Legal Judgment Prediction for Canadian Appeal Cases

by

Intisar Almuslim

Thesis submitted to the
Faculty of Engineering
In partial fulfillment of the requirements
For the Ph.D. degree in
Computer Science

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

Abstract

Law is one of the knowledge domains that are most reliant on textual material. In this age of legal big data, and with the increased availability of legal text online, many researchers started working on the development of legal intelligent systems and applications. These intelligent systems can provide great services and solve many problems in the legal domain. Over the last few years, researchers have focused on predicting judicial case outcomes using Natural Language Processing (NLP) and Machine Learning (ML) methods over case documents. Thus, Legal Judgment Prediction (LJP) is the task of automatically predicting the outcome of a court case given the case description. To the best of our knowledge, no prior research with this intention has been conducted in English for appeal courts in Canada, as of 2023. The NLP application to legal judgments, that our proposed methodology focuses on, is to predict the outcomes of cases by looking only at the text of cases. Because appeal court decisions are often binary, as in 'Allow' or 'Dismiss', the task is defined as a binary classification problem. This is the general approach in the literature as well. However, many of the previous LJP approaches utilized traditional classifiers or standard general language models (LMs). In our thesis, we constructed a Canadian Appeal-Law dataset (CanAL-DS) that contains a collection of decisions from different higher courts in Canada. In addition, we further pre-trained the LegalBERT model on our collected corpus that combines around 50,000 documents of Canadian case law and legislation which resulted in (CanAL) LegalBERT, a Canadian Appeal-Law BERT-based legal model. Moreover, we proposed a novel Ensemble-Hierarchical CanAL (EH-CanAL) architecture that simulates the actual voting setting in appellate courts showing great promise in LJP performance within Canadian case law. We improved the architecture with a multi-task component (MEH-CanAL) to help the model identify what legal paragraphs require the most attention and facilitate its explainability. Results from our study demonstrate the potential for the proposed approaches to reshape traditional judicial decision-making and the efficacy of domain-specific language models. Through this study, we hope to establish the basis for future research on the appellate law system of Canada and offer a baseline for future work.

Acknowledgements

My PhD journey has been a remarkable journey filled with challenges, including family circumstances, political upheavals, and the unprecedented global pandemic. Despite these obstacles, I could not have moved forward without resilience, determination, and commitment, transforming obstacles into opportunities for growth and learning. I would like to extend my heartfelt gratitude to the following individuals, without whom this journey would not have been possible:

To my esteemed supervisor **Prof. Diana Inkpen**, your guidance, support, and scholarly wisdom have been instrumental in shaping this research. Your mentorship has been a source of encouragement throughout my academic pursuit. You have consistently demonstrated a rare combination of mentorship and scholarly acumen. Your keen insights and critical feedback have refined the methodology and scope of my research. In expressing my gratitude to Prof. Diana Inkpen, I acknowledge the immeasurable impact you have had on my academic journey. I am privileged to have had the opportunity to work under your guidance.

To **Prof. Wolfgang Alschner** for his identification of the research gap, which became the cornerstone of my PhD thesis and also ignited a passion for exploration and inquiry. The foundation laid by Prof. Alschner has been instrumental in the development of this thesis, and I am profoundly thankful for his unwavering support and scholarly guidance.

To my scholarship employer **King Saud University** and the **Saudi Arabian Cultural Bureau in Canada** for the financial support that has eased the burden of my academic expenses. Your investment in my education has been a significant factor in my success.

To the **members of TAMALE research group**, your contributions during our insightful meetings have enriched my perspectives and sharpened the focus of my research. Your collaborative spirit has been a constant source of inspiration.

To my neighbour **Mrs.Amnah**, for generously sharing valuable information related to law and the legal domain. Your insights have been an unexpected yet crucial asset to my research.

To **my mother and brothers** for their unwavering support, encouragement, prayers and sacrifices. Your belief in my abilities has fueled my determination to reach this academic milestone.

To **my husband, Mr. Abdulaziz Alabbadi**, your patience, understanding, and love have sustained me through the challenges of this academic journey. Your presence is my greatest source of strength and joy.

To **my six beautiful children, Sarah, Omar, Deemah, Yousef, Hamed, and Nouf**, whose unwavering love, patience, and understanding have been my greatest source of strength throughout this academic journey. Their boundless joy, encouragement, and occasional study breaks filled with laughter provided the balance and inspiration needed to navigate the challenges of doctoral research. Your belief in me has fueled my perseverance. This thesis is a tribute to the resilience and unwavering support of my cherished children. Thank you for being my greatest cheerleaders and for turning even the most challenging days into moments of shared success and joy."

Each of you has played a unique and fundamental role in this academic journey, and I am profoundly grateful for your unwavering support.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Legal information is often expressed in natural language and represented in textual form. Nowadays, although this information is available online, the number of legal documents produced daily in law firms and courts is increasing enormously. It is also well known that in the legal domain, there are many complex decisions that need to be made by law practitioners. Some examples of these decisions include what case law might be relevant for a specific case, which area of law is applicable to a given case, what the outcome might be, what are the possible charges or fines of the case, etc.

One of the services that lawyers have to provide to their clients is the prediction of various types of case outcomes by assessing the overall strengths of arguments and a client's legal position in a specific lawsuit or proposition. Therefore, *"Predictability"* is a fundamental property of legal systems. Predicting the outcome or the chance of winning a legal case has always been highly desirable in legal practices. It is typically used in appeal courts or supreme courts, where appeals are cases that have already been determined by lower courts and have been escalated to the appeal courts or supreme courts for reassessment. In an appeal court, judges decide together upon a case and their judgments are documented in agreement reports named "judgments" or "decisions". Appeal decisions, similar to lower court decisions, include explanation and analysis of the facts or law, and

further metadata such as judgment date, attorneys, judges, etc. These documents are very useful for guiding lawyers and judges in understanding the constitution in the common law systems, thus helping the decision-making process (Kumar et al., 2013). For instance, judges always consider previous judicial decisions as precedents and use them as guidelines for future judgments (Leszczyński, 2020). Moreover, one of the most common and essential tasks for legal practitioners is to predict how a certain court will decide, given the facts that make up the case. For example, this is useful for preparing and tuning a case to increase the chances of a favourable decision. Hence, attorneys can rely on substantial assumptions on how judges will decide based on their submissions. However, predictions about the outcome of cases are often provided based on the experience of law professionals. In addition, although this information can be found in public decisions, the myriads of available documents make this task very complex and error-prone, even for experienced lawyers.

Given the complexity of the judiciary system worldwide, investing in the development of intelligent tools to support legal specialists and consequently facilitate and expedite legal decisions becomes crucial to many countries. If Artificial Intelligence (AI) can be used to support real-life legal decision-making, there will be a significant potential to reduce cost and processing time. An automated Legal Judgment Prediction (LJP) system can help not only legal professionals but also the general public who are not legal specialists. The system could allow everyone to predict and foresee the outcome of litigation when involved in legal disputes as people can access the system wherever, whenever. Also, the anticipated cost of LJP will be much lower than that of human legal professionals. The system is expected to provide broader access to justice for people who have limited or no access to justice.

All this being said, our research's primary objective is to develop a framework to predict judicial outcomes of the appeal cases of Canadian case law. We believe that by designing a prediction model that offers consistent and explainable results for some of the Canadian appeal courts, this model could be transferred, after fine-tuning, to any other appeal courts in Canada.

1.2 Problem Statement

Deciding on cases is a complicated and time-consuming task. It requires the court staff to look through various records to identify supporting statements for any possible case outcomes. Court judgments play a crucial role in litigation, legal study, and court decision-making because some of them are exemplars of legal usage and interpretation. Lawyers use these judgments to analyze if their clients could win the lawsuits.

In the field of AI, LJP is a challenging task for many reasons:

1. Although legal decisions are based on logical reasoning, it is far too complex and time-consuming to handcraft rules and to imitate such tasks in a computational model.

2. Working with legal documents can be challenging because of the uniqueness and complexity of the language. Some words and phrases used in legal documents have particular meanings within the legal domain, especially when they are used by legal practitioners from different cultures/countries, or within different legal systems. For example, the distinction in English law between solicitors and barristers is not made in Canadian law. In some cases, the concepts are the same, but the words used to describe them are different (British company law versus American corporate law). Another example is that in countries such as India, legal English is infused with many terms for indigenous legal concepts. Legal language is influenced by the cultural background, customs, and beliefs of a particular community. Thus, legal language reveals itself as a relatively flexible means of communication by readily adapting to the situation and place in which it is used (Tiersma, 1999).

3. Legal court cases are usually unstructured, lengthy, and noisy. It is difficult to extract and utilize facts and arguments directly. While traditional NLP methods can process long texts, they are less effective as compared to novel approaches, such as transformer-based models. However, transformer-based models are limited in that they can handle only a certain amount of data, which makes them difficult to use

in real-life scenarios in which the size of the input document exceeds the maximum input size.

4. It is difficult to obtain a public universal annotated dataset of legal judgment predictions. The majority of the related work (Aletras et al., 2016; Kowsrihawat et al., 2018; Long et al., 2019; Octavia-Maria et al., 2017; Virtucio et al., 2018) has been developed for many different languages or very different legal systems; therefore, they made their own datasets for model training and testing.

5. Due to the domain-specific lexicon used in court cases, models pre-trained on generally available texts are ineffective in such cases. Consequently, the standard models need to be adapted to the legal domain for the legal judgment prediction on court cases. To train and assess LJP models, it is necessary to develop the LJP tasks and their datasets reflecting differences in jurisdictions.

6. Explaining prediction in legal documents is considerably more challenging as it requires understanding the facts, following the arguments and applying legal rules and principles to arrive at the final decision.

In light of these considerations, it is clear that the judgment prediction and explanation tasks are far more challenging than standard text classification tasks. In this thesis, we show how we tackle the problem of providing NLP-based intelligence support to legal decision-makers in a real-world setting using transformer-based NLP. For this purpose, we constructed a dataset of LJP with regard to Canadian judgments in order to provide a reliable dataset for LJP research in Canada, as well as to further pre-train the first Canadian Legal Language Model (LM) specializing in appeal cases. Additionally, our architecture has been improved by making it hierarchical to handle lengthy documents, and ensemble to simulate real-world scenarios. We further enhanced its explainability by utilizing multi-task learning. We limit the scope of the study to the set of appeal cases wherein the court decision is either "dismissed" or "allowed" the appeal. Therefore, the problem of case outcome prediction can be framed as a binary classification task. To the best of our knowledge, the appeal court system of Canada has not been the subject of

outcome prediction using NLP-based approaches. The work described in this research aims to bridge this gap by investigating the effectiveness of large legal language models in predicting case outcomes of Canadian appeal cases.

1.3 Research Questions

This research intends, as a primary objective, to develop a framework to predict judicial outcomes of the appeal cases of Canadian case law, along with explanations of those outcome predictions. Thus, we have identified four research questions that help with achieving this objective:

1. Are country-specific legal models more effective in comparison to generic legal models or legal models originating from other countries?
2. Is it possible to build a system that predicts the outcomes of the appeal cases from the Canadian higher courts using a legal model further pre-trained on Canadian appellate law?
3. How to adapt this model to effectively deal with long documents in legal judgment prediction?
4. How to adapt this model to effectively justify its outcomes?

As an approach to address these questions, this thesis proposes a methodology involving the implementation and evaluation of a system capable of predicting legal outcomes and their rationales from appeal documents in hierarchical, multitasking, and ensembling settings.

1.4 Ethical Considerations

Firstly, as a research tool, the system is intended to assist legal professionals in their work and not to replace their role in the field. In this respect, the system does not violate

ethical considerations such as allowing artificial intelligence to make decisions about human legal rights. In addition, we do not intend to develop a system of artificial lawyers and judges that regulate human behaviour. Instead, we aim to promote the development of an explainable system for predicting appeal case outcomes and provide valuable information to legal professionals or laypersons that might assist them in making strategic decisions. It is important to note, however, that the actual decision-making process must be carried out by the legal professionals themselves. This work takes the first steps in taking this process forward by introducing the Canadian-specific corpus and baseline models. Nevertheless, further research in this area is needed to fully evaluate the social implications of such a system. Secondly, we declare that we did not use Generative AI to assist in writing or coding this work. During the preparation of this work, the author just used some paraphrasing tools such as Wordtune and Grammarly. After using these tools, the author reviewed and edited the content as needed and took full responsibility for the content of the thesis. Lastly, we used the Responsible NLP Checklist [1] from the ACL 2023 conference to evaluate the ethical implications of our work. The questions and our answers can be found in Appendix A.

1.5 Contributions

Our research studies the LJP task, a classification task that has significant practical value to legal professionals. The contributions of our study are:

1. We constructed a dataset that contains a collection of decisions from different higher courts in Canada and labeled it automatically with an outcome variable that can be used in training classifiers for LJP tasks. So far, most of the similar research for LJP has relied on a manual annotation process, which is tedious and requires domain knowledge (See Chapter 3).

2. We enhanced our dataset with additional annotations to facilitate a range of auxiliary tasks, such as statute or case law prediction and explainability (See Section 3.2.4).

[1] https://aclrollingreview.org/responsibleNLPresearch/

3. We further pre-trained a Canadian-specific transformer-based legal model on our collected corpus that combines around 50,000 documents of Canadian case law and legislation (See Chapter 4).

4. We proposed a novel hierarchical network architecture in an ensemble setting showing great promise in LJP performance within the Canadian case law. To the best of our knowledge, this task has not been studied in Canadian legal texts from the appeal courts (See Chapter 5).

5. To improve the model's explainability, we added a multi-task component with self-learning loss weights that helps the model identify which legal paragraphs should be in focus (See Chapter 6).

1.6 Thesis Structure

The rest of this thesis is organized as follows: In Chapter 2, A short overview of Predictive Modelling (PM) in the judicial system is provided, and literature related to our research is reviewed and categorized according to the legal prediction tasks addressed. Details on data collecting, preparation, and pre-processing are given in Chapter 3. Chapter 4 introduces our Canadian appeal law legal model which is further pre-trained using our collected Canadian legal corpus. The methods that we used, and their experimental settings, are given in Chapter 5, together with details on the metrics that we used to evaluate the performance. The description of the explainability component is given in Chapter 6. Chapter 7 concludes by discussing our findings, the limitations of our work, and the directions of research that can be investigated in future work.

1.7 Publications

1. Almuslim, I., & Inkpen, D. (2020). Document-level embeddings for identifying similar legal cases and laws (AILA 2020 shared task). Proceedings of FIRE, pp. 42-48.

2. Almuslim, I., & Inkpen, D. (2022). Legal Judgment Prediction for Canadian Appeal Cases. In Proceedings of the 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA) (pp. 163-168). IEEE.

3. Almuslim, I., Stilwell, S., Suresh, S. K., & Inkpen, D. (2023). uOttawa at SemEval-2023 Task 6: Deep Learning for Legal Text Understanding. In Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), pp. 580-588.

Chapter 2

Background and Related Work

2.1 Overview

The first part of this chapter briefly reviews the history of judicial predictive modeling in the legal domain, along with the milestones that have led to this current stage in the field's development. The second part of the chapter reviews and discusses related literature by classifying it into different legal prediction tasks. It focuses on studies that have used ML/DL and NLP to provide predictions about legal cases.

2.2 Background

2.2.1 Overview of the Appeal Law System

In Canada, there are separate courts that are operated by each province as well as the federal government. The method by which cases are heard is very similar in all provinces and the federal government (Government of Canada, 2021a). In general, court systems are divided into at least three levels: the trial court, the intermediate appellate court; and the supreme court. A trial court is the first court, where criminal and civil cases are introduced. During a trial, the court hears evidence, consults with witnesses, and allows both sides to present their arguments before providing a verdict. It is possible to appeal a verdict if one

of the parties to the original decision believes that the decision was wrong or unfair. Thus, the appeal process allows a judgment from a lower-level court to be considered by a higher court. It is the court of appeal that hears these cases on appeal from lower courts, which is an intermediate level of court in between trial courts and the supreme court (Government of Canada, 2021a).

It is important to note that trial courts and appeal courts have significant differences. Unlike the trial case, the appeal case must be based on an issue of law instead of a factual one. During the appeal process, no new evidence is presented, there is no jury, and no witnesses are called. This means that the evidence is not reheard by the appeal court. Instead, the appeals party, known as the appellant, presents legal arguments to the panel in a written brief, attempting to convince the judges that the trial court made substantial errors and that its decision should therefore be reversed. Courts of appeals are typically composed of a panel of judges with an odd number of judges to avoid a tie. This panel reviews the case, and the majority decision of the panel members (i.e., an agreement by more than half of the panel members) determines whether or not the law was applied correctly. The majority opinion contains the reasoning behind the court's decision as well as the decision itself. Also, the decision will incorporate any relevant precedents, such as cases previously decided by that court or the Supreme Court. The majority opinions, are incorporated into the body of case law in countries with a common law system. Once issued, these opinions act as precedents for later courts, thus opinions provide their own legal rules that become part of the case law. (Government of Canada, 2021b).

2.2.2 Predictive Modelling in The Legal Domain

The application of artificial intelligence (AI) in the legal domain is not a new idea. Since the mid-twentieth century, there has been an active group of researchers exploring the potential and methodologies of AI and applying them to law applications (Buchanan and Headrick, 1970; Buxbaum and Latin, 1973; Chien et al., 1971). The development of AI in the legal field has followed a path that is very similar to that of AI research in general. Law-related AI began with a focus on knowledge representation, then rules-based legal systems

appeared (Ashley, 2019). In the 1960s and 1970s, the earliest systems for accessing online legal information appeared, and *legal expert systems* were a major topic of study in the 1980s and 1990s (Dale, 2019).

Later, AI and law have shifted from knowledge-representation techniques toward ML-based approaches, like the AI field in general. A new wave of utilizing artificial intelligence in law has emerged in the form of data-driven ML techniques (Trasberg, 2019). One of the most exciting domains within the emerging wave of artificial legal intelligence is Predictive Modelling (PM). In Data Science, PM is simply a process that uses data mining, statistics, and probability to forecast outcomes, while these outcomes are determined by a number of variables and factors that go into the model (Kaur and Bozic, 2019).

PM studies in the legal domain have focused mainly on two directions (Ashley, 2019). On one hand, many researchers focused mostly on document review and knowledge discovery in the legal domain, also known as Electronic Discovery, or E-discovery. E-discovery is the process of searching, identifying, collecting, and organizing electronically stored legal documents which simplifies the work of lawyers and law practitioners (Mauritz, 2018). On the other hand, after the first generation of E-discovery which was focused upon platforms for analyzing previously decided cases, a new generation of E-discovery emerged. It centred around Predictive Coding (PC) technology (Chhatwal et al., 2018) and focused more on the decision-making process by trying to discover hidden patterns and making predictions using modern AI techniques. In recent years, PC of the case law has become a research hot-spot in legal intelligence. The following section will discuss how this promising technique has been utilized in legal AI research to provide appropriate legal advice, including predicting case outcomes. Figure 2.1 shows the key milestones that will be discussed.

2.2.3 Predicting Judicial Decisions

Over the past few decades, many researchers have worked actively to advance the field of legal prediction. They believed, at that time, that a judge's responsiveness to the facts and legal arguments in a case can be predicted (Ashley, 2019). Early works (Kort, 1957; Lawlor, 1963; Nagel, 1960; Ulmer, 1963) about predicting judicial decisions proposed simple

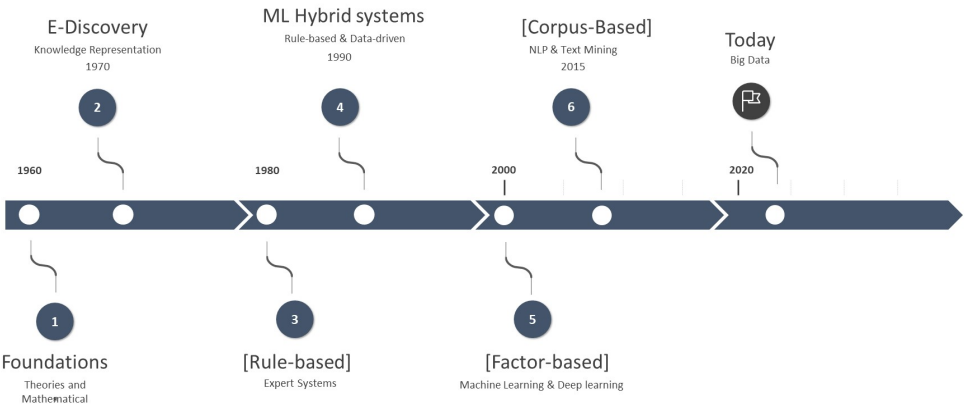


Figure 2.1: Key Milestones in Legal AI Research

mathematical and quantitative methods due to computational limitations at that time. For instance, Kort (1957) studied mathematical relationships in judicial review by manually converting the factual components of cases into numerical values, computing their sum, and predicting a decision in favour of the petitioner if the sum was above a manually-chosen threshold. He proved that the knowledge can be encoded as numerical characteristics and used to predict judgments in certain domains. Moreover, Lawlor (1963) in his research on the possible use of information technology in the legal area, anticipated that computers will one day be able to predict the results of judicial judgments. According to Lawlor (1963), making a reliable prediction of the behaviour of relevant decision-makers in law, such as the judges, is dependent on an understanding of how the law and facts affect them.

While early efforts remained as theoretical suggestions, new and more practical ways to automating law appeared after that. When ML approaches were utilized, it was primarily the methods that produced interpretable models that were preferred, such as K-Nearest Neighbors (KNN) and Decision Trees, which were frequently used for predictive judicial analytics purposes. (Mckaay and Robillard, 1974) is one of the earliest outcome prediction studies that employed the KNN approach, using 64 tax cases with 46 hand-crafted features. In this work, the similarity measure was simply the number of features with the same values in the cases. Another study (Ruger et al., 2004) was based on decision trees with six hand-crafted features, and it produced correct predictions for 75.0% of the Court's affirm/reverse results, while the experts predicted only 59.1% correctly.

Gradually, and from the beginnings of AI with the use of rule-based systems in the late 1980s, intelligent algorithms have been used in the legal domain to benefit from the automation and reasoning given by AI. Different approaches proposed this kind of reasoning in the form of legal *Expert Systems* (Ashley, 1992; Gardner, 1984; Smith et al., 1992; Susskind, 1986). With the ability of legal expert systems to translate legal rules into a machine-readable system of logic, algorithms become able to read law and apply it to specific circumstances. However, Legal rules may contain ambiguous language that can be interpreted in multiple ways. For instance, a statute stating that a person should not "willfully" cause harm, the term "willfully" may be open to interpretation, leading to debates over whether specific actions meet the statutory requirements. This characteristic of law makes it far more complex and challenging than a simple set of logical rules. As a result, only a few limited situations have shown significant success with logic-based automated legal systems (Ashley, 2019).

More recently, AI and law have shifted away from rule-based methods to data-driven techniques. One of the most promising areas within the emerging wave of artificial legal intelligence is Legal Judgment Prediction (LJP). Using LJP can help legal professionals (such as judges, attorneys, etc.) operate more efficiently while also lowering the risk of making mistakes. In addition, it can help ordinary people who do not have extensive legal expertise but are interested in knowing the expected outcome of a case they have. LJP can help people to be free from the hard tasks of information retrieval and data analysis by using the legal knowledge contained in massive law statutes and case judgment documents. However, due to the complexities of court proceedings, training an intelligent machine model to anticipate proper judgment decisions is not easy (Li et al., 2019).

The first attempts used *factor-based* ML models to predict the judgments of the legal cases. In this kind of ML, researchers are interested in whether a certain case-related variable of factor has a strong impact on the probability that a court or judge will rule in one direction or the other. Factors are usually weighted to measure how much they influence a given case towards a particular classification (Shaikh et al., 2020). An example of this is the study carried out by Martin et al. (2004) in which classification trees were utilized, and

658 case documents were tagged with some metadata factors such as type of case and judge name to be used as predictive features. The work of Katz et al. (2014) is another example which used the Extremely Randomized Tree method to form a classification algorithm, with a feature engineering approach. However, their feature engineering was extremely complex and used up to fifty feature vectors, mostly derived from sophisticated manual annotation. Ashley and Brüninghaus (2009) applied KNN approach to manually constructed summaries rather than the original text of the case decisions. The classifiers were built taking into account different factors. The proposed method for classifying and predicting cases was able to meet 91.8% of accuracy, however, the evaluation took only into account a small data set (146 cases).

More recently, the evolution of *corpus-based* text analysis techniques, along with the increased availability of very large legal text corpora, has made it possible for models to derive inferences straight from court records (Branting et al., 2015). Because of the advances in NLP and ML, we now have the tools we need to analyze legal documents automatically and construct models that accurately anticipate judicial decisions. As a result, LJP becomes an essential and representative task in the legal domain. Besides, LJP has drawn lots of attention from both AI researchers and legal professionals. In the following section, we describe the research progress and focus on recent advances in LJP, which employs NLP, ML, and other techniques to provide predictions of legal cases.

2.3 Related Work

2.3.1 Legal Language Models

In the years following the introduction of the self-attention-based transformer mechanism (Vaswani et al., 2017), we have seen a sustained development of large-scale, pre-trained language models. Devlin et al. (2019) first introduced BERT, a multi-layer transformer architecture for language modeling. On general domain corpora, notably English Wikipedia and BookCorpus, they trained a deep bidirectional language model using Masked Language

Modelling (MLM), which masks some tokens randomly in the text. They also used the Next Sentence Prediction (NSP) objective to sample pairs of contiguous sentences.

There was, however, a general lack of performance in downstream tasks in specialized domains such as healthcare and law due to these general-domain models. This led to efforts to either re-train models such as BERT (to continue pre-training from the BERT checkpoint) or pre-train them from scratch on specialized domain data. For example, by comparing models such as BioBERT (Lee et al., 2020), LegalBERT (Chalkidis et al., 2020), and SciBERT (Beltagy et al., 2019) to the general BERT model, these models demonstrate significant improvements in their domain tasks. It is evident that domain-specific pretraining is effective based on the performance of such models.

As well as domain-specific pre-training being effective, developing a new BERT model tailored to a particular language or country can also be beneficial. Models like Arabert (Antoun et al., 2020) and BERTje (De Vries et al., 2019), BERT models trained on Arabic and Dutch texts, respectively, are beneficial for tasks that are specific to these languages, for example for sentiment classification (De Kruijf et al., 2022).

A combination of domain-specific and country-specific approaches can also be applied. With regard to the legal domain in particular, efforts have been made to pre-train transformer-based models on country-specific legal systems. Garneau et al. (2021) introduced CriminelBART, a new French-Canadian Legal Language Model Specialized in Criminal Law. Chalkidis et al. (2020) pre-trained BERT-base on EU and UK legislation and court documents from the European Court of Justice (ECJ) and the European Court of Human Rights (ECtHR), and released the LegalBERT model. Zheng et al. (2021) proposed CaseLaw-BERT, pre-trained on a corpus of US case law documents and contracts. Paul et al. (2023) prepared a very large corpus of Indian case law documents and trained LegalBERT to yield the InLegalBERT model.

It is important to note that, as a result of the increased number of legal tasks and datasets released recently across a variety of legislations and law systems, including those in the United States, the European Union, China, and India, as well as the unique nature of the legal text and the need to handle issues relating to specific laws within the legal systems,

it is essential that language models not only specialize in the legal domain as a whole, but also specialize in specific laws within a specific country. Following this path of releasing BERT models suited for a specific domain, this thesis introduces a new transformer-based model CanAL-LegalBERT, for Canadian appeal law, trained on Canadian legal domain data. This model bridges the gap of having no legal models applicable for the Canadian Appeal law in English. Especially for the legal judgment prediction problem, a model trained on Canadian appeal-law data can be very beneficial.

2.3.2 Legal Judgment Prediction Task (LJP)

LJP requires an integrated relationship between case law reasoning and other types of legal reasoning, such as statute law reasoning. While case law consists of decisions and precedents derived from past court decisions, statute law consists of laws adopted by legislative bodies. These forms of legal reasoning are inherently interconnected, despite that they may have distinct functions. Thus, recognizing and incorporating the nuanced interactions between case law and statute law reasoning becomes imperative within the discourse of legal judgment prediction. Taking into account insights from both domains, predictive models are able to gain a deeper understanding of the legal landscape, which enhances their ability to anticipate judicial outcomes in the future.

The majority of the studies attempted to resolve the legal prediction task by formalizing it as a classification problem, either binary or multi-class classification. This differs from research in which the variable of interest is continuous. For example, Bala et al. (2017) attempted to predict the time it would take for an insurance claim to be settled. Another example is the work of Chalkidis et al. (2019), who predicted the importance of the cases. Such studies that look at predicting continuous target variables are less relevant to this research.

After reviewing the selected articles, we divided the application of text classification to LJP into three broad categories, as shown in Figure 2.2: predicting the case's outcome, predicting violation against law articles, and predicting the legal area or statute of a case. Among the gathered literature, we focused on studies for English texts. However, a growing

body of research exists on legal prediction of case law from international courts in different languages. So, these studies are also reviewed in order to gain a broad understanding of how NLP and ML can be applied to the legal prediction tasks.

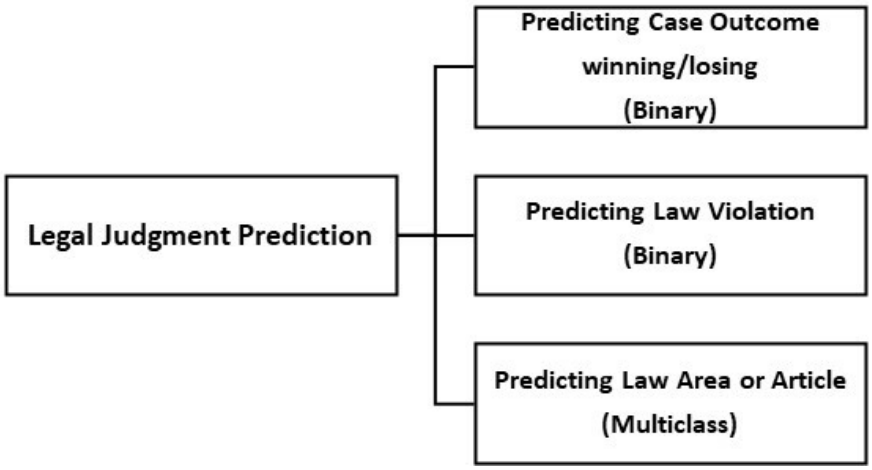


Figure 2.2: Legal Judgment Prediction Categories

Predicting Case Outcome [Win/Lose]

Who wins a case? predicting the outcome or possibility of winning a court dispute has always been appealing in legal research and practice. The aim of this task is to predict the outcome of winning or losing the case by classifying it into labels such as affirmed/reversed or allowed/dismissed. Hence, the task can be framed as a binary classification problem, as shown in Figure 2.3.

This task is usually used in the courts of appeal or supreme courts, where appeals are cases previously decided by lower courts and transferred to the higher courts for recon-

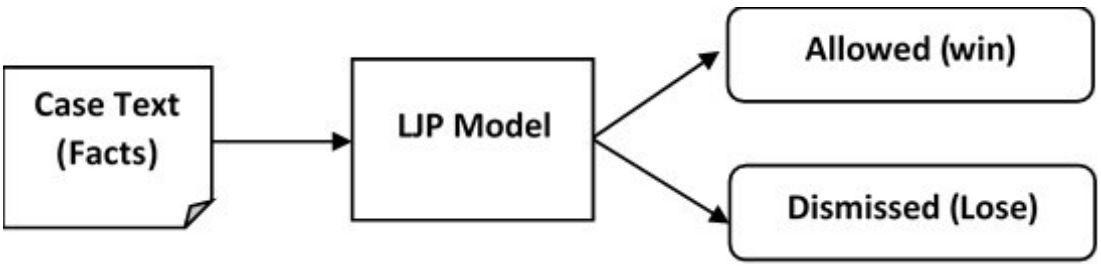


Figure 2.3: Predicting Case Outcome [Win/Lose] (Binary)

sideration. The majority of existing English studies looked at the Supreme Court of the United States (SCOTUS). The federal court system in the United States is based on an appellate review system, in which higher courts can either confirm or reverse lower court decisions (Agrawal et al., 2017). Predicting these affirm/reverse outcomes has been extensively researched in related work. However, the early previous work has largely focused on non-textual information. Many research efforts, such as (Katz et al., 2014; Sharma et al., 2015; Katz et al., 2017), employed classification models with features extracted from case profiles by manual annotation. Alternatively, many other studies have investigated such tasks through the acquisition of a richer semantic understanding of case factors based on social, political, and economic aspects contained in structured databases such as the U.S. Supreme Court Database (SCDB) (Spaeth et al., 2020).

The SCDB comprises more than two centuries of high-quality data about SCOTUS activities coded by experts. More than 200 categorical variables are contained in every case, including chronological variables, case background variables, justice-specific variables, and outcome variables. This database has been widely used and analyzed over the legal research community, including the area of judgment prediction. For instance, Katz et al. (2017) considered more than 28,000 cases of the SCOTUS, with the aim to predict the outcomes at both the case and justice level. In that work, RF classifiers were trained on categorical variables taken from the SCDB in addition to descriptive features manually extracted and engineered such as chronologically-oriented or justice-related features. It achieved 70.2% accuracy at the case outcome level and 71.9% at the justice vote level. That approach, however, was difficult to scale since it relied significantly on expert knowledge and human annotation (Virtucio et al., 2018).

On the other side, many studies on how to predict SCOTUS decisions have found that including information from legal texts can outperform prediction algorithms based solely on quantitative legal data (Ashley and Brüninghaus, 2009; Sim et al., 2015; Kaufman et al., 2017). One of the initial efforts toward explicitly integrating text features into prediction models was made in (Sim et al., 2015), where the authors estimated a topic model from the text of briefs to the SCOTUS. Additionally, Kaufman et al. (2017) made use of both

summary information and oral arguments. Oral arguments are textual data that outline the justices' oral arguments for each case, whereas summary information is data available from SCDB.

In the same vein, and in order to see if judicial decision-making was significantly influenced by the stimulation of case information extracted from the texts, researchers looked at the association between case outcomes and case facts based purely on textual content. For example, Agrawal et al. (2017) were among the first researchers to consider only the text of case documents for predicting the outcome of SCOTUS decisions. Specifically, they used documents from lower courts to predict the decisions of the higher courts by utilizing ML techniques such as Convolutional Neural Network (CNN) and RF. They predicted district-to-circuit court affirm/reverse judgments with a 79% accuracy using CNN, and by using RF, they predicted circuit-to-supreme court outcomes with a 68% accuracy. Using the RF algorithm also, Alexander et al. (2018) examined four RF models at different stages in the litigation process to predict its case-ending outcome. Features were manually extracted from 5,111 employment law cases from the U.S. District Court, and the model that performed the best achieved an accuracy of 94%.

Not all U.S. studies worked only on SCDB. There are many researchers who used other datasets extracted from law firms and companies. For instance, Branting et al. (2015, 2017) have also applied ML classification techniques such as Support Vector Machine (SVM)s, and more advanced DL approaches such as Hierarchical Attention Networks (HAN), for decision support in administrative adjudication, including routine claims for veterans benefits, social security disability, immigration status, and medicare appeals. They applied their approaches to predict outcomes from the texts of cases in a corpus of decisions of three datasets: motion-rulings, Board of Veterans Appeals (BVA) decisions; and World Intellectual Property Organization (WIPO) domain name dispute decisions. In another study, Alexander and Iannarone (2020) worked on 3,227 publicly available arbitration decisions from the Financial Industry Regulatory Authority (FINRA) in disputes between investors and broker-dealers.

In other countries than the United States, the use of text analysis methods to predict

the winning/losing outcomes of cases is also common. There are studies, for example, from Germany (Waltl et al., 2017), France (Octavia-Maria et al., 2017; Sulea et al., 2017), UK (Strickson and De La Iglesia, 2020), Brazil (Lage-Freitas et al., 2019), China (Long et al., 2019), Thailand (Kowsrihawat et al., 2018), and Philippine (Virtucio et al., 2018).

Using the metadata and textual features, Waltl et al. (2017) analyzed 5,990 documents from the German Federal Fiscal Court, and extracted 11 different features for each case. The evaluation of the performance of a Naive Bayes multinomial classifier has revealed that it can only achieve F1 scores between 0.53 and 0.58. For French text, Octavia-Maria et al. (2017) used SVMs to predict outcomes of French Supreme Court cases. On a collection consisting of over 126,000 documents, they extracted unigrams and bigrams from each case and assigned a corresponding label in terms of the result. They trained SVMs to predict the outcomes and reported an overall accuracy reaching 96% and an F1-score reaching 0.97 on predicting case outcomes. However, they studied the problem without the usual binary classification accept/reject model but employed multi-label classifications with six or eight labels. Working on the same dataset, Sulea et al. (2017) developed a mean probability ensemble system combining the output of multiple SVM classifiers. They reported accuracy scores of 98% for predicting a case ruling, 96% for predicting the law area of a case, and 87.07% for estimating the date of a ruling. In Brazil, Lage-Freitas et al. (2019) proposed a methodology for predicting Brazilian court legal decisions which was able to reach 79% of accuracy when employed for a Brazilian court dataset with 4,043 cases. Their approach was designed to predict case outcomes by using three different labels: yes, no, and partial. Moreover, their proposed method also predicted whether the decision will be unanimous.

For Chinese case outcome prediction, the application of deep learning was introduced in (Long et al., 2019) by developing a prediction system called AutoJudge. The model's architecture was composed of three Bidirectional GRU (BiGRU), each of which encodes pleas, fact descriptions, and relevant laws separately. In addition, a pairwise attention mechanism was designed to transform these encodings into more meaningful representations. Finally, a CNN layer was added to transform these representations into final classification vectors. They used 100,000 divorce proceedings from the Supreme People's Court of the People's

Table 2.1: Summary of previous works on the predicting case outcome task [Win/Lose]. Results with (%) mean accuracy; otherwise, mean F1 score.

Ref#	Data and Court	Method	Features	Best Results
Agrawal et al. (2017)	15K cases [SCOTUS]	CNN, RF	Ngrams, W2V	79% [CNN]
Katz et al. (2017)	28K cases [SCOTUS]	RF	NA	70.2%
Alexander et al. (2018)	5,111 cases [U.S. District Court]	RF	factors from text	94%
Alexander and Iannarone (2020)	3,227 cases [Financial Industry Regulatory Authority (FINRA)]	AdaBoost Regression Forest	factors from text	75%
Sim et al. (2015)	2,074 cases' briefs [SCOTUS]	logistic regression	bag of phrases	79.1%
Branting et al. (2017)	50K cases [Board of Veterans Appeals (BVA)], 5587 cases [World Intellectual Property Organization (WIPO)] [EN]	SVM, HAN	Ngrams	0.74 [BVA], 0.94 [WIPO]
Strickson and De La Iglesia (2020)	4,959 cases [UK court]	SVM, RF, LR, KNN, SLP, MLP	TFIDF, Doc2Vec	0.69 [LR]
Octavia-Maria et al. (2017)	126K cases [French Supreme court]	SVM	Unigrams+bigrams	0.96
Sulea et al. (2017)	126K cases [French Supreme court]	Ensembles SVMs	Unigrams+bigrams	0.98
Waltl et al. (2017)	5,990 cases [German fiscal courts]	Naive Bayes	Metadata and textual features [Bow, TF-IDF]	0.57
Virtucio et al. (2018)	27,492 cases [Philippine Supreme Court]	SVM, RF	Ngram and topics	59% [RF]
Kowsrihawat et al. (2018)	1,207 cases [Thai Supreme Court]	Bi-GRU	word embeddings	0.63
Lage-Freitas et al. (2019)	4,043 cases [Appellate court Brazil]	NN	TFIDF, Doc2Vec	79%
Long et al. (2019)	100K divorce cases [Supreme People's Court of China]	BiGRU + pairwise attention + CNN	TFIDF	0.83

Republic of China to train the model and other simpler DL models to predict whether the divorce will be granted. With AutoJudge, the best results were 82% accuracy and 0.83 F1-score. Kowsrihawat et al. (2018) used a similar method consisting of BiGRU encoders for encoding facts and laws, followed by additional attention and hidden layers for case outcome prediction. They tested their approach on more than 1,000 Thai Supreme Court binary classification cases. This model has the best F1-score of 0.63. Experiments have also been performed by Virtucio et al. (2018) to assess the performance of RFs and SVMs

in predicting the outcomes of Philippine Supreme Court decisions. Their best result was using a RF with an accuracy of 59%. Table 2.1 shows the summary of the datasets, features, methods, and results of the previous studies related to the case outcome prediction task.

Predicting Law Violation

In this task, the cases are categorized as either violations or non-violations of a certain article in a particular law (Aletras et al., 2016). Courts and judges usually decide the case based on a given set of facts and the applicable law (see Figure 2.4).

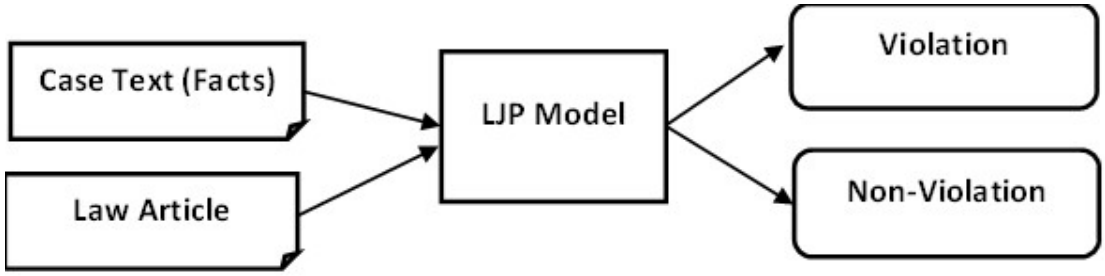


Figure 2.4: Predicting Law Violation (Binary)

In the previous few years, researchers have successfully attempted this task for the text of judicial cases in several languages such as English, French, and Chinese. The main influencer study was (Aletras et al., 2016) which was the first to apply this approach relying only on unstructured text features. The authors have used a dataset of around 600 cases from the European Court of Human Rights (ECHR). They built an SVM model with a linear kernel and used it on data represented by n-grams and topics in order to predict whether an article of the ECHR Convention (Articles 3, 6, and 8 were considered) was violated or not. They reported 79% accuracy as their highest score.

That work inspired other researchers to work on the same dataset. For example, Liu and Chen (2017) tested different classifiers on the same dataset. It compared kNN, Logistic Regression (LR), bagging, RF, and SVM in three sets of experiments for the Articles 3, 6 and 8 of the Convention. For all three sets of the experiments, SVM outperforms other models over the full feature space and over the topics feature space. Medvedeva

Table 2.2: Summary of the previous work on the predicting case violation task. Results with (%) mean accuracy; otherwise, mean F1 score.

Ref#	Data	Method	Features	Results
Aletras et al. (2016)	584 cases [ECHR]	SVM	BOW and topics	76%
Liu and Chen (2017)	584 cases [ECHR]	k-NN, LR, bagging, RF, SVM	Ngrams, topics	87.5% [SVM]
Medvedeva et al. (2018, 2020)	3,132 cases [ECHR]	SVM	Ngrams	74%
Kaur and Bozic (2019)	3,132 cases [ECHR]	CNN vs SVM	fasttext, Word2Vec,GloVe	82% [CNN]
Chalkidis et al. (2019)	11,500 cases [ECHR]	BiGRU-Att, HAN, BERT and HIER-BERT	GLOVE	0.82 [H-BERT]
O'Sullivan and Beel (2019)	9,703 cases [ECHR]	15 classical ML classifiers	Ngrams, W2V, Doc2Vec, Echr2Vec	68.83% [Avg]

et al. (2018, 2020) used also SVM but increased the number of cases and articles involved and analyzed different aspects of the same dataset. They conducted three experiments considering different features such as n-grams, the year of the cases or the judges' names. In (O'Sullivan and Beel, 2019), 15 classical ML models were constructed for 12 articles in the Convention for 9,703 cases from ECHR case judgments. The textual features used for training the models were n-grams, word embeddings, and paragraph embeddings such as doc2vec and a custom echr2vec. Using a realistic test set, the overall test accuracy, across all articles and models, was 68.83%.

The advancements in the areas of DL and NLP have made their way to this task too. For example, Kaur and Bozic (2019) developed 9 CNN models for 9 Articles of ECHR and compared them with the SVM models trained in the previous research (Medvedeva et al., 2020). CNN models outperformed SVM models as the former achieved an average accuracy of 82%, whereas the latter achieved 75%. A new dataset of ECHR was created by Chalkidis et al. (2019) totaling 11,500 cases. They reported the results of various DL architectures on three tasks: binary classification, multi-class classification and case importance prediction. They further introduced a Hierarchical- BERT, which first produced fact embeddings used with a self-attention mechanism to formulate the document embedding. Their Hierarchical-

BERT was the best-performing model in both binary and multi-label classification tasks with an F1-score of 0.82 and 0.61, respectively. A summary of the studies for the case violation prediction task is shown in Table 2.2.

Predicting Law Area, Applicable Laws or Charges

In addition to the previously mentioned tasks, some researchers have explored legal judgment prediction with the aim of simplifying the process by predicting some important information that could lead to the decision, for example, predicting the legal area, applicable law articles, or charges for a given legal case. Thus, some previous works consider it as a task of multi-class classification (see Figure 2.5).

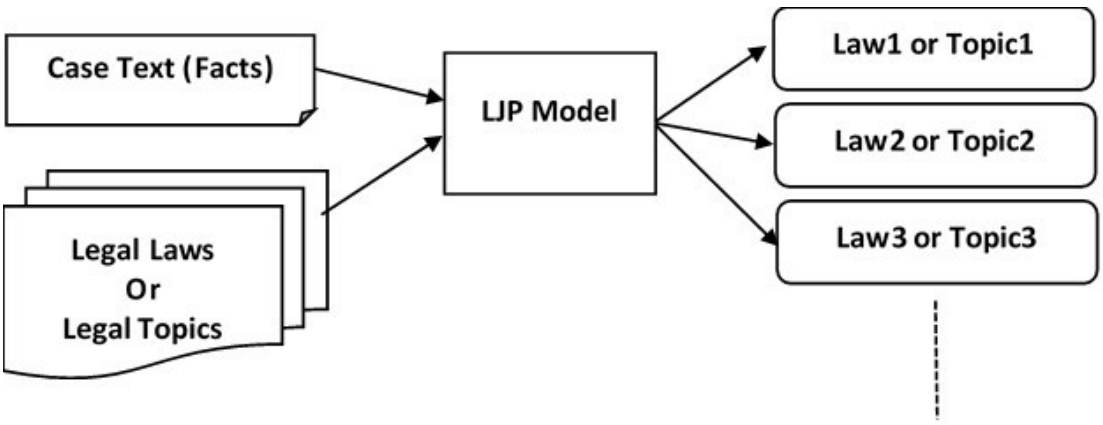


Figure 2.5: Predicting Legal Area or applicable Laws (Multi-class)

Ikram and Chakir (2019) applied four classical ML models to classify the Moroccan court rulings written in Arabic. The comparative analysis used in this study has shown that SVM algorithm is the most efficient algorithm with an accuracy of 98.11%. Octavia-Maria et al. (2017); Sulea et al. (2017) have done a similar investigation, predicting the law area of the French Supreme Court using lexical features and SVM. In (Undavia et al., 2018), deep neural networks such as CNN, LSTM and RNN have been used for classifying English legal court opinions for the SCDB. The authors compared several classical ML algorithms with several neural network systems in terms of law area classification, and they found out that CNN combined with Word2vec performed better compared to the other approaches and gave an accuracy of around 72.7%. Howe et al. (2019) explored multiple

Table 2.3: Summary of studies for Predicting Law Area, Applicable Laws, or Charges. Results with (%) mean accuracy; otherwise, mean F1 score.

Ref#	Data	Method	Features	Task	Results
Undavia et al. (2018)	8,419 opinions [U.S.] [EN]	CNN, LSTM, GRU	BoW, LDA, Word2vec,Doc2vec, FastText, Glove	Law Area Pred	72.4% [CNN]
Howe et al. (2019)	6,227 cases [Singapore] [EN]	linearSVM, Bert, ULMFiT	Topic models LSA, Glove	Law Area Pred	0.63 [SVM]
Chalkidis et al. (2019)	11.5K cases [ECHR] [EN]	BiGRU-AttHAN:LWAN, BERT and HIER-BERT	GLOVE	Articles Pred	0.82 [HIER-BERT]
Ikram and Chakir (2019)	1,452 cases [Moroccan Supreme Court] [Arabic]	NB, KNN, SVM, Decision Tree	TF-IDF	Law Area Pred	98.11 [SVM]
Sulea et al. (2017)	126,000 cases [French Supreme court]	linearSVM	unigrams +bigrams	Law Area Pred	0.96
Octavia-Maria et al. (2017)	126Kcases [French Supreme court]	Ensembles Classifiers	unigrams +bigrams	Law Area Pred	0.90
Yan et al. (2019)	20,4231 cases [CAIL dataset] [China]	CNN	Word Emb	Statute Pred	0.86
Liu et al. (2015)	1,518 cases [Chinese criminal judgments]	SVM+TextSimilarity	TF-IDF	Statues Pred	0.74 [Recall]
Zhong et al. (2018)	1M+ [CAIL2018 dataset] [China]	SVM, CNN, RNN	W2V	Charges,Articles	0.73, 0.78
Li et al. (2019)	1,367,654 cases [CAIL2018 dataset] [China]	BiRNN	W2V	Charges, Articles	0.91, 0.83
Xiao et al. (2018)	2.6M cases [CAIL2018 dataset] [China]	CNN	W2V	Charges, Articles	97.6%, 97.6%

ML approaches, DL models such as BERT and a shallow CNN to classify 6,277 Singapore Supreme Court Judgments into their 31 different legal areas. Their results showed that the statistical model performed best with an F1-score of 0.632 and an accuracy of 0.733.

A lot of work in the field of predicting law articles or charges is currently being done in China. Liu et al. (2015) proposed a three-step framework for retrieving Chinese relevant statutes, in which SVM-based multi-label classifiers combined with text similarity were utilized, and strong performance was demonstrated in real cases. Jiang et al. (2018) used RNN on China Judgment Online cases to achieve a 76.3% accuracy. Zhong et al. (2018) tried to predict law articles, charges, and terms of penalty by analyzing criminal cases from the Supreme Court of China using SVM, CNN and RNN. They found that CNN was able to make predictions with higher scores than SVM, whereas their RNN models failed to

converge. Their best CNN models predict law articles with an F1 score of 73.4%, charges with 78.3% and terms of penalty with 32.1%. Moreover, Li et al. (2019) repeated the same experiment with an RNN-based Multi-channel Attentive Neural Network (MANN) which allowed them to predict law articles with 0.83, charges with 0.91 and prison terms with 0.40 F1-scores; whereas Xiao et al. (2018) achieved 97.6%, 97.6% and 78.2% accuracies for the same dataset with CNN but the precision and recall results were not very high. A summary of the previous mentioned studies showing their dataset, methods, features, and results is shown in Table 2.3.

Legal Judicial Prediction Tasks within Canada

As mentioned before, over the last few years, researchers have been dedicated to predicting judicial case outcomes using NLP and ML methods. However, within Canada, the application of NLP to LJP is still new compared with several NLP tasks carried out on legal texts, most notably, text summarization (Farzindar and Lapalme, 2004; Farzindar, 2004; Yousfi-Monod et al., 2010), semantic search (Nejadgholi et al., 2017), information extraction (Lou et al., 2021), and information retrieval (Goebel et al., 2023).

For LJP, much of the Canadian research up to now has been conducted on the French Canadian case law. For example, Chehayeb et al. (2007) studied the classification of 567 construction claims in Canada into 12 categories. Their prediction module to forecast the outcome of a dispute using ANN achieved an accuracy of 65%. Queudot and Meurs (2018) outlined, in ongoing research, the limitations and perspectives of AI application in predictive justice and discussed the particular case of the Federal Court of Canada in the French language. Salaün et al. (2020) studied a multi-label classification task in predicting the types of verdicts in the domain of house law. The best model, based on the FlauBERT language model, achieves F1-scores of 0.937 and 0.849 in the Landlord vs. Tenant and Tenant vs. Landlord cases, respectively. With high annotation cost, Westermann et al. (2019) introduced two datasets drawn from judgments from the Régie du logement du Québec where they used factors to analyze and predict the outcome of landlord-tenant cases. For the first dataset, they identified 44 factors to tag 149 cases, then they used

a deep taxonomy of factors to annotate 39 cases drawn from the first dataset. With RF models, they were able to reach a precision of 0.665 for the first dataset and 0.70 for the second one.

To the best of our knowledge, the use of text classification to predict court rulings for the *English* Canadian case law is an under-explored area. There exist few Canadian studies in predicting legal outcomes from English case law. The recent studies (Dahan et al., 2020) and (Lam et al., 2020) are among the few examples of such attempts. Lam et al. (2020) applied multiple DL solutions to human-written case summaries that have been combined with other manually extracted factors to classify the reasonable notice period a plaintiff should be awarded. Their classification task had a total of 25 classes. The best-performing model was RoBERTa+base which achieved an accuracy of 69%. The study (Lam et al., 2020) was based on the previous work (Dahan et al., 2020) which analyzed the prediction of reasonable notice using statistical ML on a hand-annotated tabular dataset and indicated ML predictors to be poor predictors of reasonable notice.

In view of all that has been mentioned so far, and as of this writing, there does not appear to be any existing literature on using DL and NLP for the LJP of the Canadian appeal cases, and the aim of our study is to contribute to this area of research.

2.3.3 Explainability

Explainability is an important issue that is relevant to LJP. Given the importance of legal outcomes for individuals, it is essential that a model's decision is justified in order to increase the user's trust in the system. In addition, it is important for the outcome correction and debugging. Explanations, for example, assist in detecting and mitigating biases inherent in predictive models. Given the complex nature of legal reasoning and the potential for historical biases to influence judicial outcomes, transparent explanations are essential for understanding the underlying mechanisms driving predictions. By providing insights into how decisions are reached, explanations enable stakeholders to identify instances of bias, whether resulting from skewed training data, algorithmic shortcomings, or systemic disparities within the legal system. Moreover, explanations empower stakeholders to make

informed judgments about the fairness and validity of predictive models, facilitating the implementation of corrective measures to address bias and ensure equitable outcomes.

Jiang et al. (2018) used reinforcement learning to build a phrase-level rationale-augmented classification model for the charge prediction task. The model selects as its rationale the relevant textual phrases in the fact description. Using manually designed features, Zhong et al. (2020) encoded factors a human judge would usually consider when making a decision. A reinforcement learning model was presented for explainable LJP tasks on three Chinese datasets. Through a series of relevant questions related to the facts of the case, the model aims to predict the most likely crime. A post-explanation approach proposed by Malik et al. (2021) identifies the important sentences in a case relating to the system's prediction. By using the ablation approach, the importance of a sentence s is identified: if removing s changes the system's prediction significantly, then s is deemed to be important. In contrast, Chalkidis et al. (2021) presented a pre-explanation approach that predicts alleged law articles and the rationales behind them, the rationales being taken from the relevant case paragraphs. In light of this study, we will extend its approach to explainability by including paragraphs that include citations of prior cases in addition to statutes and laws. Our model was also developed in an ensemble setting and using multi-task learning, where the model is forced to pay greater attention during training to relevant paragraphs in addition to the main classification task.

2.3.4 Summary

In this chapter, several studies related to our research were reviewed. Although they share a common ground, the aforementioned related works have their differences in terms of the data and methods used. Our work mostly shares the common ground of these works. We aim to predict the outcomes of cases by looking solely at case documents provided by the courts. We use textual features and classify cases into one of two possible results, Allowed or Dismissed. These studies, while distinct in some respects, jointly demonstrate the feasibility of machine learning-based prediction and give success criteria from which we can draw conclusions. As demonstrated in this section, LJP is a new area for text

classification methods in Canadian case law and our research aims to contribute in this direction.

Chapter 3

CanAL-DS: CANadian Appeal Law Dataset

In this chapter, we describe the CANadian Appeal Law Dataset (CanAL-DS) that we created for this study. We gathered the decisions of different Canadian higher courts such as: the Supreme Court of Canada (SCC), Federal Court of Appeal (FCA), Court of Appeal for Ontario (ONCA), Nova Scotia Court of Appeal (NSCA), Court of Appeal for British Columbia (BCCA), and Court of Appeal for Saskatchewan (SKCA). The texts we have compiled are obtained from data made available online and the final corpus we have constructed spans cases from 1950 to 2022. The structure of the case texts differs for each court. The details of the pre-processing and labeling process will be discussed in this chapter.

3.1 Overview

In our approach, we focus on appeal courts in one country (Canada), like other studies (Aletras et al., 2016; Lage-Freitas et al., 2019; Strickson and De La Iglesia, 2020; Chalkidis et al., 2019; Medvedeva et al., 2018) that focused on courts within one country's legal system. Further, we share the same assumption of Aletras et al. (2016), as well as the majority of research studies, that "*there is enough similarity between (at least) certain*

chunks of the text of published judgments and applications lodged with the Court and/or briefs submitted by parties with respect to pending cases". Consequently, since we do not have access to the relevant data, we will base our predictive tasks on the text of published decisions, rather than on factual submissions. As a result, we introduce the CANadian Appeal Law Dataset (CanAL-DS), a collection of case proceedings (in the English language) from the Higher Courts of Canada. For a case filed at these courts, a decision ("allowed" vs. "dismissed") or ("accepted" vs. "rejected") is taken between the appellant/petitioner versus the respondent by judges while taking into account the facts of the case, ruling by lower court(s), if any, arguments, statutes, and precedents. Thus, the decision is relative to the appellant. In CanAL-DS, each of the case proceeding documents is labelled with the original decision made by the judges of the appeal court, which serve as the gold labels. In addition to the ground truth decision label, the dataset also includes cited precedents and regulations for each document - if any - and is complemented with paragraphs that cite these sources. Information of this type contributes to the enrichment of the constructed dataset, which can be used in other legal NLP tasks as well as to serve as the basis for the explainability experiments conducted later on.

3.2 Dataset Preparation

3.2.1 Data Collection

For all included courts, we extracted the appeal case decisions from publicly available court websites. To extract the full text of all the decisions, we downloaded them as PDF or HTML files and then converted them to text files. However, SKCA was a special case because there were no cases available on the official court website. We found the *Saskatchewan Court of Appeal Sentencing Digest Database*[1] which is a searchable resource that contains information of sentencing appeals heard by the SKCA from 1982 to the present. For each case, it provides some details such as the case ID, date, judges, the different entities involved in the trial, and short digests or summaries that include a small description of the

[1]https://www.lawsociety.sk.ca/legal-resources-library/research-tools/

Table 3.1: Details on the corpora used in this work.

Court	No. documents	Website
SCC	500	https://www.scc-csc.ca/home-accueil/index-eng.aspx
FCA	6939	https://www.fca-caf.gc.ca/fca-caf_eng.html
NSCA	6555	https://www.courts.ns.ca/
ONCA	23,259	https://www.ontariocourts.ca/coa/
BCCA	704	https://www.bccourts.ca/Court_of_Appeal/
SKCA	4236	https://sasklawcourts.ca/

facts of the cases followed by the outcome. However, there are no full texts available in this database. To obtain the full texts of the cases, we used the CanLII API [2] to extract metadata about each case which includes a link to the full text of the case which is available in PDF or HTML on the CanLII website[3]. In Canada, the Canadian Legal Information Institute (CanLII) is a non-profit organization managed by the Federation of Law Societies of Canada. CanLII aims at making Canadian law accessible for free on the Internet. The website provides access to court judgments, decisions, statutes, and regulations from all Canadian jurisdictions. We requested a researcher API key that allowed us to link the data that we got from the SKDB with the full texts in the CanLII website using the case identifiers. Table 3.1 shows the initial numbers of the collected documents from the available resources.

3.2.2 Data Cleaning and Pre-processing

Case documents are long pieces of text, and understanding the document structure can be essential for both pre-training and fine-tuning tasks. However, as discussed in Section 1.2, the legal text is known to be noisy, and is mostly written without any structure, which makes it challenging to extract the document hierarchy. Moreover, the Canadian higher courts do not follow a strict pattern in writing case judgements. Thus, working on these

[2]https://github.com/canlii/API_documentation/blob/master/EN.md
[3]https://www.canlii.org/en/

cases is more complicated and harder than working with other kinds of legal documents such as contracts for instance.

The documents of decisions are formatted as free texts but include practical information about the litigation such as date, name of the judges, and names of the parties involved. After listing the entities, decisions continue with the description of the facts and the reasons for the outcomes. The description states all the facts and procedures that led up to the appeal. It also outlines the various arguments presented by the parties and follows the court's reasoning. Lastly, in the conclusion part, the decision is stated as well as the names of the judges.

It is important to note that, even though these decisions do not follow a strict pattern or structure, there are still certain common keywords (or key phrases), usually written in all capitals or as separate titles, which identify the beginning of each section. For example, phrases like: "Introduction", "Facts", "Analysis", "Issues", "Conclusion" and so on. We search for these keywords and divide the document into 3 parts: "Case Header", "Case Body", and "Case Conclusion"; from where one of these keywords occurs. The ***"Case Header"*** consists of meta-information (e.g., initial portions of the document containing case number, judge name, dates, and other meta-information) from the proceedings. It might also state the outcome of the case which is usually mentioned again with more details in the ***"Case Conclusion"***. The part ***"Case Body"*** is used as the case description (used for training), and the parts ***"Case Header"*** and ***"Case Conclusion"*** are used for label extraction. There is no exact, non-changing set of such keywords, and the keywords we use are our choice. The set of used keywords might change with respect to the legal corpus or court. If none of the keywords are found in a document, the document is deemed un-splittable and is discarded. From those that are successfully split, a label is extracted from the ***"Case Header"*** or ***"Case Conclusion"*** parts with another keyword search (see section 3.2.3). After extracting labels from these sections, we used regular expressions to remove them and keep only the ***"Case Body"*** part. In practice, as pointed out by legal experts, the decisions are written towards the start and the end of the document. These start and end sections directly stating the decision have been deleted from the documents

in CanAL-DS since that is what we aim to predict.

Further, since we intend to annotate the cases in our dataset with the cited legislation and precedents, as well as identify the paragraphs that contain these citations, another pre-processing step was undertaken. Normally, cases of appeal courts are divided into numbered points or paragraphs. For this purpose, we used regular expression matching to identify paragraph boundaries from within the ***"Case Body"*** part by customizing the process with a set of numbering styles frequently seen in legal documents (such as [1], 1), 1., etc.). This real-world segmentation will assist in understanding the results of the explainability experiments since it is a common practice in the legal domain that paragraph-level segments of the cases are used for citations and reasoning.

A further point to be taken into account is that some of the studies in legal NLP, such as (Chalkidis et al., 2019) and (Paul et al., 2020), include in the pre-processing phase removing named entities (e.i. names of judges or locations) from the text in order to prevent the model from being biased toward demographic information. Although it is understandable that removing this bias is important when constructing a generic automated decision-making system; however, it is not preferable for predicting or classifying judgements (Medvedeva et al., 2021). Specifically, models used for prediction or classification may benefit from keeping this information in, given that the location of courts or names of the judges may offer relevant information about the case (Katz et al., 2017; Medvedeva et al., 2018). Moreover, the developers of the French Canadian Legal Language Model Specialized in Criminal Law (CriminelBART) Garneau et al. (2021) designed a cloze privacy test where the context contains a specific criminal charge and asked CriminelBART to predict the name of the defendant. The results of the experiment showed that the name most frequently predicted was the name of a judge appearing most frequently in the corpus, which suggests no concern about privacy. As a result, we decided to keep the anonymization steps for future work on other legal tasks that may require further investigation of the model's bias.

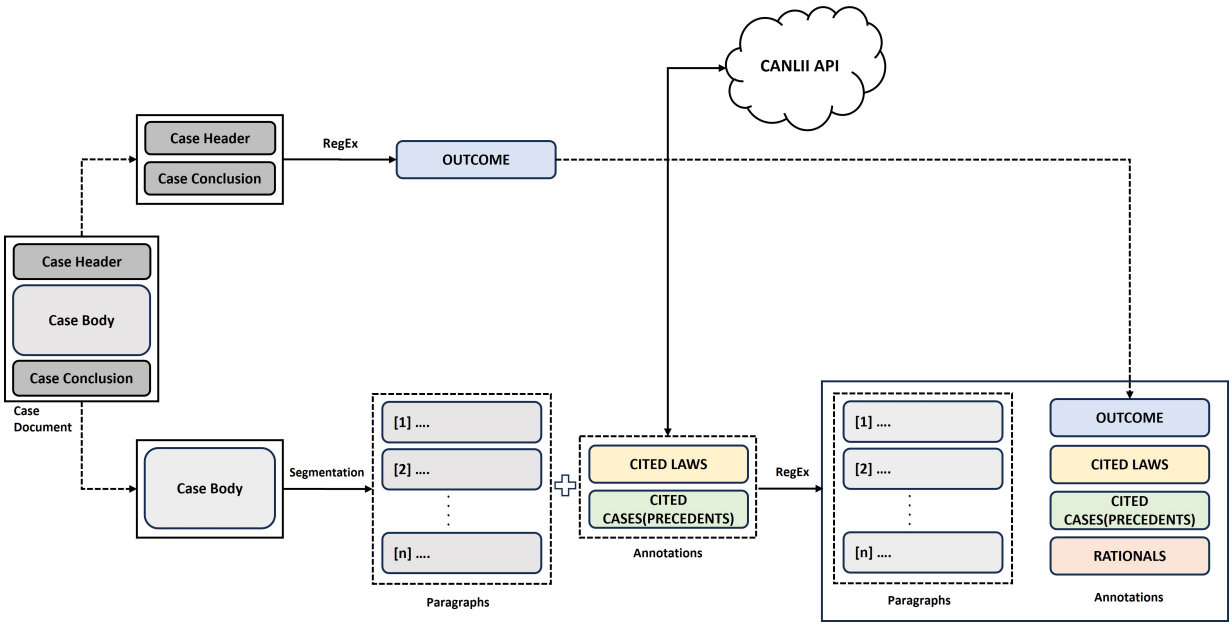


Figure 3.1: The steps of labeling a given case from our dataset.

3.2.3 Dataset Labeling

Figure 3.1 describes the steps involved in labeling a given case from the CanAL dataset. After collecting the full text of decisions, we had to extract our target variable, that is, the outcome of appeals. This information may be present within the case metadata available in the original resources, but it was not always available. Gathering correct labels for our dataset was not easy. As we mentioned before, we extract labels from the beginning of the text that is the "Case Header" and from the "Case Conclusion" starting from the end of the text. For the SKCA cases, we benefit from its digest text because it usually contains explicit expressions about the outcome, written by human experts.

Concerning judgments from appeal courts, the court will either confirm the first lower court decision (dismiss the appeal, i.e., the appellant" loses") or reverse it (allow the appeal, i.e., the appellant "wins"). However, the court can also partially confirm the judgment. In other words, the court can decide to accept only some of the appellant's requests, therefore partially reversing the decision. Empirically, we noticed that certain expressions and words are present in the majority of the decisions; thus, we resorted to hand-crafted keyphrase-based regular expressions. We choose the splitting and labeling set

of keywords from fundamental expressions denoting an outcome. Our choice of keywords is such that most of our case documents can be split and labelled successfully. If label extraction fails, those documents are discarded as well. Some examples of such keywords are listed in Appendix C.

To find such keyphrases and keywords, we used the FlashText[4] library and regular expressions. As a result, each case document will be automatically labelled with 'Allow' or 'Dismiss' outcomes using the above approach. The cases that could not be labelled with only one outcome were filtered out from the dataset. They represent cases in which the judges did not reach a final decision or cases with "Mixed" outcomes. We used several heuristics to identify and assess the quality of our labels from three data points:

- Case metadata or digest for SKCA cases;
- Header part of the document;
- Conclusion part of the document.

While creating these heuristics, we sampled the data and verified labels manually several times to ensure their quality. In the end, we used a simple voting classifier in which at least two out of three label sources must agree in order to consider that label valid. These procedures led to a lower number of cases than the total number in Table 3.1. A unified corpus consisting of 35,928 cases is obtained as a result of this compilation, 10,187 with "Allowed" as the outcome and 25,741 as "Dismissed". Table 3.2 gives a numerical overview of this collection.

After all the steps described above were carried out for each court, the resulting unified corpus was then split into training, validation, and test sets. We use the SKCA-based corpora as a test dataset and split the remaining documents into 90% for training and 10% for validation. By training and validating the model on documents from different courts, while reserving one court exclusively for testing, we ensure that the model's performance is evaluated on data it has not encountered during the training and validation phases.

[4]https://github.com/vi3k6i5/flashtext

Table 3.2: Details of the dataset before and after the cleaning, pre-processing, and labeling steps.

Court [span]	Before	After	Dismissed	Allowed
SCC [1905-2022]	500	490	294	196
ON [2014-2022]	23,259	19,894	14,953	4,941
BC [2003-2021]	704	624	433	191
NS [2003-2022]	6,555	5,788	3,905	1,883
FCA [2001-2022]	6,939	5,421	4,035	1,386
SK [1984-2021]	4,236	3,711	2,121	1,590
Total	**42,193**	**35,928**	**25,741**	**10,187**

This helps to assess the model's ability to generalize across various scenarios, ensuring its effectiveness in handling a broad range of higher courts within Canada. An overview of the statistics of the created corpora after all the splitting and labeling can be seen in Table 3.3 and Figure 3.2.

Table 3.3: Size and label distribution for the training, validation, and test datasets.

Dataset split	No. documents	Dismissed	Allowed
Train	28,995	21,258	7,737
Validation	3,222	2,362	860
Test	3,711	2,121	1,590
TOTAL	**35,928**	**25,741**	**10,187**

3.2.4 Adding Legislations, Precedents and Rationales

For the purpose of the LJP, the preparation of the dataset described in previous sections is sufficient. However, we aim to develop models whose decisions can be justified using additional steps. The inspiration for this step came from the related work (Chalkidis et al., 2021) that introduced "paragraph-level rationales" as prominent paragraphs referred to in

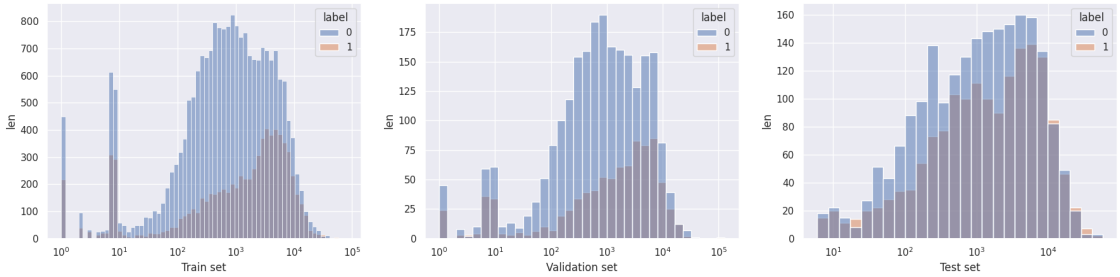


Figure 3.2: The distribution of the lengths of the dataset splits after pre-processing.

the court's decision. It made use of citations that correspond to laws referred to in the texts of the cases during the reasoning of the judgments. This legal reasoning approach, of relying on the parts that include citations to law sources, has proven beneficial in legal practice. For example, in the Canadian case-law competition COLIEE (Rabelo et al., 2022), the participating team Ma et al. (2021) achieved first place among other participants by utilizing this idea in some tasks. For this purpose, each case in our dataset was mapped with the legislations as well as the prior cases (precedents) that were cited, if any, and their cited paragraphs were extracted as the rationales. To do that, we first extracted the cited legislations or prior cases in each case using CANLII API[5]. Then, by using regular expressions, we identified paragraphs that contain these citations. Fig 3.1 shows all steps involved in labeling a given case from our dataset.

The results of this process allow us to obtain decision rationales for 45.57%, 46.80%, and 68.39% of training, validation, and test samples, respectively. Additionally, with 1,852 different legislation labels, there is a considerable class imbalance as many of these labels occur less than 50 times, of which only 1,128 occur in the training set. The distribution of the number of cases by legislation based on our exploratory data analysis is visualized in Figure 3.3. Interestingly, in all dataset splits, Act RSC 1985, c C-46, the Criminal Code, is the most commonly seen one, followed by Act Schedule B to the Canada Act 1982 (UK), 1982: The Constitution Act.

[5]https://github.com/canlii/API_documentation/blob/master/EN.md

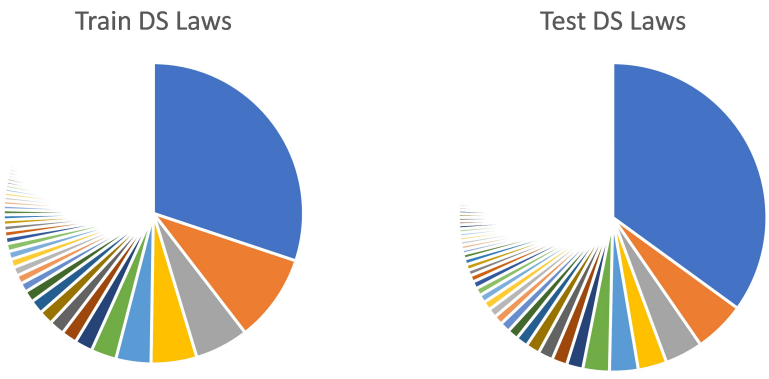


Figure 3.3: The distribution of the number of cases by law acts.

3.3 Summary

In this chapter, we introduced our dataset created for the purpose of this research. The chapter began with a description of the data preparation and collection. Next, we explained the procedure followed for data preprocessing and labeling. Finally, we illustrated how this dataset will be used in this research. In the next chapters, we will employ the described dataset for different legal NLP tasks.

Chapter 4

CanAL: CANadian Appeal-Law Model

4.1 Overview

To generate useful text embeddings using BERT-based language models, it is necessary to first train the models on extensive data. However, in cases where there is limited data available for specific domains such as the law domain, it may not be practical to train an entirely new language model from scratch. Instead, further pre-train existing state-of-the-art language models may suffice (Herrewijnen et al., 2023). Essentially, further pre-training involves continuing training a pre-trained model on a domain-specific corpus to improve the model's knowledge on that domain (De Kruijf et al., 2022). Examples of such models are JuriBERT (Douka et al., 2021) and CriminelBART(Garneau et al., 2021). In comparison with pre-training from scratch, further pre-training has two main advantages. First, further pre-training typically begins with a pre-trained model, such as BERT, which has been trained on a large amount of text data. As a result, these models have a solid foundation of generic language before being further pre-trained. Second, in contrast to pre-training from scratch, further pre-training requires fewer training steps, making it computationally simpler and potentially less costly.

In this thesis, we explore the potential benefits of further pre-training for BERT-based

models in improving downstream classification tasks in the legal domain. Our findings suggest that for these tasks, conducting additional pre-training on a narrow domain dataset can yield results comparable to those achieved by retraining models on larger domain datasets. This suggests that a complete retraining of the language model may not be essential for enhancing performance in downstream tasks. Rather, it might be enough to make small modifications to existing state-of-the-art language models, like BERT, instead of training a completely new language model. By making these slight adjustments to the current models, we have the potential to achieve the desired outcomes without the requirement of extensive pre-training on large datasets (Herrewijnen et al., 2023).

Many attempts have been made to fine-tune pre-trained language models, making them more useful in the legal domain (Chalkidis et al., 2020; Douka et al., 2021; Garneau et al., 2021). These recent advances are all distinguished by specific underlying legal characteristics, which may be related to specific language, law, or country. Therefore, in this work, we propose a new country-specific language model that specializes in Canadian **"Appellate Case-Law"**, an area that has recently gained more attention (Vazirgiannis et al., 2020; Waltl et al., 2017; Jacob de Menezes-Neto and Clementino, 2022; Sivaranjani et al., 2021; Song et al., 2022). It is essential to consider one important issue for this model; which is the type of documents it handles that vary significantly from other kinds of legal documents such as contracts or trial cases documents. Appellate law refers to the legal procedures and policies involved in appealing a judicial decision. It is the set of rules that the losing party must follow when presenting their case to a higher court. In an appeal, the applicant is responsible for presenting their case to a higher court that has more authority than the original trial court. In this process, the goal is to provide persuasive arguments in order to correct any errors made by the trial court judge or alter the interpretation of statutory law.

It is therefore essential to incorporate this specialization into a language model. However, the legal field is vast and complex, and as a lawyer cannot be an expert in every domain, we considered it unwise to develop a generic legal model for very specific tasks in the legal field, like predicting the outcome of legal appeals of a particular case-law. This

is because the aspects of legal proceedings vary widely depending on the type of case, the jurisdiction, and other factors. In order to create a truly useful tool for predicting appeal outcomes, it is necessary to develop a specialized model for the appellate law. This will enable us to provide more accurate predictions and better serve the needs of legal professionals in that field. By focusing on a specific area of law, such as Canadian appeal case law, we can create a model that is tailored to the specific needs of that area. With this in mind, we found that further pre-training legalBERT model on the Canadian appeals corpora should be considered and may prove beneficial. We believe that such a specialized legal model can perform better compared to generic or non-Canadian legal models. An overview of the stages of developing this model is shown in Figure 4.1.

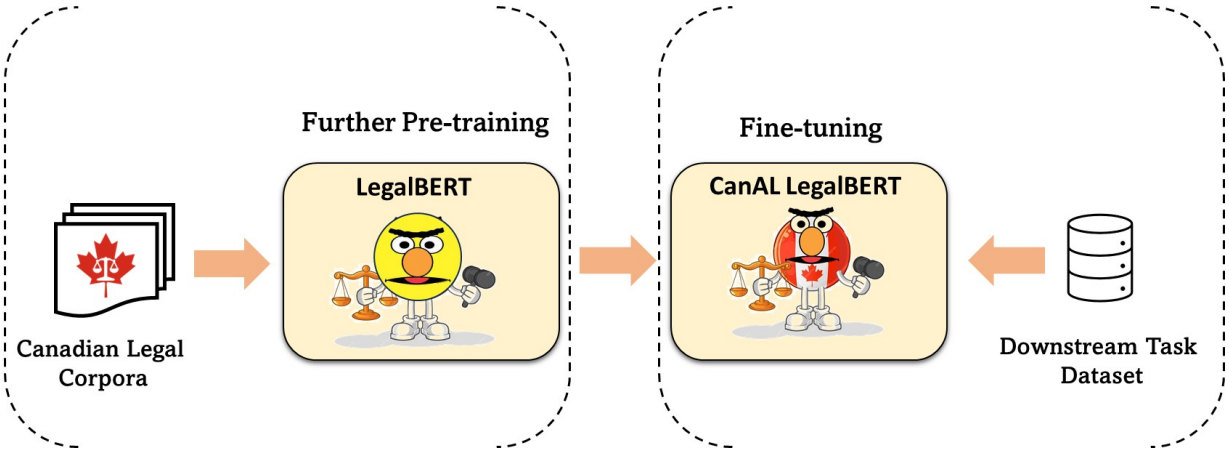


Figure 4.1: The two stages of employing CanAL LegalBERT: (a) further pre-traning LegalBERT on Canadian corpora, and (b) fine-tuning the pre-trained model for downstream Canadian NLP tasks.

The corpus we are working with (further described in the following section) is a Canadian legal corpus specific to the appellate law. It contains several appeal cases we collected from many Canadian higher courts. This corpus, comprising 50,641 judgments has been gathered from public Canadian legal websites. To construct a specialized model from this corpus, we propose to further pre-train LegalBERT (Chalkidis et al., 2020) using the transformers library for 75K steps achieving a perplexity of 2.67. We then assess the "legal expertise" of our Canadian Appeal-Law (CanAL) LegalBERT with two legal end-tasks.

Table 4.1: Further pre-training Corpus Statistics.

Type of Files	**# of Files**
Annual Statutes	600
Regulations	4,538
Consolidated Acts	918
Case Law	44,585
Total Files	50,641

4.2 Further Pre-training Stage

4.2.1 Corpus Collecting

Training Corpus. As the first step to build a Canadian Appellate Law language model, we tried to collect public Canadian Appellate-Law-related corpora. We collected a large corpus of case documents from many higher courts in Canada. These documents were collected from diverse publicly available sources on the web, such as the official websites of these courts e.g., the website of the Canada Supreme Court[1], the website of the Legal Information Institute of Canada, the popular legal repository CanLII[2], and so on (see Section 3.2.1). The court appeal cases in our dataset range from 1950 to 2022, and belong to all legal domains, such as civil, criminal, constitutional, and so on. Additionally, we collected 6,056 Central Government Acts from the Justice Laws website [3], which constitute the country's legislative body. The collection contains annual statutes: 600 files, regulations: 4,538 files, and consolidated acts: 918 files. In total, our dataset contains 50,641 Canadian legal documents (all in the English language). The raw text corpus size is around 4.5 GB. Finally, we obtained around 3.5 GB of clean legal text data to proceed with the domain-adaptive pre-training.

Corpus Pre-processing. Before we started the training, data pre-processing was

[1]https://www.scc-csc.ca/

[2]https://www.canlii.org/en/

[3]http://laws.justice.gc.ca/eng

performed. The text was cleaned from non-English characters. We split the entire corpus into train and test splits and included all the acts in our training split. The case documents were distributed randomly into the splits in the ratio of 9:1, with 46,409 files in the train split and 4,232 in the test split. The combined splits provide us with a collection of raw English legal text of size 3.5 GB that we will use to further pre-train our model. The documents in both the train and test splits were divided into equal-sized disjointed sequences or chunks of contiguous text, disregarding sentence boundaries. Each chunk is a unique training or testing sample, and the maximum chunk size was set to 512, following previous studies. The method described above gives us a total of 599,862 training sequences and 40,788 testing sequences.

4.2.2 Further Pre-training Details

Among the steps in the transfer learning workflow, pretraining a language model is the most expensive step (Horsuwan et al., 2020). Generally, this step is only performed once before fine-tuning is carried out on the downstream task. We choose to further pre-train LegalBERT[4] in order to better explore the impact of using country-specific corpora in pre-training. CanAL-LegalBERT has the same architecture as BERT and Legal-BERT with 12 layers, 768 hidden units and 12 attention heads (110M parameters). We train our model on the Masked Language Modeling (MLM) task. This means that given an input text sequence, we mask tokens with 15% probability and the model is then trained to predict these masked tokens. Following Devlin et al. (2019), we run additional pre-training steps of Legal-BERT on Canadian domain-specific corpora. Our 599862 training sequences divided over batches of size 32 equals approximately 19K steps per epoch. Therefore, further pre-training of the final model was done for 75K steps, with around 4 epochs.

A learning rate of 5e-05 was used along with an Adam optimizer (ß1=0.9, ß2=0.999) with a weight decay of 0.1, and a linear scheduler with 10,000 warm-up steps. We have trained our model using Tesla V100-PCIE-32GB with 2 GPUs. The batch size was set at 16 per GPU, so the total train batch size (w. parallel, distributed & accumulation) is

[4]https://huggingface.co/nlpaueb/legal-bert-base-uncased

32. For both further pre-training and fine-tuning tasks, we used the Datasets package[5] by HuggingFace (HF) for dealing with loading, tokenizing and formatting. We also used the Transformers package [6] by HF to load the models and run training and evaluation tasks.

4.2.3 Impact of Further Pre-training

MLM Evaluation. As an evaluation of the language modeling capabilities of the models, we use the **Perplexity** metric:

$$Perplexity = e^{L^{MLM}} \tag{4.1}$$

where L^{MLM} represents the Cross-Entropy Loss over the test set for the MLM task. Perplexity is one of the most common metrics for evaluating language models as it provides an indication of the level of confidence that the model has on the observed sequence. A lower perplexity indicates a better model.

In order to obtain an idea of the quality of further pre-training, the model is evaluated on the test set (10% of the corpus, as previously stated). For comparison, we selected three open-source language models available on HuggingFace, including the standard BERT[7] base (Devlin et al., 2019), and two adapted legal BERT models from different authors: Legal-BERT[8] (Chalkidis et al., 2020) and IndLegalBERT[9] (Paul et al., 2023). The two latter models are based on the same architecture as the BERT-base. All models are available via HuggingFace API. Their descriptions are listed in Table 4.2.

In addition to evaluating these models on our corpus, we report their performance on two other legal corpora where one is a Canadian case-law corpus from the COLIEE "Competition on Legal Information Extraction/Entailment" (Kim et al., 2022) and the second one is ILDC, the "Indian Legal Documents Corpus" (Malik et al., 2021) which is

[5]https://huggingface.co/docs/datasets/
[6]https://huggingface.co/docs/transformers/index
[7]https://huggingface.co/docs/transformers/model_doc/bert
[8]https://huggingface.co/nlpaueb/legal-bert-base-uncased
[9]https://huggingface.co/law-ai/InLegalBERT

Table 4.2: Models for comparison. The last row is about the further pre-trained Language Model developed in this work.

Model	**Type of model**	**Data Source**
BERT	Genric LM	Wikipedia and BookCorpus
LegalBERT	Domain-specific LM	(12GB) EU, UK, US Legislation, Court Cases, and Contracts.
IndLegalBERT	Country/Domain-specific LM	(27GB) Cases from Indian Supreme Court and High Courts.
CanAL-LegalBERT (this work)	Country/Domain-specific LM	(3.5GB) Cases from Canadian Appellate Case Law + Canadian Law Statutes and Acts.

related to the Indian legal system. The aim of this comparison is to prove that country-specific pre-trained models can perform better than their equivalent generalized ones in the legal domain.

Table 4.3: Quality of further pre-training measured via perplexity. All perplexity values are reported over the test sets.

Model	**Perp. [CanAL]**	**Perp. [ILDC]**	**Perp. [COLIEE]**
BERT	15.09	25.76	59.81
LegalBERT	5.04	7.13	11.07
IndLegalBERT	11.44	**5.26**	21.37
CanAL-BERT	**2.67**	8.57	**6.15**

Table 4.3 shows the perplexity values of the various models over the test sets. Generally, all the pre-trained legal models demonstrate their greater understanding of the legal language by outperforming BERT massively in terms of perplexity. The results also demonstrate that each country-specific model performed optimally on the same country dataset. As an example, the Indian model IndLegalBERT has the lowest perplexity (5.26) among the other models on the Indian dataset ILDC, which is a result of it having been

trained on the same dataset. Importantly, on Canadian datasets, the Canadian model CanAL-LegalBERT achieves lower perplexities (2.67 and 6.15) than other LMs. This makes CanAL-LegalBERT easier to adapt to Canadian case law than other general or non-Canadian legal models.

4.3 Fine-tuning Stage

We fine-tuned the further pre-trained model on several practically important end-tasks over both Canadian legal text as well as legal text from other domains/countries. For comparison, we chose the same aforementioned models in table 4.2 as baselines. Our methodology included selecting one simple architecture, replacing the encoder module in it with the previously mentioned baselines or our further pre-trained model CanAL-LegalBERT, and then comparing the performance of the models. Using this procedure, any difference in performance can be attributed solely to the LM used in the encoder module. We evaluated these models on two end-tasks namely: Legal Judgment Prediction (LJP) and Law Statute Identification (LSI) as described in the following sub-sections. For every task, we also compare results obtained using our model on its dataset with the best results reported in the original papers that introduced the non-Canadian legal models we are comparing with.

4.3.1 Model Testing on Downstream Legal NLP Tasks

Task1: Legal Judgment Prediction (Binary Classification)

Recent years have seen extensive research into automatically predicting the outcome of court cases, as discussed in Section 2.2.3. Typically, this is accomplished through the process of LJP, in which we are to predict the outcome of a case based on an analysis of the full description of the case. In this respect, the LJP task within appellate law is to predict the final decision of the court based on the case description (i.e., the input is a case judgment document with the final decision removed). Typically, in a Canadian court

case, the appellant files claims against the respondent, and the judge decides whether they should be "allowed" or "dismissed". Thus, this can be modelled as a binary text classification task.

Task2: Legal Statute Identification (Multi-label Classification)

As a fundamental principle of law, a country's legal system is governed by written laws (statutes). In such systems, these laws are often cited in court proceedings depending on the facts and evidence presented. A legal statute identification system (LSI) uses the details of the case to automatically identify the relevant statutes (i.e., the ones that might be violated). It is a widely studied problem and is often categorized as a sub-task of the larger problem of LJP. It can be modelled as a multi-label text classification task.

Datasets

We evaluated the models on the aforementioned text classification tasks using different datasets.

CanAL DS. This dataset has been described previously in detail in section 3. We used it for both LJP and LSI tasks.

ILDC DS. For LJP, we also used a subset of the ILDC dataset[10] introduced by Malik et al. (2021) and released by the organizers of the LegalEval2023 shared task (Modi et al., 2023). ILDC contains the case proceedings from the Supreme Court of India, and It has two different types of training data namely ILDC-single and ILDC-multi. For ILDC-single, one decision is reached for a petition or all the petitions together, while ILDC-multi documents include a variety of petitions involving different decisions. Document labeling assigned a single label for each document 'accept' if at least one appeal in that case document was accepted, 'reject' otherwise. For our current experiment, we consider the ILDC-multi dataset to compare with the original paper of IndLegalBERT (Paul et al., 2023).

[10]https://github.com/Exploration-Lab/CJPE

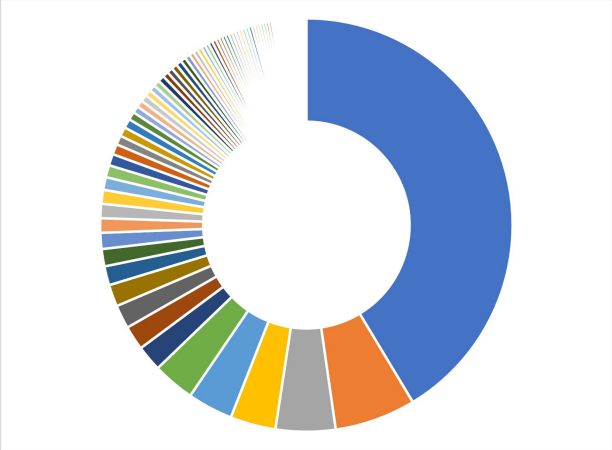


Figure 4.2: The number of citations per law in the test dataset.

ILSI DS. The Indian Legal Statute Identification (ILSI)[11] Dataset based on criminal case documents from India Paul et al. (2022). The dataset consists of 65,000 examples, split into training, development, and test sets, and a target set of 100 statutes. The facts are derived from criminal court cases from the Supreme Court and 6 High Courts of India. All 100 statutes (class labels) are part of the Indian Penal Code, which contains the defined laws for most criminal procedures in India. More details of the dataset are available in (Paul et al., 2022).

Evaluation.

We evaluated the models using three standard evaluation metrics (i.e., Precision, Recall, and F1-score), motivated both by compliance with research standards and by comparability with existing literature. More specifically, we used a macro F1-score to evaluate our binary classification task, which means that we weighed the performance of both prediction classes equally. For the multi-label classification task, however, due to the high label imbalance in identified statute frequency as indicated by Figure 4.2, we employed a micro F1-score (μ-F1), meaning that we weighed the performance of each class by the frequency of the corresponding class label.

[11] https://github.com/Law-AI/LeSICiN

Fine-tuning Details.

We use the BertForSequenceClassification[12] architecture, which is essentially a sequence classification head on top of a BERT-based model, i.e., a linear layer on top of the pooled output where the output dimensions represent the number of classes in a particular dataset. Due to the maximum input token limit of this vanilla BERT-based model, we used only the last 512 tokens as input similar to the original paper of the Indian model that we compare with. The models were trained for 4 epochs (with early stopping) with Binary Cross Entropy Loss. We also experimented with a batch size of 16, a learning rate of 5e-5, and a single NVIDIA A100-SXM GPU.

Results of the Downstream Tasks

LJP Results. The results of this downstream evaluation task are presented in Table 4.4.

Table 4.4: Performance over CanAL DS for the LJP task.

	CanAL DS		
Model	**Pr**	**R**	**F1**
BERT	50.17	50.02	39.34
LegalBERT	68.843	57.660	62.76
IndLegalBERT	68.169	60.788	64.27
CanAL	**76.88**	**74.50**	**74.95**

It is clear from the results that the vanilla BERT model, which was pre-trained over only general domain data, performs the poorest across all metrics. Other models leverage the pre-trained knowledge of legal data and obtain massive gains over BERT. Additionally, CanAL LegalBERT outperformed LegalBERT and IndLegalBERT on CanAL dataset by some margin, even though these models were trained with larger datasets and more steps. This could be because IndLegalBERT has been trained for more steps on a dataset of

[12]https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification

Table 4.5: Comparison of LJP results on ILDC Indian test DS with related work.

	Model	F1
Models from (Paul et al., 2023)	BERT	71.14
	LEGAL-BERT	78.21
	IndLegalBERT	83.09
Model from this work	CanAL LegalBERT	73.70

Indian legal documents helping it to generalize better for the Indian domain rather than the Canadian domain (as shown in Table 4.5), whereas LegalBERT was used as the base model for our model, so it performed better for the Canadian dataset. This indicates the importance of country-specific legal language models in the domain of natural language processing and law.

LSI Results. According to the results reported in Table 4.6 and Table 4.7, there is a similar pattern of behavior among all the models for this extreme multi-label classification task.

Table 4.6: Performance over CanAL test DS for LSI task.

	CanAL DS		
	PR	**RE**	**μ F1**
BERT	90.88	20.79	33.84
Legal-BERT	85.37	18.43	30.32
IndLegalBERT	91.12	20.15	32.99
CanAL	80.97	32.98	**46.87**

It is clear that although precision is increased, recall is significantly reduced. It could be attributed to the higher data dimensionality, which could be regularized in the future to possibly achieve better results. Still, our baseline fine-tuned CanAL model outperformed all models fine-tuned on CanAL dataset with a micro F1-score of 46.87%. Similarly, the Indian model has a micro F1-score of 64.58% over the ILSI dataset on which the model was first

Table 4.7: Comparison of LSI results on Indian test DS with related work.

	Model	PR	RE	**μ F1**
Models from (Paul et al., 2023)	BERT	82.12	49.07	59.11
	LEGAL-BERT	83.98	53.83	63.89
	IndLegalBERT	82.42	55.16	**64.58**
Model from this work	CanAL LegalBERT	74.77	58.84	60.63

trained on. We are therefore able to answer our first research question that country-specific legal models can perform well against generic legal models or legal models originating from other legal systems. We also believe that because the multi-label classification task is significantly more difficult than the binary prediction task, developing legal systems specifically tailored to this task that incorporate legislation's representations as well as training them on large legal text corpora will be a significant asset. The task difficulty is likely also the reason why the vanilla legal models have low performance results although they achieve better results in the binary classification task. A further improvement in multi-label classification results may be achieved by adding more data, improving architectures, or selecting fewer labels.

4.4 Summary

According to the results of our experiments, further training of LegalBERT for a more specific case law performs better than general legal language models. Therefore, we presented CanAL-LegalBERT, a model trained exclusively for Canadian appeal law texts and capable of making decent predictions on two main legal NLP tasks: Legal Judgment Prediction and Law Statutes Identification. This model can be considered as a base model that can be applied to a wide range of Canadian legal textual applications. In the following chapters, we will explain how we employ this model to develop an architecture for the task of legal judgment prediction within the Canadian appellate law area.

Chapter 5

Legal Judgment Prediction (LJP)

5.1 Overview

Legal prediction studies discussed in Chapter 2 covers a variety of issues and use different methodologies and algorithms. In our research, we conduct a series of experiments on the LJP task over the CanAL dataset and using the CanAL model. Our task for model development and evaluation contains three stages. First, we build a variety of baselines using pre-trained BERT-based language models. As a second step, because these models are only capable of processing a limited number of tokens, we proceed by developing *Hierarchical* language models (H-CanAL), which embed paragraph knowledge prior to producing an aggregate case embedding that can be used for classification. Finally, we introduce (EH-CanAL), an *Ensemble Hierarchical* Canadian Appeal Law LegalBERT model, which combines the strengths of three fine-tuned Canadian legal models using voting between them to determine the final outcome. Detailed descriptions of these models and the conducted experiments are provided in the following subsections.

5.2 Methods

Appeal documents are long and have specialized vocabulary compared to general corpora used for training general text classification models and language models. Initially, we

experimented with non-neural models based on text features (e.g., n-grams, and tf-idf) and existing pre-trained models (e.g., pre-trained word embeddings-based models), but none of them were better than a random classifier. Consequently, we developed neural models for our setting. In particular, we ran a set of experiments and came up with three different types of models: *Baseline* transformer-based model, *Hierarchical* transformer-based model, and *Ensemble Hierarchical* transformer-based model.

5.2.1 Transformer-Based Model [Truncation]

As baselines, we use already pre-trained versions of BERT (Devlin et al., 2019), and LEGAL-BERT (Chalkidis et al., 2020) models available on Hugging Face to be fine-tuned on our classification binary downstream task. We made this decision because these models provide a great baseline in different ways: BERT is the original and most used transformer-based NLP model, while LEGAL-BERT is a BERT model completely trained on legal text corpora. Additionally, all these models are based on the same basic architecture. Every input text is formatted by appending a special token - [CLS] acting as a representation of the entire sequence used for the classification task, and [SEP] token that is used to mark the end of the sequence. The main downside of these models is that the maximum sequence length is set to 512, which means that long legal texts have to be truncated. Usually, the key information of a document is at the beginning and end. We investigate two different methods of truncating text to perform fine-tuning. (1) *head-only*: keep the first 510 tokens; (2) *tail-only*: keep the last 510 tokens. We observed that using the last 512 tokens gave the best performance.

After the input sequence goes through the encoder, the [CLS] token which is regarded as the representation of the whole input sequence can be used as an input to a feed-forward neural network with a sigmoid activation function for performing single sequence classification. To classify texts according to labels, the BertForSequenceClassification[1] model can be used, which is essentially a sequence classification head on top of a BERT model, i.e., a linear layer on top of the pooled cls output where the output dimensions

[1]https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification

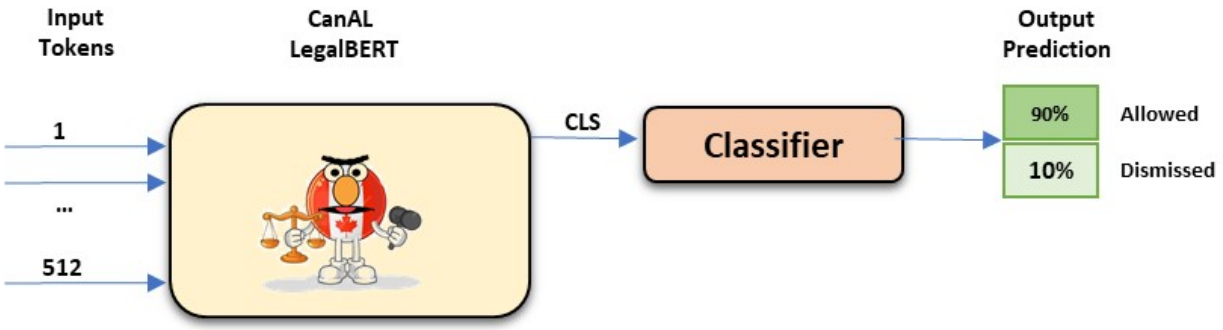


Figure 5.1: A CanAL model with a Classification linear layer on top (BertForSequenceClassification).

represent the number of classes in a particular dataset. An overview of this architecture is given in Figure 5.1.

5.2.2 Hierarchical Methods

While BERT has been proven to be very effective in NLP tasks, it still suffers from a significant drawback. It can only pass up to 512 tokens through its layers. This means that longer documents have to be truncated to the maximum sequence length. Various studies try to solve this problem by selecting specific sections of a legal document, while others propose some adaptations of the original transformer model to overcome this problem. Sun et al. (2019) finds that cutting out the middle section of a text rather than keeping just the head or tail gave the best results on the IMDb dataset. There can be a lot of information in the first paragraph of a document, but legal texts normally exceed the token limit, so truncating such a document would result in the loss of a significant amount of information. Moreover, Longformer addresses the limitation length with a new attention mechanism that scales linearly (Beltagy et al., 2020), showing a slight performance improvement over RoBERTa for QA and text classification. However, due to the large number of tokens being processed, it requires a greater amount of resources, such as GPUs and RAM. Considering that BERT is relatively light compared to these complicated models, it makes sense to attempt to improve upon existing BERT models, such as LegalBERT. To address its limitation and go beyond the 512-word token limit, we build a hierarchical architecture

which captures long-term dependencies and can use BERT-based model as a base model.

Several previous studies (Malik et al., 2021; Limsopatham, 2021; Paul et al., 2023) have introduced hierarchical models for classifying legal documents at the sentence or chunk levels. However, we developed our pre-processing approach (described in section 3.2.2) to accommodate paragraph-level segments, as is commonly used in legal document citation practices. More specifically, previous methods split the cases' text into sentences or fixed-length chunks without controlling the coherence of the text segments. On the other hand, our proposed model examines the document in terms of its actual numbered paragraphs and divides it into chunks accordingly.

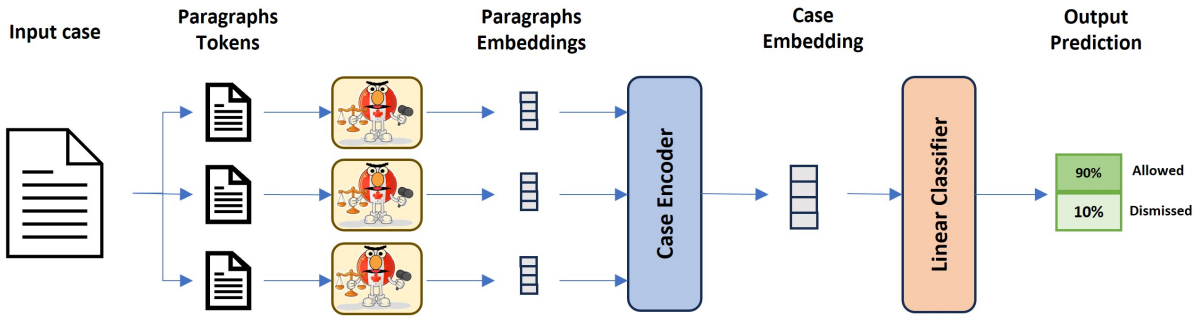


Figure 5.2: Overview of the Hierarchical model.

As illustrated in Figure 5.2, after splitting the full case document into smaller chunks $D_i = [P_1, \ldots, P_m]$, these chunks are then passed on to CanAL LegalBERT encoder to extract paragraph representations:

$$H_i = CanAL\_Encode(D_i) \tag{5.1}$$

where $H_i = [\ h_1^{P[CLS]}\ , ..., h_m^{P[CLS]}\ ]$. The resulting [CLS] representations of these chunks are then concatenated and passed on to an additional intermediate layer before the final layer, which is the linear classifier. In our hierarchical network, this additional layer works as a "Case Encoder". It receives the sequence of the chunks embeddings while its output serves as a single case embedding:

$$C_i = (Case\_Encoder(H_i)) \tag{5.2}$$

Then, the aggregated embedding C_i from the case encoder is passed to a fully connected layer with sigmoid activation to obtain the final prediction:

$$p_i = Sigmoid(Dense(C_i)) \tag{5.3}$$

where p_i is the final predicted probability $\in \{0,1\}$.

For the "Case Encoder" component, we experimented with both a BiGRU recurrent layer as well as a transformer with a simple self-attention layer :

(a) Case encoder based on Bi-GRU + attention. (b) Case encoder based on transformer.

Figure 5.3: The Case Encoder component of the Hierarchical model with different approaches of encoding.

- **Case Encoder based on BiGRU (Schuster and Paliwal, 1997):**

  A Bidirectional Gated Recurrent Unit (BiGRU) is a type of recurrent neural network (RNN) architecture that extends the traditional Gated Recurrent Unit (GRU) (Cho et al., 2014) by processing input sequences in both forward and backward directions simultaneously. Using average pooling, the obtained paragraph embeddings are aggregated in order to produce an embedding that represents the whole sequence. The aggregated embedding from the case encoder is then passed to a fully connected layer with sigmoid activation to obtain the final predictions.

- **Case Encoder based on Transformer (Vaswani et al., 2017):**

  Due to transformers' advantage over recurrent networks in capturing long-distance relationships between words in a sequence, we have experimented with replacing the

BiGRU recurrent layer with a single transformer building block that contains a self-attention layer. The attention mechanism is incorporated to improve the encoding of the case embedding. The basic concept of the attention mechanism is to reduce the weight of unimportant features while increasing the weight of important features. The produced contextualized paragraph embeddings from the self-attention layer are narrowed down by selecting the final hidden state h of the first embedding as the representation of the whole sequence. In addition to this pooling approach, we experimented with more two pooling techniques namely: Avg Pooling and Max Pooling. Avg Pooling aggregates by taking the element-wise arithmetic mean of the paragraph-level embeddings, while Max Pooling takes the max value instead. In a similar manner to the first encoder, the aggregated embedding from the case encoder is passed to a fully connected layer with sigmoid activation in order to obtain the final prediction. Figure 5.3 shows the Case Encoder component with different approaches of encoding.

5.2.3 Ensemble Methods

In general, Ensemble Learning is a technique of combining two or more algorithms of similar or dissimilar types called base learners (Montgomery et al., 2012). This is done to make a more robust system which incorporates the predictions from all the base learners. In the context of an appeal courtroom, ensembling can be similar to a panel of judges deliberating on a case. Judges in an appeal courtroom have a variety of legal perspectives that contribute to a comprehensive evaluation of the case. Each judge brings their unique perspective and interpretation of the law to the table. By combining these diverse viewpoints, the goal is to create a more robust and well-rounded decision-making process. In real-world scenarios of appellate courts, appeal case decisions are usually reached by a 3-0 or 2-1 vote of a three-judge panel as shown in Figure 5.4. Thus, such phenomena inspired us to utilize an ensemble approach to model the voting process between judges. In this study, two ensemble techniques have been explored:

1. **Hard Majority Voting:** It is the most commonly used technique for ensemble

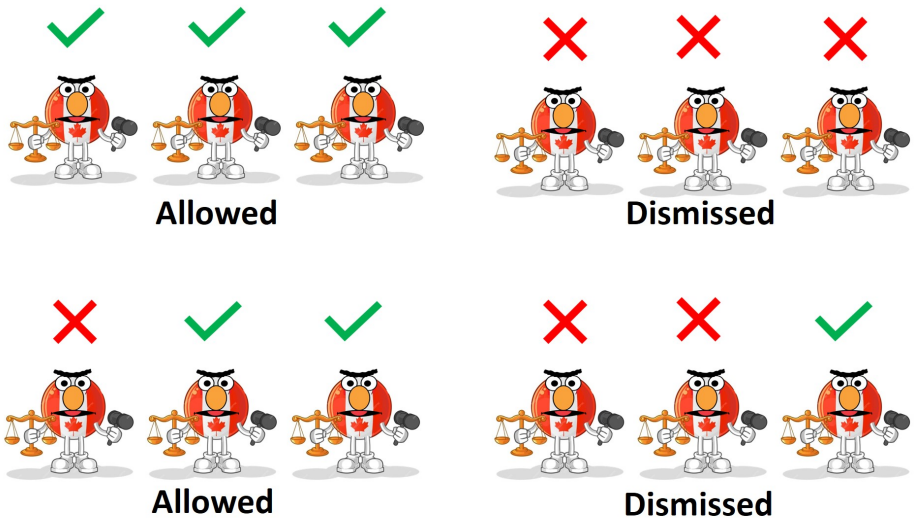


Figure 5.4: The possible outcomes from a Three-Judge-Panel system.

learning. In majority voting, the final output of the ensemble model is determined by counting for each class the number of votes of multiple models. The class with the majority of votes is predicted. For example, assuming we combine three classifiers, C1, C2, and C3, that assign the following classifications to a training sample: [0,0,1]. According to the hard voting algorithm, the class that receives the most votes is the final prediction. So, after passing the above data to a hard voting mechanism, it will categorize the sample as "class 0".

2. **Soft Majority Voting:** The second technique is based on calculating the sum of raw outputs of each model in the ensemble. It involves collecting predicted probabilities for each class label and predicting the class label with the largest probability. For example, assuming we combine three classifiers, C1, C2, and C3, that assign the following probabilities for "0" class: (0.56, 0.49, 0.6), and (0.44, 0.51, 0.4) for "1" class. Since the total probability for "0" (1.65) is greater than the total probability for "1" (1.35), the soft voting ensemble would predict the output class as "0"

The diagram in Figure 5.5 illustrates the proposed ensemble model. Firstly, a raw text of a document D_i is fed to three Hierarchical CanAL models that work as described in the previous section. Then, each model outputs probabilities for each class. Finally, the final

output is obtained by calculating the sum of the probabilities from the same class, then for each class, the argmax operation is applied to find the class with a higher probability value:

$$y_i = argmax \sum_{j=1}^{3} p_{kj} \tag{5.4}$$

where p_{kj} is the probability of the k^{th} class label of the j^{th} classifier and y_i is the predicted output for the document D_i.

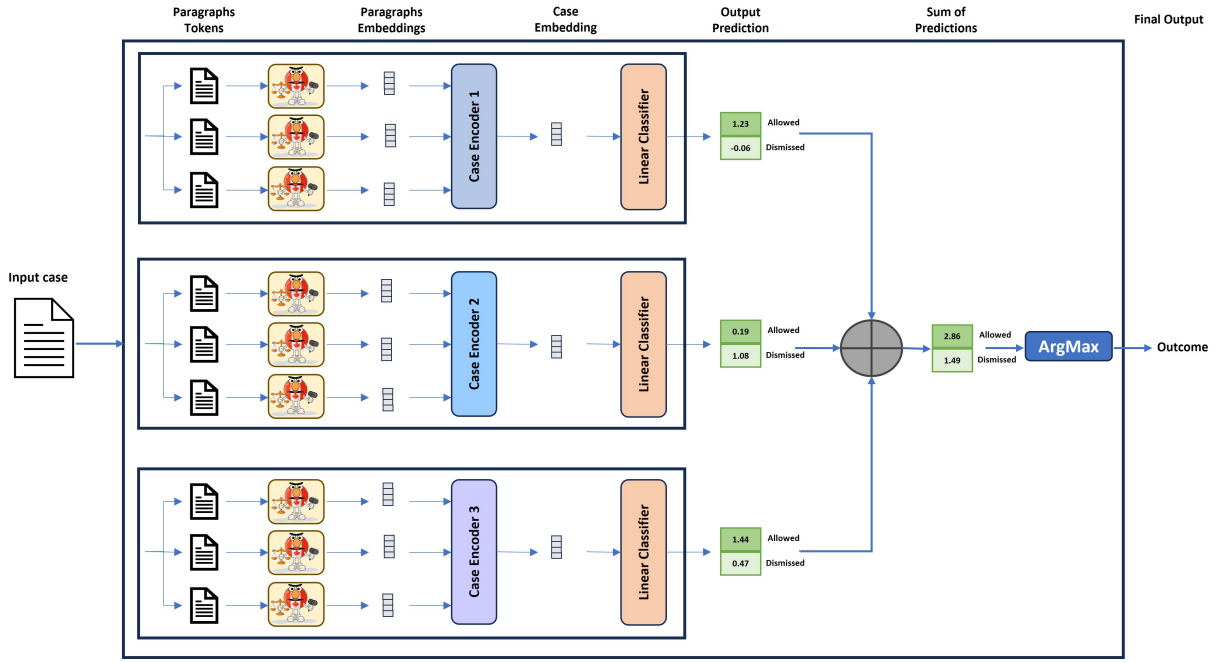


Figure 5.5: The Architecture of the Proposed Ensemble Model (EH-CanAL).

In all three models, the Case Encoder component is the key difference. In other words, it represents how each model analyzes the case and its internal components. As with the process of deciding an appeal decision between court judges, each model encodes the case in a different manner. An important point that must be taken into consideration, is to find an efficient method to combine all legal paragraph embeddings into a final case embedding while also generating attention scores over paragraphs which would make the model more interpretable as will be described in the next chapter. Taking that into consideration, we chose to use transformer-based Case Encoders with three types of pooling mechanisms for each component of the ensemble model.

5.3 Experimental Setup

Dataset and Training Details. We evaluated the models on binary text classification task using CanAL-DS described previously in section 3. To fine-tune our models for downstream task, suggestions made by Devlin et al. (2019) are followed. The number of training epochs is 4 with EarlyStopping to prevent the model from overfitting. The batch size was set to 16 for all tasks since this was the largest batch size that fit into the GPU memory. For all our models, We choose Binary Cross Entropy Loss function, Adam as the optimizer and a learning rate of 2e-5 and a dropout probability of 0.1. Based on the average number of paragraphs in the input datasets, the maximum number of paragraphs processed by the hierarchical models was set to 25 paragraphs. All the experiments were run on Tesla V100-PCIE-32GB GPU.

Evaluation. We evaluated our models using three standard evaluation metrics in a similar fashion to research in the literature, i.e., Precision (P), Recall (R), and F1-score. This decision was motivated both by compliance with research standards and by comparability with existing literature.

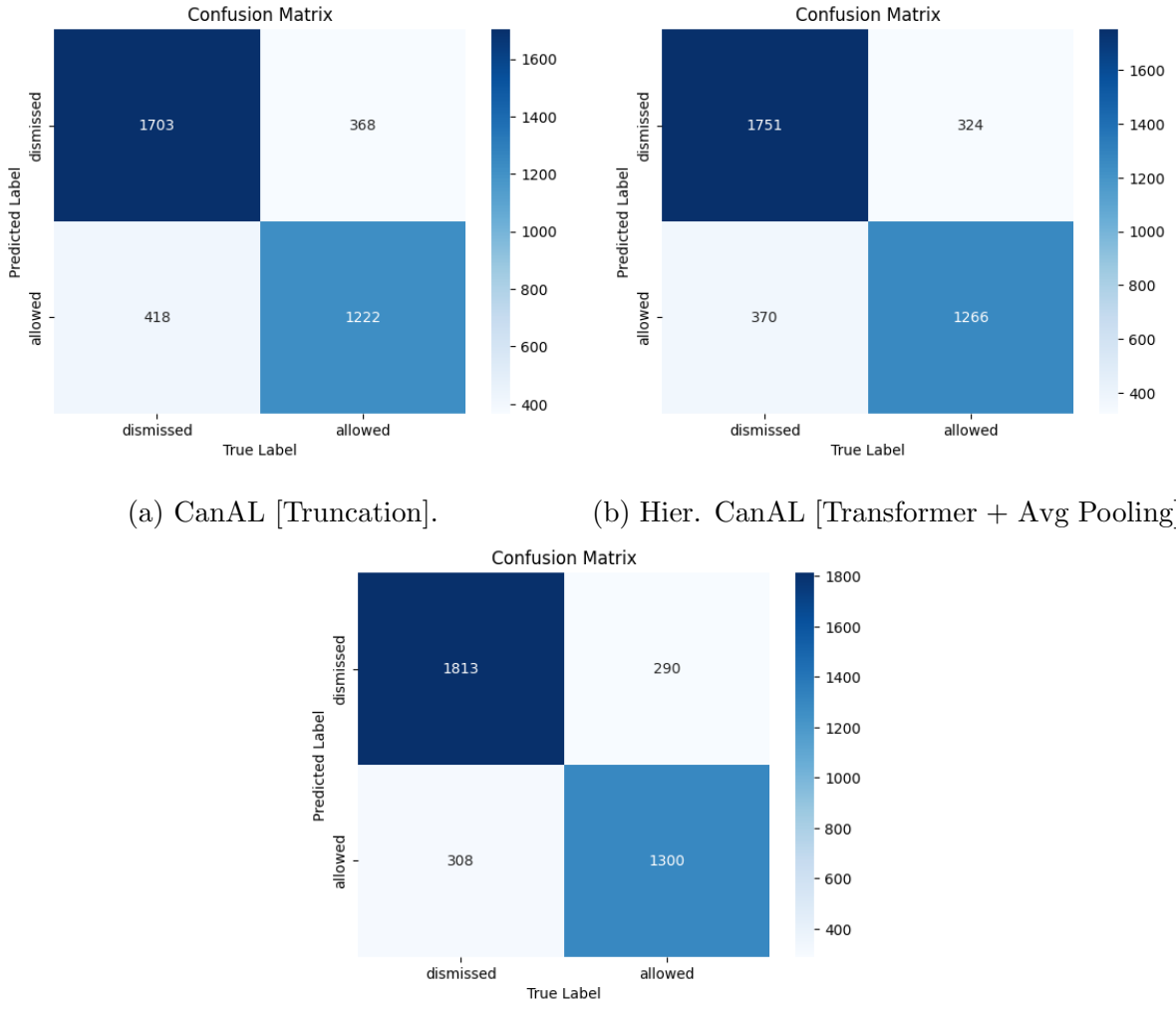
5.4 Results and Discussion

The results of all proposed models on our dataset are shown on Table 5.1. The confusion matrices for the best performers in each group of experiments are shown in Fig. 5.6. We can notice the poor performance of the generic BERT model on the legal-specific tasks even with more advanced architectures. On the other hand, the performance of legal models varies from one model to another. Moreover, the performance of all models is enhanced by the hierarchical architecture, with most of them achieving high scores when compared to baseline models. The only exception was the hierarchical CanAL with BiGRU case encoder, which performed less than the baseline CanAL. for that reason, we decided to use transformers as case encoders as they have a competitive performance among others. In particular, the hierarchical CanAL model using a case encoder as a transformer layer with average pooling achieves a relative improvement with a macro F1-score of 78.48 compared

Table 5.1: Comparison of different measures of performance on the CanAL Dataset.

	Validation Data			Test Data		
	P	**R**	**F1**	**P**	**R**	**F1**
Truncation						
Bert	82.01	66.32	65.27	50.17	50.02	39.34
LegalBERT	79.46	72.96	75.13	68.843	57.660	62.76
CanAL	80.17	73.79	76.84	76.88	74.50	74.95
Hier. [BiGRU Case Encoder]						
H-BERT	80.04	63.14	60.25	52.58	52.26	48.84
H-LegalBERT	82.50	77.08	75.00	82.33	67.35	66.08
H-CanAL	78.06	72.75	73.14	74.42	70.35	71.82
Hier. [Transformer Case Encode]						
H-BERT	77.50	69.86	69.79	60.06	45.23	51.60
H-LegalBERT	77.01	72.41	72.79	72.64	70.20	70.50
H-CanAL (CLS Pooling)	84.53	76.39	80.25	79.59	76.68	77.24
H-CanAL (Avg Pooling)	86.29	79.42	**81.94**	79.61	77.39	**78.48**
H-CanAL (Max Pooling)	80.08	79.31	79.68	83.44	75.07	76.60
Ensemble Hier.						
EH-CanAL (Hard Voting)	-	-	-	78.20	78.79	78.11
EH-CanAL (Soft Voting)	-	-	-	81.75	80.86	**81.79**

to the other used pooling techniques (Max and Cls). The hierarchical structure helps to some extent, but transformers inherently excel at capturing global dependencies due to their self-attention mechanism which can be crucial for the interpretability of the LJP task as will be described in the following chapter. Additionally, it is observed that models with ensemble settings outperform all individual models in terms of F1-score. Overall, the EH-CanAL with soft voting achieves the best performance with an F1-score of 81.79%. Another compelling observation is the ability of the ensemble learning to strike a good balance between various performance metrics. This shows the effectiveness of combining



(a) CanAL [Truncation].

(b) Hier. CanAL [Transformer + Avg Pooling].

(c) Ensemble Hier CanAL [Soft Voting].

Figure 5.6: The confusion matrices for the best performers in each group of experiments.

multiple models in tackling the complexities of a challenging task such as LJP. In addition, the success of soft voting over hard voting indicates that considering the confidence levels and unanimity among diverse models can lead to more reliable and accurate predictions.

5.5 Summary

In this section, we explored the legal judgment prediction (LJP) problem and took both the long text nature of legal documents and the voting settings of appeal judgments into

consideration. Following the real appeal-law process, we expanded our further pre-trained model and proposed a novel Ensemble Hierarchical Canadian Appeal-Law model. A series of experiments showed the effectiveness of our method over other baselines.

Chapter 6

Explainability Using Multi-Task Learning

6.1 Overview

One of the limitations of many previous studies in LJP is that neural judgement prediction models struggle to give a justification for their predictions. Because of this, we expand on our EH-CanAL model architecture to generate judgment rationales or explanations. Inspired by Chalkidis et al. (2021), Luo et al. (2017) and Niu et al. (2019), we introduce (MEH-CanAL), a Multi-task Ensemble Hierarchical CanAL model that is explicitly trained to pay attention only to relevant paragraphs. This allows the model to simultaneously predict the case outcome while also predicting which paragraphs of the case description are relevant for the final court decision.

6.2 Approach

In this section, we introduce the detailed architecture of our proposed Multi-Task Ensemble Hierarchical Canadian Appeal Law model (MEH-CanAL). First, the description of the main task and the auxiliary task is given. Following that, we explain how multitasking

works and how the attention mechanism contributes to the model's explainability. Finally, we explain how the final outcome and rationales are produced using an ensemble approach. Figure 6.1 illustrates the overall architecture of MEH-CanAL, while Figure 6.2 provides details on one component (MH-CanAL).

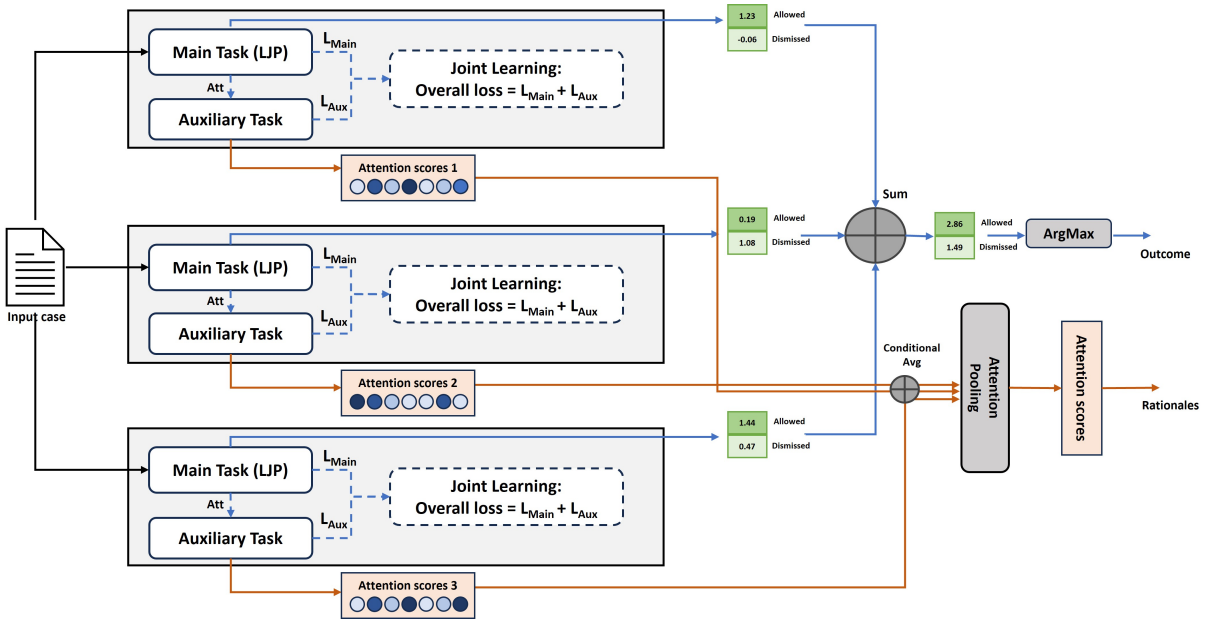


Figure 6.1: The MEH-CanAL Architecture that combines three MH-CanAL models.

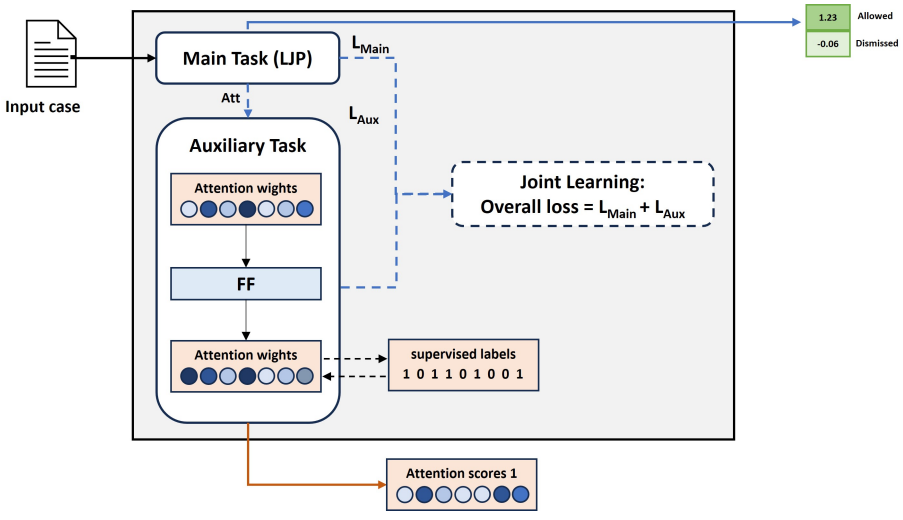


Figure 6.2: The MH-CanAL Model.

6.2.1 The Main Task

Similar to the EH-CanAL model architecture, MEH-CanAL's processes each case paragraph separately using a base CanAL LegalBERT to produce separate paragraph embeddings. For a given case document D_i, it can be viewed as a list of paragraphs $D_i = [P_1, \ldots, P_m]$. Each paragraph is a list of tokens $P_i = [w_1, \ldots, w_n]$. We first pass each paragraph through a CanAL encoder to extract context-unaware paragraph representations $P_i^{[CLS]}$, using the [CLS] embedding from CanAL encoder:

$$H_i = CanAL\_Encode(D_i) \tag{6.1}$$

where $H_i = [\ h_1^{P^{[CLS]}}, ..., h_m^{P^{[CLS]}}\ ]$.

These paragraph embeddings are then fed through the case encoder layer which is a shallow encoder with a single Transformer layer Vaswani et al. (2017) that contains a multi-head self-attention mechanism to compare paragraphs with each other. Also, a pooling layer to aggregate all paragraphs into one embedding that represents the input case. The final outputs of the transformer encoder are (1) an aggregate case embedding C_i which is a compact vector representation of all paragraphs in the case, and (2) a vector Att_i of the attention weights of the corresponding paragraphs.

$$C_i, Att_i = Pooling(Transformer(H_i)) \tag{6.2}$$

where $Att_i = [att_1^i, ..., att_m^i]$ is the paragraph attention scores vector that will be the input for the auxiliary task component, and C_i is the context-aware aggregated paragraphs embedding which is then fed through a linear classification layer with sigmoid activation function to obtain the final prediction probability p_i:

$$p_i = Sigmoid(Dense(Pooling(C_i))) \tag{6.3}$$

We use the binary cross-entropy loss function of the main task as:

$$L_{main} = -\frac{1}{k}\sum_{i=1}^{k}[y_i log p_i + (1 - y_i) log(1 - p_i)] \tag{6.4}$$

where k is the training size, y_i is the actual label and p_i is the probability predicted by the output neuron.

6.2.2 The Auxiliary Task

As mentioned above, the paragraph attention scores Att_i =[att_{i1}, ..., att_{ij} , ..., att_{im}] from the attention mechanism in the case encoder layer of the main task are linearly transformed in the auxiliary task network to produce binary relevance predictions for each paragraph. This not only allows the model to directly output a decision rationale as an auxiliary task but also provides a mechanism to explicitly teach the model which paragraphs to pay attention during training to via a multi-label paragraph classification auxiliary task.

Every element att_{ij} denotes the confidence that the paragraph in j the corresponding position is a rationale. Then, we get the attention supervised vector A_i =[a_{i1} , ..., a_{ij} , ..., a_{im}], where $a_{ij} \in \{0,1\}$. These labels may help our model to recognize paragraphs that are related to the rationales that are related to the final outcome. we define the training loss function of the auxiliary task as:

$$L_{aux} = -\frac{1}{k}\sum_{i=1}^{k}\sum_{j}^{m}[a_{ij} log att_{ij}] \tag{6.5}$$

where k is the number of the case samples, and m is the number of the labels.

6.2.3 Joint Learning of Tasks

MEH-CanAL model is trained in a multi-task setting with self-learning loss weights between the LJP binary classification task (weight σ_1 for loss L_1) and the auxiliary multi-label classification task (weight σ_2 for loss L_2). Prior approaches for building a multi-task model for LJP have used a weighted sum of these losses (Shang, 2022; Yue et al., 2021), where the loss weights are manually fine-tuned. We decided to let MEH-CanAL learns the weights between both losses on its own because, for 54.43% of training examples, judgment rationales (paragraph relevance labels) were unavailable, and varying between batches. Thus, learning the loss weights directly proves to be more stable during training according to (Dou et al., 2021). Additionally, to consider examples without judgment rationales, we created an indicator variable f, equal to one if there were paragraph labels, otherwise equal to zero. Our final loss function uses uncertainty loss (Kendall et al., 2017),

which uses homoscedastic uncertainty to combine multiple losses. Model uncertainty is a quantitative metric revealing the model's inability to make the prediction due to insufficient training data. Using this, we define our multi-task loss function for a single data sample i as:

$$L^i = f^i(\frac{L^i_{main}}{2\sigma^2_{main}} + \frac{L^i_{aux}}{2\sigma^2_{aux}} + log\sigma_{main}\sigma_{aux}) + (1 - f^i)L^i_{main} \tag{6.6}$$

where σ_{main}, and σ_{aux} are learnable parameters. If a task has a higher uncertainty, a larger sigma will decrease its contribution to the overall loss function. The log sigmas here will penalize setting the sigmas unnecessarily large.

6.2.4 The Ensemble Approach

The ensemble model follows a two-step process: first, each multi-task component provides predictions, and these are combined using soft majority voting, as explained previously in section 5.2.3. In addition, the rationales generated by the auxiliary task of each component are combined to form an overall justification for the model. This involves integrating rationales from sub-models that align with the majority vote results, ensuring a cohesive and representative justification for the ensemble model. This thoughtful approach aims to leverage the strengths of individual components while achieving consensus in both predictions and rationales, enhancing the model's overall performance and interpretability. We chose the option of averaging probabilities rather than summing them which provides a more balanced representation and can be less sensitive to extreme values. It reflects a consensus view, smoothing out variations between individual models.

$$Rationales_{Ens}(D_i) = \frac{1}{N_c}\sum_{j=1}^{N_c} Rationale_{model_j} \tag{6.7}$$

where:

- $Rationales_{Ens}(D_i)$ is the combined rationale score vector for paragraphs of document D_i according to the ensemble.
- $Rationale_{model_j}$ is the rationale score vector for predicted by model j.

- N_c is the number of models that align with the majority vote results.

6.2.5 Experimental Setup

For each case, the dataset contains paragraphs from the case description that were extracted using regular expressions. We used the pre-defined splits of our dataset which contains sets of 28,995, 3,222, and 3,711 cases between the training, validation, and test, respectively, For our model, we enrich this dataset with paragraph decision relevance labels, cited in the judges' reasons. This allows us to obtain decision rationales for 45.57%, 46.80% and 68.39% of training, validation, and test sets, respectively. The complete description of creating and preprocessing this dataset was detailed in section 3.2.4. The main task hyper-parameters are the same as the hierarchical models described before. All the experiments were run on Tesla V100-PCIE-32GB GPU.

6.3 Results and Discussion

The main classification results obtained can be seen in Table 6.1. As a result of the supervised learning of attention scores using the MEH-CanAL model, we were able to further improve our state-of-the-art results (81.79% F1 score), to achieve an 81.99% score. Even though this improvement is very slight, we are at least able to provide our model's explanations for its predictions, which is very important in the legal field. We can see that while the results of the auxiliary task still require further improvement, the results of the binary classification main task using hierarchical multi-task models have relatively improved when compared to the corresponding hierarchical models, which validates our decision to train CanAL in a multi-task setting.

By building MEH-CanAL model as a hierarchical model which first embeds paragraph meaning and then uses a multi-head attention layer to produce a final case embedding, we successfully process longer texts. Furthermore, we have enhanced MEH-CanAL to be an interpretable model by making legal rational selection an explicit part of the training procedure, as well as considering paragraph-level justifications rather than word-level following

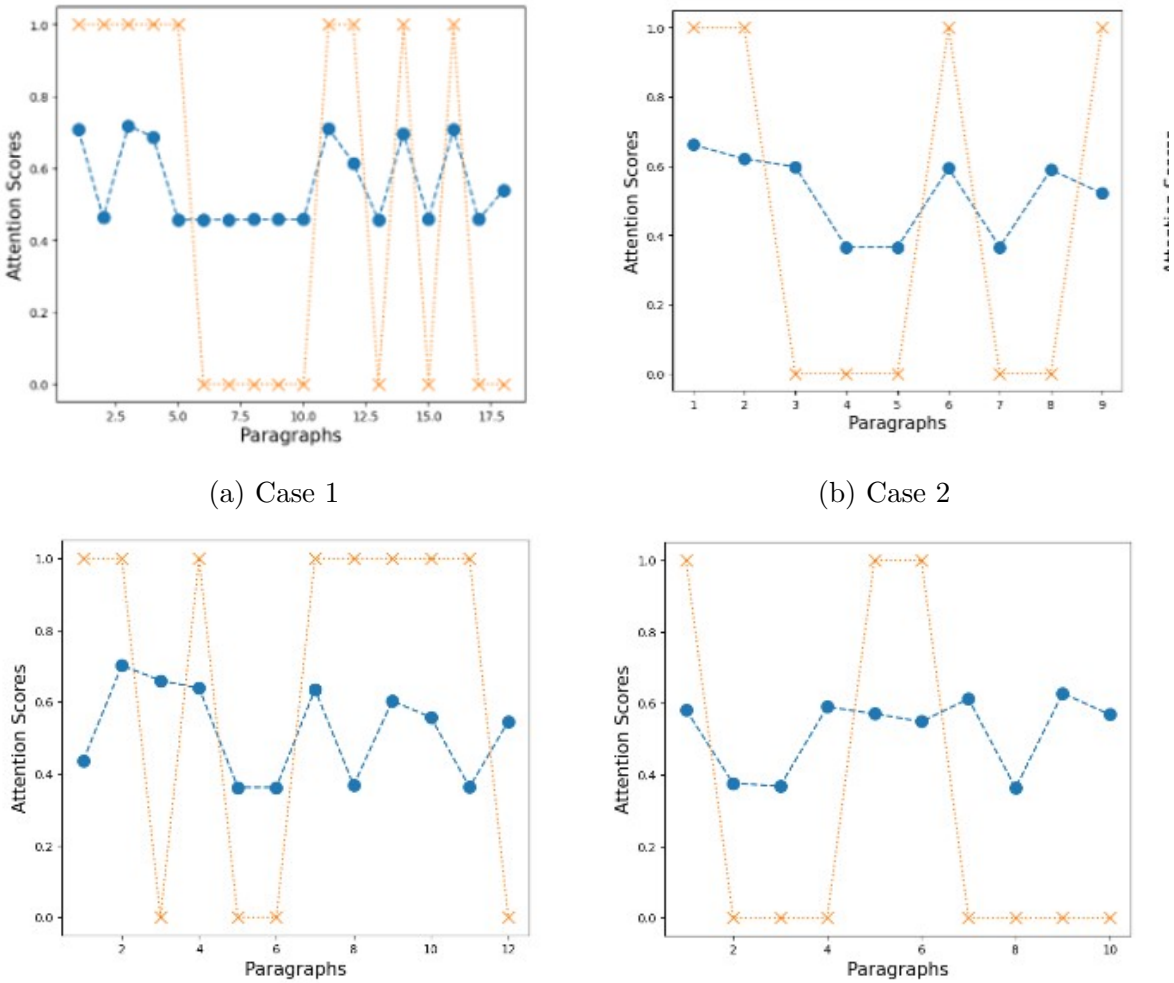
Table 6.1: Comparison of different measures of performance on the test split of the CanAL Dataset.

	Main Task					
Model	**P**	**R**	**F1**			
CanAL [Truncation]	76.88	74.50	74.95			
H-CanAL [Hier, Trans., MAX]	83.44	75.07	76.60			
H-CanAL [Hier, Trans., CLS]	79.59	76.68	77.24			
H-CanAL [Hier, Trans., Avg]	79.61	77.39	78.48			
EH-CanAL [Soft Voting]	81.75	80.86	81.79			
	Main Task			**Auxiliary Task**		
Model	**P**	**R**	**F1**	**P**	**R**	**MF1**
MH-CanAL [MAX]	83.11	75.55	76.74	61.92	50.39	36.49
MH-CanAL [CLS]	77.61	80.39	78.72	86.65	51.22	**42.30**
MH-CanAL [AVG]	84.56	75.61	79.83	78.58	50.45	36.37
MEH-CanAL	84.09	78.80	**81.99**	48.63	49.85	38.37

legal practices. Finally, we construct the model in an ensemble setting in an attempt to simulate the actual situation of higher court judgments.

To understand the detailed effect of attention mechanisms on the explainability of the LJP task, we first examined the attention weights (scores) and then visualized them over specific input case paragraph sequences from the test dataset. Attention score plots for some of the documents are shown in Figures 6.3 and 6.4. Figure 6.3 represents predicted paragraph attention scores with respect to the provided paragraphs' true labels, while Figure 6.4 represents the predicted paragraph attention scores with no true labels provided.

We can notice that the attention weights are biased toward the paragraphs around or between the true justifications as depicted in Figure 6.3. Although this bias influences task evaluation metrics, it aligns with real-world scenarios where arguments and justifications typically center around or between true justifications. This natural alignment suggests that the model's emphasis on such contextual elements may reflect a realistic understanding of



(a) Case 1

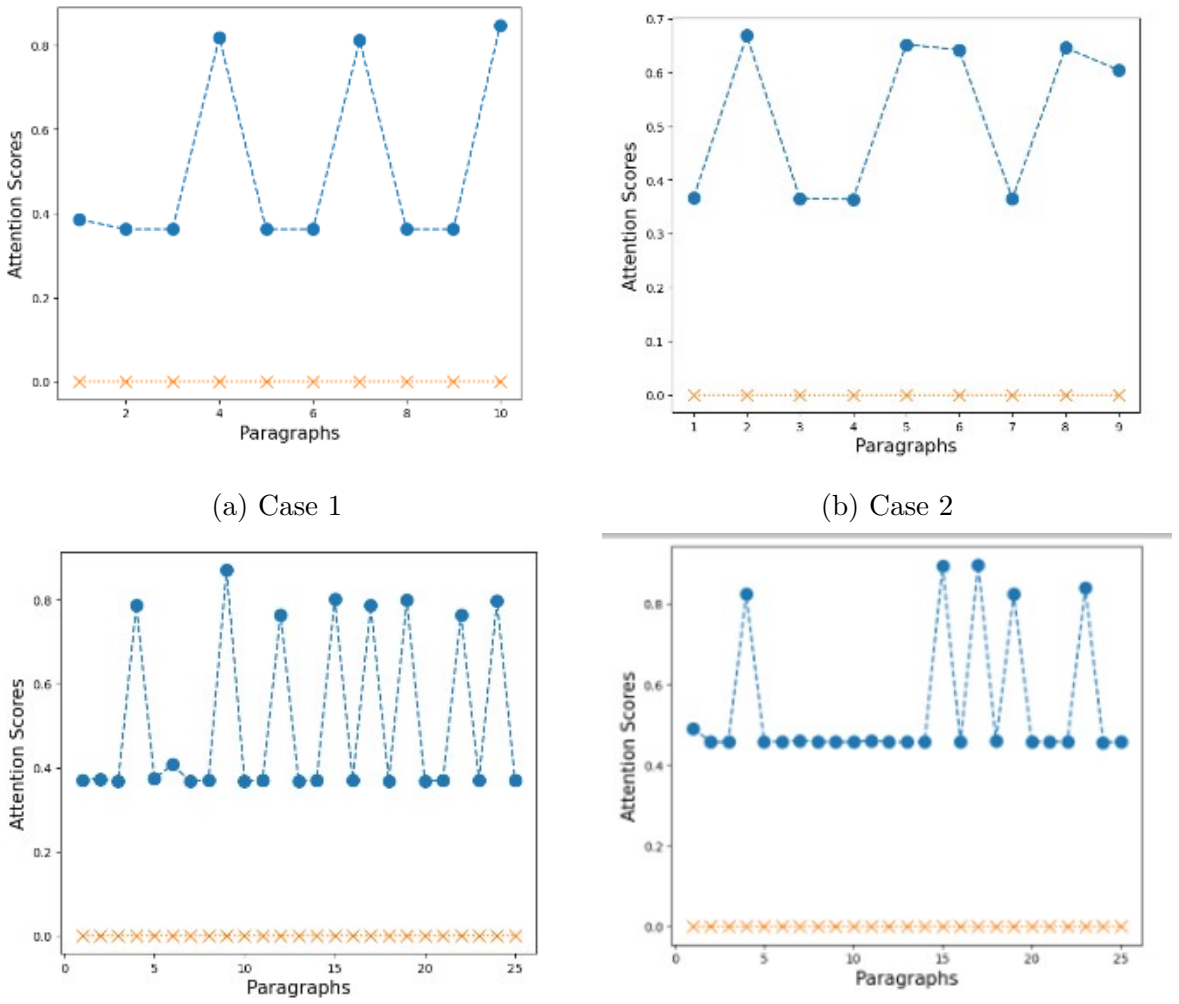
(b) Case 2

(c) Case 3

(d) Case 4

Figure 6.3: Attention scores of some cases, where the orange points are the supervised paragraphs labels and the blue circles are the predicted ones.

how information is presented and reasoned in the real world. Rather than solely considering it a drawback, recognizing and leveraging this alignment could contribute to the model's effectiveness in capturing the nuances of genuine justifications in diverse contexts. Another interesting point is that the model usually gives attention to the last paragraphs of the case where the most relevant syntactic and semantic information lies towards the end of the case. This behaviour suggests that the model is effectively recognizing the significance of the last sections in legal cases, where key insights or explanations are often provided. However, it is understandable that the most challenging task for the model during training



(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 6.4: Attention scores of some cases, where no supervised paragraph labels are provided.

was to predict the rationales for those cases with no labels provided as in Figure 6.4. To understand the model's behaviour in both situations, we visualized its attention scores for some representative instances and highlighted their corresponding paragraphs.

The first instance is case 1 in the sub-figure (a) of Figure 6.3. This case is cited as "2000 SKCA 73" and it has "Allowed" as both the ground truth label and the predicted outcome.

The highlighted paragraphs on the left sides of Figures 6.5, 6.6, and 6.7 represent the

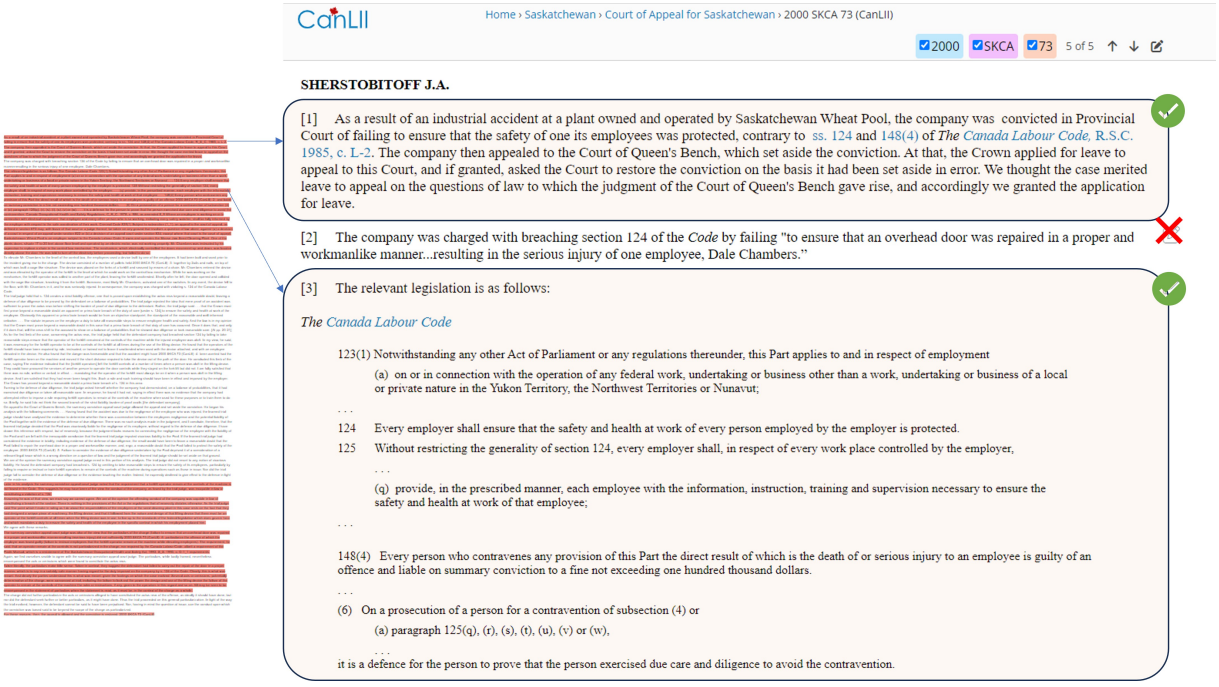


Figure 6.5: An example of visual explanation for case "2000 SKCA 73" (part 1).

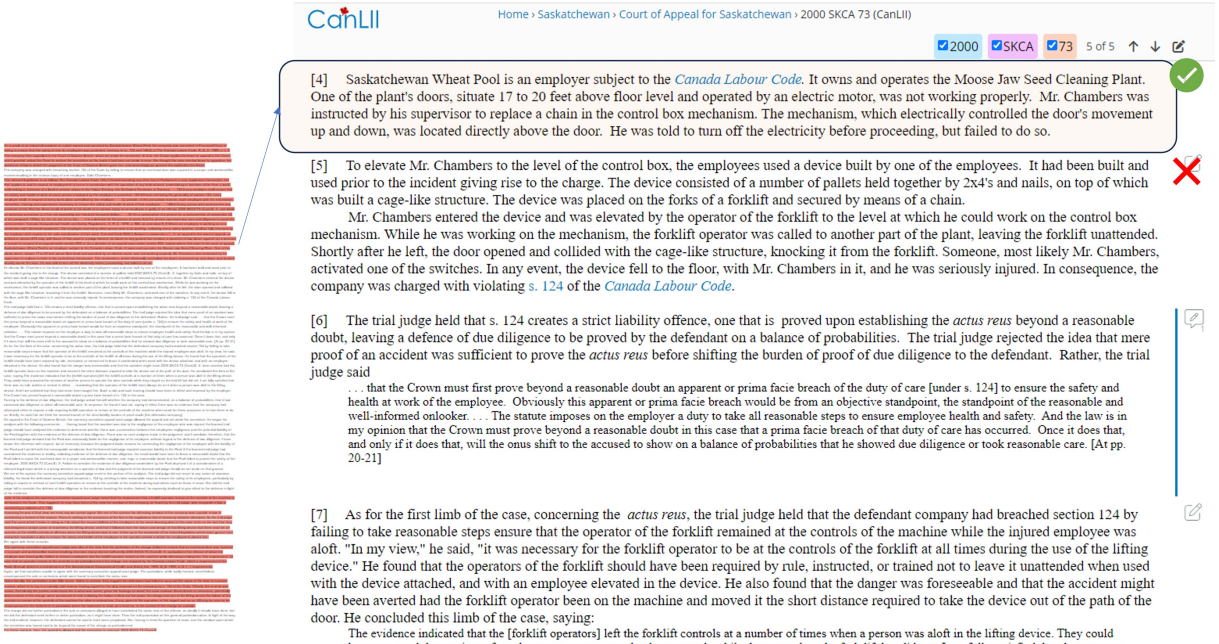


Figure 6.6: An example of visual explanation for case "2000 SKCA 73" (part 2).

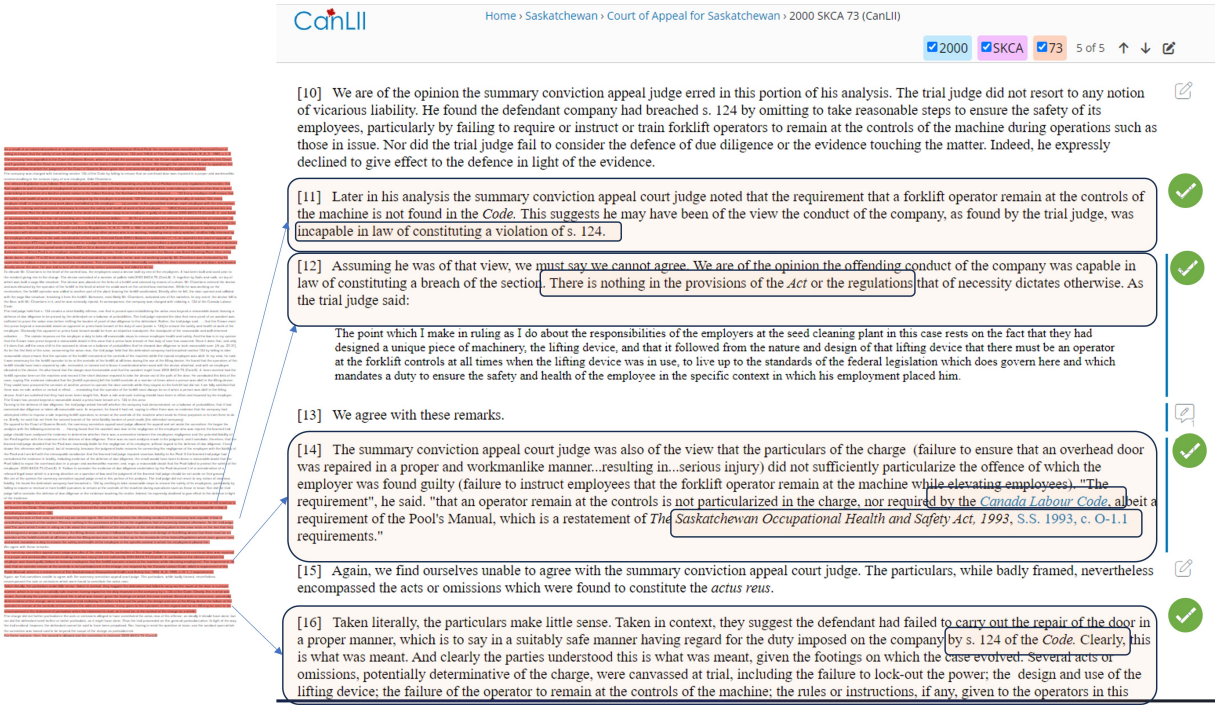


Figure 6.7: An example of visual explanation for case "2000 SKCA 73" (part 3).

predicted rationales from the model. Accordingly, we mapped these predictions onto the text of the same case on the CANLii website. We can see that the model picked up the majority of the true relevant paragraphs (with the green right check) but missed a few of them (with the red x). From these figures, it is clear that the model was able to recognize the important paragraphs that either include information related to the law of this case as in paragraphs number [3] and [4] in Figure 6.5 and Figure 6.6, or describe the argumentation that led to the final outcome (such as [12], [14] and [16] in Figure 6.7)

The second instance is case 4 in the sub-figure (d) of Figure 6.4. This case is cited as "2001 SKCA 107" and has "Dismissed" as both the ground truth label and predicted outcome. As there are no labelled rationales for this case, we get help from a useful feature on the CANLii website (the blue line on the right of some paragraphs) which points out to the paragraphs that have been cited by other cases. Such paragraphs can be considered rationales because they are used as arguments in other cases. Based on this observation, we realize that MEH-CanAL does indeed effectively select relevant paragraphs as [13], [15], and [17] in Figures 6.9 and 6.10

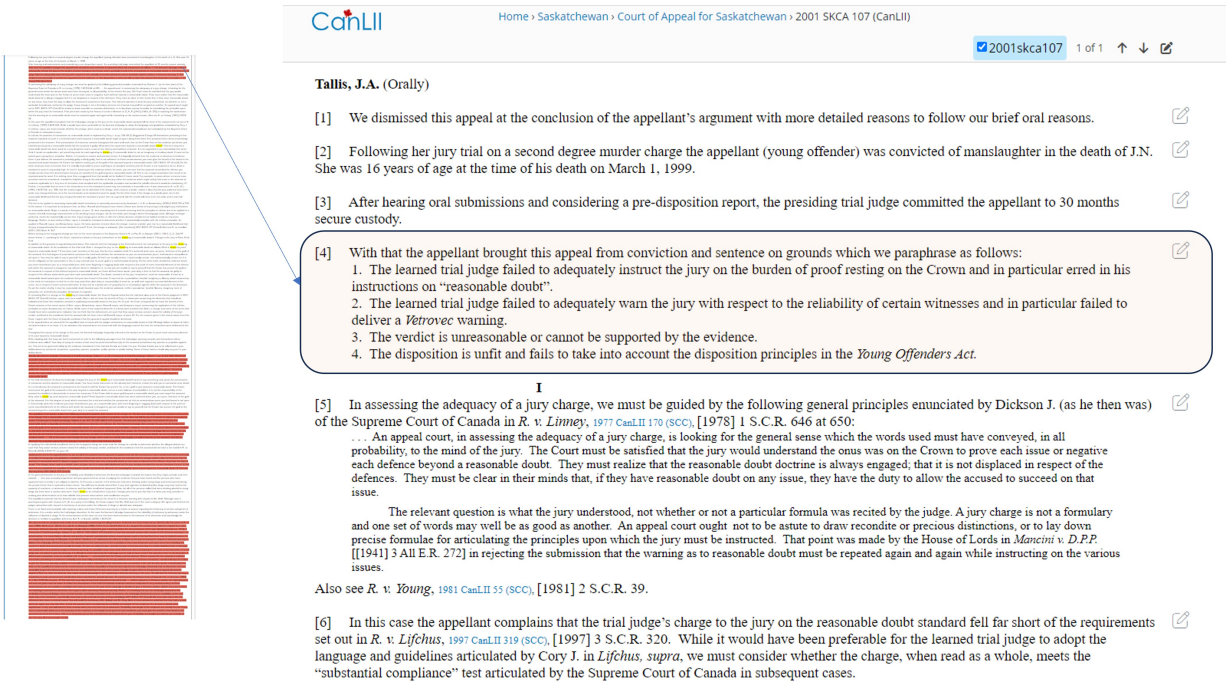


Figure 6.8: An example of visual explanation for case "2001 SKCA 107" (part 1).

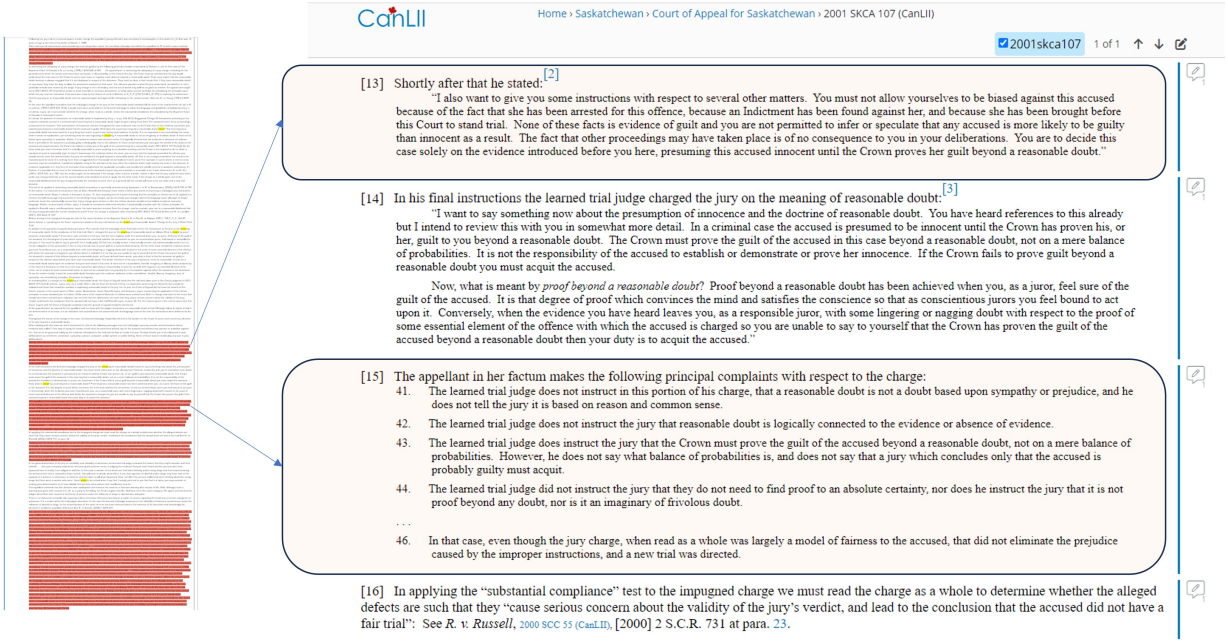


Figure 6.9: An example of visual explanation for case "2001 SKCA 107" (part 2).

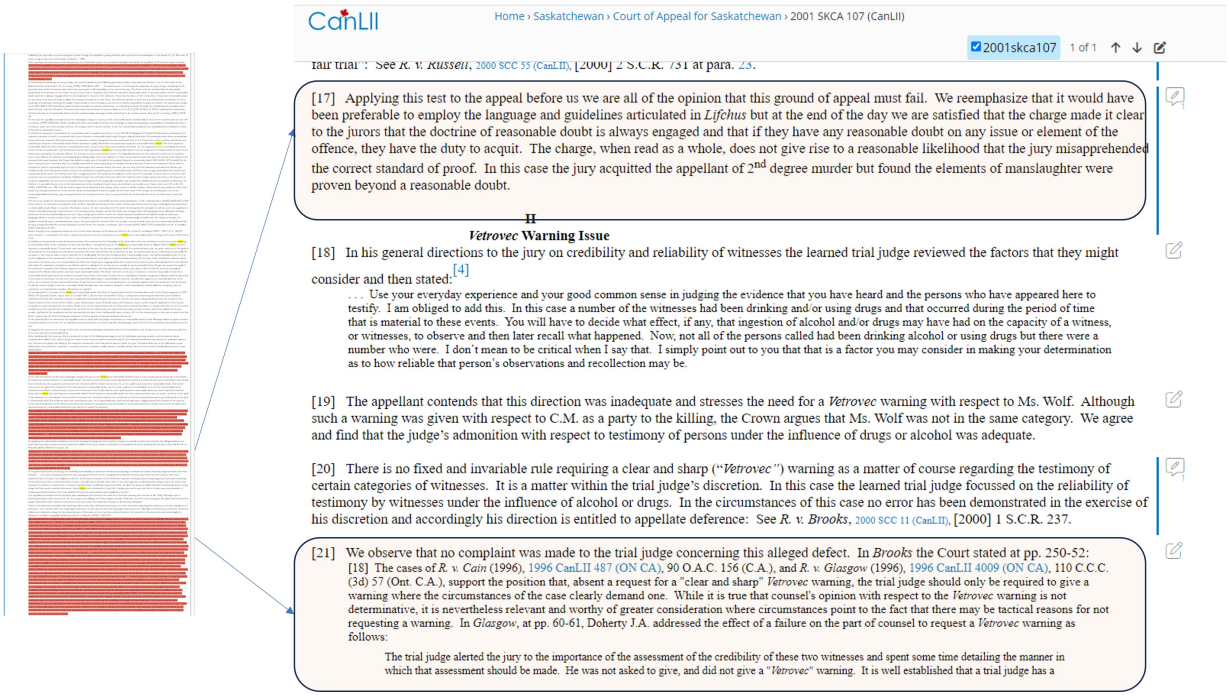


Figure 6.10: An example of visual explanation for case "2001 SKCA 107" (part 3).

Finally, by investigating the effect of paragraph attention learning on some cases that have incorrect predicted outcomes, we concluded that the model might be learned during training to pay attention to the paragraphs that include any type of citations including non-relevant ones such as the previous versions of the same case in the lower courts proceedings. These kinds of citations usually appear in the text not for the argumentation but rather to point out some facts, evidence or background information. This suggests to consider such citations during the pre-processing stage to avoid their impacts on the training and evaluation steps.

6.4 Summary

In this chapter, we discussed our approach to enhancing model explainability through multi-task learning. We introduced the main and auxiliary tasks, where the main task, focused on our H-CanAL model architecture, tackled the primary objective of Legal Judgment Prediction (LJP). Simultaneously, the auxiliary task addressed the challenge of explainability by training the model to generate judgment rationales or explanations. The ensemble approach, by combining the strengths of individual models, further enhances the overall performance and interpretability of the model. Through careful experimental setup and evaluation, we have observed promising results that underscore the effectiveness of our explainability-focused multi-task learning framework.

Chapter 7

Conclusion and Future Work

Legal case outcome prediction is a ML and NLP application in law that has not received attention in the context of the appeal courts in the legal system of Canada. In this thesis, we have systematically studied the problem of Legal Judgment Prediction for appeal court cases in Canada. We identified several research questions and attempted to explore how to answer them by taking the following steps. First, we constructed a dataset comprising decisions from various higher courts in Canada, and implemented an automatic labeling process to create an outcome variable for legal judgment prediction tasks. Subsequently, we enriched our dataset with supplementary annotations to facilitate auxiliary tasks, such as statute or case law prediction, and to enhance overall model explainability. Leveraging this dataset, we further pre-trained a Canadian-specific Bert-based legal model on a substantial corpus of around 50,000 documents encompassing Canadian case law and legislation. Then, we introduced a novel hierarchical network architecture with an ensemble setting that demonstrates significant promise in Legal Judgment Prediction (LJP) within the Canadian case law domain, marking a unique contribution to the field. To further enhance explainability, we incorporated a multi-task component with self-learning loss weights, aiding the model in identifying the relevant legal paragraphs that are crucial to the prediction process. The results of the binary classification task show that appeal court rulings in Canada can be predicted with good performance, and underscore the potential of country-specific language models in legal judgment prediction. However, the explain-

ability results are still quite low due to the complexity of this task. Given the limited time and computational resources available to us, we are confident we can further improve the results. Direct comparison with previous work is not possible because such systems were developed for other languages and very different legal systems. We hope that this research will contribute to establishing a general baseline for further studies of the Canadian appeal system. Our work could not only provide a baseline for further studies of the Canadian legal system but also a framework within which other legal systems can be analyzed and then compared with each other.

7.1 Concerns on Sensitive Data

Court case documents in the legal datasets may contain sensitive information such as personal details and party information. Leins et al. (2020) raise ethical concerns regarding the use of sensitive sources such as judgments in the development of datasets. In Canada, public access to judgment documents is allowed, and there are a lot of Canadian legal documents that have been released by many recently published works such as Henderson et al. (2022) and Chalkidis et al. (2023). Usually, publishing websites anonymize the documents before publishing a case in any database. Accordingly, sensitive information should not be contained in the databases or websites we used to collect data. However, concerns about accidental disclosure persist despite safety measures and anonymization by publishing entities. To balance data sharing risks and study reproducibility, we will share data only with researchers agreeing to strict terms of use. The same applies to releasing the CanAL language model at scale. We found that a more in-depth analysis should be conducted, which is kept as future work.

7.2 Limitations

Our study is based on the CanAL dataset, and our conclusions must be interpreted in light of the Canadian appellate law system and the dataset used. According to our analysis of

the results of the LJP, the variance in legal language in case documents from different Canadian courts affects the system's ability to predict outcomes based solely on these documents. In this regard, it is particularly challenging due to the differences between the various higher courts in Canada and their differing approaches to drafting texts of case documents that serve as a basis for predictive systems.

In addition, one of the technical difficulties associated with working with legal documents is their length. Hierarchical models based on paragraphs constrain the number of paragraphs that can be considered in training. Model performance is affected by this limitation in various types of tasks, and more exploratory work is necessary to address this issue. It was also computationally and time-consuming to conduct multiple experiments. Because we test our models on both a validation dataset and a holdout dataset, we decided to stick with one run for now and increase the number of runs in the future.

Moreover, it should be noted that our models have been designed based on pre-trained language models, and as such they inherit any biases they may contain. This requires examining all developed systems on groups of cases to determine their performance and to correct any inconsistencies before deploying them for prediction and inference purposes. In our view, further research on effective LJP of legal NLP systems is necessary to ensure these systems are robust and reliable. However, the time-consuming nature of developing and evaluating such models collaboratively with legal experts involves significant resource challenges.

Finally, it is important to note that the evolution of research methods in the field of legal judgment prediction has been characterized by a considerable emphasis on ML, particularly within the framework of DL. The pursuit of explainability in predictive models, however, requires a broader approach. Recent advances in AI methods have highlighted the importance of integrating logical, probabilistic, and neural representations into a "neurosymbolic" spectrum. There is a need to move beyond singular methodologies in order to ensure that legal reasoning is transparent and understandable. In this regard, although DL has provided valuable insights, it is necessary to synthesize various AI approaches in order to gain a comprehensive understanding of legal judgment prediction. It is imperative

that both legal and computational researchers navigate this evolving landscape, integrating insights from the neurosymbolic spectrum to enhance both predictive accuracy and understandability of legal reasoning.

7.3 Future Work

In addition to addressing the limitations outlined above, there are several potential avenues for future work in the field of LJP. Our results provide valuable insights into the effectiveness of further pre-training language models that already included legal data for capturing legal information in a specific legal system. A possible improvement in the future would be training a Canadian case law from scratch on the expanded corpus from all appeal courts in Canada. Training from scratch allows the model to develop specialized legal embeddings that are fine-tuned to Canadian case law. This can enhance the model's ability to capture and understand legal information specific to the Canadian legal landscape. Also, training from scratch minimizes the potential bias introduced during pre-training on a non-Canadian legal model. The model can adapt specifically to the characteristics of Canadian legal texts without being influenced by broader language patterns.

Moreover, an expanded corpus from all appeal courts in Canada could provide a more comprehensive and representative dataset. This inclusivity would ensure that the model is exposed to a diverse range of legal texts, covering various jurisdictions and legal contexts within the Canadian legal system. Another direction that might be worthwhile to consider and weigh in more detail is to investigate the impact of separating the dataset by province or based on timeframes of the case decisions to show how the distribution of decisions may affect model performance across tasks.

Furthermore, even though neural judgment prediction in English is a relatively new area, building a comprehensive multilingual legal corpus specifically tailored to the Canadian case law (English and French) is another possible future direction. By including both English and French legal texts, the corpus could facilitate the development of models that can effectively represent and predict legal judgments in a bilingual context. This is essential

for applications that involve understanding legal content in either official language.

In addition, LJP, as an NLP task, holds immense potential for leveraging large language models that exist today. These models, with their advanced natural language understanding capabilities, can be instrumental in automating and enhancing the legal decision-making process. By analyzing vast amounts of legal texts, case law, and statutes, these models can learn to predict judicial outcomes with remarkable accuracy. However, future directions of this task extend beyond mere prediction. Efforts can be directed towards refining these models to provide not only accurate judgments, but also transparent and interpretable explanations for their decisions. On a broader level, advances in other domain-specific tasks could enable us to tangibly advance the mission of facilitating societal problems through technology by creating systems for tasks going beyond neural legal judgment prediction. For example, we can investigate in the future the ability to generate or draft legal documents using new emerging techniques such as prompt learning and with state-of-the-art language models such as ChatGPT.

Finally, it is important to note that explanations in automated LJP systems serve as a valuable tool for debugging, as well as building user trust. Even though most current research focuses on explanation quality, it is important to acknowledge their role in identifying wrong outcomes and correcting them. While this dissertation may not delve into the complexities of utilizing explanations for debugging, it recognizes it as a significant area for future exploration. This acknowledgment opens avenues for future research to leverage explanations for error detection and resolution.

References

Agrawal, S., Ash, E., Chen, D., Gill, S. S., Singh, A., and Venkatesan, K. (2017). Affirm or reverse? using machine learning to help judges write opinions. *NBER Working Paper, National Bureau of Economic Research, Cambridge, MA, June*, 29.

Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., and Lampos, V. (2016). Predicting judicial decisions of the european court of human rights: A natural language processing perspective. *PeerJ Computer Science*, 2:e93.

Alexander, C., Al Jadda, K., Feizollahi, M. J., and Tucker, A. M. (2018). Using text analytics to predict litigation outcomes. *Law as Data: Computation, Text, and the Future of Legal Analysis (under contract, Santa Fe Institute Press, Michael Livermore & Daniel Rockmore, eds., 2018, Forthcoming)., Georgia State University College of Law, Legal Studies Research Paper*, (2018-13).

Alexander, C. and Iannarone, N. G. (2020). Winning, defined? text-mining arbitration decisions. *Cardozo Law Review, Forthcoming*.

Antoun, W., Baly, F., and Hajj, H. (2020). Arabert: Transformer-based model for arabic language understanding. *arXiv preprint arXiv:2003.00104*.

Ashley, K. D. (1992). Case-based reasoning and its implications for legal expert systems. *Artificial Intelligence and Law*, 1(2-3):113–208.

Ashley, K. D. (2019). A brief history of the changing roles of case prediction in ai and law. *Law Context: A Socio-Legal J.*, 36:93.

Ashley, K. D. and Brüninghaus, S. (2009). Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law*, 17(2):125–165.

Bala, J., Kellar, M., and Ramberg, F. (2017). Predictive analytics for litigation case management. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 3826–3830. IEEE.

Beltagy, I., Cohan, A., and Lo, K. (2019). Scibert: Pretrained contextualized embeddings for scientific text. *arXiv preprint arXiv:1903.10676*, 1(1.3):8.

Beltagy, I., Peters, M. E., and Cohan, A. (2020). Longformer: The long-document transformer. *arXiv:2004.05150*.

Branting, L. K., Yeh, A., Weiss, B., Merkhofer, E., and Brown, B. (2015). Inducing predictive models for decision support in administrative adjudication. In *AI Approaches to the Complexity of Legal Systems*, pages 465–477. Springer.

Branting, L. K., Yeh, A., Weiss, B., Merkhofer, E., and Brown, B. (2017). Cognitive assistance for administrative adjudication. In *2017 AAAI Fall Symposium Series*.

Buchanan, B. G. and Headrick, T. E. (1970). Some speculation about artificial intelligence and legal reasoning. *Stan. L. Rev.*, 23:40.

Buxbaum, R. M. and Latin, H. A. (1973). The computer and the law. *Computer*, 6(12):18–20.

Chalkidis, I., Androutsopoulos, I., and Aletras, N. (2019). Neural legal judgment prediction in english. *arXiv preprint arXiv:1906.02059*.

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*.

Chalkidis, I., Fergadiotis, M., Tsarapatsanis, D., Aletras, N., Androutsopoulos, I., and Malakasiotis, P. (2021). Paragraph-level rationale extraction through regularization: A case study on European court of human rights cases. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics:*

Human Language Technologies, pages 226–241, Online. Association for Computational Linguistics.

Chalkidis, I., Garneau, N., Goanta, C., Katz, D. M., and Søgaard, A. (2023). Lexfiles and legallama: Facilitating english multinational legal language model development. *arXiv preprint arXiv:2305.07507*.

Chehayeb, A., Al-Hussein, M., and Flynn, P. (2007). An integrated methodology for collecting, classifying, and analyzing canadian construction court cases. *Canadian Journal of Civil Engineering*, 34(2):177–188.

Chhatwal, R., Gronvall, P., Huber-Fliflet, N., Keeling, R., Zhang, J., and Zhao, H. (2018). Explainable text classification in legal document review a case study of explainable predictive coding. In *2018 IEEE international conference on big data (Big Data)*, pages 1905–1911. IEEE.

Chien, R., Maggs, P., and Stahl, F. (1971). New directions in legal information processing. In *AFIPS '72 (Spring)*.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Dahan, S., Touboul, J., Lam, J., and Sfedj, D. (2020). The unpredictable nature of termination notice: A data science experiment. *McGill Law Journal, Forthcoming*.

Dale, R. (2019). Law and word order: Nlp in legal tech. *Natural Language Engineering*, 25(1):211–217.

De Kruijf, G., de Vries, A., and Hasibi, F. (2022). Training a dutch (+ english) bert model applicable for the legal domain.

De Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., and Nissim, M. (2019). Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Dou, Q., Lu, Y., Manakul, P., Wu, X., and Gales, M. J. (2021). Attention forcing for machine translation. *arXiv preprint arXiv:2104.01264*.

Douka, S., Abdine, H., Vazirgiannis, M., El Hamdani, R., and Restrepo Amariles, D. (2021). JuriBERT: A masked-language model adaptation for French legal text. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 95–101, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Farzindar, A. (2004). Atefeh farzindar and guy lapalme,'letsum, an automatic legal text summarizing system in t. gordon (ed.), legal knowledge and information systems. jurix 2004: The seventeenth annual conference. amsterdam: Ios press, 2004, pp. 11-18. In *Legal Knowledge and Information Systems: JURIX 2004, the Seventeenth Annual Conference*, volume 120, page 11. IOS Press.

Farzindar, A. and Lapalme, G. (2004). Legal text summarization by exploration of the thematic structure and argumentative roles. In *Text Summarization Branches Out*, pages 27–34.

Gardner, A. v. d. L. (1984). *An artificial intelligence approach to legal reasoning*. PhD thesis, Stanford University.

Garneau, N., Gaumond, E., Lamontagne, L., and Déziel, P.-L. (2021). Criminelbart: A french canadian legal language model specialized in criminal law. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, pages 256–257.

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, ICAIL '23, page 472–480, New York, NY, USA. Association for Computing Machinery.

Government of Canada (2021a). The Appeal Process in Canada - About Canada's System of Justice.

Government of Canada (2021b). Where our legal system comes from.

Henderson, P., Krass, M., Zheng, L., Guha, N., Manning, C. D., Jurafsky, D., and Ho, D. (2022). Pile of law: Learning responsible data filtering from the law and a 256gb open-source legal dataset. *Advances in Neural Information Processing Systems*, 35:29217–29234.

Herrewijnen, E., Craandijk, D. F., et al. (2023). Towards meaningful paragraph embeddings for data-scarce domains: A case study in the legal domain. In *Proceedings of the 6th Workshop on Automated Semantic Analysis of Information in Legal Text co-located with the 19th International Conference on Artificial Intelligence and Law (ICAIL 2023), Braga, Portugal, 23rd September, 2023*, volume 3441, pages 13–18. CEUR-WS. org.

Horsuwan, T., Kanwatchara, K., Vateekul, P., and Kijsirikul, B. (2020). A comparative study of pretrained language models on thai social text categorization. In *Asian Conference on Intelligent Information and Database Systems*, pages 63–75. Springer.

Howe, J. S. T., Khang, L. H., and Chai, I. E. (2019). Legal area classification: A comparative study of text classifiers on singapore supreme court judgments. *arXiv preprint arXiv:1904.06470*.

Ikram, A. Y. and Chakir, L. (2019). Arabic text classification in the legal domain. In *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, pages 1–6. IEEE.

Jacob de Menezes-Neto, E. and Clementino, M. B. M. (2022). Using deep learning to predict outcomes of legal appeals better than human experts: A study with data from brazilian federal courts. *PloS One*, 17(7):e0272287.

Jiang, X., Ye, H., Luo, Z., Chao, W., and Ma, W. (2018). Interpretable rationale augmented charge prediction system. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*, pages 146–151.

Katz, D. M., Bommarito, M. J., and Blackman, J. (2017). A general approach for predicting the behavior of the supreme court of the united states. *PloS one*, 12(4):e0174698.

Katz, D. M., Bommarito II, M. J., and Blackman, J. (2014). Predicting the behavior of the supreme court of the united states: A general approach. *arXiv preprint arXiv:1407.6333*.

Kaufman, A., Kraft, P., and Sen, M. (2017). Machine learning, text data, and supreme court forecasting. *Project Report, Harvard University*.

Kaur, A. and Bozic, B. (2019). Convolutional neural network-based automatic prediction of judgments of the european court of human rights. In *AICS*, pages 458–469.

Kendall, A., Gal, Y., and Cipolla, R. (2017). Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491.

Kim, M.-Y., Rabelo, J., Goebel, R., Yoshioka, M., Kano, Y., and Satoh, K. (2022). Coliee 2022 summary: Methods for legal document retrieval and entailment. In *JSAI International Symposium on Artificial Intelligence*, pages 51–67. Springer.

Kort, F. (1957). Predicting supreme court decisions mathematically: A quantitative analysis of the "right to counsel" cases. *American Political Science Review*, 51(1):1–12.

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

Kumar, S., Reddy, P. K., Reddy, V. B., and Suri, M. (2013). Finding similar legal judgements under common law system. In *International workshop on databases in networked information systems*, pages 103–116. Springer.

Lage-Freitas, A., Allende-Cid, H., Santana, O., and de Oliveira-Lage, L. (2019). Predicting brazilian court decisions. *arXiv preprint arXiv:1905.10348*.

Lam, J. T., Liang, D., Dahan, S., and Zulkernine, F. H. (2020). The gap between deep learning and law: Predicting employment notice. In *NLLP@ KDD*, pages 52–56.

Lawlor, R. C. (1963). What computers can do: Analysis and prediction of judicial decisions. *American Bar Association Journal*, pages 337–344.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2020). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Leins, K., Lau, J. H., and Baldwin, T. (2020). Give me convenience and give her death: Who should decide what uses of nlp are appropriate, and on what basis? *arXiv preprint arXiv:2005.13213*.

Leszczyński, L. (2020). Implementing prior judicial decisions as precedents: The context of application and justification. *International Journal for the Semiotics of Law-Revue internationale de Sémiotique juridique*, 33(1):231–244.

Li, S., Zhang, H., Ye, L., Guo, X., and Fang, B. (2019). Mann: A multichannel attentive neural network for legal judgment prediction. *IEEE Access*, 7:151144–151155.

Limsopatham, N. (2021). Effectively leveraging BERT for legal document classification. In Aletras, N., Androutsopoulos, I., Barrett, L., Goanta, C., and Preotiuc-Pietro, D., editors, *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 210–216, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Liu, Y.-H., Chen, Y.-L., and Ho, W.-L. (2015). Predicting associated statutes for legal problems. *Information Processing & Management*, 51(1):194–211.

Liu, Z. and Chen, H. (2017). A predictive performance comparison of machine learning models for judicial cases. In *2017 IEEE Symposium series on computational intelligence (SSCI)*, pages 1–6. IEEE.

Long, S., Tu, C., Liu, Z., and Sun, M. (2019). Automatic judgment prediction via legal reading comprehension. In *China National Conference on Chinese Computational Linguistics*, pages 558–572. Springer.

Lou, A., Salaün, O., Westermann, H., and Kosseim, L. (2021). Extracting facts from case rulings through paragraph segmentation of judicial decisions. In *International Conference on Applications of Natural Language to Information Systems*, pages 187–198. Springer.

Luo, B., Feng, Y., Xu, J., Zhang, X., and Zhao, D. (2017). Learning to predict charges for criminal cases with legal basis. *arXiv preprint arXiv:1707.09168*.

Ma, Y., Shao, Y., Liu, B., Liu, Y., Zhang, M., and Ma, S. (2021). Retrieving legal cases from a large-scale candidate corpus. *Proceedings of the Eighth International Competition on Legal Information Extraction/Entailment, COLIEE2021*.

Malik, V., Sanjay, R., Nigam, S. K., Ghosh, K., Guha, S. K., Bhattacharya, A., and Modi, A. (2021). ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. In Zong, C., Xia, F., Li, W., and Navigli, R., editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4046–4062, Online. Association for Computational Linguistics.

Martin, A. D., Quinn, K. M., Ruger, T. W., and Kim, P. T. (2004). Competing approaches to predicting supreme court decision making. *Perspectives on Politics*, 2(4):761–767.

Mauritz, B. J. (2018). *Automatic classification of legal documents*. PhD thesis, Master's thesis, Masarykova univerzita.

Mckaay, E. and Robillard, P. (1974). Predicting judicial decisions: The nearest neighbor rule and visual representa-tion of case patterns.

Medvedeva, M., Üstün, A., Xu, X., Vols, M., and Wieling, M. (2021). Automatic judgement forecasting for pending applications of the european court of human rights. In *ASAIL/LegalAIIA@ ICAIL*, pages 12–23.

Medvedeva, M., Vols, M., and Wieling, M. (2018). Judicial decisions of the european court of human rights: Looking into the crystal ball. In *Proceedings of the Conference on Empirical Legal Studies*.

Medvedeva, M., Vols, M., and Wieling, M. (2020). Using machine learning to predict decisions of the european court of human rights. *Artificial Intelligence and Law*, 28(2):237–266.

Modi, A., Kalamkar, P., Karn, S., Tiwari, A., Joshi, A., Tanikella, S. K., Guha, S. K., Malhan, S., and Raghavan, V. (2023). Semeval 2023 task 6: Legaleval–understanding legal texts. *arXiv preprint arXiv:2304.09548*.

Montgomery, J. M., Hollenbach, F. M., and Ward, M. D. (2012). Improving predictions using ensemble bayesian model averaging. *Political Analysis*, 20(3):271–291.

Nagel, S. (1960). Using simple calculations to predict judicial decisions. *American Behavioral Scientist*, 4(4):24–28.

Nejadgholi, I., Bougueng, R., and Witherspoon, S. (2017). A semi-supervised training method for semantic search of legal facts in canadian immigration cases. In *JURIX*, pages 125–134.

Niu, J., Yang, Y., Zhang, S., Sun, Z., and Zhang, W. (2019). Multi-task character-level attentional networks for medical concept normalization. *Neural Processing Letters*, 49:1239–1256.

Octavia-Maria, Zampieri, M., Malmasi, S., Vela, M., P. Dinu, L., and van Genabith, J. (2017). Exploring the use of text classification in the legal domain. In *Proceedings of 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts (ASAIL)*, London, United Kingdom.

O'Sullivan, C. and Beel, J. (2019). Predicting the outcome of judicial decisions made by the european court of human rights. *arXiv preprint arXiv:1912.10819*.

Paul, S., Goyal, P., and Ghosh, S. (2020). Automatic charge identification from facts: A few sentence-level charge annotations is all you need. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1011–1022.

Paul, S., Goyal, P., and Ghosh, S. (2022). Lesicin: A heterogeneous graph-based approach for automatic legal statute identification from indian legal documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11139–11146.

Paul, S., Mandal, A., Goyal, P., and Ghosh, S. (2023). Pre-trained language models for the legal domain: a case study on indian law. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 187–196.

Queudot, M. and Meurs, M.-J. (2018). Artificial intelligence and predictive justice: Limitations and perspectives. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 889–897. Springer.

Rabelo, J., Goebel, R., Kim, M.-Y., Kano, Y., Yoshioka, M., and Satoh, K. (2022). Overview and discussion of the competition on legal information extraction/entailment (coliee) 2021. *The Review of Socionetwork Strategies*, 16(1):111–133.

Ruger, T., Kim, P., Martin, A., and Quinn, K. (2004). The supreme court forecasting project: Legal and political science approaches to predicting supreme court decision-making. *Columbia Law Review*, 104.

Salaün, O., Langlais, P., Lou, A., Westermann, H., and Benyekhlef, K. (2020). Analysis and multilabel classification of quebec court decisions in the domain of housing law. In *International Conference on Applications of Natural Language to Information Systems*, pages 135–143. Springer.

Schuster, M. and Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Shaikh, R. A., Sahu, T. P., and Anand, V. (2020). Predicting outcomes of legal cases based on legal factors using classifiers. *Procedia Computer Science*, 167:2393–2402.

Shang, X. (2022). A computational intelligence model for legal prediction and decision support. *Computational Intelligence and Neuroscience*, 2022.

Sharma, R. D., Mittal, S., Tripathi, S., and Acharya, S. (2015). Using modern neural networks to predict the decisions of supreme court of the united states with state-of-the-art accuracy. In *International Conference on Neural Information Processing*, pages 475–483. Springer.

Sim, Y., Routledge, B. R., and Smith, N. A. (2015). The utility of text: the case of amicus briefs and the supreme court. In *Twenty-Ninth AAAI conference on artificial intelligence*.

Sivaranjani, N., Jayabharathy, J., and Teja, P. (2021). Predicting the supreme court decision on appeal cases using hierarchical convolutional neural network. *International Journal of Speech Technology*, 24(3):643–650.

Smith, J., Gelbart, D., and Graham, D. (1992). Building expert systems in case-based law. *Expert Systems with Applications*, 4(4):335–342.

Song, L., Han, X., Liu, G., Wang, W., Cui, C., and Yin, Y. (2022). Sequential multi-task learning with task dependency for appeal judgment prediction. *arXiv preprint arXiv:2204.07046*.

Spaeth, H. J., Epstein, L., Martin, A. D., Segal, J. A., Ruger, T. J., and Benesh, S. C. (2020). Supreme court database, version 2020 release 01. *Database at http://supremecourtdatabase. org*.

Strickson, B. and De La Iglesia, B. (2020). Legal judgement prediction for uk courts. In *Proceedings of the 2020 The 3rd International Conference on Information Science and System*, pages 204–209.

Sulea, O.-M., Zampieri, M., Vela, M., and Van Genabith, J. (2017). Predicting the law area and decisions of french supreme court cases. *arXiv preprint arXiv:1708.01681*.

Sun, C., Qiu, X., Xu, Y., and Huang, X. (2019). How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Susskind, R. E. (1986). Expert systems in law: A jurisprudential approach to artificial intelligence and legal reasoning. *The modern law review*, 49(2):168–194.

Tiersma, P. M. (1999). *Legal language*. University of Chicago Press.

Trasberg, H. (2019). Quantitative legal prediction and the rule of law.

Ulmer, S. S. (1963). Quantitative analysis of judicial processes: Some practical and theoretical applications. *Law and Contemporary Problems*, 28(1):164–184.

Undavia, S., Meyers, A., and Ortega, J. E. (2018). A comparative study of classifying legal documents with neural networks. In *2018 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 515–522. IEEE.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vazirgiannis, M., Boniol, P., Panagopoulos, G., Xypolopoulos, C., El-Hamdani, R., and Amariles, D. R. (2020). Performance in the courtroom: Automated processing and visualization of appeal court decisions in france. *Revista Eletrônica da Faculdade de Direito de Franca*, 15(2).

Virtucio, M. B. L., Aborot, J. A., Abonita, J. K. C., Avinante, R. S., Copino, R. J. B., Neverida, M. P., Osiana, V. O., Peramo, E. C., Syjuco, J. G., and Tan, G. B. A. (2018). Predicting decisions of the philippine supreme court using natural language processing and machine learning. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, volume 2, pages 130–135. IEEE.

Waltl, B., Bonczek, G., Scepankova, E., Landthaler, J., and Matthes, F. (2017). Predicting the outcome of appeal decisions in germany's tax law. In *International Conference on Electronic Participation*, pages 89–99. Springer.

Westermann, H., Walker, V. R., Ashley, K. D., and Benyekhlef, K. (2019). Using factors to predict and analyze landlord-tenant decisions to increase access to justice. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law*, pages 133–142.

Xiao, C., Zhong, H., Guo, Z., Tu, C., Liu, Z., Sun, M., Feng, Y., Han, X., Hu, Z., Wang, H., et al. (2018). Cail2018: A large-scale legal dataset for judgment prediction. *arXiv preprint arXiv:1807.02478*.

Yan, G., Li, Y., Shen, S., Zhang, S., and Liu, J. (2019). Law article prediction based on deep learning. In *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 281–284. IEEE.

Yousfi-Monod, M., Farzindar, A., and Lapalme, G. (2010). Supervised machine learning for summarizing legal documents. In *Canadian Conference on Artificial Intelligence*, pages 51–62. Springer.

Yue, L., Liu, Q., Jin, B., Wu, H., Zhang, K., An, Y., Cheng, M., Yin, B., and Wu, D. (2021). Neurjudge: A circumstance-aware neural framework for legal judgment prediction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 973–982.

Zheng, L., Guha, N., Anderson, B. R., Henderson, P., and Ho, D. E. (2021). When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 159–168.

Zhong, H., Guo, Z., Tu, C., Xiao, C., Liu, Z., and Sun, M. (2018). Legal judgment prediction via topological learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3540–3549.

Zhong, H., Wang, Y., Tu, C., Zhang, T., Liu, Z., and Sun, M. (2020). Iteratively questioning and answering for interpretable legal judgment prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 1250–1257.

APPENDICES

Appendix A

Responsible NLP Checklist

We use the Responsible NLP Checklist [1] from the ACL 2023 conference to evaluate the ethical implications of our work. The checklist is displayed below with the answers to each question provided in italics.

Section A - For every submission

1. **Did you describe the limitations of your work?** *Yes, Section 7.2 of the thesis discusses limitations of this work.*

2. **Did you discuss any potential risks of your work?** *Yes, in section 1.4.*

3. **Do the abstract and introduction summarize the paper's main claims?** *Yes, the abstract and introduction both summarize the contributions of this work.*

Section B - Did you use or create scientific artifacts?

Yes, we used and created various scientific artifacts. We constructed a dataset, further pre-trained a legal language model, and created the source code used to train and evaluate

[1] https://aclrollingreview.org/responsibleNLPresearch/

the models and the explainability step. We also used existing artifacts, such as Python libraries, pre-trained models, and datasets.

1. **Did you cite the creators of artifacts you used?** *Yes, all artifacts are cited where they are mentioned in the text.*

2. **Did you discuss the license or terms for use and / or distribution of any artifacts?** *Yes, all data have been collected from the public domain and have been licensed for research purposes, and this has been discussed where the artifacts are mentioned in the text.*

3. **Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?** *Yes, we discussed that in Sections 1.4 and 7.1.*

4. **Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?** *We discussed this issue in Section 3.2.2.*

5. **Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?** *Yes, we provide this discussion in Chapters 3 and 4.*

6. **Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created?** *Yes, we provide this information in Chapter 3.*

Section C - Did you run computational experiments?

Yes, we ran computational experiments to train and evaluate the models and to conduct the explainability step. Chapters 4, 5, and 6 provide a detailed description of the experiments and the results.

1. **Did you report the number of parameters in the models used, the total computational budget, and the computing infrastructure used?** *Yes. Computing infrastructure is described in Section ??. and parameters are mentioned in Appendix B.*

2. **Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?** *Yes, in the experimental setup and training details sections in Chapters 4, 5, and 6.*

3. **Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?** *Yes.*

4. **If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?** *Yes, whenever we mention a package, we add a footnote to its original resource or URL.*

Section D - Did you use human annotators (e.g., crowd-workers) or research with human participants?

No, all annotation process was conducted by one person only.

1. **Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?** *Not applicable.*

2. **Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?** *Not applicable.*

3. **Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?** *Not applicable.*

4. **Was the data collection protocol approved (or determined exempt) by an ethics review board?** *Not applicable*

5. **Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?** *Not applicable.*

Appendix B

CanAL Models' Architectures

CanAL [BertForSequenceClassification]

```
BertForSequenceClassification(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
```

```
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
        (intermediate): BertIntermediate(
          (dense): Linear(in_features=768, out_features=3072, bias=True)
          (intermediate_act_fn): GELUActivation()
        )
        (output): BertOutput(
          (dense): Linear(in_features=3072, out_features=768, bias=True)
          (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (dropout): Dropout(p=0.1, inplace=False)
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (classifier): Linear(in_features=768, out_features=2, bias=True)
)
Total Number of Parameters = 109786368 ~ 110M
```

H-CanAL [Hierarchical]

```
HierarchicalModel(
  (base_model): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
```

```
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (pos_encoder): PositionalEncoder(
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (hier_model): TransformerEncoder(
    (layers): ModuleList(
      (0): TransformerEncoderLayer(
        (self_attn): MultiheadAttention(
          (out_proj): NonDynamicallyQuantizableLinear
          (in_features=768, out_features=768, bias=True)
        )
        (linear1): Linear(in_features=768, out_features=3072, bias=True)
        (dropout): Dropout(p=0.1, inplace=False)
        (linear2): Linear(in_features=3072, out_features=768, bias=True)
        (norm1): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        (norm2): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        (dropout1): Dropout(p=0.1, inplace=False)
        (dropout2): Dropout(p=0.1, inplace=False)
      )
    )
  )
  (self_attn): MultiheadAttention(
    (out_proj): NonDynamicallyQuantizableLinear
    (in_features=768, out_features=768, bias=True)
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (cls_head): Linear(in_features=768, out_features=1, bias=True)
  (loss_fnc): BCEWithLogitsLoss())
Total Number of Parameters = 118933253 ~ 119M
```

MH-CanAL [Multi-task Hierarchical]

```
MH_CanAL(
  (base_model): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
```

```
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (pos_encoder): PositionalEncoder(
    (dropout): Dropout(p=0.1, inplace=False)
  )
  (hier_model): TransformerEncoder_withAttention(
    (layers): ModuleList(
      (0): TransformerEncoderLayer_withAttention(
        (self_attn): MultiheadAttention(
          (out_proj): NonDynamicallyQuantizableLinear
          (in_features=768, out_features=768, bias=True)
        )
        (linear1): Linear(in_features=768, out_features=3072, bias=True)
        (dropout): Dropout(p=0.1, inplace=False)
        (linear2): Linear(in_features=3072, out_features=768, bias=True)
        (norm1): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        (norm2): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        (dropout1): Dropout(p=0.1, inplace=False)
        (dropout2): Dropout(p=0.1, inplace=False)
      )
    )
  )
  (self_attn): MultiheadAttention(
    (out_proj): NonDynamicallyQuantizableLinear
    (in_features=768, out_features=768, bias=True)
  )
  (attn_head): Linear(in_features=1, out_features=1, bias=True)
  (dropout): Dropout(p=0.1, inplace=False)
  (cls_head): Linear(in_features=768, out_features=1, bias=True)
  (class_loss_fnc): BCEWithLogitsLoss()
```

```
)
Total Number of Parameters = 118934785 ~ 119M
```

Appendix C

Lists of Keywords Used in the Dataset Labeling

We mentioned in the dataset labeling section 3.2.3, that the keywords we use for extracting outcome labels are of our choice, and they might change with respect to the legal corpus or court. Empirically, we noticed that certain expressions and words are present in the majority of the decisions; thus, we resorted to hand-crafted keyphrase-based regular expressions. Our choice of keywords is such that most of our case documents can be labelled successfully. The following are some examples of these keywords:

- appeal dismissed/allowed.
- appeal is dismissed/allowed.
- appeal should/will/must be therefore/accordingly dismissed/allowed.
- appeal should/will/must therefore/accordingly be dismissed/allowed.
- I/we * would/will dismiss/allow