Predicting Legal Case Outcomes with Deep Learning

By

Yifei Yin

Project Type: Thesis

Course Code: CISC 500

Supervisor: Farhana Zulkernine

Date: April 2020

Abstract

Predicting the outcomes of legal cases is an essential skill of lawyers. In this research project, we aimed to summarize the state-of-the-art models for both civil law and case law systems and further improve the accuracy of legal judgement prediction tasks. Previous efforts include using logical reasoning models [18,19], Support Vector Machine (SVM) [1,2,5,6,16,20,21], Decision Trees [17], Multilayer Perceptron (MLP) [15] and Recurrent Neural Networks (RNN) [4]. These studies have shown that legal reasoning models being the best overall model in legal judgement prediction tasks, followed by neural network models and then other models. However, the performance difference is narrow between logical reasoning models and deep neural networks. The small difference makes the massive development effort went into developing a logical reasoning models unjustified, as they require experts in the design process. Logical reasoning models also suffers from any structural change in the legal system which usually result in a complete redesign of the models. In this paper, we showed 1) using human extracted features, MLP can predict employment type at 91.7% accuracy, with robust stability when some criteria are missing from the case itself. 2) using deep language model Bidirectional Encoder Representation from Transformers (BERT) [30] and A Lite BERT (ALBERT) [7] over raw case text, whether there was a violation from European Court of Human Rights can be predicted at 88.29% accuracy.

Table of Contents

Chapter 1
Introduction

1.1. Motivation

Legal judgement prediction (LJP) is an essential skill for the defendant, the plaintiff and other stakeholders involved in a case to know the possible outcomes of a case in order to better plan for their actions with regards to those outcomes. However, LJP is a hard skill to acquire. Lawyers need extensive training. This is especially a problem in common law systems, where the outcome of a case is determined by interpreting previous cases, known as case law. This puts an incredible burden as people who does not have legal knowledge will not likely to gain much insight from case law. The amount of case law puts another weight on lawyers' daily work routine when dealing with a new case – finding relevant cases and reading those cases are time consuming. Reading all case law is practically impossible for any human to accomplish. This marks the necessary development of LJP models to better assist the public and legal professionals.

The development of LJP models can be roughly broken down to three stages. It started with logic-based models [18,19]. The development of these models requires experts to examine the human thought process of determining the outcome of cases. Logic models are expensive to develop and prone to small structural changes in the legal system. Automated models that uses statistical features of the texts became prevalent after logic methods. These models use text frequency-inverse document frequency (tf-idf) or Bag-of-Words (BoW) statistics with traditional machine learning methods to predict the outcome of cases. The accuracies of these models are far from optimal [6,16,20,21], especially for common law systems [2,5,17,19]. The latest kind of models are neural based models including MLP [15], RNN [4], and deep networks such as BERT [30,11,13,14].

The applications of deep neural networks are still lacking in LJP domain. Many new models such as BERT [30] and A Lite BERT (ALBERT) [7]. They have shown their abilities on understanding domain-specific information [12,13,14,15]. This research aims to give a more complete picture of deep neural network's performances on LJP with deep MLP and BERT.

1.2. Problem Description

An employee is a worker who enters into an employment contract with an employer that involves an exchange of labour for wages, and the contract is subject to the employment laws that are intended to protect employees. Independent contractors enter into a commercial contract to sell their labour in exchange of revenues and the change of profit. Legal employment systems in Canada and elsewhere have created a special legal regime to govern employment relations because employees are vulnerable and require legal protection, whereas independent entrepreneurs do not require the same level of protection [24]. Determine the type of employment status require comprehensive knowledge about the relevant case law and the inconsistency between different judges' decisions. This cannot be done by someone without legal knowledge.

We have decided to undertake this initiative of designing a platform for legal judge prediction for two main reasons. First, as we just mentioned, it is not easy, even for lawyers, to determine whether an employment relationship exists. In many cases, while Canadian courts have developed a robust and consistent legal test, workers exhibit a mix of associated features that could make the classification uncertain. Secondly, the rationale for a legal distinction between employees and independent contractors does not seem to be appropriate in the current economic context. While this discussion is outside the scope of this paper, it is important to note that over the past 40 years, the percentage of workers who enjoy the relative security of a 'standard employment relationship' has declined in Canada and elsewhere, notably, but not only, with the growth of the sharing economy. More than 40% of Canadians today work under other types of precarious work arrangements [23], including part-time or temporary work. The issue is that many of these workers may meet the technical definition of an independent contractor, and thus are excluded from most employment law's protections for employees1 [3]. As a result, we hope that this initiative may shed light on that issue for lawyers, but also for the average workers who is not equipped to determine his or her legal status. In particular, one of the medium-term objectives of this study is to develop an open-access Artificial Intelligence (AI) system that will

[1] it is important to note that many appellate jurisdictions, including in Canada with the Uber case, have to decide whether gig workers (for instance, uber drivers) are employees or independent contractors. For instance, the French Supreme Court have already considered that uber drivers, in some cases, can meet the technical legal definition of an employee.

help self-represented litigants to understand their basic legal rights as well as the likely judicial outcome if their case goes to court. In other words, the system would offer what in negotiation terms is called a Best Alternative to a Negotiated Agreement (BATNA) that may be used by the workers to leverage a settlement outside the courtroom.

Although previous studies have been investigating the effectiveness of machine learning on LJP, none of them provided the effectiveness of using language model on raw case text. Thus, we want to compare the ability of machine learning in predicting legal outcome in these two setting: use MLP 1) with human extracted features about the cases 2) with embeddings from state-of-the-art language models over the case text itself. This comparison can give us some insight about current language models' abilities to extract useful information from the cases, and if the extracted information is more useful to MLP compare to the features extracted by human.

However, due to the unavailability of the case text for the employee type classification task, the second stage of the research is replaced by predicting if there is a violation for cases from European Court of Human Rights (ECHR). In terms of the ECHR prediction task, comprehensive previous works have been done using both SVMs, RNNs and BERT-based models [13,16]. However, due to the size of the dataset being too small, the hierarchical attention mechanism [8] alongside BERT model might not achieved its full potential [13].

1.3. Key contributions

I developed a state-of-the-art model for employee type classification task that uses human extracted features as inputs.

For the ECHR prediction task, I developed a state-of-the-art model that can overcome the maximum sequence length constraint in BERT model family and achieved 88.29% accuracy. A data optimization strategy was employed to speed up the fine-tuning process which allows over 1,000 epochs of training in under 20 minutes on a single GPU.

1.4. Organization

In this report, I will discuss the research done on LJP tasks and other legal related tasks in Chapter 2. A brief overview and summary of the domain will be provided at the end of Chapter 2 including MLP, Decision Tree, SVM and deep neural networks. A number of natural language

processing and understanding techniques will be mentioned in chapter 2 as they are becoming the norm in LJP research, including tf-idf, BoW, word embeddings such as GloVe [31], language models such as BERT and ALBERT.

In Chapter 3, the implementation and experiments for the Employee Classification task will be illustrated. Chapter 4 will be implementation and experiments s for ECHR prediction.

Chapter 5 will summarize the two tasks mentioned above and critically evaluate the results. Future works will be discussed as well.

Chapter 2

Background

2.1 Background

Courts have struggled with the appropriate legal test to distinguish an employee from an independent contractor. Initially, courts looked primarily at the degree of control exercised over the worker (the 'control test'). However, that test was criticized for its inability to capture the complexity of the employment relationship. The Control test gave way to the fourfold test applied in the 1947 case of Montreal v. Montreal Locomotive Works Ltd involving (1) control; (2) ownership of the tools; (3) chance of profit; (4) risk of loss. In the 2001 671122 Ontario Ltd. v. Sagaz Industries Canada Inc., [2001] 2 SCR 983 the Supreme Court of Canada encapsulated these various tests in the following question: whether the person who has been engaged to perform the services is performing them as a person in business on his or her own account. In making this determination, the level of control the employer has over the worker's activities will always be a factor. The Court also identified the following (non-exhaustive) factors to consider: (i) whether the worker provides his or her own equipment, (ii) whether the worker hires his or her own helpers, (iii) the degree of financial risk taken by the worker, (iv) the degree of responsibility for investment and management held by the worker, and (v) the worker's opportunity for profit in the performance of his or her tasks.

2.2 Literature Study

Raghavan H. [1] used human extracted features to demonstrate that the performance of statistical models can be significantly improved even if the dataset is small. In a study by Nallapati et al. [2], Support Vector Machine (SVM) performed better with human-extracted features than with automatically-extracted features using a bag-of-words (BoW) approach for docket entry classification where each entry is an event happened on the timeline of the case. The task is to predict some aspect of the case, for example, whether the case was set on a full trial. The linear SVM with hand-crafted features was able to achieve 86.77% on F1 metrics, compared to only 79.44% achieved by the SVM using uni/bigrams features extracted using a BoW approach. However, the study did not use any neural-based machine learning models. A docket entry dataset with more cases used was investigated again by Vacek et al. using Gated Recurrent Unit (GRU) neural network model and its variations [4]. Nested-GRU outperformed SVM with a weighted F1-score of 0.91 compared to a 0.88 score achieved by the SVM. Similar situations are observed in civil law systems [5,20,21] where the outcome of a case is solely determined by the statute without considering the case law. The macro average F1-score of a GRU-based attention model is 95.42, which was significantly higher than the 88.53 achieved by SVM [5].

Lage-Freitas et al. [15] used MLP to predict whether a Brazilian State higher court (appellate court) will vote for fully favourable decisions, vote for partially favourable decisions, or deny the case appeal. They extracted the tf-idf statistics for the vocabularies presented in each case. The statistics emphasize words that are more frequent in the case description, but less frequent in the entire archive of cases. A multilayer perceptron was trained using fivefold cross-validation. The model was able to achieve a 78.99% F1-score.

Liu Z. et al. [6] compared the performances of several traditional machine learning models on predicting whether one of the articles labelled 3, 6 or 8 was violated by a case given in European Court of Human Rights (ECtHR) case law. The best performing model was SVM, which achieved an accuracy of 73.4, 87.5 and 77.7 respectively on the three articles 3, 6 and 8. The feature extraction method they used was a frequency-based N-gram method. In their study, they did not measure the performance of models using manually-extracted features.

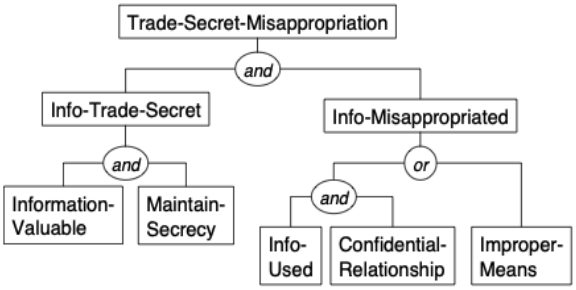
Aletras et al. [16] investigated predictive modeling to detect which of the articles from 3, 6, and 8 were violated in the ECtHR case law. They implemented linear SVM and compared the model performance when different sections of the document were converted to N-gram

representations. This frequency-based extraction along with SVM showed that the 'Circumstances' section in the ECtHR case can best inform the model to make the best prediction compared to the other sections. The 'Circumstances' section is rich in information about the fact of the case. They combined this section with topics extracted from each case to further raise the accuracy from 76% to 79%.

Martin et al. [17] implemented a group of decision trees to determine the outcome of a case. They manually extracted facts about each of the 628 United State Supreme Court cases. This extraction method is context-based, meaning that the value of each feature might not be directly present in the case, and could have been crafted by human readers according to the context of the document. Martin et al. made 11 decision trees. The first two trees determined whether a case was likely to be a unanimous liberal or conservative decision. If none of the trees predicted a unanimous decision, then the rest of the nine justice trees will vote on the case individually. Each of the nine justice trees was designed according to the past behavioral pattern of each judge in the Supreme Court. One drawback of this implementation was that the system breaks when a justice leaves the court. However, this model outperformed human experts, scoring 75.0% accuracy compared to human experts' 59.1% accuracy.

Brüninghaus et al. [18] designed a sophisticated reasoning-based model in order to predict the outcome of trade secret cases. Unlike statistical models, IBP (Fig. 1) uses case-based reasoning to construct a case-based argument. Cases have to be mapped to 26 factors and prototypical patterns. Each factor that belongs to the same issue is then evaluated as favoring either the plaintiff or the defendant. If all factors of the same issue favored one party, then that issue is said to be biased on one side. If all factors did not favor one side, the model will find cases with the same factual patterns and check which side is favored more than the other. The model then analyzes the favored side for each issue and defines a logic tree to determine the final outcome of the case. They achieved an accuracy of 91.4%. The model provides reasoning that is comprehensible by human. However, the development of this model is expensive and requires human expertise.

Figure 1: Logical structure for trade secrets law [18]



Katz et al. [19] successfully predicted whether a case will be reversed or affirmed by the Supreme Court of the United States with an average accuracy of 70.2%. Their main contribution is that the model they developed can predict the outcome of cases from 1816-2015 equally well. They extracted 61 features from each case to build a random forest model using the context-based feature extraction method. The random forest started with 125 trees. To build a forest for each year, the 120 best trees were selected from the previous year, and five more trees were randomly initialized and added to the forest. This accommodated changes in the cases and the effect of shifts in court decisions over time. The forest was trained on the data from the current year, and the growing cycle repeated.

There is a noticeable difference between models' performances in common law systems and civil law systems. Whether the features are crafted from text automatically or hand-crafted by humans also make a difference. We summarized the best-performing models in different legal domains from different legal systems. We created a table of different models' accuracy (Table 1). The reason for comparing accuracy instead of F1-score is because many of the studies did not report their F1-score. Most research which reported an F1-score did not specify whether they used a micro F1-score or a macro F1-score.

Table 1: An overview of the best accuracies of different models in legal judgement prediction tasks

Model (Accuracy)	Human Extracted	Text Feature
Logic Method	IBP (91.4) [18]	/
Statistical Method	Random Forests (70.2) [20]	SVM (79.0) [17]
Neural Network	/	Hier-BERT (83.32) [13]

[2] The accuracy is calculated from the Precision score, Recall score and the number of positive and negative examples in the dataset.

In sum, logic methods and neural networks outperformed SVM for both human-extracted features and statistical features extracted from BoW or tf-idf. It is worth mentioning that developing a logic method is expensive and requires extensive domain-specific knowledge from experts, while neural networks can learn with little human intervention.

This dataset was previously used to train several statistical tools or non-neural based machine learning models [3]. Logistic regression performed the best when outliers from the dataset were removed, and certain features that are important to judges' outcomes are enhanced by squaring the original numerical value with an accuracy of 90%. This project aimed to test the performance of a neural-based model, specifically the MLP, without removing the outliers or feature engineering.

Next, I want to provide an overview of techniques used in natural language processing tasks (NLP).

Prior to word embeddings, vocabularies from a language were represented using one-hot encodings. One-hot encoding is sparse, its size is linear to the vocabulary count in a language. It fails to capture the semantic relation between different vocabularies due to the orthogonality of the representations. Word embeddings on the other hand, represent vocabularies in a continuous space [25]. A vector is initialized with values between 0 and 1. Using skip-gram training, the vector should contain sufficient information in order to predict words surrounding itself. This method generates word embeddings that successfully capture semantic relations between words and syntactical properties of vocabularies. Word embeddings drastically decrease the complexity of word representation to O(1) while being able to extracting meaningful components from words. They are used in neural networks and other language models to represent a word, or a collection of words – a sentence or a paragraph.

Two types of models are used to encode a word sequence (later referred to as a sentence, but it can also be a paragraph or a corpus). They are Auto-Encoder (AE) and Auto-Regressor (AR). In the case of Auto-Encoder (AE), the model predicts the missing or corrupted words based on the other words in the same sentence. AE does not have the aspect of time which means it cannot perform generative tasks. However, it can incorporate context from both sides of the center word. Recurrent Neural Networks (RNNs) are networks that follow along a temporal sequence, processing the next time step in combination with the internal state generated after

previous time steps. This kind of architecture is AR. AR is unidirectional, it fails to incorporate context from both side of the center word.

An example of an Auto-Encoder is BERT, Bidirectional Encoder Representations from Transformers, is a model that performed well in several language tasks including question answering, summarization and sentiment analysis. The success of the model will not be discussed here since there are many versions of BERT, each one of them was benchmarked against several measurements (ref needed). The focus of this section will be trying to explain the success of BERT and some later versions of the same model that achieved even better results. As an AE, BERT learns about the context from both sides of the center word. For example, "Cat likes fish". It would be easy for a unidirectional and forward learning model to predict 'fish' when it already saw 'Cat'. However, if the word 'Cat' is missing, then the model will have no idea about what to put as the first word of a sentence, since it cannot predict in the reverse direction because of the way the model was trained. BERT, as it is trained both in forward and reverse directions, can see the entire sentence excluding the words that are missing. So, it can learn to predict the missing words using the available information. Previous AEs like ELMo, failed to utilize this bidirectional information [26]. As a result, they performed worse than BERT [11].

However, negative impact on model's performance has been observed when the number of layers is too large [11]. The final layer is the output layer, a sequence of output corresponding to the input sequence will be generated. The one extra token added at the beginning of the sentence before feeding it into the model is used to generate an extra token at the output sequence that can be used for different tasks. Because BERT lets each token to utilize information of the entire input sequence, the one extra token contains useful information about the entire sentence. This extracted information can be the input of a feedforward network to perform classification tasks which is in the key interest of my research.

XLNet, an Auto Regressor that performed better than BERT in a lot of Natural Language Understanding tasks [27]. One of the reasons is XLNet is a regressor. It was trained by predicting a group of missing words in different sequences. One example of this is "New York is a city". Since 'New York' is an entity, 'New' and 'York' are dependent on each other. In the case of BERT, the model would not see the significance of the word pair, rather, it sees the significance of each word with respect to the entire sentence. Another reason is, XLNet uses permutation in

its training. Previously, auto regressors can only predict the sentence in a particular sequence. However, XLNet creates permutations of the collection of words it has to predict in each sentence. This virtually augmented the training dataset and was the main contribution of XLNet. What's interesting is, XLNet used TransformerXL. However, transformers excluded the idea of sequential regression, which was the main characteristic of regressors. So, it might be important for Natural Language Understanding models to have characteristics of sequential regression and sequential prediction, i.e. AR and AE.

BERT can perform well not only in general language tasks, but also in domain specific tasks, especially when it is trained on text corpus from that domain. The original BERT trained on English Wikipedia and Book Corpus, in total 3.3B tokens [11], performed relatively poorly compare to domain specific BERTs for domain specific NLU tasks [28,12]. SciBERT [28], a BERT from scientific text, performed on average +2.11% better than BERT on a variety of tasks on Biology, Computer Science and multidomain fields. It is trained on Biomedical paper and Computer science paper, in total 3.1B tokens. BioBERT is a model trained specifically for biomedical texts [12]. Specifically, BioBERT v1.1 outperformed BERT in named entity recognition (NER) and question answering by a large margin, up to a +7.55% on question answering on the BioASQ 6b dataset [29]. Although this improvement is significant, BioBERT was trained for 1.47M steps compare to 1M steps BERT was trained on. It was also trained on more data, 21.3B compared to only 3B that BERT was trained on. However, unlike BERT where all the data were used to train the model at the same time, BioBERT used one dataset to train, and then another one later instead of training simultaneously on multiple datasets.

Chapter 3

Implementation (Employee Classification)

3.1 Data Description

The dataset was completed with the assistance of about 20 lawyers, as the objective of this project was to analyze approximately 1,200 adjudicated cases, of which nearly 900 were applicable for our use. The dataset is considered to be hand-annotated based on the Sagaz classification test mentioned. In particular, the legal analysts have collected case data going back to 2001 and analyzed it to extract 16 key features, represented in a table format for all the cases and listed in Table 2 below. These key features constitute the determining factors that judges take into consideration to determine the legal status of a worker. It is important to note that most of these features are legally relevant features, that is features recognized by the case law. However, we have also taken into consideration non-legally relevant features, such as age of the worker, industry or gender in order to improve the predictability of our models but also to understand the weight of these external variables on the judicial decision-making process.

An entry in the table is set to 0 if the corresponding feature description is not available in the case data. In Table 2, the number under each feature indicates the number of categories for a feature. Except the 'Length of service', all the listed features have categorical values in the original case description. Later we converted the length of service to a categorical value by assigning a range of values to a specific category and defined 9 such categories for this feature.

Table 2 Features extracted to create the contractor dataset used in our research

Year 10	Length of service 9	Ability to hire employees 3	Who sets the work hours? 4
Court 2	Chance of profit 6	Exclusivity of services 3	Supervision review of work 2
Province 14	Type of Job 10	Delegation of tasks 2	Where the work is performed? 2
Risk of loss 3	Industry 4	Ownership of tools 5	Is the worker required to wear uniform? 2

Each case can have four different outcomes, namely employee, independent contractor, dependent contractor and not applicable. A case was labeled as not applicable when it did not have anything in regard to employee categorization. We removed such data from the contractor dataset. The remaining data reflected the distribution given in Table 3.

Table 3: Class distribution of the dataset after screening

Class Name	Number of Observations
Class 1 – Employer	227
Class 2 – Independent Contractor	159
Class 3 – Dependent Contractor	14

3.2 Implementation Setup

In this section we describe the implementation and experimentation of the models to validate their accuracy and performance in automatically identifying employees and contractors. We consider this a classification problem and applied supervised learning to train and validate the models using the contractor data containing the 16 features extracted manually from the case descriptions. We validated our models using precision, recall and F1-score in experiment 1, and accuracy for other experiments. The equations for computing these measures are given below

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{Positive + Negative}$$

The dataset is well structured. We removed columns that contained the name of the judge and the name of the law researcher who collected the data. The remaining dataset contained only the numerical information used in our model.

Experimental Setup: The experiments were conducted on MacOS 10.15 (Catalina). We used an Intel i5-8259U CPU, running at 3.4GHz turbo frequency with four physical processing cores and scikit-learn [22] and Keras [10] to implement the models. For experiment 2, we ran the experiments in parallel on an Intel i9-9900K Windows 10 machine with eight physical processing cores at 5.0GHz to speed up the training in parallel.

Data Partitioning for Training and Validation: The dataset was partitioned into 80% training and 20% test data. However, with pilot experiments, we found that random partitioning of the dataset led to high fluctuations in model accuracy. Because of this, a fivefold cross-validation was used in which the data was randomized and partitioned into 80%-20% in a rolling manner from different starting points. The training was stopped when the validation loss started to increase or accuracy started to decrease consistently for five epochs.

3.3 Methods

Our implementations of the models are described below.

A. Logistic Regression (LR)

Logistic Regression was used to establish the baseline performance of predictive models. It was used to show the limitations of a linear regression model in a complex classification task.

B. Decision Tree (DT) [9]

Decision Tree was one of the interpretable classification models. This model tends to overfit its training set. The depth of the decision tree was set to 6 empirically because it gave the highest accuracy on the training dataset.

C. Support Vector Machine (SVM) [22]

SVM showed decent results in a few legal classification tasks [2,4,5,6]. It constructs hyperplanes in the problem space to separate points in the space according to their classes. It

usually performs better than other classification models on smaller datasets as it does not tend to overfit the training set.

D. Multi-Layer Perceptron (MLP) [22, 23]

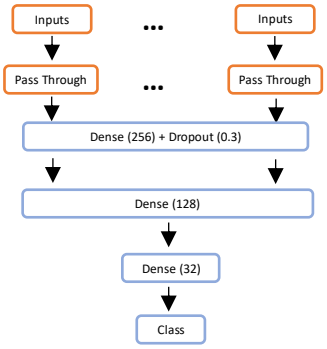


Figure 2 Multi-Layer Perceptron

Multi-layer perceptron (MLP) (Fig. 2) was used in this research to establish a baseline performance for neural network based models. It consists of four fully-connected layers or dense layers. The dense layers contain 512, 256, 128, and 32 neurons respectively. The last layer contains the same number of neurons as the number of classes in each experiment. The MLP was optimized using an Adaptive Momentum optimizer, random dropout with probability of 30% and L2 regularization.

E. Extended Multi-Layer Perceptron (eMLP)

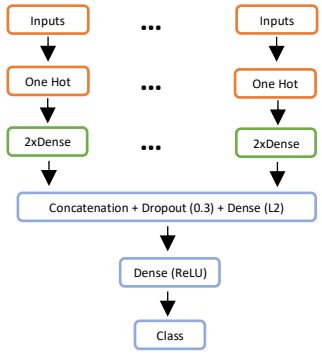


Figure 3 Extended Multi-layer Perceptron

The extended multi-layer perceptron (eMLP) (Fig. 3) converts inputs to one-hot encodings instead of letting them pass through directly. This helps the network to parallelize the computation. The encodings are passed through two dense layers. The output from the dense layers are concatenated, 30% of the tensors are dropped randomly during training to prevent overfitting. A dense layer with L2 regularization and a dense layer with ReLU nonlinearity predicts the outcome of the case, i.e. the class.

3.4 Experiments

We ran four experiments using the above-mentioned model. Experiment 1 compares the accuracy of each model when all of the features are available to the models. Experiment 2 compares the change in their accuracy when certain features are nullified. Experiment 3 tests to the extent to which each model can be trained when certain features are removed before the

training session. Experiment 4 compares the model accuracy when the model classifies all 3 classes of employment types.

3.4.1 Experiment 1 - Model Performance Comparison

In this experiment, the models were trained on five training sets and five testing sets using fivefold cross-validation method. The five predicted value sets were then merged into one confusion matrix. Precision, Recall, Accuracy and F1-score were calculated according to that matrix as shown in Table 4.

Table 4 Experiment 1: model accuracies and nullified-feature inference accuracy

Score	Precision	Recall	Accuracy	F1
eMLP	89.4%	90.6%	91.7%	90.0
Random Forest	86.4%	83.6%	87.8%	85.0
Decision Tree	87.4%	74.2%	85.0%	80.3
Logistic Regression	79.0%	71.1%	80.3%	74.8
SVM	82.5%	71.1%	81.9%	76.4
MLP	76.0%	69.8%	78.8%	72.8

3.4.2 Experiment 2 - Missing Feature Inference

Models were configured as described in Section 1. They were trained on all the features presented in the dataset just as in Experiment 1. Then one feature column was replaced by 0 entries, then used to evaluate each model's performance.

As shown in Table 5, a model trained on all features should show decreased performance when a feature is nullified. This is because the models' abilities to infer the outcomes depends on all of the features, although a minor performance increase can be seen in every model when certain features were nullified. It is worth noting that although Decision Tree outperformed Random Forest when all features were presented, its performance varied depending on which feature was nullified. A huge performance impact was observed when Type of Job was nullified for Decision Tree. eMLP showed the best performance and stable inference accuracy when certain features were nullified.

Table 5 Experiment 2: model accuracies and nullified-feature inference accuracy

Accuracy (%)	eMLP	DT	RF	MLP	LR	SVM
Full Features	91.7	85.0	87.8	78.8	80.3	81.9
Is the worker required to wear a uniform?	92.7	83.6	85.7	76.3	75.0	76.1
Length of Service	92.0	84.3	84.1	78.3	74.7	75.2
Court	91.3	85.1	85.5	62.9	54.3	71.2
Ability to hire employees	90.7	85.9	86.9	78.6	79.7	81.3
Province	90.6	84.8	86.1	76.4	68.1	76.6
Where the work is performed	89.8	85.1	81.9	69.2	67.4	72.0
Industry	89.7	84.8	84.7	72.3	71.6	82.1
Year	89.5	84.2	85.7	69.0	74.4	78.3
Type of Job	89.5	46.3	85.4	70.6	78.1	81.0
Who sets the work hours	89.4	81.3	81.8	81.7	89.2	85.3
Supervision/review of work	88.7	81.5	86.5	82.7	80.3	84.3
Risk of loss	87.4	86.2	83.4	79.1	83.0	86.3
Ownership of tools	85.5	82.9	85.0	80.2	88.2	88.4
Exclusivity of services	84.5	83.5	84.9	77.6	83.8	83.6
Chance of profit	84.2	83.7	83.7	75.7	86.4	86.1
Delegation of tasks	83.9	62.5	84.2	77.9	86.3	91.4

It is also worth noting that the nullification of Delegation of Tasks resulted in a major performance impact in both the eMLP and Decision Tree, which was identified as one of the most important factors in judges' decision making process [3].

3.4.3 Experiment 3 - Missing Feature Training

To test a model's robustness against missing features, each model was trained on a dataset with one feature removed at a time, with replacement. This experiment tested model performance independent of the nullified feature. This is different from Experiment 1, since the features were not nullified during training. In Experiment 1, the model was trained on all features, then tested on nullified features. In other words, the models from Experiment 1 were

aware of the existence of the nullified feature. But in Experiment 2, the models were trained independent of the masked feature. The results were recorded in Table 6.

All models apart from Decision Tree have differences in accuracy of less than 5%. Delegation of Tasks is one of the two important factors that go towards the judges' decisions. In the experiment, eMLP trained without Delegation of Tasks resulted in the lowest accuracy among other eMLPs.

Table 6 Experiment 3: model accuracies and missing feature model accuracy

Accuracy (%)	eMLP	RF	DT	SVM	LR	MLP
Full Features	91.7	87.8	85.0	81.9	80.3	78.8
Province	92.0	87.3	86.3	81.1	79.5	80.1
Length of Service	92.0	85.2	83.9	80.8	80.3	81.6
Type of Job	91.7	85.2	84.7	81.3	80.1	80.3
Ownership of tools	91.5	84.7	84.5	81.6	81.1	79.5
Where the work is performed	91.2	86.3	85.5	81.3	80.6	79.8
Is the worker required to wear a uniform?	91.2	86.3	85.2	81.3	81.1	78.0
Ability to hire employees	91.2	88.1	82.4	81.1	80.3	79.3
Year	91.2	88.3	84.5	81.1	79.8	79.5
Industry	90.9	87.0	85.2	81.6	79.0	76.7
Chance of profit	90.7	85.8	85.2	81.3	79.5	78.8
Risk of loss	90.7	83.7	84.5	82.1	80.1	78.5
Who sets the work hours	90.4	85.8	83.2	81.1	79.8	78.2
Court	90.4	87.0	85.0	80.8	78.5	79.0
Supervision/review of work	90.4	85.5	86.0	80.3	80.6	78.2
Exclusivity of services	89.6	85.2	83.2	79.0	78.8	78.0
Delegation of tasks	89.4	84.7	77.5	78.8	78.2	79.5
Maximum Accuracy Deviation	2.59	4.60	8.81	3.37	2.85	4.92

3.4.4 Experiment 4 - 2 Classes vs. 3 Classes Performance

Experiments 1, 2, and 3 were predicting whether an individual was an independent contractor. In essence, the models were asked to undertake a binary classification task. This choice was made because there were only 14 samples of "dependent contractor". An extra

experiment was conducted to measure the models' performance in 3-class classification. The results are shown in Table 7.

Table 7 Experiment 4: model accuracy comparison between predicting 3 classes and 2 classes

Accuracy (%)	3 Classes	2 Classes
eMLP	88.2	91.4
Random Forest	84.5	85.0
Decision Tree	79.7	85.1
Logistic Regression	78.0	78.1
SVM	76.5	78.1

Only SVM was able to make one true positive prediction of a sample that belongs to Class 3 – dependent contractor. The other models failed to make any correct prediction of the Class 3 label. However, eMLP exhibited the highest overall accuracy.

3.5 Summary

In all experiments, the best performance was achieved by eMLP. It showed minor susceptibility towards nullifying features but was still robust when trained with certain features missing. Decision Tree and Random Forests were significantly better than MLP, logistic regression and SVM. While it exhibited similar performance when all features were presented, Random Forest was less susceptible to variations resulting from nullified or missing features.

eMLP, Decision Tree and Random Forests all showed significant accuracy loss when the Delegation of Tasks feature was nullified or removed. In the case of Decision Tree, it is because Delegation of Tasks is the root node of the tree. Delegation of Tasks shows the power of the employer over the individual's tasks and timelines, which helps define employee status.

Chapter 4

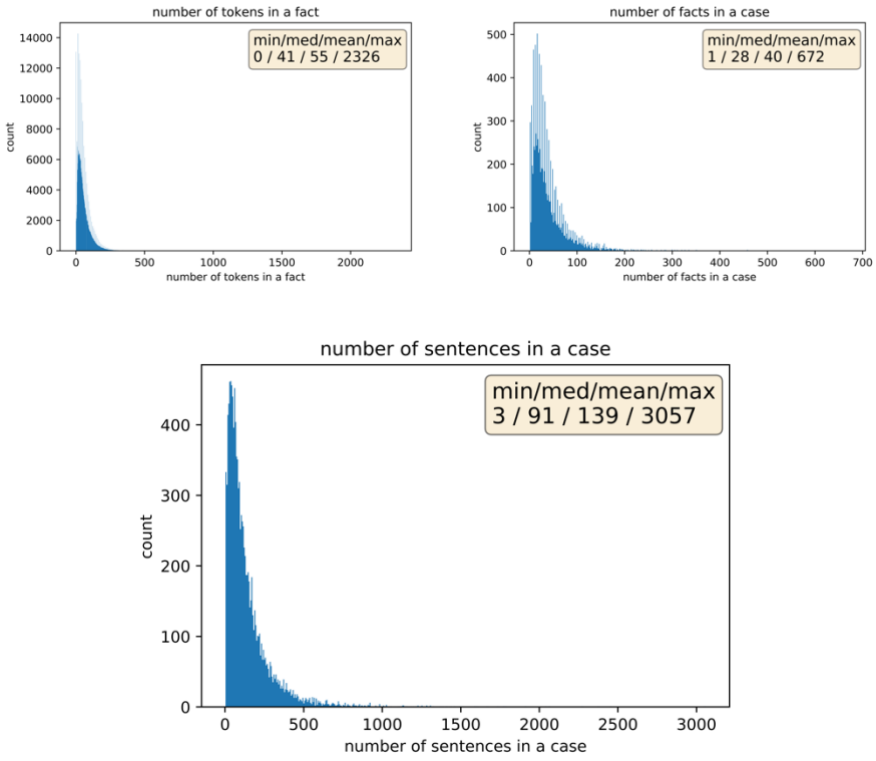
Implementation (ECHR Prediction)

4.1 Data Description

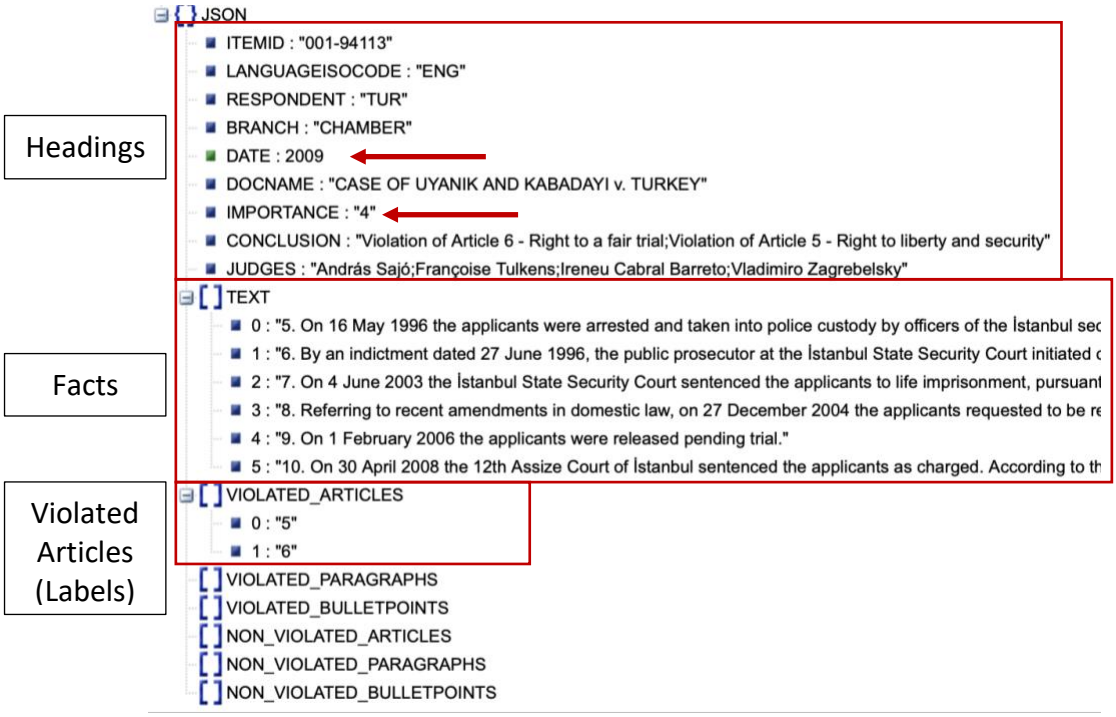
The dataset is extracted from the ECHR public dataset using regular expressions from case description as described in previous literature [16]. The dataset is divided into three parts, training (7,100 samples), development (1,380 samples) and test set (2,998 samples). The training set and development set contains cases from 1959 to 2013. The test set contains data from 2013 to 2018. The training and development set are balanced, contains equal number of cases with violations and cases without violations. The test set contains 66% cases with violations. The same dataset was previously used in [13].

Each of the cases contains couple of facts, each fact consists several sentences. The number of tokens in a fact, the number of facts in a case, and the number of sentences in a case are shown below (Fig 4)

Figure 4 Overview of the Dataset Size



Sample file (001-94113.json): The sample is a short sample. Some cases contain a few hundred facts, a few thousands of words as shown in Fig 4. Which means the dataset cannot be used with ALBERT or BERT given the maximum length of BERT is 512 tokens. Structural modification is needed.



It's worth noting that unlike the sample file, some cases in the dataset contains foreign words that cannot be indexed by BERT.

4.2 Implementation Setup

The setup used is the same as mentioned in Section 3.2, apart from the dataset partition which is described in Section 4.1 instead.

4.3 Validation

Validation was done post training using the test set partition from the dataset. The macro-average Precision, Recall, F1-score and Accuracy metrics were calculated as Section 3.2. The development set was used to help to avoid overfitting on the training set and finding the correct hyperparameters for the model itself.

4.4 Experiment

The BERT-base-v1 [30] model and ALBERT-xxl-v2 [7] were used for this experiment. BERT version 2 models were not used for the sake of fair comparison between this study and what was done before [13].

In order to avoid repeated computation and speed up the training, the embeddings from language models were computed once on the GPU then copied to the main memory before being stored on the hard drive for later epochs. This significantly speed up the training by roughly 30,000 times, since the most time-consuming part was avoided during training. This method is equivalent to converting the text dataset to embeddings first before training. The drawback of this method is language will not be able to be trained, only the fine-tuning layer was able to be trained.

The individual token embeddings were summed up for each fact. This ensure that the fact representation has fixed length for each fact, which is equal to the hidden size of the language model. Then max pooling and average pooling were used on all fact representations for a case, which becomes the case embedding. The size of the case embedding was 2 * hidden size of the language model. A simple MLP was used as the network on top of BERT with a dropout of 30% used during training to help avoiding over fitting.

The models were trained on training set for 2000 epochs. The epoch with lowest Mean Squared Error loss value on the development set will be chosen and used on the test set.

4.5 Results

Metrics	BERT base v1	ALBERT xxl v2
Precision	86.28	87.76
Recall	83.91	85.82
F1	84.89	86.65
Accuracy	86.82%	88.29%
Confusion Matrix	765 259 136 1838	799 255 126 1848

Chapter 5

Conclusion and Future Work

5.1 Summary

The problem of Legal Judgement Prediction has been studied in the past, using logical models. Those models were replaced by traditional machine learning models like SVM as they become more prevalent under the umbrella term machine intelligence. Just recently, complicated deep neural language models showed great performance in complex language tasks, which motivated us to develop applications for them in the legal domain.

We first formulated the problem of comparative Legal Judgement Prediction using human extracted features and language models. Then we identified the features of employment cases and did comparative analysis on the performance of traditional machine learning models and neural based models. An extended multilayer perceptron showed robust performance even when some features were missing at inference time.

Then we moved on to predicting binary violations for cases from European Court of Human Rights. We crafted state-of-the-art model to predict if there was any violation in a given case. BERT-base achieved a macro F1-score of 84.88, lower than that achieved by ALBERT-xxl, which is 86.65, but still being higher than the previously the best which was 82.0.

5.2 Limitations

Although summing up embeddings and using pooling to format the representation into the same shape can easily satisfy the structural requirement for backpropagation, the process does not make much sense. No previous studies have shown the orthogonality of word embeddings. Any operation that involves using addition across the same dimension will result a reduction in the amount of information in the word embeddings. Which means the network will know less information about the text than optimal.

5.3 Future Work

As mentioned in Section 5.2, the operation used to create a fix-sized case embedding have some theoretical issues. This is a prevalent issue in NLP with models that require backpropagation. An innovative way of retaining the information about the text while having the same tensor size is needed to resolve this issue.

Secondly, BERT-based model has shown marginal improvements when they are pre-trained on the text corpus [11]. Pre-training is very time consuming even on the fastest single GPUs. However, it is something that can be easily done with enough time.

References

[1] H. Raghavan, "Active Learning with Feedback on Both Features and Instances," *Current Issues in Journal of Machine Learning Research*, pp. 1655–1686, 2006.

[2] R.Nallapati and C. D. Manning, "Legal docket-entry classification," *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

[3] Maxime Cohen, Samuel Dahan and Benham Manhavi "Predicting Worker Classification with Machine Learning", 2019 (submitted) *Journal of Empirical Legal Studies*.

[4] T. Vacek, R. Teo, D. Song, T. Nugent, C. Cowling, and F. Schilder, "Litigation Analytics: Case Outcomes Extracted from US Federal Court Dockets," *Proceedings of the Natural Legal Language Processing Workshop*, 2019.

[5] B. Luo, Y. Feng, J. Xu, X. Zhang, and D. Zhao, "Learning to Predict Charges for Criminal Cases with Legal Basis," *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2017.

[6] Z. Liu and H. Chen, "A predictive performance comparison of machine learning models for judicial cases," *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017.

[7] X. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations," *arXiv:1909.11942 [cs.CL]*, Sep 2019

[8] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical Attention Networks for Document Classification," *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[9] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, pp. 81–106, 1986.

[10] F. Chollet and others, Online at *https://keras.io/*, 2015 last accessed on Mar 11, 2020.

[11] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *arXiv:1901.08746 [cs.CL]*, 2019.

[12] E. Elwany, D. Moore, G. Oberoi, "BERT Goes to Law School: Quantifying the Competitive Advantage of Access to Large Legal Corpora in Contract Understanding," *arXiv:1911.00473 [cs.CL]*, 2019.

[13] I. Chalkidis, I. Androutsopoulos, N. Aletras, "Neural Legal Judgment Prediction in English," *arXiv:1906.02059 [cs.CL]*, Jun 2019

[14] Amanda Bertucci and Lauren Butti,. Supreme Court of Canada clarifies contractor vs. employee classification in Quebec franchisee case, online at https://www.hrreporter.com/employment-law/news/supreme-court-of-canada-clarifies-contractor-vs.-employee-classification-in-quebec-franchisee-case/307929, 2019 last accessed on Mar 11, 2020.

[15] A. Lage-Freitas, H. Allende-Cid, O. Santana, L. Oliveira-Lage, "Predicting Brazilian court decisions," *arXiv:1905.10348 [cs.SI]*, 2019

[16] N. Aletras, D. Tsarapatsanis, D. Preoţiuc-Pietro, and V. Lampos, "Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective," *PeerJ Computer Science*, vol. 2, 2016.

[17] A. D. Martin, K. M. Quinn, T. W. Ruger, and P. T. Kim, "Competing Approaches to Predicting Supreme Court Decision Making," *Perspectives on Politics*, vol. 2, no. 04, pp. 761–767, 2004.

[18] S. Brüninghaus and K. D. Ashley, "Predicting outcomes of case based legal arguments," *Proceedings of the 9th international conference on Artificial intelligence and law - ICAIL 03*, 2003.

[19] D. M. Katz, M. J. Bommarito, and J. Blackman, "A general approach for predicting the behavior of the Supreme Court of the United States," *Plos One*, vol. 12, no. 4, Dec. 2017.

[20] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, J. Xu, "CAIL2018: A Large-Scale Legal Dataset for Judgment Prediction," *arXiv:1807.02478 [cs.CL]*, 2018.

[21] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, "Legal Judgment Prediction via Topological Learning," *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol.12, pp. 2825-2830, 2011.

[23] Social Development Canada, "Government of Canada," *Canada.ca*,. Online at https://www.canada.ca/en/employment-social-development/services/consultations/what-was-heard.html, 2017, last accessed on 10-Mar-2020.

[24] G. Davidov, B. Langille, "The Reports of My Death are Greatly Exaggerated: "Employee" as a Viable (Though Over-used) Legal Concept," *Boundaries and Frontiers of Labour Law*, Oxford, UK, Hart Publishing, pp. 133–152.

[25] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean "Distributed Representations of Words and Phrases and their Compositionality", *arXiv:1310.4546 [cs.CL]*, Oct 2013.

[26] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep contextualized word representations", *arXiv:1802.05365 [cs.CL]*, Feb 2018.

[27] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov and Q. V. Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding", *arXiv:1906.08237 [cs.CL]*, Jun 2019

[28] I. Beltagy, A. Cohan and K. Lo. "SCIBERT: Pretrained Contextualized Embeddings for Scientific Text", *arXiv:1903.10676 [cs.CL]*, Sept 2019

[29] Tsatsaronis,G. et al, "An overview of the BIOASQ large-scale biomedical semantic indexing and question answering competition". *BMC Bioinformatics*, 16, 138., Apr 2015.

[30] J. Devlin, M. Chang, L. Lee, K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs.CL]*, May 2019

[31] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.