Enhancing Legal Text Entailment: Evaluating Model Architectures, Training Approaches, and Interpretability

by

Michel Custeau

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the degree of
Master of Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering
University of Ottawa

Examining Committee

The following served on the Examining Committee for this thesis.

Internal Members: Wolfgang Alschner
Associate Professor, Faculty of Law
University of Ottawa

Lionel Briand
Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Supervisor: Diana Inkpen
Professor, School of Electrical Engineering & Computer Science
University of Ottawa

Declaration of Authorship

I hereby certify that this thesis is entirely my own original work except where otherwise indicated. I am aware of the University of Ottawa regulations concerning plagiarism, including those regarding consequent disciplinary actions. Any use of the works of any other author, in any form, is properly acknowledged at their point of use.

Abstract

The legal domain is a challenge for Artificial Intelligence systems as it is characterized by its complex vocabulary, intricate reasoning, and consistency with precedents. With increasing digitization, the potential for Artificial Intelligence to become a useful tool for legal projects has grown significantly. However, adoption within the legal field lags behind other industries due to cultural resistance, limited high-quality training data, and the need for interpretability in black-box systems. To contribute to the advancement of the role of AI in the legal domain, we developed and evaluated a legal entailment classification system which determines whether a paragraph from an existing legal case supports the decision in a new case, while also providing a justification for the classification.

Leveraging advanced Natural Language Processing techniques and Explainable AI methodologies, this work integrates domain-specific pretraining, lightweight adaptation techniques such as LoRA, and an ensemble technique. A dataset of over 35,000 Canadian legal cases was used for further pretraining, while fine-tuning and evaluation were performed using the COLIEE 2023 competition dataset. Our experiments highlight the trade-offs between computational efficiency and performance, and evaluate the impact of domain-specific pretraining on smaller transformer models such as RoBERTa, compared to adaptations of larger language models for the classification task.

In addition to classification, this thesis explores the role of explainablility techniques for Artificial Intelligence in legal applications by implementing the techniques of LIME and model-generated justifications. These methods were assessed using human evaluations, as well as automated sufficiency metrics, with highest scores in the human evaluation being 82.14% for adequacy, 92.86% for understandability, and 85.71% for trustworthiness, and a peak score of 99.17% for the automated sufficiency metric.

The contributions of this research include the creation of domain-specific pretrained models, a comparative evaluation of fine-tuning and lightweight adaptation techniques for large language models, and a systematic exploration of explainability methods to improve interpretability and user trust. To the best of our knowledge, this study is the first within the Canadian legal AI context to investigate the effects of further pretraining on both small and large language models, as well as to integrate language model adaptation and explainability into a unified system for legal text entailment classification.

Acknowledgements

The work that I accomplished here would not have been possible without the people who stood by me. I would like to express my sincere gratitude to my supervisor, Dr. Diana Inkpen. She has given me immense support and guidance throughout both my undergraduate Honours project and my Master's thesis. She was always present when I needed help, offering solid advice at every stage of the research. She also opened the door for me to join my team at Justice Canada. I'm deeply thankful for her patience, her encouragement, and for always believing I could pull this off.

I would also like to thank the members of my defense committee, Dr. Wolfgang Alschner and Dr. Lionel Briand, for taking the time to read and analyze my thesis. Their feedback and insights greatly contributed to sharpening the quality of this work.

I am grateful to the COLIEE organizers for allowing me to participate in the competition and for providing access to the legal data used in this study. I also wish to thank the team at CanLII for granting me access to additional legal case data. These contributions enabled the experimentation and evaluation of this research.

Finally, I wish to thank my mother, my father, Beril, Bonpa, Lucie, and all of my other family members and friends who have showed their support throughout this journey.

Table of Contents

List of Tables

List of Figures

Abbreviations

AI Artificial Intelligence 1, 2, 8, 9, 11, 13, 19, 20, 21, 25, 37, 38, 39, 40, 46, 59, 63, 73, 77, 78, 80, 81

ANLI Adversarial Natural Language Inference 27, 32, 38

BERT Bidirectional Encoder Representations from Transformers 2, 13, 14, 17, 18, 19, 25, 29, 30, 31, 37, 43, 51, 62, 70, 76, 80

BiLSTM Bidirectional Long Short-Term Memory 30

BLEU Bilingual Evaluation Understudy 20

BoW Bag-of-Words 10, 17, 30

CANLII Canadian Legal Information Institute 41, 45, 50, 57

CNNs Convolutional Neural Networks 12, 13

COLIEE Competition on Legal Information Extraction and Entailment 27, 28, 38, 41, 45, 46, 61, 62, 81

DBNs Deep Belief Networks 12

DL Deep Learning 8, 12, 13, 18, 19, 21, 24, 25, 38, 43, 44, 67

FRESH Faithful Rationale Extraction from Saliency tHresholding 37, 73, 76

GPT Generative Pre-trained Transformer 13, 14, 15, 19, 20, 25, 32

KNN K-Nearest Neighbors 11

LIME Local Interpretable Model-Agnostic Explanations 22, 33, 34, 38, 40, 60, 61, 63, 73, 74, 75, 76, 77, 81, 93

LLM Large Language Model 3, 4, 14, 15, 19, 33, 38, 43, 53, 56, 60, 61, 62, 66, 67, 68, 69, 70, 77, 79, 80, 81

LoRA Low-Rank Adaptation 3, 4, 53, 54, 55, 56, 60, 63, 65, 68, 74

ML Machine Learning 8, 10, 11, 12, 17, 19

MLM Masked Language Modeling 13, 14, 25, 30, 41, 50, 51, 70

MNLI Multi-Genre Natural Language Inference 27, 28, 33, 38

NER Named Entity Recognition 9, 17, 19

NF4 4-Bit NormalFloat 55, 56

NLP Natural Language Processing 3, 4, 5, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 19, 20, 21, 24, 25, 26, 31, 32, 33, 48, 49, 69, 78, 80, 82, 83

NMVs Neural Modular Networks 23

NSP Next Sentence Prediction 14, 25, 30

POS Part-of-Speech 17

Q&A Question Answering 7, 9, 13, 14, 19

RAG Retreival-Augmented Generation 19

RNNs Recurrent Neural Networks 12, 13

RoBERTa A Robustly Optimized BERT Pretraining Approach 3, 4, 14, 19, 29, 31, 41, 47, 48, 49, 51, 57, 58, 60, 61, 62, 64, 65, 67, 69, 70, 71, 72, 73, 74, 76, 77, 78, 79

ROUGE Recall-Oriented Understudy for Gisting Evaluation 20

SHAP SHapley Additive exPlanations 33, 34, 38, 60, 81

SNLI Stanford Natural Language Inference 26, 27, 28, 38, 45, 58, 72

SVM Support Vector Machines 11, 19

TF-IDF Term Frequency-Inverse Document Frequency 10, 17, 18

XAI Explainable Artificial Intelligence 3, 4, 5, 21, 22, 25, 33, 37, 66, 78, 80

Chapter 1

Introduction

The legal domain is known for having a highly specialized vocabulary, complex reasoning, and significant impact on real-world decisions. As legal systems are becoming more digitized, the increasing integration for tools based on Artificial Intelligence (AI) has the potential to significantly transform the way legal services are delivered and evaluated. As noted by Linna Jr (2021), "undertaking legal technology, data analytics, and artificial intelligence projects in law can help us establish standards and metrics for quality and value." However, there has been resistance in the culture of the legal field to adopt artificial intelligence based technologies in comparison to other industries such as medicine or manufacturing. For instance, Linna Jr (2021) also observes that there is a lack of supervised-learning training data in the legal field. The research cites that "to make progress with artificial intelligence and data analytics, we need (1) high-quality input and outcome data and (2) an understanding of what outcomes are optimal. There has not been enough discussion about the quality of legal industry data. A sober assessment of legal industry input, process, and outcome data would reveal serious shortcomings."

For these reasons, it is essential to further advance the legal culture toward promoting

accuracy and trust in relevant artificial intelligence applications within the legal domain. This need has driven the development of legal-specific NLP tasks, such as Legal Document Summarization, Legal Argument Mining, and Legal Judgment Prediction (Ariai and Demartini, 2024). Among these, the application we chose for this study is legal entailment classification, the classification of whether a specific paragraph from an existing legal case entails the decision of a new case. This task is relevant to both the theory and practice of law as it is compatible with the fundamental logic of law, which is that it is always evolving while simultaneously staying consistent. By nature, law is not static but dynamic. It must adapt to the new changes in culture and society, yet must simultaneously stay grounded and consistent with precedent. In other words, to study entailment and contradiction is also to study what is consistent and what is not consistent, a foundational pillar of the judicial system.

However, the classification of entailment in a legal setting offers some unique challenges. Unlike text that we find in general settings, legal documents contain sentences with nuanced meanings, complex details, and domain-specific terminology. These factors make it necessary to develop models specifically trained and adapted to data in the legal domain. Furthermore, explainability in AI remains a concern and an ongoing area of research as legal professionals require not only accurate predictions in their systems, but also clear and interpretable explanations which they can trust for their decision-making process. While black-box models can offer higher performance than symbolic methods, given the critical implications associated with legal texts, explainability in AI systems necessitates the development of methods that balance both performance and transparency. For this reason, we propose a novel system that enables a single language model to perform both classification and explanation generation. This system can be adapted for use with either a small language model based on the Bidirectional Encoder Representations from Transformers

(BERT) (Devlin et al., 2019) architecture, or a Large Language Model (LLM). The models that are employed in this study are A Robustly Optimized BERT Pretraining Approach (RoBERTa) (Liu, 2019), Llama 2 (Touvron et al., 2023), Llama 3 (Dubey et al., 2024), and GPT-4o (Hurst et al., 2024).

1.1 Objectives

The primary objective of this thesis is to develop and evaluate a robust system for legal entailment classification that leverages advanced Natural Language Processing (NLP) techniques and Explainable Artificial Intelligence (XAI) methodologies. The specific objectives are as follows:

- **Developing a Legal Entailment Classification System:** Design and implement a system that integrates multiple state-of-the-art transformer models training methods, and XAI techniques to address the complex task of legal entailment classification.
- **Pretraining and Fine-tuning with Legal Domain Data:** Investigate the impact of domain-specific pretraining on the performance of language models for legal text processing.
- **Exploring Lightweight Adaptation Techniques for LLMs:** Explore and compare adaptation strategies for LLMs, including fine-tuning, prompt engineering, and Low-Rank Adaptation (LoRA).
- **Evaluating Ensembles:** Evaluate the effectiveness of ensembles for improving classification performance by combining predictions from multiple models.

- **Incorporating Explainability:** Integrate XAI techniques to provide interpretable justifications for model predictions.

1.2 Contributions

This thesis makes several contributions to the fields of NLP and XAI. Below the key contributions of this thesis are listed:

- **Development of a unified Legal Entailment System:** This thesis introduces a novel system specifically tailored for legal entailment classification that enables a single model to perform both classification and explanation generation. The approach accommodates LLMs by employing a method involving the swapping of language modeling heads.
- **Creation of Domain-Specific Pretrained Legal Models** This thesis contributes an open-source RoBERTa and a Llama 2 model further pretrained on a large corpus of Canadian legal cases, totaling over 35,000 documents. By sharing this model with the research community, this work provides a valuable resource for future applications and studies in Canadian legal NLP.
- **Evaluation of Training Strategies and Ensembles for Specialized NLP Tasks** This thesis investigates the effectiveness of domain-specific pretraining, fine-tuning techniques, and ensemble strategies for legal entailment classification. It compares state-of-the-art models such as RoBERTa, Llama 2, Llama 3, and GPT-4o under different adaptation methods, including LoRA fine-tuning, prompt engineering, and ensemble-based approaches. Key insights are provided into the trade-offs between

computational efficiency and performance for large-scale and resource-constrained NLP systems.

- **Application and Assessment of Explainability Methods** Given the critical importance of explainability in the legal domain, this thesis explores both automated and human evaluations of model explanations.

1.3 Thesis Organization

This thesis is structured into six chapters:

- **Chapter 1: Introduction** This chapter outlines the purpose, objectives, and contributions of this thesis.
- **Chapter 2: Background** This chapter presents relevant concepts to this thesis through an overview the evolution of the field of NLP, the key stages of NLP systems, and important methods in XAI.
- **Chapter 3: Literature Review** This chapter discusses prior work and existing gaps related to entailment classification, domain-specific model adaptation, and explainability methods in the context of legal applications.
- **Chapter 4: Methodology** This chapter details the general architecture, datasets, preprocessing strategies, classification techniques, and explainability methods applied throughout this thesis.
- **Chapter 5: Discussion** This chapter discuses the results of the conducted experiments, analyzing the impact of domain-specific pretraining, fine-tuning methods, ensemble strategies, and explainability techniques.

- **Chapter 6: Conclusion** This chapter summarizes the key findings and contributions of this thesis. It also discusses the limitations of the research, and suggests future directions for further advancement in the integration of AI into legal decision-making.

1.4 Paper Publications and Submissions

- Custeau, M. & Inkpen, D. (2023). Individual models can perform better than agreement-based ensembles. In Proceedings of the Workshop of the 10th Competition on Legal Information Extraction/Entailment (COLIEE'2023) in the 19th International Conference on Artificial Intelligence and Law (ICAIL 2023).

- Custeau, M. & Inkpen, D. (2025). Enhancing Legal Text Entailment: Evaluating Model Architectures, Training Approaches, and Interpretability. In Proceedings of the 38th Canadian Conference on Artificial Intelligence (Canadian AI 2025).

Chapter 2

Background

In this chapter, we will explore key areas to the field of NLP, alongside its evolution of different methods.

2.1 Evolution of NLP

Due to the vast amount of machine readable forms of natural language, such as web pages, articles, and records, and also the growing demand for agents able to automatically interact with users using speech and text, the field of NLP has seen a surge in popularity. The field of NLP is described as the study of automating tasks involving human language, and is done using algorithms that process language in order to perform a task, such as text analytics, Question Answering (Q&A), and machine translation. A continuing challenge in NLP involves dealing with ambiguity due to the different ways a human can interpret words or sentences. Unlike computer code which is able to be exact and precise, human language can be ambiguous, as the meaning of words and sentences can vary based on context,

cultural nuances, and individual interpretation. This complexity makes it challenging for machines to achieve human-like comprehension and reasoning.

As a result, the field of NLP is one of the critical subfields of AI, and intersects with the fields of Machine Learning (ML) and Deep Learning (DL), through its different methods and training strategies to process language.

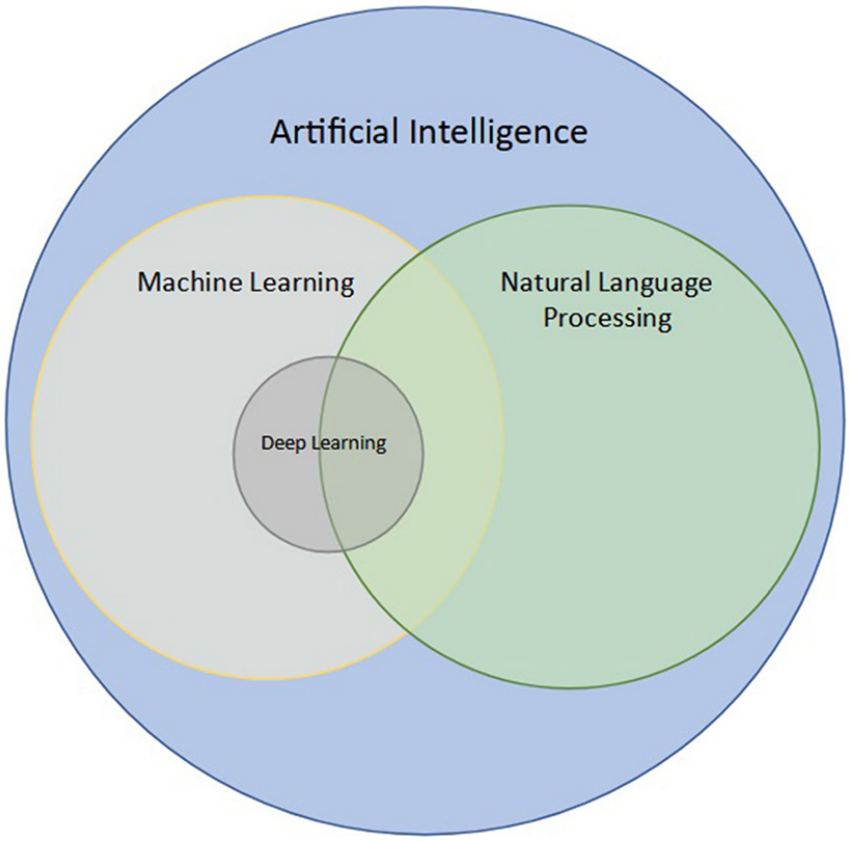

Figure 2.1: Relationship between AI, ML, DL, and NLP from Chen and Baxter (2022)

NLP methods are widely used across various domains, with applications that enhance human-computer interaction, automate writing tasks, and extract insights from textual data. These applications include:

- **Text Classification**: Categorizing text into predefined labels. This is useful in contexts where distinguishing between different types of text is essential, such as spam detection, topic categorization, and intent recognition.
- **Sentiment Analysis**: Determining the sentiment of a given text, such as positive, negative, or neutral. This is useful for contexts such as customer feedback analysis and social media monitoring.
- **Topic Extraction**: Identifying key topics or themes within a text corpus. This is useful for providing an overview of ideas from a large set of documents, or performing market analysis.
- **Named Entity Recognition (NER)**: Identifying proper nouns such as names, locations, organizations, and dates in a text.
- **Machine Translation**: Automatically translating text from one language to another. This has been exemplified by popular translation AI systems such as Google Translate.
- **Q&A**: Developing systems that can understand and answer human questions. This has been exemplified in virtual assistants like Siri and Amazon's Alexa.
- **Text Summarization**: Generating concise summaries of large documents while preserving key information, useful in any industry involving large sets of long documents, such as the medical field.

Through these applications, the versatility and importance of NLP can be demonstrated not only in an academic setting, but also across different industries, including healthcare, finance, e-commerce, and education.

2.1.1 Rule-based NLP

Rule-based NLP is an approach that relies on predefined semantic patterns and syntactic rules to analyze and extract information from text. These systems use human-crafted rules to recognize specific text structures, such as sentence patterns or keywords, enabling data extraction. Rule-based NLP models utilize task and domain specific semantic knowledge through ontologies or lexicons to interpret specialized terms and relationships (Xu and Cai, 2021). However, this approach is limited by its rigidity and struggles with the variability and ambiguity inherent in natural language. It is also much more time consuming to create in comparison to ML algorithms due to the need to create all the necessary logical rules (Gunter et al., 2022). For these reasons, other approaches can often seem like more attractive options when high levels of complexity and generalization are required for the task.

2.1.2 Statistical NLP

Statistical NLP applies mathematical models and probability theory to analyze text. It transforms text into numerical data that can be processed by computational models in order to predict the relevancy between terms in a corpus. This approach offers flexibility and generalization without extensive manual rule creation. Popular Statistical NLP methods include techniques such as Bag-of-Words (BoW), n-grams, and Term Frequency-Inverse Document Frequency (TF-IDF). However, since an importance-based score is solely used to represent each word or set of n words, this results in statistical methods that can struggle to capture semantic similarity due to similar words in meaning being treated as unrelated and separate. Another key limitation is that these methods only focus on isolated set of words and disregard the context in which the words were used (Pereira et al., 2024).

2.1.3 Machine Learning

The field of ML utilizes algorithms that enable computers to learn and make predictions based on large amounts of data. Unlike traditional programming approaches where rules are explicitly defined, ML systems identify patterns and relationships within data through iterative training processes. The field combines principles from various mathematical disciplines, such as probability and statistics, linear algebra, and differential calculus, to train models to improve their performance on specific tasks over time. This mathematical foundation, combined with modern computational capabilities, enables ML systems to handle complex problems that would be difficult or impossible to solve using traditional rule-based programming (Rebala et al., 2019). These capabilities have led to breakthroughs and rapid advancements in the field of AI and NLP.

2.1.4 Supervised Learning

Supervised learning is a ML approach where a model is trained on labeled data, meaning that each input is paired with its corresponding output. The training phase involves learning a mapping between inputs and outputs by minimizing the error between predictions and actual labels. Once trained, the model can generalize to unseen data by making predictions based on learned patterns. Supervised learning is categorized into two types: classification and regression. Classification deals with discrete labels, such as predicting the category in which a text belongs to, while regression involves continuous values, like predicting numerical outcomes based on input features. Common algorithms in the field of supervised learning include Support Vector Machines (SVM), decision trees, K-Nearest Neighbors (KNN), and Naïve Bayes (Sindhu Meena and Suriya, 2020).

2.1.5 Unsupervised Learning

Unsupervised learning is a ML approach used to uncover hidden patterns within data without the use of predefined labels. It will often use methods to group or cluster similar data points based on their intrinsic characteristics (Sindhu Meena and Suriya, 2020). These methods are particularly useful where insights can be gained from large, unstructured datasets. An example of this is the clustering of tokens to extract topics from a corpus using the LDA algorithm for topic modeling (Jelodar et al., 2019).

2.1.6 Deep Learning

The field of DL is a subfield of ML that involves training models with multiple layers of artificial neural networks to learn hierarchical representations of data. Common architectures used for DL models are Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Belief Networks (DBNs). Models based on DL architectures use stacked layers to progressively extract higher-level features from raw data during training. A key aspect of DL is its scalability. As the size of the training dataset increases, performance tends to improve also. But this boost of performance comes with its own set of potential drawbacks, such as overfitting, interpretability issues, and optimization issues due to its reliance on high-dimensional, non-convex parameter spaces. Overall, the use of DL has led to breakthroughs in various fields by enabling models to generalize effectively from large datasets, and as a result has become a very popular ML methods, though its use comes with challenges such as high computational costs and a need for vast amounts of labeled data (Dong et al., 2021).

2.1.7 Neural NLP

Neural NLP uses DL models to learn complex patterns in human language, eliminating the need for extensive manual feature engineering that we find in statistical methods. Using neural architectures, these models can generate context-sensitive word embeddings that capture semantic and syntactic relationships. RNNs and CNNs were foundational architectures, but transformer-based models, such as BERT and Generative Pre-trained Transformer (GPT), have significantly advanced the field through an attention mechanisms built within the transformer architecture, to model global word dependencies (Vaswani et al., 2017). These transformer models are pretrained on massive corpora and then fine-tuned for various NLP tasks, achieving state-of-the-art performance on tasks like machine translation, Q&A, and sentiment analysis. While neural NLP approaches are highly effective due to their scalability and adaptability across different tasks, they require vast amounts of data and computational resources, making them challenging to use in low-resource settings. Despite these challenges, their versatility and ability to generalize to unseen data have made pretrained transformer models the dominant framework in modern NLP (Lauriola et al., 2022).

2.1.8 Pretraining

Following the invention of the transformer architecture, many NLP systems utilizing this architecture have adopted the step of pretraining, and the majority of the language models available for NLP research have already gone through the pretraining step of Masked Language Modeling (MLM), giving the ability to researchers of jumping straight to the step of fine-tuning. The goal of pretraining is to minimize the need for heavy engineering of the AI model architecture to make it task-specific, allowing the model to be more

generalizable and capable of handling various types of NLP tasks. Another benefit of pretraining is the reduction in overall computation time, as pretraining on large corpera needs to be performed only once, and only the additional fine-tuning on smaller, task-specific datasets needs to be repeated. MLM consists of masking a certain percentage of all tokens in each sequence at random, and then getting the model to predict the value of each masked token. In the case of BERT based models, this is done bidirectionally by looking at both the left and right side of each masked token, while in the case of language models like GPT, this is done using conditional generation by only looking at tokens to the left of the masked token. Additionally, BERT employs a second pretraining task known as Next Sentence Prediction (NSP), which is designed to improve the model's ability to understand relationships between sentences. When using NSP, the model is presented with two consecutive sentences, and it is trained to classify whether the second sentence follows the first one in the original text, or is a randomly selected sentence (Devlin, 2018). However, later studies, such as those on RoBERTa, suggest that removing the NSP objective and instead training on more epochs, longer sequences, and changing the masking pattern on each epoch, leads to better performance in some cases (Liu, 2019).

2.1.9 Prompt Engineering

Given the text generation capabilities of LLMs that was acquired from their vast pretraining data, prompt engineering utilizes this text generation capability to perform a variety of tasks. Prompt engineering optimizes the performance of LLMs by carefully designing input prompts to guide model outputs without modifying internal parameters. This input prompt leverages task-specific instructions to create desired behaviors from models, allowing them to adapt to diverse applications such as language generation, reasoning, and Q&A. Prompts can be in the form of natural language instructions or structured examples that provide

context for the task at hand (Sahoo et al., 2024). Common strategies include:

- **Zero-Shot Prompting:** In this prompting method, the model is instructed to solve the task using only natural language description, with no additional examples provided. A model using this method will require extensive pretraining to perform the correct behavior as it generates responses based on purely its pre-existing knowledge from the texts it was pretrained on.
- **Few-Shot Prompting:** In this prompting method, the instructions for the task are provided similarly to zero-shot prompting. However, a handful of input-output examples are also provided. This provides the model with a clearer context on the task at hand and the behavior that is expected of it.

These approaches enable models like GPT and Llama to generalize across new tasks with no retraining. It also plays a crucial role in reducing the need for building large and task-specific fine-tuning datasets, making it a highly effective tool for adapting LLMs when labelled data is not available in high quantities. However, challenges such as prompt design complexity, sensitivity to input phrasing, and the fact that the model may not have enough respective documents in its training corpus match the precision of the task, remain critical areas for ongoing research and refinement (Zhao et al., 2024).

2.2 Stages of Processing Language

The field of NLP involves multiple stages that transform raw text into meaningful insights. The workflow typically starts with preprocessing raw textual data, followed by feature engineering to extract useful representations, selecting models to perform specific tasks, and

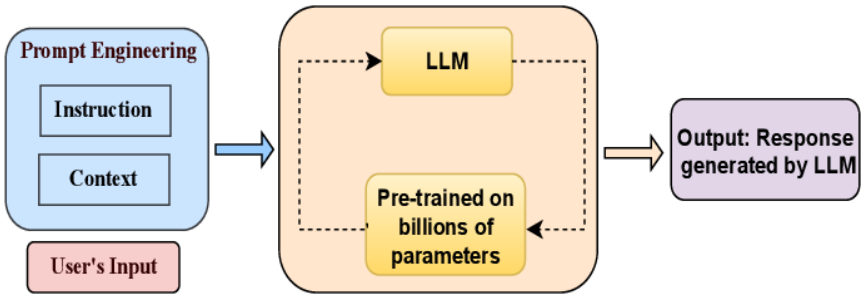

Figure 2.2: Diagram of prompt engineering from Sahoo et al. (2024)

finally evaluating and fine-tuning models to optimize their performance. Understanding these fundamental steps is crucial for researchers and practitioners in order to build NLP systems.

2.2.1 Data Preprocessing

Data preprocessing starts by ensuring that raw text is cleaned, structured, and transformed into a format suitable for analysis and modeling. Given that raw textual data often contains noise, inconsistencies, and redundancies, preprocessing is essential for improving model performance and accuracy. The main stages of NLP data preprocessing may include:

- **Tokenization**: Splitting text into individual words, phrases, or subwords to facilitate further processing. Tokenization helps break text into smaller units, making it easier for models to analyze and extract meaningful information.
- **Stopword Removal**: Filtering out common words (e.g., "the", "is", "and") that provide little contextual meaning. This reduces computational complexity and enhances the focus on significant terms.
- **Stemming and Lemmatization**: Reducing words to their root forms to standardize variations (e.g., "running" to "run"). Stemming involves chopping off word

endings, while lemmatization maps words to their base dictionary forms.

- **Part-of-Speech (POS) Tagging**: Assigning grammatical labels (e.g., noun, verb, adjective) to words in order to enhance syntactic understanding. POS tagging helps NLP models in processing word roles within sentences, improving text comprehension for downstream tasks.

- **NER**: Identifying key entities such as names, organizations, and locations within the text. This type of information extraction can be very useful when performing knowledge retrieval.

- **Text Normalization**: Converting text into a consistent format by handling contractions, misspellings, and lowercase transformations.

- **Vectorization**: Transforming text into numerical representations using techniques such as BoW, TF-IDF, and word embeddings (e.g., Word2Vec, GloVe, or BERT embeddings). This step enables ML models to interpret textual data in a structured numerical format.

- **Noise Removal**: Eliminating unwanted characters, punctuation, special symbols, and excessive whitespace. This helps in improving text quality and ensures that models do not learn from irrelevant information.

- **Handling Imbalanced Datasets**: Addressing dataset imbalances by using techniques such as oversampling, undersampling, or synthetic data generation to ensure fair model training and evaluation.

Proper data preprocessing not only makes textual data more organized, but also enhances the efficiency and effectiveness of NLP models. However, the choice of which step of

data preprocessing to apply is dependent on the chosen model and task. For instance, stopword removal and lemmatization are often recommended when using term frequency-based ranking functions such as TF-IDF, as they improve text retrieval by reducing redundancy and helps the function group words more accurately. However, for transformer-based models such as BERT and GPT, minimal preprocessing is often recommended. These models were pretrained on raw, minimally processed text and rely on contextual embeddings that capture syntactic and semantic nuances (including word order). Therefore, steps like stopword removal, stemming, lemmatization, and manual vectorization are often avoided. On the other hand, tokenization becomes essential to ensure proper text segmentation and representation. For these reasons, proper research and a clear understanding of the underlying mechanisms of the chosen model are essential prerequisites to properly choose the correct preprocessing techniques to use or leave out.

2.2.2 Feature Engineering

Feature engineering involves transforming raw textual data into meaningful features that the models can base their predictions on. Common techniques include n-gram extraction where sequences of n words from the text are captured. TF-IDF vectors are useful as they assign weights to terms based on their frequency in a document relative to their overall occurrence across the corpus. This approach highlights the significance of less common words, making them more influential in the representation, while simultaneously reducing the influence of very common words, such as "the" or "is," which typically contribute little semantic value. However, in DL models, word embeddings have become the more prominent feature engineering method, where words or phrases are mapped to dense vector representations in a continuous space. These embeddings, such as Word2Vec and GloVe, capture semantic relationships between words, improving context understanding. More

advanced models, like BERT and GPT, utilize contextualized embeddings, where the representation of a word is influenced by the surrounding text.

2.2.3 Model Selection

This phase involves selecting between traditional ML approaches, DL architectures, and pretrained transformer models. Classical ML techniques such as Naïve Bayes, SVM, Decision Trees, and Random Forests can work for smaller datasets and structured tasks like spam detection and sentiment analysis. However, modern NLP relies heavily on transformer architectures, which leverage self-attention mechanisms for context understanding and generalization. Some widely used transformer models include:

- **BERT (Bidirectional Encoder Representations from Transformers)**: A model designed to understand context bidirectionally. It is widely used for tasks such as text classification, NER, and Q&A. Compared to more recent LLMs, it is relatively not a computationally heavy model to run and can be deployed on various consumer hardware that is equipped with a GPU.
- **RoBERTa**: An optimized version of BERT with improved training strategies that has been shown to be more effective for various NLP tasks.
- **GPT**: Developed by OpenAI, GPT models are a family of LLMs that focus on generative tasks. They are a strong candidate for applications that involve text generation, conversational AI, and Retreival-Augmented Generation (RAG).
- **Llama**: A family of open-source LLMs optimized for generative tasks, making them adaptable for various NLP applications, similar to the GPT models. However, they differ from GPT models primarily in that they contain fewer parameters, making

them comparatively more efficient in terms of computation and energy consumption. Additionally, unlike GPT models, they can be downloaded and deployed locally

Given the differences between the models listed, the selection of NLP model will have a significant impact on the task results. After identifying the most suitable model for the given task, the training process can then be initiated.

2.2.4 Evaluation and Fine-Tuning

Fine-tuning is the training process where an AI model is optimized for a specific task. It often involves adjusting hyperparameters, optimizing learning rates, and employing techniques such as dropout and regularization to prevent overfitting. Most of the fine-tuning in NLP systems is done in a supervised manner, where the model will rely on task labels to optimize its parameters. In order to assess wether the fine-tuning process was effective, it is important to select the right evaluation method to ensure accuracy and reliability. The choice of evaluation metric depends on the task at hand. For sequence generation tasks, metrics like Bilingual Evaluation Understudy (BLEU) and Recall-Oriented Understudy for Gisting Evaluation (ROUGE) are often used to assess translation and summarization quality. For classification problems, metrics that are often used include accuracy, precision, recall, and F1-score. The F1-score balances precision and recall to provide a measure of a model's effectiveness when accuracy alone may not provide a reliable performance measure. Precision refers to the proportion of correctly predicted positive cases out of all positive predictions, while recall measures how well the model identifies actual positive cases in comparison to all the positive labels. A high precision with low recall indicates that the model is conservative in making positive predictions, while high recall with low precision suggests that the model is making too many false positive predictions. These factors make

the F1-score especially valuable for imbalanced datasets where we are specifically interested in the model's performance in regards to positive labels.

The F1-score is calculated as follows:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.1}$$

where:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2.2}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{2.3}$$

2.3 Explainable AI

While DL models are becoming increasingly popular in the field of NLP, their decision-making processes remains complex and ambiguous, making it difficult to understand how their predictions are made. This lack of transparency has raised concerns regarding fairness, accountability, and trust in AI-driven applications, particularly in high-stakes domains such as legal decision making. XAI aims to find a solution to these concerns by providing interpretability methods that allow users to gain insights into how these models function and why they make certain predictions.

2.3.1 Local and Global Explanations

Local XAI methods provide explanations for individual model predictions, with a focus on specific aspects such as data points, while global XAI methods have the goal to explain the overall decision-making logic of a model. According to (Vale et al., 2022), global interpretability "explains the whole logic of a model and the reasoning behind all possible outcomes," while local interpretability "explains model characteristics and the impact of input features for a specific prediction".

Local explainability is particularly useful when examining why a model made a certain prediction for an individual instance. Methods like Local Interpretable Model-Agnostic Explanations (LIME) generate explanations by approximating the local decision boundary of the model. This technique helps users understand the specific features that contributed to a particular decision. On the other hand, global explainability provides a broader understanding of how a model operates across all instances. This approach explains the model predictions through the most important rules that it learned from the training data. It will represent the explanation through the structure and parameters of the model, and can help identify general trends in model behavior. An example of global explainability would be the rules encoded in the decision tree model through its path from the root node to the leaf nodes.

While both local and global explainability methods have their advantages, they also come with challenges. Local explanations may be unstable and inconsistent across different instances, whereas global explanations can sometimes be too abstract to capture nuanced decision-making processes in complex models.

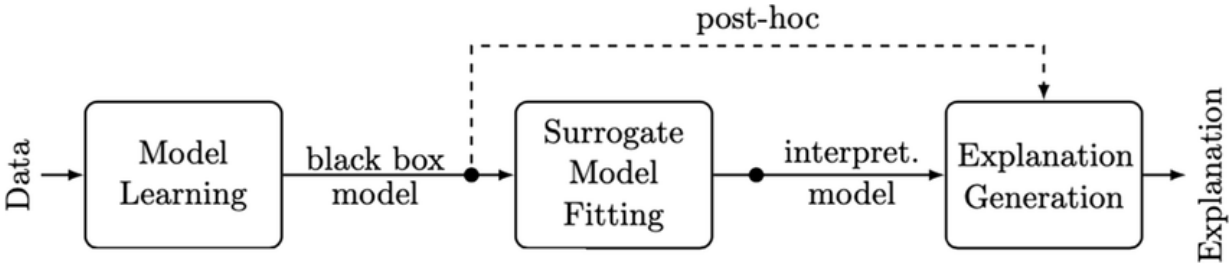

Figure 2.3: Diagram of Post-Hoc Interpretability methods by Vale et al. (2022)

2.3.2 Instrinsic Explanations

Intrinsic interpretability refers to models that are inherently explainable due to their structure and design, often described as white-box models, by directly intergrating interpretability into the architecture. An approach for this is intermediate representation explanations, where the model is constrained to produce a meaningful intermediate representations (Madsen et al., 2022). An example of this are Neural Modular Networks (NMVs) (Azam, 2000), a neural network that is divided into multiple subsystems of neural networks whose outputs are then intergrated together. This provides intermediate representation through the analysis of each output of the separate modules of the network. However, NMVs are not necessarily fully intrinsically interpretable, since as to how their intermediate representation is produced still relies on individual black-box neural network predictions.

Despite their advantages, models using intrinsic methods often lack the flexibility and predictive power deep neural networks as their architecture is by design constraint and can often be task dependent. As a result, it is not always possible to achieve this type of interpretability, while also having high performance (Madsen et al., 2022).

2.3.3 Post-Hoc Explanations

Post-hoc interpretability methods provide explanations after a model has been trained, offering insights into how decisions were made without altering the model's structure. This is different from their intrinsically interpretable models counterpart, which requires interpretability to be inserted inside the architecture of the model itself. This makes post-hoc methods model-agnostic, as they can be applied to a wide range of neural architectures without altering their structure. This greater amount of flexibility has made them a popular option for DL black-box systems.

A typical way post-hoc methods are applied is by analyzing which features or input components had the most influence on a model's prediction. For instance, feature attribution techniques assign importance scores to specific individual features. These individual features can be words or tokens in the context of NLP, and can help in understanding why a certain phrase or term led to a particular classification. Another approach involves generating counterfactual examples by creating slightly modified inputs that would result in a different prediction, highlighting the key elements that drive the model's decision-making process and how much the input would need to be altered for the prediction to be different. Additionally, some methods examine how training data impacts predictions, revealing which instances had the greatest influence on the model's learned patterns. In transformer-based architectures, attention mechanisms have also been examined as a form of explanation, though their validity as a direct interpretability tool remains debated as multiple different papers found contradicting conclusions for their effectiveness as explanations (Madsen et al., 2022).

2.4 Summary

This chapter provides an overview of of the evolution, methods, and key processing stages of NLP. It highlights the the role of different categories of AI in advancing NLP models. Various approaches, from rule-based to statistical and DL methods, are discussed, with a focus on pretraining techniques, like MLM and NSP for models such as BERT and GPT. The chapter also outlines the NLP workflow, covering data preprocessing, feature engineering, model selection, and evaluation metrics. In addition, it explores XAI, covering topics such as the differences between local and global explainability, intrinsic interpretability, and post-hoc explanations, which help uncover a model's decision-making processes. This chapter sets the foundation for understanding NLP capabilities that will have implications within this thesis.

Chapter 3

Literature Review

As the field of NLP continues to advance, various research efforts have focused on improving model training, explainability, and domain-specific adaptation. This chapter presents the research endeavors which this study was based on, and the gaps in the research that this thesis aims to solve.

3.1 Entailment Classification

Entailment and similarity classification are widely studied topic in NLP to measure a model's understanding of language. Where the measurement of entailment differs from that of similarity is that to measure entailment requires language models to determine whether the pair of text logically follow each other, rather than merely assessing their resemblance. A popular entailment classification dataset is the Stanford Natural Language Inference (SNLI) corpus (Bowman et al., 2015) which consists of 570,152 human-labeled sentence pairs. Each pair in the dataset is labelled in one of the three categories, which

consists of entailment, contradiction, and neutral. Some of the limitations of SNLI includes its lack of diversity of sentence structures and language styles found in more complex text. This is due to the fact that the sentences of SNLI were derived from image captions, which limits the description of the sentences to concrete visual scenes. The dataset Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2017) was introduced in order to address these shortcomings by dividing its 433,000 sentence pairs into ten different genres, creating a broader range of linguistic texts for models to be trained and evaluated on. A limitation of many standard natural language inference datasets, including both SNLI and MNLI, is that models trained on them can exploit spurious statistical patterns. The Adversarial Natural Language Inference (ANLI) dataset (Nie et al., 2019) was introduced to address this issue by challenging models with adversarially constructed examples that expose their weaknesses. This was done by introducing a human-and-model-in-the-loop in each round of dataset creation, where difficult examples that state-of-the-art models misclassified were created and incorporated into subsequent training rounds to enhance the model's understanding. The dataset was built in three progressively harder rounds to push the trained models toward more robust language understanding of sematic relationships for entailment.

However, while datasets like SNLI, MNLI and ANLI cover a broad range of domains, they may not always align with specialized fields requiring more domain-specific knowledge. The MedNLI dataset (Romanov and Shivade, 2018) was created to address this challenge in the clinical domain specifically. This dataset provides expert-annotated sentence pairs that were constructed using clinical notes from the MIMIC-III database, where sentences were extracted from patient histories and annotated by medical professionals. As for the legal domain, the Competition on Legal Information Extraction and Entailment (COLIEE) dataset (Goebel et al., 2023) was created to address the lack of law-related tasks and

datasets for the legal domain. The legal texts used for the legal entailment classification task, which is the task 2 of the COLIEE competition, are extracted from case law documents from the Federal Court. As a result, the texts found in the COLIEE dataset are designed to illustrate the complexities of what is found in legal documents, through their detailed content, domain-specific terminology, and legal reasoning.

3.2 Model Training and Adaptation

As for adapting models to the COLIEE 2023 dataset, the study by Kim et al. (2024), a team that includes the creator of the COLIEE competition, published a study on adapting transformer models for the task 1 and the task 4 of the 2023 dataset. For task 1, which is a legal document information retrieval task where relevant cases that are cited in the query case must be retrieved, the study used a sentence transformer to generate embeddings for each paragraph of the query and candidate documents in the training set, which it then used to construct a histograms of cosine similarities between queries and candidate documents. A Gradient Boosting binary classification model was trained on those inputs to then perform classification on the test set in order to retrieve the relevant candidate document based on the query. For task 4, a statute law entailment task based on a corpus of legal questions drawn from Japanese Legal Bar exams, the study fine-tuned a DeBERTa-large model trained initially on SNLI and MNLI, and then subsequently fine-tuned on the labeled training data of the competition. Out of the two tasks, the method for task 4 was the most successful, achieving third place out of all the eight participating teams. Additionally, they experimented with knowledge distillation techniques by evaluating a MiniLM-based model trained via teacher–student paradigm, with the larger DeBERTa models as the teacher, which demonstrated promising performance despite its smaller size.

The team also experimented with four variants of the LEGAL-BERT models. The study noted that most underperformed and also highlighted a mismatch between the corpus that the LEGAL-BERT variants were pre-trained on, consisting of legal documents from the United States and Europe, and the Japanese law corpus used in task 4 of the competition.

Recent research efforts question whether continued pretraining on domain-specific or task-specific data can further enhance performance. Further pretraining RoBERTa models on domain-specific data was explored in a study by Gururangan et al. (2020), where it adapted a RoBERTa using the approach of domain-adaptive pretraining, and task-adaptive pretraining. Domain-adaptive pretraining involves continuing the pretraining phase on a large corpus of unlabeled text, based on a specific domain. In the context of the study, it examined the domains of biomedical publications, computer science papers, news articles, and product reviews. The results demonstrate that further pretraining within these domains leads to consistent performance improvements, particularly when the target domain differs significantly from the original pretraining corpus of the model. The paper also explores task-adaptive pretraining, which fine-tunes a model on a unlabeled task-specific dataset rather than a broad domain corpus, which they showed provides performance gains, often matching or even exceeding those of domain-adaptive pretraining.

As for models pretrained specifically on a domain, the ClinicalBERT model (Huang et al., 2019) was developed for the medical domain in order to address the challenges of processing unstructured clinical text, leveraging the BERT architecture but pretrained specifically on clinical notes from the MIMIC-III dataset. Unlike standard BERT, which is trained on general language corpora, ClinicalBERT incorporates domain-specific terminology and medical jargon, allowing it to capture more accurate representations of clinical concepts. The study evaluates ClinicalBERT on the tasks of clinical language modeling and hospital readmission prediction. In the language modeling task, ClinicalBERT outper-

forms standard BERT, achieving higher accuracy in MLM with a score of 0.857 over 0.495, and also performed better for NSP with an accuracy of 0.994 over 0.539. For hospital readmission prediction, ClinicalBERT is compared against traditional methods such as BoW, Bidirectional Long Short-Term Memory (BiLSTM) with Word2Vec embeddings, and standard BERT. The results show that ClinicalBERT outperforms these baselines, achieving an AUROC of 0.714, an AUPRC of 0.701, and an RP80 score of 0.242, indicating improved predictive performance in determining whether a patient will be readmitted within 30 days. The study also argues that the model's interpretability is enhanced through its attention mechanisms, which highlight important terms in clinical notes that contribute to predictions. For example, the study found that terms in the clinical notes like "chronic heart failure" receive high attention weights, and are predictive of patient readmission classification. Overall, ClinicalBERT is an example of a BERT model pretrained for a specific domain to address the limitations of standard BERT.

A pretrained variant of BERT for the legal domain is LEGAL-BERT (Chalkidis et al., 2020), an adaptation of the BERT model specifically designed for legal text processing. The study explores three main strategies for domain adaptation. The first strategy is using the base BERT with no further pretraining, the second using further pretraining of BERT on legal corpora, and third using pretraining from scratch on legal text. The dataset used for training LEGAL-BERT includes a diverse range of legal documents, totaling 12GB, that encompass legislation, court cases, and contracts from Europe and the United States. The study demonstrates that models pretrained on legal-specific text outperform general BERT models in tasks where in-domain knowledge is more important, such as multi-label classification in the ECHR-CASES data, and named entity recognition for lease details in the CONTRACTS-NER data. The paper also highlights that smaller models trained from scratch on legal text, such as LEGAL-BERT-SMALL, achieve comparable performance to

larger models while being significantly more efficient. These findings suggest that specialized pretraining for the legal domain is beneficial for legal NLP applications, improving both accuracy and computational efficiency. Another study that explores strategies to optimize BERT for legal documents is from Limsopatham (2021). This study focuses on handling long-form legal texts and adapting pretraining approaches for improved performance. The study evaluates different models, including BERT, LEGAL-BERT, and RoBERTa, on the ECHR Violation Dataset and the Overruling Task Dataset. The study finds that pretraining on in-domain legal documents enhances classification accuracy. One of the main challenges addressed is the length limitation of standard BERT models, which struggle with legal documents exceeding 512 tokens. To mitigate this issue, the study compares approaches such as truncation, hierarchical BERT models, and alternative architectures like BigBird and Longformer, which incorporate specialized attention mechanisms to process longer documents. The results indicate that models explicitly designed for long texts outperform standard BERT adaptations. Furthermore, the study demonstrates that models pretrained on legal-specific corpora, such as ECHR-Legal-BERT and Harvard-Law-BERT, achieve superior classification performance compared to models pretrained on general corpora. However, while studies such as (Chalkidis et al., 2020) and (Limsopatham, 2021) explored further pretraining on legal corpora, they focused on legal texts from the United States and Europe, and did not examine its impact on more recent architectures of large language models. Our work addresses this gap by evaluating domain-specific further pretraining on both small and large language models, and also using Canadian legal texts for the further pretraining phase.

As for adapting large language models for a specific task, the study (Brown, 2020) demonstrates that the GPT-3 large language models can effectively learn to perform new tasks from a handful of examples provided within the input prompt, a process that they

refer to as in-context learning. This technique of adapting to a task using prompting is a major departure from conventional approaches of fine-tuning for each new downstream task. Rather than relying on additional supervised fine-tuning, GPT-3 was shown to be able to generalize across a wide variety of NLP tasks through few-shot, one-shot, and zero-shot at inference time. The results indicate that as the size of the GPT model increases in the range from 125M parameters to 175 billion parameters, its ability to adapt to unseen tasks from context alone improves significantly. The study also shows that when looking at the aggregate performance for all 42 benchmarks, zero-shot performance improves with model size. However, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at few-shot learning than zero-shot learning. However, the study indicates that prompting showed limitations. Few-shot performance is below that of fine-tuned models in some areas where it struggles when asked very specific facts whose knowledge is outside its training distribution knowledge, such as questions related to fine-grained knowledge from Wikipedia articles, or challenging datasets such as the ANLI dataset requiring complex and subtle reasoning. Recent studies have also explored comparisons between the two strategies of fine-tuning and prompt engineering for adapting large language models to specific tasks. The study from Trad and Chehab (2024) compared these approaches in the context of phishing detection, showing that prompt engineering can achieve decent performance in relation to their minimal setup requirements, but are not as effective as fine-tuned models dedicated to the task of detecting phishing URLs. The study from Pornprasit and Tantithamthavorn (2024) similarly explores fine-tuning and prompt engineering GPT-3.5 for code review automation, and found that fine-tuning outperformed prompt engineering. To the best of our knowledge, no such comparison analysis has been done for the legal domain, which our works aims to contribute to.

Recent work by Wang et al. (2024) provides a comprehensive evaluation of state-of-the-

art LLMs, such as OpenAI's o1 and Mistral-7b, applied to legal tasks across data from both the United States and China. The results in the human evaluation of the study showcase that while LLMs can exhibit a strong alignment between their generated judgments and the legal reasoning underlying the real case outcomes, they still face challenges in handling complex legal reasoning and domain-specific terminology. For example, while models like GPT-4o and O1-preview achieved high human evaluation scores of 3.54 and 4.08 on English legal texts, their ROUGE and BLEU scores remained relatively low. In addition, Lawyer-Llama-13B achieved a strong ROUGE-2 score of 0.38, but a lower human evaluation score of 2.23. The study highlights the fact that while LLMs demonstrate potential in their processing abilities for legal text, their still remains a need for improved training methodologies that integrate domain-specific legal knowledge and strengthen reasoning capabilities. This thesis situates itself in the current legal information research landscape by addressing domain adaptation and reasoning challenges in Canadian legal data, while incorporating explainability with legal entailment predictions.

3.3 Explainable AI

Common methods that have been used to explain black-box models in the domain of XAI are LIME (Ribeiro et al., 2016) and SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) which are perturbation-based methods. Despite their wide usage, many studies have shown their drawbacks and limitations. In NLP, the study by Heyen et al. (2024) evaluates LIME explanations across DeBERTaV3 models of increasing sizes on the MNLI and e-snli datasets. The study finds a misalignment between the explanations produced by the models and their true internal decision processes. While the faithfulness of the explanations increases with model size, the plausibility, measured as the alignment

between the model explanations and the true human explanations, does not increase. This lack of correlation between the faithfullness and plausibility indicates that the explanations do not fully capture the model's actual decision-making process, otherwise improvements in faithfulness would naturally lead to greater plausibility. More broadly, the results of the study also indicate a lack of expressiveness in token-based highlighting techniques for explainability, as these methods struggle with the necessary abilities to represent higher reasoning concepts in entailment classification, such as token dependencies and logical relations. The reliability of LIME and SHAP is also questioned in Alvarez-Melis and Jaakkola (2018) which examines the robustness of these techniques. Robustness, in the context of interpretability, refers to the expectation that similar inputs should result in similar explanations. Through evaluations on different datasets including the UCI repository (Lichman and Bache, 2013) and COMPAS dataset (Bao et al., 2021), the study demonstrates that LIME and SHAP often produce unstable explanations, particularly when applied to non-linear models such as neural network classifiers. The study shows that even slight perturbations to the input can result in different explanations, despite having minimal effect on the model's actual prediction. To quantify this evaluation, the authors propose a local Lipschitz continuity metric, designed to measure the sensitivity of explanations to small input variations. The study found that both LIME and SHAP scored high, indicating a lack of stability and robustness from an interpretability standpoint. In addition, another study that found concerns about robustness was Kommiya Mothilal et al. (2021) which highlights fundamental limitations in how LIME and SHAP rank features when explaining model predictions. The study challenges the assumption that high feature importance in LIME and SHAP reliably indicates causal influence on the prediction by showing that the top-ranked features identified by these methods are often neither necessary nor sufficient for the prediction, meaning that changing these features does not consistently alter the outcome, nor does keeping them fixed guarantee the same prediction.

A new method of explainability that is emerging is the usage of prompts for generating explanations which leverages the generative abilities of models pretrained on a large corpus. An example of this shift is presented in the work of Zhang et al. (2025), which moves away from the rigid knowledge-graph paths typically found in traditional product search systems, instead exploring natural language explanations generated by the system. By utilizing prompt-based methods, this approach addresses the limitations inherent in traditional path-based explanations, which often lack semantic flexibility and can hinder users' understanding. The study calls its framework P-PEG and uses subsets of the Amazon dataset including Electronics, Kindle Store, CDs & Vinyl, and Cell Phones & Accessories. The framework operates by retrieving suitable products based on information extracted from the user's query, historical search sequences, and structured knowledge obtained from a Knowledge Graph. Additionally, relevant user information and query content are combined to formulate a prompt. This prompt serves as input for a pretrained language model, generating natural language explanations for the retrieved search results that closely align with user expectations and search intent. Another study that includes prompt explanations is from Leippert et al. (2024), which examines the identification and generation of clarification requests in knowledge-based question answering systems through fine-tuning, prompt tuning, and manual prompt engineering. The study used the CLAQUA dataset, modelled as a binary classification task where the model must predict whether a given query needs a clarification request based on the conversation context and entity descriptions. For the prompt engineering section of the study, the prompts that were employed were zero shots prompts, ranging from simple to detailed informative instructions on the task. One of the insights from the study is that while manual prompt engineering with GPT-3.5 was less effective in overall classification performance, it revealed potential for generating explanations when informative prompt instructions were used. This suggests that while prompt-based explanations could not beat fine-tuned language models, or prompt-tuning

based methods, they offer interpretability by providing reasoning about why a query needs clarification. However, these explanations are not without challenges, as hallucinations, omissions, and incoherence were observed in some of the results. Another study that explores explainability through prompts is Paranjape et al. (2021). The study explores contrastive explanations as a method for improving explainability in commonsense reasoning tasks. Instead of generating free-form textual justifications, the approach in the study engineered the prompts to produce contrastive explanations where the model must explicitly contrast different possible answers when providing its explanation. The study argues that this method aligns with human reasoning, where explanations often highlight why one option is more valid than another, rather than just providing standalone justifications. To achieve this, the study uses contrastive prompt templates, such as "Q are _ while P are _" to guide the model in structuring explanations that emphasize distinguishing attributes between answer choices. The generated explanations not only improve model explainability, but the results also demonstrate that contrastive explanations improve classification performance in commonsense reasoning tasks, outperforming previous self-talk approaches and clarification-based methods. The contrastive explanations also scored well on faithfulness when flipping contrastive statements to assess the model's reliance on these explanation.

An important aspect of explainability research is the development of methods that not only provide accurate explanations, but also ensure that these explanations are meaningful and trustworthy to human users. The SELFEXPLAIN framework Rajagopal et al. (2021) is an example of this, as it introduces a self-explaining neural model that enhances interpretability by incorporating both global and local explanation layers. The study evaluates the effectiveness of these explanations based the four metrics of sufficiency, adequacy, understandability and trustability. The first metric of sufficiency measures whether ex-

planations alone are indicative of the model's prediction, evaluated through the Faithful Rationale Extraction from Saliency tHresholding (FRESH) pipeline, which demonstrated that explanations from SELFEXPLAIN showed higher predictive performance than baselines. The FRESH pipeline in the study uses a BERT model trained to perform classification using the explanations alone. The pipeline aims to measure how indicative and sufficient the explanations generated are in relation to the prediction of the label. The second metric, adequacy, evaluates whether explanations provide adequate justification for the model's predictions. Using human evaluations, SELFEXPLAIN achieved a 32% increase in perceived justification adequacy compared to saliency-based methods. The third metric, understandability, evaluates how easily human users can comprehend the explanations provided. The fourth metric, trustability, analyzed the explanations using a Likert scale where human annotators rated explanations on how they improved their trust in the model. These four metrics will similarly serve as the basis for evaluating the explainability methods employed in this thesis.

For XAI in the domain of law, the survey by Richmond et al. (2024) showcases the important challenges regarding transparency and explainability. While the applications of XAI in law has its limitations, proponents argue that XAI can provide decision support to judges, assist litigants in demonstrating the legality of algorithmic decisions, and help defendants contest AI-based administrative decisions. The survey shows that traditionally, legal reasoning has been modeled using symbolic AI, which relies on human-created rules to ensure interpretability. However, the shift from symbolic AI to machine learning in AI applications has led to a rise in more opaque decision-making systems, where deep neural networks are increasingly used to extract legal information from textual data, despite their black-box nature and large parameter spaces. While these newer legal AI models demonstrate strong performance, they remain difficult to interpret. As a result, the lack

of suitable explanations for DL models has been a major barrier to the adoption of these technologies in both public bodies and private law firms, which rely on legal reasoning that is inherently rule-guided and where decisions must align with established legal norms. For these reasons, rule-based systems are still the most prevalent form of legal AI expert systems. However, this creates a trade-off between explainability and accuracy, as early legal AI models, such as rule-based systems and decision trees, are highly interpretable but exhibit lower performance, in contrast to deep learning models that perform with high accuracy but are less interpretable. In order to have both high accuracy and interpretability, the survey highlights the necessity to advance research in regards to making deep learning models more understandable and transparent to professionals in the domain of law. To address this gap, this thesis explores a novel approach in which fine-tuned black-box classification models are repurposed for explanation generation, incorporating LLMs by using a method of swapping language modeling heads. In addition, we systematically evaluate black-box explainability methods in the context of legal text entailment classification, ensuring that high-performance models can provide meaningful and interpretable justifications in legal decision-making.

3.4 Summary

This section explored various datasets used for entailment classification, highlighting the evolution from SNLI to more complex datasets like MNLI, ANLI, and domain-specific corpora such as MedNLI and COLIEE. It then examined approaches for improving model performance through domain-specific pretraining and large language model adaptation. The section on explainability addresses the limitations of post-hoc methods like LIME and SHAP , which struggle with robustness and faithfulness, and introduces emerging ap-

proaches that leverage prompting to enhance model transparency. Within the legal domain, the review highlights the trade-off between interpretability and accuracy, emphasizing that while rule-based AI systems remain widely used due to their transparency, deep learning models significantly outperform them in predictive tasks despite their opacity, which poses a barrier to adoption in legal settings. The chapter also dives into the lack of systematic evaluations comparing prompt engineering and fine-tuning methods in the legal domain, limited exploration of domain-specific pretraining for Canadian legal texts, and insufficient investigation into methods for black-box model explanations in the legal domain.

Chapter 4

Methodology

This chapter provides a detailed explanation of the methodology employed in this study, detailing the system architecture, datasets, preprocessing techniques, and model training strategies used for legal entailment classification. It explores various classification and explainability methods, including fine-tuning and prompt engineering for text classification, and LIME and generated explanations for providing insights into model decisions. Furthermore, the study also explores human and automated evaluations of explainability techniques. By systematically assessing both classification accuracy and explainability, this research aims to contribute to the ongoing effort to bridge the gap between high-performance black-box models and the transparency required for legal AI applications.

4.1 General Architecture

The system architecture developed in this study is designed to enable a single language model to perform legal text classification and explanation. The core of the system is built

on models utilizing the transformer architecture. An overview of the system can be found in the figure 4.1

4.1.1 Data Preparation

The data preprocessing phase begins with acquiring the necessary input data for the model. The dataset is categorized into two types:

- **Pretraining Data**: This consists of an unlabeled domain-specific legal corpus obtained through the Canadian Legal Information Institute (CANLII) API, which serves as the corpus for further pretraing.
- **Fine-tuning Data**: This labeled dataset is sourced from the COLIEE 2023 competition and is comprised of paragraphs from Federal Court cases. It is structured in the format of sets of paragraphs associated with a fragment from a court case, with each set assigned a training label.

This data is then preprocessed through tokenization, truncation, batching, and padding when necessary. The text from the paragraphs is also formatted in pairs. The dataset and preprocessing steps are described in more details in 4.3 and 4.4 respectively.

4.1.2 Text Classification

The system's architecture utilizes transformer models in order to perform classification on the pairs of legal text. These models undergo the following methods:

- **Domain-specific further pretraining** For the models RoBERTa and Llama 2, they are further pretrained on a corpus of documents from the Federal Court, Federal Court of Appeal, and Supreme Court of Canada using MLM.

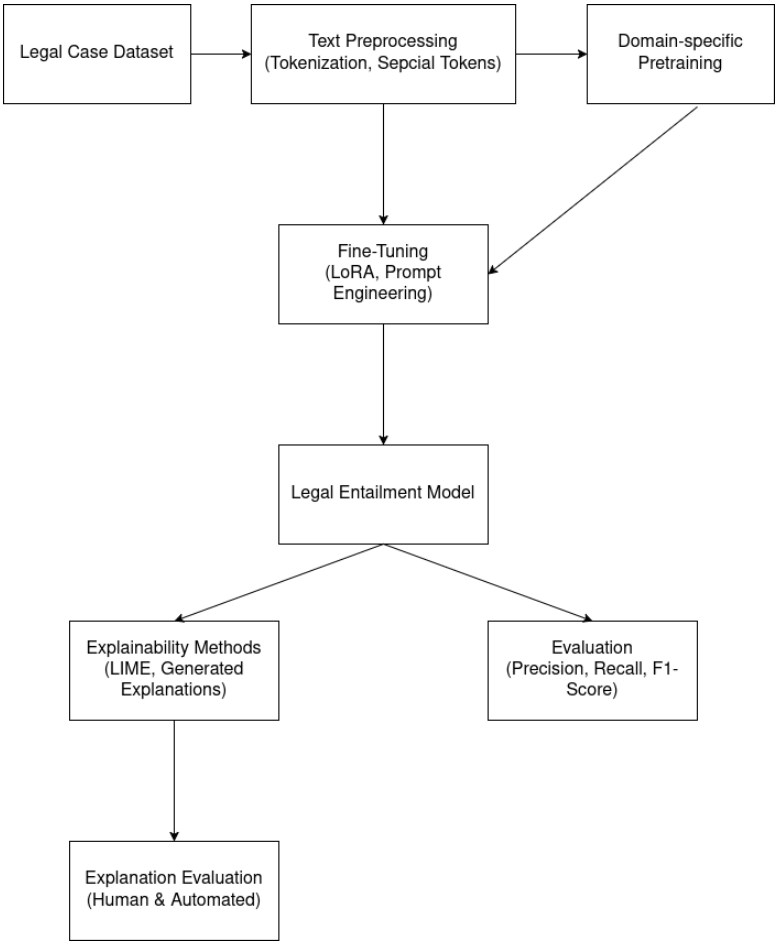

Figure 4.1: Diagram of legal entailment system.

- **Model adaptation** Models in this phase use supervised learning in order to be adapted to perform binary classification with entailment and contradiction labels as output.

These training methods are utilized in order to ultimately enable the transformer models to classify wether or not two legal texts entail each other. The classification performance is then evaluated using the metrics specified in Section 4.5.7.

4.1.3 Explainability

Since DL models function as black boxes, integrating interpretability techniques is important for getting a better understanding on why the classification predictions are made. To ensure transparency, the system incorporates different explainability approaches depending on the model architecture to provide insight into the predictions:

- **Token-based Attribution**: For BERT-based models, the system utilizes a feature attribution technique to highlight key words that influence the classification decision.
- **Generated Explanations**: For LLMs, explanations are generated in natural language using prompt engineering, providing justifications for classification predictions.

To assess the quality and effectiveness of the explanations, two evaluation methods are applied:

- **Human Evaluation**: Explanations are evaluated based on adequacy, understandability, and trustworthiness.
- **Automated Evaluation**: Explanations are evaluated using a quantitative measure to determine their relevance and consistency with classification outcomes.

By integrating these explainability techniques, the system enables the same DL model that performs classification to also generate justifications for its entailment or contradiction predictions on text pairs.

4.2 Experimental setup

For the hardware configuration, all the experiments were done using Intel Xeon Gold 5218 CPU with 64 processing cores. For the GPU, the server was comprised of two NVIDIA Tesla V100 GPUs, each with 32GB of memory. The implementation of the models and experiments was coded in Python. The most important libraries used in experiments are:

- **Transformers (Hugging Face)**: The library used for loading and fine-tuning open-source pretrained transformer-based models. This library was also used to acquire important tools such as text tokenization functions and the AdamW weight decay optimization algorithm.
- **PyTorch**: The machine learning library used for DL model training and inference. This library enabled setting up important model training aspects such as epochs, gradient computation, and data batching.
- **CUDA**: A proprietary parallel computing platform and API developed by NVIDIA, used in the context of this study for GPU acceleration in both training and inference of DL models.
- **NumPy**: The library used for numerical and scientific computing of vector operations and array manipulations.
- **Pandas**: The library used for structuring and manipulation data through dataframes.
- **OpenAI**: The API used to access and train OpenAI's GPT-4o model.
- **LangChain**: The framework used for efficient prompt engineering, which enabled the construction of templates that could have data inserted in them such as the legal

text pairs and labels, and also the training examples used for constructing few-shot prompts.

4.3 Data

The experiments from this study were built from two datasets. The first dataset used for pretraining consists of legal cases obtained through the CANLII API. The dataset includes 6742 cases from the Federal Court of Appeal, 27481 cases from the Federal Court, and 1392 cases from the Supreme Court of Canada. This amounts to a total of 35,617 cases for pretraining, forming a Canadian court case corpus with a diverse representation of legal arguments, procedural details, and judicial reasoning.

The second dataset from this study is from the second task of the COLIEE 2023 competition (Goebel et al., 2023). This dataset was used for fine-tuning the models for legal entailment classification. The dataset is composed of queries extracted from court decisions. Each query Q is associated with paragraphs that are from a separate relevant case R, which is divided into n paragraphs $P = \{P_1, P_2, \ldots, P_n\}$. Out of all these paragraphs, the task is to classify which of the paragraphs entails Q. It is important to note that just because the whole document is relevant to the query, this doesn't mean that each paragraph necessarily entails the query. There is specifically 625 queries for the training set, and another 100 queries for the test set. The relevant case associated to each query can be divided on average into 35.22 paragraphs per query in the training set, and 37.65 paragraphs per query in the test set. As a result, the number of contradicting paragraphs per query being higher compared to the numbers of entailing ones. It is worth noting that the entailment classification dataset of COLIEE only contains two categories, entailment and classification, meaning it does not have a neutral label which datasets like SNLI contain.

Type	Train	Test
Number of Queries	625	100
Total Number of Paragraphs	22,018	3,765
Average Number of Entailment Paragraphs per Query	1.174	1.2
Average Number of Contradicting Paragraphs per Query	34.054	36.45
Average Number of Candidate Paragraphs per Query	35.229	37.65

Table 4.1: Analysis of the data

As a result, even if two pairs of text do not showcase explicit contradictions, they will still be labeled under a contradiction in the dataset. While the reason for not including the neutral label is not explained by the authors of the dataset, it can be reasonably assumed that including it would have little impact on the results since the competition uses the F1-score metric for the entailment class. As a result, since the evaluation metric emphasizes exclusively the entailment class, distinguishing a neutral class would have no impact on the scoring.

As for the Explainable AI portion, the dataset used for the human evaluation was composed of a balanced subset of the COLIEE dataset. Due to the time-consuming nature of analyzing legal texts and verifying whether the models correctly identified the relevant aspects of each case, this balanced subset for the human evaluation contained exactly 28 randomly selected samples. For the automated explainability evaluation, the full dataset was used.

4.4 Text Preprocessing

The text preprocessing step involves preparing the data for training and inference after the dataset has been acquired. It ensures that the data is structured and formatted in a way that can effectively be processed by the models. For this study, preprocessing focused

on maintaining the integrity of the original legal texts while ensuring compatibility with downstream model requirements.

4.4.1 Preprocessing Text Pairs

For RoBERTa and Llama models, the preprocessing approach was adapted to fit the architectural requirements of modern transformer-based models, without applying overly aggressive modifications that could compromise the semantic richness of the data. Unlike traditional preprocessing pipelines that heavily modify the text through lowercasing, stopword removal, or lemmatization, these steps were avoided. RoBERTa and Llama employ a subword tokenization approach that handles variations in words that contain capitalization and punctuation, making their performance well suited for raw natural language. This is important in a legal setting as domain-specific nuances, such as abbreviations, named entities, and formatting, should be preserved in the input.

For model training and evaluation, the dataset was further processed by dividing the texts into pairs. For each query Q and its associated relevant case R, individual pairs were created by combining Q with each candidate paragraph P_i from R. This enabled the model to evaluate each candidate paragraph individually in relation to the query. Examples of these pairs can be found in Table 1.

In order to input these pairs of text into the RoBERTa model, it must be able to distinguish between the two different textual components. One way to achieve this is by using the `[SEP]` token to separate the two inputs. Following the formatting approach used in Sekulić et al. (2020), this method allows the model to process the input as a single sequence while preserving a distinction between the query and the paragraph. This meant that the legal text pairs were formatted by concatenating the query and the candidate

paragraph, with the [SEP] token inserted between them.

$$\text{Input Sequence} = \texttt{[CLS]}\ Q\ \texttt{[SEP]}\ P\ \texttt{[SEP]} \tag{4.1}$$

The [CLS] token, which is placed at the beginning of the input sequence, acted as an aggregate representation of the entire sequence and was used by the classification head for predicting whether the paragraph entails the query. The input sequence was also truncated at the end of the text pair if necessary to meet the 512-token maximum limit for RoBERTa models (Liu, 2019).

For Llama, input formatting differs as it does not rely on special tokens like [CLS] or [SEP]. Instead, the input sequence was concatenated using general delimiters, such as <|begin_of_text|> and <|end_of_text|>, which Llama uses to identify the boundaries of the input text. This means that the query-paragraph pairs followed this format for the Llama models:

<|begin_of_text|>Query<|begin_of_text|>Paragraph<|end_of_text|>

The maximum token limit for Llama varies by model size but typically ranges from 2,048 to 4,096 tokens. However, in this context the maximum amount of token limit was 900 for Llama 2 and 800 for Llama 3 due to computational constraints. In the case that the combined input exceeded this limit, the sequence was truncated at the end of the text pair.

4.4.2 Decisions on Preprocessing Steps

Below are typical preprocessing steps found in NLP pipelines, along with explanations of how each was approached in this study.

- **Lowercasing**: Lowercasing was not applied to the language models inputs because the tokenizer of the models are case-sensitive. This sensitivity allows the models to distinguish between contextually significant uppercase and lowercase text items, which can be especially important in legal texts.

- **Stopword Removal**: Stopwords, which are typically removed in traditional NLP pipelines, were retained. In legal texts, even common stopwords like "shall," "may," or "will" can carry significant semantic weight and affect the interpretation of legal statements. The subword tokenization of RoBERTa and Llama takes these terms into account and processes them without manual intervention.

- **Punctuation Removal**: Punctuation was preserved in the inputs because it plays a critical role in structuring legal arguments and sentences. Removing punctuation could obscure certain relationships the texts or alter the meaning of sentences, which is detrimental in a legal text processing context.

4.4.3 Tokenization and Batching

The query and paragraph pairs were tokenized using the tokenizer obtained from the HuggingFace library. The preprocessing step of tokenization involves splitting the text into subword units, applying padding to align sequence lengths within a batch, and truncating any input exceeding the maximum token limit of 512 for RoBERTa and 800 - 900 for Llama models. The tokenized inputs included:

- `input_ids`: Subword token indices that correspond to the model's vocabulary.

- `attention_mask`: A binary vector that specifies the tokens to be attended to by the model, where 1 denotes actual tokens and 0 denotes padding tokens.

Examples of tokenized text for each model can be found in Table 4.2. Once the tokenization step was completed, the tokenized inputs were then batched and paired with a corresponding label of entailment or contradiction for fine-tuning. If the length of the tokenized pair of query and paragraph did not reach the maximum allowed token limit of the model, padding was applied at the end of the input. This ensured that all sequences in a batch had uniform lengths.

4.5 Entailment Classification

This section of the methodology covers the classification portion of the system. This is the section of the system's architecture that performs binary classification on text pairs to determine whether one text entails the other.

4.5.1 Further Pretraining

Further pretraining involves continuing the MLM step of the language models, typically on a more domain-specific set of documents to the task it is being evaluated on. In this study, the selected models underwent further pretraining using the dataset obtained through the CANLII API. This additional training step followed the same MLM objective used during the model's original pretraining, with 15% of the tokens masked at each epoch. While this process could be considered a type of fine-tuning, we classify it as further pretraining for several reasons.

First, the goal of the intermediate step was not directly to optimize for the downstream task of legal entailment classification, but rather to enhance the model's exposure to domain-specific knowledge, such as the structure, terminology, and patterns characteristic

of Canadian legal texts. Second, this step leverages a large unlabeled legal dataset, which is a characteristic of pretraining, wherein fine-tuning typically involves using a smaller, labeled dataset tailored for the specific task, in the case of this study, entailment classification. Finally, this further pretraining step is intended to bridge the gap between the original pretraining corpus, which was comprised of general text data, and the specialized task of legal entailment classification. By continuing the self-supervised MLM objective on domain-relevant data, the goal is to refine the model's linguistic representations in a way that better aligns with the complexities of legal language. Hence, while the process resembles fine-tuning in its task relevance, it remains conceptually and procedurally aligned with the principles of pretraining, as it continues to leverage the self-supervised MLM objective on a large, unlabeled dataset.

4.5.2 Finetuning

Once the pretraining phase is complete, the next step is to adapt the model to the specific downstream task. For the RoBERTa models used in this study, this involved adapting the model to perform entailment classification. This adaptation is achieved by adding a classification head on top of the model, which is the standard approach for modifying BERT-based architectures for classification tasks, given their encoder-based structure (Devlin, 2018).

RoBERTa, like BERT, is designed as a transformer encoder that produces contextualized embeddings, where each token's vector captures its meaning within the broader context of the sequence. In the context of fine-tuning for classification tasks, the most important token is the [CSL] token. This token is the first token of every input sequence, and its hidden state can be used as an aggregate sequence representation. The classification head uses this representation for determining the appropriate prediction by applying a fully

connected neural network on the hidden state of the [CLS] token (Liu, 2019). The output is then processed through a softmax function to produce probabilities for each class, in this case, entailment or contradiction. The classification performance is evaluated using two strategies, one considering only the prediction with the highest confidence, and another evaluating all classification outputs without filtering by confidence. We found that taking the most confident paragraph for each query gave the best results.

$$\hat{y} = \text{softmax}(\mathbf{W} \cdot \mathbf{h}_{\texttt{CLS}} + \mathbf{b}),$$

where:

- $\mathbf{W}$ is the weight matrix of the classification head,
- $\mathbf{b}$ is the bias vector,
- $\hat{y}$ is the predicted probability distribution over the two classes.

The training objective is to minimize the **binary cross-entropy loss**, defined as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right],$$

where:

- N is the number of training examples,
- y_i is the true label for the i-th example,
- $\hat{y}_i$ is the predicted probability for the entailment class for the i-th example.

4.5.3 Finetuning Larger Models

As language models become increasingly larger, finetuning them for specific tasks becomes increasingly challenging due to their vast parameter space, making them very difficult to train on consumer hardware. For this reason, different strategies have been proposed in order to bypass this challenge.

One approach to adapting LLMs for specific tasks is prompt engineering which guides the LLM towards the desired task by carefully crafting prompts that contain clear instructions along with relevant context or examples. Two common prompting methods are zero-shot prompting, where the model receives only a task description and generates responses based on its pre-existing knowledge, and few-shot prompting, which provides a few input-output examples. As shown in the study from Brown (2020), LLMs typically perform better when using few-shot prompting. For this reason, this prompting method was employed in this study. The template for the classification prompt can be found in Table 2.

For tasks involving complexity and a highly specialized domains, optimizing the parameters of the LLM can be a more performant solution than prompt engineering. However, as previously mentioned, finetuning large models with billions of parameters requires a high amount of computing resources. This is because fine-tuning requires the entire model to be loaded into memory, along with additional memory to store gradients for backpropagation, a process that is often too demanding for consumer hardware. LoRA offers a more computational efficient approach to finetuning LLMs without the need to train all parameters. Instead, LoRA works by freezing the original model weights and performs the training on a small set of additional low-rank matrices. Specifically, instead of updating the entire weight matrix W, as we would do in regular finetuning, Lora freezes W and instead uses matrix A and matrix B for training, such that:

$$\Delta W = A \cdot B, \tag{4.2}$$

where:

- $W \in \mathbb{R}^{d \times k}$: The original pretrained weight matrix, which is frozen during fine-tuning.
- $\Delta W \in \mathbb{R}^{d \times k}$: The weight adjustment matrix learned during fine-tuning.
- $A \in \mathbb{R}^{d \times r}$: A low-rank matrix that reduces the dimensionality to r.
- $B \in \mathbb{R}^{r \times k}$: A low-rank matrix that restores the original dimensionality to k.
- r: The rank of the decomposition, typically much smaller than d or k, and chosen as a hyperparameter.

When calculating LoRA, the matrix A and B are matrices that are introduced to approximate the original weight matrix W. This is achieved by estimating the rank of W, which is the number of linearly independent columns contained in W, defined as a hyperparameter r. The closer the estimation of r is to the intrinsic rank of W, the more accurate the matrices A and B will be. This efficiently reduces computation because, ideally, if r is close to the intrinsic rank of W, we now only need to optimize the weights in the smaller matrix A containing the linearly independent columns of W, whose original dimensionality is regained using the matrix B.

Hence, during inference, LoRA integrates the learned weights of A and B to the frozen weights of W:

$$W' = W + \Delta W = W + A \cdot B, \tag{4.3}$$

where:

- $W' \in \mathbb{R}^{d \times k}$: The final weight matrix used for inference.

The matrix A is initialized to random numbers using Gaussian distribution, and B to zero. The values of A and B are then learned through the training objectives and back-propagation. Overall, this results in a significantly smaller number of trainable parameters compared to optimizing all the weights in W, as LoRA fine-tuning effectively discards the linearly dependent columns that do not contribute additional information (Hu et al., 2021).

Similar to standard fine-tuning, LoRA fine-tuning also produces classification outputs associated with confidence scores, represented by probabilities from the model's softmax layer. This also enables evaluation using either the most confident predictions or all predictions without confidence filtering. Again, we found that taking the most confident predictions gave the best results.

4.5.4 QLoRA: Efficient 4-Bit Quantization

While LoRA is able to reduce the number of trainable parameters needed for finetuning, QLoRA is able to extend this method to even further reduce computational requirements. This is achieved by reducing the precision of the model's weights, making the matrices more manageable and efficient for GPU processing.

Quantization reduces the precision of model weights. For instance, it will convert 16-bit floating-point values into 8-bit or 4-bit representations. In the study by Dettmers et al. (2024), the 4-Bit NormalFloat (NF4) format was employed, which is an optimized quantization technique that encodes weights as 4-bit integers. During computation, these integers are later dequantized into approximate floating-point values. Empirical studies

have demonstrated that NF4 outperforms traditional 4-bit quantization methods when applied to LLMs. Notably, it has been shown to achieve performance comparable to 16-bit LoRA fine-tuning, even on tasks involving models with billions of parameters. For this reason, NF4 was used for quantization in this study.

4.5.5 Models

For this study, we selected a diverse set of transformer models varying in architecture, size, and training methodology to explore their effectiveness across the legal text classification task. These models include both encoder-only and decoder-only transformers, ranging from compact, fine-tuned base models to large-scale language models with billions of parameters. The Table 4.3 summarizes the characteristics of each model used, including their architecture, parameter count, openness, and training strategy.

4.5.6 Proposed Experiments

To assess the performance of various models on the task of legal entailment classification, three experiments were conducted. These experiments aim to evaluate both regular language models and LLMs, exploring different training and adaptation strategies for each.

4.5.6.1 Experiment 1: Adapting Large Language Models

The first experiment examines the performance of LLMs, specifically Llama 2, Llama 3, and GPT-4o, for the classification task. Due to computational constraints, regular fine-tuning of these large models is not feasible. This study instead compares two alternative resource-efficient strategies as adaptation strategies, which are prompt engineering and LoRA fine-tuning. These methods enable task-specific tuning without the need for extensive hardware

resources. The experiment is designed to showcase the trade-offs between these approaches, and to identify which strategy offers the optimal performance in classifying true positives.

4.5.6.2 Experiment 2: Effects of Pretraining

The second experiment examines the impact of domain-specific pretraining on model performance, by comparing two models of differing scales, specifically a pair of RoBERTa models and a pair of Llama 2 models. All models are fine-tuned on a legal entailment dataset. However, one model from each pair undergoes additional pretraining on the CANLII corpus of legal cases prior to fine-tuning. This setup enables a controlled comparison between base and further pretrained versions, and offers insights into the value of domain adaptation when working with complex and specialized legal texts. If the further pretrained models outperform their baseline counterparts, it would suggest that targeted pretraining on legal corpora enhances model effectiveness in specialized domains like law, where general-purpose language models may lack sufficient contextual understanding. The results for the fine-tuning of the pretrained RoBERTa model was taken as the average of 10 random seeds. Due to time constraints, we did not pretrain the Llama 3 model, and also due to the significant amount of time to fine-tune Llama 2, its results were taken from one seed.

4.5.6.3 Experiment 3: Ensemble Methods for Legal Entailment

The third experiment focuses on evaluating ensemble methods that combine the predictions of different models. This agreement-based ensemble technique combines multiple models that have been fine-tuned with different datasets and architectures. These ensembles aggregate predictions using a consensus-based criterion, where a final decision is made based

on the agreement among participating models. The goal of this experiment is to determine whether an agreement-based ensemble strategy can outperform individual models by leveraging complementary strengths and reducing individual biases.

Four ensemble configurations are examined in this experiment, which allows us to examine the effectiveness of combining models with diverse strengths for the classification task. We also include a RoBERTa model fine-tuned on the SNLI dataset to explore whether fine-tuning from external and general-purpose natural language inference tasks can contribute to legal entailment classification when combined with other domain-adapted model. Below are the ensembles listed.

- A RoBERTa ensemble combining the pretrained model with a model fine-tuned on the SNLI dataset.
- An ensemble of Llama 2 and the pretrained RoBERTa model.
- An ensemble of Llama 2 and GPT-4o.
- An ensemble of GPT-4o and the pretrained RoBERTa model.

If the ensemble is able to surpass any of its individual counterparts from the previous 2 experiments, then the ensemble can be deemed successful.

4.5.7 Evaluation

For the evaluation of the classification portion of this system, the focus is specifically on the ability of the models to correctly classify pairs of text as entailment. As a result, we used the evaluation metrics of precision, recall, and F1-score. The metrics were chosen to

provide a fair and informative evaluating on the performance of the models through these aspects:

- **Precision**: Measures the proportion of correctly predicted entailments among all predicted entailments. High precision indicates that the model is good at avoiding false positives.
- **Recall**: Measures the proportion of correctly predicted entailments among all actual entailments. High recall indicates that the model is good at identifying true positives.
- **F1-Score**: Provides a harmonic mean of precision and recall, offering a balanced measure of a model's classification performance.

These metrics are well-suited for this study, as the dataset is imbalanced, with significantly more contradiction pairs than entailment pairs. Accuracy alone would not suffice to provide a meaningful evaluation in this context, as a model would be able to achieve high accuracy by predominantly predicting the majority class. Instead, precision and recall provide an evaluation of how well the models identify the minority class without being overwhelmed by the dominant class, and the F1-score provides a balanced measure of both. The best performing models on the evaluation for the classification portion were then chosen for the explainability section of the study.

4.6 Explainability

In this section of the methodology chapter, we detail the methods that allow the explainable portion of the system to gain insight on the predictions of the best performing AI models used in the classification portion of this study.

4.6.1 LIME (Local Interpretable Model-Agnostic Explanations)

LIME is a post-hoc explainability method that provides interpretable and local explanations for individual predictions made by a machine learning model. LIME works by perturbing input data and generating multiple synthetic samples to observe corresponding changes in the model's predictions. The goal of the explanations created by LIME is to highlight the most influential features that contributed to a given prediction.

LIME is conceptually similar to SHAP in that both provide local explanations based on feature importance. However, as noted by Stilwell and Inkpen (2024), SHAP does not necessarily outperform LIME in the context of text classification. For this reason, LIME was chosen over SHAP as the token attribution method for this study.

4.6.2 Generated Explanations

While LIME is well suited for smaller language models like RoBERTa, for the LLMs used in this study, we explored generated explanations, where the model itself produces natural language justifications for its predictions. This approach leverages the generative capabilities of LLMs gained from the knowledge and patterns acquired during pretraining, which we use to justify the reasoning behind their classification decisions.

To enable this, we repurposed the models fine-tuned for classification by replacing the classification head that was added during the LoRA fine-tuning process, with a causal language modeling head. To the best of our knowledge, the approach of using different heads on the same model for explanation generation has not been previously explored. This configuration enabled the same LLM model to produce natural language explanations in place of class label outputs. Each input pair was accompanied by a structured prompt that guided the model to generate a reasoning-based explanation, while intentionally omitting

explicit class labels to mitigate the risk of overfitting in the automated evaluation phase of the study. A sample of an explanation from each model can be found in Table 3 and Table 4, alongside the prompt template which can be found in Table 2.

4.6.3 Proposed Experiments for Explainability

We propose two experiments to evaluate the quality and effectiveness of the explainability methods used in this study:

4.6.3.1 Experiment 1: Human Evaluation of Explanations

The first experiment involves human evaluation of the explanations produced by RoBERTa with LIME, and the generated method using the LLMs. A balanced subset of 28 samples from the COLIEE dataset was selected for this purpose. Two human evaluators, a PhD student in Information Science and a Master's student in Computer Science, were tasked with assessing explanations using the following criteria:

- **Adequacy**: Does the explanation adequately justify the classification decision?
- **Understandability**: Does the evaluator understand the explanation?
- **Trustworthiness**: Does the evaluator trust the model based on the explanation?

Based on the evaluation methodology from Rajagopal et al. (2021), evaluators provided binary responses for adequacy and understandability, and a Likert scale rating of 1–5 for trustworthiness. To ensure consistency during human evaluation, the prompts used for explanation generation were constructed using the correct label for each example, rather than relying on the model's own prediction. This setup allows the evaluation to focus on how well the model can justify legal decisions, independent of any misclassifications.

4.6.3.2 Experiment 2: Automated Evaluation of Explanations

The second experiment on explainability focuses on evaluating the quality of the explanations across the different models using an automated method based on the sufficiency metric. The aim of this metric is to assess if the explanations generated by each model are sufficient to predict the correct label, allowing us to measure the alignment between the extracted explanations and correct predictions. This is done using the FRESH pipeline (Jain et al., 2020) where a BERT classifier is trained solely on the explanations without any prior context in order to predict the classification label. This automated approach is applied to the keyword extraction method for the RoBERTa model and the generated explanations of the LLMs. To ensure that the BERT classifier is basing its prediction on the content of the explanation to predict the label, and not the mention of the label itself in any of the explanations, the words "Entailment" or "Contradiction" in the explanations were masked with the token <label>. The sufficiency score was calculated using the complete training and testing dataset. We also compare how the automated evaluation of the explanations correlates with the results of our sample of explanations that were human evaluated.

4.7 Summary

This chapter outlined the methodology used for legal entailment classification and explanation. It presented the overall system architecture, dataset sources, and preprocessing strategies tailored for transformer-based models, with an emphasis on preserving legal language structure. The study employed both domain-specific pretraining on Canadian court cases and fine-tuning on the COLIEE 2023 dataset.

To adapt models efficiently, especially large-scale LLMs, the study used techniques such

as prompt engineering, LoRA, and 4-bit quantization via QLoRA. A diverse set of models, including encoder-only, decoder-only, and ensemble configurations, were evaluated for their classification performance. For explainability, the chapter introduced token attribution using LIME and generated explanations using a novel method that repurposes the same model for both classification and generation. Finally, it proposed both human and automated evaluation strategies to assess the interpretability of the explanations, contributing to a more transparent application of AI in the legal domain.

Input Pair	**Text 1:** It is trite law that in PRRA applications, the burden of proof is on the person claiming protection under subsection 114(1) of the Act. It is up to that person to establish that protection must be granted to him or her. **Text 2:** [19] THE COURT ORDERS THAT: - The application for judicial review be dismissed. There is no question to be certified.
RoBERTa Tokenization	'It', 'Ġis', 'Ġtr', 'ite', 'Ġlaw', 'Ġthat', 'Ġin', 'ĠPR', 'RA', 'Ġapplications', ',', 'Ġthe', 'Ġburden', 'Ġof', 'Ġproof', 'Ġis', 'Ġon', 'Ġthe', 'Ġperson', 'Ġclaiming', 'Ġprotection', 'Ġunder', 'Ġsubsection', 'Ġ114', '(', '1', ')', 'Ġof', 'Ġthe', 'ĠAct', '.', 'ĠIt', 'Ġis', 'Ġup', 'Ġto', 'Ġthat', 'Ġperson', 'Ġto', 'Ġestablish', 'Ġthat', 'Ġprotection', 'Ġmust', 'Ġbe', 'Ġgranted', 'Ġto', 'Ġhim', 'Ġor', 'Ġher', '.', '<\s>', '[', '19', ']', 'ĠTHE', 'ĠCOURT', 'ĠOR', 'D', 'ERS', 'ĠTHAT', ':', 'Ġ-', 'ĠThe', 'Ġapplication', 'Ġfor', 'Ġjudicial', 'Ġreview', 'Ġbe', 'Ġdismissed', '.', 'ĠThere', 'Ġis', 'Ġno', 'Ġquestion', 'Ġto', 'Ġbe', 'Ġcertified'
Llama 2 (7B) Tokenization	'<s>', '_It', '_is', '_tr', 'ite', '_law', '_that', '_in', '_PR', 'RA', '_applications', ',', '_the', '_bur', 'den', '_of', '_proof', '_is', '_on', '_the', '_person', '_claim', 'ing', '_protection', '_under', '_sub', 'section', '_', '1', '1', '4', '(', '1', ')', '_of', '_the', '_Act', '_.', '_It', '_is', '_up', '_to', '_that', '_person', '_to', '_establish', '_that', '_protection', '_must', '_be', '_granted', '_to', '_him', '_or', '_her', '.', '<s>', '_[', '1', '9', ']', '_THE', '_CO', 'UR', 'T', '_ORDER', 'S', '_TH', 'AT', ':', '_-', '_The', '_application', '_for', '_jud', 'icial', '_review', '_be', '_dismiss', 'ed', '.', '_There', '_is', '_no', '_question', '_to', '_be', '_cert', 'ified'
Llama 3 (8B) Tokenization	'<|begin_of_text|>', 'It', 'Ġis', 'Ġtr', 'ite', 'Ġlaw', 'Ġthat', 'Ġin', 'ĠPR', 'RA', 'Ġapplications', ',', 'Ġthe', 'Ġburden', 'Ġof', 'Ġproof', 'Ġis', 'Ġon', 'Ġthe', 'Ġperson', 'Ġclaiming', 'Ġprotection', 'Ġunder', 'Ġsubsection', 'Ġ', '114', '(', '1', ')', 'Ġof', 'Ġthe', 'ĠAct', '.', 'ĠIt', 'Ġis', 'Ġup', 'Ġto', 'Ġthat', 'Ġperson', 'Ġto', 'Ġestablish', 'Ġthat', 'Ġprotection', 'Ġmust', 'Ġbe', 'Ġgranted', 'Ġto', 'Ġhim', 'Ġor', 'Ġher', '.', '<|begin_of_text|>', '[', '19', ']', 'ĠTHE', 'ĠCOURT', 'ĠORD', 'ERS', 'ĠTHAT', ':', 'Ġ-', 'ĠThe', 'Ġapplication', 'Ġfor', 'Ġjudicial', 'Ġreview', 'Ġbe', 'Ġdismissed', '.', 'ĠThere', 'Ġis', 'Ġno', 'Ġquestion', 'Ġto', 'Ġbe', 'Ġcertified'
GPT-4o Tokenization	'It', ' is', ' tr', 'ite', ' law', ' that', ' in', ' PR', 'RA', ' applications', ',', ' the', ' burden', ' of', ' proof', ' is', ' on', ' the', ' person', ' claiming', ' protection', ' under', ' subsection', ' ', '114', '(', '1', ')', ' of', ' the', ' Act', '.', ' It', ' is', ' up', ' to', ' that', ' person', ' to', ' establish', ' that', ' protection', ' must', ' be', ' granted', ' to', ' him', ' or', ' her', '.', '[', '19', ']', ' THE', ' COURT', ' ORD', 'ERS', ' THAT', ':', ' -', ' The', ' application', ' for', ' judicial', ' review', ' be', ' dismissed', '.', ' There', ' is', ' no', ' question', ' to', ' be', ' certified'

Table 4.2: Tokenized representation of the input text pair across different models

Model	Architecture	Size	Open Source	Adaptation Strategy
RoBERTa	Encoder-only transformer	125M	Yes	Fine-tuned on task-specific data
RoBERTa + Legal Pretraining	Encoder-only transformer	125M	Yes	Further pretrained on legal corpus, then fine-tuned on task-specific data
Llama 2 (7B)	Decoder-only transformer	7B	Yes (Meta license)	Fine-tuned using LoRA or prompt engineering
Llama 2 (7B) + Legal Pretraining	Decoder-only transformer	7B	Yes (Meta license)	Further pretrained on legal corpus, then fine-tuned using LoRA
Llama 3 (8B)	Decoder-only transformer	8B	Yes (Meta license)	Fine-tuned using LoRA or prompt engineering
GPT-4o	Decoder-only transformer	over 100B (est.)	No (API only)	Prompt engineering, fine-tuning
Pretrained RoBERTa + SNLI RoBERTa Ensemble	Ensemble of encoder-only RoBERTa models	Multiple 125M	Yes	Combines predictions of RoBERTa pretrained on legal corpus and RoBERTa fine-tuned on SNLI
Pretrained Llama 2 + Pretrained RoBERTa Ensemble	Ensemble of decoder-only and encoder-only transformers	7B + 125M	Yes	Combines predictions of Llama 2 with RoBERTa model, both pretrained on legal corpus
Llama 2 + GPT Ensemble	Ensemble of decoder-only models	7B + over 100B (est.)	Partial	Combines predictions of Llama 2 and GPT-4o
GPT-4o + Pretrained RoBERTa Ensemble	Ensemble of decoder-only and encoder-only transformers	over 100B (est.) + 125M	Partial	Combines predictions of GPT-4o and the RoBERTa model pretrained on legal corpus

Table 4.3: Summary of models and ensemble configurations used in the study, including architecture, size, openness, and adaptation strategies.

Chapter 5

Results and Discussion

This chapter discusses the results of the experiments conducted for our system by addressing each research question and examining the corresponding findings. The chapter begins by analyzing the performance of the entailment classifier, followed by an evaluation of the XAI methods used to interpret the model's decisions.

5.1 Entailment Classification

The entailment classifier serves as the central component of the proposed system, as described in the methodology chapter. Its goal is to determine whether a given pair of legal texts are in an entailment or contradiction relationship. To develop this classifier, a series of experiments were conducted, which are described below.

- **Experiment 1: Adapting LLMs:** This experiment evaluates how to adapt LLMs to the task of entailment classification, exploring LORA fine-tuning and prompt engineering as computationally efficient alternatives to full parameter fine-tuning.

Team	File	Recall	Precision	F1-score
CAPTAIN	mt5l-ed.txt	0.7083	0.7870	0.7456
CAPTAIN	mt5l-ed4.txt	0.6750	0.7864	0.7265
THUIR	thuir-monot5.txt	0.6583	0.7900	0.7182
CAPTAIN	mt5l-e2.txt	0.6583	0.7596	0.7054
THUIR	thuir-ensemble_2.txt	0.6583	0.7315	0.6930

Table 5.1: Top 5 results for legal entailment classification in COLIEE 2023 (Task 2) based on F1-score.

- **Experiment 2: Effects of Pretraining on Legal Corpus:** This experiment evaluates two pairs of models, RoBERTa and Llama 2. Each pair consists of a base model and a version that undergoes additional pretraining on domain-specific legal data, with the goal of improving performance on legal text compared to the base model.
- **Experiment 3: Ensemble Methods for Legal Entailment:** This experiment evaluates the combination of the predictions of RoBERTa models and LLMs using a consensus-based criterion, in order to enhance the results and outperform the invidiual models.

Overall, the classification results in the three experiments are similar to the results of the top ranking competitors in the 2023 COLIEE competition Goebel et al. (2023) which are showcased in Table 5.1. The top 3 best ranking entries obtained F1-scores of 0.745, 0.726, and 0.718. It is important to note that their methods used DL models, but did not include systematic experiments about domain-specific pretraining strategies and did not address the issue of explaining the decisions of the entailment classifiers.

5.1.1 Results for Experiment 1

This first experiment is designed to evaluate the adaptation of LLMs for the legal classification task. The goal is to find alternative to full parameter optimization, since while the increase in

parameter size gives these models more weights to optimize for the data, their large size makes it very difficult to perform traditional model training on most consumer hardware. For this reason, this experiment investigates the results of the task using prompt engineering and LoRA fine tuning on the models in Table 5.2

Model	**Recall**	**Precision**	**F1-Score**
Llama 2 - 7B with finetuning	0.642	0.770	0.700
Llama 3 - 8B with finetuning	0.617	0.740	0.673
GPT 4o with finetuning	0.658	**0.790**	**0.718**
Llama 2 - 7B with prompt engineering	0.058	0.069	0.063
Llama 3 - 8B with prompt engineering	0.525	0.0774	0.135
GPT 4o with prompt engineering	**0.875**	0.166	0.280

Table 5.2: Comparison of model performance with fine-tuning and prompt engineering

It is worth noting that prompt engineering is considerably less time-intensive and demands fewer computational resources compared to LoRA fine-tuning, making it a more viable option in resource-constrained environments. For example, in the case of Llama 2, LoRA fine-tuning consumes approximately 49.5 GB of VRAM and requires around 20 hours to complete training on the full training dataset. In contrast, few-shot prompting uses only around 31.6 GB of VRAM and 6.752s on average based on 10 samples.

In terms of performance, the results showcase that there is a significant difference between fine-tuning and prompt engineering for adapting LLMs for the task. Fine-tuning consistently outperforms prompt engineering across all evaluated models for F1-score. Among the fine-tuned models, the GPT-4o model achieved the highest F1-score of 0.718, followed closely by Llama 2 7B at 0.700. As for the fine-tuned Llama 3 performing lower than the fine-tuned Llama 2, an empirical study on Llama 3 quantization by Huang et al. (2024) suggest that LoRA fine-tuning can lead to performance degradation in Llama 3 due to its large scale pre-training dataset of over

15T tokens, which potentially makes it more sensitive to fine-tuning on smaller datasets with low-rank parameter adjustments.

In comparison, the models adapted using prompt engineering demonstrated significantly lower F1-scores than all the fine-tuned models. Among the models using prompt engineering, the GPT-4o model similarly achieved the a higher F1-score than the other LLMs, with a F1-score of 0.280, while Llama 2 7B and Llama 3 8B both performed poorly with their respective F1-scores of 0.063 and 0.135. While GPT-4o achieved a notably high recall score of 0.875 with prompt engineering, this came at the cost of a very low precision score, indicating that a large portion of its positive predictions were incorrect, making it unreliable for the task. These results showcase the need for deeper optimization of model parameters in the case of highly specialized classification tasks like legal entailment, which prompt engineering not only did not perform well for, but had a significant drop in quality in comparison to the fine-tuned models. The results also highlight the notion that while prompt engineering can be a useful lightweight strategy for some NLP tasks, it struggles to match the robustness of fine-tuning for domain-specific applications such as this one.

5.1.2 Results for Experiment 2

The second experiment is designed to evaluate the impact of additional domain-specific pretraining on the performance of two language models of different sizes in the classification task. By comparing each model fine-tuned directly on the entailment dataset with a counterpart that first underwent further pretraining on a corpus of 35,617 federal court cases, we aim to investigate whether domain-adaptive pretraining leads to measurable performance gains. The comparison between the two RoBERTa models and the two Llama 2 models is presented in Table 5.3.

Model	Recall	Precision	F1-Score
RoBERTa	0.603	0.724	0.658
RoBERTa with further pretraining	**0.622**	**0.746**	**0.678**
Llama 2 - 7B	**0.642**	**0.770**	**0.700**
Llama 2 - 7B with further pretraining	0.625	0.750	0.682

Table 5.3: Impact of pretraining on models

The results indicate that further domain-specific pretraining using MLM leads to measurable performance gains for the RoBERTa model. Specifically, the RoBERTa model that underwent further pretraining achieved an increase of around 0.02 points in F1-score compared to the base version. As stated in the methodology section, these results were taken as the average of 10 random seeds, where specifically the untrained model achieved an average F1-score of 0.658 $\pm$ 0.019, while the further pretrained model achieved an average F1-score of 0.678 $\pm$ 0.011.

In comparison to the LLMs in experiment 1, while the further pretrained RoBERTa model did not surpass the performance of the base Llama 2 model, it outperformed the Llama 3 model from Experiment 1 for classification. The difference between the size of RoBERTa with 125 million parameters, in comparison to size of Llama 3 with 8 billion parameters shows the computational efficiency of the pretrained RoBERTa, since further pretraining a smaller model on domain specific data is less computationally expensive than fine-tuning a large language model.

This level of improvement aligns with the findings of the research on pretraining a BERT-based model by Gururangan et al. (2020), which also demonstrates that continued pretraining on domain-specific data can have a similar level of improvement over various tasks. The results from this experiment extend prior findings to the legal domain, demonstrating that smaller transformer models, such as RoBERTa, can benefit from domain-adaptive pretraining on Canadian legal documents without requiring the massive computational resources often associated with large-scale language models. These results make the approach particularly valuable in resource-constrained

settings where legal text classification is needed.

Unlike the smaller RoBERTa, the larger Llama 2 model did not show improvement with further pretraining. This could be related to its much larger initial training dataset of approximately 2 trillion tokens, indicating that our pretraining dataset was not large enough to provide additional benefit to a model of this scale.

Model	**Recall**	**Precision**	**F1-score**
RoBERTa	0.482	0.501	0.489
RoBERTa (with pretraining)	0.536	0.561	0.548
Llama 2	0.567	0.602	0.584
Llama 2 (with pretraining)	0.550	0.606	0.576
Llama 3	0.608	0.608	0.608
GPT-4o	**0.683**	**0.641**	**0.661**

Table 5.4: Performance of models using the full classification strategy (without confidence filtering).

In addition, we evaluated the models from experiment 1 and experiment 2 without confidence filtering in Table 5.4. While GPT-4o maintained its top position with an F1-score of 0.661, its result is notably lower compared to the most confident classification strategy, which had an F1-score of 0.718. The same can be said for the rest of the models, which all consistently benefited from leveraging prediction confidence scores.

5.1.3 Results for Experiment 3

This experiment evaluates whether ensemble methods can outperform the best-performing individual model in each pairing for the task of legal entailment classification. The results of the evaluated ensembles are presented in Table 5.5.

Model	Recall	Precision	F1-Score
Pretrained RoBERTa + SNLI RoBERTa	0.633	0.644	0.639
Llama 2 + Pretrained RoBERTa	0.650	0.748	0.695
Llama 2 + GPT-4o	**0.683**	**0.774**	**0.726**
GPT-4o + Pretrained RoBERTa	**0.683**	0.771	0.724

Table 5.5: Comparison of ensemble performance across different model combinations

The results indicate that the ensembles containing the GPT-4o model not only outperformed their individual counterpart, but also achieved the highest precision and F1-scores across all experiments. The ensemble with Llama 2 and GPT-4o, and the ensemble with the pretrained RoBERTa model and GPT-4o, respectively, achieved the highest F1-scores of 0.726 and 0.724, both surpassing the standalone GPT-4o model that had an F1-score of 0.718.

However, the ensembles comprised of only RoBERTa models and Llama 2 models were not able to surpass their highest performing counterparts. The ensemble with Llama 2 and pretrained RoBERTa model achieved an F1-score of 0.695, falling slightly short of Llama 2's individual F1-score of 0.700. In addition, the ensemble of the two RoBERTa models, one pretrained on legal data and one fine-tuned on the SNLI dataset, performed notably lower with an F1-score of 0.639. This result suggests that the SNLI dataset, although widely used in general natural language inference tasks, does not transfer effectively to the legal domain to provide improvements to the pretrained RoBERTa model.

Although the GPT-4o based ensembles performed well, the results do not suggest that ensembling is universally beneficial. Other ensembles that did not involve GPT-4o failed to outperform their strongest individual components. Overall, the ensemble methods provided mixed results, and indicate that the model combination alone is not guaranteed to improved performance in legal entailment classification when using the consensus-based criterion.

5.2 Explainability

As explained in the methodology chapter, the explainability part of the systems aims at providing explanation for the classification predictions of the AI models. A challenge in this area is finding a quantitative evaluation method to measure the quality of these explanations. Two experiments aim to address this, one involving human judges, and the other involving an automated method of evaluation. The experiments are conducted on the best performing individual models from the classification portion of the study.

- Experiment 1: Human evaluation of explanations: In the first experiment, human evaluation is performed on a sample of the explanations generated by the classification models, where their adequacy, understandability and trustworthiness is investigated.
- Experiment 2: Automated evaluation of explanations: In the second experiment, a method based on the FRESH pipeline is used to automate the evaluation of the explanations by measuring the sufficiency of each explanation.

5.2.1 Results for Experiment 1

In this experiment, 28 samples were evaluated by human judges, specifically looking at adequacy, understandability and trustworthiness. The results are found in Table 5.6.

From the results, GPT-4o, followed by Llama 2, were the best performing models, and Llama 3 and RoBERTa with LIME the weakest. GPT-4o achieved the highest scores across all three evaluation metrics, with a score of 82.14% $\pm$ 0.036 for adequacy, 92.86% $\pm$ 0.071 for understandability, and 85.71% $\pm$ 0.393 for trustworthiness. Llama 3 performed the lowest out of the two Llama models, with a score of 0.0% $\pm$ 0.000 for adequacy, 3.57% $\pm$ 0.036 for understandability,

Human judge	Model	Adequacy	Understandability	Trustworthiness
Human judge 1	LIME	3.57%	3.57%	26.43%
	Llama 2 - 7B	46.43%	60.71%	53.57%
	Llama 3 - 8B	0.00%	7.14%	22.14%
	GPT-4o	85.71%	89.29%	82.86%
Human judge 2	LIME	3.57%	3.57%	20.00%
	Llama 2 - 7B	46.43%	71.43%	60.00%
	Llama 3 - 8B	0.00%	0.00%	20.00%
	GPT-4o	78.57%	96.43%	88.57%
Combined Average (± SD)	LIME	3.57% ± 0.036	3.57% ± 0.036	23.21% ± 0.161
	Llama 2 - 7B	46.43% ± 0.071	66.07% ± 0.125	56.79% ± 0.304
	Llama 3 - 8B	0.00% ± 0.000	3.57% ± 0.036	21.07% ± 0.054
	GPT-4o	**82.14%** ± 0.036	**92.86%** ± 0.071	**85.71%** ± 0.393

Table 5.6: Explanation quality comparison across different models by human judges, including standard deviation (SD) of the scores.

and 21.07% ± 0.054 for trustworthiness. These low scores can be attributed to its explanations appearing vague and lacking depth, which inadequately captured the nuances of the two legal texts. In comparison, Llama 2 was able to extract important concepts such as names, locations and events, which made its generated explanations more detailed and specific than Llama 3. However, when we tested a few samples on the Llama 3 model not fine-tuned for entailment classification, the explanations were much better and detailed. To further demonstrate this contrast, Table 5.7 showcases an example output from Llama 3, comparing responses with and without LoRA fine-tuning for a pair of contradictory texts. The fine-tuned model produces a vague explanation that lacks direct reference to the input content, whereas the untuned version offers a more coherent interpretation by specifically highlighting the judge's obligations in Text 1, and the discussion of cross-examination in Text 2. This aligns with our previous observations regarding potential quality degradation in Llama 3 models due to LoRA fine-tuning. The performance of the RoBERTa

Model Setting	Explanation
Llama 3 (No fine-tuning)	The model made the decision to classify the pair because the two texts present different perspectives on the role of cross-examination in the process. Text 1 suggests that the Citizenship Judge is not obligated to provide an opportunity to file additional material, implying that the process is already complete. On the other hand, Text 2 states that there is no requirement to cross-examine a deponent on their affidavit, which could be seen as an opportunity for additional information to be presented.
Llama 3 (with LoRA Fine-tuning)	The model made the decision to classify the pair because the provided text seems to be discussing a specific aspect of a larger context, and the model has determined that the text does not accurately reflect the reality of the situation.

Table 5.7: Example of Llama 3 quality degradation.

model with LIME also had poor results, which resulted in a score of 3.57% $\pm$ 0.036 for adequacy and understandability, and 23.21% $\pm$ 0.161 for trustworthiness. This is not surprising as simply extracting 6 keywords is limited in providing a comprehensive explanation of the depth and logical connections necessary to explain the model's decision in classifying the pair of text.

Overall, the standard deviation values for adequacy and understandability were relatively low across all models, mostly ranging from 0.000 to 0.071, indicating agreement between the evaluators. For Llama 3, the standard deviation for adequacy was 0.000, due to complete agreement on its consistently poor adequacy scores. In contrast, trustworthiness scores showed greater variability,

with standard deviation values reaching 0.393 for GPT-4o and 0.304 for Llama 2, suggesting that the evaluations for trustworthiness were more subjective.

5.2.2 Results for Experiment 2

In this experiment, we calculated the sufficiency of the explanations of each model using an automated method based on the FRESH pipeline. This involves a BERT classifier that is trained exclusively on the model-generated explanations to determine whether these explanations alone are sufficient for the BERT classifier to predict the correct label. The results are shown in Table 5.8.

Model	**Recall**	**Precision**	**F1-score**
RoBERTa with LIME	0.567	0.548	0.557
Llama 2 - 7B	0.933	0.918	0.926
Llama 3 - 8B	0.683	0.656	0.669
GPT-4o	**1.000**	**0.984**	**0.992**

Table 5.8: Comparison of automated evaluation for sufficiency

From the table, we can see that GPT-4o dominates in recall, precision and F1-score. This aligns with Experiment 1, where GPT-4o consistently scored highest in adequacy, understandability, and trustworthiness. Llama 2 was the second best performing model with an F1-score of 0.9256, also reflecting its stronger performance in Experiment 1 in comparison to Llama 3 due to its more detailed and context-aware explanations. Llama 3 and RoBERTa with LIME similarly demonstrated a significant gap in performance when compared to GPT-4o and Llama 2. RoBERTa with LIME had the lowest F1-score of 0.557, reinforcing its limited ability to provide comprehensive explanations with only six extracted keywords. Overall, the results from this experiment correlate with the results from Experiment 1, highlighting that models producing higher-quality human-evaluated explanations also perform better in automated sufficiency evaluations.

5.3 Summary

This chapter analyzed the results of our experiments for legal entailment classification and explanation evaluation. It highlights that fine-tuning consistently outperformed prompt engineering across all LLMs, with GPT-4o achieving the highest classification scores. As for the effects of domain-specific pretraining, the results showed improved performance for smaller models like RoBERTa, but limited benefit for the larger language model Llama 2. As for the effects of combining models through ensemble methods, the results produced mixed outcomes where only GPT-4o-based ensembles surpassed individual models, while other ensembles showed no gains. In terms of explainability, both experiments had evaluation results that correlated together, with GPT-4o and Llama 2 providing the most effective explanations in both human and automated evaluations. In contrast, Llama 3 and LIME-based explanations underperformed. Overall, these findings highlight the effectiveness of fine-tuning for LLMs and domain adaptation for smaller models, the limitations of prompt-based and ensemble approaches in complex legal tasks, and the explainability capabilities of the models for interpretable prediction in our legal AI system.

Chapter 6

Conclusion

This chapter provides a summary of the work that was done and examines the contributions of this study. It also discusses the study's limitations and outlines directions for future research.

6.1 Conclusion

This thesis set out to explore the application of NLP and XAI to the complex task of legal entailment classification. In doing so, it addressed the challenges posed by the legal domain, including specialized language, and the critical demand for interpretable and trustworthy AI systems, by developing models that can accurately classify whether a paragraph from an existing legal case entails the decision in a new case, while also providing transparent justifications for these classifications. The main contribution of this work is the development of a unified system capable of both classifying legal entailment relationships and generating interpretable explanations for its decisions under the same model. By leveraging both small transformer models like RoBERTa and larger language models such as Llama and GPT-4o, this system demonstrates flexibility across computational and resource constraints.

The experimental results demonstrate several key findings. Domain-specific pretraining was shown to enhance the performance of smaller transformer models like RoBERTa, with an increase in F1 score, from 0.658 to 0.678. This provides a computationally efficient alternative to using much larger language models, as the pretrained RoBERTa outperformed the Llama 3 model in the classification task despite having significantly fewer parameters. However, it also highlights that such benefits do not always translate to larger models like Llama 2, which may already encapsulate extensive linguistic knowledge from their pretraining corpora. For LLMs, fine-tuning consistently outperformed prompt engineering across all evaluated models, showing that specialized tasks like legal entailment classification benefit from parameter optimization rather than just contextual prompting. Among the fine-tuned models, GPT-4o achieved the highest F1 score of 0.718, followed closely by Llama 2 with 0.700, and pretrained RoBERTa with 0.678, while Llama 3 trailed behind at 0.673. As for the ensemble, only GPT-4o-based combinations outperformed their individual counterparts, while other ensembles showed no improvement. This suggests that ensembling and combining prediction is not reliably beneficial.

The study also explored various explainability methods, finding that generated explanations from GPT-4o demonstrated superior quality in human evaluations, with high scores for adequacy of 82.14%, understandability with 92.86%, and trustworthiness with 85.71%. Explanations were also evaluate through an automated sufficiency metric, where GPT-4o again achieved the highest score of 0.992, indicating strong alignment between explanation quality and classification performance. Llama 2 also performed well, achieving a sufficiency score of 0.926 and competitive human evaluation scores for adequacy at 46.43%, understandability at 66.07%, and trustworthiness at 56.79%, further supporting its potential as a reliable and open-source alternative to GPT-4o. This affirms the potential for using a single LLMs for both classification and explanation generation.

In summary, this research demonstrates that a hybrid approach combining domain-specific pretraining, lightweight adaptation, and explainability techniques that can improve both the per-

formance and transparency of AI systems in legal contexts. The findings contribute to the growing body of work to steer the domain of law towards a future that can utilize the strong performance of black-box models while also providing interpretability and trust in the AI systems.

6.2 Summary of Contributions

This thesis makes several contributions to the fields of NLP and XAI in the domain of law. Below is a summary of the main contributions:

- **Development of a Domain-Specific Legal Entailment System:** This research introduces a novel system specifically designed for legal entailment classification, capable of leveraging both small BERT-based models and LLMs for classification and explanation tasks within a unified framework that integrates both predictive accuracy and interpretability within a single model architecture. To the best of our knowledge, this is the first system of its kind to use language modeling head swapping.
- **Creation of Domain-Specific Pretrained Legal Models:** This study developed open-source, domain-specific models for Canadian court cases by pretraining on a corpus of 35,617 Canadian federal court cases. The impact of this pretraining was systematically analyzed to assess whether it led to measurable improvements in legal text classification.
- **Comprehensive Evaluation of Model Adaptation Strategies:** This study conducted an extensive evaluation of various model adaptation techniques for legal text classification, examining both lightweight and parameter-efficient approaches. It compared a range of transformer models under different strategies such as fine-tuning, prompt engineering, and ensembling.

- **Evaluation of Explainability Techniques:** This study emphasizes the importance of interpretability in AI-driven legal decision-making and investigated various explainability methods. A comprehensive evaluation framework was implemented to assess the quality of model-generated explanations through both automated metrics and human judgment.

6.3 Limitations

This study contains some limitations that should be considered. Our human evaluation was limited to a small sample rather than the complete dataset due to time restrictions. Computational resource limitations required us to use a single random seed when fine-tuning LLMs. We also excluded SHAP from our explainability evaluations as it requires substantially more computational resources than LIME without demonstrating significantly better performance for text classification according to prior research (Stilwell and Inkpen, 2024). Furthermore, the computational requirements of perturbation-based explanation methods like LIME and SHAP made it impractical to apply them to our LLMs when working with our extensive legal dataset. An additional limitation is the potential for data leakage in LLMs like GPT-4o or Llama. Since the exact pretraining data for these models has not been publicly disclosed, we cannot confirm whether parts of the evaluation dataset, such as the COLIEE cases, were seen during pretraining.

6.4 Future Work

While this thesis presented a comprehensive system for legal entailment classification and explainable AI, several directions remain open for future exploration and improvement.

1. **Data Expansion and Augmentation:** Although a substantial corpus of 35,617 Canadian federal court cases was used for pretraining, the results indicate that this size may be

insufficient to significantly improve the performance of already extensively pretrained large models such as Llama 2. Future work could involve curating larger and more diverse legal corpora, including provincial decisions to enhance pretraining.

2. **Investigation of Llama 3 Performance Issues:** Our findings revealed unexpected performance degradation in Llama 3 after LoRA fine-tuning compared to Llama 2. A deeper investigation into this degradation could uncover underlying causes and inform strategies to enhance Llama 3's fine-tuning effectiveness.

3. **Optimized Prompt Engineering Techniques:** While this study found that fine-tuning consistently outperformed prompt engineering, future research could focus on exploring other methods found in the literature, that go beyond zero-shot and few-shot prompting, such as Chain of Though prompting. Another avenue to optimize the prompts could be to develop more specialized prompting strategies tailored to legal reasoning. This could involve legal professionals, such as lawyers or legal scholars, in the design of prompts in order to better capture domain-specific language, and expectations of justification found in legal contexts.

4. **Explanations for Incorrect Predictions:** In this study, human evaluators assessed explanations generated for correctly labeled predictions. Future work could explore how models explain their incorrect predictions, especially for the best-performing systems like GPT-4o. Analyzing the reasoning patterns behind misclassifications could offer valuable insight into model limitations, and potentially reveal systematic biases. This may also inform whether such erroneous justifications could still appear convincing or misleading to end users.

5. **Comparative Analysis with Other Legal NLP Tasks:** Future work should explore whether our findings regarding model architectures, pretraining, and explainability extend

to other legal NLP tasks beyond entailment classification, such as legal information retrieval.

6. **Generalization Across Different Settings:** Future research could examine whether the findings of this study hold under different settings, including other legal domains such as criminal, constitutional, or international law.

7. **Legal Professional Involvement in Evaluation:** Involving legal professionals such as judges, lawyers, and legal scholars in human evaluation could provide deeper insights into the explanation quality of the model-generated explanations. Beyond rating adequacy, understandability, and trustworthiness, legal professionals could also assess whether the model's predictions and explanations are sufficiently reliable and accurate to be applied in real-world legal contexts.

8. **Domain-Specific Pretraining from Scratch:** Rather than further pretraining existing models, exploring the development of legal-specific language models trained from scratch on a Canadian legal corpora could lead to deeper insights regarding the benefits of domain-specific pretraining. While computationally intensive, such models might achieve better performance with smaller parameter counts than adapted general-purpose models, especially for specialized tasks like legal entailment classification.

9. **Ensemble Approaches for Legal Classification:** Although the current study explored ensemble methods, the experiment provided mixed results. Future work could investigate more robust ensembling strategies to improve reliability and performance across diverse models. Potential directions include the use of majority voting across three or more models, and weighted ensemble techniques where model predictions are combined based on their individual performance scores. In addition, future research should investigate whether the performance gains achieved through these different ensembling strategies are statistically significant when compared to the best-performing individual classifiers.

References

Alvarez-Melis, D. and Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*.

Ariai, F. and Demartini, G. (2024). Natural language processing for the legal domain: A survey of tasks, datasets, models, and challenges. *arXiv preprint arXiv:2410.21306*.

Azam, F. (2000). *Biologically inspired modular neural networks*. PhD thesis, Virginia Polytechnic Institute and State University.

Bao, M., Zhou, A., Zottola, S., Brubach, B., Desmarais, S., Horowitz, A., Lum, K., and Venkatasubramanian, S. (2021). It's compaslicated: The messy relationship between rai datasets and algorithmic fairness benchmarks. *arXiv preprint arXiv:2106.05498*.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Brown, T. B. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., and Androutsopoulos, I. (2020). Legal-bert: The muppets straight out of law school. *arXiv preprint arXiv:2010.02559*.

Chen, J. S. and Baxter, S. L. (2022). Applications of natural language processing in ophthalmology: present and future. *Frontiers in Medicine*, 9:906554.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. (2024). Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Devlin, J. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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.

Dong, S., Wang, P., and Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, 40:100379.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Goebel, R., Kano, Y., Kim, M.-Y., Rabelo, J., Satoh, K., and Yoshioka, M. (2023). Summary of the competition on legal information, extraction/entailment (COLIEE) 2023. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 472–480.

Gunter, D., Puac-Polanco, P., Miguel, O., Thornhill, R. E., Yu, A. Y., Liu, Z. A., Mamdani, M., Pou-Prom, C., and Aviv, R. I. (2022). Rule-based natural language processing for automation of stroke data extraction: a validation study. *Neuroradiology*, 64(12):2357–2362.

Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., and Smith, N. A. (2020). Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Heyen, H., Widdicombe, A., Siegel, N. Y., Perez-Ortiz, M., and Treleaven, P. (2024). The effect of model size on llm post-hoc explainability via lime. *arXiv preprint arXiv:2405.05348*.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Huang, K., Altosaar, J., and Ranganath, R. (2019). Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*.

Huang, W., Ma, X., Qin, H., Zheng, X., Lv, C., Chen, H., Luo, J., Qi, X., Liu, X., and Magno, M. (2024). How good are low-bit quantized llama3 models? an empirical study. *arXiv preprint arXiv:2404.14047*.

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al. (2024). Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Jain, S., Wiegreffe, S., Pinter, Y., and Wallace, B. C. (2020). Learning to faithfully rationalize by construction. *arXiv preprint arXiv:2005.00115*.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., and Zhao, L. (2019). Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia tools and applications*, 78:15169–15211.

Kim, M.-Y., Rabelo, J., Babiker, H. K. B., Rahman, M. A., and Goebel, R. (2024). Legal information retrieval and entailment using transformer-based approaches. *The Review of Socionetwork Strategies*, 18(1):101–121.

Kommiya Mothilal, R., Mahajan, D., Tan, C., and Sharma, A. (2021). Towards unifying feature attribution and counterfactual explanations: Different means to the same end. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 652–663.

Lauriola, I., Lavelli, A., and Aiolli, F. (2022). An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing*, 470:443–456.

Leippert, A., Anikina, T., Kiefer, B., and Genabith, J. (2024). To clarify or not to clarify: A comparative analysis of clarification classification with fine-tuning, prompt tuning, and prompt engineering. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 105–115.

Lichman, M. and Bache, K. (2013). UCI machine learning repository. `http://archive.ics.uci.edu/ml`. Accessed: 2025-03-02.

Limsopatham, N. (2021). Effectively leveraging bert for legal document classification. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 210–216.

Linna Jr, D. W. (2021). Evaluating legal services: The need for a quality movement and standard measures of quality and value. In *Research Handbook on Big Data Law*, pages 404–431. Edward Elgar Publishing.

Liu, Y. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Madsen, A., Reddy, S., and Chandar, S. (2022). Post-hoc interpretability for neural nlp: A survey. *ACM Computing Surveys*, 55(8):1–42.

Nie, Y., Williams, A., Dinan, E., Bansal, M., Weston, J., and Kiela, D. (2019). Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Paranjape, B., Michael, J., Ghazvininejad, M., Zettlemoyer, L., and Hajishirzi, H. (2021). Prompting contrastive explanations for commonsense reasoning tasks. *arXiv preprint arXiv:2106.06823*.

Pereira, S. C., Mendonça, A. M., Campilho, A., Sousa, P., and Lopes, C. T. (2024). Automated image label extraction from radiology reports–a review. *Artificial Intelligence in Medicine*, page 102814.

Pornprasit, C. and Tantithamthavorn, C. (2024). Fine-tuning and prompt engineering for large language models-based code review automation. *Information and Software Technology*, 175:107523.

Rajagopal, D., Balachandran, V., Hovy, E., and Tsvetkov, Y. (2021). Selfexplain: A self-explaining architecture for neural text classifiers. *arXiv preprint arXiv:2103.12279*.

Rebala, G., Ravi, A., and Churiwala, S. (2019). *An introduction to machine learning*. Springer.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " 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*, pages 1135–1144.

Richmond, K. M., Muddamsetty, S. M., Gammeltoft-Hansen, T., Olsen, H. P., and Moeslund, T. B. (2024). Explainable ai and law: An evidential survey. *Digital Society*, 3(1):1.

Romanov, A. and Shivade, C. (2018). Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv:1808.06752*.

Sahoo, P., Singh, A. K., Saha, S., Jain, V., Mondal, S., and Chadha, A. (2024). A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Sekulić, I., Soleimani, A., Aliannejadi, M., and Crestani, F. (2020). Longformer for ms marco document re-ranking task. *arXiv preprint arXiv:2009.09392*.

Sindhu Meena, K. and Suriya, S. (2020). A survey on supervised and unsupervised learning techniques. In *Proceedings of international conference on artificial intelligence, smart grid and smart city applications: AISGSC 2019*, pages 627–644. Springer.

Stilwell, S. and Inkpen, D. (2024). Explainable Prompt-based Approaches for Sentiment Analysis of Movie Reviews. *Proceedings of the Canadian Conference on Artificial Intelligence*. https://caiac.pubpub.org/pub/oe1gma4v.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Trad, F. and Chehab, A. (2024). Prompt engineering or fine-tuning? a case study on phishing detection with large language models. *Machine Learning and Knowledge Extraction*, 6(1):367–384.

Vale, D., El-Sharif, A., and Ali, M. (2022). Explainable artificial intelligence (xai) post-hoc explainability methods: Risks and limitations in non-discrimination law. *AI and Ethics*, 2(4):815–826.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J., Zhao, H., Yang, Z., Shu, P., Chen, J., Sun, H., Liang, R., Li, S., Shi, P., Ma, L., et al. (2024). Legal evalutions and challenges of large language models. *arXiv preprint arXiv:2411.10137*.

Williams, A., Nangia, N., and Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Xu, X. and Cai, H. (2021). Ontology and rule-based natural language processing approach for interpreting textual regulations on underground utility infrastructure. *Advanced Engineering Informatics*, 48:101288.

Zhang, H., Zhu, Q., and Dou, Z. (2025). A unified prompt-aware framework for personalized search and explanation generation. *ACM Transactions on Information Systems*.

Zhao, H., Chen, H., Yang, F., Liu, N., Deng, H., Cai, H., Wang, S., Yin, D., and Du, M. (2024). Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–38.

Appendix A: Examples of Legal Text Pairs and Model Explanations

A: Example of Input Pairs

Text 1:	The standard of review for breach of procedural fairness is correctness.
Text 2:	[66] Although neither party has made submissions on the standard of review on this issue, I wish to note that this aspect of the decision must be held to a standard of correctness. The Applicant alleges that the Respondent breached procedural fairness by failing to give advance notice of the decision. This Court has repeatedly found that the standard of review for breaches of procedural fairness is correctness and that will be the standard applicable to this issue.
Class:	Entailment
Text 1:	Lemieux, J., declined to order consolidation, for he found that the two actions there concerned different contracts arising from the same general circumstances, different parties to the two actions, and there was no substantive commonality of legal and factual issues in the two actions.
Text 2:	[29] McKeil is not a party in the Canning action against Tripap.
Class:	Contradiction

Table 1: Examples of input pairs with their respective labels for entailment and contradiction.

B: Prompt Templates

Classification Prompt	**Natural Language Explanation Prompt**
Classify the pair as "Contradiction" if there is a contradiction within the two texts or if you are not sure. Classify the pair as "Entailment" only if you are certain the two paragraphs of text involve or are consequences of each other. Here are a few examples to help you make your decision: <EXAMPLES GO HERE> Now that you have seen the examples, classify the following two texts. It is most likely a contradiction, but if you are certain it is not, classify the pair as "Entailment" if the two paragraphs of text involve or are consequences of each other.	Give an explanation based on the information provided below for why the model made the decision to classify the pair with the chosen label. Please don't mention the actual word 'Contradiction' or 'Entailment' in your response under any circumstances, simply give a brief explanation based on the information provided on why you think the classification is appropriate. If you have to use the words 'Contradiction' or 'Entailment' in your response, please use the token <label> instead. For example, instead of saying 'The model made the decision to classify the pair as Contradiction', you can say 'The model made the decision to classify the pair as <label>'. **Format:** • **Text 1:** {query} • **Text 2:** {paragraph} • **Classification:** {classification} • **Response:** {explanation}

Table 2: Prompts used for few-shot entailment classification and natural language explanation generation.

C: Examples of Generated Explanations

Input Pair	• **Text 1:** the court may only intervene if there is no line of analysis within the reasons that could reasonably lead the board from the evidence before it to the conclusion at which it arrived. • **Text 2:** [24] Similarly, with respect to the third ground, the Registrar concludes that the burden has not been satisfied because "the probabilities of confusion favour neither one side nor the other."
Class:	Contradiction
LIME Response	The words 'may', 'court', 'evidence', 'confusion', 'analysis' and 'only' make the pair of texts seem like a contradiction to the model.
Llama2 Response	Based on the information provided, the model has classified the pair as <label>. The reasoning given in the text for the decision is "the probabilities of confusion favour neither one side nor the other." This suggests that the pair is classified as <label> because the reasoning provided does not lead logically or reasonably from the evidence before the court to the conclusion it arrived at.
Llama3 Response	The model made the decision to classify the pair because the information provided by the second text does not provide a clear and direct conclusion, unlike the first text which presents a clear and logical argument.
GPT-4o Response	The model made the decision to classify the pair as <label> because the first text suggests that the court can only step in if there's no reasonable path from the evidence to the conclusion, implying that there is a clear line of reasoning. However, the second text indicates that the Registrar concludes there is no clear probability favoring either side, suggesting an absence of a clear line of reasoning. This discrepancy in the presence of a clear line of reasoning led the model to its classification.

Table 3: Examples of generated explanations for an input pair of class Contradiction

Input Pair	• **Text 1:** the applicant would be at risk to life or at risk to cruel and unusual treatment or punishment if returned to Afghanistan. • **Text 2:** [12] In the meantime a pre-removal risk assessment (PRRA) was undertaken in respect of the Applicant's potential return to Afghanistan and it was determined on 20 July 2004 that the Applicant would be at risk to life or at risk to cruel and unusual treatment or punishment if he returns to Afghanistan.
Class:	Entailment
LIME Response	The words 'risk', 'treatment', 'applicant', and 'Afghanistan' make the pair of texts seem like an entailment to the model. Both texts share overlapping and consistent terminology which aligns them closely.
Llama2 Response	The model made the decision to classify the pair as Entailment. The explanation for this decision is that the language used in the two sentences is similar and conveys a consistent meaning. The phrase "at risk to" is used in both sentences, and it is clear that the same situation is being described in both cases. Therefore, the model has classified the pair as Entailment based on the similarity of the language and the consistency of the meaning.
Llama3 Response	The model made the decision to classify the pair as Entailment because the provided text suggests that the second statement is a direct consequence of the first statement. The first statement presents a specific condition, and the second statement reveals the potential outcome of the first. The model has determined that the second statement is a logical consequence of the first, and therefore, the two statements are connected in a way that makes the second statement a natural result of the first.
GPT-4 Response	The model made the decision to classify the pair as Entailment because both the text and the provided information align perfectly. They both state that the applicant would face danger to their life or be subjected to cruel and unusual treatment or punishment if they were to return to Afghanistan. Therefore, the information in the second text is directly derived from the first text, making the classification appropriate.

Table 4: Examples of generated explanations for an input pair of class Entailment.