Politecnico di Torino

ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and control engineering (37^{th}cycle)

Enhancing Legal Document Processing through Natural Language Understanding and Generation techniques

Improving Document Exploration, Accessibility, and Decision Support in the Legal Domain

By

Irene Benedetto

******

Supervisor(s):
Prof. Luca Cagliero, Supervisor
Dr. Vittorio Di Tomaso, Co-Supervisor
Dr. Francesco Tarasconi, Co-Supervisor

Doctoral Examination Committee:
Prof. Gerasimos Spanakis , Referee, Maastricht University
Prof. Alessandro Lenci, Referee, University of Pisa
Prof. Viviana Patti, University of Torino
Prof. Giuseppe Rizzo, Polytechnic University of Turin

Politecnico di Torino
2025

Declaration

I hereby declare that, the contents and organization of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

Irene Benedetto
2025

* This dissertation is presented in partial fulfillment of the requirements for **Ph.D. degree** in the Graduate School of Politecnico di Torino (ScuDo).

Acknowledgements

I would like to acknowledge my family, friends, and colleagues for their constant support and encouragement. In particular, I want to thank the *rebellious group of Lab 5* for their inspiration, perseverance, and the countless laughs.

Abstract

The rapid advancement of Artificial Intelligence (AI), particularly in Natural Language Processing (NLP), offers significant opportunities for innovation in the legal domain. However, the characteristics of legal language and the intricate nature of legal tasks present substantial challenges for AI adoption in real-use cases. Current legal systems often rely on manual annotation and predefined taxonomies that are time-consuming to maintain and struggle to capture the evolving nature of legal concepts. Existing AI approaches face limitations such as dependence on outdated information and data scarcity. In the realm of decision-making, legal professionals hesitate to fully trust AI-generated outcomes due to concerns about accuracy, transparency, and accountability, especially since AI systems often lack true legal reasoning. This dissertation explores how NLP and in particular Language Models can address these challenges and enhance legal workflows across three areas: automatic document exploration, content accessibility, and reasoning applications for more complex tasks. To address these challenges, this research develops a classification system, based on Language Models, to automatically infers and annotates taxonomy relationships between legal documents, effectively overcoming the limitations of traditional taxonomy-based approaches. This dissertation benchmarks state-of-the-art abstractive summarization models tailored to the Italian legal domain, enhancing content accessibility by generating high-quality summaries of lengthy and complex legal texts. The thesis also introduces AI models for court judgment prediction and explanation, incorporating legal entities to improve accuracy and explainability. Additionally, it presents novel pipelines that use Large Language Models and Retrieval Augmented Generation to align AI-generated legal solutions more closely with specific case details, thereby improving legal reasoning and decision-making accuracy. Focusing primarily on the Italian legal domain, this work employs both quantitative and qualitative evaluation, to compare different approaches, including open- and closed- source Large Language models.

Results of this work reveal that integrating domain-specific techniques—such as named-entities and specialized pre-training—significantly enhances the performance, robustness, and explainability of AI models in legal document analysis tasks like classification, summarization, and court judgment prediction. In addition, incorporating collaborative multi-model approaches and advanced retrieval techniques significantly enhances AI systems' ability to perform complex reasoning tasks, which is crucial for improving the reliability of decision-making systems. However, for these tasks, the generalization capability of larger models prevails on smaller fine-tuned models. From an applicative perspective classification tasks language models are effective due to their robustness and scalability, though they are limited in addressing complex challenges. While LLMs show promising performance in text summarization and legal reasoning with fine-tuning, they face challenges in explainability and complex reasoning.

Contents

List of Figures

List of Tables

Symbols

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CoT	Chain-of-Thought
DL	Deep Learning
EU	European Union
GPT	Generative Pre-trained Transformer
GQA	Grouped-Query Attention
LLM	Large Language Model
LM	Language Model
LoRA	Low-Rank Adaptation
LR	Logistic Regression
MoE	Mixture of Experts
MRR	Mean Reciprocal Rank
NER	Named Entity Recognition
NLG	Natural Language Generation
NLP	Natural Language Processing

NLU Natural Language Understanding

PEFT Parameter-Efficient Fine-Tuning

PLMs Pretrained Language Models

RAG Retrieval Augmented Generation

SOTA State-of-the-art

SVM Support Vector Machine

TF-IDF Term Frequency-Inverse Document Frequency

ToT Tree-of-Thought

XMC eXtreme Multi-label Classification

Chapter 1

Introduction

The rapid advancement of Artificial Intelligence (AI), particularly in Natural Language Processing (NLP) and Deep Learning (DL), has opened up new avenues for innovation across various sectors. The legal domain, with its vast repositories of complex and nuanced textual data, stands to benefit significantly from these technological advancements.

In particular, Natural Language Understanding (NLU) and Natural Language Generation (NLG) solutions offer transformative possibilities for legal tasks, such as automating legal document summarization, assisting with case law analysis, and generating responses to legal inquiries. These solutions promise to significantly reduce time spent on labor-intensive tasks. However, despite their promises, practical application of NLU and NLG in the legal domain remains limited, especially in real-use cases [2]. This is probably caused by several factors: the specialized vocabulary of legal language, which can not be fully captured by generalistic LLMs, the need for contextually accurate interpretations, and the risk of errors in sensitive areas like case outcomes or regulatory compliance. Consequently fully integrated and reliable AI systems for legal work are still in early development stages, requiring further refinement to address these domain-specific demands accurately and effectively. For these reasons, this thesis explores the application of NLP and DL techniques to address these challenges and enhance legal workflows across three interconnected areas: automated legal document exploration, content accessibility, and support for advanced tasks requiring interpretability and reasoning capabilities.

Automated legal document exploration refers to the use of Natural Language Processing techniques to systematically extract relevant information from large volumes of legal documents. It enables efficient navigation, retrieval, and understanding of relevant legal content. Current legal systems face limitations in effectively organizing and accessing the vast amounts of legal information available. Traditional methods of document exploration rely on manual annotation and pre-defined taxonomies, which are often incomplete, time-consuming to maintain, and struggle to capture the evolving nature of legal concepts. Furthermore, the verbose and technical language of legal documents poses significant barriers to comprehension for both legal professionals and the general public.

Content accessibility in the legal context refers to the ability to easily obtain, understand, and interact with legal information and documents, ensuring that a broader range of users can easily access and benefit from legal resources. This issue is compounded by the fact that legal documents can be extremely lengthy, requiring significant time and effort to comprehend. The main challenge is due to the technical nature of legal language, which often includes precise and unambiguous expressions that are difficult to transform in other forms accurately for an automated system.

Finally, *reasoning capabilities* refer to the ability of an AI system to process information, draw conclusions, make inferences, and apply logic to solve problems or answer questions, particularly in complex scenarios where understanding context and relationships among concepts is crucial, such as in legal analysis or decision-making. While AI systems, particularly those based on machine learning and natural language processing, have demonstrated considerable potential in tasks like document review, case summarization, and even preliminary judgment prediction, they often fall short in replicating the sophisticated reasoning processes that underpin human legal decision-making. The absence of true legal reasoning in AI applications poses significant risks, as decisions based solely on pattern recognition or statistical correlations may lack the depth and contextual understanding necessary for just and accurate outcomes. Support for advanced tasks involves the development of AI systems that can not only analyze complex legal scenarios and provide insights but also offer transparent explanations for their conclusions, enabling legal professionals to understand the rationale behind decisions and ensure accountability in the application of legal reasoning. Outcome reliability is a critical challenge in deploying AI systems for legal decision-making. Legal professionals, such as judges and lawyers, often hesitate to fully trust AI-generated outcomes due to concerns about accuracy,

transparency, and accountability. The stakes in legal decisions are high—errors can lead to significant consequences, including miscarriages of justice or the misapplication of law. Therefore, for AI systems to be trusted in legal contexts, they must not only predict outcomes accurately but also provide clear, reliable explanations that legal professionals can scrutinize and understand.

Most of approaches for automated legal document exploration often struggle with the evolving nature of legal systems, which can lead to outdated or unreliable information over time. They often rely on human annotation, which is a time-consuming activity. Language limitations are also a challenge, as most solutions are tailored to English, reducing their effectiveness in non-English contexts. Techniques such as word-level vector representations and contextualized embeddings often face performance issues, especially when domain-specific training data is scarce. Transformer-based models, while powerful, can have difficulty handling the complexity and specialized vocabulary of legal texts, leading to reduced accuracy in certain scenarios. Furthermore, Language Models struggle with processing long documents, managing complex multi-label classifications, and dealing with the uneven distribution of annotated data across various legal domains.

Automated summarization techniques have been developed to reduce the verbosity, the complexity and the presence of redundant information in legal documents, but they encounter limitations when applied to legal texts, particularly those written in languages other than English. Most existing summarization approaches are extractive, meaning they select and combine portions of the original text, which often results in summaries with low readability. Abstractive summarization, which generates summaries by rephrasing the content, is more suitable but less commonly used due to the challenges in capturing the nuances of legal language, especially in Italian. Additionally, current models are often trained on English data and adapted to other languages through translation, which fails to fully leverage the benefits of language-specific models. Newer models and datasets tailored specifically to the Italian legal domain aim to address these issues, but challenges remain in fine-tuning these models for different document types and ensuring they can process longer texts effectively.

Researchers are significantly improving the the reasoning capability and the trust of language models in legal contexts, particularly in court judgment prediction and explanation. Legal tasks, such as predicting court judgments, are complex due to

the nuanced and technical language of legal documents, the variability in norms across different legal domains, and the often extensive length of these documents. Transformer-based models have shown promise in understanding forensic language and predicting judgments, but issues like document length and model specialization remain problematic. Additionally, although different approaches have been proposed to enhance the interpretability of large language models, their effective application remains a subject of ongoing debate.

This thesis addresses the following research questions:

- **RQ1**: How can AI techniques be leveraged to automatically explore and organize legal documents, overcoming the limitations of traditional taxonomy-based approaches? How can LM-based solutions be used to improve scalability and robustness in real applications?
- **RQ2**: How can AI be utilized to improve the accessibility of legal documents, particularly for non-experts and in multilingual contexts?
- **RQ3**: What trade-offs exist between the generalization capabilities of closed-source Large Language Models (LLM) and the domain-specific accuracy of fine-tuned smaller models in a real-word setting?
- **RQ4**: How can AI be integrated into legal processes for solving more complex task to enhance accuracy, efficiency and faithfulness, and transparency while ensuring reliability?
- **RQ5**: How can open-source LLMs be enhanced to compete with the generalization performance of proprietary models in the legal domain?

Research contribution This research contributes to the advancement of Legal AI by addressing critical challenges in document exploration, content accessibility, and decision-making. By developing and evaluating novel AI models, this thesis provides insights into the potential of AI to transform legal practices, making legal information more accessible, efficient to process, and supporting more informed and transparent decision-making. This work has the potential to benefit legal professionals, researchers, and the wider public by promoting a more efficient and equitable legal system.

1. Develop a robust deep learning-based classification system to automatically infer and annotate taxonomy relationships between legal document pairs, addressing challenges of evolving and incomplete taxonomies in legal content retrieval. Propose a classification pipeline with ad-hoc models to effectively classify Italian legal documents of varying lengths and complexities using Pre-trained Language Models, and evaluate it in a real-use case (**RQ1**).

2. Benchmark various state-of-the-art abstractive summarization models tailored to the Italian language on a proprietary dataset of Italian legal news, highlighting the effectiveness of these models and the importance of handling longer text for generating high-quality legal news summaries. Introduce new resources for Italian legal documents summarization, with extensive empirical and manual evaluation against other language models, also in a real word context (**RQ2** and **RQ3**).

3. Develop and evaluate AI models for court judgment prediction and explanation, focusing on incorporating legal entities to improve accuracy and explainability, and addressing the challenges of handling long and complex legal documents (**RQ4**)

4. Present novel pipelines for case resolution with Large Language models and Retrieval Augmented Generation (RAG). We propose new techniques to better align AI-generated legal solutions with the specific details of a case, improving overall accuracy and improve legal reasoning by refining the retrieval and use of relevant case information (**RQ5**).

This thesis focuses primarily on the Italian legal domain, with some investigations into legal documents, including also the Italian landscape.

The legal solutions can vary depending on the underlying legal system, making the dependence on the legal framework a crucial factor to consider in any legal analysis. Additionally, the Italian language remains underrepresented in legal research, further highlighting the importance of this study. The scope of the research encompasses various legal document types, including legislation, regulations, court judgments, and legal news articles. The primary focus is on supervised machine learning, deep learning, and transformer-based models and large language models and their techniques to adapt them in downstream tasks. The limitations of the

study include the availability of annotated data, computational resources, and the generalizability of the findings to other legal domains and languages.

This thesis employs a combination of quantitative and qualitative research methods. Qualitative analysis is employed to interpret the results, understand the limitations of the proposed approaches, and gain insights into the potential impact of AI on legal practices with domain experts.

Dissertation outline This thesis discusses the role of Natural Language Processing (NLP) in addressing these issues and outlines the thesis objectives. Following this, the main topics are presented, which focus on three key areas: the automatic exploration of legal documents, improving their accessibility, and enhancing decision-making processes.

Chapter 2 covers the history of NLP, from early theoretical work and statistical approaches to the neural network breakthrough. It also delves into the concept of transfer learning and the specific challenges of applying NLP in the legal domain.

Chapter 3 reviews prior work, particularly the significance of taxonomies and document classification in the legal field. It proposes new methodology for the extraction of taxonomy relationships among legal judgments, including problem formulation, experimental setups, and results. Additionally, it highlights the importance of named entities in legal documents and presents new methodologies for scaling language models to real-world scenarios, presenting a detailed analysis of problem statements, methodologies, and outcomes.

Chapter 4 reviews existing efforts to make legal documents more accessible. It includes benchmarking Italian summarization models and presents new resources such as curated datasets and models, with detailed methodology, experimental setups, and results.

The last chapter, the thesis evaluates prior works in court judgment prediction, legal reasoning, and retrieval-augmented generation. It proposes improvements in court judgment prediction, detailing the problem statement, methodologies, experimental designs, and results. It further evaluates the performance of large language models on legal problem-solving and concludes with the development of an end-to-end pipeline for legal information retrieval and resolution.

The 6 summarizes the contributions and findings, suggesting future directions for advancements in the application of NLP to the legal domain.

Chapter 2

Background

2.1 Natural Language Processing: A brief history

2.1.1 Early Theoretical Work

Natural language processing (NLP) is a field of artificial intelligence that deals with analyzing, understanding, and generating human language. Natural Language Processing (NLP) has its roots in the 1940. The journey of NLP is intertwined with the history of machine translation, speech recognition, and artificial intelligence.

The history of machine translation dates back to the seventeenth century, with philosophers like Leibniz and Descartes proposing codes for relating words between languages. However, these remained theoretical until the mid-1930s when the first patents for "translating machines" were applied for. In the early 1900s, Ferdinand de Saussure, a Swiss linguistics professor, developed an approach describing languages as "systems" where a sound represents a concept that shifts meaning as the context changes. This laid the foundation for what has come to be called the structuralist approach. In 1950, Alan Turing published "Computing Machinery and Intelligence" [3], proposing what is now called the Turing test as a criterion of intelligence. This criterion depends on the ability of a computer program to impersonate a human in a real-time written conversation.

The Georgetown experiment in 1954 involved fully automatic translation of more than sixty Russian sentences into English. The authors claimed that within three or

five years, machine translation would be a solved problem. However, real progress was much slower, and after the ALPAC report in 1966, funding for machine translation was dramatically reduced. Three years later, in 1957, the Noam Chomsky's idea of a "universal parser" that could read and understand any text was proposed. However, the constraints of existing computing power and theoretical frameworks made this an unrealistic goal at the time. During the 1960s, some of the foundational theoretical work in NLP was laid down. This included the development of language models like noam-transformational grammar and stratificational grammar as well as early work on question answering systems.

Some notably successful NLP systems developed in the 1960s were SHRDLU, a natural language system working in restricted "blocks worlds" with restricted vocabularies developed at MIT. While limited in scope, it demonstrated the potential for computers to process and understand typed language input. In 1969, Roger Schank introduced the conceptual dependency theory for natural language understanding. In 1970, William A. Woods introduced the augmented transition network (ATN) to represent natural language input.

A major milestone in 1983 was the creation of the first statistical language model using n-grams at IBM. This marked a shift from purely symbolic and rule-based approaches to more probabilistic and data-driven methods. Throughout the 80s and 90s, there was vigorous debate between proponents of symbolic, rule-based NLP systems and those favoring statistical techniques. Both approaches made important advances, but encountered challenges with scaling up to handle broad domains and varieties of language.

2.1.2 Early statistical approaches

The 1990s saw the rise of data-driven, statistical NLP with the application of methods like decision trees, maximum entropy models, and neural networks to language tasks. One key breakthrough was the use of hidden Markov models for speech recognition by the late 1990s. However, symbolic and knowledge-based techniques continued to play an important role alongside statistical algorithms. Rule-based systems excelled at capturing linguistic constraints, while statistical methods were good at leveraging data patterns.

While applying statistical methods, language processing and generation tasks were based on the concept of n-grams. An n-gram is defined as a contiguou sequence of n items from a given text snippet. They can be defined in terms of words or characters. For example, a word-based bi-gram (where n is set to 2) for the sentence "example of a sentence" will include "example of", "of a", "a sentence" and so on. Similarly, a char-based bi-gram will be "ex", "xa", "mp" and so on. In both cases, the idea is to take a sequence of items and apply statistical methods to find out which of them are likely to occur together. This simple yet effective definition allowed the design of new innovative systems in the field of text mining, information retrieval and, text generation to name a few. Leveraging the n-gram definitions it was possible to define new way for representing documents according to their constituents' words.

Traditional Text Vectorization Approaches. Early methods in natural language processing involved mapping text into vector representations, a process known as text vectorization. Seminal approaches in this domain leveraged the co-occurrence of words within the text to generate vector representations, a technique referred to as the bag-of-words model. Given a document D containing a sequence of words $(w_1, w_2, \ldots, w_n) \in D$, the bag-of-words representation of D corresponds to a vector where each dimension refers to a specific word w_i. The vector cells are binary, with a value of 1 if the corresponding word appears in the document, and 0 otherwise. Despite their historical significance and continued use in some commercial systems, these early approaches exhibit major limitations. Firstly, the high dimensionality of the vectors, a phenomenon known as the curse of dimensionality, renders vector manipulation challenging. Secondly, the sparsity of the vectors, with many cells set to 0, makes computation difficult and memory-intensive. Lastly, these vectors cannot encode contextual information; consequently, words that are semantically related within the same context are not represented by similar vectors.

Vector space models. A fundamental issue with the bag-of-words approach is the binary encoding of word presence in a document, which fails to capture word frequency information. To address this limitation, the Vector Space Model (VSM) was developed in the context of information retrieval and has been extensively employed in various natural language processing applications. Under the VSM framework, each word in a document is encoded according to its frequency, rather than mere presence. Formally, given a document D containing a sequence of words

$(w_1, w_2, \ldots, w_n) \in D$, the representation of D is a vector where each dimension corresponds to a specific word w_i, and the value of each vector cell is set to the frequency of the corresponding word within the document. This encoding can be viewed as a generalization of the bag-of-words approach, providing a more accurate representation of the documents. Furthermore, the methodology has been extended by replacing word frequency with the Term Frequency-Inverse Document Frequency (TF-IDF), which introduces an additional term, the inverse document frequency (IDF), to down-weight words that appear frequently across many documents (i.e., less discriminative words). Although more accurate than bag-of-words, the VSM approach still fails to address issues related to sparse vectors, high dimensionality, and the inability to encode contextual information.

Latent Semantic Analysis (LSA). Latent Semantic Analysis (LSA) [4] is a technique based on Singular Value Decomposition (SVD) [5], a linear algebra method used for matrix compression and extraction of latent semantic relationships within the data. In the context of LSA, the matrix in question is the term-document matrix, where each row corresponds to a term in the vocabulary and each column corresponds to a document in the collection. LSA projects document representations into a lower-dimensional space via singular value decomposition of the term-document matrix. Consequently, the information contained in the high-dimensional matrix is condensed into a lower-dimensional space, and documents are represented as linear combinations of the latent concepts extracted from the original matrix. In contrast to bag-of-words and VSM, LSA does not consider words in isolation but rather considers the relationships among them. It overcomes the issues of data sparsity and high dimensionality by creating a new, lower-dimensional space that captures the latent concepts of the original sparse matrix. While LSA addresses the sparsity issue, it still fails to consider contextual information and word order, thus limiting its ability to identify the semantic meaning of words and their sequential relationships. With the advent of deep learning in the early 2000s, researchers began to focus on the semantic level of language, aiming to learn representations that capture latent semantic relationships among words.

2.1.3 The advent of Neural Networks

In 2010s deep learning neural networks achieved state-of-the-art performance across many NLP tasks, like machine translation, question answering, text summarization, and more.

Contextualized Word Representations. Recognizing the inherent limitations of traditional text vectorization techniques, the this generation of text embedding models leverages deep learning methodologies. These models were predicated upon the distributional hypothesis, which posits that words appearing in similar contexts tend to exhibit analogous semantic meanings.

The pioneering approaches in this domain employed shallow neural network architectures to generate dense, distributed vector representations for individual words. Prominent works of this kind include Word2Vec [6] and GloVe [7]. These models leveraged self-supervised learning paradigms to automatically derive effective word representations without necessitating manual annotation. Their training procedure relies on the definition of a context window C, which delineated the context associated with a given target word w as the concatenation of the k preceding and k subsequent words in the text sequence. The Word2Vec model comprised a single fully-connected layer that could be trained using two distinct methodologies:

- **CBOW (Continuous Bag of Words):** given a context C as input, the model is trained to predict the target word w.
- **Skip-Gram:** The model takes the target word w as input and is trained to predict the constituent words of the context C.

An illustrative depiction of both training procedures is provided in Figure 2.1. The model mapped each unique word in the training corpus to a singular vector representation. Initially, word vectors were randomly initialized and subsequently fine-tuned during the training process using the CBOW or Skip-gram training objectives. Notably, the training process did not mandate human annotation and could be executed in an unsupervised setting. Upon training completion, word embeddings demonstrated a proficiency in capturing semantic relationships among words. The similarity between two words (w_i, w_j) was commonly quantified by computing the cosine similarity between their vector representations $(\vec{w}_i, \vec{w}_j)$:

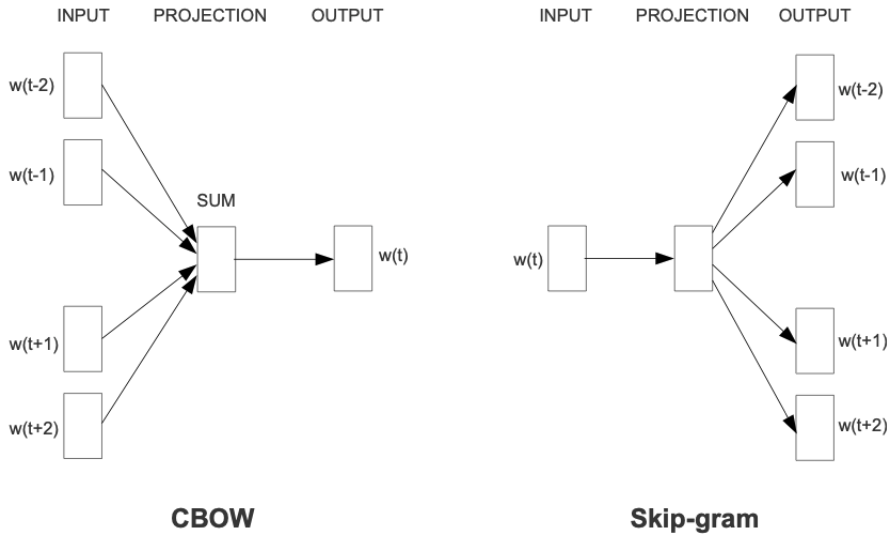


Fig. 2.1 CBOW and Skipgram training procedure.

$$\text{sim}(w_i, w_j) = \frac{\vec{w}_i \cdot \vec{w}_j}{\|\vec{w}_i\| \|\vec{w}_j\|} \tag{2.1}$$

This similarity measure emerged as the de facto standard for quantifying the degree of semantic similarity between word pairs. Word embedding models have been widely adopted in both research and commercial natural language processing systems to circumvent the shortcomings of traditional methods, such as the bag-of-words approach. Nonetheless, they exhibited lingering limitations in the domain of semantic content understanding:

1. **Out-of-vocabulary words (OOV):** Vector representations were generated solely for words present in the training data. For novel words, these models proved incapable of inferring the corresponding vector representation.

2. **Contextualized representation:** Each word was represented by a single, static vector. During training, the context was utilized to learn the vector mapping, but once trained, the vector remained invariant, irrespective of the context in which the word was encountered.

3. **Sentence and document representation:** Text snippets were vectorized by averaging the vectors of their constituent words, thereby obfuscating the sequential structure and context of the sentence.

The OOV issue was partially mitigated by leveraging sub-word units and word compositionality [8]. The sub-words were generated using character n-grams, enabling the extraction of vector representations for out-of-vocabulary words by summing the vectors of their constituent sub-words. However, limitations regarding contextualized representations and text snippet encoding remained unresolved. The static nature of word embeddings, coupled with the aggregations required for document/sentence representations, adversely impacted the semantic understanding of text.

Recurrent neural networks. Recurrent Neural Networks (RNNs)[9] have emerged as a prominent class of models for natural language processing tasks due to their ability to capture long-range dependencies while remaining computationally tractable. In the context of word prediction, the task is formulated as a discriminative problem, with the primary objective being to determine the conditional probability of a word given its context, which depends on the preceding words:

$$\mathbb{P}(w|u) = \frac{\beta_w \cdot v_u}{\sum_{w' \in \mathscr{V}} \exp(\beta_{w'} \cdot v_u)} \tag{2.2}$$

where $\beta_w \cdot v_u$ represents a dot product between two K dimensional vectors. A natural language model can be constructed from a recurrent neural network by iteratively updating the context vectors as the model progresses through the sequence. Given $x_m \widehat{=} \phi_{w_m}$ as the word embedding of the words w_m, h_m, the contextual information at position m is:

$$h_m = \text{RNN}(x_m, h_{m-1}) \tag{2.3}$$

Then the RNN language model can be defined as:

$$p(w_{m+1}|w_1, w_2, ..., w_m) = \frac{\exp(\beta_{w_{m+1}} \cdot h_m)}{\sum_{w' \in \mathscr{V}} \exp(\beta_{w'} \cdot h_m)} \tag{2.4}$$

The recurrent operation enables the model to consider all the information about the sequence when processing a word at time m, without imposing limitations on the context length. The model parameters, representing the parameters of the conditional distribution, are updated via *backpropagation through time*. The gradients at time m depend on all the previous gradients $n < m$, and the size of the computational graph is contingent upon the input length. Repeated application of non-linear functions may lead to the exploding gradients or the vanishing gradients problem. While the former can be addressed by clipping the gradients above a certain threshold, the latter, being a more complex issue, necessitates architectural modifications. In particular, two variants of RNNs have been developed: Long short-term memory (LSTM) [10] and Gated recurrent unit (GRU) [11].

LSTMs are equipped with a *memory cell* c_m, which contributes to the formation of the next states h_m. The advantage of this architecture is that the memory cell does not pass through a squashing function, allowing information to flow through the network without vanishing.

The gates are functions that control the amount of information propagating through the network. They compute the sigmoid of the linear combination of inputs and the previous hidden states. There are different types of gates:

- Forget gate: $f_{m+1} = \sigma(\Theta^{(h\to f)} h_m + \Theta^{(x\to f)} x_{m+1} + b_f)$
- Input gate: $i_{m+1} = \sigma(\Theta^{(h\to i)} h_m + \Theta^{(x\to i)} x_{m+1} + b_i)$
- Output gate: $o_{m+1} = \sigma(\Theta^{(h\to o)} h_m + \Theta^{(x\to o)} x_{m+1} + b_o)$

The cell memory is given by:

$$c_{m+1} = f_{m+1} \odot c_m + i_{m+1} \odot \tilde{c}_{m+1} \tag{2.5}$$

where $\tilde{c}_{m+1} = \tanh(\Theta^{(h\to c)} h_m + \Theta^{(x\to c)} x_{m+1})$ is the updated candidate. The output of each state is given by $h_{m+1} = o_{m+1} \odot \tanh(c_{m+1})$. The memory cell of the previous state contributes to the memory of the subsequent state, and, indirectly, also to the subsequent output, thereby mitigating the gradient vanishing problem.

The Transformer model. The most significant novelty introduced by transformer-based models is the attention mechanism, which is responsible for learning the

relationships between elements within a sequence. This mechanism enables the model to selectively focus on relevant parts of the input sequence when generating a specific output, effectively capturing long-range dependencies without the recurrent computational overhead. In contrast to recurrent neural networks, which process input sequences in a strictly sequential manner, transformer models leverage self-attention mechanisms to enable parallel computation across all elements of the input sequence. This parallelization capability, coupled with the ability to model long-range dependencies, has contributed to the transformers' superior performance and computational efficiency compared to their recurrent counterparts.

Self-Attention Mechanism. The attention mechanism enables the transformer model to selectively focus on relevant portions of the input sequence when generating a specific output representation. Considering a sequence of n ordered tokens $\mathscr{S} = w_1, w_2, \ldots, w_n$, the model represents each element w_i with a query vector q_i, a key vector k_i, and a value vector v_i. For a given token pair (w_i, w_j), the self-attention score is computed as the dot product between the query vector q_i and the key vector k_j. The self-attention scores quantify the relative importance of each token for the semantic representation of the token under consideration. To compute the context vector for a given token w_i, the self-attention scores are normalized via the softmax function to obtain a probability distribution over the tokens. The context vector is then obtained as a linear combination of the value vectors v_j of all tokens, weighted by the normalized attention scores. Unlike recurrent networks, transformer models enable parallelized computation of context vectors, improving computational efficiency.

However, self-attention requires computing attention scores between each pair of tokens in the sequence, resulting in a quadratic complexity $\mathscr{O}(n^2)$ with respect to the sequence length n. Consequently, the inputs and outputs of modern transformer architectures are limited to a fixed threshold. Recent advances in transformer architectures have introduced the concept of sparse self-attention [], allowing the model to focus on a subset of the entire text and process longer sequences, thereby improving computational efficiency.

Natural language understanding models. Natural Language Understanding (NLU), a subfield of Natural Language Processing (NLP), aims to extract struc-

tured information from natural language inputs. In contrast to sequence-to-sequence tasks, where both the input and output of the model are sequences of tokens, NLU tasks analyze the input sequence and provide a vector representation for each token in the sequence. This objective aligns precisely with the function of the encoder component in transformer models, which is trained to convert the input sequence into a sequence of vector representations.

Example of NLU models includes:

- BERT (Bidirectional Encoder Representations from Transformers) [12] is an encoder model based on the transformer architecture, designed to address natural language understanding (NLU) tasks. It employs a two-step process: pre-training and fine-tuning. During the pre-training stage, the model learns to create an effective representation of language by exploiting large unlabeled corpora. In the subsequent fine-tuning step, the pre-trained model is adapted to specific NLU tasks by leveraging task-specific annotated data. BERT leverages two tailored pre-training tasks: i) **Masked Language Modeling**, the model is presented with a text sequence containing masked words (i.e., each word is replaced with the standard [MASK] token), and its objective is to predict the original masked words based on the surrounding context, ii) **Next Sentence Prediction** the model is trained to recognize which sentence pairs should be considered contiguous.

- RoBERTa (Robustly Optimized BERT Pretraining Approach) [13] proposes modifications to the BERT architecture and training procedure to improve performance: i) it expand the BERT architecture by increasing the number of parameters and training data ii) they introduce a dynamic masking procedure to enhance variability and robustness iii) remove the Next Sentence Prediction (NSP) objective, asserting that its removal has a limited impact on performance.

Natural language generation models. Natural language generation (NLG) is the process of creating text with a specific meaning using natural language. In contrast to natural language understanding (NLU), the primary objective of NLG is not to comprehend a text, but rather to generate new text with a certain intended meaning.

The pioneering works in this field, Generative Pre-trained Transformer (GPT)[14–16] models have demonstrated remarkable performance in various NLG tasks, such

as dialog generation. These models employ the original transformer decoding architecture and incorporate self-supervised learning for model pre-training. With respect to the language generation objective, the pre-training strategy is formulated as a language modeling task, where the model is trained to predict the next word in a sequence given the previous context. These models have exhibited impressive generalization capabilities and have outperformed many prior models on various NLG tasks.

From language models to large few-shot learners and their democratization. The authors of the third GPT model (GPT-3) [15] asserted that the capabilities of a language model could not be limited to a single task at a time. On the contrary, the advantage of having a language model capable of generating text enabled tackling diverse tasks, which could be solved even with few examples. Research attention has turned to few-shot learning and zero-shot learning: in the former, the model can learn from very few examples (up to 10). In zero-shot learning, however, the number of examples seen during training is zero, hence measuring the model's ability to perform on unseen tasks. The objective of these two problem setups is to demonstrate the model's generalization capabilities. To achieve performance approaching human-level, however, it was necessary to increase the number of parameters. From here, subsequent works such as Big Science BLOOM [17], Google PALM [18] and Gemini [19], Claude (see: https://www.anthropic.com/news/claude-3-family) emerged: the size of such models quickly reached and surpassed 100B parameters.

Training such models is a task limited to companies and universities with the necessary hardware resources. A change in direction came from Meta: the Llama model family was created with the purpose of democratizing access to such resources. These models can fit on consumer GPUs, with a variable number of parameters ranging from 1 to 8B. Despite their reduced size, these models are capable of achieving and surpassing the performance of much larger competitors like GPT-3.

Among examples of reduced-size models, in addition to the Llama model family, we find:

- Mistral [20] from MistralAI: uses Grouped-query attention (GQA) [21] for faster inference and Sliding Window Attention (SWA) [22] to handle longer sequences efficiently. They introduce the Mixture of Experts models (MoEs),

a powerful technique that enhances AI models by dividing them into separate sub-networks, or "experts," each specializing in different input data.

- Phi [23] from Microsoft: Phi-1 achieved state-of-the-art performance on Python coding among existing LLMs. Phi-1.5 [24] extended the focus to common sense reasoning and language understanding with 1.3 billion parameter with performance comparable to models 5x larger. And Phi-2 (see https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/), a 2.7 billion-parameter language model, demonstrates outstanding reasoning and language understanding capabilities, showcasing state-of-the-art performance among base language models with less than 13 billion parameters. On complex benchmarks Phi-2 matches or outperforms models up to 25x larger, thanks to new innovations in model scaling and training data curation.

- Falcon [25]: is a generative large language model created by the Technology Innovation Institute (TII) in Abu Dhabi. It's designed to advance applications and use cases across various domains.

Other techniques, such as (**PEFT**) [26], allows to adapt the pre-trained model to the designated task. The benefits of PEFT include lower memory requirements, allowing mixed-task batches during inference and fine-tune a small number of (extra) model parameters while freezing most parameters of the pretrained LLMs. Specifically, we use LoRA [27] which is a lightweight fine-tuning strategy aimed to freeze the pre-trained model weights and inject trainable rank decomposition matrices into each layer of the Transformer architecture thus greatly reducing the number of trainable parameters for downstream tasks.

The Artificial General Intelligence The advent of such models marks an epochal shift in the field:

- The number of tasks on which to benchmark these models is increasing: now, to create a new benchmark, it is no longer necessary to have thousands of data points; even just 100 examples can suffice to measure a model's capabilities (in zero-shot learning).

- The types of tasks are shifting towards reasoning capabilities: tasks such as summarization and sentiment analysis are no longer sufficient; these models can already perform such tasks well. To fully assess a language model's capabilities, it is necessary to measure its ability to express reasoning of various kinds, such as logical or mathematical reasoning.

- Prompting: A significant portion of the work has focused on understanding which prompting techniques can be most effective for solving problems with few examples. From this, techniques such as Least-To-Most Prompting [28] (Decompose a complex problem into a series of simpler sub-problems and subsequently solving for each of these sub-questions), Chain-of-Thought (CoT) Prompting have emerged (introduced in [29], it enables complex reasoning capabilities through intermediate reasoning steps), Tree of Thought (ToT) [30] (for complex tasks that require exploration or strategic lookahead, ToT generalizes over chain-of-thought prompting and encourages exploration over thoughts that serve as intermediate steps for general problem-solving with language models), and Active-Prompt [31] (the most uncertain predictions of the language model become the examples to annotate and prompt via Chain-of-Thought) and Self-Consistency [32](Generate a diverse set of reasoning paths and select the most consistent output for the final answer).

- The Artificial General Intelligence and alignment with humans: the creation of a collective general intelligence at the service of humanity. Many of the most-cited AI scientists, including Geoffrey Hinton, Yoshua Bengio, and Stuart Russell, argue that AI is approaching human-like and superhuman cognitive capabilities and could endanger human civilization if misaligned. AI alignment, theorized as early as the 1960s, is a subfield of AI safety, the study of how to build safe AI systems.

The increasing scale and generalization capabilities of these models have profound implications for the field of AI. While they offer unprecedented opportunities, they also raise concerns about safety and alignment with human values, necessitating rigorous study and ethical considerations to ensure their responsible development.

2.2 Transfer learning

The availability of large-scale dataset corpora and the adoption of large pre-trained language models allow advances in different NLP tasks. Creating large datasets for a new domain, however, is often infeasible and highly costly. Thus, the ability to transfer knowledge from large pre-trained models to new domains with little or no in-domain data is necessary, especially when such models should be adopted in real-life applications.

Transfer learning [33] is the technique of knowledge transfer or transfer learning between task domains. If a domain is denoted as $\mathscr{D} = \{\mathscr{X}, P(X)\}$, where $\mathscr{X}$ is the feature space and $P(X)$ is the marginal probability distribution over that feature space and a task is represented by $\mathscr{T} = \{\mathscr{Y}, P(Y|X)\}$, where $\mathscr{Y}$ is the label space and $P(Y|X)$ is the target posterior probability distribution, a formal definition of transfer learning can be the following: defining a source domain $\mathscr{D}_{\mathscr{S}}$ and its corresponding source task $\mathscr{T}_{\mathscr{S}}$, and the target domain as $\mathscr{D}_{\mathscr{T}}$ with its target task $\mathscr{T}_{\mathscr{T}}$, the objective of transfer learning is to learn the target conditional probability distribution $P(\mathscr{Y}_{\mathscr{T}}|\mathscr{X}_{\mathscr{T}})$ (where $\mathscr{Y}_{\mathscr{T}}$ and $\mathscr{X}_{\mathscr{T}}$ are the training data of the target domain) in $\mathscr{D}_{\mathscr{T}}$ with the information gained from $\mathscr{D}_{\mathscr{S}}$ and $\mathscr{T}_{\mathscr{S}}$ where $\mathscr{D}_{\mathscr{S}} \neq \mathscr{D}_{\mathscr{T}}$ or $\mathscr{T}_{\mathscr{S}} \neq \mathscr{T}_{\mathscr{T}}$.

Chapter 3

The automatic exploration of documents in the legal domain

This chapter provides a review of the related literature in Section 3.1 and introduces the research contributions for improving content search in the legal domain. It provides the details of methodologies for the extraction of taxonomy relationship in Section 3.2, legal document classification in Section 3.4 and Section 3.3.

3.1 Prior works

3.1.1 The importance of taxonomies in the legal domain

Legal databases comprise a broad spectrum of documents, including legislative texts, regulatory guidelines, judicial decisions, and legal principles [34]. To assist legal practitioners in efficiently navigating these extensive resources, sophisticated retrieval systems are imperative. However, the intricate nature of legal language, the exponential growth of digital resources, and cross-national differences in legal frameworks render this task particularly challenging [35].

The process of legal content exploration is heavily dependent on human-generated annotations within specialized taxonomies [34]. A legal taxonomy systematically organizes key terms into hierarchical structures, where relationships such as parent-

child and sibling connections are defined. Each document may be tagged with multiple relevant terms.

Legal experts utilize these taxonomy relationships to navigate legal documents, often accessing supplementary materials through *Related-to* links. Although useful, taxonomy-based retrieval systems encounter several challenges:

- As legal systems evolve, taxonomies undergo changes, resulting in *temporal concept drift*, which can compromise the accuracy and reliability of established relationships [36, 34].
- The relationships within taxonomies, particularly *Related-to* links, are frequently incomplete.
- A significant proportion of electronic legal documents and their annotations are available only in English, with limited solutions for other languages.

Prior works analyze legal document similarities using rule-based approaches [37, 38], document-level text similarities [39], graph-based methods [40], and machine learning solutions [41, 42]. These work focus on English documents.

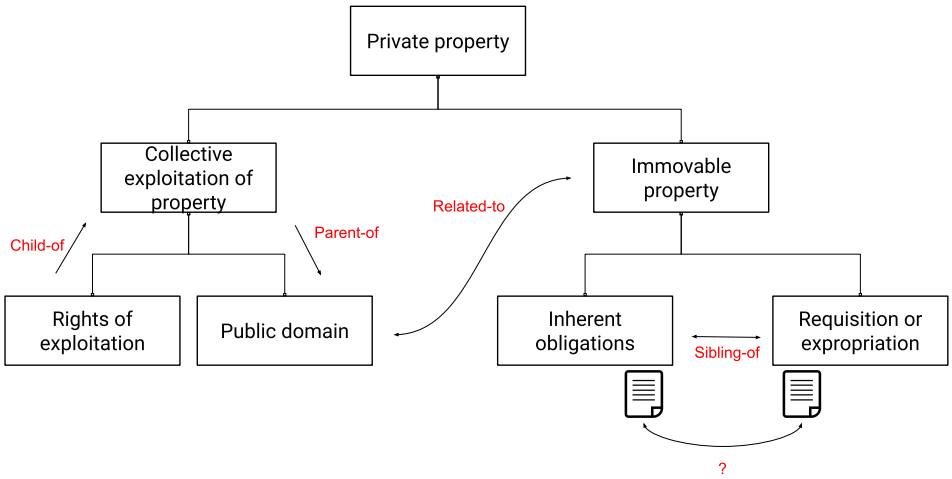


Fig. 3.1 Example of the taxonomy: in red we highlighted the relationship types we aim at identifying with our classification system.

3.1.2 Legal document classification

Classification techniques are employed to automatically assign a label from a predefined set of classes to a given document. In the legal domain, classifiers are particularly valuable as they reduce the burden on domain experts by automating the annotation process.

A prevalent application of document classification in the legal field is the automatic categorization of court cases, where the primary objective is to predict the relevant area of law for a given case. Existing research in this area has primarily focused on leveraging machine learning and deep learning methodologies [43–46]. Parallel studies have also investigated the automatic text classification of legislative documents to identify the law topic, with a particular emphasis on monolingual datasets [47–53]. Research exploring multilingual datasets of legislative documents is relatively scarce[54]. Notably, [52] examines the semantic relationship between each document and its corresponding labels, though its performance on English documents remains constrained. In contrast, transformer-based approaches, such as those proposed in [54, 47, 49] represent the current state-of-the-art for English-language legal documents.

Language Models (e.g., [55, 51]) have demonstrated significant efficiency and effectiveness, largely due to their utilization of attention mechanisms [56]. However, the application of LMs for classification within the legal domain must contend with several inherent challenges posed by the complexity of legal documents.

Firstly, legal documents can vary significantly in length, ranging from brief texts, such as maxims, to extensive ones, such as contracts or judicial decisions. Most LMs are not well-equipped to process very long texts and are often not directly transferable across different document types [57]. Secondly, legal documents are frequently associated with multiple labels simultaneously, where candidate labels may exhibit arbitrary semantic relationships, often structured within hierarchical frameworks. Thirdly, the distribution of human-annotated data is typically uneven across different areas of law and document types, complicating model training.

3.2 Extracting taxonomy relationships among legal judgment

In [58]To address the challenge of managing taxonomies, we propose a classification system leveraging Deep Natural Language Processing techniques to automatically infer taxonomy relationships between pairs of legal documents. Utilizing a proprietary dataset of Italian legal judgments, primarily focused on private property law, our system applies supervised machine learning to predict relationships such as *Parent-of/Child-of*, *Sibling-of*, or *Related-to*. This automated annotation process significantly improves the efficiency and effectiveness of legal content retrieval, particularly in scenarios where the taxonomy is incomplete or outdated.

The preliminary results obtained from a real-world application indicate that: 1) Traditional word-level vector representations, such as Doc2Vec [59], are effective in this domain, as they successfully capture syntactic relationships between terms like "Patents" and "Trademarks," thereby facilitating classification. 2) Contextualized embeddings, such as those generated by BERT [12], underperform in comparison to Doc2Vec for Italian documents, primarily due to the scarcity of domain-specific training data.

3.2.1 Problem formulation

Consider a legal document d_i and a legal taxonomy $\mathscr{T}$, which is composed of a set of labels $\mathscr{L}$ used to annotate documents. Let $L_i \subseteq \mathscr{L}$ represent the specific labels assigned to d_i. The taxonomy $\mathscr{T}$ consists of a series of hierarchical structures where each label in $\mathscr{L}$ corresponds to headwords that describe the topic of a document. An *Is-a* relationship within this hierarchy defines the *Parent-of* and *Child-of* relationships between any two labels $l_i, l_j \in \mathscr{L}$. A *Sibling-of* relationship indicates that two labels, l_i and l_j, share the same parent, while a *Related-to* relationship links labels l_i and l_j that are semantically related.

Given a pair of annotated legal documents (d_i, d_j), the goal is to infer the taxonomy relationships between the labels l_i and l_j. To achieve this, we define a classification function f that predicts the type of relationship between the labels in l_i and l_j. The possible relationships include *Sibling-of*, *Parent-of/Child-of*, *Related-to*,

or *Other* if no specific relationship exists. For simplicity, this problem is framed as a single-label classification task.

The function f is formulated as the composition of two functions $(h \circ g)(d_i, d_j) = h(g(d_i, d_j))$. The first function, $g(d_i, d_j)$, generates high-dimensional vector representations $e_i, e_j \in \mathbb{R}^N$ for the documents. A classification model is subsequently trained on these vector representations to compute $h(e_i, e_j)$, which determines the relationship type r_{l_i,l_j}. During inference, the relationship type is predicted based solely on the document content, independent of any pre-existing annotations l_i and l_j.

3.2.2 Experimental settings

Dataset The proprietary dataset utilized for empirical validation consists of Italian legal judgments and maxims, predominantly related to property law. Each document within this dataset is annotated with one or more labels that correspond to pertinent legal principles. These principles are organized into pairs based on their relationship type, categorized as *Sibling-of*, *Parent-of/Child-of*, *Related-to*, or *Other*.

The training dataset is structured as triples (d_i, d_j, r_{l_i,l_j}), where each triple represents a pair of documents (d_i, d_j) along with the relationship r_{l_i,l_j} between their associated legal topics.

Models Text representation methods considered:

- **TF-IDF**: A word-level, occurrence-based text representation technique [60].
- **Doc2Vec**: A sentence-level embedding model that extends the Word2Vec methodology [61].
- **Multilingual BERT**: A model for generating contextualized embeddings [12]. Retraining BERT from scratch on our dataset is infeasible due to the lack of sufficient domain-specific data in Italian.

For the classification task, we employ established algorithms from the scikit-learn library [62], including Support Vector Machines (SVMs), K-Nearest Neighbor (k-NN), Random Forest classifier (RF), and Logistic Regression (LR) [63].

To assess the performance of these classifiers, we compute standard evaluation metrics such as Precision, Recall, and F1-score [63]. These metrics are critical for evaluating the system's effectiveness in correctly identifying positive instances for each class.

Additionally, cosine similarity is used to measure the similarity between pairs of legal documents within the vector space [63]. Given the encodings $e_i = f(d_i)$ and $e_j = f(d_j)$ for documents d_i and d_j, the similarity between the documents is defined as follows:

$$sim_{d_i,d_j} = sim(e_i, e_j) = \frac{e_i \cdot e_j}{\|e_i\| \|e_j\|} \tag{3.1}$$

3.2.3 Results

We calculate the pairwise similarity for each document pair, categorize the results by relationship type, and analyze the mean differences in similarity distributions for each relationship category. The objective is to assess whether the text encoding methods can effectively distinguish between different relationship types.

To test the hypothesis that relationship types are distinctly separable, we conduct a two-sided t-test [64] to compare the means, with a significance level set at $\alpha = 0.05$. The findings are summarized in Table 3.1. Consistent with expectations, the mean differences between *Parent-of/Child-of* and *Sibling-of* relationships are consistently not significant.

Among the text representation methods evaluated, Doc2Vec demonstrates the highest performance. Contextualized embedding models do not surpass other methods, primarily due to the insufficient domain-specific training data available.

Table 3.2 displays the performance metrics of the classifiers on the test set. The combination of Doc2Vec text representation and Logistic Regression classifier has shown exceptional effectiveness, which can be attributed to the specific characteristics of the input data.

Model explainability To better understand the classification problem, we employ decision tree models to assess the influence of input features on the output predictions.

Method	Type 1	Type 2	P-value
BERT	Sibling-of	Parent-of/Child-of	0.073
BERT	Related-to	Parent-of/Child-of	<0.001
BERT	Related-to	Sibling-of	<0.001
BERT	Other	Parent-of/Child-of	0.848
BERT	Other	Sibling-of	0.047
BERT	Other	Related-to	<0.001
Doc2Vec	Sibling-of	Parent-of/Child-of	0.308
Doc2Vec	Related-to	Parent-of/Child-of	<0.001
Doc2Vec	Related-to	Sibling-of	<0.001
Doc2Vec	Other	Parent-of/Child-of	<0.001
Doc2Vec	Other	Sibling-of	<0.001
Doc2Vec	Other	Related-to	<0.001
TFIDF	Sibling-of	Parent-of/Child-of	0.839
TFIDF	Related-to	Parent-of/Child-of	0.889
TFIDF	Related-to	Sibling-of	0.746
TFIDF	Other	Parent-of/Child-of	<0.001
TFIDF	Other	Sibling-of	<0.001
TFIDF	Other	Related-to	<0.001

Table 3.1 Significance test for the difference of two means computed on documents' similarities groups with a α=0.05.

Representation	Classifier	Precision	Recall	F1-Score
Doc2vec	Logistic Regression	0.740	0.744	0.740
Doc2vec	SVM	0.731	0.735	0.733
Doc2vec	Random Forest	0.721	0.724	0.722
TF-IDF	Random Forest	0.702	0.706	0.703
TF-IDF	SVM	0.690	0.695	0.685
TF-IDF	Logistic Regression	0.666	0.669	0.667
Doc2vec	KNN	0.660	0.664	0.662
BERT	Random Forest	0.650	0.660	0.648
BERT	SVM	0.639	0.647	0.642
BERT	Logistic Regression	0.610	0.620	0.614
TF-IDF	KNN	0.583	0.591	0.584
BERT	KNN	0.429	0.460	0.433

Table 3.2 Classification results

Figure 3.2 highlights that domain-specific headwords, such as "Patents" and "Trademarks" are highly discriminative. This finding is advantageous for modeling,

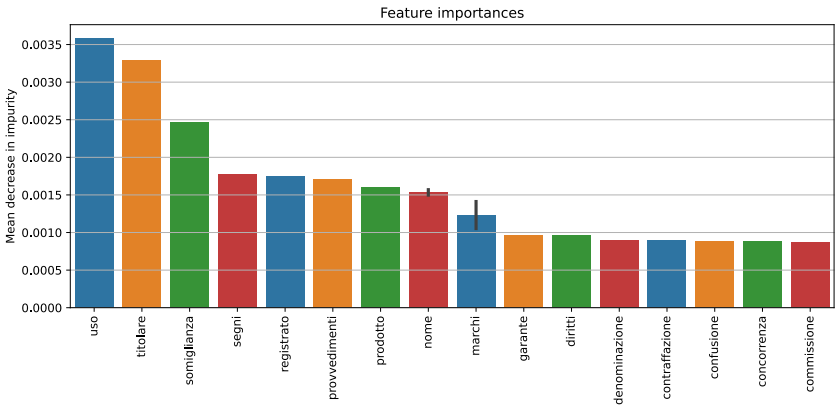


Fig. 3.2 Feature importance of Random forest classifier: most relevant terms for the classifier are mainly in-domain terms, such as “trademark” ("marchio"), “commission” (“commissioni”).

as it facilitates the early pruning of less relevant features, thereby enhancing the efficiency and accuracy of the classification process.

3.3 The importance of named entities

In [65], we address the limitations inherent in current transformer-based methods: they fail to differentiate between legal texts and general-purpose texts, thereby constraining their performance, particularly in zero-shot learning scenarios [47].

To mitigate these limitations, we propose an entity-aware attention mechanism built upon the LUKE transformer [66]. This mechanism utilizes entity embeddings to capture semantic features specific to the legal domain. By concentrating on textual dependencies involving entities, the proposed approach aims to enhance the discriminative capacity of the model in the context of legal document classification.

Our experiments using the EURLex dataset [54] reveal that incorporating entity embeddings substantially improves zero-shot extreme multi-class (XMC) performance. The proposed entity-aware attention mechanism not only surpasses current transformer-based methods but also shows superior efficacy compared to the larger Llama 2 7B model [67].

3.3.1 Methodology

In this section, we present our methodology for eXtreme Multi-label Classification of legal documents in a zero-shot setting, which implies the absence of specific training examples. To address this, we propose leveraging entity embeddings within the document text. Specifically, we utilize the pre-trained LUKE model [66], modifying its original classification layer to integrate one that has been trained on the benchmark dataset. LUKE, a pre-trained model that provides contextualized representations of words and entities based on a transformer architecture, employs an *entity-aware self-attention mechanism*. This mechanism enhances the precision of attention scores by incorporating entities directly into the representation process.

Let $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k$ be a sequence of input vectors, where $\mathbf{x}_i \in \mathbb{R}^D$, the attention score e_{ij} is computed as follows:

$$e_{ij} = \begin{cases} \mathbf{K}\mathbf{x}_j^\top \mathbf{Q}\mathbf{x}_i, & \text{if both } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are words} \\ \mathbf{K}\mathbf{x}_j^\top \mathbf{Q}_{w2e}\mathbf{x}_i, & \text{if } \mathbf{x}_i \text{ is word and } \mathbf{x}_j \text{ is entity} \\ \mathbf{K}\mathbf{x}_j^\top \mathbf{Q}_{e2w}\mathbf{x}_i, & \text{if } \mathbf{x}_i \text{ is entity and } \mathbf{x}_j \text{ is word} \\ \mathbf{K}\mathbf{x}_j^\top \mathbf{Q}_{e2e}\mathbf{x}_i, & \text{if both } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are entities} \end{cases} \tag{3.2}$$

where $\mathbf{Q}_{w2e}$, $\mathbf{Q}_{e2w}$, $\mathbf{Q}_{e2e} \in \mathbb{R}^{L\times D}$ are query matrices, $\mathbf{K} \in \mathbb{R}^{L\times D}$ is key matrix.

3.3.2 Experimental setup

Dataset In our experiments, we utilize the English subset of the EURLEX dataset[54], which is a multi-label classification dataset containing 65,000 European Union (EU) legal documents annotated with EUROVOC taxonomy labels. The EUROVOC taxonomy is a multilingual classification and thesaurus system that organizes and categorizes concepts and terms utilized in official EU documents, thereby facilitating research and information retrieval. Each document in the EURLEX dataset is annotated with one or more EUROVOC concepts.

Consistent with the methodology described in [54], we focus on the third-level taxonomy labels and employ the dataset splits provided by the authors for the training and testing phases of our models.

Models We evaluate our methodology against the following benchmarks:

- **Logistic Regression**: This baseline approach utilizes a Term Frequency-Inverse Document Frequency (TF-IDF) encoder to capture both local and global token frequencies. A logistic regression model is then trained on the encoded text to perform classification.
- **RoBERTa** [54]: This model is an enhancement of BERT [68] that excludes the next-sentence prediction objective and is trained with larger mini-batches and higher learning rates, improving performance on various tasks.
- **Llama 2 7B** [67]: A pre-trained large language model known for its strong performance in both few-shot and zero-shot learning scenarios. In line with [69], we approached the XMC task as a generative problem for comparative analysis with LLMs.

The performance of the models in our study is evaluated using the following metrics:

- **Precision@5** and **Recall@5**: These metrics assess the precision and recall at k predictions, where k is set to 5 in our dataset. This choice corresponds to the average number of labels per document in the training set. The formulas are defined as:

$$\text{Precision@5} = \frac{\text{TP}_5}{\text{TP}_5 + \text{FP}_5} \tag{3.3}$$

$$\text{Recall@5} = \frac{\text{TP}_5}{\text{TP}_5 + \text{FN}_5} \tag{3.4}$$

  where TP_5, FP_5, and FN_5 represent the true positives, false positives, and false negatives among the top 5 predictions, respectively.

- **mean Reciprocal Precision (mRP)**: This metric ranks the labels selected by the model in descending order of confidence for each document. Precision@k is computed where k equals the number of gold labels for the document, and the results are then averaged across all documents.

3.3.3 Experimental results

We conducted experiments with various training procedures to evaluate the performance of our proposed methodology and to compare it against different architectures. Initially, we fixed the 9 attention blocks and fine-tuned only the classification layer to assess the quality of the hidden representations. Subsequently, we performed a comprehensive end-to-end evaluation of the proposed model to fully explore its capabilities.

Table 3.3 presents the overall performance of our model under different training strategies. Our findings reveal that the proposed approach surpasses both the state-of-the-art model and the Large Language Model LLama 2. Notably, the model demonstrates superior performance even when the first 9 attention blocks are kept static, underscoring the effectiveness of our model in producing highly informative hidden representations that significantly enhance the classification task.

Models	**mRP**
Logistic Regression	0.21
State-of-the-art [54] (first 9 blocks frozen)	0.27
Our approach (first 9 blocks frozen)	**0.33**
State-of-the-art [54] (end-to-end training)	0.67
Our approach (end-to-end training)	**0.68**
LLama 2 7B [67]	0.65

Table 3.3 Models comparison

We performed a comparative analysis of our model against competing approaches in the context of zero-shot classification, where the labels are not present during training. In this analysis, all models were trained with no layers frozen.

Table 3.4 presents the results measured by Precision@5 and Recall@5, evaluating the models' capability to retrieve relevant labels without prior exposure to those labels.

The findings reveal that the baseline model exhibits poor performance in zero-shot scenarios, with low Precision@5 and Recall@5 scores. Although the state-of-the-art model demonstrates marginally improved performance, it still falls short in comparison to our proposed method. Our model achieves significantly higher

Precision@5 and Recall@5 scores, underscoring its superior performance in zero-shot learning contexts. These results attest to the accuracy and comprehensiveness of our model's predictions. Notably, while Large Language Models (LLMs) exhibit better Recall@5 scores, their overall performance is inferior.

	R@5	**P@5**
Logistic Regression	0.001	0.001
State-of-the-art [54]	0.028	0.006
Our approach	0.087	**0.164**
LLama 2 7B [67]	**0.253**	0.056

Table 3.4 Comparison in zero-shot learning context

Model explainability To further illustrate the effectiveness of the entity-aware self-attention mechanism, we analyze the attention scores derived from the top-performing models listed in Table 3.3. Specifically, we compute the mean token attention scores assigned by both the state-of-the-art model and the LUKE model, focusing on the final attention layer[1]. Tokens are ranked in descending order according to the attention scores provided by each model.

For each class $c \in C$, we calculate the Mean Reciprocal Rank (MRR) of model m_i using the most frequent k tokens associated with class c. The MRR is defined as:

$$\mathrm{MRR}_{m_i,c,k} = \mathrm{MRR}(\mathrm{R}_{a(m_i),k_c}) \tag{3.5}$$

In this formula, $\mathrm{R}_{a(m_i)}$ denotes the attention ranking position of the k most frequent tokens for class c in model m_i.

We then compute the Mean Reciprocal Rank difference between our model and the state-of-the-art model for various values of$k$:

$$\mathrm{MRR}_k = \frac{1}{|C|} \sum_{c \in C} (\mathrm{MRR}_{\mathrm{LUKE},c,k} - \mathrm{MRR}_{\mathrm{SOTA},c,k}) \tag{3.6}$$

where:

[1]The final attention head is considered as it is closest to the classification layer.

- $\text{MRR}_{\text{LUKE},c,k}$ denotes the Mean Reciprocal Rank computed using the LUKE model for class $c \in C$, considering the k most frequent terms.
- $\text{MRR}_{\text{SOTA},c,k}$ represents the Mean Reciprocal Rank computed using the state-of-the-art model for class $c \in C$, taking into account the k most frequent terms.

These values are illustrated in Figures 3.3 and 3.4, which present the results for frequent and zero-shot labels, respectively. Positive scores indicate that, on average, our model assigns more attention to the most frequent terms associated with each class compared to the state-of-the-art model. The results demonstrate that our model tends to emphasize terms that frequently appear within each class, with a notable advantage for zero-shot labels. However, the magnitude of these differences diminishes as k increases.

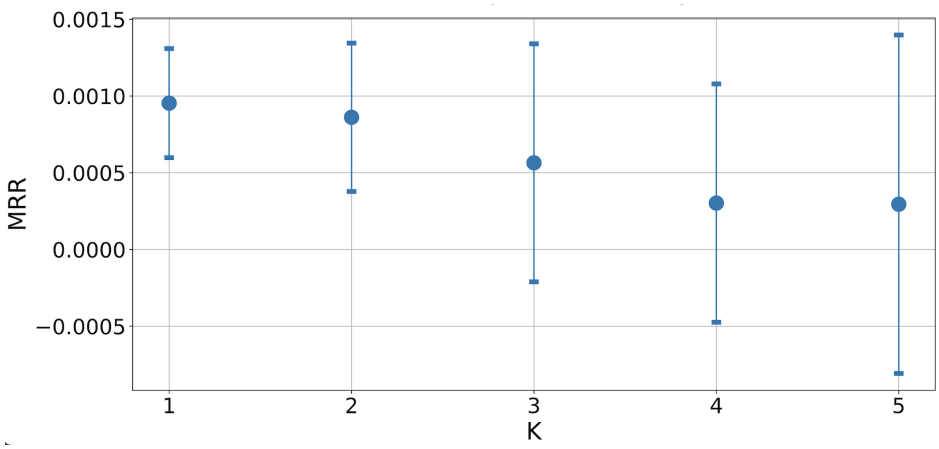


Fig. 3.3 Comparison of MRR differences in token attention scores between our proposed model and the state-of-the-art model for different values of k, specifically for frequent labels. Positive differences indicate that our model assigns more attention to the most frequent terms associated with each class.

3.4 Scaling to a real word scenario

In [70], we investigate the generalization capabilities of established pre-trained language models (LMs) across various types of legal documents. This study employs the BERT model [68] within a real-world business context focused on the classification of Italian legal documents. Our experiments utilize a proprietary dataset comprising thousands of documents, which are diverse in terms of data sources (e.g.,

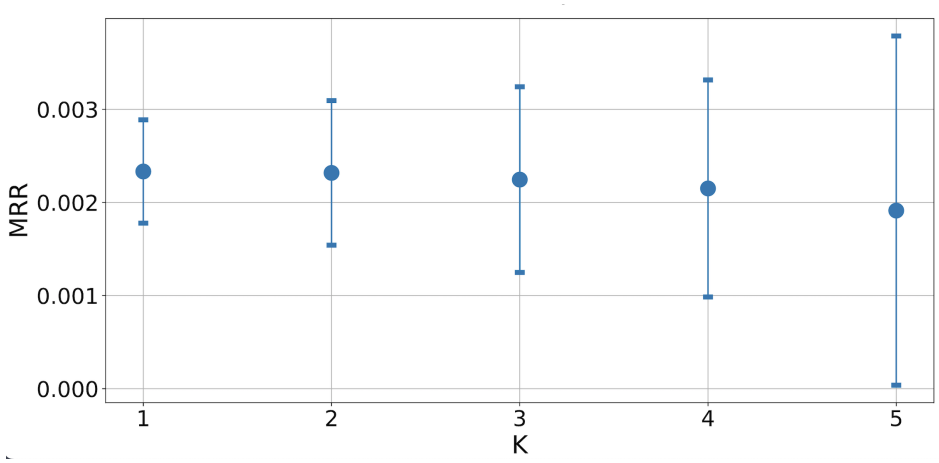


Fig. 3.4 Comparison of MRR differences in token attention scores between our proposed model and the state-of-the-art model for different values of k, specifically for zero-shot labels. Positive differences indicate that our model assigns more attention to the most frequent terms associated with each class.

legal judgments, maxims, and legal news) and legal domains. Each document is annotated with over 20000 distinct labels.

We extend a standard classification pipeline to address the multi-class problem and manage lengthy documents through hierarchical modeling and multi-label attention [71]. Our evaluation of PLM performance includes both quantitative and qualitative assessments. The results reveal that while the model performs well on maxims, its performance diminishes on longer documents.

3.4.1 Problem statement and business case

An Italian publisher specializing in legal texts and related products requires an efficient system for organizing various document types, including legal news, court judgments, contracts, and maxims. To facilitate effective retrieval and navigation of this database, documents must be annotated using labels from a proprietary taxonomy that represents an is-a hierarchy.

Previous approaches to Italian legal document classification [71] have been limited to single areas of law. Consequently, the extent to which pre-trained language models generalize across different legal domains and document types remains unclear. This paper addresses these limitations by investigating a broader spectrum of document types and legal areas.

We adhere to established best practices [72], including: (1) *Data Benchmarking* to enable thorough comparisons across a sufficiently large number of cases, and (2) *Error Analysis* to identify the limitations of Artificial Intelligence and determine where human intervention is necessary.

3.4.2 Methodology

To address the limitation of BERT model [68] we implemented the following modifications:

- **Type-specific LMs**: To assess the influence of document type on classification performance, we fine-tune distinct BERT-based multi-label classification models [68] for each legal domain. This approach allows us to evaluate the impact of specialization on the model's effectiveness across different areas of law.
- **Multi-class data**: We adapt the classification process to a multi-class setting by initially applying a minimum confidence threshold to each class probability. Specifically, we assign all classes to a test document if their predicted probabilities exceed this threshold. The model is trained to jointly classify labels across different hierarchical levels, which encourages the model to learn and predict at least one label from both the first and second hierarchical levels. While these labels may not be as detailed as third-level labels, they are essential for accurately identifying the correct sub-area of law.
- **Long documents**: We modify pre-trained language models (LMs) to process documents exceeding the standard token limit of 512 tokens. This adaptation involves employing a hierarchical model [73], which first divides each document into paragraphs. For each paragraph, an intermediate representation is generated using the hidden state of the Begin-of-Sequence token, referred to as the *paragraph attention vector*. These paragraph-level vectors are subsequently aggregated to form a comprehensive representation of the entire document, which is then input to a classification layer.
- **Multi-label attention**: To enhance the model's performance, we incorporate a multi-label attention mechanism [74]. This approach allows the model to assign varying weights to different segments of the input based on their

relevance to the predicted labels, thereby refining the model's accuracy by focusing attention on the most pertinent parts of the document.

Model's predictions are post-processed in the following manner: we remove redundant intermediate-level labels and prioritizing the most specific predictions (being hierarchical, the most specific prediction include also parent labels). For example, if the model outputs both the labels *Associations and Foundations* and *Associations and Foundations - Committees*, we retain only the more detailed label *Associations and Foundations - Committees* since it provides more precise information.

3.4.3 Experimental setting

Dataset The proprietary dataset comprises Italian legal judgments, maxims, and legal news across ten distinct areas of law:

- *Public Administration*: Documents addressing issues of public interest related to the organization and functions of public administration and its interactions with private individuals.
- *Lease and Housing*: Documents concerning rulings on contracts between private parties involving real estate assets.
- *Family*: Documents dealing with legal relationships among individuals defined by familial connections under the law.
- *Civil Liability*: Documents pertaining to offenses that contravene the civil code provisions.
- *Labour*: Documents addressing regulations governing relationships between employees and employers.
- *Civil and Telematic Process*: Documents related to norms governing procedural mechanisms, jurisdiction, and laws ensuring fair justice within civil judicial processes.
- *Criminal*: Documents related to facts punished with penalties or sanctions based on the severity of criminal offenses.

- *Corporate*: Documents related to regulations concerning the formation, governance, oversight, dissolution, and liquidation of companies. This includes corporate responsibility, shareholder property relations, extraordinary corporate transactions, and management of company crises.
- *Tax*: Documents pertaining to procedures and regulations related to taxation.
- *Bankruptcy*: Documents addressing the regulation of business insolvency and crisis management.

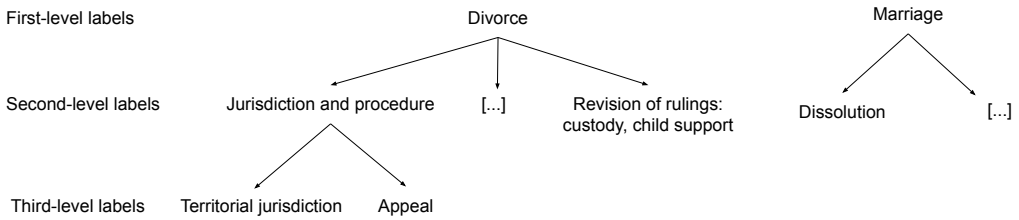


Fig. 3.5 Example of label taxonomy related to the *family* law area

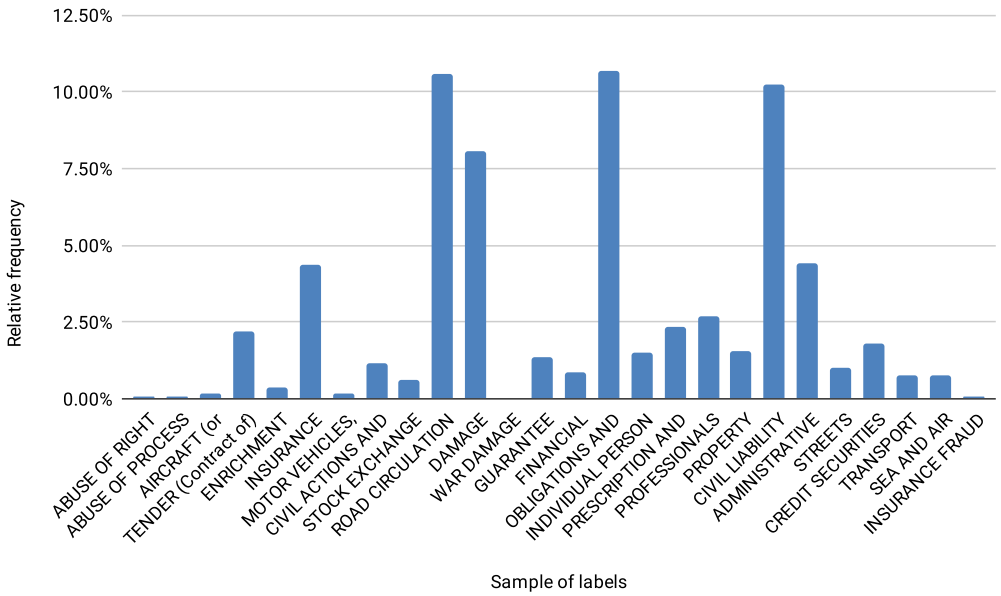


Fig. 3.6 Distribution of first-level labels over the documents (legal judgments and maxims) related to civil liability law area.

Each legal domain is mapped to a segment of a proprietary taxonomy, which is organized as an is-a hierarchy. This taxonomy structures the set of labels used to

denote pertinent legal principles. It is hierarchically organized into three layers of concepts, with each successive layer offering a more granular characterization of legal topics (refer to Figure 3.5). It is important to note that while some second-level labels are associated with third-level labels, all top-level labels are further specialized into second-level labels.

A portion of the label hierarchies is shared across multiple legal domains. On average, legal judgments and maxims are annotated with five labels per document, whereas legal news remains unannotated. Legal judgments are pre-processed by focusing exclusively on paragraphs from the *FactLaw* corpus, which includes sections detailing facts and applied rulings, while preambles and conclusions are excluded.

Table 3.5 provides a summary of dataset statistics, including the distribution of labels across various hierarchical levels within the taxonomy. Additionally, the table reports the average document lengths, measured in characters per document, which surpass those of established text classification datasets [75–78]. The number of labels per legal domain varies significantly, having an imbalanced class label distribution (see Figure 3.6).

Experimental setup We conducted a train-validation-test simulation for each law area individually. Following the approach outlined in [51], we maintained a fixed number of 15,000 test documents per area and allocated 5% of the data for the validation set.

For our experiments, we utilized the *bert_multi_cased* model[2], a multilingual version of BERT that includes Italian in its pretraining. This model comprises 12 transformer blocks, each with a hidden size of 768 and 12 attention heads. The models were trained using sequence cross-entropy loss and the AdamW optimizer [79] with a weight decay rate of 0.01. Training was performed over a small number of epochs (3) with a learning rate of 5×10^{-5}. During inference, a confidence threshold of 0.30 was applied. To mitigate overfitting, we employed dropout in the final classification layer with a probability of $p = 0.1$ and used early stopping during training.

[2]Available on TensorFlow Hub: https://tfhub.dev/

Dataset cardinality			
Law area	**Judgments**	**Maxims**	**Legal news**
Public administration	555749	280471	392
Lease and Housing	22192	35662	150
Family	34454	37800	502
Civil liability	111751	71842	470
Civil and telematic process	581480	421202	171
Criminal	377397	300252	431
Labour	279397	136639	244
Corporate	28703	9474	314
Tax	706343	459822	305
Bankruptcy	405553	290714	337
All	**3103019**	**2043878**	**3316**
Labels cardinality			
Law area	**First level classes**	**Second level classes**	**Third level classes**
Public administration	51	832	1641
Lease and Housing	13	163	259
Family	35	230	243
Civil liability	26	400	470
Civil and telematic process	126	1650	2569
Criminal	147	1301	1698
Labour	28	512	784
Corporate	24	129	233
Tax	106	1631	2608
Bankruptcy	66	1028	1762
All	**622**	**7876**	**12267**
Average documents length			
Law area	**Legal judgements**	**Maxims**	**Legal news**
Public administration	15174.47	646.11	1100.28
Lease and Housing	15572.34	598.09	5967.18
Family	9862.95	640.55	9044.72
Civil liability	15174.47	646.11	12084.64
Civil and telematic process	11606.35	631.87	9258.03
Criminal	11850.67	606.92	3994.46
Labour	13730.64	632.02	1547.79
Corporate	18115.98	557.3	13981.53
Tax	12266.11	594.09	11646.29
Bankruptcy	12289.62	630.08	10149.7
All	**13564.36**	**618.31**	**7877.46**

Table 3.5 Overview of the dataset statistics, including the count of documents, the number of associated labels, and the average document length across various law areas.

Hardware and execution times The experiments were performed on a single NVidia® Tesla® A100 GPU with 80 GB of memory. The training and inference execution times varied significantly depending on the specific law area, ranging from 1 hour to a full day.

Evaluation metrics We evaluate the performance of PLMs from both quantitative and qualitative perspectives. Quantitative analysis relies on labeled data (i.e., ground truth), which is available only for legal judgments and maxims. It counts the overlap between the predicted and ground truth labels per document with precision and recall and f1-score metrics [80]. Precision is defined as the proportion of correctly classified samples out of all samples classified as positive. The recall measures the proportion of correctly classified positive samples out of all gold standard positive samples. We calculate the weighted F1-score, which represents the mean of the F1-score values (the harmonic mean of precision and recall) across all input labels.

The qualitative assessment focuses on ensuring that the PLM outcomes align with the expectations of human experts. To complement the quantitative results, this qualitative evaluation is conducted on both an unlabeled legal news dataset (which is not included in the quantitative analysis) and the labeled legal judgments and maxims. In this process, a group of 10 expert evaluators is engaged to review the predicted document labels. They are asked to categorize the labels as *correct*, *partially correct*, or *incorrect*, thereby providing an additional layer of validation.

For each area of law, we calculate the percentage of legal documents that fall into the following categories:

- Documents where all predicted labels are correct, with no missing labels.
- Documents where predictions are partially correct. In these cases, the evaluator considers the classification incomplete or identifies that some of the labels returned by the model are incorrect.
- Documents where all predictions are incorrect, meaning none of the labels provided by the model are deemed acceptable.

3.4.4 Results

Comparison between data sources

Table 3.6 presents the weighted f1-score along with the corresponding confidence intervals for each law area, evaluated separately for different data sources (legal judgments or maxims) and their combination (*all*). The f1-score values, which range from 0.48 to 0.76, suggest that the type of legal source significantly influences PLM performance. The variability in results is fairly consistent across all law areas. Notably, maxims, while less numerous, are easier to classify. These concise statements, which encapsulate general legal principles, are frequently used in legal reasoning and decision-making. The relative simplicity of their linguistic structure likely contributes to the higher performance of PLMs in these cases.

In contrast, classification performance on judgments is generally lower across all law areas. Judgments are comprehensive written decisions that detail the facts of a case, present legal arguments, and conclude with a ruling. These documents often include extraneous information not pertinent to the classification task and may feature domain-specific terms that are less common in other types of legal documents. Law areas such as *Family*, *Civil Liability*, *Criminal*, *Labour*, and *Corporate* exhibit relatively higher standard deviations in performance, suggesting that PLM performance in these domains is more influenced by specific cases or variations within the texts.

In contrast, other law areas like *Public Administration* and *Lease and Housing* demonstrate more consistent performance across different documents (lower standard deviations). This consistency suggests that the PLM performs more reliably in these areas. The higher standard deviation observed in *Maxims* compared to legal judgments may be due to the greater variability in the wording, structure, and complexity of maxims. Unlike the more standardized and structured legal judgments, maxims can differ significantly in their language and content, which could lead to increased variability in performance scores.

Comparison between areas of law

The performance of PLMs, as measured by the f1-score, shows no significant correlation with the number of labels in each law area, as indicated by the Pearson

correlation coefficient. This suggests that simply considering the area of law without accounting for the document types is generally insufficient for achieving satisfactory classification results.

Law area	Legal Judgments	Maxims	All
Public administration	0.53 ($\pm$ 0.16)	0.65 ($\pm$ 0.31)	0.61 ($\pm$ 0.28)
Lease and Housing	0.59 ($\pm$ 0.21)	0.68 ($\pm$ 0.31)	0.64 ($\pm$ 0.28)
Family	0.76 ($\pm$ 0.25)	0.68 ($\pm$ 0.32)	0.72 ($\pm$ 0.32)
Civil liability	0.55 ($\pm$ 0.20)	0.68 ($\pm$ 0.36)	0.63 ($\pm$ 0.33)
Civil and telematic process	0.53 ($\pm$ 0.15)	0.67 ($\pm$ 0.32)	0.61 ($\pm$ 0.28)
Criminal	0.60 ($\pm$ 0.20)	0.64 ($\pm$ 0.32)	0.63 ($\pm$ 0.32)
Labour	0.60 ($\pm$ 0.18)	0.64 ($\pm$ 0.30)	0.63 ($\pm$ 0.28)
Corporate	0.48 ($\pm$ 0.14)	0.65 ($\pm$ 0.32)	0.62 ($\pm$ 0.30)
Tax	0.51 ($\pm$ 0.16)	0.62 ($\pm$ 0.30)	0.59 ($\pm$ 0.28)
Bankruptcy	0.56 ($\pm$ 0.16)	0.63 ($\pm$ 0.31)	0.61 ($\pm$ 0.27)
All	**0.57 ($\pm$ 0.19)**	**0.654 ($\pm$ 0.31)**	

Table 3.6 PLM performance on different law areas in terms of weighted f1-score

Comparing between levels of label granularity

Table 3.7 presents the performance of PLMs across different levels of granularity within the expert-provided taxonomy. The models demonstrate a conservative prediction pattern, assigning a higher proportion of first- and second-level labels compared to third-level labels. On average, each third-level label is associated with two first- and second-level labels.

First-level label predictions generally achieve high f1-scores ($\geq 80\%$), with recall values marginally surpassing precision. However, there is a notable decline in performance for second- and third-level label classifications, with f1-scores decreasing by approximately 30% from level-1 to level-3. This drop can be attributed to the increased complexity of multi-class classification at higher levels, where the number of labels is significantly greater.

Furthermore, as detailed in Section 3.4.2, the label trees are expanded, and the model is trained to classify all labels simultaneously. Consequently, labels from higher levels are more frequently considered during training, which may contribute to the observed performance disparities.

Law area	Level 1 labels			Level 2 labels			Level 3 labels		
	P	R	F1	P	R	F1	P	R	F1
Public administration	0.82	0.84	0.83	0.60	0.57	0.56	0.48	0.39	0.40
Lease and Housing	0.93	0.92	0.92	0.65	0.52	0.57	0.51	0.33	0.38
Bankruptcy	0.77	0.84	0.80	0.57	0.57	0.55	0.46	0.43	0.42
Family	0.86	0.90	0.88	0.69	0.72	0.70	0.45	0.44	0.43
Labour	0.83	0.86	0.85	0.58	0.59	0.57	0.46	0.42	0.41
Civil and telematic process	0.76	0.83	0.79	0.57	0.57	0.55	0.46	0.43	0.42
Corporate	0.82	0.86	0.84	0.57	0.60	0.57	0.45	0.39	0.39
Tax	0.74	0.82	0.77	0.57	0.55	0.54	0.45	0.40	0.40
Civil liability	0.85	0.81	0.83	0.65	0.52	0.56	0.57	0.41	0.45
Criminal	0.78	0.82	0.80	0.57	0.60	0.57	0.49	0.46	0.45
All	**0.81**	**0.86**	**0.83**	**0.60**	**0.58**	**0.57**	**0.48**	**0.40**	**0.42**

Table 3.7 Performance of PLMs (Precision, Recall, and F1-score) across various law areas and taxonomy levels of granularity

Qualitatitive evaluation

Table 3.8 provides a summary of the qualitative assessment results, presenting the accuracy (percentage of correct, partially correct, and incorrect predictions). The annotators achieved approximately 70% agreement, meaning that in 70% of the cases, they concurred on whether a classification was entirely correct, partially correct, or completely incorrect. The results for legal judgments and maxims align closely with the quantitative findings, with about 80% of assignments being correct. Additionally, approximately 10% of the predictions were deemed partially correct, indicating that the PLMs' predictions are generally reliable.

However, the areas of *Corporate* and *Tax* stand out with a notable number of incorrect predictions, likely due to the specialized terminology used in these domains. In contrast, for legal news, the performance of the PLMs declined, particularly in the areas of *Public Administration* and *Criminal Law*. Despite this, in 8 out of 10 law areas, the number of incorrect predictions remained stable, suggesting that most automatic assignments were either correct or partially correct.

Furthermore, the number of predictions generated by the model for legal news decreased to an average of three per document. Annotators noted that the most common cause of errors was the lack of sufficiently detailed predictions. In summary, while PLMs are effective in correctly identifying the sub-areas of law represented by first-level labels, achieving a higher level of detail would require some form of domain adaptation, especially when dealing with unseen data sources.

	Legal judgement/maxims			Legal news		
Law area	**C**	**PC**	**I**	**C**	**PC**	**I**
Public administration	80%	14%	6%	49%	44%	7%
Lease and Housing	78%	12%	10%	77%	15%	8%
Family	81%	16%	3%	75%	20%	5%
Civil liability	83%	9%	8%	78%	13%	9%
Civil and Telematic process	79%	14%	7%	80%	18%	2%
Criminal	83%	12%	5%	59%	33%	8%
Labour	82%	13%	5%	77%	16%	7%
Corporate	79%	13%	8%	60%	12%	28%
Tax	82%	12%	6%	67%	18%	15%
Bankruptcy	81%	13%	6%	71%	24%	5%

Table 3.8 Qualitative validation results in terms of Correct (C), Parial Correct (PC), Incorrect (I) percentage.

Errors and models limitations

In this section, we summarize the main errors and limitations identified in the models:

- **Performance Variation**: The performance of LMs varies depending on the type of legal source. Maxims, which are concise and express general truths about the law, tend to be easier to classify, leading to higher average performance scores but with greater variability. On the other side, judgments, which contain more detailed information, generally show lower more stability in classification performance across all law areas.
- **Law Area vs. Document Types**: the LMs' performance, as measured by the F1-score, does not correlate with the number of labels in each law area. Therefore, it is crucial to take both the area of law and the document types into account to improve classification outcomes.
- **Label Granularity**: LMs tend to be conservative in assigning labels. An higher proportion of first- and second-level label is predicted, compared to third-level labels. The F1-score significantly decreases from level-1 to level-3. This decline is largely due to the greater complexity of multi-class classification with a larger number of labels at higher levels.

- **Performance on Legal News**: The performance of LMs decreases notably on legal news, especially in areas like Public Administration and Criminal Law. Despite this, the number of incorrect predictions is stable across most law areas, suggesting that while LMs are generally reliable, they may require domain adaptation when dealing with unseen data sources.

3.5 Summary of results and key insights

Several key findings emerge regarding automated legal document classification using natural language processing techniques. The research demonstrates that classification approach effectiveness varies significantly depending on the specific task and data characteristics: Doc2Vec with Logistic Regression achieved the highest performance (F1-score of 0.740) for taxonomy relationship extraction in Italian legal texts, surprisingly outperforming contextualized BERT embeddings due to insufficient domain-specific training data. In contrast, entity-aware attention mechanisms through LUKE-based models substantially enhanced zero-shot learning capabilities, outperforming even larger models like Llama 2 7B. However, comprehensive BERT-based classification models revealed significant performance degradation with increased taxonomic complexity, dropping approximately 30% from first-level to third-level classifications, and notable variations across document types, with maxims consistently outperforming complex legal judgments. The experiments collectively conclude that successful legal document classification requires domain-specific adaptations rather than relying solely on generic pre-trained models, with simpler embedding methods sometimes proving more effective than complex architectures in specialized legal corpora, particularly for low-resource languages. Furthermore, the incorporation of legal entities and attention mechanisms shows promise for zero-shot scenarios, while hierarchical modeling approaches become essential when dealing with fine-grained legal taxonomies, emphasizing that document type and domain complexity are crucial factors in determining optimal classification strategies.

Chapter 4

Content accessibility of legal documents

This chapter provides a review of the related literature in Section 4.1 and introduces the research contributions for improving text summarization in the legal domain. It provides the details of methodologies for benchmarking (Section 4.2), and improving (Section 4.3) Language Models.

4.1 Prior works

Legal documents, including legal news, statutes, patents, and judicial decisions, are frequently distinguished by their extensive length and intricate structure. Additionally, legal text tends to exhibit considerable redundancy and verbosity. These characteristics present significant challenges in terms of content accessibility, as navigating such documents can be exceedingly time-consuming, even for those with legal expertise.

Automated text summarization techniques utilize Deep Learning and Natural Language Processing (NLP) methods to enhance the accessibility and exploration of legal documents [81]. For instance, legal information providers frequently release news articles, some of which contain overlapping content, such as the repetition of the same facts by different news sources or the reference to previous events in more

recent articles. Summarizing these legal news articles enables legal professionals to efficiently gain insights into the content, thereby conserving both time and resources.

The problem of legal document summarization has recently attracted significant interest within the Legal AI community. For instance, studies such as [82–85] examine the effectiveness of summarization algorithms when applied to legal cases and court judgments. The majority of existing methodologies concentrate on summarizing legal documents written in English [81], with comparatively fewer efforts directed toward summarizing legal texts in other languages.

Summarization of English legal documents In [81] a comprehensive survey of legal document summarization techniques is presented, covering a range of summarization methods tailored to various types of legal documents, including court judgments [82] and legal case documents [85]. [82] compares multiple unsupervised and supervised summarization algorithms using a large corpus of Indian Supreme Court judgments. In [85], the authors explore the effectiveness of extractive and abstractive summarization techniques on legal case documents, conducting extensive experiments on three legal summarization datasets and providing insights into long-document summarization, particularly for English court rulings. The study presented in [86] emphasizes capturing the argumentative structure of legal documents by incorporating argument role labeling into the summarization process. [83] introduces CaseSummarizer, a summarization engine designed for English-written legal cases. The authors of [84]developed a novel train-attribute-mask pipeline using a CNN classifier for summarizing legal cases.

Legal Document Summarization using Large Language Models There have been limited attempts to apply Large Language Models (LLMs) to the task of legal document summarization, as noted in the literature [87, 88]; however, none of these efforts have focused on Italian legal documents. In particular, [87] tested GPT-3 [16] on the summarization of Indian court judgments, revealing issues such as inconsistencies and hallucinated information in the generated summaries. Their findings suggest that neither pre-trained abstractive summarizers nor LLMs are currently suitable for summarizing real case judgments. Our study complements the work of [87] by focusing on the summarization of Italian legal news, rather than English court judgments. Meanwhile, ChatLaw [88] is a domain-specific language

model developed for the Chinese legal domain. To address the challenge of model hallucinations in legal data retrieval, [88] propose combining vector-based retrieval with keyword-based methods.

Legal AI from Italian documents Limited works focuses on the italian legal landscape. ITALIAN-LEGAL-BERT [89] is the first pre-trained Transformer language model specifically designed for the Italian legal domain. It has undergone additional pre-training on Italian civil law corpora, leading to enhanced performance across various domain-specific tasks when compared to the general-purpose Italian BERT model. Similarly, LamBERTa [90] is another BERT-based architecture developed for law article retrieval, with training specifically focused on the Italian civil code. However, neither ITALIAN-LEGAL-BERT nor LamBERTa is intended for legal document summarization.

EUR-Lex-Sum [91] is a multilingual and cross-lingual dataset aimed at long document summarization within the legal domain. The dataset comprises legal acts from the European Union law platform (EUR-Lex), each paired with manually curated summaries, providing up to 1,500 document/summary pairs per language and 375 cross-lingually aligned legal acts. The paper evaluating this dataset assesses the performance of baseline extractive summarization methods.

In contrast to [91], which focuses on extractive summarization approaches, our work explores the application of abstractive techniques. Additionally, while EUR-Lex-Sum centers on the summarization of legal acts, our research is dedicated to summarizing legal news.

4.2 Benchmarking italian summarization models for the legal domain

Although several Large Language Models (LLMs) support multilingual content, including Italian, the proportion of training examples in languages other than English is typically low. For instance, only 0.11% of the data used to train Llama 2 [67] consists of Italian text. Consequently, the effectiveness of state-of-the-art summarization techniques on Italian legal documents remains uncertain. In [92, 93], we address the challenge of generating headlines and abstracts for Italian legal news

	Original Italian news content	**English translation**
Headline	Sostenibilità: disponibili le traduzioni degli standard emessi da EFRAG	Sustainability: translations of the standards issued by EFRAG are available
Abstract	Il CNDCEC ha pubblicato le traduzioni delle regole a cui le imprese devono conformarsi nel rendicontare impatti, rischi e opportunità legati alla sostenibilità in base alla Corporate Sustainability Reporting Directive.	The CNDCEC has published the translations of the rules that companies must comply with when reporting impacts, risks and opportunities related to sustainability based on the Corporate Sustainability Reporting Directive.
Main body	Il CNDCEC ha pubblicato le traduzioni di cinque degli standard ESRS (European Sustainability Reporting Standards) elaborati da EFRAG (European Financial Reporting Advisory Group) su incarico della Commissione europea che definiscono le regole a cui le imprese sono tenute a conformarsi nel rendicontare impatti, rischi e opportunità legati alla sostenibilità, secondo quanto previsto dalla CSRD, Corporate Sustainability Reporting Directive (Dir. 2022/2464/EU). Più in particolare, gli standard tradotti dal Consiglio sono i due standard cross cutting: - ESRS 1 General Requirements, - ESRS 2 General Disclosures, e i primi tre standard topic e sector agnostic sull'ambiente: [...]	The CNDCEC has published the translations of five of the ESRS (European Sustainability Reporting Standards) developed by EFRAG (European Financial Reporting Advisory Group) on behalf of the European Commission which define the rules with which companies are required to comply in reporting impacts, risks and opportunities related to sustainability, in accordance with the provisions of the CSRD, Corporate Sustainability Reporting Directive (Dir. 2022/2464/EU). More specifically, the standards translated by the Council are the two cross cutting standards: - ESRS 1 General Requirements, - ESRS 2 General Disclosures, and the first three standard topics and sector agnostic on the environment: [...]

Table 4.1 Example of Italian legal news and corresponding headline and abstract.

documents. Consider, for example, the article illustrated in Table 4.1. For clarity, both the original Italian content and its English translation are provided. The news article pertains to a specific legal domain (*Accounting*) and includes a headline and an abstract. These elements convey the most salient information in different textual forms: the headline encapsulates the main news points as key phrases, while the abstract offers a concise reformulation of the news, omitting marginal or redundant details.

Our objective is to automatically generate the headline and abstract (in Italian) by summarizing the original content of the news article. Both the headline and abstract typically consist of brief text segments that emphasize the key aspects of the news. We consider abstractive summarization methods to be more suitable than extractive ones, as they rephrase the original text rather than merely extracting portions of it.

4.2.1 Methodology

We conducted experiments using the following open-source Large Language Models (LLMs) that support Italian-language documents:

- **LLaMA 2** [67] is a collection of LLMs comprising both pretrained and fine-tuned models with parameter counts ranging from 7 billion to 70 billion. These models are primarily optimized for dialogue-based use cases. LLaMA 2 builds upon the architecture and pretraining settings of its predecessor, LLaMA 1, and demonstrates superior performance compared to LLaMA 1 and other competing models across various tasks, including commonsense reasoning, word knowledge, and reading comprehension.
- **Flan-T5** [94] is an enhanced version of the T5 model [95], which has been fine-tuned on numerous tasks, including text summarization. Flan-T5 exhibits improved zero-shot and few-shot capabilities relative to earlier models. The authors of Flan-T5 demonstrate that instruction fine-tuning can significantly enhance performance across a diverse range of models, prompting configurations, and evaluation tasks.

We initially explore the application of **Few-Shot In-Context Learning** (**ICL**) [96], a technique that allows pretrained LLMs to perform tasks that they have not previously encountered. This is achieved without any gradient-based training by providing

a small number of training examples directly as part of the input. However, the number of examples that can be included is constrained by the model's maximum context size. Additionally, because these input text-summary pairs must be included every time the model generates a prediction, the approach is considered computationally inefficient.

To address the limitations of Few-Shot In-Context Learning, we employ **Parameter-Efficient Fine-Tuning** (**PEFT**) [26], which enables the adaptation of pretrained models to specific tasks with minimal computational overhead. PEFT offers several advantages, including reduced memory usage, the ability to handle mixed-task batches during inference, and the fine-tuning of a small subset of additional parameters while keeping the majority of the pretrained model parameters frozen. Specifically, we utilize LoRA [27], a lightweight fine-tuning approach that involves freezing the weights of the pretrained model and introducing trainable rank decomposition matrices into each layer of the Transformer architecture. This method significantly decreases the number of parameters that need to be updated for downstream tasks. Additionally, to further minimize memory requirements, we apply 8-bit quantization [97] during the training of the LLM.

For comparative analysis, we evaluate the performance of the following established language models in the context of abstractive summarization:

- **BART-It** [98] is an adaptation of BART [99] for the Italian language. BART employs a denoising autoencoder during pretraining to reconstruct corrupted text, which enhances its ability to understand linguistic structures and improves performance in downstream tasks such as text summarization.
- **IT5** [100] is the Italian variant of T5 [95]. IT5 utilizes a unified text-to-text framework, which supports transfer learning and enhances the model's adaptability across various tasks.
- **mBART** [101] is a multilingual sequence-to-sequence model pretrained using the BART objective across a diverse set of multilingual corpora, allowing it to handle multiple languages effectively.
- **LSG-BART-It** [98] is a variant of BART-It that integrates Sparse Global Attention [102] in place of the traditional Self-Attention [103]. This modification aims to address the limitations of Transformers in processing long legal documents by approximating self-attention with an efficient $O(n)$ approach, where

n represents the number of input tokens, and is capable of handling documents up to 16,384 tokens in length.

Baseline methods We also evaluate the following extractive summarization methods as baseline approaches. These methods generate summaries by selecting relevant segments from the original news content without performing any additional text paraphrasing:

- **KL-Sum** [104] seeks to minimize the Kullback-Leibler divergence between the word distributions in the sentences and the entire news document. The core principle is to identify sentences that best represent the overall content of the news, thereby reducing the distance between the original document and its summary.
- **TextRank** [105] is a graph-based method that evaluates the importance of sentences based on their connectivity within a textual graph.
- **LexRank** [106] is another graph-based technique that assigns importance scores to sentences through random walks, based on their mutual similarity.
- **LSA** [107] employs Latent Semantic Analysis to select sentences that contain the most words relevant to a topic-level description of the document.
- **BERT Extractive** [108, 109] utilizes BERT [110] for extractive summarization. This model first generates sentence-level embeddings and then clusters them to identify the most representative sentences. For these experiments, we use the Italian legal adaptation of BERT [90].

4.2.2 Experimental setup

Dataset Table 4.2 provides an overview of the key characteristics of the legal news articles included in the proprietary dataset utilized in this study. This dataset encompasses the following areas of law: *Accounting, Finance, Tax, Business, Employment*, and *Digital Law*. The number of news articles varies considerably across these areas, with a minimal count for *Digital Law* and up to 1709 articles for *Tax Law*.

Table 4.3 offers insights into the reference summaries used in this study, which are categorized as either *headline* or *abstract*. Although the number of input tokens

Law area	Number of legal news
Accounting	112
Finance	234
Tax	1709
Business	265
Job	916
Digital	60

Table 4.2 News distribution across law areas.

	Abstract	Headline
Input length (tokens)	1,087.36	1,087.36
Output length (tokens)	61.879	15.639
% novel unigram	14.286	11.111
% novel bigram	42.222	37.500
% novel trigram	57.778	57.143
% novel fourgram	68.182	66.667
Coverage	0.317	0.086
Density	0.833	0.129
Compression ratio	0.982	0.984

Table 4.3 Characteristics of the proprietary dataset.

remains consistent regardless of the summary type, there are notable differences in output length, as headlines are considerably shorter than abstracts.

Abstractive summaries frequently introduce new terms not present in the original news documents. To assess the relevance of these summaries for the given use case, we conduct a preliminary analysis of n-gram co-occurrence frequencies[1] in both the original news content and the target summary. The analysis reveals that abstracts contain a higher percentage of original n-grams compared to headlines; however, in both cases, the original content is substantially rephrased. To corroborate these findings, we also calculate established metrics such as extractive fragment coverage, density, and compression ratio [111, 112].

[1]An n-gram refers to a sequence of *n* tokens in a row. For our experiments, *n* ranges from 1 to 4.

The *extractive fragment coverage* metric quantifies the degree to which the summary retains elements from the original content:

$$\text{Coverage}(A,S) = \frac{1}{|S|} \sum_{f \in \mathscr{F}(A,S)} |f| \tag{4.1}$$

where $A = \langle a_1, a_2, ..., a_n \rangle$ is the input sequence of tokens, S is the corresponding summary $S = \langle s_1, s_2, ..., s_m \rangle$ of length $|S|$, and $\mathscr{F}(A,S)$ is the set of token n-grams in common between A and S.

The *density* metric represents the average length of the extractive fragments, calculated as follows:

$$\text{Density}(A,S) = \frac{1}{|S|} \sum_{f \in \mathscr{F}(A,S)} |f|^2 \tag{4.2}$$

The *compression ratio* is defined as the ratio of the number of tokens in the input text to the number of tokens in the output summary:

$$\text{Compression}(A,S) = \frac{|A|}{|S|} \tag{4.3}$$

It indicates the reduction in token count from the input to the output. Generally, abstracts exhibit higher values for both coverage and density compared to headlines.

Experimental settings The training set comprises 70% of the proprietary dataset, 10% of this data is utilized as a validation set to identify the optimal model checkpoint and hyperparameters. The remaining 20% of the dataset serves as the test set. Stratification based on law areas is applied when generating these splits to ensure balanced representation across different legal domains. We optimize the model hyperparameters based on validation set performance. Specifically, we evaluate the following learning rates: 5×10^{-5}, 1×10^{-5}, and 1×10^{-6}. All models are trained for a maximum of 10 epochs, with the objective of maximizing log-likelihood using sequence cross-entropy loss and the AdamW optimizer [113], employing a weight decay of 0.01. The batch size is adjusted according to the model size and available hardware resources, ranging from 8 to 32. For fine-tuning Llama 2, we employ LoRA fine-tuning for a single epoch, setting the rank to $R = 64$ and the LoRA scaling factor

to $\alpha = 16$. This process begins with the base version of the Llama 2 model. Due to constraints on computational resources, we restrict the model size to 7 billion parameters. Under these conditions, the memory requirements for training Llama 2 are comparable to those of BART-It and IT-5.

Quantitative evaluation metrics We utilize the well-established ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric [114] to measure the syntactic similarity between human-generated and model-produced summaries. Specifically, we evaluate the following:

- **ROUGE-N**: This metric assesses the overlap of token n-grams between the reference and generated summaries. The score is computed as the ratio of the number of shared n-grams to the total number of n-grams in the summary.
- **ROUGE-L**: This metric evaluates the Longest Common Subsequence (LCS) of tokens between the reference and generated summaries, emphasizing the matching of longer sequences.

In addition, we employ BERTScore [115] to evaluate the semantic coherence of the generated summaries. BERTScore utilizes contextual embeddings [110] to compare generated and reference summaries within the latent space.

Qualitative evaluation metrics We conduct a human evaluation of the automatically generated summaries. Following the methodology outlined in [116], we ask human experts to assess the summaries based on the following established metrics:

- **Grammaticality** (GR): This metric evaluates the linguistic correctness and fluency of the generated text.
- **Summary Usefulness** (SU): This metric measures the extent to which the generated summary is informative and valuable relative to the original content.
- **Summary Coherence** (SC): This metric assesses the logical and semantic consistency of the generated summary.
- **Non-Redundancy** (NR): This metric evaluates the presence or absence of repetitive or redundant linguistic elements within the generated text.

Evaluation criterion	Questions
Grammaticality	How grammatically correct is the generated text? How fluent and natural is the language used in the generated text?
Summary Usefulness	How useful or valuable is the generated summary compared to the original content? Does the summary effectively capture the key points of the original content?
Summary Coherence	How logically coherent and semantically consistent is the generated summary? Does the summary flow in a logical and organized manner?
Non-Redundancy	How much redundant or repetitive information is present in the generated text? Does the generated text avoid unnecessary repetition of ideas or phrases?
Overall Quality	How would you rate the overall quality of the generated summary?

Table 4.4 Evaluation criteria for assessing text generation quality.

- **Overall Quality** (OQ): This metric provides a comprehensive evaluation of the overall quality of the generated summary.

Expert evaluators rate the quality of the summaries using a 5-point Mean Opinion Score (MOS) scale. Table 4.4 presents the questions used to guide the human validation process.

4.2.3 Results

Abstractive summarization Table 4.5 provides a summary of the performance of various models on headline and abstract generation across all quantitative metrics.

For abstract generation, LSG-BART-It demonstrates superior performance in syntactic metrics, such as ROUGE-1, ROUGE-2, and ROUGE-L, while LLama 2 (7B) excels in semantic similarity with the reference summary, as indicated by

BERTScore. Notably, Flan-T5-XL consistently underperforms relative to traditional models.

In headline generation, LLama 2 (7B) achieves the highest scores across all performance metrics. Traditional language models, including IT5-large and BART-It, exhibit comparable performance in this task. Although LSG-BART-It excels in abstract summarization, its performance in headline generation is closely matched by IT5-large and BART-It. Conversely, Flan-T5-XL significantly lags behind other models, particularly in the headline summarization task.

Summary	Model	Size	R-1	R-2	R-L	BERT-score
Abstract	IT5-small	60M	0.368	0.136	0.335	0.559
	IT5-base	220M	0.390	0.159	0.358	0.572
	IT5-large	738M	0.400	0.166	0.368	0.573
	BART-It	140M	0.441	0.231	0.411	0.625
	mBART-large	880M	0.433	0.221	0.395	0.633
	LSG-BART-It	140M	**0.456**	**0.249**	**0.427**	0.631
	Flan-T5-XL	3B	0.113	0.062	0.107	0.557
	LLama 2	7B	0.393	0.216	0.295	**0.746**
Headline	IT5-small	60M	0.245	0.003	0.232	0.541
	IT5-base	220M	0.270	0.021	0.258	0.550
	IT5-large	738M	0.296	0.034	0.282	0.555
	BART-It	140M	0.292	0.047	0.285	0.595
	mBART-large	880M	0.265	0.035	0.256	0.591
	LSG-BART-It	140M	0.298	0.051	0.290	0.597
	Flan-T5-XL	3B	0.053	0.048	0.051	0.542
	LLama 2	7B	**0.317**	**0.138**	**0.271**	**0.743**

Table 4.5 Quantitative evaluation of abstractive summarizers.

Insights into the performance of Large Language Models. We conduct an in-depth analysis of the performance variability of the top-performing open-source LLM, Llama 2 7B, across different legal domains in our proprietary dataset (see Table 4.6).

For abstract generation, the LLM's performance appears to be influenced by the specific legal area under consideration, likely reflecting the degree of domain-specific knowledge encoded in the pre-trained model. The model demonstrates strong

performance in certain domains (e.g., *Finance*, *Tax*), while others (e.g., *Accounting*, *Employment*) present greater challenges. A similar pattern emerges in the headline generation task, with the best performance observed in the *Digital* and *Accounting* domains, and the weakest in the *Tax* and *Employment* areas.

Target	Law area	R-1	R-2	R-L	BERT-score
Abstract	Accounting	0.367	0.209	0.277	0.738
	Finance	0.404	0.217	0.293	0.741
	Tax	0.398	0.227	0.301	0.748
	Corporate	0.397	0.215	0.303	0.744
	Job	0.385	0.200	0.284	0.747
	Digital	0.376	0.216	0.297	0.745
Headline	Accounting	0.348	0.152	0.279	0.741
	Finance	0.308	0.128	0.260	0.737
	Tax	0.290	0.128	0.242	0.724
	Corporate	0.326	0.149	0.281	0.737
	Job	0.288	0.115	0.240	0.729
	Digital	0.339	0.161	0.296	0.747

Table 4.6 LLama 2 performance across law areas.

Few-shot Learning and Lightweight Fine-Tuning We evaluate the performance of Large Language Models (LLMs) under the following configurations:

- **Zero-Shot Learning**: The LLM generates summaries without any additional training examples.
- **Few-Shot Learning**: We fine-tune the LLM using a small number of examples (N), which range from 5 to 1000, selected from the in-domain training dataset. This is achieved using LoRA for efficient training.
- **Full Training**: We fine-tune the LLM with the entire set of available in-domain training examples, exceeding 3300 samples, employing LoRA for efficient training.

For each setting, we ensure stratification across legal domains to appropriately select additional training examples when feasible.

The results are presented in Figures 4.2 and 4.1. Generally, both abstract and headline generation performance improves as the number of training examples increases. Specifically, performance metrics show a positive trend with the number of few-shot training examples. The *full training* configuration consistently achieves the highest scores across all performance metrics, indicating that a comprehensive training approach significantly enhances the accuracy of generated abstracts. Notably, BERTScore improvements are most pronounced with approximately 500-1000 training examples, with additional examples contributing only marginal gains. These findings suggest that the benefits of LLM fine-tuning approach a plateau, likely due to the sufficiency and diversity of the training dataset.

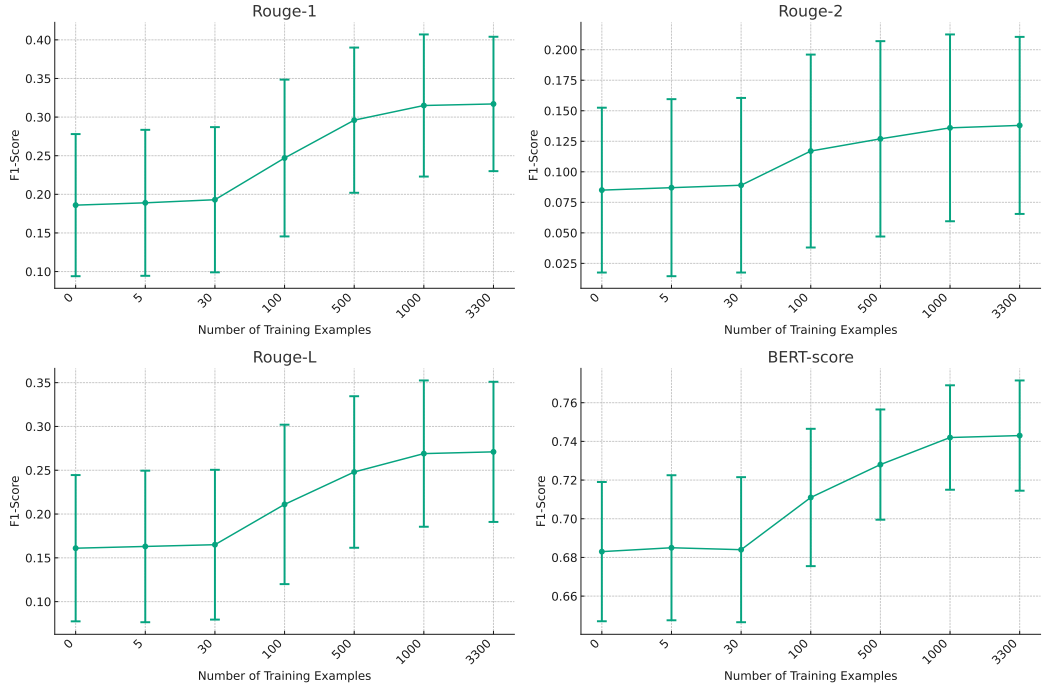


Fig. 4.1 Fine-Tuning Llama 2 7B for headline generation.

Human evaluation. We evaluate a subset of automatically generated summaries, including 250 headlines and abstracts, with the assistance of domain experts. The qualitative results for abstract and headline generation are summarized in Tables 4.7 and 4.8, respectively.

Our analysis reveals that as the number of training samples used for fine-tuning the LLM increases, all performance metrics show a consistent improvement, reflect-

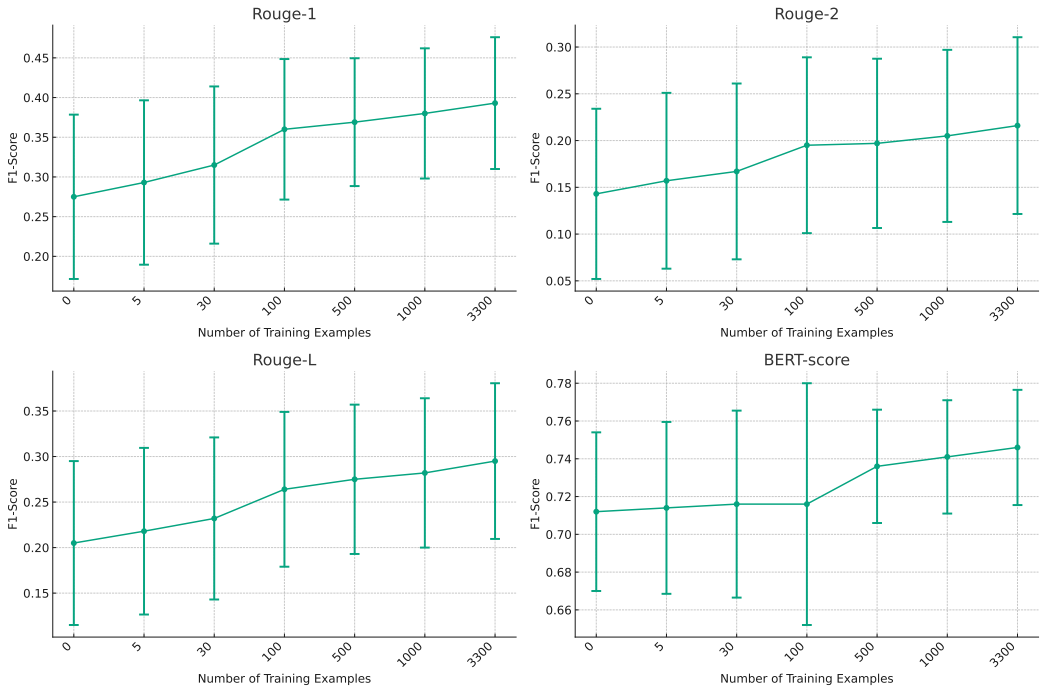


Fig. 4.2 Fine-Tuning Llama 2 7B for abstract generation.

ing enhanced content quality. Both abstracts and headlines exhibit satisfactory levels of grammatical accuracy and informativeness across all training conditions. However, non-redundancy scores remain relatively stable with minimal variation. Notably, headlines consistently receive higher overall quality scores compared to abstracts. This disparity can be attributed, in part, to the differences in length between abstracts and headlines.

We also ask annotators to express their model preference. They were instructed to select the optimal model, favoring those with fewer training data in cases of parity. For abstract generation, the analysis revealed that 51% of the abstracts were preferred when produced by the Full Training model. In contrast, 34% of the abstracts were deemed best when generated by the model fine-tuned with 1000 training samples, and 11% were favored when using the model fine-tuned with 500 samples. Less than 4% of the outputs were considered unsatisfactory by the annotators.

Regarding headline generation, 7% of the results were classified as unsatisfactory. For the remaining summaries, model preferences were fairly evenly distributed. The Full Training model was preferred for 36% of the headlines, while the few-shot

models with 1000 and 500 training samples were favored in 35% and 22% of cases, respectively.

A summary of the common errors identified by domain experts is outlined below:

- **Incomplete Sentences and Token Overflow**: around 60% of errors in abstracts are caused by truncated or incomplete sentences, likely caused by the language model exceeding the maximum token limit during output generation. This limitation not only disrupts the flow and coherence of the text but may also lead to a loss of essential information, compromising the usability of the abstracts in professional legal contexts.

- **Headline Tone and Effectiveness**: The tone and stylistic choices in some headlines are not suitably tailored to the expectations and conventions of the legal community in 5% of error cases. For instance, overly informal or sensational phrasing may undermine the perceived credibility of the content when presented to legal professionals such as lawyers, judges, or regulatory analysts. Effective communication in this domain requires a more precise, neutral, and authoritative tone.

- **Relevance of Abstracts**: A small subset (5%) of the generated abstracts fails to include critical legal references, such as citations to relevant statutes, case law, or regulatory provisions. This omission limits the abstract's usefulness for legal practitioners who rely on accurate contextual grounding to assess the applicability and importance of the summarized content. Incorporating these domain-specific elements is essential for ensuring the abstracts meet the informational needs of their intended audience.

- **Factual Correctness and Hallucinations**: Around 30% of summaries contain factual inaccuracies or fabricated information, a phenomenon commonly referred to as "hallucination" in large language models. These errors include incorrect citations of legal codes, mischaracterization of case outcomes, and invented regulatory references. Such inaccuracies can significantly undermine trust in the generated content, especially in legal contexts where precision and factual integrity are paramount.

	GR	SC	SI	NR	OQ
Few-shot (N=500)	4.96	2.76	2.76	4.99	2.88
Few-shot (N=1000)	4.96	3.01	3.04	4.97	3.04
Full training	4.97	3.156	3.17	4.98	3.23

Table 4.7 Human evaluation of Llama 2 7B for abstract generation

	GR	SC	SI	NR	OQ
Few-shot 500	4.87	4.64	4.66	4.98	4.68
Few-shot 1000	4.96	4.69	4.69	5	4.76
Full training	4.94	4.75	4.75	4.98	4.76

Table 4.8 Human evaluation of Llama 2 7B for headline generation.

Extractive summarization Table 4.9 provides a comparative analysis of various extractive summarization models. The evaluation employs the same reference summaries as those used for assessing the abstractive summarization methods. The extractive models were tested against these reference standards, which were initially established for the abstractive approaches. Generally, extractive models underperform relative to their abstractive counterparts. Among the extractive models evaluated, BERTExtr demonstrates the highest performance.

Table 4.10 presents a detailed performance analysis of BERTExtr across different law areas and summary types. Similar to the findings with abstractive models, the performance metrics for abstract generation show greater stability compared to headline generation. Overall, BERTExtr performance is inferior to that of the abstractive models, which are more adept at capturing text nuances and contextual information.

4.3 New resources for italian legal document summarization

In this study [117], we address the gap in models and datasets for training Italian language models specifically tailored for the legal domain. The key contributions of our work are as follows:

Target	Model	Rouge-1	Rouge-2	Rouge-L	BERT-score
Abstract	KL-Sum	0.304	0.146	0.218	0.711
	TextRank	0.350	0.188	0.257	0.731
	LSA	0.347	0.187	0.254	0.730
	LexRank	**0.356**	**0.194**	**0.260**	**0.733**
	BERTExtr	0.31	0.156	0.224	0.718
Headline	KL-Sum	0.145	0.054	0.118	0.661
	TextRank	0.172	0.072	0.137	0.674
	LSA	0.174	0.071	0.138	0.675
	LexRank	0.176	0.073	0.140	0.674
	BERTExtr	**0.186**	**0.080**	**0.150**	**0.679**

Table 4.9 Quantitative evaluation of extractive summarization models

- **Introduction of New Pre-trained Models**: we introduce `LegItBART`, a novel pre-trained model based on BART, specifically adapted for the legal domain. This model, due to its domain-specific and language-specific pretraining, effectively captures the intricacies of Italian legal texts.
- **Development of Benchmark Datasets**: to facilitate the fine-tuning of abstractive summarization models capable of producing concise summaries of Italian legal documents, we release two benchmark datasets, `LawCodes` and `LegItConcepts`. These datasets include a range of legal document types with diverse characteristics, such as civil and penal codes and procedural texts.
- **Fine-tuning of Models**: we fine-tune the pre-trained model on the newly introduced datasets for the abstractive summarization task. Given the long length of the input documents, we also fine-tune a model equipped with sparse attention mechanism to handle extended textual sequences (up to 16k tokens).
- **Comprehensive Empirical Evaluation**: We perform an extensive quantitative and qualitative analysis of the fine-tuned model's performance in comparison with other language models [98, 100, 67]. Our empirical evaluation examines various factors, including the type of pre-training and fine-tuning data, model complexity in terms of the number of parameters, and the effectiveness of Large Language Models for this specific application.

Task	Law area	Rouge-1	Rouge-2	Rouge-L	BERT-score
Abstract	Accounting	0.295	0.152	0.220	0.720
	Finance	0.328	0.183	0.239	0.725
	Taxation	0.300	0.160	0.211	0.719
	Business	0.315	0.162	0.226	0.722
	Job	0.293	0.140	0.201	0.716
	Digital	0.267	0.144	0.198	0.713
Title	Accounting	0.239	0.120	0.193	0.698
	Finance	0.200	0.100	0.157	0.687
	Taxation	0.181	0.075	0.144	0.677
	Business	0.175	0.069	0.148	0.672
	Job	0.190	0.081	0.153	0.680
	Digital	0.192	0.090	0.177	0.683

Table 4.10 Performance analysis of the BERTExtr extractive summarization model: evaluation across different tasks and legal domains

4.3.1 New curated datasets

In this section we discuss the characteristics of the newly released datasets namely *LawCodes* and *LegItConcepts*.

LawCodes This dataset is derived from the online versions of the Italian legal codes, including the Civil Code[2], Penal Code[3], Civil Procedure Code[4], and Criminal Procedure Code[5]. For each article within these legal codes, the corresponding article headlines—extracted from the main text—serve as the reference summaries. An example illustrating the article-summary pair is provided in Table 4.11.

LegItConcepts This dataset encompasses 155 broad, cross-topic categories from the Italian Wikipedia, each related to Italian law, such as "Corte costituzionale" (Constitutional Court), "Teoria del diritto penale" (Criminal Law Theory), and "Obbligazioni" (Obligations). Each category summary consists of the top two sentences from the introductory section of the respective Wikipedia page, providing

[2] https://it.wikisource.org/wiki/Codice_civile (accessed October 2023)
[3] https://it.wikisource.org/wiki/Codice_penale (accessed October 2023)
[4] https://it.wikisource.org/wiki/Codice_di_Procedura_Civile (accessed October 2023)
[5] https://it.wikisource.org/wiki/Codice_di_procedura_penale (accessed October 2023)

Law article	Law article title
Original: Nell'applicare la legge non si può ad essa attribuire altro senso che quello fatto palese dal significato proprio delle parole secondo la connessione di esse, e dalla intenzione del legislatore. Se una controversia non può essere decisa con una precisa disposizione [...]	Interpretazione della legge
English: In applying the law, no other meaning can be attributed to it than the one clearly derived from the literal meaning of the words, considering their connection and the intention of the legislator. If a dispute cannot be resolved with a specific provision [...]	Interpretation of the law.
Original: Nuove disposizioni che prevedono reati possono essere introdotte nell'ordinamento solo se modificano il codice penale ovvero sono inserite in leggi che disciplinano in modo organico la materia.	Principio della riserva di codice
English: New provisions that establish offenses can only be introduced into the legal system if they modify the penal code or are included in laws that systematically regulate the subject matter.	Principle of legality

Table 4.11 *LawCodes*: examples of law articles and their corresponding article titles.

a high-level overview of the category. These summaries are extracted from the main document for our analysis. An example of a category-summary pair is illustrated in Table 4.12. Data retrieval was conducted using the PetScan tool[6], with the maximum search depth set to zero to ensure content relevance.

Statistics Table 4.13 provides key statistics for the newly introduced datasets, as outlined below:

- **Average Number of Tokens per Input Text and Output Summary**: This metric reflects the typical complexity involved in encoding the input text and generating the corresponding summary. A greater average length generally indicates more demanding modeling requirements.
- **Percentage of Novel n-grams**: This metric represents the percentage of sequences composed of *n* tokens in the summary that do not appear in the input text. The value of *n* is varied between 1 (unigrams) and 4 (fourgrams). This statistic is particularly informative for distinguishing between the demands for extractive versus abstractive summarization techniques.

[6] https://petscan.wmflabs.org/ (accessed October 2023)

Concept/fact description	Summary
Original: La presentazione della dichiarazione di successione e della voltura catastale, oltre ad eventuali altre dichiarazioni e/o comunicazioni, quali ad esempio quelle previste in caso di “beni culturali”, sono obbligatorie per legge. La mancanza di detti adempimenti comporta sanzioni amministrative e problematiche che vengono scoperte dagli interessati soltanto dopo tempo, con rilevanti sanzioni e limiti. La presentazione tardiva, ma non eccessivamente procrastinata nel tempo viene risolta in modo agevole [...]	La Dichiarazione di successione , nel diritto civile italiano, è una dichiarazione che gli eredi e/o legatari di un deceduto sono obbligati a presentare, all’Ufficio dell’Agenzia delle Entrate competente, entro 12 mesi dalla morte nel caso il defunto abbia lasciato anche soltanto un immobile o denaro per almeno 100 000 euro.
English: The filing of the inheritance declaration and the cadastral transfer, as well as any other declarations and/or communications, such as those required for "cultural assets," are legally mandatory. Failure to comply with these obligations results in administrative sanctions and issues that are discovered by the parties involved only after some time, leading to significant penalties and limitations. However, a slightly delayed submission can be easily and not excessively burdensomely [...]	In Italian civil law, the inheritance declaration is a statement that the heirs and/or legatees of a deceased person are obliged to submit to the competent Office of the Revenue Agency within 12 months from the date of death, in cases where the deceased has left at least one property or cash amounting to at least €100,000.
Original: A partire dagli articoli 168-177 della Costituzione albanese, la Corte costituzionale decide: 1. la compatibilità della legge con la Costituzione o con gli accordi internazionali di cui all’articolo 126 2. la compatibilità degli accordi internazionali con la Costituzione, prima della loro ratifica 3. la compatibilità degli atti normativi degli organi centrali e locali con la Costituzione e gli accordi internazionali 4. i conflitti di competenze tra poteri, così come tra governo centrale e governo locale [...]	La Corte costituzionale d’Albania, è il più alto organo giudiziario che garantisce il rispetto della Costituzione. La Corte costituzionale è composta da 9 membri, che sono nominati, a partire dalla riforma costituzionale del 2016: 3 dal presidente della Repubblica, 3 dal’Assemblea e 3 dalla Corte Suprema.
English: Starting from articles 168-177 of the Albanian Constitution, the Constitutional Court decides: 1. The compatibility of laws with the Constitution or with international agreements referred to in Article 126. 2. The compatibility of international agreements with the Constitution before their ratification. 3. The compatibility of legislative acts of central and local bodies with the Constitution and international agreements. 4. Conflicts of jurisdiction between powers, as well as between the central government and local government. [...]	The Constitutional Court of Albania is the highest judicial body that guarantees the respect of the Constitution. The Constitutional Court is composed of 9 members, who are appointed following the constitutional reform of 2016: 3 by the President of the Republic, 3 by the Assembly, and 3 by the Supreme Court.

Table 4.12 Illustrative examples of input documents and corresponding summaries in the LegIt Wiki dataset

- **Coverage**: Defined as the percentage of tokens in the summary that are part of an extractive fragment present in the original document, this metric provides insight into the extent of content preservation in the summary.

$$\text{coverage}(A,S) = \frac{1}{|S|} \sum_{f \in \mathscr{F}(A,S)} |f| \tag{4.4}$$

where A denotes the input text, represented as $A = \langle a_1, a_2, \ldots, a_n \rangle$, S is the corresponding summary, expressed as $S = \langle s_1, s_2, \ldots, s_m \rangle$ with length $|S|$, and $\mathscr{F}(A,S)$ represents the set of token sequences that are common to both A and S. This measure quantifies the degree to which the summary is representative of the original document.

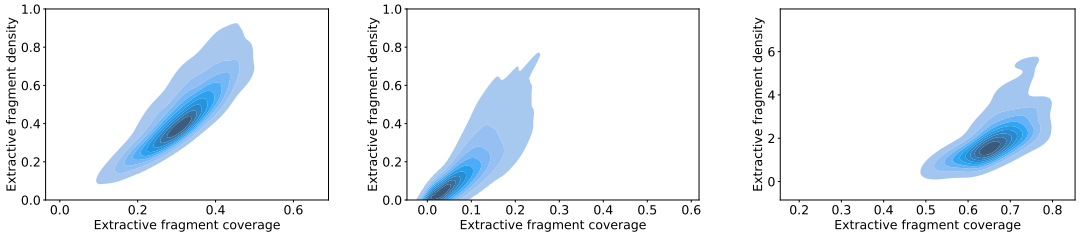
- **Density**: This metric quantifies the concentration of content from the input document that is captured within the summary. It is formally defined as follows:

$$\text{density}(A,S) = \frac{1}{|S|} \sum_{f \in \mathscr{F}(A,S)} |f|^2 \tag{4.5}$$

- **Compression ratio**: is defined as the ratio of the length of the original document (measured in tokens) to the length of the summary. This ratio provides insight into the degree of reduction in content from the input text to the generated summary.

	Law codes		**Law procedures**		**LegItconcepts**	**EUR-Lex-Sum**
	Penal	**Civil**	**Penal**	**Civil**		
Input tokens	135.75	98.57	181.96	142.27	2346.39	23938.8
Output tokens	10.52	8.74	9.85	8.99	96.38	1435.23
New 1-grams (%)	53.86	43.83	38.37	44.51	45.36	40.21
New 2-grams (%)	78.93	74.30	70.30	76.01	81.88	64.21
New 3-grams (%)	87.14	85.97	81.45	87.00	93.00	78.40
New 4-grams (%)	91.54	91.92	88.94	92.07	96.70	84.59
Coverage	0.08	0.08	0.08	0.07	0.31	0.61
Density	0.18	0.15	0.16	0.13	0.45	2.37
Compression Ratio	10.86	12.02	13.26	12.48	9.48	5.21

Table 4.13 Summary of the key dataset statistics



(a) LegItConcepts (b) LawCodes (c) EURLEX-sum (it)

Fig. 4.3 Density estimation of extractive diversity scores. The y-axis reflects variability in the length of source sequences included in the summary, while the x-axis shows the variation in the length of extractive fragments containing summary content. Significant variability on either axis suggests differences in how source content is summarized.

Comparison between existing dataset We compare the statistics of LawCodes and LegItConcepts (see Table 4.13) with those of the leading benchmarks for legal document summarization, specifically the EUR-Lex-Sum dataset [91] and the one and the Multi-Legal Pile dataset [118].

The output summaries in EUR-Lex-Sum are significantly longer—by two orders of magnitude compared to those in LawCodes and one order of magnitude longer than those in LegItConcepts. For the Multi-Legal Pile dataset, ground truth summaries are not provided, limiting its use to model pretraining rather than fine-tuning. While EUR-Lex-Sum is suitable for training abstractive models, its fine-tuning process is unlikely to produce concise summaries, such as headings or brief definitions, due to the substantial difference in the characteristics of its training samples.

The percentages of new *n*-grams in the dataset summaries, as presented in Table 4.13, further validate this observation. Summaries in LegItConcepts exhibit a substantial number of *n*-grams absent from the input documents and a high compression ratio of 9.48, indicating a significant level of abstraction. In contrast, the EUR-Lex-Sum dataset has a lower compression ratio of 5.21, reflecting less abstraction in its summaries.

These findings underscore the necessity for specialized summarization models tailored to the unique characteristics of each dataset.

Figure 4.3 presents a comparative analysis of the density versus coverage scatter plots for the LawCodes, LegItConcepts, and EUR-Lex-Sum datasets. The EUR-Lex-Sum dataset is characterized by relatively high coverage values but low-density

values, suggesting that while the summaries are predominantly extractive, they may omit important content, leading to some inaccuracies.

In contrast, LawCodes and LegItConcepts exhibit lower coverage values, which aligns with the nature of abstractive summarization models. As anticipated, LegItConcepts displays higher density values compared to LawCodes, likely because the introductory sections in LegItConcepts provide more comprehensive information than mere document headings.

Moreover, it is important to note that the input documents in LegItConcepts are considerably more verbose than those in LawCodes and related procedures (2300 tokens vs. 100-200 tokens). This highlights the necessity for text encoding strategies capable of efficiently handling longer texts.

4.3.2 Methodology: BART-IT

Pretraining for Legal Domain Adaptation We utilize the denoising objectives from the BART pre-training framework, including document rotation, sentence permutation, token infilling, token masking, and token deletion [99]. These perturbations create corrupted versions of the input texts in a self-supervised manner. The pre-training objective for LegItBART is to reconstruct the original sentence from its corrupted form, thereby improving the model's resilience to noise and enhancing its comprehension capabilities.

For pre-training, we employ a subset of Italian legal documents from the Multi-Legal Pile dataset [119], with the aim of adapting the model to understand the specialized language and terminology used in legal contexts. Within this subset, we specifically extract documents categorized as *contracts*, *legislation*, *caselaw*, and *other*, to ensure a focused and relevant pre-training process.

To ensure relevance within the *other* category, we apply document clustering and BERT-based topic modeling [120]. Specifically, we group documents into clusters based on thematic similarity. Each cluster is characterized by extracting the top-10 most representative keywords. Clusters deemed irrelevant to the legal domain, along with any outliers, are filtered out. This topic-level filtering eliminates fewer than 4% of the original documents.

We explore two alternative strategies for pre-training the model:

- **Train LegItBART from Scratch**: This strategy involves training LegItBART exclusively on legal texts. This approach is limited by the relatively small number of available legal documents.
- **Specialize an Existing Model**: We initialize LegItBART using the pre-trained weights of the general-purpose Italian model BART-IT [98][7]. This model already possesses an understanding of Italian language patterns and terminology, and the goal is to adapt it to the Italian legal domain.

The model is trained using standard cross-entropy loss to optimize the reconstruction of original inputs from their corrupted versions. During training, LegItBART applies text corruption techniques at both the sentence and token levels. At each training iteration, a type of corruption—either sentence-level or token-level—is randomly chosen to ensure the model encounters a diverse range of perturbations.

Sentence-level corruptions Two methods are employed to induce sentence-level variations through *document rotation* and *sentence permutation*.

- *Document Rotation*: This method involves segmenting the input document into individual sentences by splitting on full-stop punctuation. A random sentence is then chosen as the starting point, with the remaining sentences following in their original sequence. This process creates a modified document structure that retains the original sentence order but starts from a different point.
- *Sentence Permutation*: This approach also begins by dividing the text into sentences. However, the sentences are then randomly shuffled, resulting in numerous potential orderings. Unlike document rotation, which preserves some degree of the original order, sentence permutation introduces complete variability in the arrangement of sentences.

These sentence-level corruption techniques are employed during the pre-training phase to expose LegItBART to a range of structural alterations. By encountering these variations, the model enhances its capability to interpret legal texts even when the ordering or emphasis of sentences is modified. This is particularly pertinent

[7]We initialize LegItBART with the pre-trained weights of BART-IT and continue training on legal documents.

in legal documents, where critical information can be conveyed through specific phrasing or the arrangement of clauses.

The application of sentence-level corruptions thereby improves the model's robustness in identifying and retaining significant facts, irrespective of superficial changes in sentence order or structure. This approach also aids LegItBART in comprehending complex legal terminology across various possible textual configurations.

Token-level corruptions LegItBART incorporates three token-level perturbations during pre-training to enhance its capacity to reconstruct texts from incomplete information.

- **Token Infilling**: In this perturbation, a span of tokens is selected randomly, with the span length L drawn from a Poisson distribution with a mean $\lambda = 3$. This span, a continuous sequence of tokens of length L, has a 15% probability of being replaced entirely with a single `[MASK]` token. This method introduces variability by substituting entire spans of text, challenging the model to predict the omitted content.
- **Token Masking**: This technique operates at a more granular level than token infilling. Here, each individual token in the input sequence has a 15% probability of being replaced with a `[MASK]` token. Unlike token infilling, which affects entire spans, token masking modifies only one token at a time, requiring the model to infer the masked tokens from the surrounding context.
- **Token Deletion**: For token deletion, each token in the input text is independently removed with a 15% probability. The model is then tasked with predicting the missing tokens that have been randomly omitted. This perturbation simulates scenarios where tokens are absent, enhancing the model's ability to handle incomplete or erroneous data.

These token-level perturbations are designed to improve LegItBART 's resilience and accuracy in real-world applications where portions of legal terminology might be unclear, misspelled, or missing. By incorporating various types of noise, the model is trained to robustly handle diverse contexts involving incomplete or unreliable information.

Encoding long legal documents

Transformer-based models previously employed for abstractive legal document summarization [101] are typically constrained by their capacity to handle a limited number of tokens (e.g., 1204 tokens). This constraint presents significant challenges in the legal domain, where documents such as legal codes, procedures, and articles frequently exceed this token limit.

To address this limitation, LegItBART incorporates an encoder-based Local-Sparse-Global (LSG) attention mechanism [102]. The LSG attention mechanism reduces computational complexity to linear time relative to the number of input tokens, enabling the model to process sequences of up to 16,384 tokens—16 times longer than the token capacity of traditional models.

The LSG mechanism achieves this by employing i) Local Attention, where each token attends only to a fixed-size window of surrounding tokens, thereby constraining the attention scope and reducing computational demands and ii) Sparse Global Attention where LSG selectively establishes a sparse set of global connections between tokens, rather than attending to all tokens. This selective connectivity ensures that the model captures essential long-range dependencies while maintaining efficiency.

Fine-Tuning for Abstractive Summarization

We fine-tune the pre-trained model using a supervised approach with the legal document-summary pairs obtained from the newly released benchmarks. In this context, the reference summary serves as the target sequence, guiding the model towards the generation of precise abstractive summaries.

During fine-tuning, the model is trained to predict the subsequent token in the summary sequence based on the preceding tokens and the encoded input document. This process enables the model to effectively learn to generate summary tokens that accurately reflect the key information from the input legal documents.

The model optimizes its performance using the autoregressive cross-entropy loss function, which is formulated as follows:

$$\mathscr{L} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T-1} y_{nt}\log(p_{t|t-1}) \tag{4.6}$$

where N represents the number of training examples per batch, T denotes the length of the summary, y_{nt} is the target token at position t for the n-th training example, and $p_{t|t-1}$ is the predicted probability distribution over tokens at timestep t, conditioned on the preceding tokens.

4.3.3 Experimental details

Metrics We use the following established metrics to quantitatively evaluate the output summaries under complementary aspects:

- **ROUGE** [114]: it quantifies the syntactic similarity between generated text and reference summary by counting their n-gram overlap. Specifically, hereafter we will consider the F1-scores of the ROUGE-1 (unigram overlap), ROUGE-2 (bigram overlap), and ROUGE-L (longest common subsequence).
- **BERTScore** [121]: it measures the semantic similarity between generated and reference summaries based on the BERT embeddings [110].

Experimental details The pre-training phase for the model was performed on a high-performance computing setup featuring an Intel® Core™ i9-10980XE CPU and dual Nvidia® RTX A6000 GPUs, complemented by 128 GB of RAM. This configuration was essential to accommodate the substantial computational requirements inherent to pre-training transformer-based models. The entire pre-training process was completed within approximately two days.

For the fine-tuning phase, a single Nvidia® RTX A6000 GPU was utilized, given the relatively reduced computational demands in comparison to pre-training. Each model variant was fine-tuned on the legal summarization datasets until no further improvement was observed in validation performance.

Competitors: Italian Sequence-to-Sequence models. IT5 [100] and BART-IT [98] are two pre-trained sequence-to-sequence models designed for general-purpose Italian natural language processing (NLP) and natural language generation

(NLG) tasks. IT5 is based on the T5 architecture [122], while BART-IT is derived from the BART architecture [99]. Both models are pre-trained on the Italian subset of the mC4 Corpus [123], which has been curated to exclude documents with offensive content, code snippets, or excessively brief sentences.

IT5 [100] utilizes span masking objectives during pre-training and has demonstrated strong performance across a range of downstream tasks, including summarization. IT5 is available in various model sizes, including small, base, and large configurations, allowing for flexibility depending on computational resources and task requirements. The base model configuration (IT5-base) contains 220 million parameters, making it the second largest publicly available IT5 model.

In this study, we compare the performance of our proposed LegItBART model against IT5-base. Both models are similarly sized, with IT5-base having 220 million parameters and LegItBART containing 141 million parameters. This comparison allows us to evaluate the effectiveness of each model while controlling for size-related differences.

BART-IT [98] is pre-trained using various text denoising objectives and has demonstrated robust performance on multiple Italian abstractive summarization datasets. The authors of BART-IT assert that it achieves an optimal balance between efficiency and effectiveness for Italian abstractive summarization.

Both LegItBART and BART-IT are based on the BART architecture and share similar pre-training methodologies. However, LegItBART extends the pre-training process by incorporating domain-specific Italian legal documents. This comparison of LegItBART with BART-IT serves to assess the benefits of tailoring a model to the legal domain through two different approaches: one that trains LegItBART from scratch on Italian legal texts (LegItBART-LPT) and another that continues the pre-training of BART-IT on legal documents (LegItBART).

Evaluating LegItBART against BART-IT, alongside its variants, allows for an investigation into whether initial or continued pre-training on legal texts enhances summarization capabilities for legal content, independent of the model's parameter size.

The finetuning process employed a maximum learning rate of 5×10^{-5}, with a learning rate scheduler that implemented a linear warmup over the initial 500 steps, followed by a linear decay phase. Each model was trained for up to 10 epochs, and

the model checkpoint that achieved the highest performance on the validation set was selected.

By applying a uniform optimization strategy and hyperparameter configuration across all models during fine-tuning, we ensure that the performance evaluation is unbiased and not influenced by variations in training procedures among the models.

Competitors: Open source LLMs. LLaMA 2 [67] comprises a series of pre-trained large language models (LLMs) with parameter counts ranging from 7 billion to 70 billion, designed specifically for dialogue-related tasks. These models build upon the foundational pretraining and architectural principles established by LLaMA 1 [67]. Notably, LLaMA 2 demonstrates superior performance compared to LLaMA 1 and other models across benchmarks evaluating commonsense reasoning, world knowledge, and reading comprehension.

In contrast to IT5 and BART-IT, which were trained exclusively on Italian text, LLaMA 2 was trained on a diverse multilingual corpus that includes Italian among over 20 other languages. This broad training base allows LLaMA 2 to leverage in-context learning (ICL) [96], a technique that enables the model to handle previously unseen tasks by providing a few examples without requiring gradient-based adjustments.

However, using ICL involves significant computational costs as it requires including all input-target pairs for each prediction. To mitigate this issue, parameter-efficient fine-tuning (PEFT) techniques [26] have been developed. PEFT adapts pre-trained models to specific downstream tasks by only fine-tuning a minimal subset of parameters. This approach reduces memory requirements, allows for mixed-task batches during inference, and entails far fewer parameters to fine-tune compared to traditional methods. In this study, we employ the LoRA (Low-Rank Adaptation) fine-tuning method [27], which involves freezing the weights of the pre-trained model and incorporating trainable low-rank decomposition matrices into each transformer block. This approach substantially reduces the number of parameters that need to be adjusted for downstream tasks.

Specifically, we apply LoRA fine-tuning for three epoch, with a rank $R = 64$ and a scaling factor $\alpha = 16$, using the 7 billion parameter LLaMA 2 base model. Due to computational constraints, our experiments are limited to this model size. The efficient LoRA technique ensures that the memory footprint during training remains

comparable to that of BART-IT and IT5. To align with our computational resources, we set the maximum input length to 2048 tokens while fine-tuning the LLaMA 2 model on the legal summarization task. This setup allows us to effectively manage memory usage while still leveraging the model's capabilities for the task at hand.

4.3.4 Results

Comparison between pre-training strategies We assess the efficacy of the pre-training phase in adapting the model to the legal domain.

Figure 4.4 illustrates the training and validation loss variations for LegItBART , where one variant is initialized with pre-trained weights from BART-IT (LegItBART), and the other is initialized with random weights (LegItBART-LPT).

In the initial stages of training, LegItBART-LPT shows considerably higher training and validation losses compared to LegItBART. This indicates that initiating training from scratch on legal texts presents greater challenges than fine-tuning a well-established Italian language model with continued pre-training. As training progresses, the loss gap between LegItBART-LPT and LegItBARTnarrows, with LegItBART-LPT gradually approaching the loss values of LegItBART. Nevertheless, LegItBART-LPT does not fully match the loss levels achieved by LegItBART, even after complete pre-training.

These findings suggest that continuing the pre-training of an Italian language model on legal texts is more effective than training from scratch exclusively on legal data. The model that builds on pre-existing Italian language knowledge reaches robust performance more efficiently, demonstrating the advantages of leveraging prior general language understanding for specialized tasks.

Effect of encoding longer sequence inputs We compare a standard summarization model, constrained to a maximum of 1024 input tokens, with a variant capable of handling up to 16384 input tokens. The evaluation results are presented in Table 4.14, where # input and # output denote the maximum token limits for the input and output texts, respectively.

LegItBART outperforms other models even when limited to 1024 input tokens, achieving a $+2.57$ improvement in the ROUGE-1 score over the second-best per-

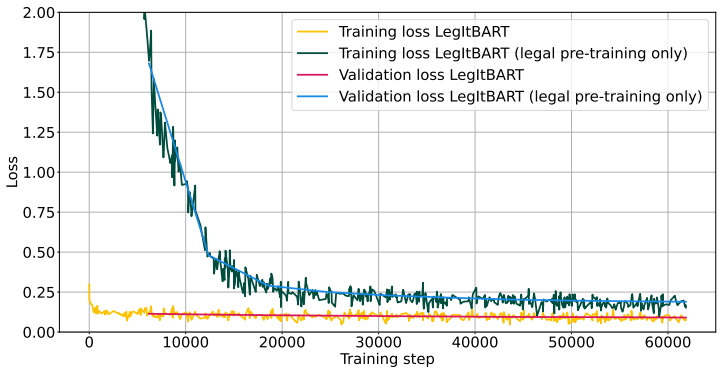


Fig. 4.4 Pre-training and validation loss of LegItBART .

forming model, BART-IT. This suggests that LegItBART is generally more adept at modeling and understanding legal texts. When extended with local-sparse-global (LSG) attention, the performance of LegItBART improves substantially across all tested conditions, with ROUGE-1 score enhancements ranging from +7 to +14 points. The LLaMA 2 model shows competitive performance; however, it falls short of the LSG-enhanced LegItBART, highlighting the advantages of processing longer documents with LSG. LegItBART-LPT , which was pre-trained solely on legal text, does not achieve the performance levels of LegItBART. This indicates that specialized pre-training alone may be insufficient for developing comprehensive text understanding and generation capabilities.

	# input	# output	R-1	R-2	R-L	BERT-score
BART-IT	1024	1024	28.04	11.94	18.85	71.38
IT5-base	1024	1024	11.84	6.57	9.08	54.76
LegItBART-LPT	1024	1024	27.77	11.89	19.21	69.22
LegItBART	1024	1024	30.61	13.59	19.18	72.75
BART-IT	16384	1024	42.57	20.58	22.99	**76.62**
LegItBART-LPT	16384	1024	34.81	16.67	21.99	72.59
LegItBART	16384	1024	**43.05**	**20.99**	**23.29**	76.58
Llama 2	2048		34.35	11.92	20.81	59.45

Table 4.14 Evaluation of LegItBART versus baseline models on the EUR-Lex-Sum dataset

Finetuning on LawCodes As discussed earlier, the documents and summaries in the dataset are relatively short. Given the short lengths of both the inputs and outputs, we will focus on evaluating the LegItBART version without extended input sequences.

Table 4.15 summarizes the performance of various models on the LawCodes dataset. The results reveal that LegItBART outperforms all other models across all evaluation metrics. This superior performance indicates that the domain-specific legal language and structures encountered during LegItBART 's pre-training significantly enhance its ability to summarize documents effectively.

Notably, both IT5 and BART-IT perform relatively well on this domain-specific dataset, with IT5 showing superior performance compared to BART-IT in terms of ROUGE metrics. However, LegItBART-LPT, which was pre-trained solely on legal data, shows lower performance compared to other models. This suggests that general-purpose pre-training provides a substantial advantage even in datasets with a high degree of domain specificity. LegItBART-LPT's results are probably due to the lack of diversity in domain-specific data, which might not capture the full range of language structures and patterns required for effective summarization.

Overall, the results for the LawCodes dataset underscore the effectiveness of LegItBART in handling highly abstractive legal texts. Its superior performance can be attributed to its domain-specific pre-training, which equips the model with the necessary capabilities to manage the specialized language and structures of legal documents. The relatively good performance of general-purpose models on this dataset is also noteworthy, indicating that while these models were not pre-trained on legal data, they can still perform reasonably well on domain-specific tasks. Combining general-purpose pre-training with domain-specific fine-tuning, as demonstrated by LegItBART , may offer a path to further enhancing model performance.

Finetuning on LegItConcepts The experimental results presented in Table 4.16 reveal that when models are constrained to a maximum input length of 1024 tokens, BART-IT achieves a higher ROUGE-1 score compared to LegItBART , while LegItBART performs better in terms of ROUGE-2. When input lengths are extended to accommodate up to 16384 tokens using Local-Sparse-Global (LSG) attention, the

	# input	# output	R-1	R-2	R-L	BERT-score
BART-IT	1024	128	32.04	17.54	31.55	78.3
IT5-base	1024	128	33.72	18.19	33.01	76.9
LegItBART-LPT	1024	128	29.97	15.97	29.26	77.65
LegItBART	1024	128	**35.42**	**20.66**	**34.92**	**79.32**
Llama 2	2048		25.08	12.03	23.65	73.33

Table 4.15 LegItBART and baseline competitors performance on LawCodes dataset.

performance improvements across all models are modest. This suggests that the increased input length may not significantly enhance performance on this dataset.

Interestingly, BART-IT consistently outperforms all other models, including LegItBART, by a small margin (e.g., +0.42 in ROUGE-1 and +0.06 in ROUGE-2). This indicates that the general-purpose BART-IT model may be more adept at handling the broad range of topics present in the LegItConcepts dataset.

Furthermore, LegItBART-LPT, which was pre-trained exclusively on legal data, shows notably lower performance compared to other models. This suggests that the LegItConcepts dataset is not purely legal in nature. Instead, it seems to target a broader audience, with content that is less technical and more accessible. As a result, the specialized legal terminology may not be as beneficial in this context, highlighting the importance of aligning the model's training data with the dataset's characteristics and audience.

In this context, additional pre-training on legal data may offer limited benefits and could potentially degrade performance. This is evident from the notably lower performance of LegItBART-LPT, which was exclusively pre-trained on legal texts. General-purpose models, which include diverse data sources such as Wikipedia, benefit from their broad coverage of topics and varied writing styles. For instance, BART-IT, which was pre-trained on general-purpose Italian data, performs optimally on the LegItConcepts dataset. Its superior performance can be attributed to its exposure to a wide range of language patterns and writing styles. The efficacy of general-purpose models is further illustrated by IT5, which, although slightly inferior to BART-IT, still achieves commendable results. In contrast, while LLaMA 2 demonstrates competitive performance, it does not surpass LegItBART. This indicates that, for the LegItConcepts dataset, the advantage lies in models trained on a broader spectrum of data rather than those solely specialized in legal texts.

	# input	# output	R-1	R-2	R-L	BERT-score
BART-IT	1024	128	32.5	14.32	24.35	73.29
IT5-base	1024	128	24.31	7.38	17.69	67.52
LegItBART (scratch)	1024	128	28.06	10.93	20.6	70.53
LegItBART	1024	128	32.02	14.77	24.23	73.13
BART-IT	16384	128	**32.74**	**14.89**	**24.85**	**73.33**
LegItBART-LPT	16384	128	29.35	11.94	21.66	70.94
LegItBART	16384	128	32.32	14.83	24.59	73.15
Llama 2	2048		31.50	14.76	21.341	68.20

Table 4.16 LegItBART and baseline competitors performance on LegItConcepts dataset.

4.4 Summary of results and key insights

Several key findings emerge that demonstrate the effectiveness of different approaches and their practical implications. The experiments demonstrate that combining general-purpose pre-training with targeted legal domain adaptation represents the most effective strategy, with LegItBART (initialized from BART-IT and further pre-trained on legal texts) consistently outperforming both general-purpose models and models trained exclusively on legal data from scratch. For abstractive summarization specifically, LSG-BART-It achieved superior syntactic performance (ROUGE scores) for abstract generation, while Llama 2 (7B) excelled in semantic similarity (BERTScore) and dominated headline generation across all metrics. The integration of Local-Sparse-Global (LSG) attention mechanism proved particularly valuable, enabling processing of significantly longer documents (up to 16,384 tokens) and resulting in substantial performance improvements of ROUGE-1 points. Importantly, the study reveals that parameter-efficient fine-tuning using LoRA significantly improves performance with benefits plateauing around 500-1000 training examples, while traditional smaller models like BART-It and IT5 often matched or outperformed much larger models like Flan-T5-XL (3B parameters), indicating that model architecture and training methodology can influence the performance. However, human evaluation uncovered critical limitations: 30% of summaries contained factual hallucinations, 60% of abstract errors involved incomplete sentences due to token limits, and challenges persisted with domain-appropriate tone, though grammatical quality remained satisfactory. The consistent underperformance of extractive methods compared to abstractive approaches confirms that legal text sum-

marization benefits from paraphrasing and abstraction capabilities, yet the prevalence of factual errors highlights the critical need for additional safeguards when deploying these systems in legal contexts where accuracy is paramount, particularly given that optimal performance depends on dataset characteristics with highly technical content favoring domain-adapted models while more accessible legal content shows comparable results across different approaches.

Chapter 5

Explainability and reasoning for complex tasks

This chapter provides a review of the related literature in Section 4.1 and introduces the research contributions for improving court judgment prediction (Section 5.2), and legal reasoning applications for solving legal issues (Section 5.4) Language Models.

5.1 Prior works

5.1.1 Court Judgement Prediction and Explanation

Existing research has primarily focused on the jurisprudence of the U.S. Supreme Court [124, 125], cases of the European Court of Human Rights [126–128] and the Indian judicial system [129, 1]. In [130] transformer-based models have demonstrated superior performance in most application scenarios [131–134], but the constraints on the maximum document length processable by a transformer-based architecture are often impractical for legal court cases. Also those models lacks in interpretability, therefore the need of Explainable AI (XAI) models to enhance the transparency of AI models [135] in the legal domain is urgent [136]. Interpretable-by-design XAI methods integrate explainability during the AI model development, resulting in models with inherent interpretability [137]. Post-hoc XAI approaches provide explanations for existing, trained models. These explanations can be provided at either a global level, offering a comprehensive view of model behavior [138–140]

or at instance-level [141–144]. [145] adopts interpretable-by-design approach, introducing a model-agnostic interpretable layer to predict interpretable intermediate scores. Similarly, [126] train an SVM classifier with a linear kernel for its direct interpretability in assessing human rights convention violations.

Despite these efforts, the superior performance of black-box models like transformer-based deep-learning models has led many studies to turn to post-hoc explanation methods. For instance, [146] apply the Grad-CAM explanation method to interpret neural network predictions in legal texts. [147] leveraged LIME [142] and Layer Integrated Gradients [148] to explain legal judgment predictions of XGBoost and Longformer models. [149] focused on evaluating the explanations provided by LIME [142] and SHAP [141]. [150] adopt local explanation methods for a legal text classification problem, measuring the plausibility of explanations with the assessments of legal professionals. A (local) occlusion-based approach is proposed in [1]: it involves masking a portion of embeddings and measuring their influence on classification probability distribution, then they assess the plausibility of the explanations compared to human rationales.

5.1.2 Legal Retrieval Augmented Generation

The Retrieve-Read [151] RAG research paradigm represents the earliest methodology: it includes indexing, retrieval, and generation.

More *advanced* RAG employs pre-retrieval and post-retrieval strategies, that includes query transformation or re-ranking [152–154].

Modular RAG add specialized components to enhance retrieval and processing capabilities. The orchestration of Modular RAG Flow showcase the benefits of adaptive retrieval techniques such as Retrieve-Read-Retrieve-Read flow of ITER-RETGEN [155], FLARE [156] and Self-RAG [157]. Although these methods introduce greater flexibility into the RAG system, they necessitate increased computational resources and substantial data volumes, particularly when fine-tuning is required [158–162]. Given the constraints of our problem, specifically the limited availability of data and resources, we will not consider these last methods.

Few works have been proposed for improving Large Language models abilities with legal knowledge. [163] propose a methodology aimed at generating extensive responses to any statutory law inquiries, employing a Retrieval-Augmented Gener-

ation pipeline. The primary objective of this research is not to resolve legal issues per se, but rather to develop an assistive system for navigating the legal domain and bridging the gap in legal literacy through the advancement of automated legal aid systems. It is worth noting that the scope of this study is confined to the French language. Similarly, [164] employs a RAG pipeline on LLM-generated question-answer pairs derived from the Australian Open Legal Corpus dataset. The study investigates various representations and comparison techniques with the aim of extracting information from court cases. In [165], the authors introduce Eval-RAG, a novel evaluation methodology for texts generated by LLMs.

In our latest work[UNDER REVIEW] we show that GPT-4 outperforms other models across all methods. Keyword search with tensor re-ranking emerges as the most effective retrieval method. We also present two effective strategies for improving the search and the generation of the most correct answer.

5.1.3 Legal Reasoning

Researchers have recently started exploring whether large language models have the capability to carry out legal reasoning. Unlike BERT-based models, LLMs are evaluated on their ability to learn tasks in-context, primarily through prompting [26]. Studies have explored the role of prompt-engineering for Legal Judgment Prediction [166], statutory reasoning [167] legal exams [168]. Several case studies [169–173] highlight the potential and the limitations of GPT models in real use cases. However, to the best of our knowledge, limited effort has been devoted to analyzing the effectiveness of smaller and open-source language models (e.g., Llama 3 [174]) in this domain [175], and how they can effectively be employed in conjunction with closed-source foundational models, such as GPT-4 [176].

5.2 Improving court judgment prediction

Applying Deep Learning and Natural Language Understanding techniques to legal court cases poses significant challenges due to several factors:

- *Language style*: the nuanced and technical nature of forensic language in legal documents makes deep content comprehension more difficult than in other domains [177, 37]
- *Model specialization*: variability of laws and regulations across different areas requires custom fine-tuning of models for each specific domain [178, 58].
- *Document length*: documents are often very lengthy, potentially exceeding the token limits that state-of-the-art AI models can handle [132, 70].
- *Outcome reliability*: Because of the critical importance of legal outcomes, experts are generally hesitant to rely on AI tools in the legal field [130, 84].

Transformer-based models have proven effective in comprehending the forensic language used in cases from the U.S. Supreme Court, the European Court of Human Rights, and the Indian judicial system [1]. This progress has partly addressed the challenge of understanding complex legal language. Simultaneously, research communities have been increasingly focusing on applying Large Language Models to legal AI tasks [179–181]. However, integrating automatic judgment prediction with form of explanations, remains problematic. Consequently, the last challenge is still unresolved.

In [182, 183], the focus is specifically on court judgment prediction and explanation using the Indian Legal Documents Corpus (ILDC), which contains English-written case proceedings from the Supreme Court of India from 1947 to 2020. The main contribution of this work is the integration of a legal named entity recognition step to enhance both judgment prediction and explanation, addressing challenge (4). Legal entities describe domain-specific concepts and facts that are crucial for a deep understanding of court case proceedings. In this study, we aim to explain court judgment predictions using a local, model-agnostic approach. Notably, post-hoc model-agnostic methods, which are independent of the classification model used for explanation, enable us to select the most suitable and best-performing model for our task, regardless of its inherent interpretability and architecture.

The proposed pipeline includes:

- **Entity Recognition:** We utilize an entity-aware attention mechanism [184] to precisely identify legal named entities in court judgments. We evaluate the

performance of domain-specific models that have been pre-trained and fine-tuned in the legal domain, examining how pre-training improves the quality of representations for the Named Entity Recognition (NER) task.

- **Entity Masking:** After identifying legal entities, we mask the corresponding text snippets. This serves two purposes: (1) Enhancing the generalization capabilities of predictive models by hiding information irrelevant to the case outcome, such as personal names. (2) Limiting the spread of potentially sensitive information, like the names of individuals involved in the court judgment.

- **Leverage Entities in Judgment Prediction and Explanation:** We use the presence of legal entities to aid in selecting specific sentences within the case proceedings. Sentences rich in semantically meaningful legal entities are considered more likely to be useful in explaining the predicted judgment. We compare the performance of a newly proposed explainable AI method, called the *NER Boosting Explainer*, with existing post hoc input attribution explanation methods.

We employ both hierarchical transformers and open-source Large Language Models (LLMs). Hierarchical transformers enable us to manage legal documents that surpass the token limits of BERT-based models [132]. Additionally, we also study Parameter-efficient fine-tuning (PEFT) [26] to tailor LLMs to the legal domain and the specific tasks at hand [185, 1]. Our empirical findings indicate that entity-aware methods significantly outperform in both prediction and explanation tasks.

5.2.1 Problem statement

The Court Judgment Prediction and Explanation (CJPE) task [1] encompasses two objectives: Court Judgment Prediction (CJP) and Prediction Explanation (PE).

Court Judgment Prediction. Given a legal judgment $X_i = \{x_{i,1}, x_{i,2}, ..., x_{i,N}\}$, where N represents the number of sentences in the judgment, CJP seeks to estimate a function f that predicts the court's decision, y_i. The decision can be *allowed*, indicating a ruling in favor of the appellant or petitioner, or *dismissed*, a decision in favor of the respondent.

Prediction Explanation. Given a legal judgment prediction y_i generated by a CJP model, PE provides an explanation for y_i. This explanation consists of a subset of K sentences $X_K \subseteq X_i$, where $K \leq N$, that support the decision made by the $\hat{f}$ model. Explanations are evaluated by comparing them to human-generated sentence selections.

5.2.2 Methodology

In [182, 183] we introduced a natural language processing pipeline designed to tackle both CJP and PE. An example of the pipeline execution is illustrated in Figure 5.1.

The initial step of the pipeline involves Legal Named Entity Recognition (NER) masking. This step serves two primary purposes: identifying relevant legal entities and masking information that is not pertinent to the legal context. Specifically, legal entities in the input text are replaced with meaningful tags that preserve essential information (e.g., [COURT] instead of *the High Court of Judicature for Rajasthan Jaipur in Criminal Appeal*).

The annotated text is then segmented into sentences to facilitate the generation of sentence-level explanations.

In the second step, a hierarchical transformer-based model is utilized to predict the court's decision (e.g., *Allowed*).

The final step involves generating prediction explanations. During this process, sentences containing relevant legal entities (e.g., [CASE_NUMBER] is preferred by [PETITIONER]...) are given priority to enhance the quality of the explanations.

In the following paragraphs we will detail these steps.

NER Masking The objectives of NER masking are twofold: (1) To focus on domain-specific entity-related information during the inference process, thereby enhancing the transformer's ability to differentiate between court judgments; (2) To obscure detailed descriptions of entities in the raw data, thereby removing redundant or potentially sensitive information (e.g. the appellant name).

Initially, we fine-tune the NER model provided by [185] using the Indian court judgment dataset. This model is specifically adapted to recognize a subset of legal domain entities, which are detailed in Section 5.2.3.

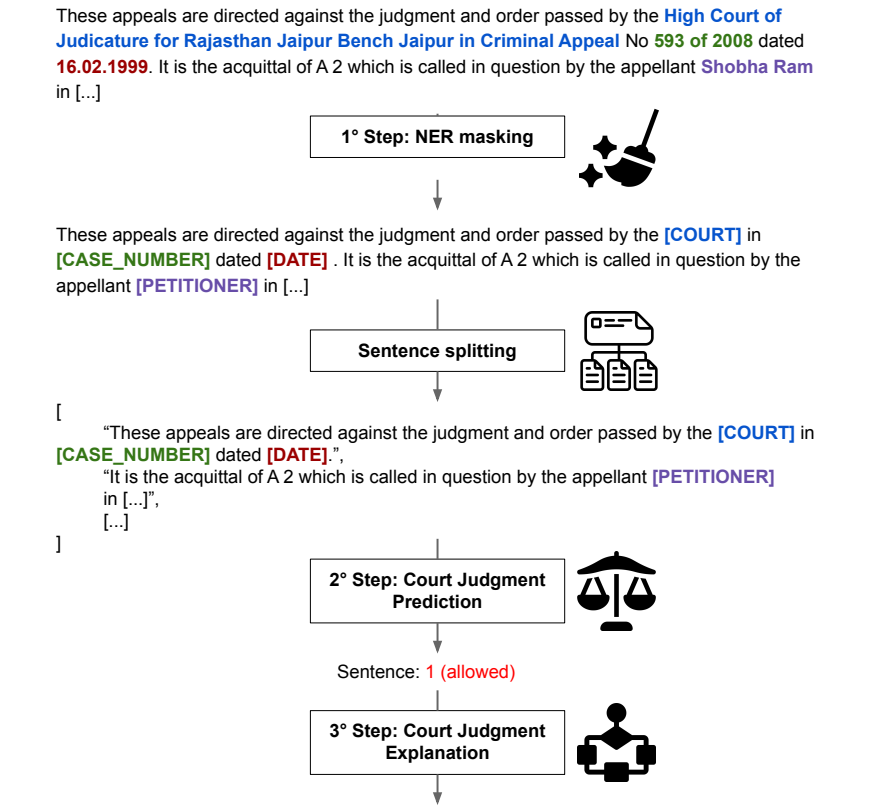


Fig. 5.1 Pipeline describing the proposed methodology.

To generalize the information in the input text, we replace the spans of text associated with recognized entities with designated tags, referred to as `ENTITY_TYPE`. Each tag corresponds to a specific type of entity identified by the NER model. These masked entity tags are then treated as individual tokens and incorporated into the vocabulary of the sentence tokenizer.

Court Judgment Prediction. The Court Judgment Prediction (CJP) module comprises two main components: a sentence encoder and a hierarchical model.

The sentence encoder processes each sentence from the original document and produces a vector representation. It utilizes two advanced transformer-based models: the general-purpose `RoBERTa-l` (large) and the domain-specific `LegalBERT`. These

models have recently been identified as the top-performing sentence encoders for the CJP task [186].

Both pretrained models are fine-tuned specifically for the CJP task. However, a limitation of using transformer-based models is their maximum length constraint, which restricts the processing of lengthy legal documents such as actual court cases. To address this issue, we incorporate a hierarchical component that aggregates the sentence vector representations to produce the final prediction. It consists of several attention layers and linear layers. Sentence embeddings are combined through average pooling, and the resulting document representation is fed into the classification layer to generate the final prediction.

For the sake of comparison, we also investigate the use of Large Language Models (LLMs) for CJP. Specifically, we fine-tuned three open-source LLMs: Llama 2 7B [67], Mistral 7B v0.1 [187], and Zephyr 7B β [188].

Court Judgment Prediction Explanation and our NER Boosting Explainer. We utilize post-hoc input attribution methods (model-agnostic). These methods analyze a model's prediction for a given target class to estimate the contribution of each input token to that outcome. In our scenario, however, the model inputs are sentence embeddings rather than individual tokens. Therefore, we apply input attribution methods at the sentence level to determine which sentences are most relevant to the prediction. This approach not only simplifies the explanation process but also reduces computational costs, which would be significantly higher if performed at the token level for long documents.

Standard post-hoc attribution methods typically depend solely on the input documents and the prediction model, often ignoring external knowledge such as domain-specific entities that legal experts might consider relevant. To address this limitation, we propose incorporating entity information into the generation of prediction explanations. The core idea is that sentences featuring legal entities are more likely to provide meaningful explanations for the predicted outcome.

Our approach, namely *NER Boosting Explainer*, is composed of the following steps:

1. We compute a score, named the *NER score*, to each sentence based on the proportion of legal entities present in its token representation. Sentences

with a higher concentration of legal entities are considered more relevant for explaining judgment predictions.

2. We enhance the explanations provided by the post-hoc attribution methods using the NER scores. For a given sentence x_i, let $e(x_i)$ denote the attribution score from the method, $n(x_i)$ represent the NER score, and $\beta \in \mathbb{R}_{\geq 0}$ be a boosting parameter. The boosted sentence score is calculated as $e(x_i)(1 + \beta n(x_i))$. A higher β value increases the emphasis on sentences with frequent legal entities.

3. We construct the explanation by selecting the top-k sentences, ranked according to their boosted sentence scores, in descending order.

5.2.3 Experimental design

Legal NER We utilize the named entity dataset provided by [185, 1]. This dataset includes annotations for two tasks of the Indian legal system. The annotations include the identification and the classification of legal entities, as well as explanatory elements pertinent to understanding legal judgments.

The dataset for the L-NER task includes 14,444 Indian court judgments and 2,126 judgment preambles, annotated with 14 types of legal named entities. Indian court judgments are divided into two sections: the preamble, which contains metadata such as names, dates, and court details, and the judgment itself, which includes the substantive text. The following list outlines the selected named entities.

- `COURT`: Name of the court which has delivered the current judgment;
- `PETITIONER`: Name of the petitioners/appellants/revisionist from the current case;
- `RESPONDENT`: Name of the respondents/defendants/opposition from the current case;
- `JUDGE`: Name of the judges from the current case;
- `LAWYER`: Name of the lawyers from both the parties;
- `DATE`: Any date mentioned in the judgment;

- `ORG`: Name of organizations mentioned in text apart from the court;
- `GPE`: Geopolitical locations which include names of states, cities, villages;
- `STATUTE`: Name of the act or law mentioned in the judgment;
- `PROVISION`: Sections, sub-sections, articles, orders, rules under a statute;
- `PRECEDENT`: All the past court cases referred to in the judgment as precedent;
- `CASE NUMBER`: All the other case numbers mentioned in the judgment (apart from precedent);
- `WITNESS`: Name of witnesses in current judgment;
- `OTHER PERSON`: Name of all the persons that are not included in petitioner, respondent, judge, and witness.

For L-NER masking, we consider all entity types except for `PROVISION` and `STATUTE`.

CJP and PE Our focus is on the Indian Legal Documents Corpus (ILDC) [1], which includes case proceedings from the Supreme Court of India in English, spanning from 1947 to 2020.

The ILDC documents are categorized into two main types: single petitions (7,593 documents) and multiple petitions (34,816 documents).

Each document is labeled with the original decision made by the Supreme Court of India, which serves as the ground truth. Cases can be either accepted or rejected based on various factors such as case facts, lower court rulings, arguments, statutes, and precedents.

In addition to the decisions, a subset of 56 test documents has been annotated with explanations by five legal experts. These annotations, ranked by importance, provide the ground truth for prediction explanations and are referred to as the *explain* set. Note that these explanation annotations are only available for the test set.

To highlight the importance of Named Entity Recognition (NER) in court judgment prediction, we analyzed the frequency of entities in both the documents and

Legal named entity	f_{REL}
PROVISION	0.980
PRECEDENT	0.980
STATUTE	0.961
ORG	0.424
GPE	0.294
OTHER_PERSON	0.272
RESPONDENT	0.255
COURT	0.238
DATE	0.236
WITNESS	0.140
JUDGE	0.110
CASE_NUMBER	0.108
PETITIONER	0.079
LAWYER	0.027

Table 5.1 Frequency count ratio of the analyzed entities.

the ground-truth explanations. Specifically, Table 5.1 presents the *frequency count ratio* for each entity, defined as the ratio of the number of occurrences of the entity in the explanation gold standard to its occurrences in the input documents. The results show that many entities frequently appear in both the documents and the explanation annotations, underscoring their significant role in model explainability.

Metrics The evaluation of Named Entity Recognition (NER) models involves calculating both strict and type-match F1 scores based on the combined preamble and judgment sentences. The *strict* F1 score measures the exact match of both entity boundaries and types. Conversely, the *type-match* F1 score evaluates the overlap between predicted and actual entities, considering their types. This latter metric reflects how closely the model's predictions align with the ground truth.

In line with [1], the performance of classifiers on binary court judgment prediction tasks is assessed using macro Precision, Recall, and F1 score metrics.

Explanation evaluation employs established metrics of plausibility and faithfulness.

Plausibility assesses how well the explanations align with human reasoning [189]. We measure the alignment of generated explanations with gold annotations using

several quality metrics: ROUGE-1, ROUGE-2, ROUGE-L [114], BLEU [190], METEOR [191], Jaccard Similarity, Overlap Maximum, and Overlap Minimum [1].

Faithfulness evaluates how accurately the explanation reflects the model's reasoning process. It is measured using two complementary metrics: comprehensiveness and sufficiency.

Comprehensiveness indicates whether the explanation includes the sentences that the model actually used to make its prediction [189]. We measure comprehensiveness by removing the sentences in the explanation and observing the change in the prediction probability. Specifically, let $x_{i,1}, x_{i,2}, ..., x_{i,N}$ be the sentences of the legal judgment X_i and $f(X_i)_{y_i}$ be the probability of the CJP model f for the prediction y_i. Let $X_K \subseteq X_i$ be the explanation, with $x_k \in \{x_{i,1}, x_{i,2}, ..., x_{i,N}\}$ for all $x_k \in X_K$. Comprehensiveness is defined as $f(X_i)_{y_i} - f(X_i \setminus X_K)_{y_i}$, where $X_i \setminus X_K$ represents X_i with the sentences $x_k \in X_K$ removed. A high comprehensiveness value indicates that the sentences in X_K are important for the prediction.

Sufficiency measures whether the sentences in the explanation are sufficient for the model to make the prediction [189]. It is quantified as the change in prediction probability when only the sentences in the explanation are used as input, i.e., $f(X_i)_{y_i} - f(X_K)_{y_i}$. A low score suggests that the sentences in the explanation are crucial for driving the prediction.

Models We experimented with the recently proposed Large Language Models and the most established transformer-based models. The motivations behind our choice are summarized below: As open access pre-trained Large Language Models we consider Llama 2 7B [67], Mistral 7B v0.1 [187], and Zephyr 7B β [188] for both court judgment prediction and explanation tasks.

As baseline methods for Legal NER we consider:

- LUKE [184]: We test both its base (`LUKE-b`) and large (`LUKE-l`) versions, with and without additional fine-tuning on the well-known generic named entities recognition dataset (CoNLL-2003 [192], Open entity [193], and TACRED [194]).
- BERT [68] and RoBERTa [195]: fine-tuned generically for the NER task in mono- and multi-lingual setting ($\texttt{BERT-b-multilingual}_{\texttt{FC}}$ and $\texttt{RoBERTa-b-multilingual}_{\texttt{FC}}$).

- A set of models specifically pre-trained on legal corpora (EURLEX, ECHR, and contracts) [132].

To address CJP we adopt `RoBERTa-l` and `LegalBERT`. The former model is larger and general-purpose, whereas the latter is smaller but pre-trained in the legal domain. For both model tokenizers we inject new tokens (e.g., `[ENTITY_NAME]`) corresponding to the NER tags used to mask the documents for part of the experiments. All pre-trained checkpoints of these models are obtained from the Hugging Face hub repository[1].

Large Language Models leverage Parameter-efficient fine-tuning (PEFT) [26] to adapt the model parameters to specific tasks with lower memory requirements. Specifically, we employ LoRA [27], which integrates trainable rank decomposition matrices into each Transformer layer. This approach significantly reduces the number of trainable parameters required for downstream tasks. To further decrease memory usage, we implement 8-bit quantization during training [97]. Additionally, we include the pre-trained model in a zero-shot setting as a baseline for comparison.

The competitor adopts a XLNet with BiGRU and an attention mechanism. For PE, it proposes an occlusion-based method that operates at the level of chunks of 512 tokens of the document. It estimates the relevance of a chunk as the differences in output probability between the masked chunk input and the unmasked one. We leverage and evaluate the following post-hoc feature attribution methods: Gradient [196] (also named as *Saliency*), Integrated Gradient [197], and leave-one-out (LLO) methods. At the implementation level, we employ the *ferret* [198] XAI library to generate and benchmark explanations for Transformers models. We extend the *ferret*'s Explainer APIs to support sentence inputs rather than token inputs.

For LLM explanation, we iterate over each sentence and we ask LLM to identify whether this sentence can be used as a explanation, i.e.:

```
You are provided with a **single sentence** taken from a legal
judgment.
The goal is to determine whether this sentence can serve as a valid
explanation for the final prediction of the judgment:
* **1 (allowed)** - the appeal, request, or claim was accepted.
```

[1] https://huggingface.co/models latest access: January 2024

```
* **0 (dismissed)** - the appeal, request, or claim was rejected.

A sentence is a valid explanation if it:
* Provides legal reasoning, justification, or causation tied to the
outcome.
* References legal principles, factual findings, procedural
reasoning, or statutes that support the final decision.
* Helps a reader understand *why* the case was allowed or dismissed.

### Instructions:
Given the sentence, output only one number:
 * ‘1‘: if the sentence provides a valid explanation for the final
 prediction of the judgment.
* ‘0‘ : if the sentence should be dismissed as not explanatory for
the final prediction.

### Input:
<legal_sentence></legal_sentence>

### Prediction:
<predicted_label></predicted_label>

### Output:
```

5.2.4 Results

Legal NER Results Table 5.2 presents a comparative analysis of the performance metrics for several models applied to the L-NER task on the validation dataset (performance metrics for the test set across the complete ILDC dataset are not publicly accessible [1]). The models are categorized according to their pre-training and fine-tuning approaches, distinguishing between those designed for general NER tasks and those specifically tailored for legal domain applications.

The results indicate that the LUKE-l-openentity$_{\text{FC}}$ model outperforms all other models evaluated. It achieves the highest scores across both evaluation metrics, with a Strict F1-score of 88.82% and a Type Match F1-score of 93.69%. This model

Model type	Model	Strict F1-Score	Type Match F1-Score
Generic pre-training	LUKE-b	86.42	92.31
	LUKE-l	88.59	93.38
Legal pre-training	BERT-s-legal$_{PT}$	82.61	90.07
	BERT-b-legal$_{PT}$	87.28	93.28
	BERT-b-contracts$_{PT}$	87.56	93.20
	BERT-b-echr$_{PT}$	86.66	92.90
	BERT-b-eurlex$_{PT}$	86.86	92.34
	RoBERTa-b-legal$_{PT}$	83.90	91.20
NER-generic fine-tuning	BERT-b-conll$_{FC}$	82.75	90.23
	BERT-l-conll$_{FC}$	84.30	91.13
	BERT-b-multilingual$_{FC}$	85.94	91.14
	LUKE-l-conll$_{FC}$	88.81	93.25
	LUKE-l-openentity$_{FC}$	**88.82**	**93.69**
	LUKE-l-tacred$_{FC}$	88.03	93.04
	MLUKE-l-conll$_{FC}$	67.52	78.37
	RoBERTa-b-multilingual$_{FC}$	54.19	69.97
	RoBERTa-l-conll$_{FC}$	<u>88.14</u>	<u>92.98</u>

Table 5.2 Comparison of various models based on their performance on validation data.

leverages the LUKE architecture and has been fine-tuned using the Open Entity dataset.

Based on the Type Match F1-score, the second highest-performing model is LUKE-l-conll$_{\text{FC}}$, another general-purpose fine-tuning model built on the LUKE architecture. This model achieves a Strict F1-score of 88.81% and a Type Match F1-score of 93.25%. The LUKE-l model, which has not undergone additional fine-tuning on a specific dataset, also demonstrates strong performance, with a Strict F1-score of 88.59% and a Type Match F1-score of 93.38%.

General-purpose fine-tuning models exhibit a wide range of performance outcomes. For example, MLUKE-l-conll$_{\text{FC}}$ performs the poorest among all models, with a Strict F1-score of only 67.52% and a Type Match F1-score of 78.37%. In contrast, BERT-b-multilingual$_{\text{FC}}$ shows significantly better performance, achieving a Strict F1-score of 85.94% and a Type Match F1-score of 91.14%.

When evaluating legal-specific models, it is evident that those pre-trained on legal data tend to underperform compared to their general-purpose counterparts on this particular dataset. Among the legal pre-training models, BERT-b-contracts$_{\text{PT}}$ achieves the best performance, slightly surpassing BERT-b-legal$_{\text{PT}}$.

The overall results indicate that the model architecture and fine-tuning strategy relevantly affect the performance of the models compared to the choice of pre-training. Despite the legal-specific models being pre-trained specifically for legal text, they generally exhibit worse performance than the LUKE models, which are fine-tuned on generic datasets. This indicates that the fine-tuning process, where the models are adapted to the specific L-NER task, has a stronger impact on performance than the choice of pre-training data. To comprehensively investigate the influence of named entity recognition masking on the subsequent stage of the proposed pipeline while effectively managing the experiment complexity, we use two specific NER models, namely LUKE-l-openentity$_{\text{FC}}$ and RoBERTa-l-conll$_{\text{FC}}$. These models were chosen based on their promising performance in terms of type match metrics, ensuring a thorough analysis of the impact of NER masking without overwhelming the experimental setup.

Classification Model	NER Model for masking	Dev Set Precision	Recall	F1 Score
BERT-b-legal$_{PT}$	-	76.21	75.86	75.77
BERT-b-legal$_{PT}$	RoBERTa-l-conll$_{FC}$	76.10	75.75	75.67
BERT-b-legal$_{PT}$	LUKE-l-openentity$_{FC}$	74.94	74.65	74.57
RoBERTa-l	-	**80.69**	**79.38**	**79.15**
RoBERTa-l	RoBERTa-l-conll$_{FC}$	76.60	76.26	76.18
RoBERTa-l	LUKE-l-openentity$_{FC}$	77.27	76.38	76.16

Table 5.3 Results for the CJP model obtained by applying a textual encoder to the document tails. RoBERTa-l model reaches the highest performance without the NER masking approach.

		w/o masking		w/ masking	
Attention Layers	MLP Layers	Dev Set	Test Set	Dev Set	Test Set
2	2	79.40	75.83	**80.58**	78.15
2	4	**80.41**	76.89	**80.58**	77.12
2	8	79.06	75.69	79.07	76.10
4	2	78.22	74.67	79.53	75.74
4	4	80.06	74.90	80.33	**78.47**
4	8	65.60	63.61	80.17	77.35
8	2	78.97	74.58	78.82	76.45
8	4	80.01	76.81	79.86	77.80
8	8	78.77	**77.02**	79.77	78.27

Table 5.4 CJP results obtained applying a hierarchical approach.

CJP Results Table 5.3 presents the results of the CJP model of the initial component of the CJP step. Predictions are generated by applying transformer-based sentence encoders—namely, RoBERTa and LegalBERT—to document tails. Previous research indicates that document tails are particularly advantageous for prediction tasks within the legal domain [1, 186].

Before utilizing the sentence encoders, the document tails undergo NER masking using two distinct NER models: RoBERTa (RoBERTa-l-conll$_{FC}$) and LUKE (LUKE-l-openentity$_{FC}$), the latter being the top performer for the L-NER task. Both the training and evaluation phases were conducted using the complete dataset, which includes both single and multiple petition documents.

Notably, model size exerts a greater influence on performance than in-domain legal pre-training. For instance, the large version of RoBERTa (`RoBERTa-l`) outperforms the smaller LegalBERT (`BERT-b-legal`$_{\texttt{PT}}$), despite the latter's advantage from legal domain pre-training.

We also investigate the impact of the NER masking operation (done with the `RoBERTa-l` model for both encoding and NER masking of document sentences) by repeating the experiments on the raw document tails. With non-hierarchical models, the results obtained using a masked text are slightly worse than those reached using the original document for the prediction task. Such a behavior can be attributed to the presence of a few entities within document tails that seemingly do not contribute significantly to the prediction task.

In the following discussion, the model that employs NER-masked sentences will be referred to as H-NER-MASKED, whereas the model that utilizes raw document sentences will be designated as H-UNMASKED.

Training	**Model**	**Precision**	**Recall**	**F1-score**
Zero-shot	Llama 2 7B	50.31	50.11	40.60
Zero-shot	Llama 2 7B-MASKED	49.97	49.99	41.32
Zero-shot	Mistral 7B	52.29	51.62	47.88
Zero-shot	Mistral 7B-MASKED	51.99	51.58	48.96
Zero-shot	Zephyr 7B β	50.76	50.74	50.27
Zero-shot	Zephyr 7B β-MASKED	52.40	52.26	51.56
Fine-tuning	Llama 2 7B β	51.96	50.93	43.37
Fine-tuning	Llama 2 7B β-MASKED	55.27	52.16	43.98
Fine-tuning	Zephyr 7B β	52.33	52.32	52.27
Fine-tuning	Zephyr 7B β-MASKED	**55.86**	**55.73**	**55.48**

Table 5.5 CJP results achieved by the LLMs on the test set. Model versions' with name suffix -MASKED integrate entity masking.

The results presented in Table 5.4 validate the efficacy of the hierarchical approach for processing long legal documents, achieving an F1 score of 80.41% on original texts and 80.58% on masked texts. These scores represent improvements of 1.26% and 4.40%, respectively, over their corresponding non-hierarchical methods. Unlike predictions based solely on document tails, the model trained with NER-masked sentences demonstrates superior performance on test data, with an F1 score of 78.47% compared to 77.00% for the unmasked text. This performance

enhancement is also observable in the test set, where the model's results are slightly lower than those on the validation set. These findings indicate that the removal of sensitive information, such as defendants' names, can be effectively managed without compromising system performance.

Table 5.6 provides a comparative analysis of performance between competitor models—employing XLNet with BiGRU and attention mechanisms—trained on both single-petition and multi-petition documents, and the top-performing H-NER-MASKED.

Model	Precision	Recall	F1-score
XLNet + BiGRU + att. (competitor multi)	77.32	76.82	77.07
XLNet + BiGRU + att. (competitor single)	75.26	75.22	75.25
H-NER-MASKED	**78.49**	**78.47**	**78.47**

Table 5.6 Performance of the best model (hierarchical approach with four attention layers and MLP layers) compared to competitors' models trained on *single* and *multi* datasets.

Competitor models are evaluated based on their training on either single-petition or multi-petition documents separately. In contrast, our approach utilizes a unified model trained on the entire dataset, which includes both single-petition and multi-petition documents.

Our model, featuring a hierarchical approach with four attention layers and MLP layers, demonstrates superior prediction accuracy compared to all tested competitors, including those based on Large Language Models (refer to Table 5.5). This enhanced performance can be attributed to two key factors: (1) the complexity of legal reasoning tasks for pre-trained Large Language Models (LLMs) [175], and (2) the length of legal documents, which frequently exceeds the maximum capacity of these models.

In general, all approaches utilizing entity masking outperform their respective versions that do not incorporate entity masking.

CJPE Results We conducted judgment prediction on the validation dataset utilizing the H-NER-MASKED model, followed by an evaluation of the explanation quality for the predictions. It is noteworthy that the model employs masking based on the large variant of RoBERTa (`RoBERTa-l`) and is structured with two attention

CJP model	explainer	ROUGE 1	ROUGE 2	ROUGE L	Jaccard
	random	30.35	19.80	33.98	24.37
[1]	OCCL	45.07	29.70	42.42	32.39
H-NER-MASKED	LOO	61.43	46.54	59.19	47.89
	GxI	61.62	47.01	59.51	48.17
	G	62.45	48.30	60.61	49.12
H-UNMASKED	LOO	58.33	43.17	56.02	44.54
	GxI	60.58	45.53	58.17	46.88
	G	62.24	47.76	60.2	48.52
H-NER-MASKED	G+NER	62.46	48.31	60.62	49.10
H-UNMASKED	LOO+NER	58.36	43.20	56.03	44.55
	GxI+NER	60.74	45.72	58.33	47.04
	G+NER	**62.82**	**48.39**	**60.68**	**49.16**

CJP model	explainer	Overlap min	Overlap max	BLEU	METEOR
	random	68.66	27.97	9.38	16.35
[1]	OCCL	71.85	38.31	17.60	22.29
H-NER-MASKED	LOO	76.84	56.05	42.74	38.98
	GxI	77.21	56.33	42.71	38.59
	G	**77.95**	57.14	44.34	39.91
H-UNMASKED	LOO	74.75	52.83	38.22	35.64
	GxI	76.06	55.21	40.97	37.18
	G	77.31	56.75	43.62	39.18
H-NER-MASKED	G+NER	77.94	57.12	44.34	39.90
H-UNMASKED	LOO+NER	74.75	52.84	38.28	35.66
	GxI+NER	76.15	55.38	41.26	37.39
	G+NER	77.57	**57.38**	**44.99**	**40.20**

Table 5.7 Explanation plausibility of LOO (Leave-one-out), GradientXInput (GxI), Gradient (G), and gradient β-boosted via NER (G+NER-β) considering 40% of the sentences as explanation.

layers and four MLP layers. Additionally, we examined the explanation quality of the highest-performing H-NER-MASKED model, comparing it to the masked version. This optimal model similarly consists of two attention layers and four MLP layers.

We applied the two models to a separate *explain* dataset consisting of 50 documents, for which gold annotations provided by five legal experts are available. The H-UNMASKED and H-NER-MASKED models achieved F1 scores of 0.71 and 0.73, respectively. Subsequently, we evaluated the quality of the explanations generated by the leave-one-out (LOO), gradient (G), and gradient × input (GxI) methods for both models, as well as those enhanced with named entity recognition (Explainer-NER-β). Tables 5.7 and 5.9 present the results for plausibility and faithfulness, respectively.

Comparison with the baselines. We initially assess the plausibility of the proposed explanations by comparing them against established baselines. In particular, for all approaches, 40% of the sentences are designated as explanations, following the methodology described in [1]. Two baseline methods are utilized for comparison. The first baseline, serving as a sanity check, is a random model where explanations are generated by randomly selecting 40% of the sentences. The second baseline involves occlusion-based explanations generated for the XLNet+BiGRU model as proposed in [1].

For the H-NER-MASKED model, all explanation methods outperform both the random baseline and the occlusion-based explanations (refer to Table 5.7). Our approach generates more plausible explanations that align more closely with human-provided explanations.

The LOO explanations generated by our method and the occlusion-based explanations from [1] exhibit notable similarities, as both techniques rely on omitting individual inputs and measuring the change in the CJP model's output probability. The primary difference lies in the portion of the input that is occluded: our method occludes sentence embeddings, while [1] occludes chunk embeddings. Despite this distinction, our approach yields higher quality results across all plausibility metrics.

We attribute this improvement to two main factors. First, the CJP models we present achieve superior overall performance, which suggests that they have better learned relevant associations between the input document and the final decision, thereby producing explanations that are more aligned with human reasoning. Second,

the finer granularity of sentence-based occlusion appears to be more effective than using chunk embeddings, which cover multiple sentences, leading to more precise and plausible explanations.

Impact of NER masking. NER masking leads to the highest CJP performance, with the H-NER-MASKED outperforming the H-UNMASKEDẆe now assess whether explanations derived from NER-masking also exhibit a better alignment with experts' explanations. The ground truth explanations correspond to original sentences within the judgment. To ensure a fair comparison, the masked sentences selected for explanation are remapped to their original versions.

The plausibility metrics, both without and with masking, are reported in the 3rd and 4th blocks of Table 5.7. Across all metrics, each explanation method achieves higher results when H-NER-MASKED is utilized. This indicates that NER masking not only enhances prediction accuracy but also improves the plausibility of the explanations. These findings empirically demonstrate the increased robustness of the proposed models, which prioritize relevant facts and causes over the identities of involved parties, regulations, or precedent cases when predicting outcomes.

Impact of NER boosting. We now turn to the evaluation of whether enhancing explanations via NER tagging improves their overall quality. NER tagging boosting amplifies the significance of sentences that contain entities, based on the premise that such entities represent crucial information that experts typically use to draw conclusions from legal documents, thereby aligning with human rationales.

The boosting parameter β is set to 7, as this value provides an optimal balance in the proportion of sentences affected by the boosting. A detailed analysis of the β hyperparameter tuning, along with its impact on the quality of explanations, is presented later in this section.

We first assess the impact of NER boosting on the H-NER-MASKED. NER boosting is applied across the three explanation methods; however, only the results for the gradient (G) method are presented in the 5th block of Table 5.7, as the boosted explanations for the LOO and GxI methods are identical to their original versions. The gradient explanation method shows a slight improvement with the adoption of NER boosting. This outcome suggests that the H-NER-MASKED already effectively encodes the relevance of entities due to the NER masking, thereby diminishing the additional benefits of boosting. Next, we evaluate the effect of boosting on

the H-UNMASKED, with results reported in the final block of Table 5.7. In this case, NER boosting proves highly beneficial, with all explanation methods showing improved quality metrics when boosting is applied. The NER-boosted gradient (G) explanations for the H-UNMASKED achieve the highest scores across all evaluated configurations, though the differences with the H-NER-MASKED remain marginal. These findings indicate that NER boosting can be effectively used to introduce post-hoc information after training, enhancing the quality of explanations. Moreover, this technique can be applied to masked texts with only a negligible impact on performance.

Plausibility separately for experts. To this point, we have evaluated the average plausibility results across five experts. We now analyze the variability in results among individual experts. Focusing on the best configuration for the H-NER-MASKED, specifically the NER-boosted gradient (G) explanations, Table 5.8 presents the plausibility metrics for each expert separately, alongside the occlusion-based explanations for the XLNet + BiGRU + attention method.

Our G+NER model achieves the highest performance. However, the results exhibit variability across different experts. This variation highlights that legal experts may identify different parts of a document as relevant to their judgments, influenced by their individual experience and expertise. These discrepancies underscore the importance of providing explanations that can assist legal experts in validating and trusting model decisions based on their own professional judgment.

Faithfulness of explanations. Table 5.9 provides the metrics for comprehensiveness and sufficiency, representing the faithfulness of the explanations. Consistent with the plausibility analysis, the results for NER-boosted Leave-One-Out (LOO) and Gradient × Input (GxI) explanations for H-NER-MASKED are not presented, as they are identical to the non-boosted explanations.

Comprehensiveness measures the extent to which explanations encompass the relevant inputs for the model. The Gradient × Input (GxI) method yields the highest comprehensiveness scores for H-NER-MASKED among the evaluated methods, followed by the Gradient (G) method, in both its original and boosted forms.

Sufficiency evaluates how well the explanations alone can account for the model's predictions, with lower sufficiency values indicating better performance. The Gradi-

	Method and Explainer	User 1	User 2	User 3	User 4	User 5	avg
ROUGE 1	ours - G+NER	**65.22**	**60.57**	**62.98**	**62.49**	**61.03**	**62.46**
	[1] - OCCL	44.40	51.67	40.14	39.07	50.08	45.07
ROUGE 2	ours - G+NER	**51.70**	**42.71**	**51.87**	**53.40**	**41.85**	**48.31**
	[1] - OCCL	30.33	29.53	29.65	29.65	29.37	29.70
ROUGE L	ours - G+NER	**63.80**	**54.97**	**63.25**	**66.08**	**54.99**	**60.62**
	[1] - OCCL	43.92	40.72	42.31	44.45	40.72	42.42
Jaccard	ours - G+NER	**53.16**	**43.58**	**52.43**	**53.23**	**43.10**	**49.10**
	[1] - OCCL	33.27	31.70	32.79	32.41	31.77	32.39
Overlap min	ours - G+NER	**79.42**	**71.43**	**83.67**	**86.86**	**68.31**	**77.94**
	[1] - OCCL	74.41	58.87	80.93	83.37	61.69	71.85
Overlap max	ours - G+NER	**62.01**	**53.22**	**58.46**	**57.90**	**54.01**	**57.12**
	[1] - OCCL	38.97	41.42	36.00	35.11	40.06	38.31
BLEU	ours - G+NER	**48.85**	**46.20**	**41.11**	**39.23**	**46.31**	**44.34**
	[1] - OCCL	15.97	27.95	9.92	9.31	24.83	17.60
METEOR	ours - G+NER	**39.94**	**47.33**	**33.97**	**33.48**	**44.79**	**39.90**
	[1] - OCCL	21.10	29.18	17.23	16.91	27.03	22.29

Table 5.8 The quality of explanations is evaluated separately for each user using the gradient method with NER boosting ($\beta = 7$), denoted as G+NER) for the H-NER-MASKED and compared with the occlusion-based method applied to the XLNet + BiGRU + attention model from [1]. In both methods, 40% of the sentences are used as explanations.

Masking	Boosting	Explainer	Comprenhensiveness (↑)	Sufficiency (↓)
✗	-	random	0.126	0.295
✓	-	random	0.113	0.254
✗	✗	LOO	-0.059	0.578
		G	0.211	0.122
		GxI	0.211	0.122
✓	✗	LOO	-0.033	0.470
		G	<u>0.237</u>	**0.034**
		GxI	**0.295**	<u>0.046</u>
✗	✓	LOO	-0.058	0.578
		G	0.218	0.114
		GxI	0.213	0.120
✓	✓	G	<u>0.237</u>	**0.034**

Table 5.9 Faithfulness of explanations for the Leave-One-Out (LOO), Gradient × Input (GxI), and Gradient (G) methods, with and without NER boosting ($\beta = 7$). Explanations are based on 40% of the sentences. The caption highlights the best-performing method in terms of faithfulness.

ent (G) method, both in its original and boosted versions, achieves the most favorable sufficiency results for H-NER-MASKED.

The results indicate that the masked judgments yield improved comprehensiveness and sufficiency metrics compared to the unmasked version. Specifically, improvements of up to +0.084 in comprehensiveness and -0.088 in sufficiency are observed for the Gradient × Input (GxI) and Gradient (G) methods. The masking of entities not only helps to maintain privacy but also enhances the faithfulness of the explanations.

LLMs explanations. We perform a separate analysis of the interpretability offered by Large Language Models (LLMs), using the optimal predictive model as input. Explanations, alongside predictions, are generated following the protocol outlined in [199]. Table 5.10 presents plausibility metrics collected from five legal experts, as well as their average assessment.

Compared to the results reported in Table 5.8—which include transformer-based explanation methods—the LLM-generated explanations consistently score lower on plausibility. This discrepancy likely stems from the nature of the evaluation setting: while transformer methods select from a fixed pool of candidate sentences (typically aligned with the gold standard), the LLM explanations are dynamically generated, even when prompted to select rather than generate text.

Although we instructed the LLM to select approximately 40% of the most contributive sentences, minor output variations are difficult to eliminate entirely due to the generative nature of LLMs. Moreover, current evaluation metrics—designed around comparison to a gold standard—may be less suited for free-form or flexible outputs. This mismatch between dynamic generation and static evaluation likely contributes to the observed performance gap.

To mitigate this, several strategies could be explored. Improved prompting techniques, such as few-shot learning with demonstrations or the use of more explicit selection constraints (e.g., via in-context chain-of-thought selection tasks), might increase alignment with the gold standard. Additionally, alternative output formats, such as producing binary sentence-level justifications or explicitly ranking candidate sentences, could help guide the LLM toward more evaluation-compatible outputs. Lastly, adopting more flexible evaluation protocols, such as expert consistency checks or agreement with auxiliary model rationales (rather than a static gold set), could

	User 1	**User 2**	**User 3**	**User 4**	**User 5**	**avg**
ROUGE-1	25.38	35.24	21.66	20.58	29.69	26.51
ROUGE-2	16.65	19.54	15.42	14.90	15.07	16.32
ROUGE-L	30.17	32.54	27.70	27.54	27.18	29.03
Jaccard	20.96	23.85	19.02	17.86	19.88	20.31
Overmin	80.96	67.52	86.32	86.76	63.61	77.03
Overmax	22.52	27.47	19.86	18.66	22.97	22.30
BLEU	4.32	10.78	2.12	1.97	8.07	5.45
METEOR	12.01	17.82	9.25	8.82	15.14	12.61

Table 5.10 Evaluation of Zephyr 7B β (masked) explanation quality for each individual user.

offer a more faithful assessment of the explanatory value of dynamically generated text.

Impact of boosting parameter. We optimize the boosting hyperparameter, β on the validation set. The parameter β is varied across a range from 0 to 30, and its impact on the explanations generated for the validation set is systematically analyzed.

As β increases, the relevance of sentences with NER taggings also increases. Figure 5.2 shows that a higher β leads to a greater proportion of documents with explanations influenced by NER boosting. The impact varies by masking modality and explanation method. Notably, Gradient (G) explanations without masking are most affected by boosting, with up to 85% of documents showing an increase. We identify a knee point at $\beta \approx 7$, where approximately 60% of Gradient explanations, 17% of Gradient × Input (GxI) explanations, and 7% of Leave-One-Out (LOO) explanations are influenced. In contrast, explanations with masking exhibit minimal impact, with the maximum effect reaching only 2%. This limited effect is attributed to the fact that the NER-masked model already effectively encodes the relevant information regarding the presence of entities.

We investigate the impact of β on faithfulness using the validation set, as illustrated in Figure 5.3. The analysis excludes Leave-One-Out (LOO) explanations without masking due to their inferior performance compared to other methods, as detailed in Table 5.9. For the unmasked model, we observe an increase in comprehensiveness along with a decrease in sufficiency as β varies. In contrast, boosting has a minimal effect on the masked model's explanations, likely because only a small proportion of documents are affected by the boost.

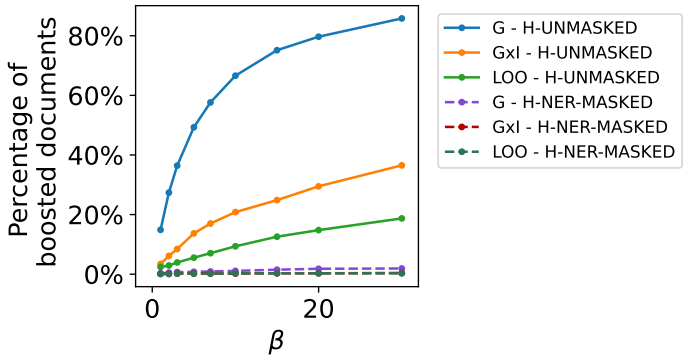


Fig. 5.2 Percentage of explanations affected by boosting as the degree of boosting β varies, based on the validation set.

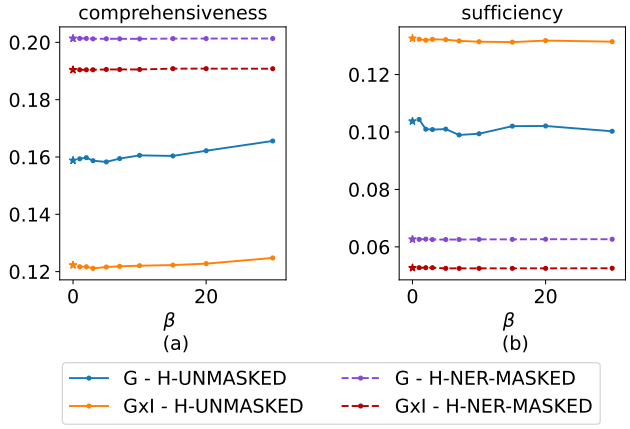


Fig. 5.3 Faithfulness results as the degree of boosting β varies on the validation set. Results with no boosting applied ($\beta = 0$) are marked with a star.

These results highlight the significance of including entity information for generating faithful explanations. Specifically, explanations from the NER-masked model generally demonstrate higher quality. Meanwhile, for models that do not utilize NER, applying NER boosting enhances both the quality and faithfulness of the explanations.

Given the minimal impact on faithfulness, we select the β value based on the percentage of documents affected by boosting. Specifically, we set β to 7, as this provides an optimal balance between the extent of boosting and its impact. As detailed in Table 5.7, NER boosting significantly enhances the plausibility of the results, particularly for the unmasked model.

In these experiments, tuning the β value for plausibility measures is not feasible due to the lack of ground truth explanations for the validation set. However, an examination of the impact of β on plausibility in the explain set is presented in the Appendix.

5.3 Evaluating LLM performance on legal problem solving

There is an urgent need to advance and refine legal reasoning capabilities in Large Language Models (LLMs). However, practitioners encounter difficulties in evaluating these capabilities due to the limitations of existing legal benchmarks, which often fail to encompass the full range of legal tasks [175]. To address this challenge, the SemEval-2024 Task 5 organizers introduce a novel task and dataset of U.S. civil procedure domain [200, 201]. Each instance in this dataset includes a case introduction, a specific question, a proposed solution argument, and a detailed analysis justifying the argument's relevance to the case. The goal of the proposed task is to assess the accuracy of the given answer based on the provided case introduction, question, and potential answer.

To address this task, we initially reformulate the dataset as a multiple-choice question answering problem using the multiple-choice prompting (MCP) method [202]. We tested various open-source language models on this adapted dataset, including Flan T5 XXL [203, 94], LLama 7B and 13B [67], Zephyr 7B [67], and Mistral 7B [187]. These models were specifically trained to address legal problems and provide explanations for their predictions, utilizing the analyses provided in the dataset.

In this work we propose the *CLUEDO* approach, which stands for "**C**hoosing **L**egal o**U**tcome by **E**xplaining **D**ecisions through **O**versight".

This framework employs a set of collaborative models to aggregate the final outcome based on the predictions from each individual model. These models are trained to predict the label of the correct answer and to generate an explanation of this prediction. The final component, referred to as the *"detective"* model, operates in a zero-shot manner, utilizing the outputs from the collaborative models. This

model evaluates the answers and explanations provided by all collaborators and deduces the final answer.

In this work we investigate the following research questions (RQs):

- **RQ1.** Is the multiple-choice approach more effective compared to the single-choice approach?
- **RQ2.** Does incorporating the analysis into both the training and generation processes enhance performance?
- **RQ3.** Does our detective model, *CLUEDO*, demonstrate greater effectiveness than individual models in a zero-shot setting? Additionally, are the results from *CLUEDO* more stable?

Our results on the challenge dataset indicate that the proposed methodology outperforms individual models trained with standard fine-tuning. Moreover, our approach achieved second place in the public competition, with a final test F1 macro score of 0.77.

5.3.1 Dataset

[201] introduce a new dataset sourced from the U.S. civil procedure domain. This dataset, derived from a textbook designed for law students, is characterized by its complexity and relevance for evaluating modern legal language models. Each instance in the dataset comprises i) a *general introduction to the case*: provides an overview of the case to establish context. ii) a *specific question*: presents a particular legal query related to the case. iii) *proposed solution argument*: offers a potential answer to the posed question. iv) an *annotated label*: Indicates whether the proposed solution is correct (1) or incorrect (0). v) a *detailed analysis*: includes a comprehensive explanation for why the solution argument is applicable to the case. The task is formulated as a binary classification problem, where the objective is to predict the correctness of the provided answer based on the accompanying textual information. During the test phase, both the analysis and labels are not accessible.

Model	P	R	F1	A
Llama 2 7B	0.57	0.60	0.56	0.64
Mistral v0.1 7B	0.61	0.63	0.62	0.73
Zephyr beta 7B	0.62	0.65	0.63	0.73
Llama 2 13B	0.65	0.69	**0.66**	0.75

Table 5.11 Performance of trained models on the development set. All models are trained to produce both labels and analyses within the framework of a multiple-choice setting.

5.3.2 System Overview

This section offers a detailed overview of the proposed methodology. Initially, we describe our approach to transforming the question-answering problem into a multiple-choice format and how this adaptation is applied to our specific scenario. Subsequently, we present the *CLUEDO* framework, including an explanation of the various models incorporated into our study.

Multiple-choice. Building on the methodology proposed by [202], we reformulate the dataset as a multiple-choice question-answering task and employ multiple-choice prompting (MCP) [202]. In the MCP framework, the language model is presented with not only the question but also a set of candidate answers, similar to a traditional multiple-choice test. Each candidate answer is associated with a label, such as *"A,"* *"B,"* or *"C."* This setup allows the model to explicitly compare the available options and reduces computational costs associated with generation.

When only one candidate answer is available, the system generates an additional option labeled *"None of the above is true"*. These additional answers are excluded from the test and validation metrics. Our experiments aim to assess whether the multiple-choice approach offers a performance advantage over the single-choice method. In the single-choice scenario, the model is provided with a single answer and must predict its correctness directly.

CLUEDO. This approach entails the development and training of multiple collaborative models, each tasked with predicting the appropriate label for responses to legal queries. During the training process, each model provides a corresponding explanatory analysis. In the final stage, a zero-shot model consolidates the predic-

tions and justifications generated by these collaborative models to derive the most accurate final decision, leveraging the combined efficacy of the individual models' outputs. The structure of the *CLUEDO* system is as follows:

- *N Collaborative Models*: These models are trained to predict the label corresponding to the candidate answer that best addresses the legal question, while also generating a rationale for their selection. In our implementation, we employ three collaborative models, chosen based on their performance on the development set.
- *The Final "Detective" Model*: This model operates in a zero-shot manner, utilizing the outputs and justifications provided by the collaborative models to determine the most accurate final answer. It receives the same introduction, legal question, and candidate answers as the collaborative models and is tasked with synthesizing and assessing the inputs from the collaborators to arrive at the final decision.

Examples of prompts utilized for both the collaborative and detective models are provided in Table 5.12.

5.3.3 Experimental design

Models We conducted an evaluation of several open-source models, utilizing both zero-shot and fine-tuning techniques. The models analyzed include Flan T5 XXL [203, 94], LLama 7B and 13B [67], Zephyr 7B [67], and Mistral 7B [187], chosen based on their distinctive characteristics and performance metrics. Additionally, GPT-4 [176] was incorporated into our evaluation in a zero-shot setting.

We applied a Supervised Fine-Tuning (SFT) strategy, incorporating precision enhancement through 8-bit quantization. The models underwent training for three epochs using Parameter-Efficient Fine-Tuning (PEFT) [204], batch size of 4 and a learning rate of 5e-5. The sequences were processed with a context length of 4096, which optimizes the model's capability to capture long-range dependencies within the data.

Models	Example Prompt
Collaborative Models	<s>[INST] <<SYS>>Given the following explanation and the question, which of the candidate answers is correct? The correct answer is the one that is true according to the explanation. <</SYS>> <explanation>Although discovery usually extends to [...] </explanation> <question>Confidential chat. Shag [...] </question> <candidate_answers> 1 - Shag will not have to answer any of the interrogatories [...] </candidate_answers> [/INST] <correct_answer>
CLUEDO	<s>[INST] <<SYS>>You are a legal supervisor tasked with resolving legal queries. You are working alongside three artificial intelligence models, named m1, m2, and m3. Given an introductory context, a question, and a set of candidate answers, these three models must choose the correct answer and provide justification for their choice. Your responsibility is to assess the models' responses and determine whether they are correct or not. To do so, you must read the context (enclosed within the tags <context> </context>), the question (within <question></question>tags), and the candidate answers (within <candidate_answers> </candidate_answers>tags), and identify the correct answer among them (using the <supervisor_answer> tag). Additionally, you must provide reasoning for your choice (using the <supervisor_explanation>tag). While collaborating with the models and considering their advice, the ultimate decision rests with you. For each response, use the following format: <supervisor_answer>SUPERVISOR ANSWER</supervisor_answer> <supervisor_explanation>SUPERVISOR ANSWER</supervisor_explanation> <</SYS>> <context>Although discovery usually extends to [...] </context> <question>Confidential chat. Shag [...] </question> <candidate_answers> 1 - Shag will not have to answer any of the interrogatories [...] </candidate_answers> [/INST] </candidate_answers> <m1_answer>1</m1_answer> <m1_explanation>[...] </m1_explanation> <m2_answer>1</m2_answer> <m2_explanation>[...] </m2_explanation> <m3_answer>2</m3_answer> <m3_explanation>[...] </m3_explanation> <supervisor_answer>

Table 5.12 Example of prompts for collaborative models and our CLUEDO approach.

Model	Classification task	Prec	Rec	F1	Acc
Flan T5 XXL	Multiple choice	0.60	0.67	**0.59**	0.64
Flan T5 XXL	Single choice	0.54	0.53	0.32	0.32
GPT-4	Multiple choice	0.66	0.73	**0.66**	0.57
GPT-4	Single choice	0.40	0.50	0.44	0.80
Llama 2 13B	Multiple choice	0.64	0.58	**0.59**	0.79
Llama 2 13B	Single choice	0.55	0.58	0.54	0.61
Llama 2 7B	Multiple choice	0.51	0.51	0.51	0.74
Llama 2 7B	Single choice	0.53	0.52	**0.52**	0.73
Mistral v0.1 7B	Multiple choice	0.55	0.59	**0.54**	0.61
Mistral v0.1 7B	Single choice	0.55	0.58	0.52	0.57
Zephyr beta 7B	Multiple choice	0.54	0.56	**0.50**	0.69
Zephyr beta 7B	Single choice	0.40	0.50	0.44	0.80

Table 5.13 Zero-shot models on the development set. The highest performance within each model family, measured by F1 macro score, is highlighted in bold. Notably, the multiple-choice approach yields superior performance in five out of the six evaluated cases.

5.3.4 Results

We conduct individual tests for each configuration and present the obtained results on the development set, to show the efficacy of the multiple-choice setting and model selection criteria. The following paragraphs address the research questions previously presented.

RQ1: Impact of the multiple-choice setting. Table 5.13 presents the performance of zero-shot models on the development set. Within each model family, the multiple-choice question-answering approach consistently surpasses the single-choice method in terms of F1 Macro scores. The performance of models within the same family shows variability, with larger models generally demonstrating better generalization capabilities compared to their smaller counterparts.

RQ2: Impact of analysis inclusion in model training. Table 5.14 illustrates the effect of incorporating analysis into the training process of models. To evaluate the

Model	Classification task	Analysis included	Prec	Rec	F1	Acc
Llama 2 7B	Multiple choice	x	0.49	0.48	0.47	0.56
Llama 2 7B	Multiple choice	✓	0.57	0.60	**0.56**	0.64
Llama 2 7B	Single choice	x	0.40	0.50	0.44	0.80
Llama 2 7B	Single choice	✓	0.40	0.50	0.44	0.80
Llama 2 13B	Single choice	x	0.55	0.58	0.52	0.57
Llama 2 13B	Multiple choice	✓	0.65	0.69	**0.66**	0.75

Table 5.14 Trained models on the development set. The highest F1 Macro scores are highlighted in bold. For both the 7B and 13B models, incorporating the generation of the analysis results in superior performance.

impact across different model sizes and classification tasks, we maintained a fixed model family.

For both the 7B and 13B models, the generation of the analysis (✓) improves performance when adopting the multiple-choice setting. It balances precision and recall metrics, improving the overall F1 Macro score. Conversely, for both Llama 2 7B and Llama 2 13B, the F1 Macro scores for single-choice tasks do not exhibit significant improvement with the addition of analysis. This suggests that these models may be less responsive to the benefits of additional analysis in single-choice scenarios. Furthermore, training the Llama 2 13B model with analysis results in an additional $+0.07$ F1 score compared to its zero-shot counterpart. In contrast, the performance of the 7B models deteriorates with the inclusion of analysis during training.

RQ3: CLUEDO results. The selection of collaborative models is based on the performance metrics reported on the development set, as detailed in Table 5.11. All models are configured to produce both labels and analyses within the multiple-choice framework. Among the models evaluated, Llama 2 13B demonstrates the highest F1 Macro score, reflecting its superior performance across various evaluation metrics, followed by the Mistral and Zephyr models. For the supervisory model, GPT-4 is selected, as it shows the best performance in the zero-shot setting, as indicated in Table 5.13.

In Table 5.15 the results on the test set are presented. When the corrections based on the consensus of the second and third collaborators (Mistral and Zephyr)

Method	Dev		Test	
	F1	Acc	F1	Acc
Best collaborator	0.66 ($\leq$ 0.01)	0.75 ($\leq$ 0.01)	0.69 ($\leq$ 0.01)	0.75 ($\leq$ 0.01)
Collaborators agreement	0.65 ($\leq$ 0.01)	0.75 ($\leq$ 0.01)	0.65 ($\leq$ 0.01)	0.75 ($\leq$ 0.01)
Zero-shot detective	0.63 ($\pm$ 0.04)	0.71 ($\pm$ 0.02)	**0.77** ($\pm$ 0.02)	0.83 ($\pm$ 0.02)
CLUEDO	**0.74** ($\pm$ 0.01)	0.78 ($\pm$ 0.01)	**0.77** ($\pm$ 0.01)	0.82 ($\pm$ 0.01)

Table 5.15 Results on dev and test sets: collaborators within CLUEDO are trained to generate the analysis along with the labels and adopt the MCP approach.

are applied, the result slightly decreases to 0.65 on both the development and test sets, indicating that the initial predictions from the collaborator were already highly accurate.

The zero-shot model, GPT-4, achieves an F1 score of 0.63 on the development set. However, it outperforms all other methods on the test set, attaining an impressive F1 Macro score of 0.77, which highlights its exceptional generalization capabilities. The *CLUEDO* method demonstrates superior performance on the development set with the highest F1 Macro score of 0.74, and it achieves the second-highest score on the test set, reflecting its effective performance across both evaluation phases.

To evaluate the stability of predictions, we conducted five separate experiments on both the validation and test sets, assessing the models' performance. Despite employing a greedy decoding strategy, minor discrepancies in floating-point operations can result in divergent outputs, particularly for larger models [205]. This issue is notably relevant for GPT-4, as discussed in the OpenAI community [2]. Consequently, despite setting the temperature to 0 for all experiments, users frequently report significant variations in the model's output.

Although predictions from the trained models remained consistent, significant variations were observed in the predictions of GPT-4, especially when used without collaborators (with the temperature set to zero and no sampling). These results are

[2]An example of discussion on model variability: https://community.openai.com/t/why-the-api-output-is-inconsistent-even-after-the-temperature-is-set-to-0/329541, https://community.openai.com/t/run-same-query-many-times-different-results/140588

summarized in Table 5.15. The proposed CLUEDO approach notably reduces the standard deviation by 50%. Furthermore, the error estimates on the development set are consistent with those obtained on the test set. In conclusion, while CLUEDO may not achieve the highest performance on the test data, it provides enhanced stability in predictions.

5.4 An end-to-end pipeline for legal information retrieval and problem resolution

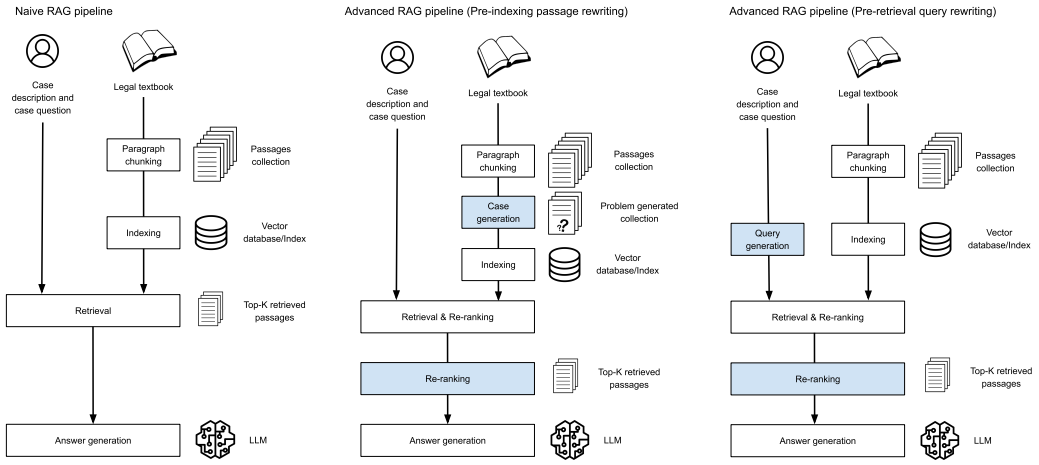


Fig. 5.4 RAG pipeline comparison: Naive approach (left) vs. proposed enhancements - case generation (center) and user input rewriting (right)

The aforementioned challenge oversimplifies the duties of a legal professional by reducing their role to merely identifying the correct response to a given legal case. In reality, resolving a legal case requires a more comprehensive approach. Firstly, the expert should retrieve the most pertinent information, such as locating and extracting the specific statutory provisions, precedents, or doctrinal explications that directly address the legal issue at hand. Secondly, the expert must conduct a thorough analysis of the retrieved materials to discern the most appropriate resolution, taking into account the nuances and complexities of the particular case.

To address this limitation, we propose a Retrieval Augmented Generation (RAG) pipeline to more accurately simulate the process of resolving legal cases. A generic RAG pipeline first *retrieves* the most relevant contextual information coming from

this dataset (the topic introduction), to provide the necessary knowledge base for addressing the legal issue. Subsequently, the pipeline selects the most appropriate resolution by analyzing and *generate* an answer using the retrieved context within the specific factual circumstances of the case.

If the user's query and the passages in the knowledge base have a semantic mismatch, it can lead to a query-passage misalignment [206]. This misalignment can negatively impact the performance of dense retrieval methods, as they rely on capturing semantic similarities between the query and the passages, therefore causing a subsequent performance drop in the generation step of a RAG.

To overcome this issue, in our RAG pipeline, we propose two alternative solutions in both retrieval and generation phases (see Figure 5.4). A *Problem generation* step to automatically generating fictitious legal cases from the original passages to improve the retrieval phase and better match the input sequence. A *query re-writing* phase, that automatically rewrite the user input and extract the most important terms for the retrieval phase. We evaluate the proposed approaches using the aforementioned legal reasoning dataset derived from U.S. civil procedure domain [200, 201]. We compare the performance of different retrieval methodologies and language models, including GPT-4, SAULM, and Llama-3 8B.

5.4.1 Problem formulation

We assume that we do not possess prior information regarding the relevance of the legal topic description associated to a specific legal issue (henceforth referred to as *passages*). Our objective is to construct a pipeline that, given a legal case inquiry, is capable of i) retrieving the most pertinent passage and ii) leveraging the retrieved passage to enhance the accuracy in identifying the correct answer with a Large Language Model. Therefore, given a legal case description and a question q_i, accompanied by its corresponding alternative answers $a_{i,1}, a_{i,2}, \ldots a_{i,n}$, the goal is to predict the correct answer $a_{i,j}$ and provide the most appropriate passage $p_i \in \mathscr{P}$ from the set of passages $\mathscr{P}$ (the legal textbook to explore) in a zero-shot fashion, i.e., without utilizing any training dataset. The rationale behind adopting the zero-shot setting stems from the fact that obtaining an annotated dataset of this nature is generally a costly endeavor.

5.4.2 Methodology

A Naive Retrieval-Augmented Generation (RAG) pipeline applied to legal problem resolution is presented and depicted in Figure 5.4.

- The retrieval module aims to retrieve the most pertinent passage with respect to the case description and the question provided by the user. Given a user input q_i, which in our case is the case description, the case question, the retriever computes the score $s(q_i, p_j)$ for each passage $p_j \in \mathscr{P}$. Subsequently, the retriever ranks each passage score and returns the top k passages with the highest scores.
- The generation module takes the first k passages selected by the retriever, the user input q_i, and returns its answer according to the information provided by the passages.

Recent approaches [206] found that query-passage misalignment can lead to lower performance in the retrieval phase, causing a subsequent performance drop in the generation phase.

Given the limited data availability, infeasibility of employing methodologies that necessitate training or fine-tuning a retrieval model [206] for solving this problem, we propose two alternative approaches for addressing this issue:

- **Case generation**: Instead of indexing the original passages, we first automatically generate fictitious legal cases starting from the original passage (simulating the user input) and index them instead of the original one. More formally, for each passage $p_j \in \mathscr{P}$, we generate the corresponding fictitious legal problem $p'_j \in \mathscr{P}'$ with a LLM. Then, the retriever computes the score $s(q_i, p'_j)$ for each new passage $p'_j \in \mathscr{P}'$, ranks each passage score, and returns the top k original passages p_j associated with the generated legal problems p'_j with the highest scores. We generate them with the following prompt:

```
Given the following text, generate two
(2) distinct legal issues or problems
that could realistically arise within
the context described.
```

```
The legal issues should be plausible
and logically connected to the details
provided in the text.

Text:
{}

Legal problem 1: [...]
```

- **Query generation**: A LLM is employed to rewrite the user input q_i, extracting the most important terms q_i'. Therefore, the retrieval module computes the score $s(q_i', p_j)$ for each passage $p_j \in \mathscr{P}$, ranks each passage score, and returns the top k original passages p_j with the highest scores. The generation component takes the first k passages selected by the retriever, the original user input q_i, and returns its answer according to the information provided by the passages. We generate the query with the following prompt:

```
Given the following legal problem,
generate a single high-quality search
query that can help find relevant
information in a book to solve
the problem.

The search query should:

1. Summarize the key concepts and
topics related to the legal problem.
2. Include relevant keywords, legal
terms, and phrases that capture the
essence of the problem.
3. Avoid being too broad or too narrow
, striking a balance between
specificity and comprehensiveness.
4. Generate 1 single query, without
punctuation or special characters.
```

```
The goal is to construct a
query that can effectively retrieve
passages or sections from the book
index that are highly relevant and
useful for addressing the given legal
problem.

Text:
{}

Query:  [...]
```

5.4.3 Results

Dataset [201] introduce a novel dataset of U.S. civil procedures. This dataset has been created from a textbook designed for law students. From this dataset, the following components are extracted:

- *Legal case question*: a description of a legal case and a specific question pertaining to that case.
- *Introduction of the context*: an overview of the most relevant civil procedures for resolving the aforementioned case.
- *Possible solution argument*: a potential answer associated with the question is provided.
- *Annotated label*: it determines whether the possible solution is correct (1) or incorrect (0).

The authors formulate the problem as a binary classification task, where the objective is to predict the correctness of the provided answer, i.e., the label, based on the textual information. We convert the problem from binary classification to multiple-choice question answering problem, adopting the multiple-choice prompting (MCP) [202]. In MCP, the language model see the question and the set of

candidate answers, akin to a multiple-choice examination. Each answer is linked to a symbol, such as *"A," "B,"* or *"C"*. This approach enables the model to explicitly compare answer choices and reduces computational expenses associated with generation tasks. In cases where none of the given options are correct, we generate an additional option *None of the above is true*. In these cases, this option is labeled as the correct answer.

The average number of candidate answers is 4 (with a maximum of 6). The mean number of characters in the input legal problems is 760, with some problems reaching up to 1980 characters. For sake of clarity, we report an example below. An example of **passage** (topic introduction):

> Under Rule 56(a) summary judgment should be granted only if there is "no genuine dispute as to any material fact" and the moving party is entitled to judgment as a matter of law. This is a somewhat misleading provision, unless you parse it very, very closely. Consider this question. In most cases, the plaintiff's complaint does state a claim for which relief can be granted. It isn't usually that hard to allege a valid cause of action. But it is one thing to allege the required elements and quite another to prove that they are true. [...]

An example of **legal case and question** (the user input):

> Tang sues Crucible Laboratories, Inc., for injuries suffered in an accident with Jericho, who was driving a Crucible van at the time of the accident. His complaint alleges that Jericho's negligence caused the accident, that Jericho was acting in the scope of his employment for Crucible at the time, and that Tang suffered a serious back injury as a result of the accident. In its answer, Crucible denies that Jericho was negligent, denies that Tang suffered a back injury from the accident, and denies that Jericho was acting in the scope of employment at the time of the accident. [...] Summary judgment for Crucible should

The associated possible **answers**:

> 1 - be denied, because Tang's supporting evidence shows that there are genuine issues of material fact concerning Jericho's negligence and the

cause of Tang's back injury.
2 - be denied, because the pleadings show that three issues of fact are in dispute: Jericho's negligence, whether he was in the scope of employment, and the cause of Tang's back injury.
3 - be denied, because Tang has submitted evidentiary materials with his opposition to the motion.
4 - be granted.

Experimental setup Three main retrieval methods were explored: 1) dense retrieval using embeddings using the LegalBERT [49], a BERT [110] model pre-trained on the legal domain 2) standard keyword search, and 3) a hybrid approach combining dense and keyword retrieval with re-ranking. The retrieval process involved several hyperparameters:

- Top-K passages to retrieve, ranging from 5 to 10;
- Re-ranking operation: determining whether the re-ranking scores should be added or multiplied to the original query score;
- Top-N passages for re-ranking, from 10 to 100, applicable only for the hybrid search;
- Minimum should match parameters, specifying the percentage of query words that should match with passages, ranging from 0% to 70%.

These hyperparameters were tuned on a validation set. Three generative language models were used in the experiments: the open-source Llama 3 8B [174] and SauLM 7B models [207], as well as the closed-source GPT-4 model from Openai.

RAG pipeline results In table 5.16 we show the results of the complete RAG pipeline. GPT-4 performs better than SAULM and Llama-3 8B across all methods, both SAULM and Llama-3 8B exhibit lower performance compared to GPT-4, with SAULM performing particularly poorly across all methods. Although this model is pre-trained on legal domain, demonstrate lower generalization capability in zero-shot task that involves reasoning skills.

Model	Method	**A**	**P**	**R**	**F1**
SAULM	Lower Bound	0.270	0.170	0.160	0.160
	Upper Bound	0.390	0.330	0.320	0.325
	Naive RAG	0.250	0.220	0.190	0.200
	Case Generation	0.250	0.284	0.250	0.240
	Query Re-writing	0.170	0.115	0.172	0.138
Llama-3 8B	Lower Bound	0.351	0.333	0.378	0.346
	Upper Bound	0.484	0.587	0.490	0.517
	Naive RAG	0.401	0.389	0.377	0.382
	Case Generation	0.400	0.399	0.403	0.401
	Query Re-writing	**0.410**	**0.404**	**0.400**	**0.402**
GPT-4	Lower Bound	0.347	0.252	0.244	0.247
	Upper Bound	0.618	0.560	0.554	0.560
	Naive RAG	0.515	0.414	0.413	0.414
	Case Generation	0.550	0.448	0.449	0.448
	Query Re-writing	**0.590**	**0.550**	**0.560**	**0.550**

Table 5.16 Comparison of RAG performance across different models and retrieval methods.

The query re-writing method yields the highest scores for GPT-4 across all evaluation metrics. This suggests that reformulating the queries to better match the information in the knowledge base can significantly improve the performance of the GPT-4 model. For sake of completeness we also show the upper bound scores, i.e. the theoretical maximum performance that the models could achieve on the task, assuming perfect information retrieval. On the other side, the lower bound represent the minimum performance that the models could achieve without any contextual information (textbook passages).

These results highlight the importance of having access to relevant contextual information for legal problem-solving tasks. While generative alone can provide a baseline performance, access to external knowledge sources and effective information retrieval techniques are crucial for maximizing the performance of these models in this domain.

Information retrieval results Table 5.17 presents the results of the information retrieval phase.

Setting	Search	MRR	R@1	R@3	R@5	R@10	P@1
Baseline	Tensor re-ranking	0.337	**0.219**	**0.414**	**0.516**	**0.633**	**0.219**
	Keyword	**0.341**	**0.219**	**0.414**	0.508	0.625	**0.219**
	Keyword re-ranking	0.068	0.023	0.078	0.117	0.188	0.023
	Tensor	0.152	0.078	0.203	0.250	0.328	0.078
Case generation	Keyword search, Tensor re-ranking	**0.359**	**0.211**	**0.422**	**0.500**	**0.633**	**0.221**
	Keyword	0.323	**0.211**	0.391	0.453	0.594	**0.211**
	Keyword re-ranking	0.085	0.055	0.078	0.125	0.195	0.055
	Tensor	0.161	0.117	0.172	0.211	0.305	0.117
Query re-writing	Tensor re-ranking	**0.368**	**0.234**	**0.453**	**0.563**	**0.688**	**0.234**
	Keyword	0.351	0.227	0.422	0.516	0.680	0.227
	Keyword re-ranking	0.138	0.078	0.156	0.211	0.344	0.078
	Tensor	0.176	0.102	0.195	0.266	0.391	0.102
Case generation and Query re-writing	Tensor re-ranking	**0.353**	**0.203**	**0.445**	0.555	**0.711**	**0.203**
	Keyword	0.347	0.195	**0.445**	**0.570**	0.688	0.195
	Keyword re-ranking	0.106	0.070	0.109	0.156	0.242	0.070
	Tensor	0.158	0.086	0.195	0.242	0.359	0.086

Table 5.17 Information retrieval performance metrics across different methods and search techniques.

Keyword search with tensor te-ranking performs best: this method consistently outperforms or matches the best performance across all settings and metrics. This hybrid approach, which combines keyword matching and dense vector re-ranking, appears to be the most effective for retrieving relevant information from the knowledge base in the legal domain. On the other side, the methods that rely solely or primarily on dense vector representations ("Tensor" and "Tensor search, Keyword re-ranking") consistently underperform compared to the keyword-based and hybrid methods. This suggests that while dense vector representations can capture semantic similarities, they may struggle to effectively retrieve relevant information in the legal domain without the aid of explicit keyword matching.

The query re-writing method consistently improves the retrieval performance compared to the baseline setting, especially for the methods that involves the keywords. This highlights the importance of query reformulation in improving the

alignment between the queries and the knowledge base. The case generation method yields modest improvements over the baseline, particularly for the "Keyword search, Tensor re-ranking" and "Keyword" methods. However, the improvements are not as substantial as those achieved by the query re-writing method.

The combination of problem-generation and query re-writing does not consistently outperform the query re-writing method alone. This finding supports the hypothesis that aligning the query with relevant passages is crucial for the retrieval phase, especially when using tensor search.

Generation quality In table 5.18 we tested the goodness of the generation. To evaluate the quality of authomatically generated problems we consider the overlap between problem generated and true. For queries, verlap between query and problems from which they have been generated from. While the problem generation method achieves higher scores on the ROUGE metrics, suggesting better text quality, the query re-writing method appears to be more effective in the overall RAG pipeline for legal problem-solving. This indicates that the ability to reformulate queries to better match the knowledge base is more crucial for the downstream performance of the complete system.

	R-1	R-2	R-L	R-Lsum	BERT score	Compression ratio
Problem generation	25.16	4.46	14.46	14.89	82.50	114.34
Query re-writing	17.49	7.87	13.38	13.38	82.29	660.69

Table 5.18 Generation quality evaluation for problem generation and query re-writing techniques.

5.5 Summary of results and key insights

Based on the experimental results from three comprehensive legal AI studies, several critical findings emerge that advance our understanding of effective approaches for legal natural language processing tasks. The experiments consistently demonstrate that architectural choices and fine-tuning strategies have greater impact than domain-specific pre-training, with LUKE-based models outperforming legal-specific

pre-trained models in NER tasks, while GPT-4 significantly outperformed domain-specific SAULM models in legal reasoning, suggesting that general reasoning capabilities may be more valuable than legal domain specialization for zero-shot tasks. For court judgment prediction systems, hierarchical approaches with NER masking prove most effective (80.58% F1-score on development, 78.47% on test sets), not only enhancing prediction accuracy but also improving explanation quality and privacy protection by effectively managing sensitive entity information without compromising performance. The studies reveal that collaborative multi-model frameworks, exemplified by the CLUEDO approach with its supervisory "detective" component, achieve superior stability by reducing prediction variance by 50% while maintaining competitive performance, addressing the significant prediction instability observed in individual high-performing models like GPT-4. In information retrieval tasks, query re-writing consistently outperforms other RAG enhancement methods across all models, while hybrid approaches combining keyword search with tensor re-ranking prove more effective than pure dense vector methods, indicating that explicit keyword matching remains crucial in legal domains. Finally, gradient-based explanation methods with NER boosting consistently outperform baseline approaches across multiple evaluation metrics (ROUGE, BLEU, METEOR). Effective information retrieval remains the primary bottleneck in legal RAG systems, emphasizing that successful legal AI systems must balance accuracy, explainability, privacy preservation, and stability while prioritizing query-passage alignment for optimal performance.

Chapter 6

Conclusion

This dissertation explored various aspects of automatic analysis of legal documents, addressing challenges related to the goal of automatic exploration of legal documents, improving content access and exploration, and improving decision-making processes.

When addressing the challenge of enhancing document exploration, the type of legal information has a great impact on LM performance and their performance decreases while considering labels at deeper levels of granularity in the taxonomy because the classification problem gets much more complex.

Classification problems can be effectively deployed in production environments, as they maintain a degree of error that is generally acceptable for most applications. The uncertainty and variability observed in the outputs of LLMs are often comparable to the level of disagreement found in human evaluations. This makes classification tasks a robust and scalable application of LLMs in real-world scenarios. However, it is important to note that, while classification is effective for well-defined problems, it is not sufficient to address more complex challenges.

For improving content accessibility, the presented works demonstrate that combining general and specialized model pre-training yields higher-quality domain adaptation compared to exclusively training summarization models on legal data from scratch, and extending the model's ability to attend to longer sequences of tokens improves legal summarization performance on benchmark datasets.

In text summarization tasks, language models demonstrate good performance, particularly when pre-trained on in-domain data and fine-tuned for specific use

cases. These strategies significantly enhance the quality of the generated summaries. However, trust and transparency remain concerns, as the outputs often lacks the level of explainability required in high-stakes applications.

When facing decision-making, the AI systems face challenges related to calibration, fairness, and transparency. In general, NER methodologies for text preprocessing are beneficial both to obtain more accurate and more plausible predictions. The model, when predicting outcomes, prioritizes the facts and causes over the involved parties, regulations, or previous cases.

Adopting an approach that combines different models for case outcome prediction, enhances prediction stability, provides more accurate explanations, and makes it easier to identify prediction errors.

Large language models, including those specifically fine-tuned for legal reasoning, continue to face difficulties in solving complex reasoning problems. Although closed-source, general-purpose LLMs often outperform smaller, fine-tuned models in terms of generalization capabilities, they still fail to generate reliable reasoning in intricate scenarios. Furthermore, the current generation of models lacks the flexibility to be easily updated or scaled to reflect rapidly changing regulatory frameworks, posing challenges for dynamic and evolving domains such as law.

Limitations Despite the promising results and novel methodologies presented in this dissertation, several limitations remain. First, many of the experiments rely on proprietary datasets or internal taxonomies, which may limit reproducibility and generalization to other legal systems or jurisdictions. While efforts were made to include multilingual and diverse document typologies, most studies focused primarily on Italian legal texts, which may not capture the full variability of global legal corpora.

Second, the evaluation of interpretability and summarization is still largely dependent on human judgment or gold-standard annotations, which may introduce subjectivity or mismatch when applied to dynamically generated outputs from LLMs. This highlights the need for more robust and automated evaluation protocols tailored to legal contexts.

Third, while classification tasks show consistent and scalable performance, more complex reasoning tasks continue to reveal shortcomings in current LLM architec-

tures, including hallucinations, lack of legal rigor, and difficulties with multi-step inference. Fine-tuned models often struggle to generalize across unseen legal scenarios, and closed-source models, while more robust, lack transparency and adaptability for domain-specific customization.

Finally, many of the advanced techniques proposed remain in the experimental phase. Their integration into real-world legal pipelines presents challenges in terms of infrastructure, interpretability, and regulatory compliance, all of which must be carefully addressed in future research.

Future research directions In conclusion, while several benchmarks have been proposed [175], the legal domain lacks a unified and structured benchmark capable of encompassing diverse tasks and providing comprehensive domain coverage, unlike other fields such as the medical domain (e.g., https://huggingface.co/blog/leaderboard-medicalllm). Furthermore, there is a pressing need for custom automated evaluation methods tailored to the legal context, emphasizing the enhancement of evaluation metrics and reporting capabilities by exploring performance measures beyond standard accuracy. These advancements would significantly improve the reliability and applicability of AI systems in legal tasks.

One promising avenue for advancing LLM research involves the integration of symbolic and rule-based methods, creating hybrid approaches. While LLMs excel at generating fluent and contextually relevant text, their output can sometimes lack the logical rigor required for high-stakes applications like legal reasoning. Symbolic reasoning [208] systems, grounded in formal logic and rule-based frameworks, could provide an additional layer of verification or enhancement. For instance, symbolic systems might check LLM-generated arguments against predefined legal principles or regulations, flagging inconsistencies or errors. Moreover, these approaches can play a crucial role in evaluation. Existing benchmarks for LLMs tend to rely on statistical metrics or human judgment, which may not fully capture the quality of legal reasoning. Symbolic methods could help establish more objective criteria by evaluating outputs against structured legal frameworks.

Also, LLM Agents [209] represent a new and promising research direction, particularly in the context of automating complex, multi-step tasks by leveraging the capabilities of large language models in tandem with external tools or APIs. While such agents have been explored in general-purpose domains, their potential

in legal applications remains largely unexplored [210]. The development of LLM Agents for legal reasoning, document drafting, and case analysis could significantly transform the efficiency and accessibility of legal services. Future work should investigate the challenges and opportunities of adapting LLM Agents to the legal domain, addressing issues such as interpretability, adherence to jurisdictional rules, and integration with domain-specific tools.

References

[1] Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4046–4062, Online, August 2021. Association for Computational Linguistics.

[2] Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S. Yu. Large language models in law: A survey. *AI Open*, 5:181–196, 2024.

[3] A. M. TURING. I.—computing machinery and intelligence. *Mind*, LIX(236):433–460, 10 1950.

[4] Thomas K. Landauer and Susan T. Dumais. Latent semantic analysis. *Scholarpedia*, 3:4356, 2008.

[5] Steven Brunton and J. Kutz. *Singular Value Decomposition (SVD)*, pages 3–46. 02 2019.

[6] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[8] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 06 2017.

[9] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 11 1997.

[11] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.

[13] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[14] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[15] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[16] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim,

Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

[17] BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel

van Strien, David Ifeoluwa Adelani, Dragomir Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tam-

mour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sang-aroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. Bloom: A 176b-parameter open-access multilingual language model, 2023.

[18] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck,

Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways, 2022.

[19] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam,

Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü,

Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen,

Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez,

Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Brid-

son, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam

Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2024.

[20] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[21] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, Singapore, December 2023. Association for Computational Linguistics.

[22] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer, 2020.

[23] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, et al. Textbooks are all you need. *arXiv preprint arXiv:2306.11644*, 2023.

[24] Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: **phi-1.5** technical report. *arXiv preprint arXiv:2309.05463*, 2023.

[25] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. The falcon series of open language models, 2023.

[26] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning, 2022.

[27] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.

[28] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023.

[29] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

[30] Jieyi Long. Large language model guided tree-of-thought, 2023.

[31] Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. Active prompting with chain-of-thought for large language models, 2024.

[32] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.

[33] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[34] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. Multieurlex - a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. In *EMNLP*, 2021.

[35] Marc Van Opijnen and Cristiana Santos. On the concept of relevance in legal information retrieval. *Artif. Intell. Law*, 25(1):65–87, mar 2017.

[36] Ugo Mattei. Three Patterns of Law: Taxonomy and Change in the World's Legal Systems. *The American Journal of Comparative Law*, 45(1):5–44, 01 1997.

[37] Alistair Johnson, Tom Pollard, Lu Shen, Li-wei Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Celi, and Roger Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 05 2016.

[38] K. Raghav, Krishna Reddy, and V. Balakista Reddy. Analyzing the extraction of relevant legal judgments using paragraph-level and citation information. 2016.

[39] Arpan Mandal, Raktim Chaki, Sarbajit Saha, Kripabandhu Ghosh, Arindam Pal, and Saptarshi Ghosh. Measuring similarity among legal court case documents. In *Proceedings of the 10th Annual ACM India Compute Conference*, Compute '17, page 1–9, New York, NY, USA, 2017. Association for Computing Machinery.

[40] Wagh RS and Anand D. Legal document similarity: a multi-criteria decision-making perspective. *PeerJ Computer Science*, 6:e262, 2020.

[41] Jörg Landthaler, Bernhard Waltl, Patrick Holl, and Florian Matthes. Extending full text search for legal document collections using word embeddings. In *JURIX*, 2016.

[42] Malte Ostendorff, Elliott Ash, Terry Ruas, Bela Gipp, Julian Moreno-Schneider, and Georg Rehm. *Evaluating Document Representations for Content-Based Legal Literature Recommendations*, page 109–118. Association for Computing Machinery, New York, NY, USA, 2021.

[43] Octavia-Maria Sulea, Marcos Zampieri, Shervin Malmasi, Mihaela Vela, Liviu P. Dinu, and Josef van Genabith. Exploring the use of text classification in the legal domain. *CoRR*, abs/1710.09306, 2017.

[44] Jiaming Gao, Hui Ning, Zhongyuan Han, Leilei Kong, and Haoliang Qi. Legal text classification model based on text statistical features and deep semantic features. In Parth Mehta 0001, Thomas Mandl 0001, Prasenjit Majumder, and Mandar Mitra, editors, *Working Notes of FIRE 2020 - Forum for Information Retrieval Evaluation, Hyderabad, India, December 16-20, 2020*, volume 2826 of *CEUR Workshop Proceedings*, pages 35–41. CEUR-WS.org, 2020.

[45] Haihua Chen, Lei Wu, Jiangping Chen, Wei Lu, and Junhua Ding. A comparative study of automated legal text classification using random forests and deep learning. *Information Processing & Management*, 59(2):102798, 2022.

[46] André Aguiar, Raquel Silveira, Vládia Pinheiro, Vasco Furtado, and João Araújo Neto. Text classification in legal documents extracted from lawsuits in brazilian courts. In André Britto and Karina Valdivia Delgado, editors, *Intelligent Systems*, pages 586–600, Cham, 2021. Springer International Publishing.

[47] Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. Extreme multi-label legal text classification: A case study in EU legislation. In *Proceedings of the Natural Legal Language Processing Workshop 2019*, pages 78–87, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[48] Eneldo Loza Mencía and Johannes Fürnkranz. *Efficient Multilabel Classification Algorithms for Large-Scale Problems in the Legal Domain*, page 192–215. Springer-Verlag, Berlin, Heidelberg, 2010.

[49] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics.

[50] Christos Papaloukas, Ilias Chalkidis, Konstantinos Athinaios, Despina-Athanasia Pantazi, and Manolis Koubarakis. Multi-granular legal topic classification on greek legislation. *CoRR*, abs/2109.15298, 2021.

[51] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. Multieurlex - A multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer. *CoRR*, abs/2109.00904, 2021.

[52] Xin Huang, Boli Chen, Lin Xiao, and Liping Jing. Label-aware document representation via hybrid attention for extreme multi-label text classification. *CoRR*, abs/1905.10070, 2019.

[53] Wei Zhao, Haiyun Peng, Steffen Eger, Erik Cambria, and Min Yang. Towards scalable and reliable capsule networks for challenging NLP applications. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1549–1559, Florence, Italy, July 2019. Association for Computational Linguistics.

[54] Ilias Chalkidis, Manos Fergadiotis, and Ion Androutsopoulos. Multieurlex – a multi-lingual and multi-label legal document classification dataset for zero-shot cross-lingual transfer, 2021.

[55] Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. Large-scale multi-label text classification on EU legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322, Florence, Italy, July 2019. Association for Computational Linguistics.

[56] Hsiang-Fu Yu, Kai Zhong, Inderjit S. Dhillon, Wei-Cheng Wang, and Yiming Yang. X-bert: extreme multi-label text classification using bidirectional encoder representations from transformers. In *NeurIPS 2019 Workshop on Science Meets Engineering of Deep Learning*, 2019.

[57] David Grangier and Dan Iter. The trade-offs of domain adaptation for neural language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3802–3813, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[58] Irene Benedetto, Luca Cagliero, and Francesco Tarasconi. Automatic inference of taxonomy relationships among legal documents. In Silvia Chiusano, Tania Cerquitelli, Robert Wrembel, Kjetil Nørvåg, Barbara Catania, Genoveva Vargas-Solar, and Ester Zumpano, editors, *New Trends in Database and Information Systems*, pages 24–33, Cham, 2022. Springer International Publishing.

[59] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014.

[60] Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*, pages 986–987. Springer US, Boston, MA, 2010.

[61] Quoc V. Le and Tomás Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.

[62] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn:

Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[63] Jiawei Han and Micheline Kamber. *Data Mining. Concepts and Techniques*. Morgan Kaufmann, 2nd ed. edition, 2006.

[64] Giselle B Limentani, Moira C Ringo, Feng Ye, Mandy L Bergquist, and Ellen O McSorley. Beyond the t-test: statistical equivalence testing, 2005.

[65] Irene Benedetto, Luca Cagliero, and Francesco Tarasconi. Extreme classification of european union law documents driven by entity embeddings. volume 3651, 2024. Cited by: 0.

[66] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention. In *EMNLP*, 2020.

[67] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[68] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[69] Taehee Jung, Joo-Kyung Kim, Sungjin Lee, and Dongyeop Kang. Cluster-guided label generation in extreme multi-label classification. In *EACL 2023*, 2023.

[70] Irene Benedetto, Gianpiero Sportelli, Sara Bertoldo, Francesco Tarasconi, Luca Cagliero, and Giuseppe Giacalone. On the use of pretrained language models for legal italian document classification. *Procedia Computer Science*,

225:2244–2253, 2023. 27th International Conference on Knowledge Based and Intelligent Information and Engineering Sytems (KES 2023).

[71] Francesco Tarasconi, Milad Botros, Matteo Caserio, Gianpiero Sportelli, Giuseppe Giacalone, Carlotta Uttini, Luca Vignati, and Fabrizio Zanetta. Natural language processing applications in case-law text publishing. In *International Conference on Legal Knowledge and Information Systems*, 2020.

[72] Sarah Friedrich and Tim Friede. On the role of benchmarking data sets and simulations in method comparison studies, 2022.

[73] Jinghui Lu, Maeve Henchion, Ivan Bacher, and Brian Mac Namee. A sentence-level hierarchical bert model for document classification with limited labelled data. In Carlos Soares and Luis Torgo, editors, *Discovery Science*, pages 231–241, Cham, 2021. Springer International Publishing.

[74] Ronghui You, Suyang Dai, Zihan Zhang, Hiroshi Mamitsuka, and Shanfeng Zhu. Attentionxml: Extreme multi-label text classification with multi-label attention based recurrent neural networks. *CoRR*, abs/1811.01727, 2018.

[75] Xin Li and Dan Roth. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002.

[76] Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*, 2001.

[77] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, pages 722–735, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[78] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[79] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2017.

[80] Pang-Ning Tan, Michael S. Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.

[81] Deepali Jain, Malaya Dutta Borah, and Anupam Biswas. Summarization of legal documents: Where are we now and the way forward. *Computer Science Review*, 40:100388, 2021.

[82] Paheli Bhattacharya, Kaustubh Hiware, Subham Rajgaria, Nilay Pochhi, Kripabandhu Ghosh, and Saptarshi Ghosh. A comparative study of summarization algorithms applied to legal case judgments. In *European Conference on Information Retrieval*, pages 413–428. Springer, 2019.

[83] Seth Polsley, Pooja Jhunjhunwala, and Ruihong Huang. CaseSummarizer: A system for automated summarization of legal texts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 258–262, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[84] Linwu Zhong, Ziyi Zhong, Zinian Zhao, Siyuan Wang, Kevin D. Ashley, and Matthias Grabmair. Automatic summarization of legal decisions using iterative masking of predictive sentences. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, ICAIL '19, page 163–172, New York, NY, USA, 2019. Association for Computing Machinery.

[85] Abhay Shukla, Paheli Bhattacharya, Soham Poddar, Rajdeep Mukherjee, Kripabandhu Ghosh, Pawan Goyal, and Saptarshi Ghosh. Legal case document summarization: Extractive and abstractive methods and their evaluation. In *The 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, 2022.

[86] Mohamed Elaraby and Diane Litman. ArgLegalSumm: Improving abstractive summarization of legal documents with argument mining. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6187–6194, Gyeongju, Republic of Korea, October 2022. International Committee on Computational Linguistics.

[87] Aniket Deroy, Kripabandhu Ghosh, and Saptarshi Ghosh. How ready are pretrained abstractive models and llms for legal case judgement summarization?, 2023.

[88] J. Cui, Z. Li, Y. Yan, B. Chen, and L. Yuan. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*, 2023.

[89] Daniele Licari and Giovanni Comandè. ITALIAN-LEGAL-BERT: A Pretrained Transformer Language Model for Italian Law. In Danai Symeonidou, Ran Yu, Davide Ceolin, María Poveda-Villalón, Davide Audrito, Luigi Di Caro, Francesca Grasso, Roberto Nai, Emilio Sulis, Fajar J. Ekaputra, Oliver Kutz, and Nicolas Troquard, editors, *Companion Proceedings of the 23rd International Conference on Knowledge Engineering and Knowledge Management*, volume 3256 of *CEUR Workshop Proceedings*, Bozen-Bolzano, Italy, September 2022. CEUR. ISSN: 1613-0073.

[90] Andrea Tagarelli and Andrea Simeri. Unsupervised law article mining based on deep pre-trained language representation models with application to the italian civil code. *CoRR*, abs/2112.03033, 2021.

[91] Dennis Aumiller, Ashish Chouhan, and Michael Gertz. EUR-lex-sum: A multi- and cross-lingual dataset for long-form summarization in the legal domain. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7626–7639, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[92] Irene Benedetto, Luca Cagliero, Francesco Tarasconi, Giuseppe Giacalone, and Claudia Bernini. Benchmarking abstractive models for italian legal news summarization. In *International Conference on Legal Knowledge and Information Systems*, 2023.

[93] Irene Benedetto, Luca Cagliero, Michele Ferro, Francesco Tarasconi, Claudia Bernini, and Giuseppe Giacalone. Leveraging large language models for abstractive summarization of italian legal news. *Artificial Intelligence and Law*, February 2025.

[94] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[95] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jun 2022.

[96] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314, Toronto, Canada, July 2023. Association for Computational Linguistics.

[97] Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. *CoRR*, abs/2110.02861, 2021.

[98] La Quatra and Cagliero. Bart-it: An efficient sequence-to-sequence model for italian text summarization. *Future Internet*, 15(1):15, Dec 2022.

[99] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics.

[100] Gabriele Sarti and Malvina Nissim. IT5: Large-scale text-to-text pretraining for italian language understanding and generation. *ArXiv preprint 2203.03759*, mar 2022.

[101] Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020.

[102] Charles Condevaux and Sébastien Harispe. Lsg attention: Extrapolation of pretrained transformers to long sequences. In *Advances in Knowledge Discovery and Data Mining: 27th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2023, Osaka, Japan, May 25–28, 2023, Proceedings, Part I*, pages 443–454. Springer, 2023.

[103] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[104] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[105] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[106] Günes Erkan and Dragomir R. Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Int. Res.*, 22(1):457–479, dec 2004.

[107] J. Steinberger and Karel Jezek. Using latent semantic analysis in text summarization and summary evaluation. *Proceedings of ISIM'04*, pages 93–100, 01 2004.

[108] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China, November 2019. Association for Computational Linguistics.

[109] Derek Miller. Leveraging BERT for extractive text summarization on lectures. *CoRR*, abs/1906.04165, 2019.

[110] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[111] Christopher D. Manning and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT Press, 2001.

[112] Alexander R. Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R. Radev. Multi-news: a large-scale multi-document summarization dataset and abstractive hierarchical model. *CoRR*, abs/1906.01749, 2019.

[113] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[114] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[115] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. *CoRR*, abs/1904.09675, 2019.

[116] Neslihan Iskender, Tim Polzehl, and Sebastian Möller. Reliability of human evaluation for text summarization: Lessons learned and challenges ahead. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 86–96, Online, April 2021. Association for Computational Linguistics.

[117] Irene Benedetto, Moreno La Quatra, and Luca Cagliero. Legitbart: a summarization model for italian legal documents. *Artificial Intelligence and Law*, February 2025.

[118] Joel Niklaus, Veton Matoshi, Matthias Stürmer, Ilias Chalkidis, and Daniel E. Ho. Multilegalpile: A 689gb multilingual legal corpus, 2023.

[119] Peter Henderson, Mark Simon Krass, Lucia Zheng, Neel Guha, Christopher D Manning, Dan Jurafsky, and Daniel E. Ho. Pile of law: Learning responsible data filtering from the law and a 256GB open-source legal dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[120] Maarten Grootendorst. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

[121] Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020.

[122] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[123] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics.

[124] Benjamin Strickson and Beatriz De La Iglesia. Legal judgement prediction for uk courts. In *Proceedings of the 3rd International Conference on Information Science and Systems*, ICISS '20, page 204–209, New York, NY, USA, 2020. Association for Computing Machinery.

[125] Kankawin Kowsrihawat, Peerapon Vateekul, and Prachya Boonkwan. Predicting judicial decisions of criminal cases from thai supreme court using bi-directional gru with attention mechanism. *2018 5th Asian Conference on Defense Technology (ACDT)*, pages 50–55, 2018.

[126] Nikolaos Aletras, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro, and Vasileios Lampos. Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93, 2016.

[127] Andrea Visentin, Alessia Nardotto, and Barry O'Sullivan. Predicting judicial decisions: A statistically rigorous approach and a new ensemble classifier. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1820–1824, 2019.

[128] Alexandre Quemy and Robert Wrembel. On integrating and classifying legal text documents. In Sven Hartmann, Josef Küng, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil, editors, *Database and Expert Systems Applications*, pages 385–399, Cham, 2020. Springer International Publishing.

[129] Rafe Athar Shaikh, Tirath Prasad Sahu, and Veena Anand. Predicting outcomes of legal cases based on legal factors using classifiers. *Procedia Computer Science*, 167:2393–2402, 2020. International Conference on Computational Intelligence and Data Science.

[130] Junyun Cui, Xiaoyu Shen, Feiping Nie, Zheng Wang, Jinglong Wang, and Yulong Chen. A survey on legal judgment prediction: Datasets, metrics, models and challenges. *arXiv preprint arXiv:2204.04859*, 2022.

[131] Ilias Chalkidis, Ion Androutsopoulos, and Nikolaos Aletras. Neural legal judgment prediction in English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4317–4323, Florence, Italy, July 2019. Association for Computational Linguistics.

[132] Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, Nikolaos Aletras, and Ion Androutsopoulos. LEGAL-BERT: The muppets straight out of law school. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2898–2904, Online, November 2020. Association for Computational Linguistics.

[133] Arshdeep Kaur and Bojan Bozic. Convolutional neural network-based automatic prediction of judgments of the european court of human rights. In *Irish Conference on Artificial Intelligence and Cognitive Science*, 2019.

[134] Masha Medvedeva, Ahmet Üstün, Xiao Xu, Michel Vols, and Martijn Wieling. Automatic judgement forecasting for pending applications of the european court of human rights. In *ASAIL/LegalAIIA@ ICAIL*, pages 12–23, 2021.

[135] Waddah Saeed and Christian W. Omlin. Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities. *Knowl. Based Syst.*, 263:110273, 2023.

[136] Adrien Bibal, Michael Lognoul, Alexandre De Streel, and Benoît Frénay. Legal requirements on explainability in machine learning. *Artificial Intelligence and Law*, 29:149–169, 2021.

[137] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[138] Francesco Ventura, Salvatore Greco, Daniele Apiletti, and Tania Cerquitelli. Trusting deep learning natural-language models via local and global explanations. *Knowledge and Information Systems*, 64(7):1863–1907, 2022.

[139] Mattia Setzu, Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. Glocalx - from local to global explanations of black box ai models. *Artificial Intelligence*, 294:103457, 2021.

[140] Davide Napolitano and Luca Cagliero. GX-HUI: global explanations of AI models based on high-utility itemsets. In Hossain Shahriar, Yuuichi Teranishi, Alfredo Cuzzocrea, Moushumi Sharmin, Dave Towey, A. K. M. Jahangir Alam Majumder, Hiroki Kashiwazaki, Ji-Jiang Yang, Michiharu Takemoto, Nazmus Sakib, Ryohei Banno, and Sheikh Iqbal Ahamed, editors, *47th IEEE Annual Computers, Software, and Applications Conference, COMPSAC 2023, Torino, Italy, June 26-30, 2023*, pages 292–297. IEEE, 2023.

[141] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus,

S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[142] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.

[143] Eliana Pastor and Elena Baralis. Explaining black box models by means of local rules. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 510–517, New York, NY, USA, 2019. Association for Computing Machinery.

[144] Eliana Pastor, Alkis Koudounas, Giuseppe Attanasio, Dirk Hovy, and Elena Baralis. Explaining speech classification models via word-level audio segments and paralinguistic features. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2024.

[145] Rohan Bhambhoria, Hui Liu, Samuel Dahan, and Xiaodan Zhu. Interpretable low-resource legal decision making. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11819–11827, 2022.

[146] Łukasz Górski, Shashishekar Ramakrishna, and Jędrzej M. Nowosielski. Towards grad-cam based explainability in a legal text processing pipeline. extended version. In Víctor Rodríguez-Doncel, Monica Palmirani, Michał Araszkiewicz, Pompeu Casanovas, Ugo Pagallo, and Giovanni Sartor, editors, *AI Approaches to the Complexity of Legal Systems XI-XII*, pages 154–168, Cham, 2021. Springer International Publishing.

[147] Rohan Bhambhoria, Samuel Dahan, and Xiaodan Zhu. Investigating the state-of-the-art performance and explainability of legal judgment prediction. In *Canadian Conference on AI*, 2021.

[148] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR, 2017.

[149] Chu Fei Luo, Rohan Bhambhoria, Samuel Dahan, and Xiaodan Zhu. Evaluating explanation correctness in legal decision making. In *Proceedings of the Canadian Conference on Artificial Intelligence (5 2022). https://doi.org/10.21428/594757db. 8718dc8b*, 2022.

[150] Łukasz Górski and Shashishekar Ramakrishna. Explainable artificial intelligence, lawyer's perspective. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, ICAIL '21, page 60–68, New York, NY, USA, 2021. Association for Computing Machinery.

[151] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore, December 2023. Association for Computational Linguistics.

[152] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting for retrieval-augmented large language models, 2023.

[153] Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Derong Xu, Tong Xu, and Enhong Chen. Large language model based long-tail query rewriting in taobao search, 2024.

[154] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels, 2022.

[155] Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy, 2023.

[156] Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation, 2023.

[157] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023.

[158] Haoyan Yang, Zhitao Li, Yong Zhang, Jianzong Wang, Ning Cheng, Ming Li, and Jing Xiao. PRCA: Fitting black-box large language models for retrieval question answering via pluggable reward-driven contextual adapter. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5364–5375, Singapore, December 2023. Association for Computational Linguistics.

[159] Zichun Yu, Chenyan Xiong, Shi Yu, and Zhiyuan Liu. Augmentation-adapted retriever improves generalization of language models as generic plug-in, 2023.

[160] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[161] Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, Luke

Zettlemoyer, and Scott Yih. Ra-dit: Retrieval-augmented dual instruction tuning, 2024.

[162] Jianyi Zhang, Aashiq Muhamed, Aditya Anantharaman, Guoyin Wang, Changyou Chen, Kai Zhong, Qingjun Cui, Yi Xu, Belinda Zeng, Trishul Chilimbi, and Yiran Chen. Reaugkd: Retrieval-augmented knowledge distillation for pre-trained language models. In *ACL 2023*, 2023.

[163] Antoine Louis, G. van Dijck, and Gerasimos Spanakis. Interpretable long-form legal question answering with retrieval-augmented large language models. In *AAAI Conference on Artificial Intelligence*, 2023.

[164] Nirmalie Wiratunga, Ramitha Abeyratne, Lasal Jayawardena, Kyle Martin, Stewart Massie, Ikechukwu Nkisi-Orji, Ruvan Weerasinghe, Anne Liret, and Bruno Fleisch. Cbr-rag: Case-based reasoning for retrieval augmented generation in llms for legal question answering, 2024.

[165] Cheol Ryu, Seolhwa Lee, Subeen Pang, Chanyeol Choi, Hojun Choi, Myeonggee Min, and Jy-Yong Sohn. Retrieval-based evaluation for LLMs: A case study in Korean legal QA. In Daniel Preoțiuc-Pietro, Catalina Goanta, Ilias Chalkidis, Leslie Barrett, Gerasimos (Jerry) Spanakis, and Nikolaos Aletras, editors, *Proceedings of the Natural Legal Language Processing Workshop 2023*, pages 132–137, Singapore, December 2023. Association for Computational Linguistics.

[166] Cong Jiang and Xiaolei Yang. Legal syllogism prompting: Teaching large language models for legal judgment prediction, 2023.

[167] Andrew Blair-Stanek, Nils Holzenberger, and Benjamin Van Durme. Can gpt-3 perform statutory reasoning?, 2023.

[168] Fangyi Yu, Lee Quartey, and Frank Schilder. Exploring the effectiveness of prompt engineering for legal reasoning tasks. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13582–13596, Toronto, Canada, July 2023. Association for Computational Linguistics.

[169] John J. Nay. Large language models as fiduciaries: A case study toward robustly communicating with artificial intelligence through legal standards, 2023.

[170] Jakub Drápal, Hannes Westermann, and Jaromir Savelka. Using large language models to support thematic analysis in empirical legal studies, 2023.

[171] Jaromir Savelka. Unlocking practical applications in legal domain: Evaluation of gpt for zero-shot semantic annotation of legal texts. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL 2023. ACM, June 2023.

[172] Jaromir Savelka, Kevin D. Ashley, Morgan A. Gray, Hannes Westermann, and Huihui Xu. Explaining legal concepts with augmented large language models (gpt-4), 2023.

[173] Hannes Westermann, Jaromir Savelka, and Karim Benyekhlef. Llmediator: Gpt-4 assisted online dispute resolution, 2023.

[174] AI@Meta. Llama 3 model card. 2024.

[175] Neel Guha, Julian Nyarko, Daniel E Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N Rockmore, et al. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *arXiv preprint arXiv:2308.11462*, 2023.

[176] OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey

Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2023.

[177] Christopher Williams. *Tradition and Change in Legal English*. Peter Lang Verlag, Lausanne, Switzerland, 2005.

[178] Ilias Chalkidis and Anders Søgaard. Improved multi-label classification under temporal concept drift: Rethinking group-robust algorithms in a label-wise setting. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2441–2454, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[179] Yongfu Dai, Duanyu Feng, Jimin Huang, Haochen Jia, Qianqian Xie, Yifang Zhang, Weiguang Han, Wei Tian, and Hao Wang. Laiw: A chinese legal large language models benchmark (A technical report). *CoRR*, abs/2310.05620, 2023.

[180] Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen, Zongwen Shen, and Jidong Ge. Lawbench: Benchmarking legal knowledge of large language models. *CoRR*, abs/2309.16289, 2023.

[181] Irene Benedetto, Luca Cagliero, Francesco Tarasconi, Giuseppe Giacalone, and Claudia Bernini. Benchmarking abstractive models for italian legal news summarization. In Giovanni Sileno, Jerry Spanakis, and Gijs van Dijck,

editors, *Legal Knowledge and Information Systems - JURIX 2023: The Thirty-sixth Annual Conference, Maastricht, The Netherlands, 18-20 December 2023*, volume 379 of *Frontiers in Artificial Intelligence and Applications*, pages 311–316. IOS Press, 2023.

[182] Irene Benedetto, Alkis Koudounas, Lorenzo Vaiani, Eliana Pastor, Elena Baralis, Luca Cagliero, and Francesco Tarasconi. PoliToHFI at SemEval-2023 task 6: Leveraging entity-aware and hierarchical transformers for legal entity recognition and court judgment prediction. In Atul Kr. Ojha, A. Seza Doğruöz, Giovanni Da San Martino, Harish Tayyar Madabushi, Ritesh Kumar, and Elisa Sartori, editors, *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1401–1411, Toronto, Canada, July 2023. Association for Computational Linguistics.

[183] Irene Benedetto, Alkis Koudounas, Lorenzo Vaiani, Eliana Pastor, Luca Cagliero, Francesco Tarasconi, and Elena Baralis. Boosting court judgment prediction and explanation using legal entities. *Artificial Intelligence and Law*, pages 1–36, 03 2024.

[184] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online, November 2020. Association for Computational Linguistics.

[185] Prathamesh Kalamkar, Astha Agarwal, Aman Tiwari, Smita Gupta, Saurabh Karn, and Vivek Raghavan. Named entity recognition in Indian court judgments. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 184–193, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[186] Irene Benedetto, Alkis Koudounas, Lorenzo Vaiani, Eliana Pastor, Elena Baralis, Luca Cagliero, and Francesco Tarasconi. PoliToHFI at SemEval-2023 task 6: Leveraging entity-aware and hierarchical transformers for legal entity recognition and court judgment prediction. In *Proceedings of the The 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1401–1411, Toronto, Canada, July 2023. Association for Computational Linguistics.

[187] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[188] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.

[189] Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics.

[190] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[191] Alon Lavie and Abhaya Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[192] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147, 2003.

[193] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[194] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*, 2017.

[195] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[196] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[197] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR, 06–11 Aug 2017.

[198] Giuseppe Attanasio, Eliana Pastor, Chiara Di Bonaventura, and Debora Nozza. ferret: a framework for benchmarking explainers on transformers. In Danilo Croce and Luca Soldaini, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: System*

Demonstrations, pages 256–266, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[199] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. Explainability for large language models: A survey, 2023.

[200] Lena Held and Ivan Habernal. SemEval-2024 Task 5: Argument Reasoning in Civil Procedure. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[201] Leonard Bongard, Lena Held, and Ivan Habernal. The legal argument reasoning task in civil procedure. In Nikolaos Aletras, Ilias Chalkidis, Leslie Barrett, Cătălina Goanță, and Daniel Preoțiuc-Pietro, editors, *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 194–207, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics.

[202] Joshua Robinson, Christopher Michael Rytting, and David Wingate. Leveraging large language models for multiple choice question answering, 2023.

[203] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[204] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

[205] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna M. Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks. *CoRR*, abs/2107.03342, 2021.

[206] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering, 2020.

[207] Pierre Colombo, Telmo Pessoa Pires, Malik Boudiaf, Dominic Culver, Rui Melo, Caio Corro, Andre F. T. Martins, Fabrizio Esposito, Vera Lúcia Raposo, Sofia Morgado, and Michael Desa. Saullm-7b: A pioneering large language model for law, 2024.

[208] Marius-Constantin Dinu, Claudiu Leoveanu-Condrei, Markus Holzleitner, Werner Zellinger, and Sepp Hochreiter. Symbolicai: A framework for logic-based approaches combining generative models and solvers, 2024.

[209] Matthias Baldauf, Schahram Dustdar, and Florian Rosenberg. A survey on context-aware systems. *Information Systems*, 2, 01 2007.

[210] Zhitao He, Pengfei Cao, Chenhao Wang, Zhuoran Jin, Yubo Chen, Jiexin Xu, Huaijun Li, Kang Liu, and Jun Zhao. AgentsCourt: Building judicial decision-making agents with court debate simulation and legal knowledge augmentation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9399–9416, Miami, Florida, USA, November 2024. Association for Computational Linguistics.