UNIVERSITY OF CAMBRIDGE

Department of Computer
Science and Technology

Discovery and Ontology Matching of Financial Regulatory Information

Tim Luka Horstmann

Lucy Cavendish College

June 2024

Submitted in partial fulfillment of the requirements for the
Master of Philosophy in Advanced Computer Science

Total page count: 96

Main chapters (excluding front-matter, references and appendix): 53 pages (pp 10–62)

Word count: 14988

Methodology used to generate that word count:

```
\newcommand{\wordcount}{
  \immediate\write18{
    texcount -0 -sum=1,1,1,0,0,0,0 -inc -merge
    -q report.tex > report_words
  }
  \input{report_words}
}
% %TC:ignore comments were used to ensure that only the
% main body of the report (excluding data listings) was counted.
% The below commands were used to also include text in tables:
% %TC:group table 0 1
% %TC:group tabular 1 1
% Integrated Overleaf word count: 14391
Word count: \wordcount
```

Declaration

I, Tim Luka Horstmann of Lucy Cavendish College, being a candidate for the Master of Philosophy in Advanced Computer Science, hereby declare that this project report and the work described in it are my own work, unaided except as may be specified below, and that the project report does not contain material that has already been used to any substantial extent for a comparable purpose. In preparation of this project report I did not use text from AI-assisted platforms generating natural language answers to user queries, including but not limited to ChatGPT. I am content for my project report to be made available to the students and staff of the University.

The raw data (documents containing financial regulations) and a baseline algorithm were provided by Regulatory Genome Development LTD as a starting point for this project. Figures in this report are my own work unless specified differently in their caption.

Signed

Horstm

Date June 3, 2024

Abstract

The growing complexity and volume of financial regulations pose significant challenges for financial institutions striving to maintain compliance. Manual processing of these intricate regulations is increasingly impractical, and research has shown that contemporary natural language processing (NLP) tools struggle with the complexity of legal language.

This study addresses the above challenges by advancing the research of NLP within the growing field of regulatory technology (RegTech) to discover financial regulatory information (FRI) in legal documents and optionally match it with labels from an ontology created by the University of Cambridge Regulatory Genome Project (RGP). Specifically, we introduce the **FRI D**iscovery and **A**nnotation s**Y**stem (**FRIDAY**) — the first ever end-to-end NLP system for discovering and classifying FRI in unseen legal documents.

Using 1,149 expertly annotated documents containing anti-money laundering (AML) regulations from 75 jurisdictions around the world, this work conducts a comprehensive data analysis to define relevant FRI in legal documents. We further present a robust preprocessing algorithm that remedies discrepancies between annotated FRI extracted via optical character recognition and document content. Five novel NLP systems are introduced, including models built on existing text segmentation tools and novel combinations of modern machine learning techniques, merging elements from text segmentation, text zoning, and sentence boundary detection to discover FRI in unseen legal documents.

We evaluated ten model configurations on 6,930 pages of unseen AML regulations. Our best system, FRIDAY, operates on a token level using a pre-trained RoBERTa model optimised for text segmentation. It identifies and optionally classifies FRI with labels from the RGP ontology. FRIDAY achieves mid-80s ROUGE scores against gold-standard annotations, outperforming baseline approaches by 37%. Furthermore, it generalises well to novel domains, maintaining strong performance on an additional 50,000+ pages of unseen financial cybersecurity regulations with only a $< 5\%$ drop in performance.

FRIDAY demonstrates the potential of advanced NLP techniques in RegTech, significantly improving regulatory compliance's efficiency and accuracy. As the first system designed to identify FRI in legal documents, it advances research in legal NLP and provides a ready-to-use practical tool for various stakeholders to navigate regulatory frameworks. Future research could seek to handle broader content and optimise FRIDAY's components.

Acknowledgements

I would like to extend my deepest gratitude to the following individuals and groups who made this research project possible:

First and foremost, I am immensely grateful to my supervisor Professor Paula Buttery, for her invaluable guidance throughout this project. Her insights, support, and encouragement were instrumental in shaping the direction and outcome of this research. I would also like to extend my thanks to Philip Moore for his feedback and for bringing new perspectives to each of our meetings.

This research project would not have been possible without Henry Garner (CTO) and the entire data science team at RegGenome. Their provision of annotated data and a baseline algorithm to compare my work against provided the starting point for this project. Additionally, their guidance and advice, especially during the first few weeks of this project, were invaluable and significantly enriched my understanding and approach to the research. I would also like to thank the broader RegGenome team for their support and contributions. I look forward to our continued collaboration after this project.

I am profoundly thankful to my family for their unwavering support. Their encouragement helped me navigate challenging times and remain focused on my goals and aspirations.

Additionally, I would like to thank the Department of Computer Science and Technology at the University of Cambridge and its staff for providing the resources and support needed to work on this project. Lastly, I would like to acknowledge the German Academic Scholarship Foundation (”Studienstiftung des deutschen Volkes”) for supporting my academic studies.

Contents

List of terms and acronyms

Glossary

BIO Named entity recognition (NER) technique, where the *Beginning, Inside*, and *Outside* of a relevant unit of interest are tagged. Also referred to as *IOB*. The BIO approach is sometimes extended to other approaches such as BIOE, additionally tagging ”End/Ending” elements.

Block Continuous unit of text extracted from the page of a PDF document using the PDF extraction and manipulation library PyMuPDF. Depending on the circumstances, blocks may represent anything from single characters to entire paragraphs. In this work, concatenated blocks provide the ground truth for the textual content of a page.

RegGenome Financial regulatory services company *Regulatory Genome Development LTD* (RegGenome) — a University of Cambridge spin-out. RegGenome provided the annotated legal documents containing financial regulations as well as a baseline algorithm as a starting point for this work.

Region An exact text segment, manually identified by regulatory experts working with RegGenome, containing essential financial regulatory information (FRI) like requirements and obligations directed at entities such as individuals, businesses, and financial institutions. In this work, regions represent the ground truth for the FRI to be identified by the snippet identifier system.

Snippet A text segment identified by the snippet identifier system that contains relevant financial regulatory information (FRI). Ideally, snippets match the regions annotated by experts.

Acronyms

AI	Artificial Intelligence
AML	Anti-money Laundering
BERT	Bidirectional Encoder Representations From Transformers
Bi-LSTM	Bidirectional LSTM
BPE	Byte-Pair Encoding
CRF	Conditional Random Field
CSD3	Cambridge Service For Data Driven Discovery
DL	Deep Learning
ESG	Environmental, Social And Governance
FRI	Financial Regulatory Information
FRIDAY	**F**inancial **R**egulatory **I**nformation **D**iscovery And **A**nnotation S**Y**stem
HPC	High Performance Computing
HPO	Hyperparameter Optimisation
KDD	Knowledge Discovery In Databases
LLM	Large Language Model
LSTM	Long Short-Term Memory
ML	Machine Learning
MLM	Masked Language Model
MTL	Multitask Learning
NER	Named Entity Recognition
NLP	Natural Language Processing
NSP	Next Sentence Prediction
OCR	Optical Character Recognition
RegTech	Regulatory Technology
regex	Regular Expression
RGP	Regulatory Genome Project
RNN	Recurrent Neural Network
RoBERTa	Robustly Optimized BERT Pretraining Approach
ROUGE	Recall-Oriented Understudy For Gisting Evaluation
SBD	Sentence Boundary Detection
TS	Text Segmentation
TZ	Text Zoning

> ”With a new regulatory alert issued every 7 minutes, how do I ensure compliance?”
>
> —*Thomson Reuters report on RegTech [29]*

Chapter 1

Introduction

For decades, financial regulations have steadily increased in number and complexity [58]. The global financial crisis of 2008, much like previous crises, only accelerated this process once more and made regulatory compliance a top priority for financial institutions [60, 84]. In response, authorities worldwide implemented stricter and ever more comprehensive financial regulations such as BASEL III [6], the Markets in Financial Instruments Directive II [73] and the Dodd-Frank Act [27]. The latter alone, with its 848 pages, exemplifies only too well the extent of the complexity of financial regulation today, and recent reports indicate this trend is set to continue [81, 100].

The growing complexity of financial regulations poses significant challenges and concerns for banks and other financial institutions globally as they struggle to keep up and comply with new regulations [3]. To meet these increasing demands, financial institutions — now more than ever — require tools that are capable of automatically and accurately processing and interpreting financial regulatory information (FRI) in a scalable and efficient manner.

The advent of machine learning (ML), a subdomain of artificial intelligence (AI), and specifically natural language processing (NLP) techniques applied in the domain of law and regulatory compliance, technology known as regulatory technology (RegTech), offer promising solutions to the above problems [34, 58]. The sector is growing rapidly, and by 2026, RegTech is projected to account for 50% of global compliance budgets [60]. The immense demand for RegTech solutions highlights the wide variety of stakeholders, including financial institutions, regulatory bodies, compliance officers, and legal professionals around the world, who can benefit significantly from the automated processing of FRI.

However, the complex domain and language of financial regulations make it challenging to meet the demand for RegTech [53]. Despite the apparent benefits of RegTech and advancements in general NLP technology, there currently exists no efficient tool to automatically identify FRI in legal documents or descriptions of what constitutes ”relevant FRI” in terms of its characteristics like location, scope or content within documents. Existing tools either have entirely distinct objectives or operate in different domains, such as identifying different textual content in job advertisements or emails [39, 50].

This work seeks to fill the identified gap by addressing the following central research question: How can ML and, specifically, NLP be applied to identify and, as an extension goal, classify text segments containing relevant FRI within structured legal documents?

For clarity, we refer to these text segments containing FRI using a parallel terminology:

- *Regions*: Segments manually identified by experts, considered the ideal.
- *Snippets*: Segments automatically identified by the NLP system developed in this study. Ideally, i.e. in the case of an optimal system, snippets match the regions.

This distinction is further explained in Chapter 3.

Based on a comprehensive real-world dataset of over a thousand expertly annotated documents containing anti-money laundering regulations published in 75 jurisdictions worldwide, the primary goal of this study is to develop a novel end-to-end snippet identifier system that can automatically discover FRI in unseen legal documents. Additionally, the system aims to optionally match the identified FRI with labels from an open-source ontology created by the University of Cambridge Regulatory Genome Project (RGP) [12].

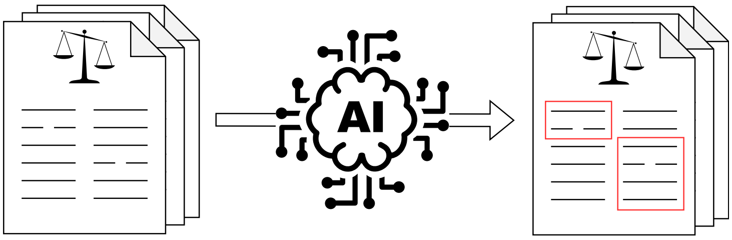


Figure 1.1: The goal of this work is to introduce a novel natural language processing (NLP) system capable of discovering financial regulatory information (FRI) in legal documents.

To the best of our knowledge, this is not only the first work to ever address this task of FRI discovery and classification but also the first to approach it based on this comprehensive dataset. The dataset for our work was provided by the financial regulatory services company *Regulatory Genome Development LTD* (RegGenome)[1].

Concretely, this study makes the following main contributions:

1. **Data analysis:** As part of a comprehensive data analysis, we describe the different textual elements contained in this work's dataset and define what constitutes the FRI to identify in the documents at hand. We synthesise the information needed for developing a snippet identifier system.
2. **Custom data pre-processing:** A robust data pre-processing algorithm that suits the idiosyncrasies of FRI is presented. This algorithm is capable of identifying erroneous annotations (i.e. regions) in the actual document content and remedying discrepancies between the two.

[1] `https://reg-genome.com/`

3. **Development and evaluation of FRIDAY**: This work introduces the **F**inancial **R**egulatory **I**nformation **D**iscovery and **A**nnotation s**Y**stem (FRIDAY). FRIDAY is a novel hybrid NLP system leveraging modern ML approaches to not only discover but also classify FRI in legal documents. As part of its development, five novel contrasting NLP models are presented. These include two models built on existing text segmentation tools and novel combinations of modern ML techniques, merging elements from text segmentation, text zoning, and sentence boundary detection. Evaluating ten different configurations of these models, the final version of FRIDAY outperforms baseline approaches from industry and research by nearly 37% with ROUGE scores of up to 0.86 against gold annotations (regions) in the test dataset. Additionally, we show that FRIDAY generalises effectively to novel domains and performs similarly well on over 50,000 pages of unseen cybersecurity regulations.

The ambition of this work extends far beyond scientific research. The project ultimately aims to contribute to facilitating global financial information sharing, increasing the productivity of compliance departments, and enhancing adherence to regulatory obligations. Furthermore, it enables deeper insights into the capabilities of modern NLP techniques in the challenging and under-researched domain of legal language and financial regulations.

The rest of this work is structured as follows: Chapter 2 presents relevant related work that informed the development of FRIDAY and provides an overview of the employed ML techniques. The subsequent main part loosely follows the traditional knowledge discovery in databases (KDD) methodology illustrated in Figure 1.2, providing a framework for the development of FRIDAY.

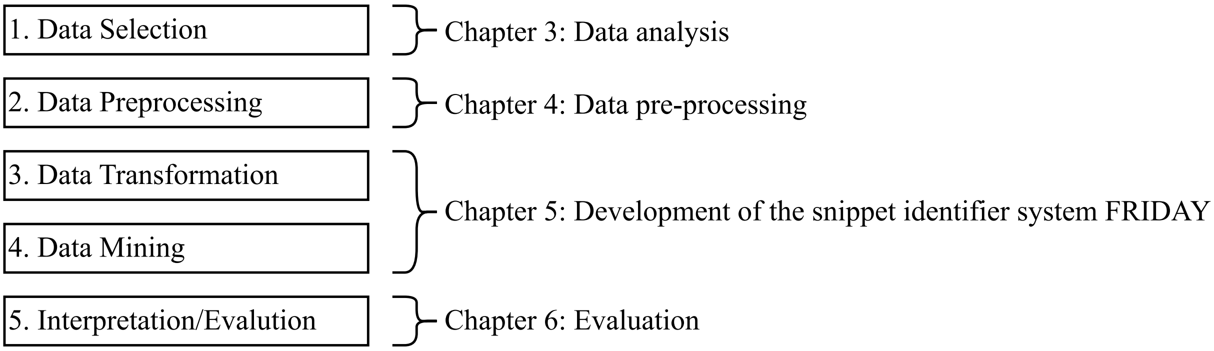

1. Data Selection
2. Data Preprocessing
3. Data Transformation
4. Data Mining
5. Interpretation/Evalution
Chapter 3: Data analysis
Chapter 4: Data pre-processing
Chapter 5: Development of the snippet identifier system FRIDAY
Chapter 6: Evaluation

Figure 1.2: The KDD process and its relevance to this report's structure. Based on [31].

Chapter 3 describes and analyses the data selected for use in this work as well as the characteristics of FRI in our dataset. Based on these insights, the custom data preprocessing algorithm is presented in Chapter 4. Chapter 5 introduces the development of the snippet identifier system FRIDAY, followed by an evaluation of its performance in Chapter 6. We conclude the report in Chapter 7 and provide directions for future work.

Chapter 2

Background and related work

2.1 NLP in the legal domain

RegTech is rapidly advancing, with the potential to revolutionise the financial industry [58]. Yet, the field is still in its infancy and faces many challenges that remain unresolved [53]. While state-of-the-art NLP models have enabled breakthrough success in many fields, they often fall short in RegTech and the legal domain in general [53].

A major reason for the deficiencies of current NLP methods in the legal domain is the unique characteristics of legal language, rendering it significantly more complex than other forms of natural language [34, 53]. Legal language is often characterised by technical, prescriptive, and sometimes multilingual language, lengthy sentences, cross-references as well as expansive and specially formatted documents [15, 34, 65, 72, 90, 109]. An additional challenge is the lack of sufficiently large annotated datasets in the legal domain, which prevents the adequate training of ML models [34]. Given the highly specialised form of language, the manual annotation process in this domain is time-consuming and expensive due to the significant expert knowledge required [15, 34, 97].

The above challenges make the application of ML techniques to legal documents substantially more difficult than other NLP tasks and are part of the reason there is only relatively little coverage — although growing rapidly — of the application of NLP to the legal domain in contemporary research [8, 34, 53, 65]. This not only underlines the particularly difficult challenges that need to be solved in RegTech but also highlights a critical research gap this work seeks to address [34, 65].

2.2 Related NLP tasks and their limitations

The intricacies of legal language pose unique challenges on several key NLP tasks, which are essential for processing legal texts effectively. The analysis of documents containing FRI to discover snippets involves elements from three prominent NLP tasks in particular: text segmentation (TS), text zoning (TZ), and sentence boundary detection (SBD). The following sections detail each task's purpose, review past approaches and discuss the limitations of these methodologies in the legal domain and the context of this work.

2.2.1 Text segmentation (TS)

TS divides text into meaningful units, usually based on topical coherence, and is an important NLP task often serving as a precursor for further downstream tasks like information retrieval, text understanding, and language modelling [4, 19, 43]. It is the task closest related to this work. TS can be categorised into two types: linear TS and hierarchical TS. Linear TS divides text into non-overlapping segments, while hierarchical TS breaks them down into subtopics [37]. Since most TS research deals with linear TS [28, 37], hierarchical TS will not be further covered here. Readers interested in hierarchical approaches to TS are referred to Yaari [107], Eisenstein [28], and Lawless and Bayomi [64].

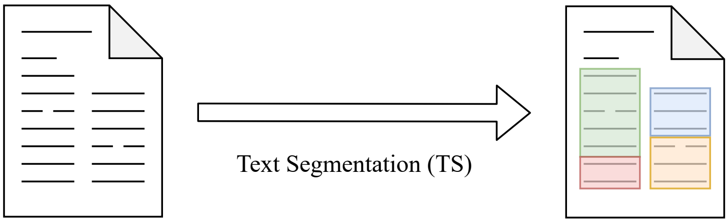


Figure 2.1: Text segmentation (TS) divides text into topic-based units.

Since Hearst's initial research on TS in 1994 [43], many approaches have been investigated. Early work mainly focused on estimating the lexical cohesion of different units through lexical features for which similarity metrics were calculated [19, 43, 52]. Similar research presented statistical models for topic modelling [7, 9, 18, 20, 26, 75, 83, 98, 105], where the task is to detect latent topics, for example through models like Latent Dirichlet Allocation [75, 83, 98], (Probabilistic) Latent Semantic Analysis [9, 20], or dynamic programming approaches [105]. Next to these unsupervised statistical-model-based approaches, further unsupervised methods using semantic relatedness graphs were proposed [37]. More recently, following trends in general NLP, scholarship started to investigate supervised models for TS employing modern deep learning (DL) techniques [1, 4, 38, 57, 69, 72]. Notably, TS usually works with sentences or even broader units like paragraphs as the elementary unit to be analysed for segmentation.

2.2.2 Text zoning (TZ)

TZ is an NLP task closely related to TS and aims at classifying text segments into predefined categories based on their content and rhetorical function. It was first introduced by Teufel [101] in 1999 as *argumentative zoning*. While Teufel analysed scientific papers by classifying the "rhetorical status of a sentence" [102], the technique of zoning was subsequently applied more widely to a variety of domains and text types such as news articles [5], job advertisements [39, 40], theatre reviews [71], emails [50, 62] and the legal domain [42]. Hence, the task is often also referred to more broadly as *text zoning* [e.g. 39, 40, 71]. Similar to TS, TZ traditionally works with sentences as the unit to classify.

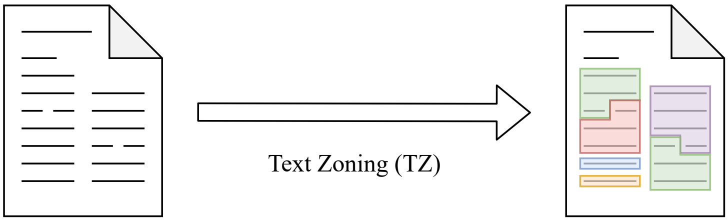


Figure 2.2: Text zoning (TZ) classifies text into categories by content and function.

Recent research has shown that sequence labelling models such as Conditional Random Fields (CRFs) [61] and Long Short-Term Memory (LSTM) networks [46], as well as their bidirectional variant Bi-LSTM [41], are generally highly effective in TZ [39, 40, 45, 50]. In both, TS and TZ, hierarchical ML approaches have been employed [e.g. 38, 50]. For example, Jardim *et al.* [50] use a sentence-level encoder in conjunction with a second-level segmentation model to identify functional zones in emails. Unlike TS, however, recent studies in TZ often also address the challenge as a token-level classification task, leveraging traditional named entity recognition (NER) techniques [e.g. 1, 39, 40]. These include the *BIO* approach where the *Beginning*, *Inside*, and *Outside* of a relevant unit of interest are tagged [63].

2.2.3 Sentence boundary detection (SBD)

Like many other NLP tasks, TS and TZ have in common that they are rarely employed in the domain of complex legal texts. Furthermore, both tasks tend to be addressed utilising sentences as the underlying textual unit. Notably, most studies assume a predefined split of documents into sentences (Figure 2.3) without detailing the methods used for this division. For an application of TS and TZ in the legal domain, however, this step of splitting text into sentences alone represents a non-trivial challenge and algorithms commonly used in other NLP domains often perform poorly on legal documents [86]. In fact, SBD in legal texts is a complex area of research in its own right [10, 86, 90, 95]. Similar to TZ, SBD also employs the BIO tagging approach to identify sentence boundaries [86].

Although SBD is not the primary focus of this work, it should not be neglected because it

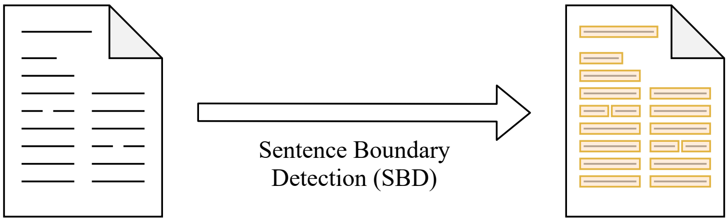


Figure 2.3: Sentence boundary detection (SBD) enables splitting text into sentences.

significantly influences the performance of any potential sentence-based snippet identifier algorithm. As one of the first steps in the model development, an incorrect sentence split can have significant negative effects on overall model performance [10, 90].

2.2.4 Summary

Consolidated overview of limitations and research gaps

Although the NLP tasks described above are highly relevant to this work, their objectives differ from the goal of identifying snippets in legal documents. Table 2.1 summarises the limitations of the presented approaches with regard to the research question at hand and suggests that a new methodology is needed for the task of identifying snippets.

Task	Limitations and gaps in related research
TS	• Usually focuses on segmenting text into larger segments based on sentences or paragraphs. As Chapter 3 will show, snippets can be shorter than this. • Existing datasets for TS are often small in size or artificial [e.g. 19, 33, 37]. • Linear TS generally classifies every part of the text as part of a segment. Snippets, however, do not need to cover a full page. (Chapter 3) • The Number of segments to be created is not always flexible and is sometimes fixed. The number of snippets, however, can vary widely. (Chapter 3) • Some approaches rely on handcrafted rules and features, which are infeasible for complex legal documents with intricate formatting. (Section 3.2.2)
TZ	• Limited recent research on TZ in the legal domain. • Limited research into the use of modern DL models.
SBD	• Significantly more challenging in the legal domain due to long, complex sentences and the intricacies of legal language [86]. • Assumptions often made in SBD approaches do not hold in the legal domain [86].

Table 2.1: Overview of limitations of the presented NLP tasks. Although TS, TZ, and SBD offer good starting points for this study, a tailored approach combining these techniques is needed.

Towards a robust snippet identifier: integrating NLP techniques

Based on the identified research gaps, the snippet identifier system presented in this work requires a tailored integration of several approaches from TS, TZ, and SBD. We aim to combine their strengths and mitigate the limitations presented in Table 2.1 to ultimately develop a robust tool for snippet identification in the legal domain. As such, previous studies that stand in between these NLP tasks, such as ”Unit Segmentation of Argumentative Texts” by Ajjour *et al.* [1], are of particular relevance for this work.

Given the substantial challenge of identifying poorly defined snippets containing FRI within complex legal documents, we define the following desirable features for the snippet identifier system based on the limitations of previous research:

I **Adaptive boundary detection:** The snippet identifier system should be flexible with respect to snippet boundaries and capable of recognising boundaries between words, sentences, paragraphs, or even pages. The system should be able to accommodate the various forms and structures of textual units found in legal documents.

II **Unrestricted snippet identification:** The system should not be constrained to classifying every part of a text or limited to a predetermined number of snippets.

III **Legal understanding:** The system should be capable of interpreting the complex format, language, and structure of legal documents.

IV **Advanced NLP techniques:** The system should go beyond handcrafted rules and features and employ state-of-the-art sequence models, offering improved accuracy.

V **Robust training datasets:** To achieve high reliability and performance, the system needs to be trained on extensive, expertly annotated data containing FRI.

2.3 Comparative analysis of ML techniques used in this work

This work mainly builds upon two ML architectures that have been successfully applied to a variety of NLP tasks relevant to this study: Long Short-Term Memory (LSTM) [46] and transformers [106]. Specifically, we use bidirectional versions of these architectures, which consider context in both directions of the input: bidirectional LSTM (Bi-LSTM) [41] and pre-trained transformer models based on the Robustly Optimized BERT Pretraining Approach (RoBERTa) [68] — given their strong performance in sequence classification tasks, including TS, TZ, and SBD [14, 40, 47, 50, 55, 66, 70, 110]. Further details on these ML techniques are provided in Appendix A.

Table 2.2 compares the LSTM and transformer-based approaches with respect to aspects critical to this study. Considering the state-of-the-art performance of these models, both architectures offer promising foundations for the development of the snippet identifier system.

Characteristic	**LSTMs** **(incl. Bi-LSTMs & CRFs)**	**Transformers** **(incl. pre-trained models)**
Processing	Sequential processing: each state is dependent on the previous.	Parallel processing: handles all inputs simultaneously.
Context Understanding	Good at capturing long-range dependencies. Bi-LSTMs can capture context in both directions.	Ability to capture wider context using attention mechanism. Advanced models can capture context in both directions.
Scalability	Less scalable due to recurrent nature.	Highly scalable due to attention mechanism.
Suitability for NLP Tasks	Strong performance in sequence prediction and tagging tasks.	State-of-the-art performance across many NLP tasks.
Recent Innovations	Integration with CRFs improves structured prediction capabilities.	Advancements in model architecture and training techniques.
Model Complexity	Relatively simpler model architecture.	More complex model architecture.
Resource Requirements	Generally lower memory and processing requirements compared to transformers.	Generally higher memory and processing requirements due to model size and parallelism.

Table 2.2: Comparative analysis of LSTMs (incl. Bi-LSTMs and CRFs) and transformers (incl. pre-trained models such as RoBERTa). Both architectures provide good foundations for a snippet identifier system, with pre-trained transformers likely performing best.

Despite the extensive application of Bi-LSTM models in related NLP tasks, we hypothesise that pre-trained transformers will outperform Bi-LSTM models in our work due to their ability to handle wider (bidirectional) contexts and their general language understanding. However, literature reveals mixed results regarding their efficacy in the legal domain. Studies by Chalkidis *et al.* [16] and Malik *et al.* [72] highlight that pre-trained transformers, despite their success in general NLP, often struggle with the specialised language and structures of legal documents. Thus, while promising, their effectiveness in processing legal texts and FRI is not guaranteed. This study, therefore, aims to critically assess and address these limitations, contributing to a deeper understanding of pre-trained transformers' applicability in the legal and regulatory domain.

Chapter 3

Data analysis

The architectural design of the desired snippet identifier must not only overcome the gaps in contemporary research outlined in Chapter 2 but also accommodate the specific characteristics of FRI in the available dataset. Hence, it is imperative to establish a profound understanding of what constitutes a ”region” or a ”snippet”.

This chapter provides a detailed overview of this study’s data, the three major textual elements specified for every page of the legal documents — **blocks, regions, and (baseline) snippets** — and the characteristics of FRI as identified by regulatory experts.

3.1 Data acquisition and document processing

The data for this study was provided by the financial regulatory services company *Regulatory Genome Development LTD (RegGenome)*[1] and consists of officially published documents containing financial regulations from various institutions worldwide. For the purpose of this work, RegGenome shared a subset of English-language documents. Based on the types of financial regulations they cover, these documents can be divided into three *themes*:

1. **AML:** Documents related to anti-money laundering regulations
2. **CYBER I:** Documents related to financial cybersecurity regulations (1st subset)
3. **CYBER II:** Documents related to financial cybersecurity regulations (2nd subset)

While the anti-money laundering (AML) documents were utilised for the development of the snippet identifier, the CYBER I and CYBER II documents were strictly isolated as test data. Given their distinct content, these documents enable an independent evaluation of the generalisation capabilities of the snippet identifier developed solely on AML documents.

[1] `https://reg-genome.com/`

3.1.1 Data acquisition pipeline

The documents from all three themes, real-world financial regulations, are generally published by official government institutions and made accessible as PDF documents through dedicated websites, such as the U.S. Federal Register[2]. Therefore, RegGenome systematically collected the documents from all three themes via web crawling. The final datasets used in this study consist of a total of 2,843 documents from 277 distinct publishers around the world, distributed as shown in Table 3.1. Figure 3.1 illustrates the origin of the AML documents, demonstrating the wide jurisdictional coverage. The geographic origins of the CYBER documents are visualised in Appendix B.1.

Theme	**#Documents**	**#Pages**	**#Jurisdictions**	**#Publishers**
AML	1,149	48,075	75	181
CYBER I	729	24,512	15	88
CYBER II	965	29,058	66	169
Total	**2,843**	**101,645**	**84**	**277**

Table 3.1: Number of *distinct* documents, pages, jurisdictions, and publishers across this study's datasets. While the comprehensive dataset ensures robust model training and evaluation, its diverse character makes the development of the snippet identifier system significantly more challenging.

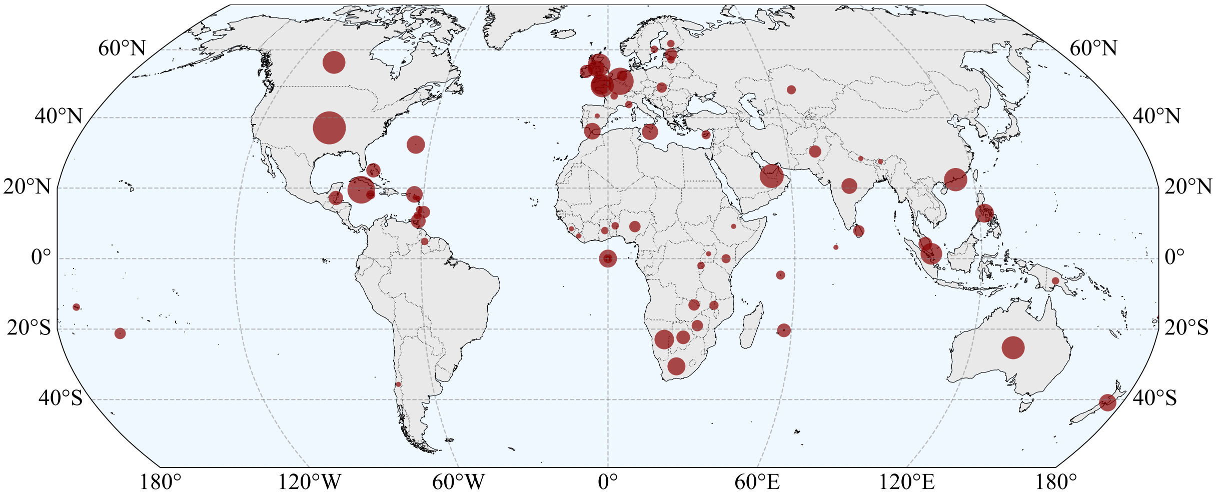


Figure 3.1: Origin of this work's documents containing AML regulations. The circle's diameter indicates the number of documents originating from each country. Data from the European Union is aggregated with that of Belgium.

The dataset's diversity allows the model to learn from a wide range of regulatory document formats and formulations, which should enhance the snippet identifier's generalisation capabilities to understand financial regulations regardless of their origin. However, this diversity also introduces two significant challenges:

1. **Document imbalance:** The dataset is heavily imbalanced, with some jurisdictions, like the US and EU, known for publishing significantly more financial reg-

[2] `https://www.federalregister.gov/`

ulations than smaller jurisdictions, such as Rwanda and Zambia (see Table B.1). This imbalance may negatively affect the model's performance on documents from less-represented jurisdictions.

2. **Significant structural differences:** Research shows that financial regulations often vary greatly in structure/format and content from jurisdiction to jurisdiction [21, 78]. Anecdotally, data science experts at RegGenome highlight that generalising to these different structures is one of their biggest challenges. Excessive differences between regulations can prevent the model from learning and generalising the essential knowledge needed for identifying snippets across various jurisdictions.

Figure 3.2 underscores the first of the above challenges: a select few publishers are primarily responsible for a significant portion of regulatory documents. These include the Financial Crimes Enforcement Network[3] and Treasury[4] from the United States, the Financial Reporting Authority from the Cayman Islands[5] and the European Parliament[6]. Together, these institutions account for nearly 30% of all AML documents.

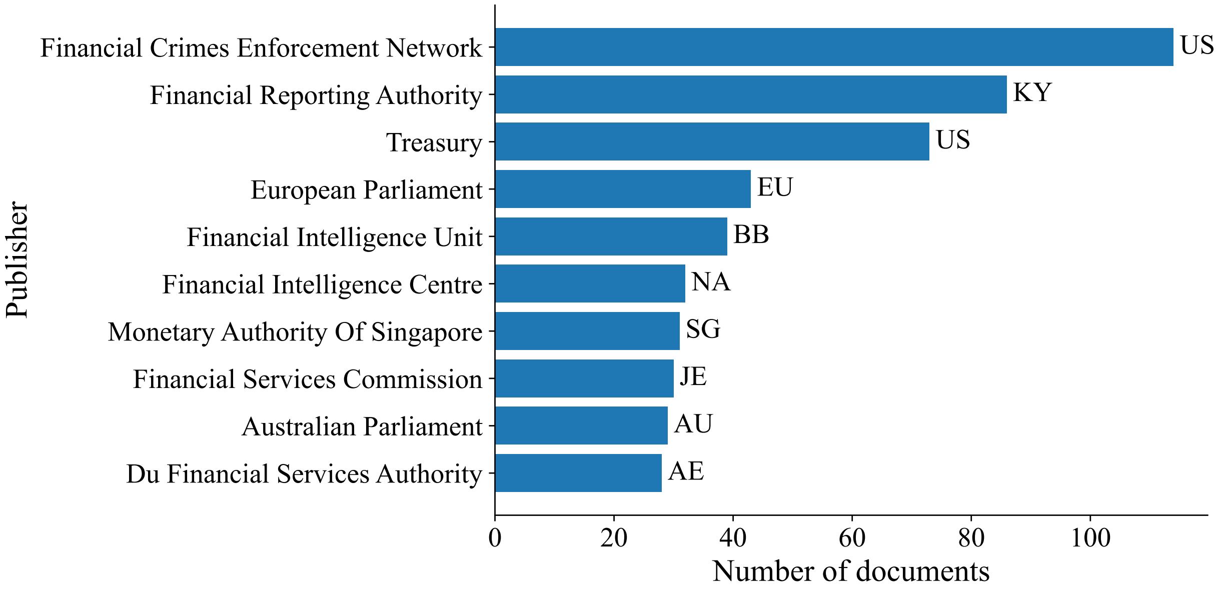


Figure 3.2: Top ten publishers by number of documents in the AML theme (jurisdictions indicated as ALPHA-2 codes (ISO 3166)[7]). The figure highlights the dataset's imbalance, with a few publishers from major jurisdictions contributing nearly 30% of all AML documents.

We provide additional insights into the diverse nature of this study's dataset in Appendix B. Section B.2 shows the largest publishers in the CYBER themes, while Table B.1 provides an overview of the origin and quantitative distribution of all documents across all themes.

[3] https://www.fincen.gov/

[4] https://home.treasury.gov/

[5] https://fra.gov.ky/

[6] https://www.europarl.europa.eu/portal/en

[7] https://www.iso.org/obp/ui/#search

3.1.2 Document preparation

All documents for this study were provided by RegGenome in the XML file format, which represents the starting point for this work. Next to relevant metadata, such as the title, id, publisher, and country of origin of each document, each page (element) of a document consists of two central structures containing the textual content of a page: **blocks** and **regions**. Figure 3.3 illustrates the document preparation pipeline on a page level and details how RegGenome obtained blocks and regions.

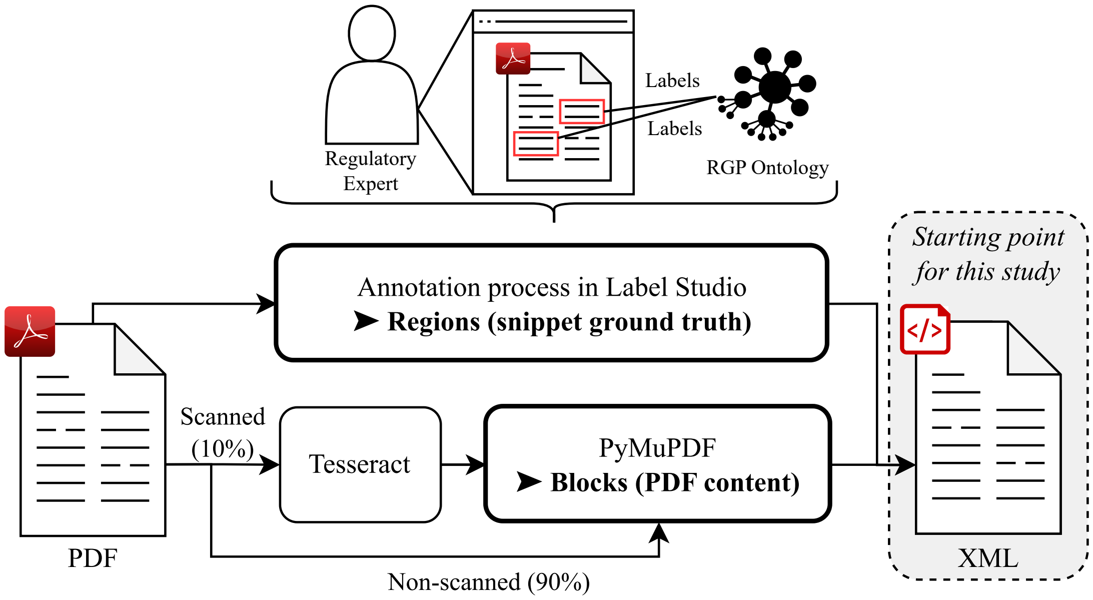


Figure 3.3: Preparation steps for each page of a document as executed by RegGenome. Scraped PDF documents were made machine-readable through a combination of the *Tesseract* and *PyMuPDF* libraries. Additionally, regulatory experts working with RegGenome manually identified relevant FRI on these pages.

Blocks

To turn the raw PDF documents into machine-readable content, all documents were processed through the PDF extraction and manipulation library *PyMuPDF*[8]. In the case of scanned documents (usually approximately 10 % of all documents[9]) the document contents were first made extractable via the optical character recognition (OCR) engine *Tesseract*[10].

PyMuPDF implements heuristic algorithms to pre-structure the pages of PDF documents into so-called *blocks* [93]. These blocks correspond to structural elements that PyMuPDF identified during the extraction process and can represent any contiguous grouping of text that the library identified based on the spatial layout and other formatting cues within the document. Therefore, these blocks can correspond to anything from single characters like page numbers to paragraphs or even entire pages. We obtained the full text of a page through simple concatenation of all of its blocks.

[8] `https://pymupdf.readthedocs.io/en/latest/index.html`

[9] Estimate provided by RegGenome.

[10] `https://github.com/tesseract-ocr/tessdoc`

Regions

RegGenome collaborates with regulatory experts who have manually reviewed each page of the legal documents used in this work to identify FRI. Using the data labelling platform *Label Studio*[11], the human annotators drew bounding boxes around the relevant passages containing FRI. These annotated sections of a page of a document are referred to as *regions* and represent the "gold snippet", i.e. the ground truth for the development of the snippet identifier system.

In Chapter 1 we introduced *regions* and *snippets*. For clarity, we reiterate their definition as follows:

Definition

A *region* is a text segment within legal documents that contains relevant financial regulatory information (FRI), such as requirements and obligations directed at entities like individuals, businesses, and financial institutions. These regions are manually identified by regulatory experts.

A *snippet* is the text segment predicted by the snippet identifier system to match the regions. Ideally, snippets and regions should be identical, with regions being the expert-identified ideal and snippets being the system-generated approximations.

Exemplary regions are shown in Figure 4.1 and Figure B.7.

Figure 3.4 depicts the typical content of regions. Due to their regulatory character, regions primarily address the parties concerned ("person", "customer", "business", "(financial) institution") and contain words defining the regulation/obligation ("may", "shall", "must", ...). The character "b" and "c" belonging to the 30 most common "words" across all regions in the AML theme emphasises the frequent usage of textual markers to structure legal texts. We note that "a" does not appear in this list due to it also being a removed stopword. We do not remove stopwords in the data used for training.

Annotators also tagged each region with a *detailed label* to define the regulation and the types of requirements it describes. The detailed labels are based on regulatory standards and originate from the University of Cambridge Regulatory Genome Project (RGP)'s ontology [12]. The RGP developed this information structure to standardise and globally compare regulatory content [12]. The detailed labels used to annotate the regions in this work's data usually consist of up to four *hierarchical levels*, separated by hyphens, that correspond to increasing levels of detail. While the first-level label ("Level 0") defines the theme (i.e. "aml" for all AML documents), the second label describes the broad topic of the regulation, with subsequent levels adding more detail. Appendix B.3 provides an overview of these levels and their corresponding distributions. An exemplary detailed label could be *aml-customeridentification-verification-individuals*.

[11] `https://labelstud.io/`

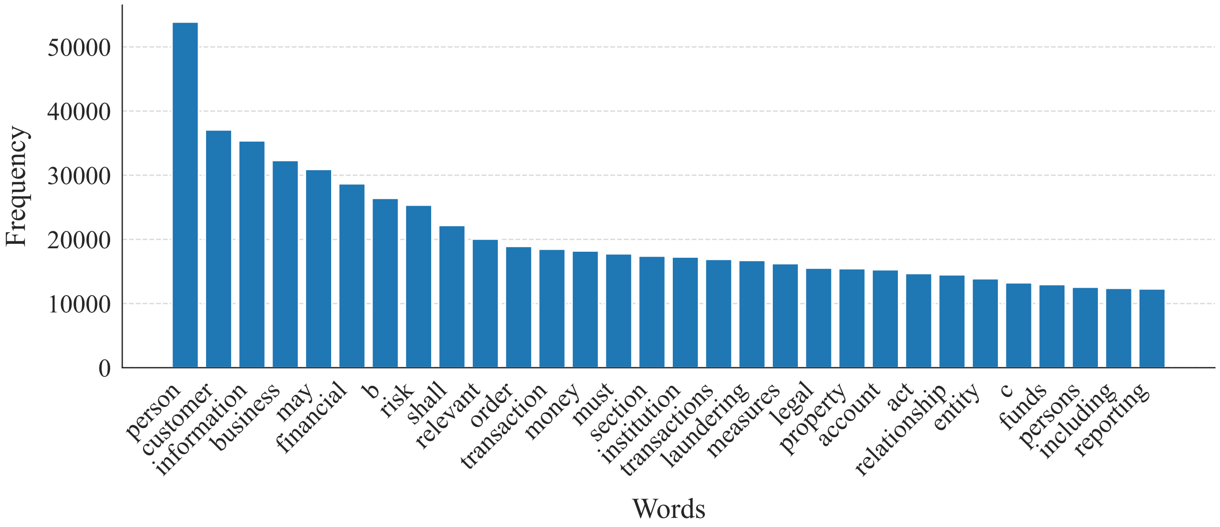


Figure 3.4: 30 most frequent words (excl. stopwords) across all regions in the AML theme. Regions usually contain common terminology, used to define regulations for different parties. They are typically presented in a structured format.

3.2 Snippet creation process

The complex formatting and content of legal documents make the identification and creation of snippets for financial regulations an extremely challenging task (Chapter 2). Yet, algorithms based on simple heuristics, such as regular expressions (regex), are still used to address such tasks. While these algorithms offer advantages in terms of speed and ease of implementation, they are typically less accurate than comparable modern ML methods. As a baseline for this work, RegGenome provided such an algorithm that was used with the goal of segmenting the pages of legal documents into region-like sections.

3.2.1 The baseline model: RegGenome's snippeting algorithm

RegGenome's snippeting algorithm takes the page of a document as input and returns its snippets as output. This process can be roughly divided into two steps (pseudocode for the algorithm is provided in Algorithm 1):

1. **Identify the "character type" of a page.** Using pre-compiled regular expressions, the algorithm attempts to either locate a table of contents on the page, or identify the textual components defining the structure of the page, such as different types of numbering or labelling formats like "1)", "1.1", and "Article I".

2. **Split the page into snippets.** If a character type is identified, it is used to divide the page into the sections defined by the markers of this type. If no character type was found, the algorithm simply segments the page into elements of similar size.

As the above snippeting algorithm was employed to segment pages into textual units similar to regions, this algorithm, coupled with a custom post-processing pipeline to choose the snippet most likely representing a region, provides a good baseline model for this study. Therefore, as part of the document pre-processing process, the text of each page of all documents used in this work was manually passed through the snippeting algorithm to obtain the **baseline snippets** for each page.

3.2.2 Limitations of the snippeting algorithm

Despite its computational efficiency, the snippeting algorithm has three major limitations:

1. **Legacy code:** The algorithm contains unreachable code segments due to conditions that are impossible to meet, thus rendering these segments non-functional.
2. **Heuristic rules and regex:** The algorithm relies entirely on heuristic rules and highly specific regex patterns, which are rarely triggered due to the complex formatting and origin-dependent differences of legal documents. Figure 3.5 illustrates this fact by highlighting that the algorithm failed to identify the character type of the pages in the AML dataset in more than 60 % of all cases. The majority of its pages were simply split by size.
3. **Size-based splitting:** When simply splitting pages by size, the algorithm employs hard character length boundaries, which do not account for the different formatting of documents from different jurisdictions.

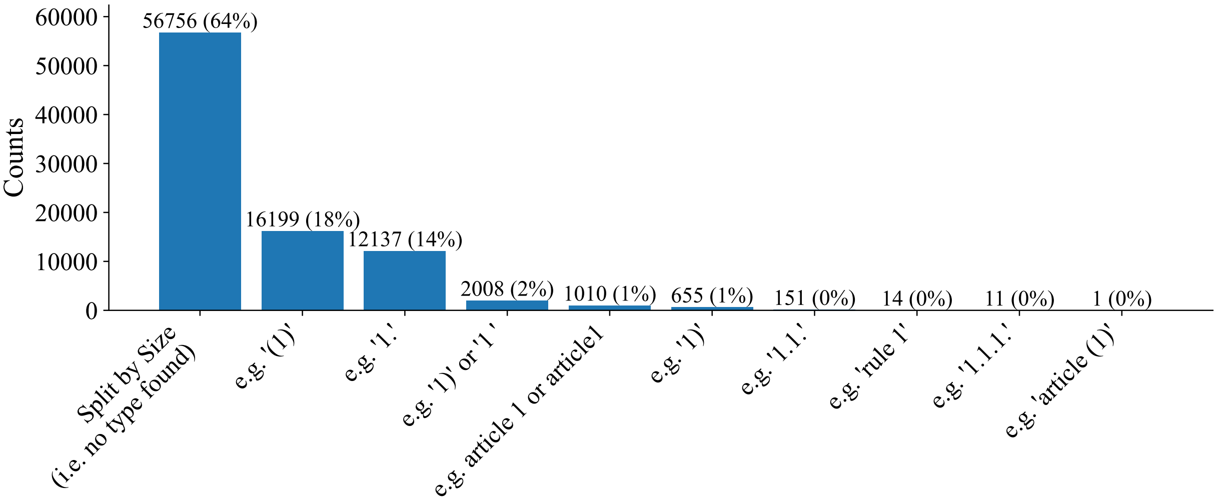


Figure 3.5: Total number of snippets identified through the snippeting algorithm by character type across all AML documents. The algorithm failed to identify character types in over 60% of cases, defaulting to a basic size-based split instead.

In summary, the snippeting algorithm is too static to handle the diverse legal documents from various jurisdictions. Additionally, it lacks any logic to decide whether a separated text segment constitutes a relevant snippet. Therefore, while the snippeting algorithm represents a basic TS algorithm, used in the same use case as the snippet identifier system, it requires post-processing and additional statistical modelling. These additions are necessary to identify which snippets — which themselves may be inaccurate — could represent regions. Our study aims to surpass the performance of the snippeting algorithm as a baseline.

3.3 Comparative Analysis

For this project, to aid the comparison of textual elements of all 2,843 unique documents in the AML dataset, we developed a custom comparison tool. With this tool, a user can interactively load and display a PDF page of any document in this study's datasets. On this page, all regions, if present, including their detailed label(s), are marked with a red bounding box as drawn by the human annotator. Furthermore, the tool offers a textual comparison of the full-page text, blocks, baseline snippets created by the snippeting algorithm, and regions. An example is shown in Figure B.7.

As part of the data analysis, the characteristics of all three textual elements of a page — **blocks, regions, and baseline snippets** — were quantitatively analysed. Table 3.2 presents relevant statistics for these elements in the AML dataset, with token counts obtained using RoBERTa's Byte-Pair Encoding (BPE) tokenizer[12]. Figure B.8 and Figure B.9 show the detailed structure and distribution of these elements.

Element	Total number	Averages	Minimum # Tokens	Maximum # Tokens
Documents	1,149	–	–	–
Pages	48,075	• Pages per document: 41.84 • Tokens per page: 662.40 • Pages with Regions 23,894	0	3,007
Blocks	163,498	• Blocks per document: 142.30 • Blocks per page: 3.40 • Tokens per block: 194.77	0	3,007
Regions	43,053	• Regions per document: 37.47 • Regions per page: 0.90 • Tokens per region: 234.55	3	1,985
Baseline Snippets	88,942	• Snippets per document: 77.41 • Snippets per page: 1.85 • Tokens per snippet: 358.33	0	2,029

Table 3.2: Overview of key metrics for all textual elements of the AML data set. On average, baseline snippets encapsulate more content than regions, while blocks are more granular than regions. Improved segmentation methods are needed to better approximate the regions.

The quantitative analysis allows insight into numerous properties of the dataset that are crucial for developing a snippet identifier system.

[12] `https://huggingface.co/docs/transformers/en/model_doc/roberta#transformers.RobertaTokenizerFast`

Firstly, it emphasises the different average granularities of the elements, which can be coarsely defined as *Baseline Snippets* > *Regions* > *Blocks* in terms of size. The snippeting algorithm usually creates segments larger than regions, while blocks tend to be smaller or similar in size to regions. Secondly, pages comprise around 662 tokens on average, which exceeds the input size limit of many transformer-based models such as Bidirectional Encoder Representations from Transformers (BERT) and RoBERTa [25, 68]. Thirdly, most element distributions are strongly left-skewed (Appendix B.3), with a considerable portion of elements falling into smaller value ranges. There are some significant outliers, such as pages with 18 regions or up to 3,007 tokens of content. Finally, our analysis shows that regions and, thus, the snippets to be identified, can range in size from parts of a sentence to a full page.

3.4 Implications for the snippet identifier system

This chapter has shown that the snippets we aim to identify are the approximation to the regions marked by human regulatory experts in legal documents. Therefore, it can be formalised that a *perfect* snippet identifier system would be capable of identifying these regions on unseen pages, effectively replicating the task of human experts.

Furthermore, it has become evident that RegGenome's snippeting algorithm (Section 3.2), while functional, lacks the capabilities to accurately identify snippets in regulatory text. This further motivates the work at hand and underscores the necessity for a more flexible and accurate replacement system.

Extending the list of desirable features

Based on this chapter's findings, the list of desirable features for the snippet identifier system outlined in Section 2.2.4 can be extended as follows:

VI Page-level snippet identification: The system should work on a page level to replicate the job of regulatory experts who annotate documents page by page. This also enables the model to work with inputs close to the 512-token limit imposed by most modern transformer-based ML models. Still, the snippet identifier system must take into account that the average page length, for example, in the AML theme, exceeds this limit. An adapted pre-processing approach is required.

VII Flexibility in snippet size: The system should dynamically determine the number and size of snippets per page (0 to n), as snippet sizes can vary greatly.

VIII Broad regulatory understanding: The system must accommodate diverse document formats from nearly a hundred different jurisdictions worldwide.

Similarity between blocks and regions

The data analysis and manual observations of this study's data indicate that the blocks generated by PyMuPDF oftentimes appear fundamentally similar to regions.

To test this hypothesis, all regions of the AML dataset were extracted and page-wise compared to the page's blocks with the highest *Jaccard similarity* (see below) to the region. We evaluated these block-region pairs by calculating metrics commonly used in NLP research to assess the similarity of two texts, as follows, with R_{tokens} and S_{tokens} being multisets containing all tokens of the region and snippet respectively:

- **Jaccard similarity: measures the similarity between two sets.** The Jaccard similarity is frequently used in related work to evaluate the performance of NLP systems in creating text similar to a reference text [24, 30, 103]. We calculated the metric as

$$\text{Jaccard Similarity} = \frac{|R_{\text{tokens}} \cap S_{\text{tokens}}|}{|R_{\text{tokens}} \cup S_{\text{tokens}}|} = 1 - \text{Jaccard Distance} \tag{3.1}$$

  utilising the NLTK library[13]. Values were averaged across all snippet-region pairs.

- **Precision, Recall, and F1-score: commonly used metrics to assess model performance in classification tasks.** Instead of averaging, *true positives*, *false positives*, and *false negatives* were collected across all pairs and precision, recall, and F1-score were calculated on a global level (i.e. across all matches) [24]:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.2}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.3}$$

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

  where True Positives $= |R_{\text{tokens}} \cap S_{\text{tokens}}|$, False Positives $= |S_{\text{tokens}}| -$ True Positives, and False Negatives $= |R_{\text{tokens}}| -$ True Positives.

- **Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score: measures the quality of machine-generated text through comparison with human-generated reference text.** Developed by Lin [67], ROUGE is widely used in evaluating NLP tasks (e.g. natural language generation, summarisation, translation) by comparing generated text against human references, making it highly suitable to evaluate the quality of snippets against regions [30, 85]. Furthermore, its recall-oriented character is preferable for this study as capturing false positives (mistakenly marking text as FRI) is preferred over missing relevant FRI. We calculated

[13] https://www.nltk.org/api/nltk.metrics.distance.html#nltk.metrics.distance.jaccard_distance

ROUGE scores as follows:

$$\text{ROUGE-N F1-Score} = \frac{2 \cdot \text{Precision}_\text{N} \cdot \text{Recall}_\text{N}}{\text{Precision}_\text{N} + \text{Recall}_\text{N}} \tag{3.5}$$

Where:

$$\text{Precision}_\text{N} = \frac{\text{Overlap}(N)}{\text{Total}(N)_\text{candidate}}, \quad \text{Recall}_\text{N} = \frac{\text{Overlap}(N)}{\text{Total}(N)_\text{reference}} \tag{3.6}$$

Concretely, scores were calculated by comparing the N-grams (i.e. contiguous sequences of N words) of both texts using the *rouge* library[14]. For this study, the F1-score for ROUGE-1 (unigram overlap), ROUGE-2 (bigram overlap), and ROUGE-L (longest common subsequence overlap) were assessed for each model. Pairwise obtained values were averaged across all samples.

The average scores across all block-region-pairs are presented in Table 3.3 and indicate a significant similarity and overlap between the two elements. This confirms the hypothesis of close relatedness between blocks and regions, providing relevant information for the development of the snippet identifier system (Chapter 5).

Jaccard Similarity	**ROUGE-L-F**	**Precision**	**Recall**	**F1-score**
80.81%	87.45%	86.13%	81.83%	83.93%

Table 3.3: Metrics on the similarity between blocks (PyMuPDF) and regions (ground truth). Blocks can often closely match regions, which indicates their potential for accurate snippet identification.

The high similarity can be attributed to the fact that PyMuPDF creates blocks primarily based on spatial information [93]. Although these blocks are not always precise due to the simplicity of the heuristic algorithms utilised by the library, they usually correspond to isolated sentences, enumerations, paragraphs, or similar textual blocks. In many cases, this approach corresponds closely to that of human experts, who also usually annotate regions taking into account the spatial conditions of text units on a page in order to enclose self-containing units with (rectangular) bounding boxes.

[14] `https://pypi.org/project/rouge/`

Chapter 4

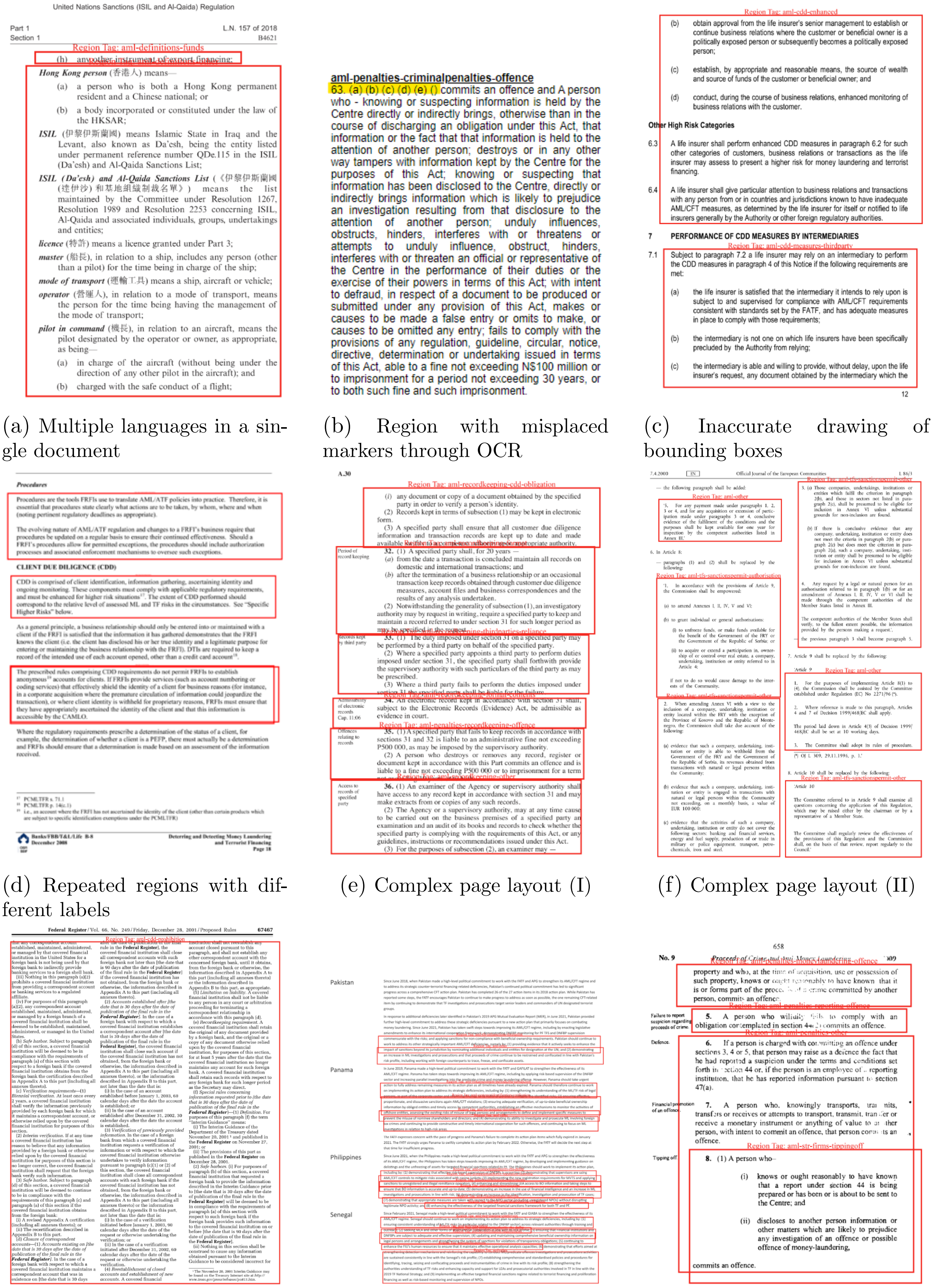
Data pre-processing

4.1 Motivation

The previous chapter demonstrated that each page of the dataset builds upon two different sources of textual content. Regions provide the “ground truth snippets” as the output of the annotation process for training the snippet identifier. However, these regions rely on OCR to convert annotated text back into a machine-readable format. As this OCR process effectively attempts to capture the text marked by the annotators by reading it from an image of a PDF page, this process can introduce errors due to image quality or font recognition. In contrast, blocks provide the ”ground truth text” through direct text extraction from PDF documents, resulting in a more accurate machine-readable collection of the page text compared to the OCR process. Consequently, the texts of regions often do not precisely align with the corresponding text from blocks, even though they are derived from the same page and should, in theory, contain identical text.

The difficulty of using OCR to accurately capture the text of regions is exacerbated even further by the high complexity of the legal documents analysed in this study. Figure 4.1 illustrates examples of highly complex pages and annotation deficiencies in the AML dataset, underscoring not only the immense challenges of textual discrepancies but also the intricate documents that need to be handled in this work.

The mismatch between the captured text of blocks and regions prevents a straightforward matching between the two. Consequently, it is impossible to straightforwardly identify which text elements of a page truly represent a region. Therefore, a robust strategy is needed to locate regions within the page text. Given the size of this study’s datasets containing over 100,000 pages, a manual correction of the OCR-identified region texts is not viable. Hence, a custom pre-processing algorithm is required to identify regions within pages while accounting for mismatches caused by OCR.



(a) Multiple languages in a single document

(b) Region with misplaced markers through OCR

(c) Inaccurate drawing of bounding boxes

(d) Repeated regions with different labels

(e) Complex page layout (I)

(f) Complex page layout (II)

(g) Large amount of content in a single region

(h) Significant number of annotations on one page

(i) Example of scanned page with hard to identify content

Figure 4.1: Examples of highly complex pages and annotation errors in the AML dataset. The examples illustrate the diverse document formatting and annotation challenges across different jurisdictions. The dataset poses significant challenges for developing a robust snippet identifier system.

4.2 The custom pre-processing algorithm

This section details the components of the custom pre-processing algorithm developed to identify regions in pages.

The main idea of the pre-processing algorithm is to precisely locate the beginning and end of a region within a page. Once these two indices have been identified, the text between these two markers, as given by the concatenated blocks' contents, can be extracted as the "correct" region. To enable this type of matching and index a page, we first employ the RoBERTa BPE tokenizer for consistent tokenization of both blocks and regions. As shown in the example below, the tokenizer splits text into subword tokens, which allows for the matching of text on a level more granular than words:

Exemplary input text:

```
Assess anti-money-laundering (AML) risks within the organization's portfolio.
```

Tokenized output:

```
['Ass', 'ess', 'Ġanti', '-', 'money', '-', 'l', 'aundering', 'Ġ(', 'AM', 'L', ')', ...]
```

To identify a region within a page, the pre-processing algorithm follows a two-step approach:

1. Find an initial approximate match for the region (i.e. "get close enough").
2. Locate the exact beginning and end of the region around the initial match by adjusting the indices and possibly refining the region content.

4.2.1 Initial approximate match estimation

Given a tokenized page P and region R as input, the algorithm initially attempts to find an exact match for the first $min(50, |R|)$ tokens of the region. This approach facilitates fast and efficient identification of the region in cases where OCR scans are accurate. Sliding a window of the region tokens across all indices of P, a region is considered identified if the following condition is met:

$$\sum_{i=1}^{\min(|r_c|,|p_c|)} \mathbb{1}(r_c[i] \neq p_c[i]) + ||r_c| - |p_c|| \leq \text{allowed\_diff} \tag{4.1}$$

Here r_c and p_c refer to the cleaned strings of the region and the page, respectively, that are being compared at each character position i. Cleaning ensures that characters highly prone to OCR misrecognition (e.g. ""*°") are not taken into account, while the *allowed_diff* parameter permits minor differences stemming from OCR errors. By default, *allowed_diff* is set to 2, ensuring that only practically exact matches are considered valid.

If no exact match can be found, the approximate location of R in P is identified as follows:

Definition 1. *We define the n-gram set for a sequence S and an integer n as:*

$$\mathit{Ngrams}(S, n) = \{(S[i], S[i+1], \ldots, S[i+n-1]) \mid 0 \leq i < \mathit{len}(S) - n + 1\} \tag{4.2}$$

Definition 2. *We define the overlap score between two n-gram sets A and B as:*

$$\mathit{Overlap}(A, B) = \frac{|A \cap B|}{|B|} \tag{4.3}$$

To find the approximate location of R within P, a window w of size $|R|$ is slid over P for each start index i_{start} in $[0, |P| - |R|]$. The best match index i^*_{start} is the index with the highest overlap score exceeding a minimum threshold T_{min} (default: $T_{min} = 0.2$):

$$\begin{aligned} i^*_{start} = \arg \max_{0 \leq i \leq |P|-|R|} & (\text{Overlap}(\text{Ngrams}(P[i : i + w], n), \text{Ngrams}(R, n))) \\ & \text{where the overlap score exceeds threshold } T_{min} \end{aligned} \tag{4.4}$$

By default, we use $n = 5$ and $n = 2$ if $|R| \leq 16$.

The results of the initial approximate match estimation define the search space for the refinement process that follows. If no approximate match or multiple matches for the region were found within the page, the refinement process, which is more precise but also computationally significantly more expensive, is applied to the entire page. Otherwise, the refined search process will consider up to 60 tokens before and 60 tokens after the estimated region's tokens on the page.

4.2.2 Match refinement

The match refinement procedure can be divided into two similar steps: **adjusting the start index** and **adjusting the end index**.

Adjusting the start index

1. Identify the region start. To adjust the start index, the pre-processing algorithm iterates through the search space $s_{start} = \{max(0, i^*_{start} - span), min(i^*_{start} + span, |P|)\}$, with $span = 60$. For each tested index $i \in s_{start}$, the tokens $p_{w_k} = P[i : i + w_k]$, with $w_0 = 64$, are extracted and transformed into a single string. Characters likely to cause OCR mismatches are removed. Subsequently, if $|p_{w_k}| \geq 8$, the algorithm tests if p_{w_k} starts with $R[iteration \times 2 : |R|]$, $iteration = 0$, or vice-versa. If no match is found in the search space, $iteration$ is incremented, and the procedure is repeated until $iteration = 16$. This approach allows an algorithmic estimation of the closest match of the region in the page text while taking into account that the beginning of a region might be flawed itself. If no match is found — even after 16 iterations — the window is re-calculated as $w_k = \frac{w}{2^k}$ with $k = k + 1$, and the full process is repeated until $k = 3$. By dynamically adjusting the number of characters to match downwards, the algorithm increases the likelihood of finding a match. If this method is ultimately unable to identify the start index of a region inside the page text, the match refinement process stops prematurely. In this case, the original region text, as extracted via OCR, is kept.

2. Refining the region beginning. Once the precise start index of a region is located, the pre-processing algorithm determines whether this region's beginning can and should be further refined in a linguistically sensible manner. For example, if an enumeration marker such as "x)" is in close proximity to the initially identified start index, this marker is set as the new beginning of the region. Alternatively, the algorithm attempts to determine the beginning of a previous sentence or other manually defined "region starters" as a new region beginning if the newly identified region text does not already commence with an appropriate character. Although this process of optionally further refining the start of a region mostly relies on manually defined simple heuristics and carries the risk of introducing noise, its conservative implementation is expected to overall aid in correcting OCR mismatches. This process aims to help restore the original intent of the human annotator.

Adjusting and refining the end index

In contrast to the start index adjustment, the pre-processing algorithm first modifies the search space for the region end index before adjusting it. This step is necessary to account for potential special whitespace and error tokens that might strongly inflate the number of region tokens captured by OCR. Hence, the search space for the region end index is defined as:

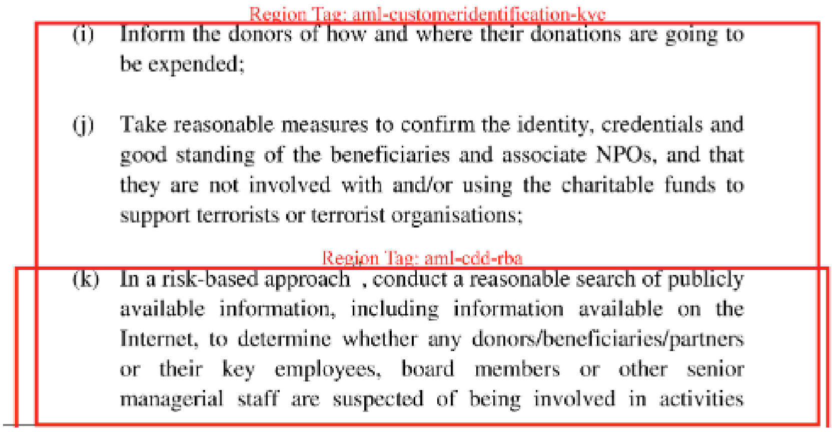
$$s_{end} = \{\max(0, i^*_{start}), \min(i^*_{start} + |R| + \text{span} + \#\text{whitespaces} + \#\text{error\_tokens}, |P|)\} \quad (4.5)$$

Based on this search space, the optimal end index of a region i^*_{end} is identified by mirroring the start index identification approach in the opposite direction. If no precise identification of a region's end index is possible, the end index remains $i^*_{end} = min(i^*_{start} + |R|, |P| - 1)$ by default. Similar to the start index refinement, the end token of a region is moved to the next sentence ending or enumeration marker if the identified region does not already end with a suitable character such as ".".

4.2.3 Practical effects

The custom pre-processing algorithm can be used to obtain the refined region text, its tokens, as well as the start and end index of the refined region within the corresponding full text of a page. Table 4.1 provides an example of the effect of the algorithm on the textual content of the regions. The refined region correctly represents the ground truth annotation visualised in the "annotated page screenshot".

Annotated page screenshot (regulatory expert)



Region Tag: aml-customeridentification-kyc

(i) Inform the donors of how and where their donations are going to be expended;

(j) Take reasonable measures to confirm the identity, credentials and good standing of the beneficiaries and associate NPOs, and that they are not involved with and/or using the charitable funds to support terrorists or terrorist organisations;

Region Tag: aml-cdd-rba

(k) In a risk-based approach , conduct a reasonable search of publicly available information, including information available on the Internet, to determine whether any donors/beneficiaries/partners or their key employees, board members or other senior managerial staff are suspected of being involved in activities

Region (OCR via Label Studio)	**Refined region** (after pre-processing)
(i) G) (k) Inform the donors of how and where their donations are going to be expended; Take reasonable measures to confirm the identity, credentials and good standing of the beneficiaries and associate NPOs, and that they are not involved with and/or using the charitable funds to support terrorists or terrorist organisations; In a risk-based approach , conduct a reasonable search of publicly available information, including information available on the Internet, to determine whether any donors/beneficiaries/partners or their key employees, board members or other senior managerial staff are suspected of being involved in activities	(i) Inform the donors of how and where their donations are going to be expended; (j) Take reasonable measures to confirm the identity, credentials and good standing of the beneficiaries and associate NPOs, and that they are not involved with and/or using the charitable funds to support terrorists or terrorist organisations; (k) In a risk-based approach13, conduct a reasonable search of publicly available information, including information available on the Internet, to determine whether any donors/beneficiaries/partners or their key employees, board members or other senior managerial staff are suspected of being involved in activities

Table 4.1: Exemplary comparison of the original region text extracted via OCR and the refined region created by the pre-processing algorithm. Discrepancies are highlighted in yellow. The refined region better matches the annotation. Text from [76].

The data pre-processing algorithm was used to match the OCR-extracted text of all regions with the page texts extracted via PyMuPDF. In the following sections, references to ”regions” or ”snippets” therefore generally imply the use or identification of refined regions. Out of the 58,157 regions contained in this work's dataset (across all themes), 32 regions could not be matched and refined within the PyMuPDF-extracted page texts. This corresponds to an error rate of the prep-processing algorithm of 0.055%.

Chapter 5

Development of the snippet identifier system FRIDAY

Building on insights from the previous chapters on the state of the art in NLP tasks related to this study and the characteristics of the data from RegGenome, this chapter introduces the methodology for developing the **F**inancial **R**egulatory **I**nformation **D**iscovery and **A**nnotation s**Y**stem (FRIDAY).

First, it is crucial to define the textual underlying unit for the model. Research shows that it is by no means certain at which level — token, sentence, or larger segment — the NLP model should operate [e.g. 4, 38, 72]. Therefore, this chapter introduces four innovative snippet identification methods, systematically exploring different options in a coarse-to-fine strategy. All models follow a page-wise prediction strategy to replicate the annotation process conducted by human regulatory experts.

Two models are based on fundamental TS approaches and work with larger segments (Section 5.1), while the sentence-level (Section 5.2) and token-level (Section 5.3) approaches are more granular and introduce more sophisticated NLP models that incorporate components from various NLP tasks. The TS-based models were limited to the main objective of this work, the discovery of snippets, while the token-level and sentence-level models were trained to also classify snippets.

The AML datasets for each model are divided into training, validation, and test sets at the document level, managed through a centralised JSON file to ensure consistency and objective performance comparisons. Following best practices, the training-validation-test split was independently established as a stratified 70-15-15 split, considering the number of pages in each document to ensure the split does not become skewed by disproportionately large or small documents. This results in 804 documents (34,404 pages) for training, 172 documents (6,741 pages) for validation, and 173 documents (6,930 pages) for testing.

5.1 Text segmentation-based approaches

The TS-based approaches to identifying snippets rest on the hypothesis that existing TS methods are capable of segmenting pages of regulatory documents into segments that are similar enough to regions. Based on these segments, a classifier can then be trained to predict whether such a segment should be considered a region or not.

The following sections present two approaches to obtaining such text segments that could form the basis for a downstream snippet identifier (Section 5.1.3).

5.1.1 GraphSeg-based TS

GraphSeg, introduced by Glavaš *et al.* [37], is an unsupervised graph-based TS model that uses word embeddings and semantic relatedness to merge sentences into segments. Due to its unsupervised nature, the model can be applied to this work's texts without additional dedicated training. As GraphSeg is used as a baseline to compare with state-of-the-art TS approaches in related research [4, 38], we employ the GraphSeg-based model as an additional baseline next to the snippeting algorithm.

For this study, the Java implementation of the GraphSeg model[1] was used to page-wise segment this study's data, providing the first input option for the snippet identifier model based on pre-determined text segments.

5.1.2 Block-based TS

Section 3.4 demonstrated that blocks generally align well with the format and location of regions. Therefore, blocks, i.e. text segments created by PyMuPDF, form the second input option for the snippet identifier model based on pre-determined text segments.

Since blocks are typically more granular than regions (Section 3.3), the block prediction model could be refined to merge adjacent blocks before comparing them to regions.

5.1.3 Snippet prediction

To leverage *GraphSeg segments* or blocks to predict snippets, we need to develop a statistical model to classify these segments as representing a region or not. This requires the creation of a custom dataset for training.

Dataset creation

The dataset for the TS-based snippet identifier system is created using the pre-processed AML data. Each sample is a text segment labelled as 1 if it represents a region and 0 otherwise. Building on Section 3.4, labels are defined using Jaccard similarity between Graph-

[1] `https://bitbucket.org/gg42554/graphseg/src/master/`

seg segments/blocks and regions. Segments with similarity above the optimal thresholds of 0.82 for blocks and 0.58 for GraphSeg are labelled 1, others 0. These thresholds were determined using the elbow method [104] implemented in the *kneed* package[2]. Details on our exact methodology are provided in Appendix C.

This method results in two datasets: one for the ”block prediction” model (”Blocks”) and another for the ”GraphSeg segments prediction” model (”GraphSeg”).

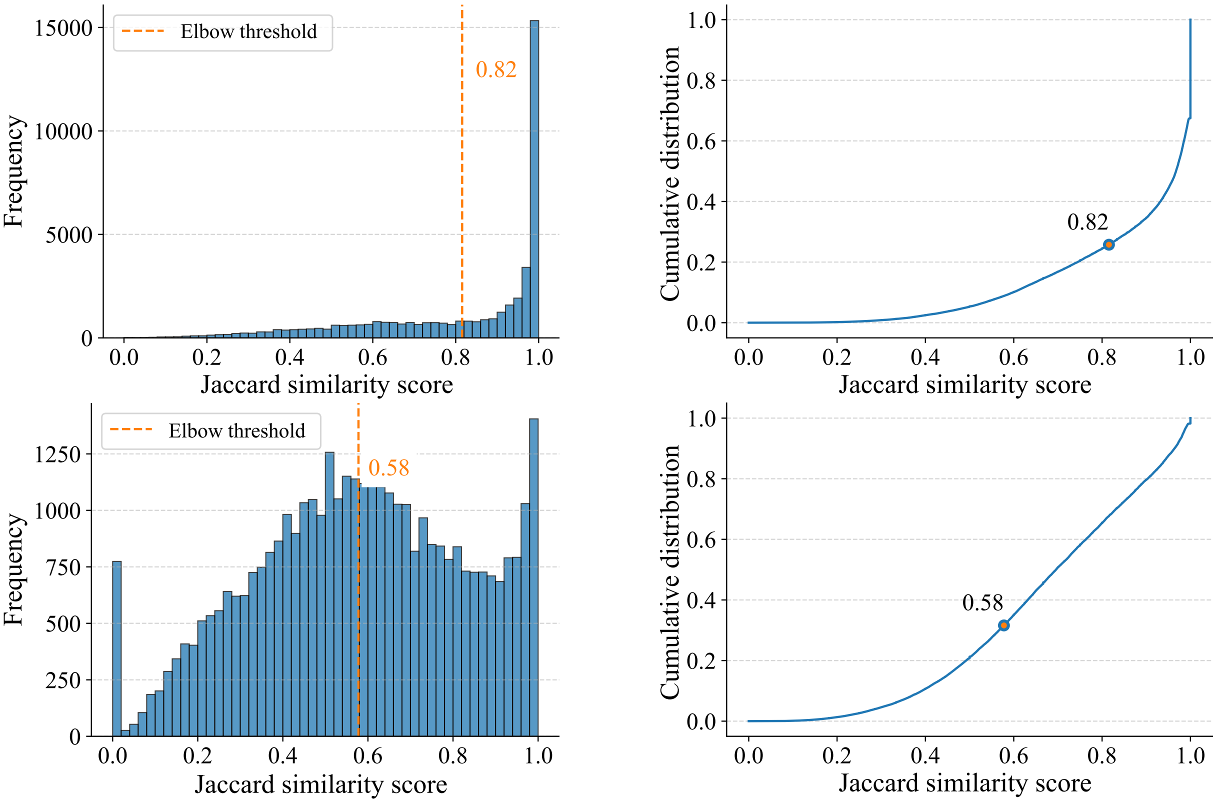


Figure 5.1: Distribution of max. similarity scores against regions: blocks (top row) and GraphSeg segments (bottom row), shown as absolute (left) and cumulative (right) values. Blocks generally align well with regions, whereas many GraphSeg segments do not.

In addition to the threshold values, Figure 5.1 confirms the strong similarity between blocks and regions. Over 70% of blocks are more than 80% similar to regions, while GraphSeg segments show a wider distribution of similarity values. This analysis indicates that while GraphSeg divides text into segments that can be very similar to regions, it also segments text into portions that do not match the regions well.

Segment prediction

Based on the two separate labelled datasets, a RoBERTa [68] model for sequence classification was trained using the HuggingFace implementation[3] for both Blocks and GraphSeg. Both models were trained on one NVIDIA A100-SXM4-80GB GPU for three epochs, with a batch size of 128, 500 warmup steps, and a weight decay of 0.01.

[2] https://pypi.org/project/kneed/

[3] https://huggingface.co/docs/transformers/v4.40.2/en/model_doc/roberta#transformers.RobertaForSequenceClassification

5.2 Sentence-level model

Chapter 2 demonstrated that many NLP tasks related to this work typically operate at the sentence level. This includes numerous TS models [e.g. 9, 19, 37, 38, 54, 57, 72, 83] as well as TZ approaches [e.g. 50, 71]. In the context of applying NLP in the legal domain, Castano et al. argue that ”a single sentence is a good candidate to become a document chunk” [15]. Based on these findings, this study also introduces a sentence-level model, which assumes that snippets are typically at most as small as what an SBD tool would identify as a sentence in a given text.

5.2.1 Model architecture

The sentence-level model builds upon the key concepts of TS but also incorporates significant components from TZ and SBD (Section 2.2). The basis of our architecture is inspired by Glavaš and Somasundaran's TS model *CATS* [38]. We present a novel hierarchical neural model, combining a first-level sentence transformer built on SBD techniques with a downstream second-level encoder network that consumes the output of the sentence transformer to make the final classifications in a TZ-inspired manner.

The following sections describe the three central components of the sentence-level model. The full architecture is illustrated in Figure 5.2.

Sentence splitting

As the sentence-level model relies on a list of sentences as input, the precursor to the first-level transformer is dividing the full text of a page into sentences. Chapter 2 highlighted the difficulty of accurately splitting legal text into sentences. Therefore, based on related research in legal SBD [8, 10, 86, 94, 95], three different sentence splitting methods were employed in this study (see Appendix C.3 for details).

1. **Customly trained Punkt**: An unsupervised SBD approach developed by Kiss and Strunk [56] and customly trained on the AML dataset using NLTK's Punkt implementation[4].

2. **Extended SpaCy**: The SpaCy[5] English language model *en_core_web_sm* with a custom component to handle consecutive newline characters (”\n”).

3. **MultiLegalSBD**: A transformer-based model specifically introduced for SBD within the legal domain by Brugger *et al.* [10] in 2023[6].

To further enhance the sentence-splitting performance, the sentence-level model can leverage the existing crude division of a page into blocks. Manual experiments showed that

[4] https://www.nltk.org/api/nltk.tokenize.punkt.html

[5] https://spacy.io/models/en

[6] https://huggingface.co/rcds/distilbert-SBD-fr-es-it-en-de-judgements-laws

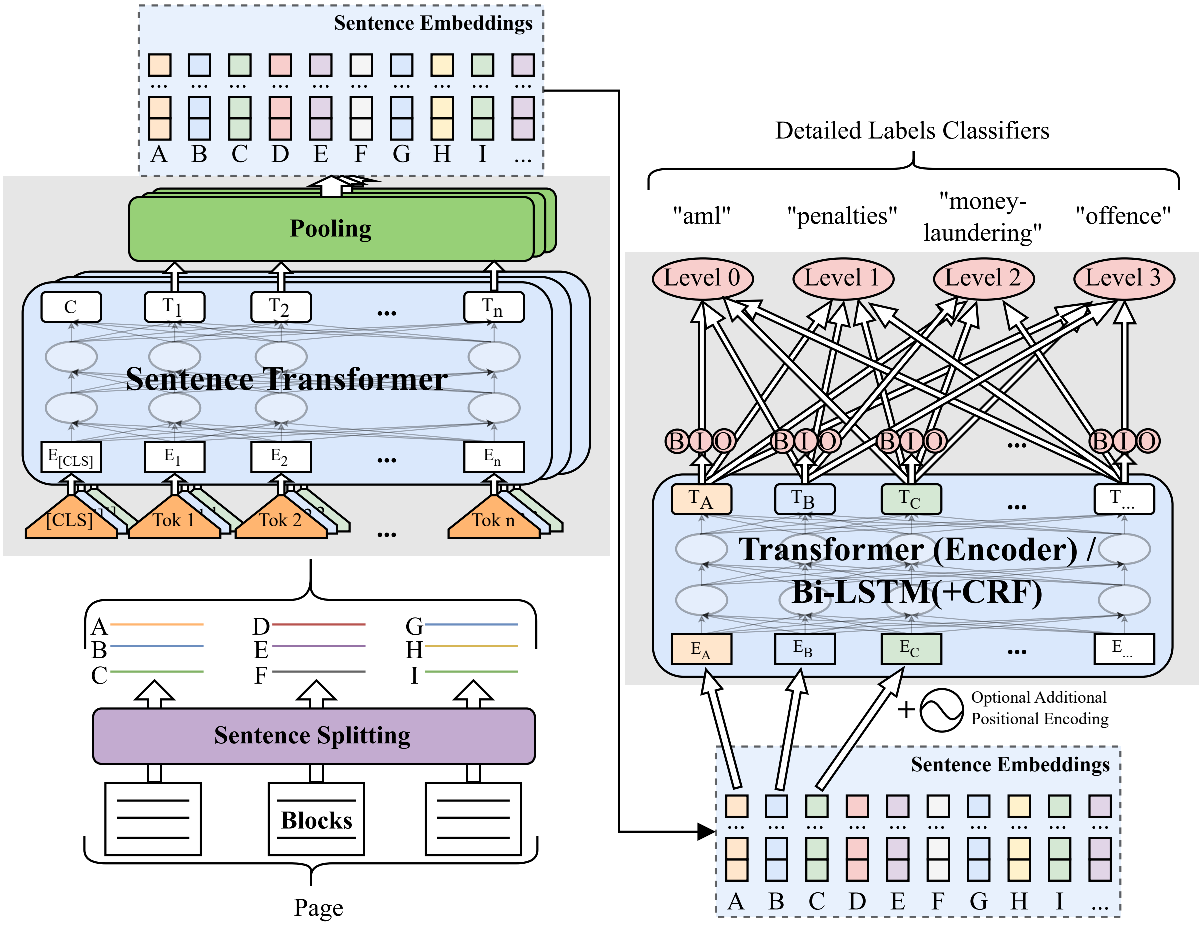


Figure 5.2: The sentence-level model architecture. Illustration based on [38, 44].

processing the page block-wise, rather than as a full text, prevented errors where all three SBD methods struggled to separate complex headings from the first content-carrying sentence as PyMuPDF would separate such headings as distinct blocks. Therefore, block-wise segmentation is expected to improve overall SBD accuracy. We note that processing text through an SBD tool removes formatting cues (e.g. "\n"), which means the sentence-level model cannot retain this information.

First level (sentence) transformer

The sentences of a page serve as inputs for the first part of the sentence-level model: a transformer model whose objective is to generate multidimensional vector representations of each sentence. These *sentence embeddings* capture the semantic content and position of each sentence within the page. This methodology aligns with hierarchical models in related TS and TZ literature [38, 50, 57]. Unlike these approaches, however, we follow Aumiller *et al.* [4] in leveraging pre-trained models for the benefit of transfer learning, specifically Reimers and Gurevych's *S-BERT* [82]. We employ the *sentence-transformers/all-MiniLM-L12-v2* model from the *sentence-transformers* library[7], which balances speed and accuracy in calculating 384-dimensional sentence embeddings [91].

[7] https://sbert.net/docs/pretrained_models.html

Second level encoder network

The context-enriched sentence embeddings from the sentence transformer are consumed by a downstream encoder network, serving as the second-level model. This model learns how to accurately transform (i.e. contextualise) the embeddings to predict snippets in regulatory documents. We evaluated various architectures for this second-level encoding network, including Bi-LSTM, Bi-LSTM+CRF, and transformer models, following state-of-the-art approaches in TS and TZ [e.g. 1, 38, 40, 50, 57]. We considered adding optional sinusoidal positional encoding to the sentence embeddings, as per Vaswani *et al.* [106], but found it did not enhance performance and sometimes even deteriorated it. This could possibly be due to the redundancy of positional data already encoded in the embeddings or the introduction of noise.

To make the sentence-level model usable for the intended downstream task of snippet identification while allowing the model to additionally learn how to classify snippets with detailed labels, final linear classifiers were implemented on top of the second-level encoder network. Building on the work by Glavaš and Somasundaran [38] and Malik *et al.* [72], we introduce a multitask learning (MTL) setting with two parallel objectives:

- **Main objective: predict snippets.** A single linear classifier predicts the main objective. Contrasting with Glavaš and Somasundaran [38], the main objective in this study cannot be to predict whether a sentence starts a new segment or not. This approach aimed at linear TS would not provide enough information to classify certain sentences as not belonging to a snippet. Instead, we employ a TZ/SBD-inspired approach, where sentences are tagged with BIO labels as either *Beginning (B)*, *Inside (I)*, or *Outside (O)* of a region. Alternatively, B and I labels can be merged into a binary setting, in which a sentence is simply classified as belonging to a region or not.

- **Auxiliary objective: predict detailed labels.** The auxiliary objective is to predict the detailed ontology labels for the snippets.

  The intuition behind the presented MTL setting is to enable the model to learn shared input representations in latent space, informing both the main and detailed label classifiers, especially given that these two objectives are interdependent. Each sentence belonging to the same snippet shares the same detailed label, while ”O” predictions indicate sentences outside snippets with no detailed label. This shared learning aims to improve performance on both objectives.

  Due to the wide variety of detailed labels in the RGP ontology, the auxiliary objective classification is split by level. A separate neural network on top of the encoder network is designated for each level of detailed label prediction. This should not only result in better performance at each level and overall but also enable dynamic adjustment of the architecture to different level predictions. Our model can be configured using a single parameter (”levels”) to predict various combinations of detailed labels. It can predict only the main objective ($levels = [\ ]$), specific levels (e.g. $levels = [1]$), or all levels simultaneously ($levels = [0, 1, 2, 3]$). Matching the AML ontology, the detailed classifiers predict 1 (trivial), 19, 130, and 154 labels for each respective level.

During training, the model's loss is a weighted sum of the main and auxiliary objectives:

$$L_{\text{total}} = \alpha \cdot L_{\text{Main}} + (1 - \alpha) \cdot \sum_{i=1}^{N} w_i \cdot L_{\text{Auxiliary}_i} \tag{5.1}$$

where each objective's loss is the average cross-entropy loss over all samples [51]:

$$L = \frac{1}{M} \sum_{o=1}^{M} \text{CrossEntropyLoss}_o \tag{5.2}$$

$$\text{CrossEntropyLoss}_o = - \sum_{c=1}^{C} y_{o,c} \log(p_{o,c}) \tag{5.3}$$

Here $y_{o,c}$ is 1 if class label c is correct for observation o, otherwise 0, and $p_{o,c}$ is the predicted probability of o being of class c. Each set of detailed labels i is weighted by w_i. The parameter α (alpha) controls the balance between the two objectives.

5.2.2 Dataset creation

The sentence-level model dataset creation involves a three-step approach: 1. block-wise splitting a page into sentences, 2. computing the sentence embeddings, and 3. labelling the sentences.

As the first two steps are detailed above, the labelling logic can be summarised as follows: Each sentence on a page is mapped to token indices through a series of string cleaning and matching steps. Subsequently, the start and end indices of each region obtained during data pre-processing (Chapter 4) are used to label each sentence by comparing their boundaries with those of the regions:

Let $s_i = (start_i, end_i)$ be the start and end indices of the i-th sentence, and $r_j = (start_{r_j}, end_{r_j})$ be the start and end indices of the j-th region of a page. The label L_i for the i-th sentence is determined as follows:

$$L_i = \begin{cases} B & \text{if } \exists j : start_i \leq start_{r_j} < end_i \\ I & \text{if } \exists j : start_{r_j} < start_i \wedge end_i \leq end_{r_j} \\ O & \text{otherwise} \end{cases}$$

Sentences starting a new region are labelled ”B”, those entirely within a region are labelled

"I". These sentences are also assigned the detailed labels of the corresponding region. Other sentences are labelled as "O" and assigned the detailed label "N/A".

The final dataset is split according to the centralised split definitions and saved in a HuggingFace DatasetDict[8]. Each sample contains the sentences, sentence embeddings, bio-labels, detailed labels, and metadata (document ID and page ID) for one page.

5.2.3 Model optimisation and hyperparameter tuning

The sentence-level model variants were optimised in a comprehensive study following a step-wise coarse-to-fine approach. First, we analysed the impact of basic model components. Once the main setup was established, we tested the three different second-level encoder network architectures and optimised their parameters.

In the initial phase, the three sentence-splitting methods were tested, and their performance was evaluated on the AML validation set. Table 5.1 shows the validation performance of the model in the final training epoch when training for 25 epochs with an *early_stopping_patience* of 5 epochs based on validation loss[9].

	Sentence split (main objective)	F1	Precision	Recall	Cohen's kappa
	SpaCy (Binary)	0.551	0.629	0.490	0.387
+5.58%	NLTK (Binary)	0.582	0.631	0.540	0.397
+2.02%	**MultiLegalSBD (Binary)**	**0.594**	**0.631**	**0.560**	**0.394**
-6.07%	MultiLegalSBD (BIO)	0.558	0.648	0.490	0.349

Table 5.1: Comparison of different sentence-level model architectures. Using MultiLegalSBD for sentence splitting and following a binary main objective yielded the best results.

Using a transformer encoder network with standard parameters[10] and binary labels for the main objective, MultiLegalSBD emerged as the most effective SBD tool, based on the validation F1-score. This aligns with its strong performance in the legal domain [10]. Experimenting with a BIO schema decreased performance, likely because the higher granularity of sentences does not require a BIO approach and the binary setting better addresses class imbalance by reducing the dominance of "O" samples.

We followed related research in calculating the following metrics for each model by introducing a custom *compute_metrics* function into the HuggingFace *trainer* class[11] for evaluation on the AML training, validation and test datasets [4, 7, 37, 38, 72, 83, 88].

[8] `https://huggingface.co/docs/datasets/en/package_reference/main_classes#datasets.DatasetDict`

[9] `https://huggingface.co/docs/transformers/en/main_classes/callback#transformers.EarlyStoppingCallback`

[10] Parameters: $nhead = 8$, $hidden\_dim = 1024$, $nlayers = 8$, $dropout = 0.1$

[11] `https://huggingface.co/docs/transformers/en/main_classes/trainer#transformers.Trainer.compute_metrics`

It should be noted that — in contrast to the metrics presented in Section 3.4, which are employed to evaluate the similarity between two texts — the below metrics are calculated to compare model predictions (i.e. BIO or detailed labels) versus the true labels in the AML dataset.

- **Precision, recall, and F1-score:** Calculated across all samples using the standard classification formulas shown in Equations 3.2-3.4.
- **Cohen's kappa:** Measures inter-annotator agreement, i.e. predicted snippet labels versus region labels, calculated using the scikit-learn library[12] as

  $$\kappa = \frac{P(A) - P(E)}{1 - P(E)} \tag{5.4}$$

  where $P(A)$ is the observed agreement, and $P(E)$ is the expected agreement by chance [13, 88, 96]. Scores typically range between 0 and 1, with 1 indicating perfect agreement [88].
- **pK:** Metric introduced by Beeferman *et al.* [7] that measures segmentation accuracy as the proportion of times segmentation boundaries are placed incorrectly and is thus typically utilised in TS tasks [77]. We calculated pK using the *segeval*[13] library by first transforming BIO predictions into (binary) segmentation boundaries and subsequently comparing the number of snippet boundaries within a sliding window of size k:

  $$\text{pK} = \frac{1}{N-k} \sum_{i=1}^{N-k} \mathbb{I}(b(i, i+k) \neq b'(i, i+k)), \ k = \frac{N}{2M} \tag{5.5}$$

  where N is the total number of labels, M is the number of segments, $b(i, i+k)$ is the reference segmentation boundaries (regions), and $b'(i, i+k)$ is the predicted segmentation boundaries (snippets) within the window [7].

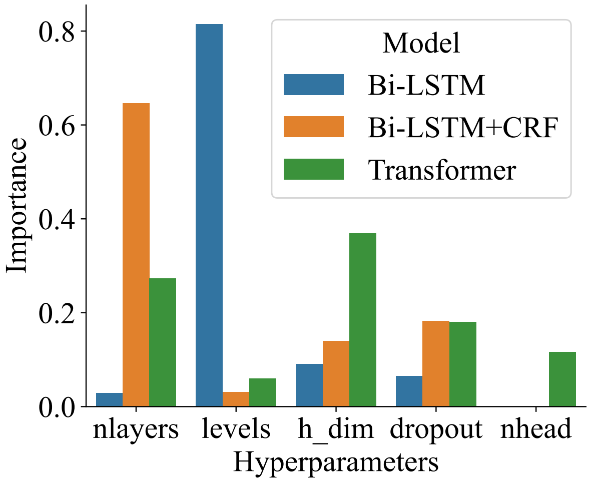
  We specifically evaluate the snippet identifier systems using pK due to this study's close relatedness to TS. By treating snippet identification as a TS task, the pK metric allows us to statistically determine the system's accuracy in identifying the correct beginning and end of a snippet — a capability precision, recall, and f1-score, for example, lack [7]. We also considered *WindowDiff*, introduced by Pevzner and Hearst [77] as an improvement over pK, but in line with related literature only report pK here [4, 38]. A lower pK value indicates better performance [7].

Given the results in table 5.1, the sentence-level model with MultiLegalSBD as its SBD-component and binary main prediction objective formed the basis for a comprehensive hyperparameter optimisation (HPO) in which differently configured Bi-LSTM, Bi-LSTM+CRF, and transformer networks were explored as the second-level encoder. The HPO's objective was intentionally defined as minimising the loss of the main prediction

[12] `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.cohen_kappa_score`

[13] `https://segeval.readthedocs.io/en/latest/`

objective while testing if predicting additional detailed labels as part of the auxiliary objective would improve performance.



Params	**Bi-LSTM**	**Bi-LSTM +CRF**	**Trans-former**
levels	N/A	[1, 2]	[1]
h_dim	256	512	2048
nlayers	5	3	8
dropout	0.20	0.31	0.31
w_lvl_0	N/A	0.14	0.55
w_lvl_1	N/A	0.20	N/A
alpha	N/A	0.49	0.90
n_head	N/A	N/A	4
Best value	0.46	0.39	**0.32**

Figure 5.3: Hyperparameter importance (left) and best parameters with validation loss (right) for each encoder network as identified during HPO. The transformer emerged as the best model.

The HPO was conducted leveraging the HPO framework *Optuna*[14] with default configurations (e.g. *TPESampler* and *MedianPruner*). We optimised key parameters of the encoder networks, including hidden dimensions (*h_dim*), number of layers (*nlayers*), dropout rate (*dropout*), and number of heads for the transformer model (*n_head*), along with different weighted auxiliary objectives (*levels* and *w_lvl_<level>*). Appendix D details all evaluated settings, while Figure 5.3 illustrates hyperparameter importance and the best results per encoder network type. We trained 50 distinct models per architecture.

The Bi-LSTM's best performance on the validation set was inferior to the Bi-LSTM+CRF and transformer models. The most important hyperparameter for the Bi-LSTM was *levels*, while the other two models were mainly influenced by the number of layers and hidden dimensions, suggesting that the auxiliary objective overwhelmed the Bi-LSTM. In contrast, the BiLSTM+CRF and transformer-based sentence-level models benefited from the additional information. Overall, a transformer-based encoder network, slightly influenced by predicting the *level 1* detailed label, emerged as the best encoder.

Based on these analyses, the final sentence-level model utilises MultiLegalSBD for sentence splitting and a transformer encoder network with parameters as outlined in Figure 5.3. The main objective follows a binary prediction strategy.

5.3 Token-level model

The token-level model more strongly incorporates key elements from TZ (Section 2.2.2) and SBD (Section 2.2.3). Namely, unlike its sentence-level counterpart, it treats snippet

[14] `https://optuna.org/`

identification as a token-level sequence labelling problem, employing strategies typical of NER tasks. This allows for more granular text analysis, building on methods detailed by Ajjour *et al.* [1], Gnehm [39], Gnehm and Clematide [40], and Sanchez [86]. Additionally, the model leverages elements from the sentence-level architecture, incorporating the same MTL framework and utilising pre-trained large language models (LLMs).

5.3.1 Model architecture

The token-level model, shown in Figure 5.4, only consists of a single BERT-based transformer network, which directly consumes tokens as input. To meet the 512 token limit, pages are tokenized in a sliding window procedure. Pages with more than 510 tokens — two tokens are reserved for BERT-like models' special start ("[CLS]"), and end tokens ("[SEP]") — are divided into 510-token windows [49]. To mitigate the context loss from splitting, we use overlapping windows, experimenting with overlaps of 64, 128, and 256 tokens [48]. Pages with fewer than 512 tokens are padded to the full sequence length.

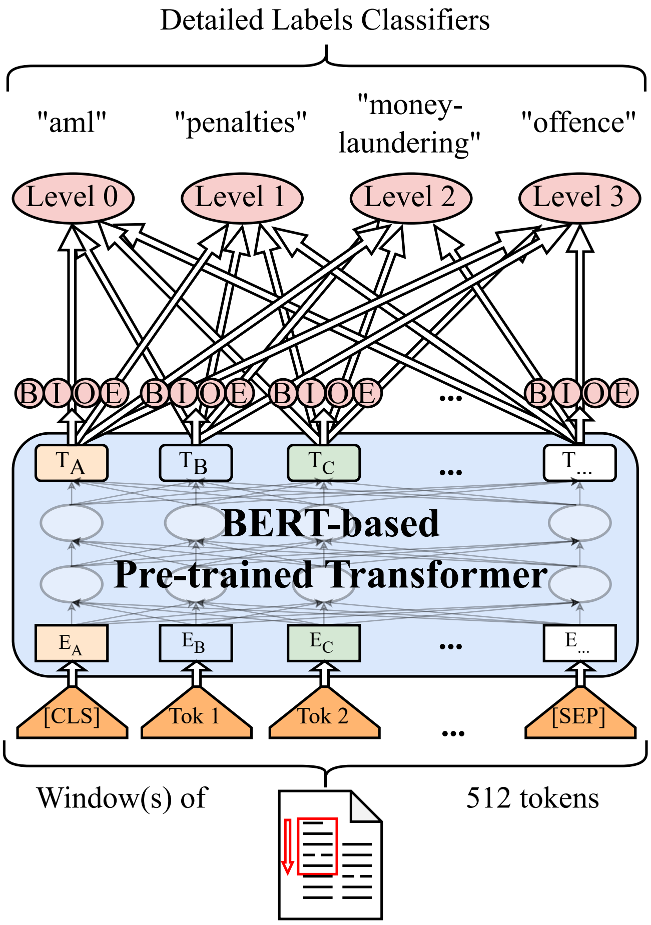


Figure 5.4: Architecture of the token-level model. Illustration based on [25, 44].

Leveraging pre-trained LLMs from HuggingFace's *AutoModel* class[15], the token-level model does not require custom word embeddings or positional encoding, as these are handled

[15] https://huggingface.co/docs/transformers/model_doc/auto#transformers.AutoModel

internally based on the pre-trained data[16]. We experimented with RoBERTa models in *distil*[17], *base*[18], and *large*[19] versions, as well as two RoBERTa-based models, pre-trained on legal documents: *saibo/legal-roberta-base*[20] and *lexlms/legal-roberta-base*[21], introduced by Chalkidis *et al.* [17]. For all models, default settings[22] were used.

The classification architecture mirrors the sentence-level model but incorporates a different main objective, where, next to the standard BIO classification objective, a BIOE approach was tested. While distinguishing ending tokens may not be practical at the sentence level due to the granularity required, it becomes relevant for token-based snippet identification because special "E" (Ending) tokens, such as periods ("."), frequently indicate region boundaries.

5.3.2 Dataset creation

A sample for the token-level model includes all tokens in one 512-token window of a page, labelled with BIOE tags using the start and end indices of regions identified during data pre-processing. For each region, the start token is labelled "B", the end token "E" (or "I" in the BIO schema), intermediate tokens "I", and all other tokens "O".

Research shows that BIO(E)-based labelling approaches, including those used in this study, cannot straightforwardly handle nested elements [2]. This may lead to issues with overlapping regions (Figure 4.1), where consecutive tokens may be incorrectly labelled as "B", for example. Due to the rarity of such cases (also see Table 6.4) and for the sake of simplicity, addressing this issue is kept for future work.

Samples are saved in a HuggingFace DatasetDict. Each window-based sample contains the tokenizer-assigned input IDs of the tokens included, the corresponding attention mask to neglect padding tokens, the BIO(E) and detailed labels, and metadata (document ID, page ID, and window ID). The final AML training dataset comprises 74,450 samples.

5.3.3 Model optimisation and hyperparameter tuning

Mirroring Section 5.2.3, the token-level model's components were successively optimised on the AML validation dataset. Specifically, different sliding window sizes, main objectives (BIOE vs. BIO), and pre-trained models were evaluated (Table 5.2). Validation scores were taken from the final training epoch after 15 epochs with $early\_stopping\_patience = 4$.

Increasing window overlap improved performance marginally between 64, 128, and 256 tokens. The BIOE schema yielded slightly better results than BIO while additionally offering

[16] `https://huggingface.co/docs/transformers/model_doc/bert`
[17] `https://huggingface.co/distilbert/distilroberta-base`
[18] `https://huggingface.co/FacebookAI/roberta-base`
[19] `https://huggingface.co/FacebookAI/roberta-large`
[20] `https://huggingface.co/saibo/legal-roberta-base`
[21] `https://huggingface.co/lexlms/legal-roberta-base`
[22] `https://huggingface.co/docs/transformers/v4.40.2/en/main_classes/configuration`

	Model	F1	Precision	Recall	Cohen's kappa
	RoBERTa Win 64 (BIOE)	0.873	0.836	0.915	0.798
+0.30%	RoBERTa Win 128 (BIOE)	0.876	0.837	0.919	0.802
+0.35%	**RoBERTa Win 256 (BIOE)**	**0.879**	**0.857**	**0.902**	**0.810**
-0.91%	RoBERTa Win 256 (BIO)	0.871	0.855	0.888	0.798
-0.05%	Saibo Win 256 (BIOE)	0.871	0.851	0.891	0.796
-11.97%	Lexlms Win 256 (BIOE)	0.766	0.703	0.842	0.629

Table 5.2: Comparison of different token-level model architectures. A RoBERTa-based model with 256 token overlap in a BIOE setting yielded the best results.

greater label granularity — a characteristic beneficial for downstream applications. Interestingly, the pre-trained models *saibo/legal-roberta-base* and *lexlms/legal-roberta-base* did not outperform RoBERTa-base, likely due to smaller pre-training datasets or overfitting to specific legal texts, reducing their transferability to the regulatory content analysed in this study [17, 68]. Overall, the token-level model achieved strong validation performance, with F1 scores close to 0.9 (+30 pp compared to the sentence-level model in Table 5.1). Notably, the best model achieved a Cohen's kappa score of 0.81, which is close to the 0.836 reported by Saravanan and Ravindran [88] for human annotators in related legal tasks, and surpasses Krippendorff's 0.80 threshold for good reliability [59].

Using RoBERTa-based transformer networks with default configurations, the token-level model's Optuna-HPO was solely focused on the evaluation of the effects of the auxiliary objective. Here too, the HPO objective was defined as reducing the main objective loss. Due to higher training costs, we evaluated 18 different token-level models. Results showed no performance improvement including the auxiliary objective, possibly due to the added complexity diverting the model's focus from the main task. The details of the token-level HPO are outlined in Table D.4.

Based on the optimisation study, a RoBERTa-based model with 256-token overlaps and a BIOE main objective was chosen as the best token-level model for snippet discovery.

5.4 Post-processing and snippet creation

To identify FRI in legal documents and predict snippets, we developed a custom post-processing algorithm to convert the systems' BIO(E) predictions into snippets. The TS-based models do not require post-processing as they directly predict snippet-like segments.

The post-processing algorithms for both the sentence-level model and token-level models follow a similar strategy. The algorithm sequentially evaluates the predicted labels to collect and merge the textual content for the snippets. "B" labels mark the start of a snippet, "I" labels continue it, "E" labels end it, and "O" labels mark non-snippet content. Figure 5.5 exemplifies this process.

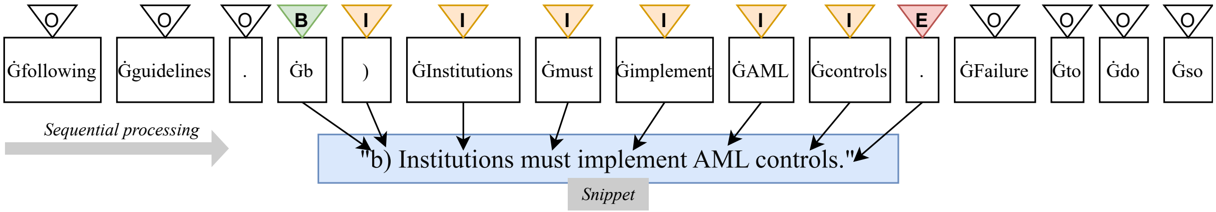


Figure 5.5: Simplified example illustrating the post-processing algorithm for the main objective. BIO(E) labels are used to construct snippets.

Our post-processing implementation accounts for a variety of prediction inaccuracies. We allow ”I” labels to start snippets if they follow ”O” labels and end snippets if ”O” labels follow ”B” or ”I” labels. To manage incorrect ”O” tokens within snippets, the *ignore_o_threshold* parameter and a look-ahead mechanism were introduced. This parameter regulates how many ”O” sentences/tokens can be overlooked to continue a snippet considering the following labels. Another parameter, *min_snippet_length*, ensures snippets reach a minimum size.

To identify the optimal values for *ignore_o_threshold* and *min_snippet_length*, a grid search was conducted for the token-level model. Exploring all 25 combinations of *ignore_o_thresholds* $= [0, 2, 4, 8, 10]$ and *min_snippet_lengths* $= [0, 4, 8, 10, 12]$, the optimum on the AML validation dataset, based on the highest Jaccard similarity of snippets against regions, was identified as 4 and 10 respectively. For the sentence-level model, *ignore_o_threshold* was set to 0. *Min_snippet_length* was not used due to the higher granularity of sentences. Although allowing to skip certain ”O”-sentences would increase recall, it results in a significant drop in precision and is thus discouraged.

For the auxiliary objective, a majority voting mechanism was implemented: detailed labels for a snippet are determined by the most frequent label among its predicted elements.

5.5 Ensemble approach

In an effort to form a technical symbiosis between sentence-level and token-level approaches for snippet identification, we introduce an ensemble model. Figure 5.6 illustrates the architecture of this model. By creating a mapping between the tokens and sentences of a page, we can independently process the same input page through both models before binarising the token-level predictions and utilising the mapping to aggregate the token-level model's predictions to a sentence-level.

For all samples i, the final sentence-level predictions of both models, P_{token} and P_{sentence}, are combined through a normalised weighted average (default: $w_{\text{token}} = w_{\text{sentence}} = 0.5$):

$$P_{\text{combined}} = \left(\frac{P_{\text{token}} \cdot w_{\text{token}} + P_{\text{sentence}} \cdot w_{\text{sentence}}}{\sum_i \left(P_{\text{token}} \cdot w_{\text{token}} + P_{\text{sentence}} \cdot w_{\text{sentence}}\right)_i} \right) \tag{5.6}$$

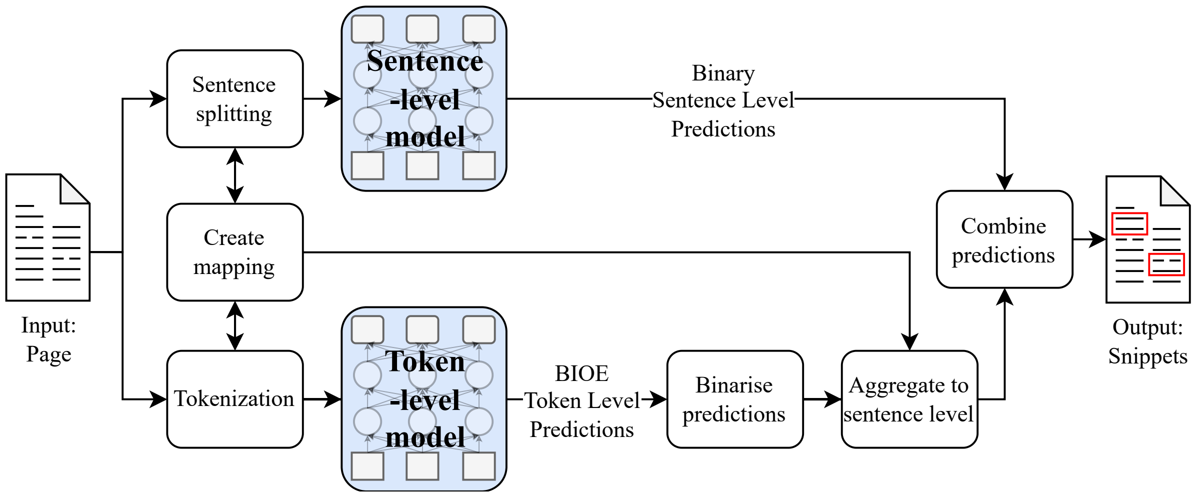


Figure 5.6: Inference process using the ensemble model.

It should be noted that the ensemble does not require a dedicated training/dataset. Instead, it relies on a pre-trained token-level and sentence-level model. Given the ensemble model ultimately works on a sentence level, the sentence-level model's post-processing algorithm can be applied to obtain the final snippets.

5.6 The snippet identifier system: FRIDAY

This chapter has presented the **F**inancial **R**egulatory **I**nformation **D**iscovery and **A**nnotation s**Y**stem (FRIDAY). Figure 5.7 illustrates how the different components and chapters of our study integrate to form an end-to-end snippet identifier system. While preprocessing (Chapter 4) and evaluation (Chapter 6) are crucial for development and training, only the components highlighted in green are necessary for the practical deployment of FRIDAY.

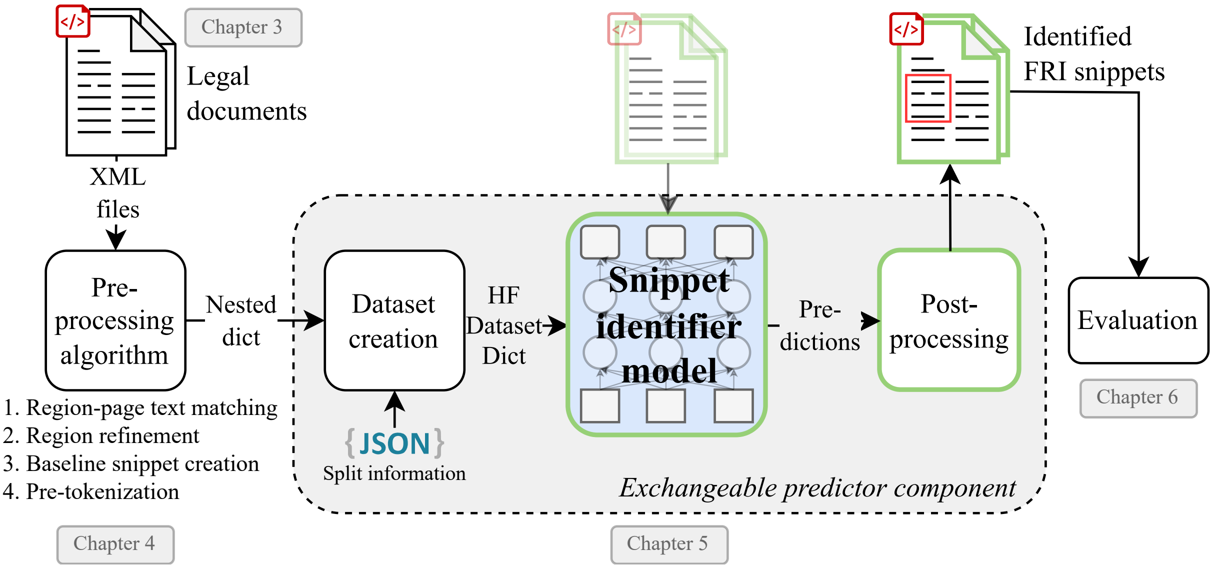


Figure 5.7: Overview of the snippet identifier system FRIDAY.

FRIDAY is designed to be highly modular. Not only is each component of the system readily maintainable and adjustable but, as shown in this chapter, the predictor component, i.e. the snippet identifier model, is also easily exchangeable. Notably, all our models are fully integrated and compatible with the HuggingFace ecosystem[23]. We evaluate different model variants in the following chapter. In addition to the ML models presented in this work, other architectures are also conceivable.

With FRIDAY we contribute a ready-to-use end-to-end system with significant applications in both academia and industry.

[23] `https://huggingface.co/`

Chapter 6

Evaluation

FRIDAY, in various configurations, was evaluated primarily on the main objective of identifying FRI snippets in legal documents while also assessing auxiliary objective performance. The evaluation included both quantitative and qualitative methods.

Quantitatively, the focus was on identifying the best FRIDAY version using linguistically informed metrics, with a TS-optimised token-level model showing the best performance. The system's performance in predicting detailed labels was robust, achieving significant accuracy even for complex levels of the RGP ontology. Additionally, FRIDAY demonstrated strong generalisation capabilities to the unseen cybersecurity datasets, maintaining similar performance to the AML theme.

Qualitatively, we analysed the system's prediction behaviour, recognising its strengths in correctly labelling continuous text segments. We also examined the impact of the post-processing algorithm and potential areas for further improvement.

6.1 Quantitative evaluation

As part of the quantitative evaluation, all different system identifier architectures, including the baseline models, were considered. Experiments were run on the High Performance Computing (HPC) platform provided by the Cambridge Service for Data Driven Discovery (CSD3). Specifically, one NVIDIA A100-SXM4-80GB GPU was utilised in combination with 32 cores of an AMD EPYC 7763 64-Core Processor 1.8GHz and 250 GiB of RAM[1].

6.1.1 Evaluation methodology and metrics

A custom evaluation process was developed to evaluate FRIDAY's capabilities in identifying snippets in unseen legal documents. Predicted snippets were mapped to the best-matching region using maximum Jaccard similarity to determine the region that the model

[1] https://docs.hpc.cam.ac.uk/hpc/user-guide/a100.html

most likely aimed to predict.

Given the importance of linguistic accuracy in this study, we used metrics that can account for the quality of matching texts and accurately capturing content. Concretely, we calculated the seven different metrics for evaluating text similarity described in Section 3.4 for all identified region-snippet pairs to assess downstream task performance.

It is noteworthy that calculating metrics based on tokens created by the RoBERTa BPE tokenizer establishes a significantly more challenging evaluation objective than evaluating unformatted text. This is because the RoBERTa tokenizer not only creates sub-word tokens but also includes formatting cues in its tokens (e.g. ”Ġ”, or ”Ċ” prefixes)[2]. For consistency, the sentence-level and ensemble models, whose SBD components remove formatting elements, were evaluated using the tokenized concatenated sentences of a page.

6.1.2 Predicting snippets in unseen AML documents

All variants of FRIDAY were first evaluated based on their performance in predicting unseen AML documents. For this purpose, the models made predictions on the 173 test documents comprising 6,930 pages of unseen AML regulations.

Main Objective

Table 6.1 summarises the performance of all models developed as part of this study, evaluated as described in Section 6.1.1 and ordered by their ROUGE-1-F score.

Model	ROUGE-1-F ↑	ROUGE-2-F	ROUGE-L-F	Jaccard	F1	P	R	Time/Page (ms)**
GraphSeg (baseline)	0.63	0.57	0.63	0.51	0.62	0.52	0.79	136.42
Snippeting alg. (baseline)	0.66	0.61	0.66	0.53	0.61	0.48	0.84	24.92
Sentence*	0.72	0.67	0.72	0.61	0.68	0.62	0.75	18.71
Blocks	0.75	0.70	0.74	0.69	**0.78**	**0.74**	0.83	21.07
Token-base-main+auxil.	0.75	0.71	0.75	0.72	0.70	0.63	0.79	20.48
Token-large-main	0.78	0.74	0.78	0.67	0.69	0.58	0.84	56.26
Ensemble*	0.79	0.75	0.79	0.70	0.75	0.64	**0.90**	87.46
Token-distilroberta-main	0.84	0.81	0.84	0.74	0.72	0.64	0.81	**8.28**
Token-base-main	0.84	0.81	0.84	0.74	0.72	0.64	0.82	17.02
Token-base-pk-main	**0.86**	**0.83**	**0.86**	**0.77**	0.74	0.65	0.84	17.48

*evaluated on unformatted text
**Maximum batch sizes used during inference: 128 (Blocks, GraphSeg), 512 (Token-large-main), 2048 (other token-level models), all sentences of a page (Sentence, Ensemble)

Table 6.1: Comparison of the different snippet identifier systems' performances on the AML test dataset. The token-level models achieved the highest ROUGE and Jaccard similarity scores.

The examined models include the *Blocks*, *sentence-level* (”Sentence”), *token-level* (”Token”), and *Ensemble* model as well as the *Snippeting algorithm* and *GraphSeg* model as

[2] https://huggingface.co/docs/transformers/en/model_doc/roberta#transformers.RobertaTokenizer

baselines. For the token-level model, different RoBERTa variants were tested. Specifically, we tested *RoBERTa-base*, *RoBERTa-large* [68], and the knowledge-distilled version *distilroberta* [87] to optimise for different resource scenarios. Additionally, a RoBERTa-base model was trained on both the main and auxiliary objectives ($levels = [1]$).

Based on the systems' performances on unseen AML documents, the following observations can be made:

- **GraphSeg:** The GraphSeg baseline model exhibits the lowest performance across nearly all metrics. In line with Section 5.1.3, this confirms that GraphSeg segments are often insufficient to accurately represent regions. Furthermore, the model demonstrates the by far slowest inference time per page due to the need to execute the unsupervised GraphSeg algorithm [37] and the RoBERTa sequence classifier.
- **Snippeting algorithm:** The baseline snippeting algorithm runs without GPU acceleration and demonstrates an acceptable inference time. It performs similarly to GraphSeg but has a high recall of 84% due to its tendency to split page texts by size, creating large segments likely to capture snippets (Section 3.2.1).
- **Sentence-level model:** Despite having the worst recall (0.75), the sentence-level model outperforms the baseline models across all other metrics and is faster. Yet, it is inferior to this study's other main models. This is likely due to its rather complex architecture and data pipeline, including two transformers and an SBD component whose accuracy strongly impacts the performance of the whole model. Furthermore, the model is less flexible than the token-level model due to its higher abstraction.
- **Blocks:** Given the close similarity between blocks and regions (Section 3.4), it is unsurprising that the Block model achieves competitive performance, outperforming both baselines and the sentence-level model. Among all models, it achieves the highest precision and F1-score, with a high recall of 0.83 and good inference speed.
- **Token-level models:** The token-level models overall perform the best, significantly outperforming both baselines and the sentence-level model. They achieve up to 20 pp higher ROUGE and Jaccard similarity scores than the baseline models while maintaining strong recall and F1 scores. Even the token-level model integrating the auxiliary objective is superior to the previously mentioned models, although this model and the RoBERTa-large-based model show the lowest performance among the token-level models. The distilroberta and RoBERTa-base token-level models exhibit a remarkably stronger performance while also demonstrating the most efficient inference speeds, as low as 8.28 ms per page.
- **Ensemble:** While the ensemble model does not outperform the sentence-level and token-level models — likely due to the comparatively poor performance of the former — integrating both models significantly improved performance compared to the sentence-level model alone. In fact, the ensemble model is the only one to achieve

a recall of 0.9 with solid performance overall. However, this hybrid approach comes at the expense of computational efficiency.

In summary, the token-level model-based snippet identifier system shows the strongest performance in identifying snippets in unseen legal documents, with ROUGE and Jaccard similarity metrics of up to 0.86 and 0.77, respectively. We reiterate that recall should be prioritised over precision in the context of this work (Section 3.4), and the token-level model exhibits desirable characteristics in this respect. Additionally, it performs well while maintaining the original text formatting, making the results even more noteworthy. Depending on use cases and priorities, certain models may be preferred. For instance, our distilroberta-based system offers near-best performance while being twice as fast and computationally more efficient [87].

Among all models, *token-base-pk-main* emerged as the best and is detailed below.

The Token-base-pk-main model

To further enhance the strong performance of our token-level architecture, we experimented with a variant optimised for a TS-related objective. Given the close relationship between snippet discovery and TS, optimising for TS should improve downstream performance. Therefore, we trained a RoBERTa-base token-level model on the main objective as before. However, instead of using validation loss for early stopping, we monitored the pK metric (Section 5.2.3), typically employed in TS tasks [4, 38, 77]. Figure 6.1 shows the training process of this model, dubbed *token-base-pk-main*. The model continued training even as the validation loss increased, which would usually indicate overfitting, and stopped once the pK validation score stopped decreasing.

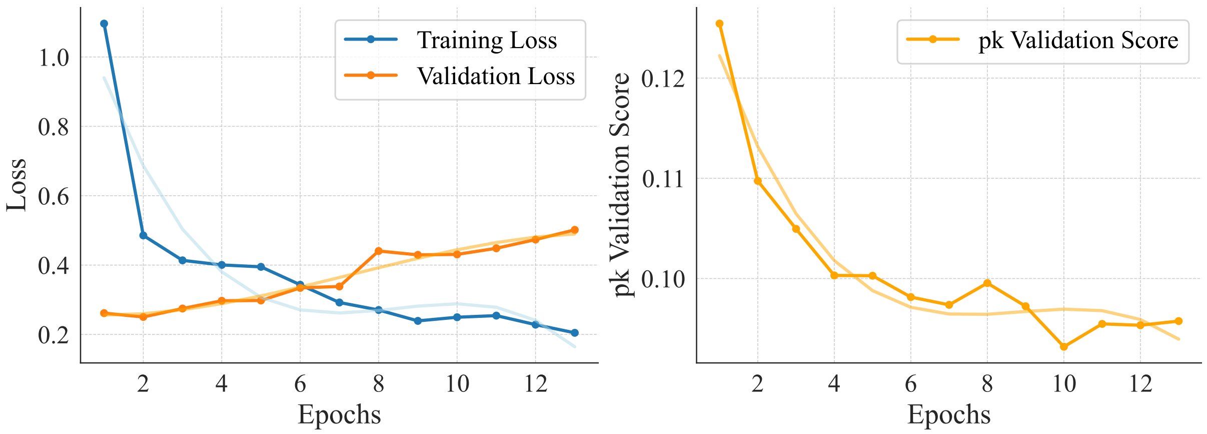


Figure 6.1: Training process of the token-base-pk-main model. The model continued training despite increasing validation loss as the validation pK score decreased.

The pK validation results are robust, considering similar models, although limited in comparability, achieved pK values of 12.50% at best on related TS tasks [4]. Table 6.1 confirms that a TS-optimised token-level model outperforms traditionally trained models.

Given its superior performance, we select the token-based pk-main model as the final predictor component for FRIDAY.

Auxiliary Objective

Although this work focuses on developing a first-of-its-kind NLP system to discover FRI in legal documents, the introduction of the auxiliary objective into the token-level model allows FRIDAY to also address our extension goal of matching the identified snippets with labels from the RGP ontology.

Figure 6.2 illustrates the F1-score, precision, and recall on the AML validation set for a version of FRIDAY extended with the auxiliary objective of predicting the detailed labels for the second ("Level 1") and third level ("Level 2") of the ontology. The BIOE labels ("Main") performance is shown in grey for comparison. The model was trained with *alpha* = 0.65 to maintain focus on the main objective. Hence, the auxiliary objective performance could likely be improved with further optimisations.

For the 19 labels of the second level, the model achieves an F1 score of around 0.6 on the validation set. This performance drops by about 0.26 on average for predicting one of the 130 labels in the third level, where the high number of classes and elevated level of detail significantly aggravate the prediction task.

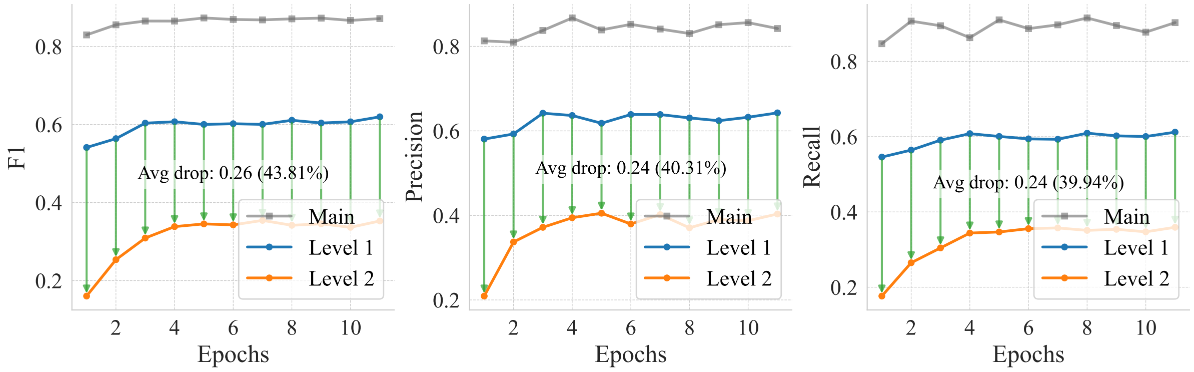


Figure 6.2: Validation performance for the auxiliary objective. Performance decreases with added detail. Predicting the first ontology level ("aml") is trivial, while the fourth level is even more complicated than the third.

On the AML test set, the performance of the model remains stable, with F1 scores of 0.58 and 0.33, precision of 0.6 and 0.36, and recall of 0.59 and 0.35 for the first and second levels, respectively.

6.1.3 Generalisation capabilities

To further understand the capabilities of FRIDAY, the system was tested on 1,694 documents from the CYBER I and CYBER II datasets, expanding the previous evaluation with an additional 53,570 pages of unseen FRI. This methodology aimed to determine how well FRIDAY, solely trained on AML documents, can generalise to new domains.

We also evaluated the two baselines (GraphSeg and snippeting algorithm) and Blocks on this data.

Table 6.2 and Table 6.3 show the results for CYBER I and CYBER II, respectively. For simplicity, we only report the ROUGE score for the longest common subsequence (ROUGE-L-F). The percentage values indicate the performance change compared to the AML test set.

Model	ROUGE-L-F ↑	Jaccard	F1	Precision	Recall
GraphSeg (baseline)	0.61 (-3%)	0.50 (-2%)	0.63 (+0%)	0.54 (+5%)	0.74 (-7%)
Snippeting alg. (baseline)	0.64 (-2%)	0.52 (-3%)	0.60 (-3%)	0.46 (-4%)	0.83 (-1%)
Blocks	0.77 (+5%)	0.73 (+5%)	**0.80** (+3%)	**0.75** (+2%)	**0.86** (+4%)
FRIDAY	**0.83** (-3%)	**0.74** (-3%)	0.73 (-0%)	0.67 (+3%)	0.81 (-4%)

Table 6.2: Comparison of different model performances on the unseen CYBER I dataset.

Model	ROUGE-L-F ↑	Jaccard	F1	Precision	Recall
GraphSeg (baseline)	0.63 (+1%)	0.53 (+3%)	0.66 (+5%)	0.61 (+18%)	0.71 (-10%)
Snippeting alg. (baseline)	0.70 (+7%)	0.58 (+10%)	0.67 (+9%)	0.55 (+15%)	**0.83** (-0%)
Blocks	0.76 (+3%)	0.71 (+3%)	**0.80** (+2%)	**0.79** (+7%)	0.81 (-3%)
FRIDAY	**0.84** (-2%)	**0.75** (-2%)	0.76 (+3%)	0.72 (+10%)	0.80 (-5%)

Table 6.3: Comparison of different model performances on the unseen CYBER II dataset.

Contrary to Ajjour *et al.* [1], who reported significant performance drops in cross-domain tasks due to the reliance on domain-specific vocabulary, FRIDAY maintained strong performance in the unseen domain of cybersecurity, similar to the AML results (Table 6.1). The system still outperformed all other models on ROUGE-L-F and Jaccard similarity.

Notably, almost all models exhibited lower recall and higher precision, especially in the more diverse CYBER II domain. This trend suggests that in a cross-domain generalisation setting, the models struggle to identify all relevant regulations — likely due to domain-specific language differences. However, identified regulations were still recognised with high confidence. Thus, while the models are precise in recognising familiar patterns, they may miss relevant content expressed differently in the new domain.

Overall, the analysis shows that FRIDAY has effectively learned and generalised a robust understanding of the important linguistic and textual features needed to identify FRI in unseen legal documents.

6.2 Qualitative Evaluation

To gain qualitative insights into FRIDAY's predictive behaviour, we evaluated its BIOE-based main objective through an analysis of consecutive token predictions, following related research [1]. Table 6.4 presents a confusion matrix for these predictions on the AML test dataset.

	Predictions														
Gold	B-B	B-I	B-O	I-B	I-I	I-E	I-O	E-B	E-I	E-E	E-O	O-B	O-I	O-E	O-O
B-B	**0**	0	0	0	2	0	0	0	0	0	0	0	0	0	0
B-I	6	**2.9k**	101	4	2.6k	2	31	0	8	0	71	23	172	0	1.3k
B-O	0	0	**0**	0	0	0	0	0	0	0	0	0	0	0	0
I-B	0	2	0	**26**	535	61	4	0	0	0	2	59	11	2	60
I-I	1	1.4k	36	289	**1.8m**	844	4.9k	0	189	0	637	1.1k	5.2k	9	181.1k
I-E	0	2	0	1	2.0k	**3.7k**	142	0	1	8	18	0	62	62	812
I-O	0	0	0	0	0	0	**0**	0	0	0	0	0	0	0	0
E-B	0	1	0	0	28	0	2	**0**	1	0	19	1	0	0	4
E-I	0	0	0	0	56	0	7	0	**3**	0	40	0	0	0	4
E-E	0	0	0	0	0	0	0	0	0	**0**	0	0	0	0	0
E-O	0	0	0	0	1.4k	27	527	0	64	0	**3.6k**	0	24	0	936
O-B	2	25	1	143	1.7k	16	19	0	0	0	7	**2.8k**	259	0	1.4k
O-I	0	0	0	0	0	0	0	0	0	0	0	0	**0**	0	0
O-E	0	0	0	0	2	0	0	0	0	0	0	0	0	**0**	0
O-O	1	1.0k	149	93	505.8k	1.1k	5.5k	0	51	5	1.1k	1.1k	5.7k	70	**4.4m**

Table 6.4: Confusion matrix of consecutively predicted labels. Overall, FRIDAY exhibits robust performance, with the precise localisation of the beginning and ending of snippets being more challenging than within-snippet predictions.

The qualitative analysis reveals strengths and potential for improvement in FRIDAY's snippet identification capabilities.

As strengths, we first highlight the model's strong ability to correctly label continuous segments within or outside of snippets, learning that ”I” labels within snippets and ”O” labels outside of snippets typically follow each other in groups. In the unseen AML documents, FRIDAY correctly labels these tokens in 1.8 million and 4.4 million cases, respectively. Secondly, FRIDAY was capable of locating the beginning and end of snippets precisely to the exact token in the majority of the cases, as indicated by its predictions for the B-I, I-E, E-O, and O-B label pairs. The confusion mainly occurs with I-I or O-O in these cases and results from the challenging nature of this precise task (Section 2.2.3), sometimes leading to slightly offset snippets or entirely missed ”B” or ”E” labels.

Areas for improvement can be seen in the confusion among inner tokens. Although the overall performance of FRIDAY in this regard is very robust, the model makes notable confusions between O-O and I-I ($>$ 500k confusions) and vice versa ($>$ 180k confusions). I-O and O-I sequences can also be included here.

We argue that most confusions, such as imperfectly identified beginnings or endings and interruptions of ”I” labels by ”O” labels or vice versa, generally do not significantly impact downstream snippet creation. These issues can be effectively handled through the post-processing algorithm (Section 5.4). The *ignore_o_threshold*, for instance, mitigates confusions between I-I and O-O, O-I, or I-O. Similarly, *min_snippet_length* helps manage the opposite scenario. Additional measures could further refine snippet creation.

6.3 Summary

This chapter demonstrated that the snippet identifier system developed in this work is capable of accurately discovering FRI in legal documents. Furthermore, the system can additionally handle the auxiliary objective of matching the identified snippets with detailed labels from the RGP ontology.

Reflecting on the requirements defined in Section 2.2.4 and Section 3.4, we conclude that our **F**inancial **R**egulatory **I**nformation **D**iscovery and **A**nnotation s**Y**stem (FRIDAY) is not only the first NLP tool capable of solving the research question posed in this study, but also, in contrast to the baseline models mentioned, fulfils all the desired characteristics outlined in this report and summarised again in Table 6.5.

Desirable Features		**Explanation**	**Snip. Alg.**	**Graph-Seg**	**FRIDAY**
I	**Adaptive boundary detection**	Recognises boundaries between words, sentences, paragraphs, or pages to accommodate different forms of textual units.	□	□	☑
II	**Unrestricted snippet identification**	Identifies an unrestricted number of snippets and is not constrained to classifying every part of a text.	□	☑	☑
III	**Legal understanding**	Interprets the complex format, language, and structure of legal documents.	□	□	☑
IV	**Advanced NLP techniques**	Utilises state-of-the-art sequence models beyond handcrafted rules and features.	□	□	☑
V	**Robust training datasets**	Trained on extensive, expertly annotated datasets of financial regulations.	□	□	☑
VI	**Page-level snippet identification**	Identifies snippets page by page, even when the page length exceeds 512 tokens.	☑	☑	☑
VII	**Flexibility in snippet size**	Accommodates varying sizes and numbers of snippets on a page.	☑	☑	☑
VIII	**Broad regulatory understanding**	Accurately handles differently formatted documents from nearly a hundred jurisdictions worldwide.	□	□	☑

Table 6.5: Evaluation of different snippet identifier systems against the desired features.

As humans will continue to play a decisive role in the analysis and interpretation of legally relevant financial regulations — at least for the foreseeable future — FRIDAY can significantly enhance their efficiency and accuracy, reducing the time and effort required to process complex legal documents.

Chapter 7

Summary, conclusion, and future work

The growing complexity and volume of financial regulations pose significant challenges for financial institutions striving to maintain compliance. While the demand for RegTech to alleviate this burden is immense, the intricacies of legal language complicate legal NLP and the automated processing of financial regulatory information (FRI). To bridge this key research gap, this study explored how NLP can be applied to identify and classify text segments containing relevant FRI within structured legal documents. We introduce the **F**inancial **R**egulatory **I**nformation **D**iscovery and **A**nnotation s**Y**stem (FRIDAY) — the first end-to-end NLP system capable of discovering and classifying FRI in unseen legal documents.

7.1 Summary and conclusion

FRIDAY was developed based on a real-world dataset of 1,149 expertly annotated documents containing financial regulations in the domain of anti-money laundering from around the world. This work analysed the idiosyncrasies of these legal texts and introduced a tool for dynamic comparison of different textual components from individual pages. Additionally, we present a robust pre-processing algorithm capable of accurately identifying and refining erroneous OCR-extracted text.

During the development of FRIDAY, five novel NLP systems capable of discovering FRI in legal documents were introduced, and ten different configurations were evaluated. The most successful version of FRIDAY operates on a token level and incorporates a state-of-the-art pre-trained RoBERTa model optimised for text segmentation in a multitask learning setting. FRIDAY can be trained to discover and optionally match FRI with labels from different node levels of the Regulatory Genome Project ontology.

With ROUGE scores in the mid-80s, FRIDAY significantly outperforms baseline approaches from research and industry by nearly 37%. It also generalises well to unseen domains while maintaining strong performance. When tested on an additional 53,570 pages of financial cybersecurity regulations, FRIDAY's performance remained almost stable, with a decrease of at most 5%.

7.2 Limitations and future work

While this study has made significant contributions in the automated discovery of FRI, several limitations and potential areas for future work have been identified:

- **Extended snippet spans and context:** The systems proposed in this work were intentionally restricted to identifying snippets within individual pages to replicate the human annotation process. Yet, the same FRI can span multiple pages. While extending the model's scope would most likely require an updated annotation process, enhancing the model to consider the content, context, and structure of an entire document could improve snippet identification and classification. However, this would require architectures capable of handling significantly larger context windows and a different approach to the task in general.

- **Overlapping regions:** Due to the way FRIDAY is set up around the inherently sequential NER-inspired BIO(E)-labelling approach, the model cannot handle overlapping FRI (i.e. text segments covering more than one type of FRI). Future work may investigate and develop methods to address nested elements effectively. Here, exploring alternative labelling schemes (e.g. nested NER), or more sophisticated sequence labelling models could provide sensible starting points.

- **Improved label dependencies analysis:** FRIDAY may produce incorrect label sequences like "I-O-I", due to insufficient learning of dependencies between labels. Although relatively rare and already mitigated through post-processing, future approaches could investigate different techniques, such as incorporating a CRF layer on top of the encoder, to further improve label consistency.

- **Extended and diversified dataset:** While this study's dataset already represents a comprehensive repository of legal documents from around the world, the robustness and performance of FRIDAY could be further enhanced by adding data containing regulations from additional themes such as cryptocurrency or environmental, social and governance (ESG). Furthermore, multi-lingual versions of the system could be explored. A larger and more diverse dataset may enable smaller models to achieve results comparable to or better than those presented in this study.

- **Further investigation of the auxiliary objective and its integration:** While this study investigated matching snippets with ontology labels (auxiliary objective), it focused on discovering FRI in legal documents and optimising models for this purpose. Therefore, future work may concentrate on optimising the auxiliary objective of classifying the discovered FRI. Since this study only integrated detailed labels from the AML theme, investigating FRIDAY's generalisation capabilities for detailed labels across different themes, such as cybersecurity, would be valuable. Furthermore, the integration of both losses warrants further exploration. Different integration techniques and the introduction of regularisation terms are conceivable and could be used to constrain the objectives in beneficial ways.

- **Improved sentence-level models:** The inferior performance of the sentence-level model compared to the token-level approaches may partly be attributed to the accuracy of the underlying sentence splits. Improving these splits could significantly enhance sentence-level systems. Additionally, alternative embedding models could be explored to obtain more accurate sentence embeddings and improve overall performance.
- **Further pre-processing optimisations:** Although the custom pre-processing algorithm (Chapter 4) handles various OCR mismatches, it is not faultless and occasionally makes wrong refinements that unintentionally introduce noise into the training data. Further optimisations of this algorithm or upstream changes ensuring cleaner training data, such as improved annotation processes, could further improve FRIDAY's performance on the downstream task.

7.3 Implications

FRIDAY significantly advances research in the field of legal NLP and RegTech by addressing the unique challenges of automatically and accurately processing financial regulatory documents. With FRIDAY, we introduce a first-of-its-kind system: an NLP tool designed to discover FRI in unseen legal documents. This tool integrates techniques from text segmentation, text zoning, and sentence boundary detection. FRIDAY offers a ready-to-use practical solution for regulatory compliance, enhancing the ability of a wide spectrum of stakeholders like financial institutions and regulatory bodies to automatically process and interpret FRI. The system can streamline compliance efforts, reduce manual labour and increase the overall accuracy of regulatory analysis and compliance with financial regulations.

Bibliography

[1] Y. Ajjour, W.-F. Chen, J. Kiesel, H. Wachsmuth, and B. Stein, “Unit Segmentation of Argumentative Texts,” in *Proceedings of the 4th Workshop on Argument Mining*, I. Habernal *et al.*, Eds., Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 118–128. DOI: 10.18653/v1/W17-5115.

[2] B. Alex, B. Haddow, and C. Grover, “Recognising Nested Named Entities in Biomedical Text,” in *Biological, translational, and clinical language processing*, K. B. Cohen, D. Demner-Fushman, C. Friedman, L. Hirschman, and J. Pestian, Eds., Prague, Czech Republic: Association for Computational Linguistics, Jun. 2007, pp. 65–72. [Online]. Available: https://aclanthology.org/W07-1009.

[3] Z. Amadxarif, J. Brookes, N. Garbarino, R. Patel, and E. Walczak, “The Language of Rules: Textual Complexity in Banking Reforms,” *SSRN Electronic Journal*, 2019. DOI: 10.2139/ssrn.3475418.

[4] D. Aumiller, S. Almasian, S. Lackner, and M. Gertz, “Structural Text Segmentation of Legal Documents,” in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, Jun. 2021, pp. 2–11. DOI: 10.1145/3462757.3466085.

[5] D. Baiamonte, T. Caselli, and I. Prodanof, “Annotating Content Zones in News Articles,” in *Proceedings of the Third Italian Conference on Computational Linguistics CLiC-it 2016*, A. Corazza, S. Montemagni, and G. Semeraro, Eds., Accademia University Press, 2016, pp. 40–45. DOI: 10.4000/books.aaccademia.1695.

[6] “Basel III: International regulatory framework for banks,” Dec. 2017. [Online]. Available: https://www.bis.org/bcbs/basel3.htm (visited on 04/14/2024).

[7] D. Beeferman, A. Berger, and J. Lafferty, “Statistical Models for Text Segmentation,” *Machine Learning*, vol. 34, no. 1, pp. 177–210, Feb. 1999. DOI: 10.1023/A:1007506220214.

[8] M. J. Bommarito, D. M. Katz, and E. M. Detterman, “LexNLP: Natural language processing and information extraction for legal and regulatory texts,” in *Research Handbook on Big Data Law*, Section: Research Handbook on Big Data Law, Edward Elgar Publishing, May 2021, pp. 216–227, ISBN: 978-1-78897-282-6.

[9] T. Brants, F. Chen, and I. Tsochantaridis, “Topic-based document segmentation with probabilistic latent semantic analysis,” in *Proceedings of the eleventh international conference on Information and knowledge management*, ser. CIKM ’02, New

York, NY, USA: Association for Computing Machinery, Nov. 2002, pp. 211–218. DOI: 10.1145/584792.584829.

[10] T. Brugger, M. Stürmer, and J. Niklaus, *MultiLegalSBD: A Multilingual Legal Sentence Boundary Detection Dataset*, May 2023. DOI: 10.48550/arXiv.2305.01211.

[11] Cambridge Regulatory Genome Project, *Regulator engagement: Building an RGP regulator community - Regulatory Genome Project.* [Online]. Available: https://www.jbs.cam.ac.uk/faculty-research/centres/regulatory-genome-project/rgp-regulator-engagement/ (visited on 04/27/2024).

[12] Cambridge Regulatory Genome Project, *Regulatory Genome Project - Centres and initiatives.* [Online]. Available: https://www.jbs.cam.ac.uk/faculty-research/centres/regulatory-genome-project/ (visited on 11/12/2023).

[13] J. Carletta, "Assessing Agreement on Classification Tasks: The Kappa Statistic," *Computational Linguistics*, vol. 22, no. 2, J. Hirschberg, Ed., pp. 249–254, 1996. [Online]. Available: https://aclanthology.org/J96-2004.

[14] S. Casola, I. Lauriola, and A. Lavelli, "Pre-trained transformers: An empirical comparison," *Machine Learning with Applications*, vol. 9, p. 100 334, Sep. 2022. DOI: 10.1016/j.mlwa.2022.100334.

[15] S. Castano *et al.*, "Enforcing legal information extraction through context-aware techniques: The ASKE approach," *Computer Law & Security Review*, vol. 52, p. 105 903, Oct. 2023. DOI: 10.1016/j.clsr.2023.105903.

[16] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "LEGAL-BERT: The Muppets straight out of Law School," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 2898–2904. DOI: 10.18653/v1/2020.findings-emnlp.261.

[17] I. Chalkidis, N. Garneau, C. Goanta, D. M. Katz, and A. Søgaard, *LeXFiles and LegalLAMA: Facilitating English Multinational Legal Language Model Development*, May 2023. DOI: 10.48550/arXiv.2305.07507.

[18] H. Chen, S. Branavan, R. Barzilay, and D. R. Karger, "Global Models of Document Structure using Latent Permutations," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, M. Ostendorf, M. Collins, S. Narayanan, D. W. Oard, and L. Vanderwende, Eds., Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 371–379. [Online]. Available: https://aclanthology.org/N09-1042.

[19] F. Choi, "Advances in domain independent linear text segmentation," in *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, 2000. [Online]. Available: https://aclanthology.org/A00-2004.

[20] F. Choi, P. Hastings, and J. Moore, "Latent Semantic Analysis for Text Segmentation," *Proceedings of EMNLP*, 2001. [Online]. Available: https://aclanthology.org/W01-0514.

[21] Clifford Chance, "Globalisation and Financial Regulation: Challenges and Trends," Clifford Chance, London, Tech. Rep., Oct. 2019. [Online]. Available: https://www.cliffordchance.com/content/dam/cliffordchance/briefings/2019/10/globalisation-and-financial-regulation-challenges-and-trends.pdf (visited on 05/19/2024).

[22] Clifford Chance, "Cyber security - what regulators are saying around the world," Tech. Rep., Dec. 2020. [Online]. Available: https://www.cliffordchance.com/content/dam/cliffordchance/briefings/2018/06/cyber-security-what-regulators-are-saying-around-the-world.pdf (visited on 05/29/2024).

[23] M. Collins, "Log-Linear Models, MEMMs, and CRFs,"

[24] J. Cui, X. Shen, and S. Wen, "A Survey on Legal Judgment Prediction: Datasets, Metrics, Models and Challenges," *IEEE Access*, vol. 11, pp. 102 050–102 071, 2023. DOI: 10.1109/ACCESS.2023.3317083.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, May 2019. DOI: 10.48550/arXiv.1810.04805.

[26] S. Dharanipragada, M. Franz, J. McCarley, S. Roukos, and T. Ward, *Story segmentation and topic detection for recognized speech*. Sep. 1999. DOI: 10.21437/Eurospeech.1999-535.

[27] *Dodd-Frank Act — CFTC*. [Online]. Available: https://www.cftc.gov/LawRegulation/DoddFrankAct/index.htm (visited on 04/14/2024).

[28] J. Eisenstein, "Hierarchical Text Segmentation from Multi-Scale Lexical Cohesion," in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, M. Ostendorf, M. Collins, S. Narayanan, D. W. Oard, and L. Vanderwende, Eds., Boulder, Colorado: Association for Computational Linguistics, Jun. 2009, pp. 353–361. [Online]. Available: https://aclanthology.org/N09-1040.

[29] S. English and S. Hammond, "Fintech, Regtech and the Role of Compliance in 2019," Thomson Reuters, Tech. Rep., 2019.

[30] L. Ermakova, J. V. Cossu, and J. Mothe, "A survey on evaluation of summarization methods," *Information Processing & Management*, vol. 56, no. 5, pp. 1794–1814, Sep. 2019. DOI: 10.1016/j.ipm.2019.04.001.

[31] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From Data Mining to Knowledge Discovery in Databases," *AI Magazine*, vol. 17, no. 3, 1996. [Online]. Available: https://www.mat.ucsb.edu/~g.legrady/academic/courses/17w259/KDD.pdf.

[32] S. Fernández, A. Graves, and J. Schmidhuber, "Sequence labelling in structured domains with hierarchical recurrent neural networks," in *Proceedings of the 20th in-*

ternational joint conference on Artifical intelligence, ser. IJCAI'07, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jan. 2007, pp. 774–779. (visited on 05/28/2024).

[33] M. Galley, K. R. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse Segmentation of Multi-Party Conversation," in *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan: Association for Computational Linguistics, Jul. 2003, pp. 562–569. DOI: 10.3115/1075096.1075167.

[34] D. Ganguly *et al.*, "Legal IR and NLP: The History, Challenges, and State-of-the-Art," in *Advances in Information Retrieval*, J. Kamps *et al.*, Eds., ser. Lecture Notes in Computer Science, Cham: Springer Nature Switzerland, 2023, pp. 331–340. DOI: 10.1007/978-3-031-28241-6_34.

[35] F. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3, Jul. 2000, 189–194 vol.3. DOI: 10.1109/IJCNN.2000.861302.

[36] F. A. Gers, J. A. Schmidhuber, and F. A. Cummins, "Learning to Forget: Continual Prediction with LSTM," *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, Oct. 2000. DOI: 10.1162/089976600300015015.

[37] G. Glavaš, F. Nanni, and S. P. Ponzetto, "Unsupervised Text Segmentation Using Semantic Relatedness Graphs," in *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, C. Gardent, R. Bernardi, and I. Titov, Eds., Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 125–130. DOI: 10.18653/v1/S16-2016.

[38] G. Glavaš and S. Somasundaran, *Two-Level Transformer and Auxiliary Coherence Modeling for Improved Text Segmentation*, Jan. 2020. [Online]. Available: http://arxiv.org/abs/2001.00891.

[39] A.-S. Gnehm, "Text zoning for job advertisements with bidirectional LSTMs," Jun. 2018. DOI: 10.5167/UZH-186646.

[40] A.-S. Gnehm and S. Clematide, "Text Zoning and Classification for Job Advertisements in German, French and English," in *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, D. Bamman, D. Hovy, D. Jurgens, B. O'Connor, and S. Volkova, Eds., Online: Association for Computational Linguistics, Nov. 2020, pp. 83–93. DOI: 10.18653/v1/2020.nlpcss-1.10.

[41] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Networks*, IJCNN 2005, vol. 18, no. 5, pp. 602–610, Jul. 2005. DOI: 10.1016/j.neunet.2005.06.042.

[42] C. Grover, B. Hachey, and C. Korycinski, "Summarising legal texts: Sentential tense and argumentative roles," in *Proceedings of the HLT-NAACL 03 on Text*

summarization workshop -, vol. 5, Not Known: Association for Computational Linguistics, 2003, pp. 33–40. DOI: `10.3115/1119467.1119472`.

[43] M. A. Hearst, "Multi-Paragraph Segmentation Expository Text," in *32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico, USA: Association for Computational Linguistics, Jun. 1994, pp. 9–16. DOI: `10.3115/981732.981734`.

[44] H. Hettiarachchi, M. Adedoyin-Olowe, J. Bhogal, and M. Gaber, *DAAI at CASE 2021 Task 1: Transformer-based Multilingual Socio-political and Crisis Event Detection*. Jan. 2021.

[45] K. Hirohata, N. Okazaki, S. Ananiadou, and M. Ishizuka, "Identifying Sections in Scientific Abstracts using Conditional Random Fields," in *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, 2008. [Online]. Available: `https://aclanthology.org/I08-1050`.

[46] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. DOI: `10.1162/neco.1997.9.8.1735`.

[47] Z. Huang, W. Xu, and K. Yu, *Bidirectional LSTM-CRF Models for Sequence Tagging*, Aug. 2015. [Online]. Available: `http://arxiv.org/abs/1508.01991`.

[48] A. S. Imran, H. Hodnefjeld, Z. Kastrati, N. Fatima, S. M. Daudpota, and M. A. Wani, "Classifying European Court of Human Rights Cases Using Transformer-Based Techniques," *IEEE Access*, vol. 11, pp. 55 664–55 676, 2023. DOI: `10.1109/ACCESS.2023.3279034`.

[49] A. Jaiswal and E. Milios, *Breaking the Token Barrier: Chunking and Convolution for Efficient Long Text Classification with BERT*, Oct. 2023. DOI: `10.48550/arXiv.2310.20558`.

[50] B. Jardim, R. Rei, and M. S. C. Almeida, *Multilingual Email Zoning*, Feb. 2021. DOI: `10.48550/arXiv.2102.00461`.

[51] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Jul. 2023.

[52] M.-Y. Kan, J. L. Klavans, and K. R. McKeown, "Linear Segmentation and Segment Significance," p. 1998,

[53] D. M. Katz, D. Hartung, L. Gerlach, A. Jana, and M. J. Bommarito II, *Natural Language Processing in the Legal Domain*, Feb. 2023. DOI: `10.48550/arXiv.2302.12039`.

[54] A. Kehagias, P. Fragkou, and V. Petridis, "Linear Text Segmentation using a Dynamic Programming Algorithm," in *10th Conference of the European Chapter of the Association for Computational Linguistics*, A. Copestake and J. Hajič, Eds., Budapest, Hungary: Association for Computational Linguistics, Apr. 2003. [Online]. Available: `https://aclanthology.org/E03-1058`.

[55] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023. DOI: `10.1007/s11042-022-13428-4`.

[56] T. Kiss and J. Strunk, "Unsupervised Multilingual Sentence Boundary Detection," *Computational Linguistics*, vol. 32, no. 4, pp. 485–525, 2006. DOI: 10.1162/coli.2006.32.4.485.

[57] O. Koshorek, A. Cohen, N. Mor, M. Rotman, and J. Berant, *Text Segmentation as a Supervised Learning Task*, Mar. 2018. DOI: 10.48550/arXiv.1803.09337.

[58] KPMG, "There's a revolution coming," Tech. Rep., 2018. [Online]. Available: https://assets.kpmg.com/content/dam/kpmg/uk/pdf/2018/09/regtech-revolution-coming.pdf (visited on 04/07/2024).

[59] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology.* SAGE Publications, Inc., 2019. DOI: 10.4135/9781071878781.

[60] L. Labeis, *Regtech is growing – but what next for the sector?* Feb. 2023. [Online]. Available: https://fintechmagazine.com/articles/regtech-is-growing-but-what-next-for-the-sector (visited on 05/02/2024).

[61] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., Jun. 2001, pp. 282–289, ISBN: 978-1-55860-778-1.

[62] A. Lampert, R. Dale, and C. Paris, "Segmenting email message text into zones," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*, ser. EMNLP '09, USA: Association for Computational Linguistics, Aug. 2009, pp. 919–928, ISBN: 978-1-932432-62-6.

[63] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, *Neural Architectures for Named Entity Recognition*, Apr. 2016. DOI: 10.48550/arXiv.1603.01360.

[64] S. Lawless and M. M. Bayomi, "C-HTS: A Concept-based Hierarchical Text Segmentation Approach," 2018. [Online]. Available: http://www.tara.tcd.ie/handle/2262/86849.

[65] E. Leitner, G. Rehm, and J. Moreno-Schneider, "Fine-Grained Named Entity Recognition in Legal Documents," in *Semantic Systems. The Power of AI and Knowledge Graphs*, M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack, and Y. Sure-Vetter, Eds., Cham: Springer International Publishing, 2019, pp. 272–287. DOI: 10.1007/978-3-030-33220-4_20.

[66] N. Limsopatham and N. Collier, "Bidirectional LSTM for Named Entity Recognition in Twitter Messages," in *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, B. Han, A. Ritter, L. Derczynski, W. Xu, and T. Baldwin, Eds., Osaka, Japan: The COLING 2016 Organizing Committee, Dec. 2016, pp. 145–152. [Online]. Available: https://aclanthology.org/W16-3920.

[67] C.-Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *Text Summarization Branches Out*, Barcelona, Spain: Association for Computational

Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013.

[68] Y. Liu *et al.*, *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, Jul. 2019. DOI: 10.48550/arXiv.1907.11692.

[69] A. Lyte and K. Branting, "Document Segmentation Labeling Techniques for Court Filings," Jun. 2019.

[70] U. B. Mahadevaswamy and P. Swathi, "Sentiment Analysis using Bidirectional LSTM Network," *Procedia Computer Science*, International Conference on Machine Learning and Data Engineering, vol. 218, pp. 45–56, Jan. 2023. DOI: 10.1016/j.procs.2022.12.400.

[71] M. Maignant, T. Poibeau, and G. Brison, "Text Zoning of Theater Reviews: How Different are Journalistic from Blogger Reviews?" In *Proceedings of the Workshop on Natural Language Processing for Digital Humanities*, M. Hämäläinen, K. Alnajjar, N. Partanen, and J. Rueter, Eds., NIT Silchar, India: NLP Association of India (NLPAI), Dec. 2021, pp. 138–143. [Online]. Available: https://aclanthology.org/2021.nlp4dh-1.16.

[72] V. Malik *et al.*, *Semantic Segmentation of Legal Documents via Rhetorical Roles*, Nov. 2022. DOI: 10.48550/arXiv.2112.01836.

[73] *MiFID II — European Securities and Markets Authority.* [Online]. Available: https://www.esma.europa.eu/publications-and-data/interactive-single-rulebook/mifid-ii (visited on 04/14/2024).

[74] T. Mikolov, K. Chen, G. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*, Sep. 2013. DOI: 10.48550/arXiv.1301.3781.

[75] H. Misra, F. Yvon, J. Jose, and O. Cappé, "Text segmentation: A topic modeling perspective," *Information Processing and Management*, vol. 47, pp. 528–544, 2011. DOI: 10.1016/j.ipm.2010.11.008.

[76] Narcotics Division, *An Advisory Guideline on Preventing the Misuse of Charities for Terrorist Financing*, Sep. 2018. [Online]. Available: https://www.sb.gov.hk/eng/special/moneylaundering/index.html.

[77] L. Pevzner and M. A. Hearst, "A Critique and Improvement of an Evaluation Metric for Text Segmentation," *Computational Linguistics*, vol. 28, no. 1, pp. 19–36, Mar. 2002. DOI: 10.1162/089120102317341756.

[78] S. Pugliese, "Divergences between EU and US in the Financial Regulation: What Effects on the TTIP Negotiations?" *European Journal of Risk Regulation*, vol. 7, no. 2, pp. 285–289, Jun. 2016. DOI: 10.1017/S1867299X00005699.

[79] PyTorch, *Advanced: Making Dynamic Decisions and the Bi-LSTM CRF — PyTorch Tutorials 2.2.2+cu121 documentation.* [Online]. Available: https://pytorch.org/tutorials/beginner/nlp/advanced_tutorial.html (visited on 04/16/2024).

[80] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018.

[81] D. C. for Regulatory Strategy, “Regulatory Outlook 2024,” Tech. Rep., 2024. [Online]. Available: https://www2.deloitte.com/uk/en/pages/financial-services/articles/regulatory-outlook.html (visited on 04/14/2024).

[82] N. Reimers and I. Gurevych, *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*, Aug. 2019. DOI: 10.48550/arXiv.1908.10084.

[83] M. Riedl and C. Biemann, “TopicTiling: A Text Segmentation Algorithm based on LDA,” in *Proceedings of ACL 2012 Student Research Workshop*, J. C. K. Cheung, J. Hatori, C. Henriquez, and A. Irvine, Eds., Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 37–42. [Online]. Available: https://aclanthology.org/W12-3307.

[84] R. Romano, *Regulating in the Dark*, SSRN Scholarly Paper, Rochester, NY, Mar. 2012. DOI: 10.2139/ssrn.1974148.

[85] A. B. Sai, A. K. Mohankumar, and M. M. Khapra, “A Survey of Evaluation Metrics Used for NLG Systems,” *ACM Computing Surveys*, vol. 55, no. 2, 26:1–26:39, Jan. 2022. DOI: 10.1145/3485766.

[86] G. Sanchez, “Sentence Boundary Detection in Legal Text,” in *Proceedings of the Natural Legal Language Processing Workshop 2019*, N. Aletras *et al.*, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 31–38. DOI: 10.18653/v1/W19-2204.

[87] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter*, Feb. 2020. DOI: 10.48550/arXiv.1910.01108. (visited on 05/29/2024).

[88] M. Saravanan and B. Ravindran, “Identification of Rhetorical Roles for Segmentation and Summarization of a Legal Judgment,” *Artificial Intelligence and Law*, vol. 18, no. 1, pp. 45–76, Mar. 2010. DOI: 10.1007/s10506-010-9087-7.

[89] V. Satopää, J. Albrecht, D. Irwin, and B. Raghavan, “Finding a ”Kneedle” in a Haystack: Detecting Knee Points in System Behavior,” in *2011 31st International Conference on Distributed Computing Systems Workshops*, Minneapolis, MN, USA: IEEE, Jun. 2011, pp. 166–171. DOI: 10.1109/ICDCSW.2011.20.

[90] J. Savelka, V. R. Walker, M. Grabmair, and K. D. Ashley, “Sentence Boundary Detection in Adjudicatory Decisions in the United States,” *Traitement Automatique des Langues*, vol. 58, no. 2, A. Nazarenko and A. Wyner, Eds., pp. 21–45, 2017. [Online]. Available: https://aclanthology.org/2017.tal-2.2.

[91] SBERT, *Pretrained Models — Sentence Transformers documentation.* [Online]. Available: https://sbert.net/docs/sentence_transformer/pretrained_models.html (visited on 06/01/2024).

[92] E. Schubert, “Stop using the elbow criterion for k-means and how to choose the number of clusters instead,” *ACM SIGKDD Explorations Newsletter*, vol. 25, no. 1, pp. 36–42, Jul. 2023. DOI: 10.1145/3606274.3606278.

[93] *Set space threshold to decide text blocks · pymupdf/PyMuPDF · Discussion #1358.* [Online]. Available: https://github.com/pymupdf/PyMuPDF/discussions/1358 (visited on 05/02/2024).

[94] R. Sheik, S. R. Ganta, and S. J. Nirmala, "Legal sentence boundary detection using hybrid deep learning and statistical models," *Artificial Intelligence and Law*, Mar. 2024. DOI: 10.1007/s10506-024-09394-x.

[95] R. Sheik, G. T, and S. Nirmala, "Efficient Deep Learning-based Sentence Boundary Detection in Legal Text," in *Proceedings of the Natural Legal Language Processing Workshop 2022*, N. Aletras, I. Chalkidis, L. Barrett, C. Goan\textcommabelowtă, and D. Preo\textcommabelowtiuc-Pietro, Eds., Abu Dhabi, United Arab Emirates (Hybrid): Association for Computational Linguistics, Dec. 2022, pp. 208–217. DOI: 10.18653/v1/2022.nllp-1.18.

[96] S. Siegel, *Nonparametric statistics for the behavioral sciences* (Nonparametric statistics for the behavioral sciences). New York, NY, US: McGraw-Hill, 1956.

[97] R. Sil, A. Roy, B. Bhushan, and A. Mazumdar, "Artificial Intelligence and Machine Learning based Legal Application: The State-of-the-Art and Future Research Trends," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Oct. 2019, pp. 57–62. DOI: 10.1109/ICCCIS48478.2019.8974479.

[98] Q. Sun, R. Li, D. Luo, and X. Wu, "Text Segmentation with LDA-Based Fisher Kernel," in *Proceedings of ACL-08: HLT, Short Papers*, J. D. Moore, S. Teufel, J. Allan, and S. Furui, Eds., Columbus, Ohio: Association for Computational Linguistics, Jun. 2008, pp. 269–272. [Online]. Available: https://aclanthology.org/P08-2068.

[99] I. Sutskever, O. Vinyals, and Q. V. Le, *Sequence to Sequence Learning with Neural Networks*, Dec. 2014. [Online]. Available: http://arxiv.org/abs/1409.3215.

[100] E. G. R. N. executive team, "EY Global financial services regulatory outlook 2023," Tech. Rep., 2023. [Online]. Available: https://www.ey.com/en_uk/financial-services/regulatory-outlook (visited on 04/14/2024).

[101] S. Teufel, "Argumentative Zoning: Information Extraction from Scientific Text," PhD, University of Edinburgh, 1999.

[102] S. Teufel, *Argumentative Zoning.* [Online]. Available: https://www.cl.cam.ac.uk/~sht25/az.html (visited on 04/19/2024).

[103] V. U. Thompson, C. Panchev, and M. Oakes, "Performance evaluation of similarity measures on similar and dissimilar text retrieval," in *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*, vol. 01, Nov. 2015, pp. 577–584. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7526978 (visited on 05/24/2024).

[104] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, Dec. 1953. DOI: 10.1007/BF02289263.

[105] M. Utiyama and H. Isahara, "A Statistical Model for Domain-Independent Text Segmentation," in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France: Association for Computational Linguistics, Jul. 2001, pp. 499–506. DOI: 10.3115/1073012.1073076.

[106] A. Vaswani *et al.*, *Attention Is All You Need*, Jun. 2017. DOI: 10.48550/arXiv.1706.03762. (visited on 12/28/2023).

[107] Y. Yaari, *Segmentation of Expository Texts by Hierarchical Agglomerative Clustering*, Sep. 1997. DOI: 10.48550/arXiv.cmp-lg/9709015.

[108] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019. DOI: 10.1162/neco_a_01199.

[109] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, and J. Mylopoulos, "GaiusT: Supporting the extraction of rights and obligations for regulatory compliance," *Requirements Engineering*, vol. 20, no. 1, pp. 1–22, Mar. 2015. DOI: 10.1007/s00766-013-0181-8.

[110] H. Zhang and M. O. Shafiq, "Survey of transformers and towards ensemble learning using transformers for natural language processing," *Journal of Big Data*, vol. 11, no. 1, p. 25, Feb. 2024. DOI: 10.1186/s40537-023-00842-0.

Appendix A

Relevant machine learning techniques

A.1 Long Short-Term Memory (LSTM)

LSTMs, developed by Hochreiter and Schmidhuber [46] in 1997 and iteratively optimised in the following years [108], are advanced recurrent neural networks (RNNs) [51]. Unlike basic feed-forward neural networks, RNNs chain nodes to handle sequential inputs of varying lengths, making them effective for NLP tasks like language modelling, sequence classification and language generation [51]. Conventional RNNs, however, are unable to retain long-term dependencies as the sequential computing process leads to gradients becoming significantly large or small over time, rendering training of RNNs for long-range tasks extremely difficult or even impossible (*exploding/vanishing gradient problem*) [46].

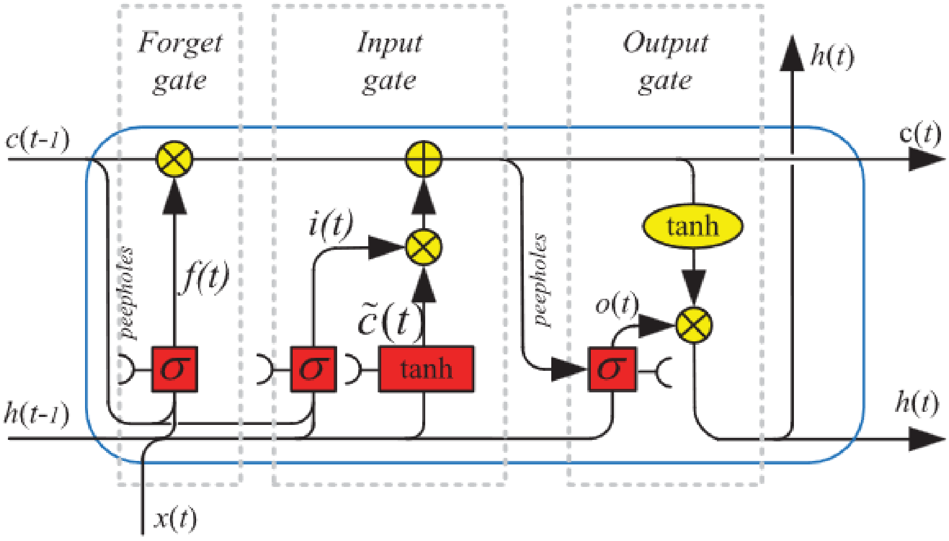


Figure A.1: Structure of a typical LSTM layer with three gates and peephole connections. Figure taken from Yu *et al.* [108].

A.1.1 Core Architecture

LSTMs address the challenge of retaining long-range dependencies by introducing an explicit context layer for extended memory [51, 108]. Each LSTM cell includes both short-term and long-term memory paths, allowing the model to learn what relevant information to retain over time while forgetting unnecessary details [46]. Research shows that LSTMs are effective for processing sequential data and often achieve state-of-the-art performance in numerous NLP tasks [40, 50].

Figure A.1 illustrates the unique architecture of LSTM [46] networks based on three *gates*: forget, input, and output gates, which regulate the flow of information [108]. Each gate within the LSTM unit operates based on a combination of the current input x_t, the previous output h_{t-1} (short-term memory), and the previous cell state c_{t-1} (long-term memory) [108]. Furthermore, peephole connections P can be added to each gate, enabling LSTMs to consider c_{t-1} as additional context for their gate decisions [35].

The forget gate of an LSTM determines what percentage of the long-term memory is discarded [36]. It takes the previous output h_{t-1} and the current input x_t as arguments and applies a sigmoid function σ to it, which results in values between 0 and 1 [108]. The resulting value f_t is subsequently multiplied by the previous cell state to determine what "long-term memory" to retain: $f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + P_f c_{t-1} + b_f)$ [108].

The input gate controls the addition of new information to the cell state (i.e. it updates the long-term memory) [46]. It consists of a sigmoid layer and a tanh layer [108]. While the latter creates a potential long-term memory, the sigmoid layer determines what percentage of that potential memory to add to the long-term memory [108]:

$$\begin{aligned} i_t &= \sigma(W_{ih}h_{t-1} + W_{ix}x_t + P_i c_{t-1} + b_i) \\ \tilde{c}_t &= \tanh(W_{ch}h_{t-1} + W_{cx}x_t + b_c) \end{aligned} \tag{A.1}$$

Together with the forget gate, the input gate creates the updated cell state $c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t$ [108]. The final gate, the output gate, determines the next hidden state h_t, which is passed to the next LSTM cell or used for predictions [46]. It is the product of a sigmoid and tanh function and contains information based on the last cell's hidden state h_{t-1}, the input x_t as well as the updated cell state c_t [108]:

$$\begin{aligned} o_t &= \sigma(W_{oh}h_{t-1} + W_{ox}x_t + P_o c_t + b_o) \\ h_t &= o_t \cdot \tanh(c_t) \end{aligned} \tag{A.2}$$

The components above make up one LSTM layer [108]. To improve performance, these layers can be stacked, where the input $x_t^{(l)}$ for the l-th layer (where $l \geq 2$) is derived from the hidden state $h_t^{(l-1)}$ of the preceding layer, adjusted by a dropout factor $\delta_t^{(l-1)}$ [32, 108].

In the ready-to-use version of an "unrolled" LSTM (i.e. many (stacked) LSTM cells are chained together), each cell can then be used to process one input of a sequence [32, 108].

A.1.2 Bidirectional LSTMs

Conventional RNNs and LSTMs capture only previous context [41]. To utilise both past and future contexts, bidirectional RNNs and, subsequently, Bi-LSTMs were introduced [41]. Research shows that Bi-LSTMs often outperform unidirectional LSTMs in sequence labelling tasks where understanding both contexts can provide crucial information (e.g. NER) [47, 66, 70].

In contrast to unidirectional LSTMs, Bi-LSTMs consist of two LSTM layers that process the input sequence in opposite directions (forward and backward) [41]. Following Yu *et al.* [108], this architecture is represented mathematically as follows:

$$\overrightarrow{h_t} = \overrightarrow{o}_t^L \cdot tanh(\overrightarrow{h}_t^L), \quad \overleftarrow{h_t} = \overleftarrow{o}_t^L \cdot tanh(\overleftarrow{h}_t^L) \tag{A.3}$$

At each time step t, the outputs from the forward and backward pass can be combined through, for example, concatenation, simple addition, or a linear transformation of both hidden states: $y_t = W_{\overrightarrow{h}y}\overrightarrow{h_t} + W_{\overleftarrow{h}y}\overleftarrow{h_t} + b_y$ [108].

A.1.3 Integration with Conditional Random Fields (CRFs)

Conditional Random Fields (CRFs), introduced by Lafferty *et al.* [61] in 2001, are often employed with LSTMs to enhance their capabilities, particularly in structured prediction tasks where the relationship between labels is crucial [39, 47, 50]. CRFs provide a probabilistic method to model the dependencies between labels and allow predictions that consider the context of the entire sequence [61]. Research shows that (Bi-)LSTM+CRF architectures often outperform conventional (Bi-)LSTM-based approaches in tasks with interdependent labels, such as TZ, NER, and other sequence labelling tasks [47, 50].

Lafferty *et al.* [61] define a Conditional Random Field (CRF) as follows:

"**Definition.** Let $G = (V, E)$ be a graph such that $Y = (Y_v)_{v \in V}$, so that Y is indexed by the vertices of G. Then (X, Y) is a *conditional random field* in case, when conditioned on X, the random variables Y_v obey the Markov property with respect to the graph: $p(Y_v \mid X, Y_w, w \neq v) = p(Y_v \mid X, Y_w, w \sim v)$, where $w \sim v$ means that w and v are neighbors in G."

When combined with CRFs, LSTMs are used to generate latent feature representations of the input data [79]. Instead of directly using the LSTM's output for predictions, they are fed into a CRF layer (i.e. they provide the features for a CRF model) [50, 79]:

$$P(y|x) = \frac{\exp(\sum_i \phi(x, y_i, y_{i-1}))}{\sum_{y'} \exp(\sum_i \phi(x, y'_i, y'_{i-1}))} \tag{A.4}$$

Here, ϕ represents a feature vector derived from LSTM outputs, and y_i are the labels for each element in the sequence [23]. Research has shown that the combination of LSTMs with CRFs sequence models usually provides strong performance in NER and other sequence-prediction tasks [40, 69, 79].

A.2 Transformer

The transformer architecture, introduced by Vaswani *et al.* [106] in 2017, represents a significant milestone and shift away from the previously dominant RNN-based architectures in sequence processing tasks. It has significantly improved performance across a wide range of NLP tasks, often setting a new state-of-the-art, and consequently making transformers a highly popular NLP architecture [14, 55, 110]. Through its *attention* mechanism, transformers are able to capture long-range dependencies and contexts without being limited by long training times like RNNs and LSTMs [106]. Unlike inherently sequential architectures, transformers leverage attention to process all inputs simultaneously, which allows them to benefit from a global context [14, 106]. This not only enhances performance but also allows for high parallelism and scalability, enabling training on large datasets [106].

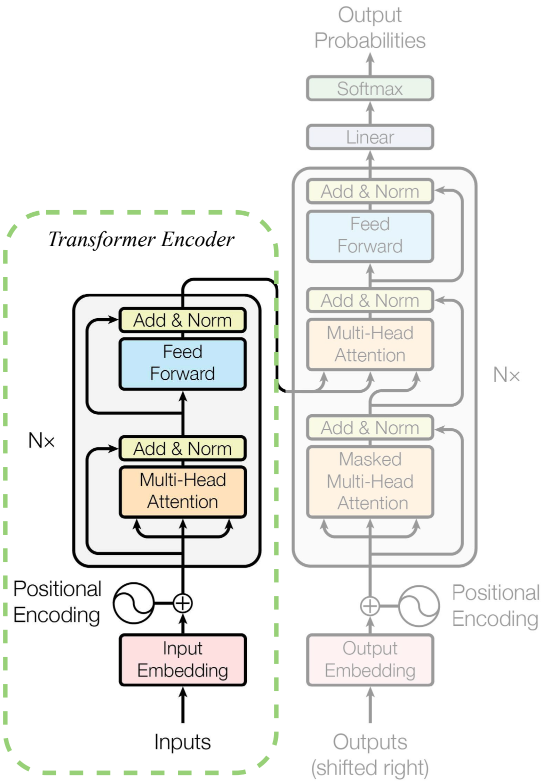


Figure A.2: Original transformer architecture. Figure taken from Vaswani *et al.* [106] and encoder highlighted.

A.2.1 Core architecture

The original transformer model, shown in Figure A.2, uses an *encoder-decoder* architecture typical for *sequence-to-sequence* models [99]. However, as the decoder is used for language generation, it is not relevant for this study [106].

The transformer processes context-enriched tokenized input, usually *word embeddings* [74] enhanced by *positional encoding*, through its encoder [106]. This combination of word embeddings and positional encoding produces a high-dimensional vector representation of each token's semantic meaning and position in the input [106]. Specifically, the input consists of the sum of word embeddings and positional encoding for each token [106]. The original transformer architecture relies on alternating sine and cosine functions to add information about the position of each token in the input to the word embeddings:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{A.5}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{A.6}$$

where pos is the position of the token in the input sequence, i is the index of the dimension within the positional encoding vector, and d_{model} is the total dimension of the model, with $d_{model} = 512$ in the original transformer [106]. This combination of parameters ensures that the positional encoding remains unique for each position in the sequence. In the absence of techniques like recurrence, positional encoding ensures that the transformer can consider the position of a token in the input for its predictions [106].

To transform the input, the encoder first calculates *Query* (Q), *Key* (K), and *Value* (V) matrices for each token as linear transformations of the input matrix (i.e. all word embeddings with positional encoding added) [106]. These matrices are combined into the *Scaled Dot-Product Attention* as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{A.7}$$

where d_k represents the dimensionality of the key vectors [106]. The obtained attention values represent how similar each input token is to all other tokens, including itself [106]. As the weights used to calculate Q, K, and V are the same for all words, all matrices and attention values can be computed in parallel, significantly speeding up operations and highlighting the contrast to the sequential processing of traditional RNN and LSTM architectures [106]. Moreover, the above attention *head* can be duplicated and different attention values calculated in parallel for h linear transformed versions of Q, K, V to obtain multiple different contextualisations of the input [106]. According to Vaswani *et al.* [106], this *Multi-Head Attention* can be represented as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) \tag{A.8}$$

where

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{A.9}$$

Here, W_i^Q, W_i^K, and W_i^V are the learned projection matrices for the i-th attention head [106]. In the original transformer, $h = 8$ heads are employed in parallel [106].

The Multi-Head Attention output is consumed by a simple feed-network, which generates the final encoder outputs [106]. The authors also added residual connections with subsequent layer normalisation around both sublayers, the Multi-Head Attention and the feed-forward network, to preserve the original input information. Hence, each sublayer's output can be described as $LayerNorm(x + Sublayer(x))$ [106].

The full encoder component can be stacked as identical layers, with the original transformer employing six encoder layers [106].

A.2.2 Pre-trained transformer models

Pre-trained transformer models have revolutionised NLP, significantly improving the efficiency and accuracy of modern NLP techniques [110]. These models follow a two-step training process: *pre-training* a model on large amounts of unlabelled data, learning to solve different pre-training tasks, and subsequently *fine-tuning* it on a specific downstream task through supervised learning [25]. This approach creates versatile base models with a general understanding of language, which are then tailored to various NLP tasks [14].

While pre-training LLMs requires substantial computational resources, fine-tuning is relatively resource-efficient [14]. Among the pre-trained models, BERT [25] and its optimised version RoBERTa [68] stand out as particularly successful NLP models, given their strong performance across a variety of tasks [14]. Unlike models like OpenAI's GPT [80], which are unidirectional, BERT and RoBERTa are bidirectional and can capture context in both directions of the input [25].

BERT

Bidirectional Encoder Representations from Transformers (BERT), introduced by Devlin *et al.* [25] in 2018, has revolutionised NLP, often delivering state-of-the-art performances in a variety of tasks [14, 25, 68]. BERT is a pre-trained large language model (LLM) utilising the transformer [106] encoder architecture(Section A.2) [25]. Unlike traditional transformers, however, BERT can leverage bidirectional context, i.e. context in both directions of each input token [25]. The model was pre-trained on the *BooksCorpus* and *English Wikipedia*, with 800 million and 2.5 billion words respectively and was initially published in two versions: BERT-base (110 million parameters) and BERT-large (340 million parameters) [25]. Following the release of BERT, further variants were developed. The architectures relevant to this work include its knowledge-distilled version *DistilBERT* [87], its version with optimised pre-training RoBERTa [68], and BERT-based models pre-trained on legal documents, such as *LEGAL-BERT* [16].

BERT's strong performance mainly stems from its innovative pre-training tasks:

- *Masked Language Model (MLM):* For the MLM pre-training task, the authors randomly masked 15 % of all input tokens and tasked BERT with predicting these tokens [25]. This objective allows the model to learn the context in both directions of the input [25].
- *Next Sentence Prediction (NSP):* Devlin *et al.* [25] argue that understanding sentence-relationships is a crucial part of many NLP tasks. To learn this relationship, BERT was also trained on predicting the next sentence in the pre-training data for each given input sentence [25].

Adding a single output network on top of the pre-trained BERT model is enough to successfully fine-tune the model for numerous NLP tasks, including text zoning [25, 40].

RoBERTa

Liu *et al.* [68] presented the Robustly Optimized BERT Pretraining Approach (RoBERTa) in 2019 as an enhanced approach to training BERT [25] models [68].

RoBERTa's optimisations rest on four main modifications to BERT:

1. Liu *et al.* [68] used a larger training text corpora, which is about ten times bigger in file size than the original corpora used by Devlin *et al.* [25]. Additionally, RoBERTa was trained with larger batch sizes for an extended training time [68].
2. In contrast to the original BERT pre-training approach, RoBERTa is not trained on the NSP objective [68].
3. RoBERTa uses *dynamic masking* [68]. Here, instead of the static masking approach used by Devlin *et al.* [25], the mask is created dynamically when a new sequence is introduced to the model [68].
4. Instead of the 30,000 token vocabulary on character-level used by Devlin *et al.* [25], RoBERTa utilises a Byte-Pair Encoding (BPE) tokenizer with a vocabulary of 50,000 byte-level tokens [68].

RoBERTa typically achieves similar or superior performance to BERT in numerous NLP tasks, making it a suitable starting point for this study [14, 50, 68].

Appendix B

Data analysis details

The following figures provide additional information about the data used in this study as well as the data analysis conducted in Chapter 3.

Section B.1 illustrates the origin of documents, while Section B.2 provides insights into the top publishers of the CYBER I and CYBER II datasets. Table B.1 summarises this information in a comprehensive table outlining the number of unique documents and publishers per theme for each country. Section B.3 explains further details about this study's dataset.

B.1 Origin of documents in the CYBER themes

Figure B.1 and Figure B.2 illustrate the origin of this work's documents in the CYBER I and CYBER II themes. The circles' diameters indicate the number of documents originating from each country. Data from the European Union is aggregated with that of Belgium.

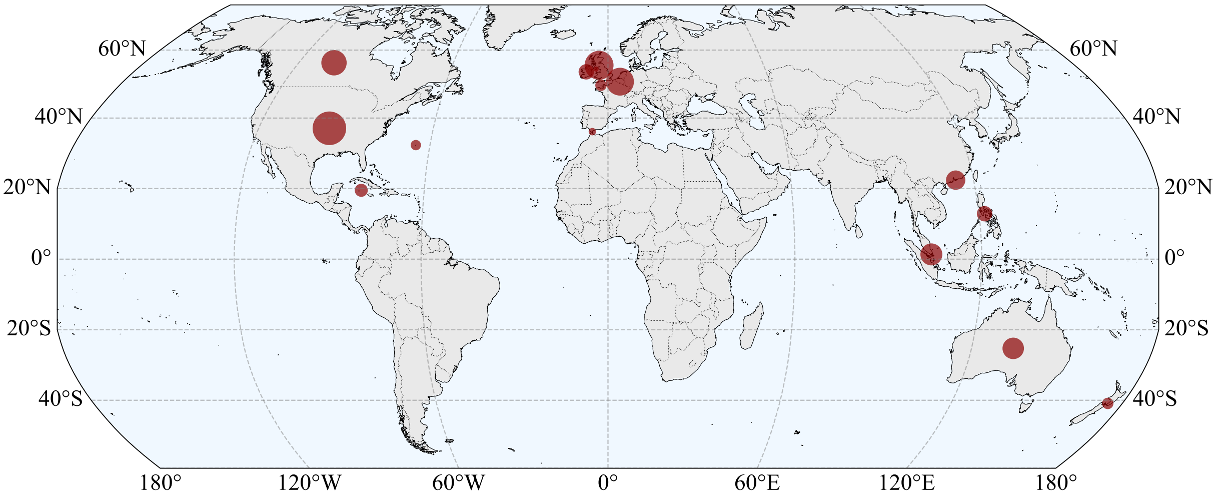


Figure B.1: Origin of the CYBER I documents.

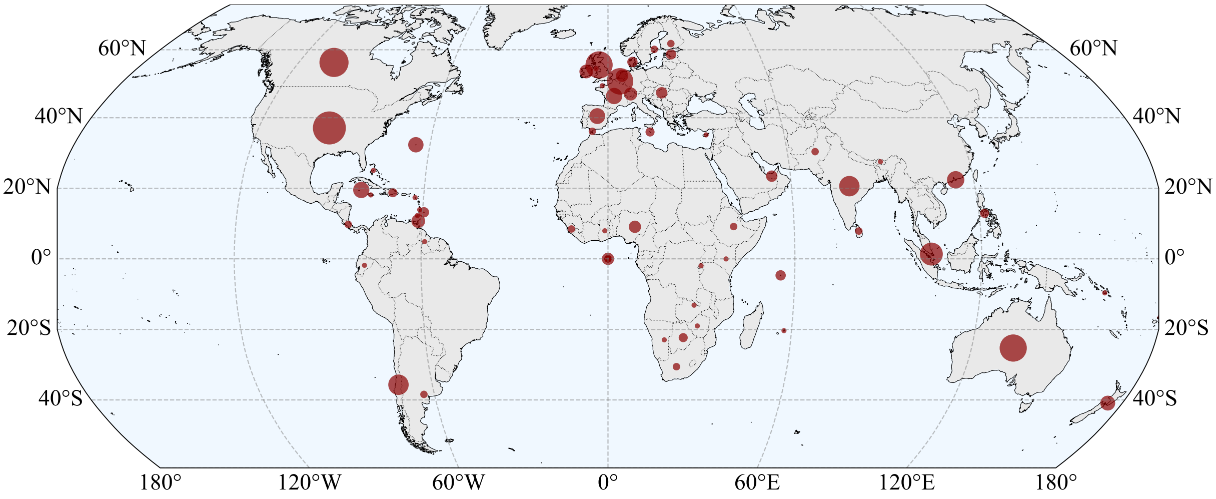


Figure B.2: Origin of the CYBER II documents.

While the CYBER I dataset is only marginally smaller than the CYBER II dataset in terms of the total number of documents and pages, the CYBER II dataset is significantly more diverse, containing documents from 66 different jurisdictions. In contrast, the CYBER I dataset only contains documents from 15 jurisdictions, comprising about half the number of distinct publishers as the CYBER II dataset (see Table 3.1).

B.2 Largest publishers in the CYBER themes

Figure B.3 and Figure B.4 show the largest publishers of cybersecurity regulations in the CYBER I and CYBER II theme, respectively. The jurisdictions are indicated as ALPHA-2 codes (ISO 3166)[1].

Given the strategic importance of cybersecurity in these jurisdictions [22], it is unsurprising that the majority of regulations in the CYBER I and CYBER II datasets originate from the United States, the European Union, the United Kingdom, and Canada.

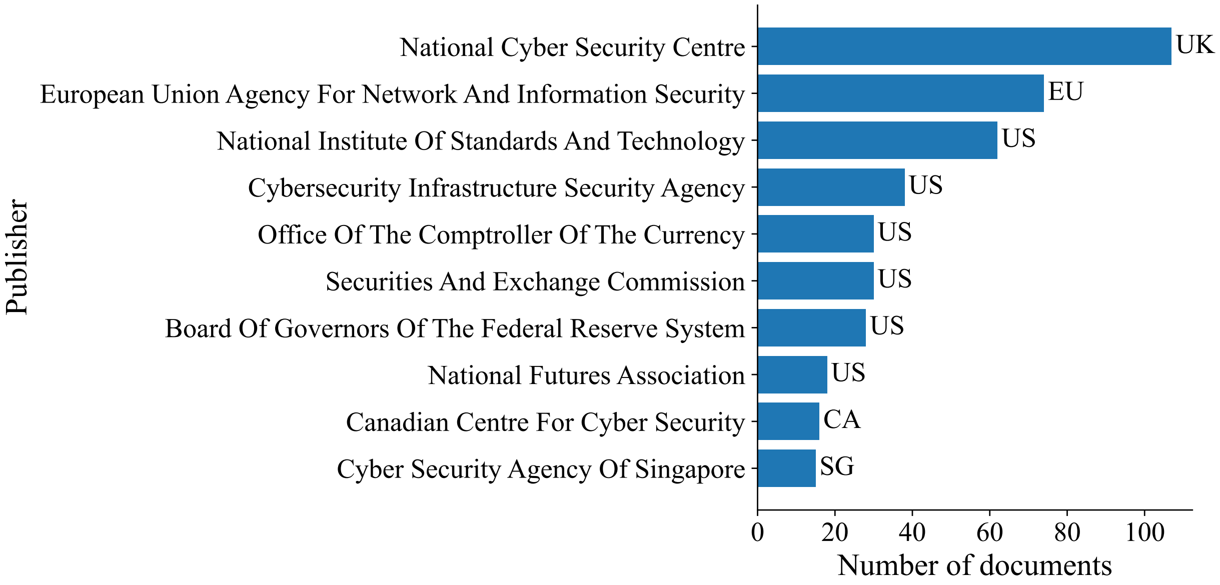


Figure B.3: Top ten publishers by number of documents in the CYBER I theme.

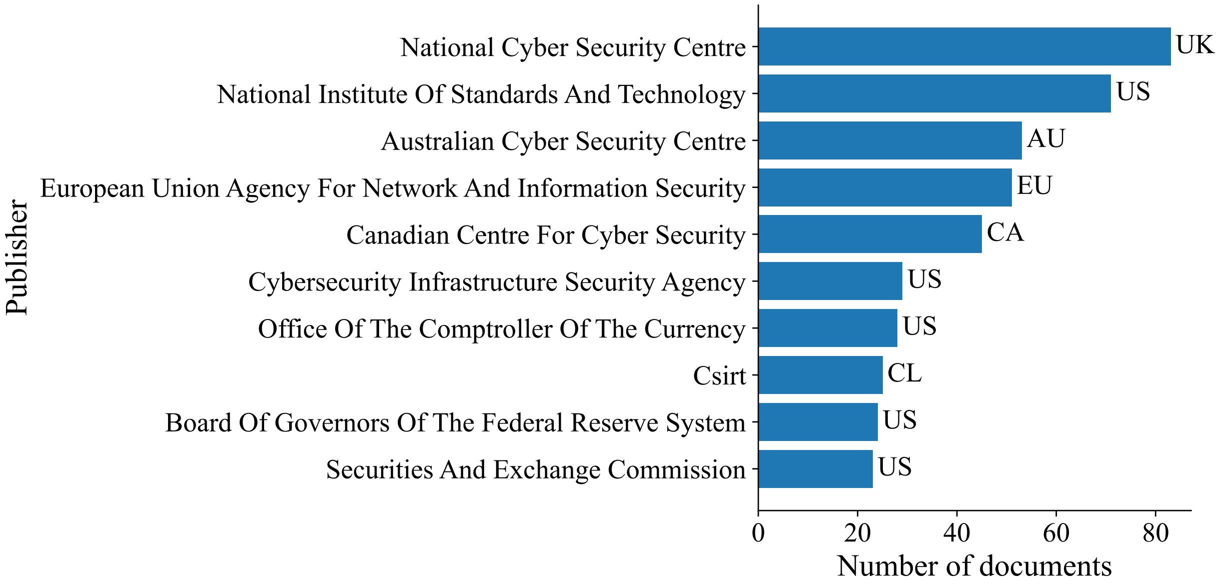


Figure B.4: Top ten publishers by number of documents in the CYBER II theme.

[1] https://www.iso.org/obp/ui/#search

ID	Country/Jurisdiction	AML	CYBER I	CYBER II	Publishers
0	Anguilla	14	0	0	2
1	Antigua and Barbuda	1	0	0	1
2	Argentina	0	0	2	2
3	Australia	45	34	94	10
4	Bahamas	9	0	1	2
5	Barbados	6	0	4	7
6	Belgium	12	0	5	4
7	Belize	9	0	0	3
8	Benin	2	0	0	1
9	Bermuda	19	4	11	4
10	Bhutan	1	0	1	2
11	Bolivia, Plurinational State of	0	0	2	1
12	Botswana	8	0	3	4
13	Brunei Darussalam	1	0	6	2
14	Canada	43	68	132	29
15	Cayman Islands	101	7	13	4
16	Chile	1	0	28	3
17	Cook Islands	5	0	0	2
18	Costa Rica	0	0	2	1
19	Cyprus	3	0	1	2
20	Côte d'Ivoire	0	0	1	1
21	Denmark	0	0	4	1
22	Dominican Republic	0	0	3	3
23	Ecuador	0	0	1	1
24	Estonia	6	0	4	3
25	Eswatini	1	0	0	1
26	Ethiopia	1	0	2	2
27	European Union	83	105	78	11
28	Fiji	1	0	1	2
29	Finland	2	0	2	3
30	France	2	0	13	3
31	Ghana	2	0	1	2
32	Gibraltar	14	2	2	4
33	Grenada	2	0	1	2
34	Guernsey	24	5	0	2
35	Guyana	2	0	1	3
36	Hong Kong	47	23	16	11
37	Hungary	0	0	5	1
38	India	12	0	28	7
39	Ireland	7	11	8	5
40	Isle of Man	13	1	1	3
41	Jamaica	3	0	1	2
42	Jersey	46	0	1	2
43	Kazakhstan	3	0	0	2
44	Kenya	3	0	1	3
45	Latvia	2	0	0	1
46	Liberia	1	0	0	1
47	Malawi	3	0	0	2
48	Malaysia	8	0	0	1
49	Maldives	1	0	0	1
50	Malta	13	0	3	3
51	Mauritius	8	0	1	3
52	Monaco	2	0	0	1
53	Namibia	24	0	1	4
54	Nepal	1	0	0	1
55	Netherlands	4	0	6	5
56	New Zealand	16	5	10	6
57	Nigeria	5	0	6	3
58	Pakistan	6	0	2	3
59	Papua New Guinea	2	0	0	1
60	Philippines	21	11	3	5
61	Rwanda	2	0	1	2
62	Saint Kitts and Nevis	1	0	1	2
63	Saint Lucia	2	0	1	2
64	Samoa	2	0	0	2
65	Seychelles	3	0	4	4
66	Sierra Leone	1	0	2	2
67	Singapore	35	37	44	5
68	Slovakia	4	0	0	2
69	Solomon Islands	0	0	1	1
70	South Africa	18	0	2	4
71	Spain	1	0	12	3
72	Sri Lanka	5	0	2	3
73	Sweden	2	0	2	2
74	Switzerland	0	0	7	2
75	Tanzania, United Republic of	3	0	1	2
76	Trinidad and Tobago	10	0	8	5
77	Uganda	1	0	0	1
78	United Arab Emirates	54	0	5	9
79	United Kingdom	43	127	94	10
80	United States	263	289	262	26
81	Virgin Islands, British	18	0	2	3
82	Zambia	5	0	1	3
83	Zimbabwe	5	0	1	2

Table B.1: Dataset structure: number of documents containing FRI by theme and number of publishers per country/jurisdiction.

B.3 Additional dataset details

The following figures provide additional insights into the structure of this study's data.

Figure B.5 illustrates the different hierarchical levels in the Regulatory Genome Project (RGP) ontology, explaining the structure of the detailed labels that are part of the snippet identifier system's auxiliary objective of matching snippets to the RGP ontology. Figure B.6 reveals how these detailed labels are distributed across levels 0 to 3 in the AML dataset.

Figure B.7 shows a screenshot of the comparison tool developed as part of this work to gain detailed intuitive insights into individual pages of this study's dataset. Figure B.8 and B.9 present various different data distributions, outlining the quantitative distribution of different textual elements in the AML dataset as well as the distribution of the length of the texts contained in these elements.

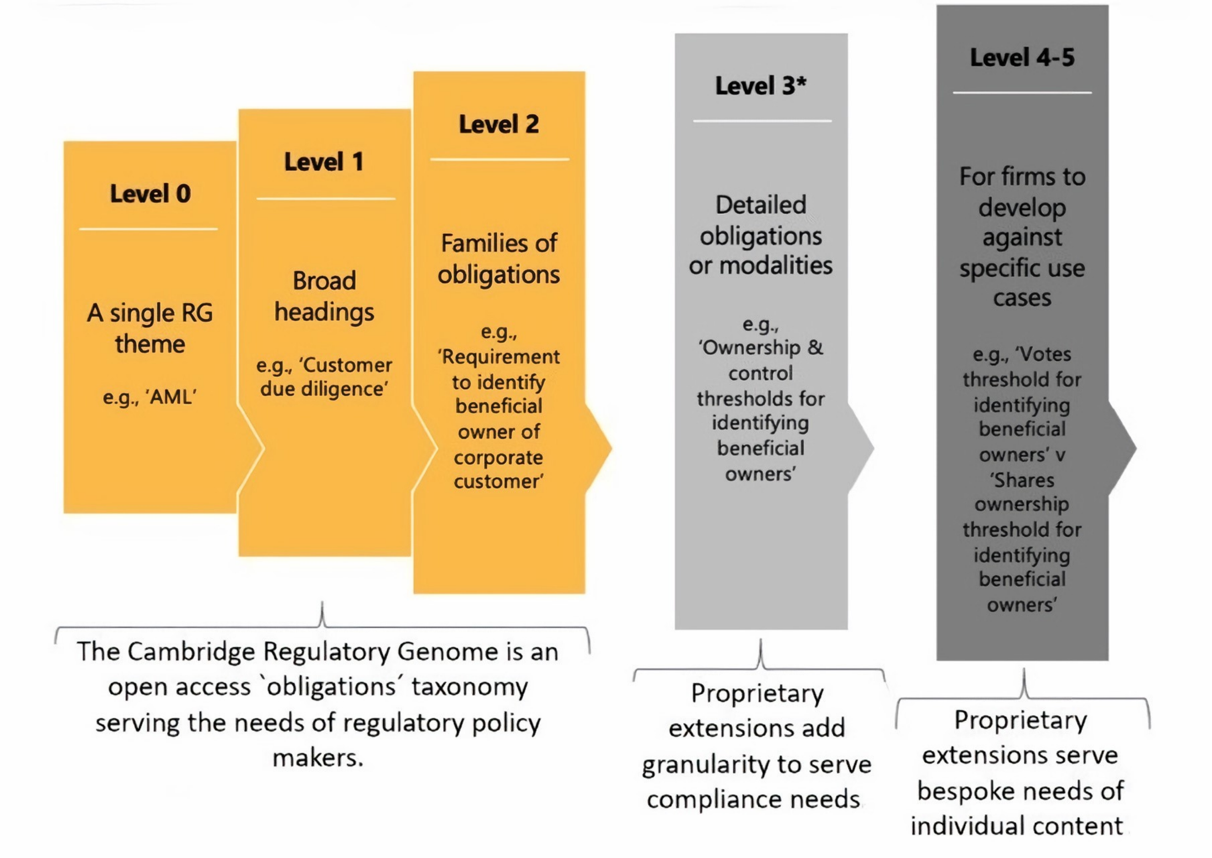

Level 0
A single RG theme
e.g., 'AML'
Level 1
Broad headings
e.g., 'Customer due diligence'
Level 2
Families of obligations
e.g., 'Requirement to identify beneficial owner of corporate customer'
Level 3*
Detailed obligations or modalities
e.g., 'Ownership & control thresholds for identifying beneficial owners'
Level 4-5
For firms to develop against specific use cases
e.g., 'Votes threshold for identifying beneficial owners' v 'Shares ownership threshold for identifying beneficial owners'
The Cambridge Regulatory Genome is an open access `obligations´ taxonomy serving the needs of regulatory policy makers.
Proprietary extensions add granularity to serve compliance needs
Proprietary extensions serve bespoke needs of individual content

Figure B.5: Different hierarchical levels according to the Regulatory Genome Project (RGP). Graphic taken from Cambridge Regulatory Genome Project [11].

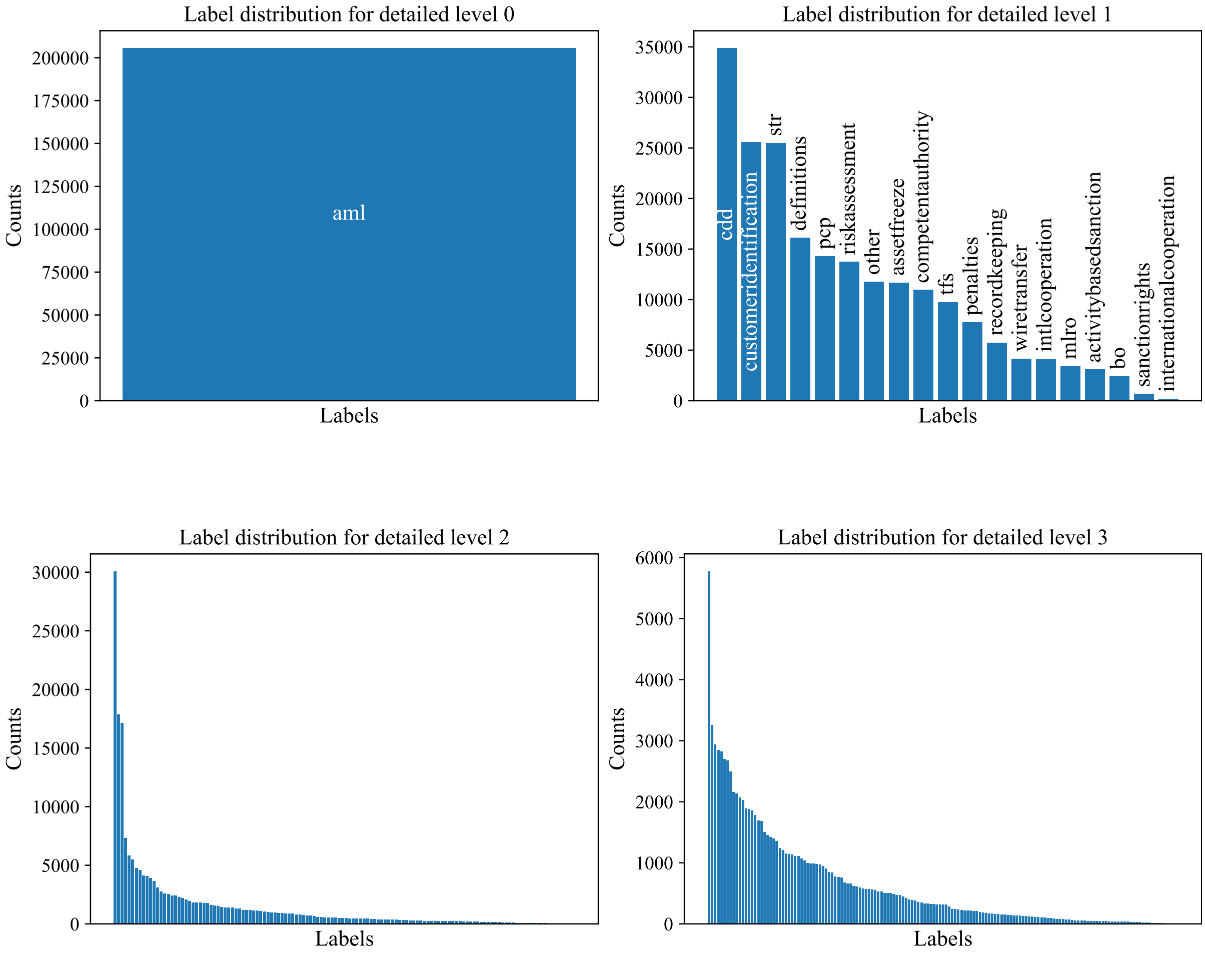
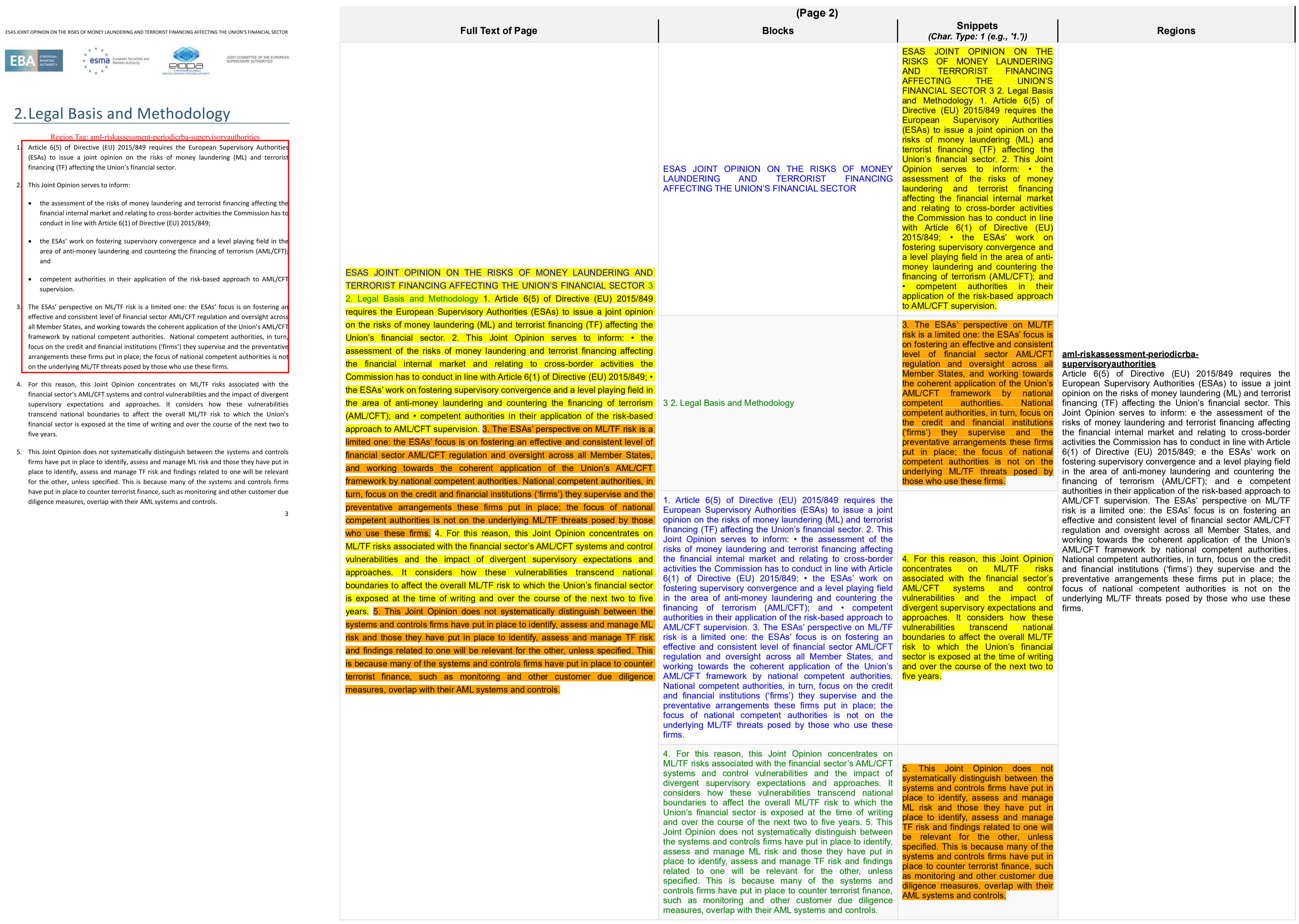


Figure B.6: Distribution of *detailed labels* per level in the AML dataset as defined by the Regulatory Genome Project (RGP) ontology. The number of labels increases for more granular levels, with 1, 19, 130, and 154 distinct labels for levels 0 to 3, respectively. The label distributions reveal a strong class imbalance, particularly in levels 2 and 3, where one label, followed by a small group of labels, accounts for a large proportion of the samples. The samples in this figure correspond to the samples in the sentence-level model dataset (Section 5.2.2) but are indicative of the general label distribution.



ESAS JOINT OPINION ON THE RISKS OF MONEY LAUNDERING AND TERRORIST FINANCING AFFECTING THE UNION'S FINANCIAL SECTOR

EBA EUROPEAN BANKING AUTHORITY
esma European Securities and Markets Authority
eiopa EUROPEAN INSURANCE AND OCCUPATIONAL PENSIONS AUTHORITY
JOINT COMMITTEE OF THE EUROPEAN SUPERVISORY AUTHORITIES

2. Legal Basis and Methodology

Region Tag: aml-riskassessment-periodicrba-supervisoryauthorities

1. Article 6(5) of Directive (EU) 2015/849 requires the European Supervisory Authorities (ESAs) to issue a joint opinion on the risks of money laundering (ML) and terrorist financing (TF) affecting the Union's financial sector.

2. This Joint Opinion serves to inform:

   - the assessment of the risks of money laundering and terrorist financing affecting the financial internal market and relating to cross-border activities the Commission has to conduct in line with Article 6(1) of Directive (EU) 2015/849;
   - the ESAs' work on fostering supervisory convergence and a level playing field in the area of anti-money laundering and countering the financing of terrorism (AML/CFT); and
   - competent authorities in their application of the risk-based approach to AML/CFT supervision.

3. The ESAs' perspective on ML/TF risk is a limited one: the ESAs' focus is on fostering an effective and consistent level of financial sector AML/CFT regulation and oversight across all Member States, and working towards the coherent application of the Union's AML/CFT framework by national competent authorities. National competent authorities, in turn, focus on the credit and financial institutions ('firms') they supervise and the preventative arrangements these firms put in place; the focus of national competent authorities is not on the underlying ML/TF threats posed by those who use these firms.

4. For this reason, this Joint Opinion concentrates on ML/TF risks associated with the financial sector's AML/CFT systems and control vulnerabilities and the impact of divergent supervisory expectations and approaches. It considers how these vulnerabilities transcend national boundaries to affect the overall ML/TF risk to which the Union's financial sector is exposed at the time of writing and over the course of the next two to five years.

5. This Joint Opinion does not systematically distinguish between the systems and controls firms have put in place to identify, assess and manage ML risk and those they have put in place to identify, assess and manage TF risk and findings related to one will be relevant for the other, unless specified. This is because many of the systems and controls firms have put in place to counter terrorist finance, such as monitoring and other customer due diligence measures, overlap with their AML systems and controls.

3

(Page 2)

Full Text of Page	Blocks	Snippets *(Char. Type: 1 (e.g., '1.'))*	Regions
ESAS JOINT OPINION ON THE RISKS OF MONEY LAUNDERING AND TERRORIST FINANCING AFFECTING THE UNION'S FINANCIAL SECTOR 3 2. Legal Basis and Methodology 1. Article 6(5) of Directive (EU) 2015/849 requires the European Supervisory Authorities (ESAs) to issue a joint opinion on the risks of money laundering (ML) and terrorist financing (TF) affecting the Union's financial sector. 2. This Joint Opinion serves to inform: • the assessment of the risks of money laundering and terrorist financing affecting the financial internal market and relating to cross-border activities the Commission has to conduct in line with Article 6(1) of Directive (EU) 2015/849; • the ESAs' work on fostering supervisory convergence and a level playing field in the area of anti-money laundering and countering the financing of terrorism (AML/CFT); and • competent authorities in their application of the risk-based approach to AML/CFT supervision. 3. The ESAs' perspective on ML/TF risk is a limited one: the ESAs' focus is on fostering an effective and consistent level of financial sector AML/CFT regulation and oversight across all Member States, and working towards the coherent application of the Union's AML/CFT framework by national competent authorities. National competent authorities, in turn, focus on the credit and financial institutions ('firms') they supervise and the preventative arrangements these firms put in place; the focus of national competent authorities is not on the underlying ML/TF threats posed by those who use these firms. 4. For this reason, this Joint Opinion concentrates on ML/TF risks associated with the financial sector's AML/CFT systems and control vulnerabilities and the impact of divergent supervisory expectations and approaches. It considers how these vulnerabilities transcend national boundaries to affect the overall ML/TF risk to which the Union's financial sector is exposed at the time of writing and over the course of the next two to five years. 5. This Joint Opinion does not systematically distinguish between the systems and controls firms have put in place to identify, assess and manage ML risk and those they have put in place to identify, assess and manage TF risk and findings related to one will be relevant for the other, unless specified. This is because many of the systems and controls firms have put in place to counter terrorist finance, such as monitoring and other customer due diligence measures, overlap with their AML systems and controls.	ESAS JOINT OPINION ON THE RISKS OF MONEY LAUNDERING AND TERRORIST FINANCING AFFECTING THE UNION'S FINANCIAL SECTOR	ESAS JOINT OPINION ON THE RISKS OF MONEY LAUNDERING AND TERRORIST FINANCING AFFECTING THE UNION'S FINANCIAL SECTOR 3 2. Legal Basis and Methodology 1. Article 6(5) of Directive (EU) 2015/849 requires the European Supervisory Authorities (ESAs) to issue a joint opinion on the risks of money laundering (ML) and terrorist financing (TF) affecting the Union's financial sector. 2. This Joint Opinion serves to inform: • the assessment of the risks of money laundering and terrorist financing affecting the financial internal market and relating to cross-border activities the Commission has to conduct in line with Article 6(1) of Directive (EU) 2015/849; • the ESAs' work on fostering supervisory convergence and a level playing field in the area of anti-money laundering and countering the financing of terrorism (AML/CFT); and • competent authorities in their application of the risk-based approach to AML/CFT supervision.	**aml-riskassessment-periodicrba-supervisoryauthorities** Article 6(5) of Directive (EU) 2015/849 requires the European Supervisory Authorities (ESAs) to issue a joint opinion on the risks of money laundering (ML) and terrorist financing (TF) affecting the Union's financial sector. This Joint Opinion serves to inform: e the assessment of the risks of money laundering and terrorist financing affecting the financial internal market and relating to cross-border activities the Commission has to conduct in line with Article 6(1) of Directive (EU) 2015/849; e the ESAs' work on fostering supervisory convergence and a level playing field in the area of anti-money laundering and countering the financing of terrorism (AML/CFT); and e competent authorities in their application of the risk-based approach to AML/CFT supervision. The ESAs' perspective on ML/TF risk is a limited one: the ESAs' focus is on fostering an effective and consistent level of financial sector AML/CFT regulation and oversight across all Member States, and working towards the coherent application of the Union's AML/CFT framework by national competent authorities. National competent authorities, in turn, focus on the credit and financial institutions ('firms') they supervise and the preventative arrangements these firms put in place; the focus of national competent authorities is not on the underlying ML/TF threats posed by those who use these firms.
	3 2. Legal Basis and Methodology	3. The ESAs' perspective on ML/TF risk is a limited one: the ESAs' focus is on fostering an effective and consistent level of financial sector AML/CFT regulation and oversight across all Member States, and working towards the coherent application of the Union's AML/CFT framework by national competent authorities. National competent authorities, in turn, focus on the credit and financial institutions ('firms') they supervise and the preventative arrangements these firms put in place; the focus of national competent authorities is not on the underlying ML/TF threats posed by those who use these firms.	
	1. Article 6(5) of Directive (EU) 2015/849 requires the European Supervisory Authorities (ESAs) to issue a joint opinion on the risks of money laundering (ML) and terrorist financing (TF) affecting the Union's financial sector. 2. This Joint Opinion serves to inform: • the assessment of the risks of money laundering and terrorist financing affecting the financial internal market and relating to cross-border activities the Commission has to conduct in line with Article 6(1) of Directive (EU) 2015/849; • the ESAs' work on fostering supervisory convergence and a level playing field in the area of anti-money laundering and countering the financing of terrorism (AML/CFT); and • competent authorities in their application of the risk-based approach to AML/CFT supervision. 3. The ESAs' perspective on ML/TF risk is a limited one: the ESAs' focus is on fostering an effective and consistent level of financial sector AML/CFT regulation and oversight across all Member States, and working towards the coherent application of the Union's AML/CFT framework by national competent authorities. National competent authorities, in turn, focus on the credit and financial institutions ('firms') they supervise and the preventative arrangements these firms put in place; the focus of national competent authorities is not on the underlying ML/TF threats posed by those who use these firms.	4. For this reason, this Joint Opinion concentrates on ML/TF risks associated with the financial sector's AML/CFT systems and control vulnerabilities and the impact of divergent supervisory expectations and approaches. It considers how these vulnerabilities transcend national boundaries to affect the overall ML/TF risk to which the Union's financial sector is exposed at the time of writing and over the course of the next two to five years.	
	4. For this reason, this Joint Opinion concentrates on ML/TF risks associated with the financial sector's AML/CFT systems and control vulnerabilities and the impact of divergent supervisory expectations and approaches. It considers how these vulnerabilities transcend national boundaries to affect the overall ML/TF risk to which the Union's financial sector is exposed at the time of writing and over the course of the next two to five years. 5. This Joint Opinion does not systematically distinguish between the systems and controls firms have put in place to identify, assess and manage ML risk and those they have put in place to identify, assess and manage TF risk and findings related to one will be relevant for the other, unless specified. This is because many of the systems and controls firms have put in place to counter terrorist finance, such as monitoring and other customer due diligence measures, overlap with their AML systems and controls.	5. This Joint Opinion does not systematically distinguish between the systems and controls firms have put in place to identify, assess and manage ML risk and those they have put in place to identify, assess and manage TF risk and findings related to one will be relevant for the other, unless specified. This is because many of the systems and controls firms have put in place to counter terrorist finance, such as monitoring and other customer due diligence measures, overlap with their AML systems and controls.	

Figure B.7: Screenshot of an exemplary comparison of the textual contents of a page from this study's AML dataset using the interactive comparison tool. The page is dynamically pulled from the original PDF document via an API provided by RegGenome, while blocks, snippets, and regions stem from the dataset files. Blocks and snippets are highlighted in the full text of the page, which itself is a simple concatenation of the blocks.

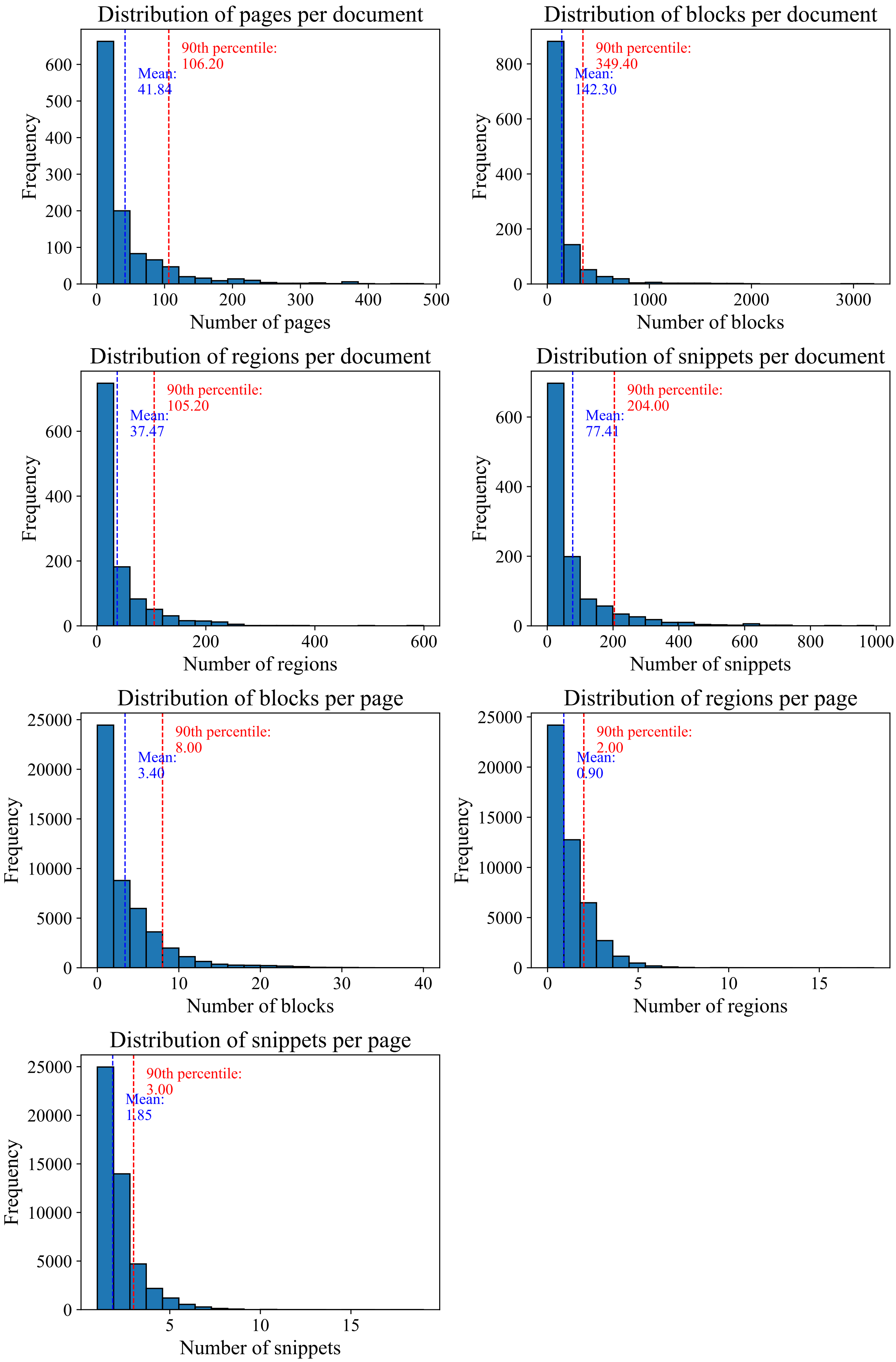


Figure B.8: Distributions for number of different textual elements in the AML dataset.

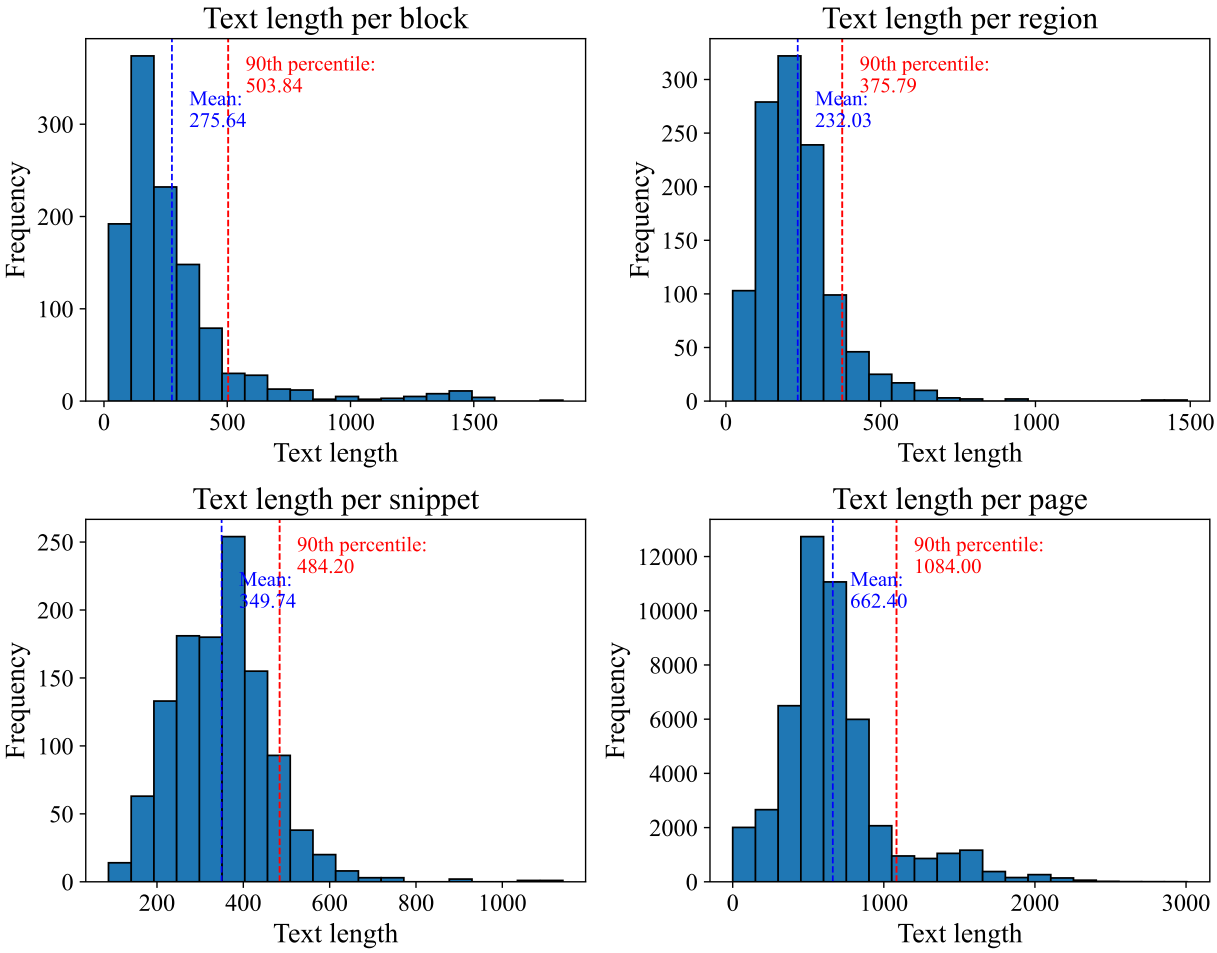


Figure B.9: Distributions of average text lengths in the AML dataset.

Appendix C

Details on methodologies

This chapter provides details on some of the algorithms and methodologies employed throughout this study. Specifically, further details on RegGenome's snippeting algorithm (Section C.1), the approach to determining the optimal threshold to label GraphSeg segments and blocks (Section C.2) as well as the sentence splitting strategies of the sentence-level model (Section C.3) are given.

C.1 RegGenome's snippeting algorithm

Algorithm 1 Pseudocode for RegGenome's snippeting algorithm

```
Input: Page of text
Output: List of snippets
function SPLIT_PAGE_INTO_SNIPPETS(page)
    regexes ← Compile regular expressions (regex) to identify table of contents (ToC)
    character_types ← Load regexes for headings & sections (e.g., ”1)”, ”1.1”, ...)
    # Step I: Identify character type of page
    if ToC is present then
        toc_type ← Determine ToC type by regex matching
    else
        longest_span ← Find longest span for each character type
        selected_type ← Select type with the longest span
    end if
    # Step II: Create snippets
    if character type of text identified then
        snippets ← Form text sections according to identified pattern
    else
        snippets ← Split by size and simple split points
    end if
    # Step III: Create final output
    snippets ← Add snippet index, start and end position for each snippet
    return snippets
end function
```

C.2 Threshold determination for segment labelling in GraphSeg and Blocks models

The dataset creation for the *GraphSeg* and *Blocks* models presented in Section 5.1 requires a statistical approach to labelling GraphSeg segments and blocks as representing regions or not. The following describes the labelling approach we followed in more detail.

Given the Jaccard similarity scores (Section 3.4) between each Graphseg segment or block $S(p)$ and region in the dataset $R(p)$ for all pages $P(d)$ of a document d across all documents in the dataset D as well as an optimal splitting threshold T_{split}, we can label all segments whose Jaccard similarity exceeds T_{split} as 1 and all other segments as 0.

To determine T_{split}, we used the elbow method [104], as implemented in the *kneed* package[1]. Despite criticism of its potential subjectivity [92], we contend its effectiveness in providing an adequate splitting point in this context.

We calculated T_{split} by first collecting the maximum Jaccard similarity scores for each region of each page of each document into an ordered set $\{s_i\}_{i=1}^{N}$:

$$\{s_i\}_{i=1}^{N} = \text{sort}\left(\bigcup_{d\in\mathcal{D}}\bigcup_{p\in\mathcal{P}(d)}\left\{\max_{s\in S(p)} \text{Jaccard}(s,r) \mid r\in R(p)\right\}\right) \tag{C.1}$$

Furthermore, we calculated the cumulative distribution C_j based on $\{s_i\}_{i=1}^{N}$:

$$C_j = \frac{\sum_{i=1}^{j} s_i}{\sum_{i=1}^{N} s_i}, \quad \forall j \in \{1,2,\ldots,N\} \tag{C.2}$$

Using the elbow method, $T_{split} = s_K$ was determined through the calculation of the "Knee Point" K [89]:

$$K = \arg\max_{j\in\{1,2,\ldots,N\}} |C_{j+1} - C_j| \tag{C.3}$$

Based on this method, T_{split} was determined as 0.82 for blocks and 0.58 for GraphSeg segments. These threshold values were consequently used to label all segments for the TS-based snippet identifier models.

[1] `https://pypi.org/project/kneed/`

C.3 Detailed description of sentence-splitting components

The performance of the sentence-level model (Section 5.2) is strongly dependent on the sentence-splitting component used to split the text of a page into sentences. To ensure robust performance, this study employed the following three sentence boundary detection (SBD) methods:

1. **Customly trained Punkt**: Punkt is an unsupervised SBD approach developed by Kiss and Strunk [56], which is widely adopted for the task of sentence splitting. Research on the use of such off-the-shelf models in the legal domain, however, has shown that approaches like Punkt often perform poorly in this field [86, 90]. According to Sanchez [86], for instance, Punkt needs to be trained and updated before it can be used for legal SBD, and Savelka *et al.* [90] similarly highlight that training Punkt is not only relatively cheap due to its unsupervised nature but also encouraged to improve performance in the legal domain. Therefore, as part of this work, NLTK's Punkt implementation[2] was used, and the model was trained on the AML dataset. Subsequently, it could be employed as the SBD component of the sentence-level model.

2. **Extended SpaCy**: As an additional sentence splitting method, the English language model *en_core_web_sm* of the NLP library SpaCy[3] was implemented into the model pipeline. Beyond the default implementation of SpaCy's pipeline, a custom component was introduced to refine SBD based on observations from the behaviour of the default system on the dataset used in this study. Specifically, an additional rule was introduced to enforce a new sentence boundary when encountering two consecutive newline characters ("\n") with optional whitespace characters in between.

3. **MultiLegalSBD**: The final SBD method employed in this study is *MultiLegalSBD*, a state-of-the-art transformer-based model specifically developed for SBD within the legal domain. The model was introduced in 2023 by Brugger *et al.* [10] and shows a robust performance across a diverse range of legal documents. For this work, we utilised the multilingual version of the model, which is accessible through the Hugging Face Transformers library token classification pipeline[4].

[2] `https://www.nltk.org/api/nltk.tokenize.punkt.html`

[3] `https://spacy.io/models/en`

[4] `https://huggingface.co/rcds/distilbert-SBD-fr-es-it-en-de-judgements-laws`

Appendix D

Hyperparameter optimisation details

This chapter details the hyperparameter optimisation (HPO) studies conducted for the sentence-level (Section 5.2.3) and token-level (Section 5.3.3) models. All HPO studies used the *Optuna* framework[1] with default configurations. The objective for each trial was to minimise the main objective's loss. The following summarises the search spaces for both model types. Tables D.1-D.3 list the exact configurations explored for the sentence-level model (50 trials each), while Table D.4 shows the token-level model study (18 trials).

For the sentence-level model:

- Detailed labels levels (*levels*): [], [1], [2], [1,2] (level 0 is trivial, level 3 too complex)
- Weights for detailed labels loss (*weight_level_1* & *weight_level_2*): floating-point number between 0.1 and 1
- Number of attention heads (*nhead*): 2, 4, 8, 16 (transformer-based model only)
- Hidden dimensions (*hidden_dim*): 64, 128, 256, or 512 (for Bi-LSTM-based models), 512, 1024, 2048, 4096 (for transformer-based model)
- Number of layers (*nlayers*): integer between 1 and 8 (for Bi-LSTM-based models), integer between 1 and 24 for (transformer-based model)
- Dropout rate (*dropout*): floating-point number between 0.1 and 0.5
- α loss parameter (*alpha*): floating-point number between 0 and 1

For the token-level model:

- Detailed labels levels (*levels*): [], [1], [2], [1,2] (level 0 is trivial, level 3 too complex)
- Weights for detailed labels loss (*weight_level_1* & *weight_level_2*): floating-point number between 0.1 and 1
- α loss parameter (*alpha*): floating-point number between 0 and 1

[1] `https://optuna.org/`

ID	Levels	Weight _level_0	Weight _level_1	Hidden _dim	Nlayers	Dropout	Alpha	State	Value
0	[1, 2]	0.820	0.132	512	6	0.461	0.266	COMPL.	0.653
1	None	NaN	NaN	128	6	0.226	NaN	COMPL.	0.467
2	[2]	0.382	NaN	512	6	0.235	0.759	COMPL.	0.650
3	[1]	0.734	NaN	256	6	0.219	0.423	COMPL.	0.657
4	[1]	0.620	NaN	512	1	0.329	0.289	COMPL.	0.606
5	[1, 2]	0.519	0.902	256	3	0.476	0.996	COMPL.	0.460
6	None	NaN	NaN	64	5	0.380	NaN	COMPL.	0.480
7	None	NaN	NaN	256	7	0.300	NaN	COMPL.	0.456
8	[1]	0.609	NaN	512	5	0.468	0.706	COMPL.	0.646
9	[1]	0.124	NaN	512	3	0.164	0.382	COMPL.	0.647
10	None	NaN	NaN	256	8	0.123	NaN	COMPL.	0.457
11	None	NaN	NaN	256	8	0.103	NaN	COMPL.	0.456
12	None	NaN	NaN	256	8	0.106	NaN	COMPL.	0.456
13	None	NaN	NaN	256	8	0.102	NaN	COMPL.	0.456
14	[2]	0.970	NaN	128	8	0.167	0.024	COMPL.	0.522
15	None	NaN	NaN	64	7	0.164	NaN	COMPL.	0.479
16	None	NaN	NaN	256	3	0.282	NaN	COMPL.	0.457
17	None	NaN	NaN	256	7	0.105	NaN	COMPL.	0.456
18	[2]	0.106	NaN	256	4	0.183	0.022	COMPL.	0.614
19	[1, 2]	0.322	0.418	64	8	0.346	0.647	COMPL.	0.630
20	None	NaN	NaN	128	1	0.420	NaN	COMPL.	0.499
21	None	NaN	NaN	256	8	0.101	NaN	COMPL.	0.456
22	None	NaN	NaN	256	7	0.139	NaN	COMPL.	0.456
23	None	NaN	NaN	256	7	0.139	NaN	COMPL.	0.456
24	None	NaN	NaN	256	7	0.136	NaN	COMPL.	0.456
25	None	NaN	NaN	256	7	0.142	NaN	COMPL.	0.456
26	None	NaN	NaN	256	5	0.201	NaN	COMPL.	0.455
27	[2]	0.970	NaN	256	4	0.198	0.985	COMPL.	0.457
28	[1, 2]	0.277	0.983	128	5	0.267	0.566	COMPL.	0.631
29	[1, 2]	0.445	0.609	64	6	0.258	0.127	COMPL.	0.539
30	None	NaN	NaN	256	2	0.199	NaN	COMPL.	0.462
31	None	NaN	NaN	256	7	0.146	NaN	COMPL.	0.456
32	None	NaN	NaN	256	6	0.139	NaN	COMPL.	0.456
33	None	NaN	NaN	256	6	0.233	NaN	COMPL.	0.456
34	None	NaN	NaN	256	6	0.230	NaN	COMPL.	0.456
35	None	NaN	NaN	128	6	0.248	NaN	COMPL.	0.467
36	[1]	0.819	NaN	512	5	0.211	0.840	COMPL.	0.646
37	[2]	0.230	NaN	256	4	0.183	0.836	COMPL.	0.653
38	None	NaN	NaN	256	6	0.235	NaN	COMPL.	0.457
39	None	NaN	NaN	256	6	0.330	NaN	COMPL.	0.457
40	[1]	0.679	NaN	512	5	0.213	0.174	COMPL.	0.647
41	None	NaN	NaN	256	7	0.131	NaN	COMPL.	0.457
42	None	NaN	NaN	256	5	0.175	NaN	COMPL.	0.455
43	None	NaN	NaN	256	5	0.175	NaN	COMPL.	0.455
44	None	NaN	NaN	256	5	0.161	NaN	COMPL.	0.455
45	[1, 2]	0.495	0.102	256	5	0.181	0.523	COMPL.	0.655
46	None	NaN	NaN	64	4	0.160	NaN	COMPL.	0.476
47	None	NaN	NaN	256	5	0.196	NaN	COMPL.	0.455
48	[1]	0.848	NaN	512	5	0.197	0.402	COMPL.	0.647
49	None	NaN	NaN	256	4	0.177	NaN	COMPL.	0.457

Table D.1: Overview of Optuna trials for the HPO of the Bi-LSTM model. The best trial is highlighted in grey.

ID	Levels	Weight _level_0	Weight _level_1	Hidden _dim	Nlayers	Dropout	Alpha	State	Value
0	None	NaN	NaN	256	6	0.142	NaN	COMPL.	0.439
1	[2]	0.610	NaN	256	3	0.336	0.086	COMPL.	0.445
2	[1, 2]	0.536	0.162	64	7	0.116	0.588	COMPL.	0.468
3	[1, 2]	0.969	0.612	256	7	0.147	0.008	COMPL.	0.438
4	[1, 2]	0.135	0.200	512	3	0.313	0.492	COMPL.	0.393
5	None	NaN	NaN	256	3	0.249	NaN	PRUNED	484.512
6	[1, 2]	0.899	0.536	256	7	0.325	0.839	PRUNED	249.600
7	[1]	0.937	NaN	512	5	0.339	0.455	COMPL.	0.443
8	[1, 2]	0.582	0.436	64	1	0.434	0.905	PRUNED	718.294
9	None	NaN	NaN	256	5	0.460	NaN	PRUNED	476.150
10	[2]	0.124	NaN	512	2	0.228	0.384	PRUNED	672.324
11	[1, 2]	0.210	0.948	128	8	0.196	0.053	PRUNED	694.799
12	[1, 2]	0.371	0.733	512	4	0.412	0.239	COMPL.	0.442
13	[1]	0.764	NaN	128	4	0.270	0.649	PRUNED	638.878
14	[1, 2]	0.337	0.183	512	8	0.173	0.311	PRUNED	150.862
15	[1, 2]	0.780	0.385	512	1	0.387	0.723	PRUNED	674.111
16	[1, 2]	0.453	0.682	128	6	0.282	0.198	PRUNED	540.388
17	[1]	0.763	NaN	64	3	0.375	0.537	PRUNED	644.638
18	[2]	0.988	NaN	256	6	0.201	0.979	PRUNED	677.464
19	[1, 2]	0.245	0.323	512	2	0.302	0.013	COMPL.	0.400
20	[1, 2]	0.204	0.294	512	2	0.494	0.171	PRUNED	641.796
21	[1, 2]	0.273	0.283	512	2	0.304	0.020	PRUNED	137.310
22	[1, 2]	0.110	0.632	512	4	0.361	0.003	COMPL.	0.443
23	[1, 2]	0.221	0.490	512	3	0.295	0.120	COMPL.	0.393
24	[1, 2]	0.210	0.432	512	3	0.303	0.137	PRUNED	626.406
25	[1, 2]	0.339	0.286	512	2	0.256	0.321	PRUNED	364.159
26	None	NaN	NaN	512	1	0.226	NaN	PRUNED	712.433
27	[1]	0.427	NaN	512	3	0.304	0.403	COMPL.	0.398
28	[1]	0.472	NaN	512	4	0.356	0.436	PRUNED	635.109
29	[1]	0.427	NaN	512	3	0.390	0.690	PRUNED	320.332
30	[1]	0.136	NaN	128	5	0.319	0.281	PRUNED	530.981
31	[1]	0.285	NaN	512	2	0.279	0.513	PRUNED	350.849
32	[2]	0.265	NaN	512	3	0.303	0.118	PRUNED	333.850
33	None	NaN	NaN	512	2	0.338	NaN	PRUNED	357.184
34	[1, 2]	0.194	0.102	64	4	0.239	0.390	PRUNED	694.589
35	[2]	0.400	NaN	512	3	0.287	0.110	PRUNED	633.667
36	[1, 2]	0.674	0.501	512	3	0.267	0.573	PRUNED	635.162
37	[1]	0.164	NaN	64	2	0.323	0.232	PRUNED	591.747
38	[1, 2]	0.297	0.338	512	1	0.350	0.077	PRUNED	661.073
39	None	NaN	NaN	512	3	0.319	NaN	PRUNED	148.923
40	[1, 2]	0.502	0.190	256	4	0.123	0.767	PRUNED	472.548
41	[1, 2]	0.674	0.798	256	3	0.102	0.047	PRUNED	688.181
42	[1, 2]	0.245	0.611	256	7	0.158	0.164	PRUNED	687.719
43	[1, 2]	0.848	0.446	256	5	0.209	0.621	PRUNED	449.029
44	[1, 2]	0.573	0.528	256	7	0.174	0.077	PRUNED	467.254
45	[1, 2]	0.159	0.234	128	1	0.412	0.003	PRUNED	692.269
46	[1]	0.340	NaN	256	8	0.133	0.375	PRUNED	690.430
47	[1, 2]	0.634	0.601	512	6	0.237	0.467	PRUNED	180.310
48	[2]	0.231	NaN	64	5	0.258	0.232	PRUNED	666.506
49	None	NaN	NaN	512	2	0.444	NaN	PRUNED	637.144

Table D.2: Overview of Optuna trials for the HPO of the Bi-LSTM+CRF model. The best trial is highlighted in grey.

ID	Levels	Weight _level_0	Weight _level_1	Nhead	Hidden _dim	Nlayers	Dropout	Alpha	State	Value
0	[1]	0.188	NaN	8	2048	23	0.181	0.445	COMPL.	0.335
1	[1, 2]	0.266	0.684	8	4096	24	0.340	0.672	COMPL.	0.438
2	[1]	0.548	NaN	4	2048	8	0.307	0.900	COMPL.	0.320
3	[1, 2]	0.328	0.620	2	1024	14	0.244	0.742	COMPL.	0.333
4	None	NaN	NaN	4	4096	19	0.437	NaN	COMPL.	0.333
5	None	NaN	NaN	8	1024	17	0.138	NaN	PRUNED	0.428
6	[2]	0.636	NaN	16	2048	16	0.361	0.351	COMPL.	0.341
7	[2]	0.249	NaN	4	2048	8	0.116	0.599	COMPL.	0.339
8	[1, 2]	0.256	0.371	2	4096	2	0.310	0.403	PRUNED	1.248
9	None	NaN	NaN	8	4096	3	0.455	NaN	PRUNED	0.349
10	[1]	0.938	NaN	4	512	9	0.239	0.994	PRUNED	0.378
11	[1, 2]	0.542	0.999	2	1024	11	0.241	0.882	PRUNED	0.767
12	[1]	0.520	NaN	2	1024	13	0.248	0.022	COMPL.	0.382
13	[1, 2]	0.728	0.128	16	512	7	0.390	0.792	PRUNED	0.785
14	[1]	0.405	NaN	4	1024	13	0.282	0.752	PRUNED	0.301
15	[1, 2]	0.367	0.660	2	2048	5	0.191	0.997	PRUNED	0.369
16	[1]	0.795	NaN	2	1024	11	0.305	0.591	COMPL.	0.347
17	[2]	0.395	NaN	4	2048	16	0.399	0.838	PRUNED	0.350
18	[1, 2]	0.468	0.958	16	512	5	0.499	0.273	COMPL.	0.389
19	[1]	0.122	NaN	4	1024	21	0.199	0.692	PRUNED	0.302
20	[1]	0.642	NaN	2	2048	14	0.283	0.905	PRUNED	0.597
21	None	NaN	NaN	4	4096	20	0.468	NaN	PRUNED	0.448
22	None	NaN	NaN	4	4096	20	0.395	NaN	PRUNED	0.439
23	None	NaN	NaN	4	4096	18	0.437	NaN	PRUNED	0.416
24	None	NaN	NaN	4	1024	10	0.346	NaN	PRUNED	0.357
25	None	NaN	NaN	4	4096	15	0.160	NaN	PRUNED	0.350
26	[1, 2]	0.986	0.465	2	2048	18	0.219	0.565	PRUNED	0.339
27	[2]	0.643	NaN	16	512	6	0.322	0.726	PRUNED	0.967
28	[1, 2]	0.319	0.736	4	1024	10	0.268	0.208	COMPL.	0.361
29	[1]	0.835	NaN	8	2048	22	0.430	0.904	PRUNED	0.396
30	None	NaN	NaN	2	4096	19	0.500	NaN	PRUNED	0.445
31	[1]	0.128	NaN	8	2048	24	0.183	0.469	COMPL.	0.330
32	[1]	0.145	NaN	8	2048	24	0.212	0.506	PRUNED	0.221
33	[1]	0.101	NaN	8	2048	23	0.185	0.646	PRUNED	0.283
34	[1]	0.204	NaN	8	2048	21	0.151	0.493	COMPL.	0.338
35	[1]	0.473	NaN	8	2048	17	0.101	0.812	PRUNED	0.341
36	[2]	0.322	NaN	8	2048	23	0.162	0.199	COMPL.	0.336
37	[1, 2]	0.463	0.239	4	4096	15	0.374	0.920	PRUNED	0.369
38	None	NaN	NaN	16	2048	1	0.325	NaN	PRUNED	0.376
39	[1]	0.195	NaN	8	1024	24	0.225	0.430	COMPL.	0.347
40	[1, 2]	0.335	0.831	2	4096	12	0.262	0.526	COMPL.	0.332
41	[1, 2]	0.316	0.791	2	4096	12	0.257	0.512	COMPL.	0.334
42	[1, 2]	0.187	0.544	2	4096	8	0.128	0.352	PRUNED	2.552
43	[1, 2]	0.607	0.844	2	4096	12	0.283	0.637	PRUNED	0.237
44	[1, 2]	0.582	0.591	2	4096	14	0.236	0.716	PRUNED	0.265
45	[1, 2]	0.277	0.874	2	512	9	0.266	0.368	PRUNED	0.696
46	[2]	0.421	NaN	4	1024	4	0.206	0.542	PRUNED	0.418
47	[1]	0.235	NaN	2	4096	7	0.297	0.274	COMPL.	0.351
48	[1, 2]	0.712	0.437	4	2048	16	0.348	0.457	COMPL.	0.340
49	[1]	0.368	NaN	16	1024	11	0.172	0.850	PRUNED	0.630

Table D.3: Overview of Optuna trials for the HPO of the transformer model. The best trial is highlighted in grey.

ID	Levels	Weight _level_0	Weight _level_1	State	Value
0	[1]	0.646	NaN	COMPL.	0.106
1	[2]	0.973	NaN	COMPL.	0.118
2	None	NaN	NaN	COMPL.	0.091
3	[1]	0.280	NaN	COMPL.	0.117
4	[1]	0.555	NaN	COMPL.	0.106
5	[1]	0.656	NaN	COMPL.	0.095
6	[2]	0.770	NaN	PRUNED	1.070
7	[1]	0.778	NaN	PRUNED	0.998
8	None	NaN	NaN	COMPL.	0.097
9	[1]	0.112	NaN	PRUNED	1.003
10	None	NaN	NaN	COMPL.	0.097
11	[1, 2]	0.336	0.599	COMPL.	0.095
12	[1, 2]	0.344	0.601	COMPL.	0.098
13	[1, 2]	0.390	0.193	PRUNED	0.594
14	None	NaN	NaN	COMPL.	0.097
15	[1, 2]	0.121	0.992	PRUNED	0.615
16	[1, 2]	0.464	0.561	PRUNED	0.628
17	None	NaN	NaN	COMPL.	0.097

Table D.4: Overview of optuna trials for the HPO of the token-level model. The best trial is highlighted in grey.