Using Machine Learning to Predict Judicial Decisions

Conor O'Sullivan BBusSc

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Data Science)

Supervisor: Joeran Beel

August 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Conor O'Sullivan

August 14, 2019

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Conor O'Sullivan

August 14, 2019

Acknowledgments

Firstly, I would like to thank the team at vizlegal for giving me access to their API and allowing me to discuss some of my results and ideas. Specifically, Gavin Sheridan who authorised and initiated the whole process and José Lopez who helped with the technical details and any questions I had about the data. Without your help and access to the data, the process of writing the dissertation would have been far less enjoyable.

Secondly, I would like to thank my supervisor Professor Dr Joeran Beel who provided valuable insight and advice throughout the process of writing the dissertation.

Lastly, I would like to thank my parents Kevin and Linda O'Sullivan. Your love and encouragement carried me throughout the entire MSc program and gave me much needed support, especially during the dissertation.

Conor O'Sullivan

University of Dublin, Trinity College
August 2019

Using Machine Learning to Predict Judicial Decisions

Conor O'Sullivan , Master of Science in Computer Science

University of Dublin, Trinity College, 2019

Supervisor: Joeran Beel

In this study, machine learning models were constructed to predict whether judgements made by the European Court of Human Rights (ECHR) would lead to a violation of an Article in the Convention on Human Rights. The problem is framed as a binary classification task where a judgement can lead to a "violation" or "non-violation" of a particular Article. Using auto-sklearn, an automated algorithm selection package, models were constructed for 12 Articles in the Convention. To train these models, textual features were obtained from the ECHR Judgment documents using N-grams, word embeddings and paragraph embeddings. Additional documents, from the ECHR, were incorporated into the models through the creation of a word embedding (echr2vec) and a doc2vec model. The features obtained using the echr2vec embedding provided the highest cross-validation accuracy for 5 of the Articles. The overall test accuracy, across the 12 Articles, was 68.83%. As far as we could tell, this is the first estimate of the accuracy of such machine learning models using a realistic test set. This provides an important benchmark for future work. As a baseline, a simple heuristic of always predicting the most common outcome in the past was used. The heuristic achieved an overall test accuracy of 86.68% which is 29.7% higher than the models. Again, this was seemingly the first study that included such a heuristic with which to compare model results. The higher accuracy achieved by the heuristic highlights the importance of including such a baseline.

Summary

The study looked at predicting the outcome of judgements made by the ECHR using machine learning. This was to address the problem of a large application backlog faced by the Court which subsequently leads to large delays. To address this problem, the study seeks to determine how well the judgements made by the Court can be predicted.

Some of the main shortcomings of previous works were addressed. This includes the lack of a realistic test set used to determine the accuracy of the models. Similarly, previous studies did not include a simple heuristic as a baseline. These studies have compared their cross-validation accuracies to 50%, the accuracy of a random guess if judgements were balanced. This comparison is flawed as in the past judgements have not been balanced. Additionally, it was determined that model selection and feature engineering could be improved.

The problem is framed as a binary classification task where a judgement can lead to a "violation" or "non-violation" of a particular Article. The auto-sklearn package was used for model selection and hyper-parameter tuning. Models were constructed for 12 of the Articles in the European Convention on Human Rights. To train models, textual features were obtained from ECHR documents. The most important documents are the Judgments. This is because the models aim to predict the outcome of the judgements made by the Court. The textual features obtained include N-grams, average word embedding and paragraph embedding features. In creating the average word embeddings, three different word embedding models were used. These are a general GloVe embedding and two legal embeddings: law2vec and echr2vec. The echr2vec embedding was created using all the ECHR documents obtained. Similarly, these documents were used to train doc2vec model which was used to obtain the paragraph embedding features.

The models achieved a weighted average accuracy of 68.83% across the test sets of all 12 Articles. In comparison, the average accuracy of the heuristic was 29.7% higher than

the models. Ultimately, the accuracy achieved was deemed to be too low for the models to be used by the Court to make judgements. However, it was argued that the models could still be used by the Court to prioritise applications.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

The European Court of Human Rights (ECHR) is an international court that examines potential breaches of the European Convention on Human Rights. The Convention consists of numerous "Articles". These Articles and the rights they protect are shown in Table 1.1. The majority of the judgements made by the court concern Article 6 where about 50% of all violations are due to breaches of this Article (Council of Europe 2014).

Article	**Title**
Article 2	Right to life
Article 3	Prohibition of torture
Article 4	Prohibition of slavery and forced labour
Article 5	Right to liberty and security
Article 6	Right to a fair trial
Article 7	No punishment without law
Article 8	Right to respect for private and family life
Article 9	Freedom of thought, conscience and religion
Article 10	Freedom of expression
Article 11	Freedom of assembly and association
Article 12	Right to marry
Article 13	Right to an effective remedy
Article 14	Prohibition of discrimination
Article 18	Limitation on use of restrictions on rights

Table 1.1: Articles in the European Convention on Human Rights (Council of Europe 1950)

According to Council of Europe (2014), for a potential breach of the Convention to be

investigated an application must first be made. This means the ECHR cannot investigate potential violations on its own accord. Any State or individual can make an application but cases can only be made against a State. Specifically, one of the 47 States that has ratified the Convention. Since its founding, the Court has been very successful leading to a growing number of cases. In the Court's own words:

> "The Court has been a victim of its own success: over 50,000 new applications are lodged every year. The repercussions of certain judgments of the Court, on a regular basis, and the growing recognition of its work among nationals of the States Parties, have had a considerable impact on the number of cases brought every year (Council of Europe 2014, p. 7)."

The impact of the Court's success can be seen in Figure 1.1. Here the number of applications made every year since 2004 is shown. In recent years there have been fluctuations. However, this still an overall positive trend. Particularly, from the years 2004 to 2013, the number of applications increased by 103%.

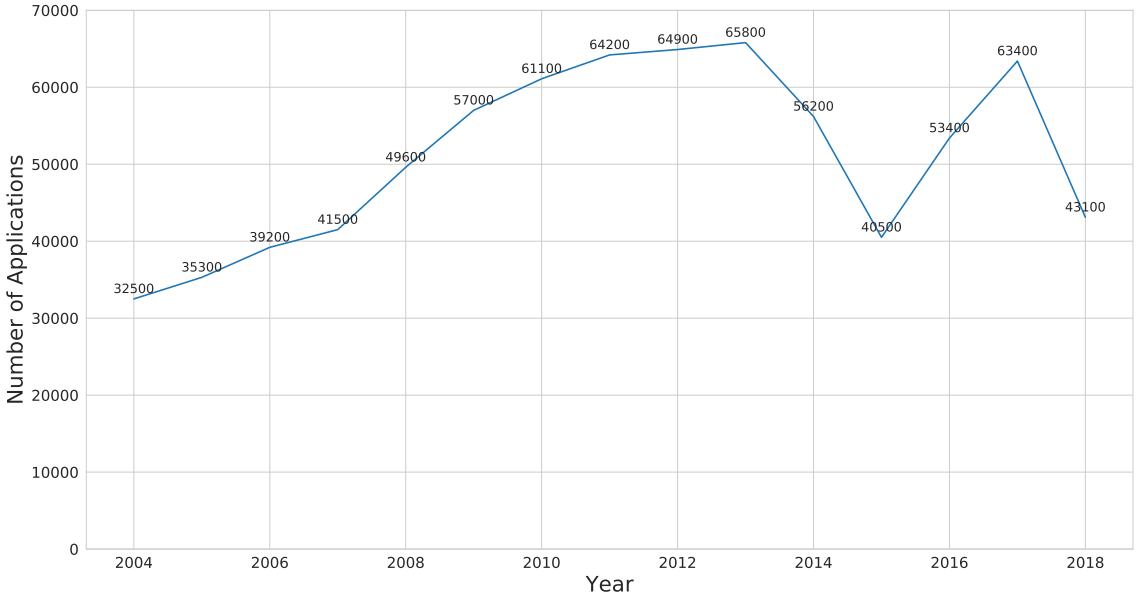

Figure 1.1: Number of Applications Made to the ECHR (Council of Europe 2019*a*)

1.2 Research Problem

The problem is that the large number of applications made every year has led to a backlog of applications. This has subsequently led to significant time delays in Court proceedings. According to the Council of Europe (2019*a*), the number of pending applications at the start of 2018 was 56250. A pending application is one that has not had an initial examination yet. By the end of 2018, the number of pending applications remained

relatively constant at 56350. Due to this backlog, applications can take up to a year before an initial examination can take place. After this examination, the application has to go through a further process before the Court can determine whether there was a breach of the Convention (Council of Europe 2016). Ultimately, it can take over a year for the ECHR to make a final judgement.

1.3 Research Question

How accurately can the judgements made by the European Court of Human Rights be predicted?

1.4 Research Objective

The research objective is to use a predictive model to help address the backlog of applications. The ECHR could use an accurate predictive model to make or help make judgements. Such a model could also be used to prioritise cases. That is cases which indicate a high likelihood of violation can be prioritised.

Chapter 2

Background

2.1 Court Procedure

According to the Council of Europe (2018), the ECHR has the following procedure:

1. An application is lodged by either an individual or State. A single application can involve multiple potential violations of Articles in the Convention.

2. The application then undergoes an initial examination to determine whether it is admissible. This decision is made by a single judge.

3. If an application is deemed admissible it will be allocated to one of three judicial formations: a single judge, a Committee (3 judges) or a Chamber (7 judges). The judicial formation chosen will depend on the type of case. In some circumstances, the application can be referred to a Grand Chamber of 17 judges.

4. Once allocated to a judicial formation, a verdict will be made for each of the potential Article violations.

5. Once a verdict has been made the ECHR produces a Judgment document.

In 2018, 42761 applications were decided by the ECHR. 40023 of these were by decision. This means they were deemed inadmissible at step 2 of the above procedure. The remaining 2738 were by judgement (Council of Europe 2019*a*). These applications go through the entire procedure and a final Judgment is produced. A Judgment is a legal document and the details of this document are discussed in the next section.

2.2 Judgment Structure

The contents of the Judgments are defined by the Council of Europe (2019*b*) in the Rules of the Court. The rules state that the Court should have, in the following order,

a "PROCEDURE", "THE FACTS", "THE LAW" and a verdict section. Throughout this study, these sections are referred to as procedure, facts, law and verdict respectfully. The facts section, seen in Figure 2.2, is commonly made up of two subsections: "I. THE CIRCUMSTANCES OF THE CASE" and "II. RELEVANT DOMESTIC LAW". Throughout this study, these subsections are referred to as circumstances and relevant, respectfully. An example of a procedure section can be seen in Figure 2.1. This section details the procedure followed by the ECHR throughout the applications life. An example of the law and verdict sections can be seen in Figure 2.3 and Figure 2.4 respectfully. The verdict section gives the judgement made by the ECHR. In Figure 2.4, we can see the Court investigated two potential violations of Article 6 and found no violations.

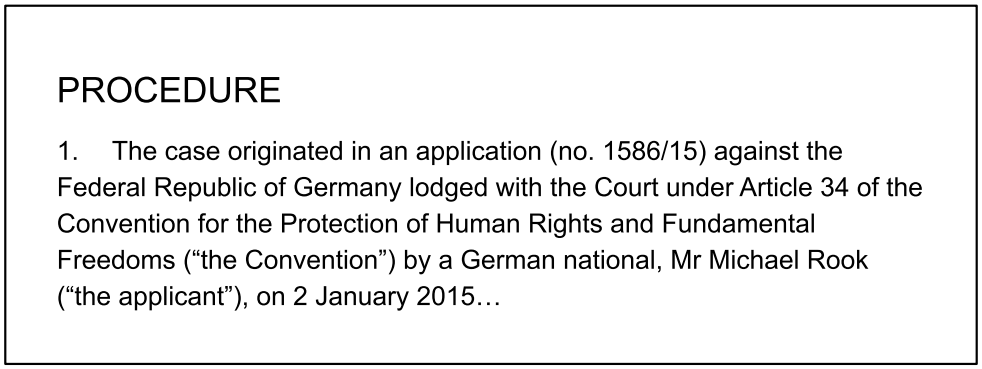

Figure 2.1: Example of Procedure Section of a Judgment (*Rook v. Germany* 2019)

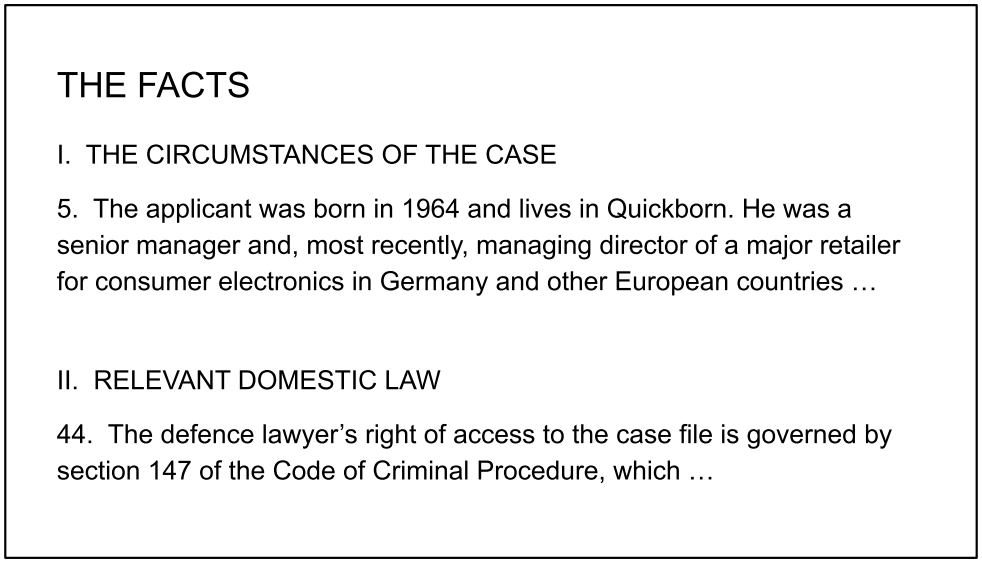

Figure 2.2: Example of Facts Section of a Judgment (*Rook v. Germany* 2019)

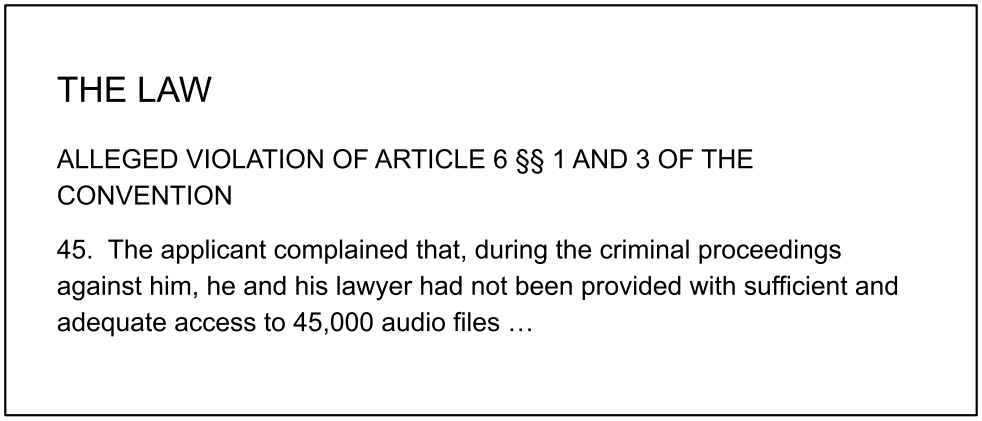

Figure 2.3: Example of Law Section of a Judgment (*Rook v. Germany* 2019)

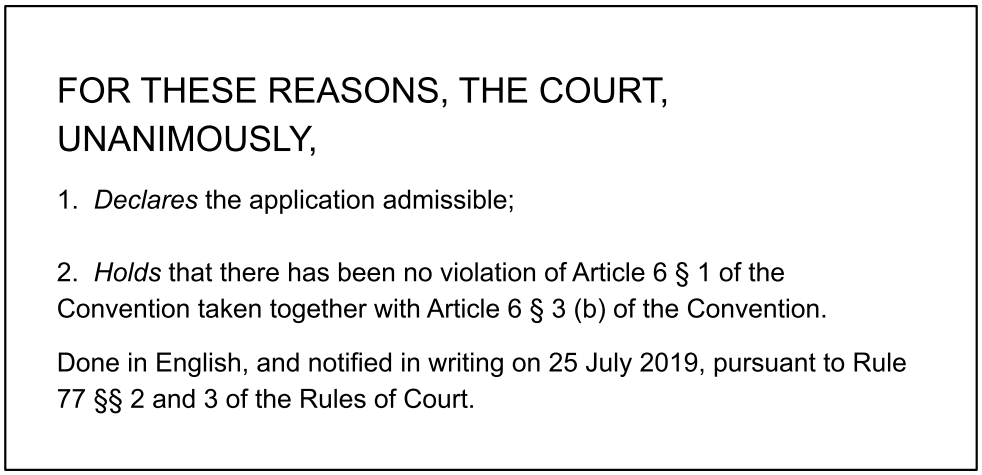

Figure 2.4: Example of Verdict Section of a Judgment (*Rook v. Germany* 2019)

Chapter 3

Related Work

The related work section focuses on studies that have used machine learning to predict the outcome of legal cases. The results of the studies can be seen in Table 3.1. The "Court" column gives the legal court considered by the study. The majority of the studies looked at either the ECHR or the Supreme Court of the United States (SCOTUS). The "Train. Acc." and "Test Acc." give the training and test accuracy, respectfully, achieved by the study. For some studies,the validation accuracy is not present. This is because it has not been reported by the researchers. Similarly, in some cases, the test accuracy is missing. This is because the researchers have not included a test set in their study. The "Data", "Target Variable" and "Algorithm" columns are explained in detail in the remaining sections of this chapter.

The first three studies in Table 3.1 looked at predicting the outcome of ECHR cases. At the time this study was done, these constituted all the previous works that aimed to predict the decisions made by this Court. Ultimately, they were scrutinised in more depth than the other studies in Table 3.1. This is because they are most relevant to the research problem. By analysing these other studies we can gain useful information on best practises when it comes to legal case prediction.

Table 3.1 includes only the studies that have constructed classification models to predict the outcome of legal cases. This is opposed to studies where the target variable is continuous such as in the study conducted by Bala et al. (2017). In this study, the researchers attempted to predict the time it would take for an insurance claim to be settled. A legal firm could use this model to avoid cases that would likely take longer to settle and reduce litigation costs. The outcome of ECHR judgements are discrete and so predicting the outcome is a classification task. In other words, the studies in Table 3.1 are most relevant to the research question. Subsequently, studies that look at predicting continuous target variables are less relevant.

Author	**Court**	**Data**	**Target Variable**	**Algorithm**	**Train. Acc.**	**Test Acc.**
Aletras et al. (2016)	ECHR	Case Documents	Violation, Non-Violation	SVM	80.1%	NA
Liu & Chen (2017)	ECHR	Case Documents	Violation, Non-Violation	SVM	79.5%	NA
Medvedeva et al. (2018)	ECHR	Case Documents	Violation, Non-Violation	SVM	75.0%	74.0%
Ruger et al. (2004)	SCOTUS	Summary Information	Affirmed, Reversed	Decision Tree	NA	75%
Guimer'a & Sales-Pardo (2011)	SCOTUS	Summary Information	Justice Decsion: Affirmed, Reversed	Stochastic Block Model	NA	83%
Katz et al. (2017)	SCOTUS	Summary Information	Affirmed, Reversed	Random Forest	NA	70%
Kaufman et al. (2017)	SCOTUS	Summary Information and Oral Arguments	Affirmed, Reversed	Random Forest	74.04%	NA
Agrawal et al. (2017)	US Circuit Court	Case Documents	Affirmed, Reversed	CNN	79%	NA
Agrawal et al. (2017)	SCOTUS	Case Documents	Affirmed, Reversed	Random Forest	68%	NA
Virtucio et al. (2018)	Philippine Supreme Court	Case Documents	Affirmed, Reversed	SVM	NA	62%
Sulea et al. (2017)	French Supreme Court	Case Documents	Rejet, Cassation, Irrecevabilite, qpc, Annulation, Non-lieu	SVM	96.9%	NA

Table 3.1: Summary of Previous Works

A broader body of research that looked at how machine learning can be applied to the legal industry, in general, was also consulted. This was to gain an understanding

of how machine learning can be used to address the research problem. In other words how machine learning models could be used to address the case backlog. Researchers explain that "the growth of big data, artificial intelligence, and machine learning will have important effects that will fundamentally change the way law is made, learned, followed, and practised (Alarie et al. 2016, p. 424)" . Some of the prominent research areas include legal prediction, legal analytics, document classifications and clustering and legal document generation (Surden 2014, McGinnis & Pearce 2013). Specifically, legal prediction models can benefit both lawyers and clients. By utilising these models, lawyers will be able to generate more accurate legal opinions in a shorter amount of time (Yoon 2016). For the ECHR, the implications of this are that legal prediction models could be used to decrease the amount of time it takes the Court to make decisions. This would ultimately have a positive effect on the case backlog and decrease time delays.

3.1 Datasets

As mentioned above, the "Court" column in Table 3.1 gives the legal court considered by the study. Besides the ECHR and SCOTUS, models were constructed for the United States (US) Circuit Court (Agrawal et al. 2017), the Philippine Supreme Court (Virtucio et al. 2018) and the French Supreme Court (Sulea et al. 2017). The data available to construct legal models differ for each Court. For example, ECHR documents are publicly available in the HUDOC database. Documents available include the Judgments discussed in the Background chapter and other documents such as decisions and legal summaries (*HUDOC database* 2018). These are all text-based data sources. In comparison, SCOTUS data, downloaded from the Supreme Court Database (SCDB), consists of different variables summarising the cases. For example, the variable "Lower Court Disposition", which can take on 12 different values, gives the decision made by the lower Court (Spaeth et al. 2018).

The "Data" column in Table 3.1 gives the type of data used in the study. "Case documents" refers to text documents that outline the cases heard by the Court. For example, the first three studies use the ECHR Judgment documents (Aletras et al. 2016*b*, Liu & Chen 2017, Medvedeva et al. 2018*b*). Similarly, Sulea et al. (2017) and Virtucio et al. (2018) used case documents. The majority of the studies that looked at SCOTUS decisions used "Summary Information" (Ruger et al. 2004, Guimerà & Sales-Pardo 2011, Katz et al. 2017, Kaufman et al. 2017). Where summary information are those variables available from the SCDB. There are some exceptions. Kaufman et al. (2017) used both summary information and oral arguments. Where oral arguments is textual data that detailed justices' oral arguments for each case. This source includes information such as the questions asked to petitioners and respondents.

Agrawal et al. (2017) are the only researchers to considered case documents when

trying to predict the outcome of SCOTUS decisions. Specifically, they use documents from lower courts predict the decisons of the higher courts. The researchers use case documents from District Court cases appealed to a Circuit Court to predict the outcome at the Circuit Court. Similarly, the researchers use Circuit Court case data to predict the outcome of the Supreme Court decision. The type of data available is important as it likely influences choices made in the study's Methodology. In particular, we'll see in the Feature Engineering section how the types of features that can be extracted depends on the available data.

We have mentioned that the studies looking at the ECHR have considered the Judgment documents. Specifically, Aletras et al. (2016*b*) considered only Judgments with the structure outlined in the Background chapter. That is the Judgments must have a procedure, facts, law and verdict section. The facts section must also consist of two subsections: circumstances and relevant. According to Aletras et al. (2016*b*), considering Judgments with the same structure simplifies the text-based analysis. We will see in the Feature Engineering section how this structure has helped in creating textual features. Subsequently, Liu & Chen (2017) and Medvedeva et al. (2018*b*) have also chosen to only consider Judgments with this structure.

It is not necessarily the case that Judgments must have this structure. Importantly, neither Aletras et al. (2016*b*), Liu & Chen (2017) nor Medvedeva et al. (2018*b*) have provided details such as what proportion of the Judgments have this structure. If only a small proportion of all Judgments have this structure, then the researchers would have significantly reduced the amount of data available to train models. Additionally, the Judgments used are not necessarily representative of all Judgments. That is there may be systematic differences between the Judgments with an without the structure. For example, we may find that the Judgments with the structure tend to have more violations than those with a different structure.

Another important detail is that the ECHR studies all construct individual models for the different Articles shown in Table 1.1. For example, Aletras et al. (2016*b*) consider a subset of ECHR Judgments which related to Articles 3, 6 and 8 of the convention. Subsequently, the researchers construct models to predict the outcome of judgements with respect to each Article. That is whether there is a 'violation" or "non-violation" of the Article. Aletras et al. (2016*b*) looked at only these Articles because they determined there to be an insufficient number of Judgments for the other Articles to construct machine learning models. Liu & Chen (2017) has used the same dataset as Aletras et al. (2016*b*) and so they have also only considered Article 3, 6 and 8. In comparison, Medvedeva et al. (2018*b*) considered 9 Articles, including Article 3, 6 and 8, of the Convention. Consequentially, we can expect the models constructed by Medvedeva et al. (2018*b*) to be more useful in practice. This is because the models can be used by the ECHR to make predictions for more Judgments.

For training, Aletras et al. (2016*b*), Liu & Chen (2017) and Medvedeva et al. (2018*b*) have all used balanced datasets. That is they have an equal number of violation and non-violation Judgments in each Article's training sets. In their final training sets, Aletras et al. (2016*b*) had 250, 80 and 254 cases for Articles 3, 6 and 8 respectfully. The number of Judgments for Article 6 is peculiar. In the Introduction, it was mentioned that the majority of the Judgments made by the ECHR involved Article 6 (Council of Europe 2014). This is not the case for the Judgments collected by Aletras et al. (2016*b*) and this suggests their dataset is not representative of all Judgments. In comparison, Medvedeva et al. (2018*b*) included 568, 916 and 458 Judgments in their training sets for Article 3, 6 and 8 respectfully. Consequentially, the researchers have trained their models on significantly more data. Specifically, their Article 6 training set has over 11 times more Judgments. There are still some issues with the dataset collected by Medvedeva et al. (2018*b*).

Medvedeva et al. (2018*b*) has made their dataset publicly available (Medvedeva et al. 2018*a*). Looking through their dataset reveals that they have included the Judgments that do not have the structure defined in the Background section. For example, they have included Judgment for the case of *Szerdahelyi v. Hungary* (2012) in their Article 11 training set. This Judgment has no relevant subsection. That is the facts section only consists of the circumstances subsection. It is not clear how many Judgments, with a different structure, have been included. Ultimately, we will see that including these Judgments may have negative consequences in the Feature Engineering section.

The issues with dataset collection could be a result of how data has been made available by the ECHR. It is not possible to download Judgments and other documents from the HUDOC database in bulk (*HUDOC database* 2018). This means researchers have to create their own tools to download the data which can be unreliable. For example, (Medvedeva et al. 2018*b*, p .8) states: "We used a rather crude automatic extraction method, so it is possible that a few cases might be missing from our dataset." This issue may be addressed by Quemy (2018). In their paper, the researcher presents the European Court of Human Rights Open Data project (ECHR-OD). This project aims to provide a complete dataset to aid machine learning and other data analysis. Ultimately, having a reliable source for the ECHR data will mean that any results presented by researchers are more reliable.

An important aspect of the machine learning process is to include a test set. Models can become biased towards the training set or, in other words, they have been over-fitted to the training set. So by including a test set we obtain an unbiased estimate of how well the models perform (Kuhn & Johnson 2013, p. 67). Additionally, to obtain a realistic estimate of the model's performance a realistic test set should be used. Where a realistic set is one where the target variables are in the same proportion to what we would expect in the future. For example, Ruger et al. (2004) trained their model using SCOTUS

cases before the 2002 term. This was done before the start of the 2002 term. Once the term started, the researchers tested their models on the cases, as they transpired, throughout the term. This is inherently a realistic test as the proportion of "Affirmed" and "Reversed" cases are the same as in reality. For the ECHR, a less elaborate way of obtaining a realistic test set would be to choose the set so that it had the same proportion of "violation" to "non-violations" as in the past. This is assuming that future Judgments will have a similar proportion.

In Table 3.1, we see that neither Aletras et al. (2016*b*) nor Liu & Chen (2017) have included a test set. Medvedeva et al. (2018*b*) has included a test set but it is not a realistic. As mentioned, Medvedeva et al. (2018*b*) has used a balanced training set for each Article. They obtained the largest training sets possible. For example, Article 6 has more violations than non-violations and so the training set, for this Article, contains all the non-violations. The remaining Judgments are used as the test set. Ultimately, what this means is that, depending on the Article, the test sets contain only either violation or non-violations and not both. This is not a realistic test set as we would not expect all future Judgments to have the same outcome. Consequentially, as far as we can tell, no study that looks at predicting ECHR judgements has used a realistic test set to evaluate their models. In terms of the research question, this means that we do not have a realistic estimate of how well machine learning models can predict the judgements made by the ECHR.

3.2 Baselines

The accuracy achieved on test sets should be compared to that of a baseline. This is because a baseline puts the model's accuracy into perspective. If a similar or higher accuracy can be achieved by a baseline it suggests improvements need to be made to the model. In machine learning, a baseline could be a simple model or a heuristic. For example, (Katz et al. 2017) used the heuristic of choosing the most frequent decision within a moving window of the last 10 years. That is if the SCOTUS made more affirmed decisions, within a 10-year window, then any test cases, within that window, would be predicted as being affirmed. This baseline had an accuracy of 65% on the test set. In comparison, their model achieved an accuracy of 70% which is 7.7% higher.

As an alternative Ruger et al. (2004) used the predictions of legal experts as a baseline. As mentioned above, the researchers used all the SCOTUS cases, as they transpired, throughout the 2002 term as a test set. At the same time, the researchers asked a panel of legal experts to predict the outcome of these same cases. The experts achieved an overall accuracy of 59.1%. In comparison, the model had a statistically higher accuracy of 75%. This baseline is effective as it allows us to compare the model's results to expert opinion. In other words, we are comparing the model to the best available predictions.

The results achieved by other studies could also be used as a baseline. For instance, Ruger et al. (2004) and Katz et al. (2017) both aimed to predict SCOTUS decisions. Looking at Table 3.1, we can see that Ruger et al. (2004) achieved an accuracy that was 7.14% higher. This suggests that the earlier work by Ruger et al. (2004) has obtained better results. However, other differences in the studies should be considered. For instance, the model constructed by Ruger et al. (2004) can only be applied to the same natural court. A natural court is a period during which no SCOTUS justices change. In comparison, Katz et al. (2017) could be applied over a much longer time period and not just one natural court.

Similarly, it is difficult to compare the approaches of the studies done on ECHR prediction. This is both due to the lack of realistic test sets and the fact that they have varying training sets. To best compare one approach or algorithm to another it is necessary to hold other factors constant. In other words, a standardised training and test set is needed. For example, the GLUE benchmark allows researchers to test NLP algorithms on standardised training and test sets across a variety of tasks. This allows for a more direct comparison between different algorithms. This highlights the need for the ECHR-OD discussed above, which will ultimately provide more standardise datasets for ECHR research (Quemy 2018).

In terms of baselines, all the researchers who looked at the ECHR have compared their training accuracy 50% (Aletras et al. 2016*b*, Liu & Chen 2017, Medvedeva et al. 2018*b*). This is the accuracy of a random guess given a balanced dataset. However, in reality, judgement outcomes are not perfectly balanced. For instance, the Medvedeva et al. (2018*b*) collected 720 cases for Article 2. 559 of these cases were label as "violation" and 161 as "non-violation". This means that 77.6% of Article 2 cases are labelled as a violation. If future cases have a similar distribution, we can expect the heuristic of always predicting violation to produce an accuracy close to 77.6%. In comparison, the training accuracy achieved by the researchers for Article 2 was only 73%. This suggests that a baseline of 50% is too low. It may have been appropriate to also compare the models results against the simple heuristic of always predicting the most common case outcome.

3.3 Target Variables and Algorithms

The "Target Variable" column in Table 3.1 gives the case outcome that the court is trying to predict. Most researchers have framed the problem as a binary classification task. This often requires a simplification. For example, Ruger et al. (2004) simplified SCOTUS decisions by labelling all votes to reverse, partly reverse, vacate or remand as "reversed". This led to a binary classification problem where cases are labelled as either "affirmed" or "reversed". This was also done in the other studies that looked at either the SCOTUS or US Circuit Court. The target variable in the study by Guimerà

& Sales-Pardo (2011) is slightly different. Here the researchers aimed to predict the outcome of individual justices' votes and not the Court's decision as a whole. Similarly, the studies that look at the ECHR have labelled Judgments as either "violations" or "non-violations". Sulea et al. (2017) were the only researchers that did not use a binary target variable. Their model was trained to predict 6 possible outcomes. Modelling a more complicated target variable, such as this one, could provide more practical value as such a model would be a better representation of reality.

To tune hyper-parameters, most of the studies in 3.1 have used k-fold cross-validation. The only exceptions are Ruger et al. (2004) and Guimerà & Sales-Pardo (2011) but they did not report their training accuracy. Consequentially, the accuracies reported in the "Train. Acc." column is the cross-validation accuracy achieved by the studies. Specifically, for the studies that look at the ECHR, Aletras et al. (2016*b*) and Medvedeva et al. (2018*b*) have used 10-fold cross-validation and Liu & Chen (2017) has used 5-fold cross-validation.

For each study, the algorithms that achieved the highest accuracy can be seen in the "Algorithm" column of Table 3.1. For some studies, such as Katz et al. (2017), Ruger et al. (2004) and Kaufman et al. (2017), only the algorithm shown in Table 3.1 was applied. In other studies, multiple algorithms were applied. For instance, Agrawal et al. (2017) applied Gradient Boosting, Multinomial Naive Bayes, Random Forest, Linear Support Vector Machine (SVM), Logistic Regression and Convolutional Neural Networks (CNN) algorithms. Subsequently, they found CNNs to performed the best for the US Circuit Court and Random Forests for the SCOTUS.

For all the studies that used case documents, except Kaufman et al. (2017), SVMs were found to produce the highest accuracy. Specifically, in all these studies Linear SVMs were used. For studies by Aletras et al. (2016*b*), Medvedeva et al. (2018*b*), Sulea et al. (2017) this was the only algorithm applied. The choice to use this algorithm may have followed from the prominence of SVMs and their success in a variety of text classification tasks (Joachims 1998, Liu et al. 2010). Virtucio et al. (2018) applied both Linear SVMs and Random Forests. Lastly, Liu & Chen (2017) applied 4 other algorithms besides a linear SVM. These are k-nearest neighbour (KNN), Logistic Regression, Random Forests, Bagging (Bootstrap aggregation) and SVM with a radial basis function (RBF).

This means only 5 classification algorithms have been applied to predicting the outcome of ECHR judgements. Additional algorithms, such as Naive Bayes Classifier, could be applied. Similarly, hyper-parameter tuning has been limited. For instance, Liu & Chen (2017) has applied a KNN algorithm using only 1 value for k (i.e. k = 5). Similarly, Medvedeva et al. (2018*b*) considered 3 values for the SVM penalty parameter: 0.1, 1 and 5. The exact values for the other algorithms in the study by Liu & Chen (2017) and for the SVM in the study by Aletras et al. (2016*b*) have not been reported. Ultimately, by applying additional algorithms and parameters the cross-validation of the models could

be improved.

3.4 Feature Engineering

Due to the nature of the data, the studies that used summary information required minimal feature engineering. For example, Katz et al. (2017) used features available directly from the SCDB such as Justice ID, Natural lower court disposition and issue area. This is common for the papers that looked at the SCOTUS (Katz et al. 2017, Ruger et al. 2004). Kaufman et al. (2017) used SCDB features but they also obtained features from the justices' oral arguments. Features were obtained using indicators (e.g. indicating if petitioners were asked more question than respondents) and ratios (e.g. the ratio of the number of questions asked to petitioners and respondents). This is one approach to structuring textual data. The researchers could have also used alternative NLP techniques. Such as, Agrawal et al. (2017) who obtained N-gram features from the case text for both the US Circuit Court and SCOTUS. N-gram features are also known as the bag-of-words model.

A similar approach has been used by Aletras et al. (2016*b*) to obtain features from the ECHR Judgments. Specifically, the 2000 most frequent N-grams of length 1 to 4 are obtained. Features are then obtained from the Judgments by determining the frequency that these N-grams occur within each Judgment section. To be clear, different sets of N-gram features are obtained from the different Judgment sections discussed in the Background chapter. A similar approach has been taken by Liu & Chen (2017) and Medvedeva et al. (2018*b*). In fact, Aletras et al. (2016*b*) have made their feature matrices available online (Aletras et al. 2016*a*). Liu & Chen (2017) have used these matrices to train their models without considering the original Judgments. Medvedeva et al. (2018*b*) has used their own dataset to obtain the N-gram features. This is where the inconsistencies in the dataset collected by the researchers are a concern. As mentioned, the Medvedeva et al. (2018*b*) have included Judgments with a different structure to that discussed in the Background chapter. Hence, it is unclear how the researchers have managed to obtain features for all the sections for each Judgment.

Virtucio et al. (2018) and Sulea et al. (2017) have also used N-gram features to train their models. Evidently, N-gram features are popular amongst the researchers that have used case documents. These features have their limitations. For instance, they do not consider the semantics of words. The word order of the legal documents is also lost Le & Mikolov (2014). As an alternative, word embeddings could be used to obtain features. The goal of word embeddings is to represent words as vectors to capture their semantics. Words with similar vectors have similar meanings (Li & Yang 2018). They have also shown to provide improvements in performance for a variety of text classification task (Lilleberg et al. 2015, Ge & Moh 2017).

Features could be obtained using pre-trained embeddings such as the GloVe embeddings trained by Pennington et al. (2014). Alternatively, the pre-trained legal embedding, law2vec, could be used. These embeddings have been trained on a variety of legal document (Chalkidis & Kampas 2019). As a result, these embeddings could have captured the legal semantics of words better than the GloVe embeddings which were trained using more general documents. Another option would be to train a new word embedding using the documents obtained from the ECHR. This is one way of incorporating additional ECHR data, not just Judgments, into the machine learning models.

Word embeddings may provide an advantage to the N-gram model but they also do not consider the word order of documents. Paragraph embeddings could be used as an alternative. These are similar to word embeddings but they consider the word order of documents (Le & Mikolov 2014). Word order may be an important factor. A paragraph embedding model could, similarly, be trained using the documents available for the ECHR. An additional advantage of both word and paragraph embeddings is that they can be used to represent Judgments using vectors of a smaller dimension than N-gram features (Li & Yang 2018). For instance, both 100 and 200 dimension law2vec embeddings are available (Chalkidis & Kampas 2019). This is opposed to the 2000 dimension vectors obtained using N-grams.

Aletras et al. (2016*b*) has used features called "topics" to reduce the feature dimensionality. These are essentially features created using clusters of N-grams. The clusters are obtained using the N-gram feature matrices and spectral clustering (Von Luxburg 2007). The reproducibility of these features may be an issue. This is because the researchers have omitted some key information needed to obtain the features. Firstly, Aletras et al. (2016*b*) did not mention from what section of the Judgments the topics were obtained. Secondly, the researchers have omitted important technical details. They mention that they follow a tutorial on spectral clustering provided by Von Luxburg (2007). However, this tutorial requires you to make various technical decisions, such as what algorithm is used to obtain the "similarity graph", before obtaining the final clusters.

A final consideration for feature engineering is data leakage. Data leakage is when data, that could only be obtained after an outcome is known, is used to train models. That is textual features should not be obtained using any text that could only be obtained after a verdict has been made because, in reality, this text would not be available to make predictions. Doing so would likely lead to higher prediction accuracies than could be achieved otherwise (Nisbet et al. 2009, p. xv). For example, in Table 3.1 we see that Sulea et al. (2017) achieved a training accuracy of 96.9%. This is 21% higher than the second-best training accuracy, 80.1%. The researchers addressed the high accuracy in their paper. One potential reason is that the French Supreme Court is predictable as it offers judges less freedom when it comes to interpretation (Sulea et al. 2017). Another potential reason is that their method of removing text, that could only be obtained after

a verdict, from cases is flawed. Specifically, the researchers stated:

> In the future we would like to investigate more sophisticated methods of masking features in the original text data that explicitly list and "give away" the desired target prediction to simulate realistic application scenarios, where text classification predicts the target features from "draft" case descriptions that do not yet contain the target predictions (Sulea et al. 2017, p .6).

The inclusion of text features that directly relate to the case outcome in the model would explain the high accuracy. This is because a machine learning model would identify these features and give them a high weighting. Hence, this mistake should be avoided. For the ECHR Judgments, the verdict contains the final outcome of the case. Hence, textual features should not be obtained from this section. Additionally, the law section also contains information that could only be known after a judgement has been made. In previous work, researchers have taken different approaches when it comes to the law section. Aletras et al. (2016*b*) used regular expressions to remove the text that, could be directly related to the outcome, from the section. Medvedeva et al. (2018*b*) took an alternative approach and simply excluded this section.

Chapter 4

Methodology

The study builds onto the work done by Aletras et al. (2016*b*), Liu & Chen (2017) and Medvedeva et al. (2018*b*). Particular aspects of the methodology in this study have been motivated by the choices these researchers have made. This is because their approach has proved to be a good initial step in addressing the research problem. It will also allow for a better comparison of the results to those of previous studies. Differences in the methodology include the introduction of new data, new textual features and an automated model selection process. By keeping other aspects of the methodology the same, the effect of these changes can be better understood.

4.1 Data Preparation

Obtaining the ECHR Documents

All ECHR documents are publicly available in the HUDOC database (*HUDOC database* 2018). The database, however, does not provide the functionality to download documents in bulk. To avoid the time-consuming process of downloading documents manually, documents were obtained using an API provided by vizlegal. vizlegal is a legal technology company that specialises in legal search (vizlegal 2019). The API allows HTTP requests to be made to vizlegal's server were all the ECHR documents have been processed and stored. This makes it simple to automatically download all the documents. In previous studies, researchers built their own scraping tools to obtain the data (Aletras et al. 2016*b*, Medvedeva et al. 2018*b*). However, these have not been made available by the researchers. Other tools such as echr-scraping , created by van der Heijden & Kapfer (2016) were research. The repository for this tool has not been updated in three years and there is minimal documentation. Ultimately, the vizlegal API was used as it was the most reliable and efficient method of obtaining the data that could be found.

The number of documents downloaded using the API are shown in Figure 4.1. 14071 Judgment documents were obtained. Besides these, additional documents were also ob-

tained. Decisions make up approximately 40% of all the documents. According to the Council of Europe (2017), these documents give the rulings on the admissibility of applications. Communicated cases describe the communications that took place with the State responding to an application. Legal summaries are summaries of important judgements or decisions and the resolution documents describe proposals made by the Court (Council of Europe 2017). The other documents include Reports and Advisory Opinions. In total, 56688 documents were obtained. The Judgments are the primary data source as this study attempts to predict the outcome the judgements detailed in these documents. The other documents will still be incorporated. This is through the process of creating word and paragraph embeddings which are described in more detail in the textual features section of the Methodology.

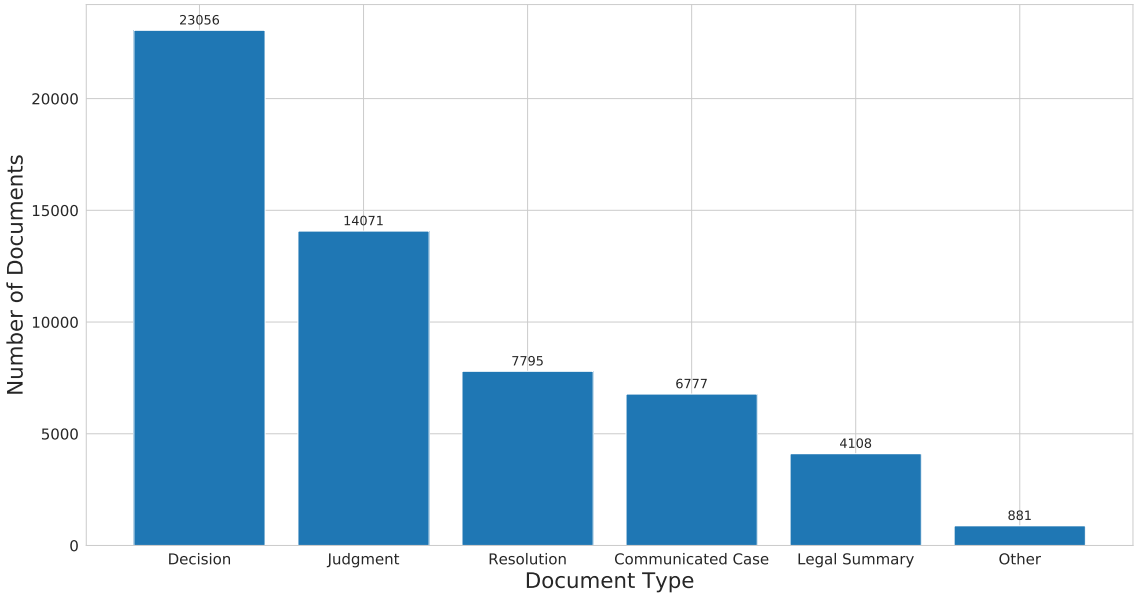

Figure 4.1: Number of ECHR Documents

Obtaining the Dataset

Only Judgments with the structure outlined in the background section are used to train models. That is they must have a procedure, facts, law and verdict section. The facts section must also consist of two subsections: circumstances and relevant. 9703 of the 14071 Judgment have this structure. The difference in the structure of 98% of the remaining 3970 Judgments is due to variations in the facts section. That is the section has either only one of the subsections, none of the subsections or the subsections are combined into one subsection. For some Judgments, such as *Beck, Copp and Bazeley v. The United Kingdom* (2002), the facts section is missing entirely. The remaining 2% consist of Judgments of a variety of different structures. For instance, *Belziuk v. Poland* (1998) has no law section. The downside to this decision is that the models are trained using only

69% of the available Judgment documents. The choice was made as all previous papers discussed have decided to used only Judgments of this structure (Aletras et al. 2016*b*, Liu & Chen 2017, Medvedeva et al. 2018*b*). This is because a standard structure for the Judgments simplifies the process of cleaning and extracting different textual features from the documents.

The 9703 Judgments are put into groups based on what Articles the Judgments address. The Judgments are then labelled according to the target variable described in Table 4.1. A single Judgment can involve multiple potential violations for the same Article. Hence, within an Article group, a Judgment is labelled as a violation if there is at least 1 violation for that Article. A single Judgment can also address potential violations for multiple different Articles. For example, within the same Judgment, the ECHR could rule that a State is in violation of Article 6 and in non-violation of Article 3. Hence, a single Judgment can appear in multiple Article groups and have different labels in each group. Ultimately, the problem has been framed as a binary classification problem with respect to each Article. Separate models are subsequently constructed to predict the outcome for each of the Articles.

Code	**Outcome**	**Description**
0	Non-Violation	There were no violations for that Article
1	Violation	There was atleast 1 violation for the Article

Table 4.1: Target Variable Code

The number of Judgments, by outcome, for each Article can be seen in Figure 4.2. Article 6 has the most Judgment documents: 3912 violations and 560 non-violations. In comparison, Articles such as 18 and 11 have significantly fewer Judgment documents. Specifically, Article 18 has less than 1% of the number of Judgments used for Article 6. For most of the articles, there is an imbalance between the number of violations and non-violations labels where there tend to be more violations. For instance, Article 3 has approximately 6 times more violations than non-violation. For each Article, Judgments are divided into a training and test set. During this process, the Judgments are randomised to avoid confounding with factors such as the date the Judgment was made.

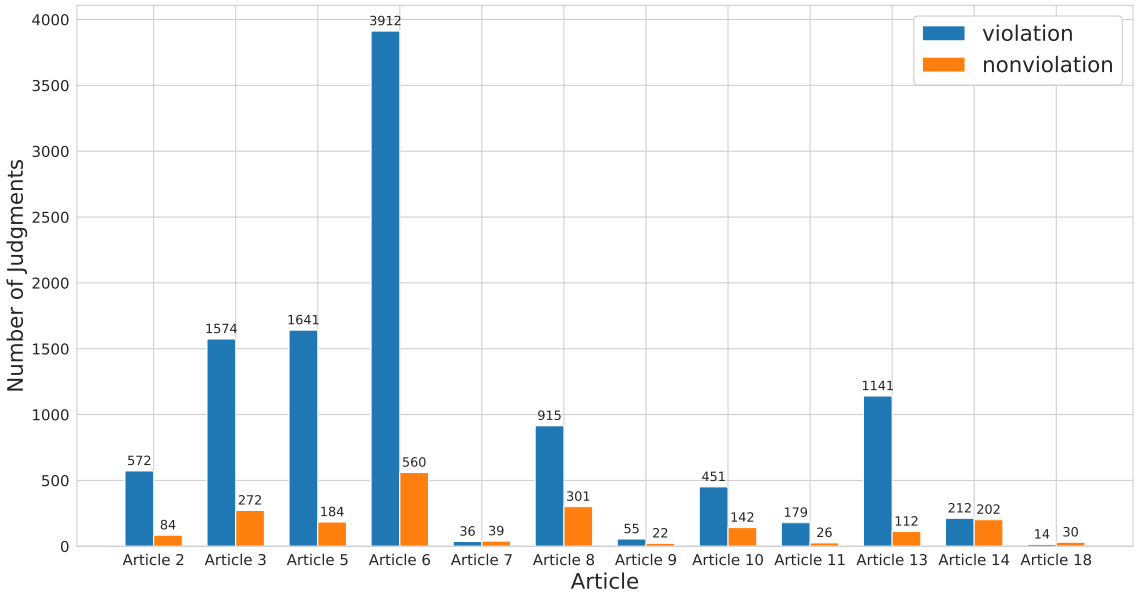

Figure 4.2: Number of Judgments in Dataset

A balance training set is selected. This is to avoid models becoming biased towards one of the outcomes. Additionally, previous studies have used balanced training sets and so a better comparison of training accuracies can be made. Realistic test sets were selected. That is, they were chosen so that they had the same violation to non-violation ratio as seen in the past. To be clear, they have the same ratio as the original set of Judgments and not the Judgments with the structure defined in the Background chapter. For example, Article 5 had 1959 violations and 210 non-violations. After selecting only those documents with the structure described, the number of violations and non-violation were 1641 and 184 respectfully. The test set is chosen to have a ratio of 1959:210 or 9.3:1. The sets are chosen in such a way as to maximise the number of training documents while still having at least 10% of each label in the test sets. The final number of Judgments in the training and test set for each article can be seen in table 4.2. We see for article 5 the number of non-violations in the test set is 18 which is 10% of 184. The ratio of violations to non-violations in the test set is 168:18 or 9.3:1.

	Training Set		Testing Set	
Article	**Violation**	**Non-violation**	**Violation**	**Non-violation**
Article 2	76	76	58	8
Article 3	245	245	175	27
Article 5	166	166	168	18
Article 6	504	504	539	56
Article 7	32	32	4	5
Article 8	271	271	93	30
Article 9	20	20	5	2
Article 10	128	128	45	14
Article 11	23	23	14	3
Article 13	101	101	138	11
Article 14	182	182	20	20
Article 18	13	13	1	2

Table 4.2: Number of Judgments in Training and Testing Sets

The number of Judgments in the training set can be compared to that of previous studies. The training sets' sizes are significantly larger than those used by Aletras et al. (2016*b*) and Liu & Chen (2017). On average, 4.5 times as many Judgments were used for each Article. In particular, 12.6 times as many Judgments were used for Article 6. Compared to Medvedeva et al. (2018*b*), 8.1% more Judgments were used across all the Articles on average. Even with the additional Judgments, some of the Articles have a small number of Judgments in their training sets.

It may be difficult to construct reliable machine learning algorithms for those Articles with relatively few Judgments. Previous studies have suggested a minimum threshold of 100 Judgments to train the models (Medvedeva et al. 2018*b*). This would exclude Articles 7, 9, 11 and 18. However, other researchers have trained models using fewer Judgments. Specifically, we saw in the Background section that Aletras et al. (2016*b*) used 80 Judgments for Article 6. So, the threshold of 100 Judgments is considered but we have still attempted to model the outcome of all the Articles. The results of the models for Articles 7, 9, 11 and 18 where analysed with the small training data sets in mind.

4.2 Feature Engineering

Before predictive models can be constructed, the Judgment text must be processed and features must be created from this text. From each Judgment, the text form the procedure, facts, circumstances and relevant sections are obtained. A combination of the

text from the procedure and facts sections (procedure+facts) is also created. To avoid data leakage, the text from the verdict section is not used. As mentioned in the Related Work, text from the law section also contains information that would only be available after a judgement is made. It was also mentioned that Aletras et al. (2016*b*) used regular expressions to remove this text. After doing so their law section models achieved an average accuracy of 62% which is 21.5% lower than their highest accuracy. On the other hand, Medvedeva et al. (2018*b*) simply excluded the entire law section. For simplicity and because the law section produced relatively low accuracies, this section was also not used to create predictive models.

Next, the text from each section is cleaned by making it lower case and removing all punctuation and numbers. A version of this text with stop-words removed is also obtained. The set of English stop-words provided by the NLTK python package was used (Loper & Bird 2002). In the end, 10 different variations of the text are obtained for each Judgment (2 stop-word variations and 5 case sections). This means we have 10 different corpora of cleaned Judgment text for each Article. Each corpus is divided into a training and test set. Different textual features are subsequently created using these different corpora.

N-grams

The NLTK Python package is used to create N-gram feature matrices using the cleaned Judgment text (Loper & Bird 2002). From the Judgments in the training set, the 2000 most frequent N-grams of length 1 to 4 are obtained. We only consider the N-grams from the training set as, in a realistic scenario, the Judgments in the test set would not be available at the time models are trained. Including the N-grams from the test set would be considered data leakage. This is because we would be incorporating information into the models that would not be available before a prediction is made. Using the training set N-grams, the Judgments are vectorised to obtain feature matrices for both the training and test sets. An example of the resulting matrix can be seen in Figure 4.3. Here, the rows represent the individual Judgments and the columns give the frequency of the particular N-gram for that Judgment. For instance, the N-gram 'yet' occurs 3 times in Judgment 1. For each Article, this is done for each of the 10 different corpora.

	abduction	ability	able	abroad	···	years imprisonment	years old	yet	zone	target
0	0	0	0	0	···	0	0	0	0	0
1	2	0	3	0	···	1	0	3	0	0
2	0	1	2	1	···	0	0	1	0	0
3	0	0	0	0	···	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
538	0	0	0	0	···	0	0	0	0	1
539	1	0	0	0	···	2	0	1	0	1
540	0	7	0	0	···	3	0	2	0	1
541	0	0	0	0	···	0	0	0	0	1

Figure 4.3: N-gram Feature Matrix Example

These feature matrices are then normalised using Min-Max feature scaling. Each value of a column is scaled using the following equation:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{4.1}$$

Here, for the given column, X is the original value and X' is the scaled value. X_{min} and X_{max} are the column's minimum and maximum values from the training set. The same minimum and maximum values are used to scale the testing matrix. Again, this is because in a realistic scenario the Judgments in the test set would not be available at the time the minimum and maximum values were determined.

Word Embeddings

N-grams are useful rudimentary textual features but they do have some disadvantages. For instances, they do not capture the semantics of words in the dataset. That is, they do not tell us anything about how similar words are to each other. By representing words as a vector, word embeddings can capture the semantics of words. Additionally, embeddings can also help to reduce the dimensionality of the features used in machine learning models (Li & Yang 2018).

The different word embeddings used are summarised in Table 3. 'Corpus' gives the documents that are used to train the word embeddings and 'No. Tokens' are the number of lower case words that make up the corpus. The tokens are used to train the word embeddings. The 'vocabulary size' is the number of words that have vector representations for that embedding. The GloVe and law2vec embeddings were trained by other researchers and the echr2vec embeddings were trained specifically for this paper. For each embedding, a 100 dimension and 200 dimension version are used. Where, the dimension is the embedding vector size. The reason for this is that the law2vec embeddings were only available in these dimensions. The same dimensions were used for the other embeddings so that a better comparison of the embeddings could be made. That is, any differences

in model accuracies could be attributed to the corpus used to train the embeddings and not the dimension.

The GloVe embeddings were trained by Pennington et al. (2014). The researchers created the embeddings using the GloVe algorithm discussed in their paper. For training data, they used the Gigaword 5 and Wikipedia 2014 dumps which provided 6 Billion tokens. They limited the vocabulary size to 400,000 of the most frequent tokens. On the other hand, the law2vec and echr2vec embeddings both used the word2vec embedding algorithm (Mikolov et al. 2013). Hence, It would have been preferable to use the word2vec algorithm instead of the GloVe algorithm. That is, use word2vec embeddings trained on a similar corpus to the one used for the GloVe embeddings. However, such embeddings with both a 100 and 200 dimensions version could not be found.

The law2vec embeddings were trained by Chalkidis & Kampas (2019). The researchers used a corpus of 123,066 legal documents from the UK, EU, Canada, Australia, USA, and Japan. These provided 492 Million tokens and embeddings with a vocabulary size of 169439. The law2vec embeddings were trained using roughly 8.2% of the number of tokens used for the GloVe embeddings. This may mean that the law2vec embeddings would not perform as well when used for text classification problems. However, the law2vec embeddings, having been trained on legal documents only, may better capture the semantics of words in a legal context.

Following from this, the semantics of words may differ depending on the area of law or the country the legal documents come from. The ECHR documents were not included in the law2vec corpus and so an additional embedding, echt2vec, using these documents were created. When creating the embeddings, all of the 56688 ECHR documents obtained were considered. To avoid data leakage it was necessary to exclude certain documents or sections of documents. For any Judgment in a training set, we exclude the law and verdict section. For any Judgment in a testing set, the entire Judgment was excluded. Additionally, the summary of any of the above Judgments was excluded. This is because the summary also contains details of the judgement made by the Court. Ultimately, this means that the echr2vec embedding was not trained on any text that would not be available before a judgement was made.

The embedding was created using these documents and the gensim implementation of the word2vec model (Řehůřek & Sojka 2010). A 5-word window and a minimum threshold of 10 occurrences was used. This is because these are the same parameters used to create the law2vec embedding. In the end, the documents provided 84 Million tokens and produced embeddings with a vocabulary size of 47587 words. Again, these embeddings were trained using fewer tokens than the others but they may have better captured the semantics of the words used in the ECHR documents. The echr2vec embeddings are also an attempt at incorporating the remaining documents available for the ECHR, and not just Judgments, into the models.

Embedding	Corpus	No. Tokens	Vocabulary Size
GloVe	Gigaword 5 and Wikipedia 2014	6 Billion	400,000
law2vec	123,066 legislation documents	492 Million	169,439
echr2vec	ECHR Judgment documents	84 Million	47,587

Table 4.3: Summary of Word Embeddings

Average Embedding Values

Word embeddings allow us to represent words as vectors. For this problem, predictions are made using entire documents. In other words, it is necessary to represent the documents as vectors. A common way of doing this is by finding the average of the word embedding vectors. This process is shown in Figure 4.4. We start with a section of the Judgment shown in the first block. In step 1 each word, of a cleaned version of this section, is cycled through. Each word is replaced by a vector representation of the word, shown in step 2. Then, in step 3, the average all these vectors are taken. This average vector is the vector representation of the Judgment. The vector representation for each word will differ depending on the embeddings used. For some embeddings, certain words will not have a vector representation. For example, in Figure 4.4, we see that 'quickborn' is replaced with 'none'. This is because the embedding does not have a representation of the word 'quickborn'. This word is consequentially not considered in the document mean vector.

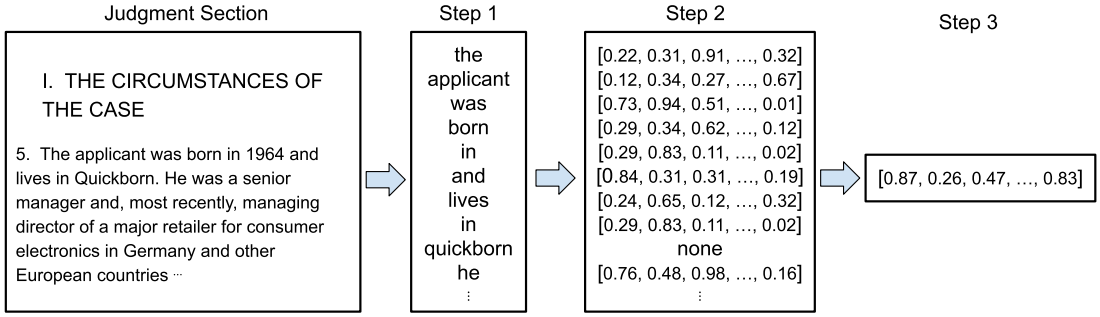

Figure 4.4: Average Word Embedding Process

This is done for both the training and testing sets. An example of a feature matrix obtained from this process is shown in Figure 4.5. As before, the rows represent each Judgment. The columns represent each element of the Judgement vector. In this example, a word embedding of dimension 100 has been used and so the corresponding average word embedding is also of dimension 100. As with the N-gram matrices, Min-Max feature scaling is used to normalise these feature matrices. For each Article, this process is repeated for each of the 10 corpora and using both the 100 and 200 dimension versions of each of the 3 word embeddings in Table 4.3.

	0	1	2	3	···	96	97	98	99	target
0	0.630378	0.284068	0.302572	0.674881	···	0.490479	0.539651	0.423404	0.272755	0
1	0.818343	0.286323	0.317915	0.405087	···	0.566218	0.524251	0.799147	0.539372	0
2	0.554219	0.390814	0.694171	0.656655	···	0.456615	0.298366	0.471229	0.557426	0
3	0.564305	0.239607	0.349915	0.556049	···	0.504377	0.469037	0.53324	0.375996	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1004	0.778099	0.384332	0.73037	0.677611	···	0.335773	0.501317	0.43763	0.655851	1
1005	0.482419	0.468934	0.462533	0.436581	···	0.40751	0.077488	0.390037	0.50174	1
1006	0.403575	0.692856	0.821801	0.718761	···	0.499702	0.553085	0.47815	0.378926	1
1007	0.458592	0.481098	0.326912	0.859364	···	0.368788	0.38387	0.72615	0.875544	1

Figure 4.5: Example of Average Word Embedding Feature Matrix

Doc2vec Embeddings

A downside to using the average of word embeddings to represent documents is that the document's word order is lost. Therefore, as an alternative, a doc2vec model has also been used to represent the Judgments as vectors. This model considers both the semantics of words as well as the word order of the Judgments (Le & Mikolov 2014). The gensim implementation was used to train the doc2vec model (Řehůřek & Sojka 2010).

The same documents used to train the echr2vec embeddings were used. That is, excluding the same documents and sections to avoid data leakage. Certain hyper-parameters have to be set to train the doc2vec model. For a better comparison to the average word embeddings, doc2vec models of dimension 100 and 200 are trained. The other hyper-parameters were informed by the work done by Lau & Baldwin (2016) where the researchers trained doc2vec models using a variety of corpora. Specifically, they used a window size of 15, a minimum word frequency of 10 and 20 training epochs. Judgment vectors are inferred using 20 epochs. These same parameters were used to train the doc2vec models in this paper. This resulted in feature matrices similar to the one shown in Figure 4.5. Only now, for columns give the doc2vec embedding representation of the Judgment. Again, these feature matrices are normalised using Min-Max feature scaling. For each Article, this is done for all 10 corpora.

4.3 Modelling

4.3.1 Classification Models

Using the textual features discussed above, machine learning models are constructed for each of the Articles. This was done using of the auto-sklearn python package. This package uses efficient Bayesian optimization methods to automate algorithm and hyper-parameter selection (Feurer et al. 2015). It is based on the scikit-learn machine learning framework (Pedregosa et al. 2011) and it considers 15 different algorithms, including

Linear SVM, Gradient Boosting and Random Forest. The number of hyper-parameters considered depends on the algorithm. For instance, the package selects the values of 4 different hyper-parameters for the Linear SVM algorithm including the penalty parameter and type of penalty: l1 or l2. 10-fold cross-validation is used to select both the algorithm and associated hyper-parameters. Ultimately, this package provides an alternative to grid search. Using the package, a wider range of models and parameters can be tested than tested in previous papers (Feurer et al. 2015).

As discussed in the Feature Engineering section, multiple different feature matrices are obtained for each Article. For example, feature matrices are constructed using both GloVe and law2vec embeddings. In terms of modelling, these differences can be considered hyper-parameters. All these hyper-parameters are shown in Table 4.4. Ultimately, varying each of these hyper-parameters results in a different feature matrix for an Article. A distinction should be made between these hyper-parameters and those selected by the auto-sklearn package. The auto-sklearn package can only be used to select the algorithm and the algorithm's associated hyper-parameters. Hence, for each of an Article's feature matrices, auto-sklearn is used to find the classification algorithm and associated hyper-parameters that maximises cross-validation accuracy. Then, all these cross-validation accuracies are compared to obtain the model with the highest overall cross-validation accuracy.

Ultimately, at the end of this process we will have one model for each Article. This model would have been trained using one combination of the hyper-parameters in 4.4. The classification algorithm and it's associated algoithm would have been selected by the auto-sklearn package. For each Article, this model is then re-trained using the entire training set and used to make predictions on the test set. This is to provide an estimation of how well the model performs on a realistic out-of-sample data set.

Hyper-parameter	**Values**
Feature Type	N-gram, GloVe,,law2vec, echr2vec, doc2vec
Dimension	100, 200 and 2000 (for N-gram only)
Judgment Section	procedure+facts ,procedure, facts, circumstances, relevant
Stop-words	Yes, No

Table 4.4: Model Hyper-parameters

To train models, the auto-sklearn time parameters "time_left_for_this_task" and "per_run_time_limit" have to be set. These give the maximum amount of time, in seconds, dedicated overall to fitting models and the maximum amount of time dedicated to each classification algorithm. Essentially there is a payoff, between the time it takes the package to fit models and the overall cross-validation accuracy. Not giving the package enough time would

mean that the optimal model and parameters may not be found. To choose the parameters some initial experimentation was done. 10 random feature matrices were taken, and the package was applied using variations of the time parameters. It was found that values of 360 and 120, respectfully, were appropriate. Increasing the time value's further resulted in an average increase in the cross-validation accuracy of less than 0.1%.

The package can also be used to automatically select data and feature preprocessing methods. The data preprocessing methods include aspects such as how missing data is imputed and how features are scaled. Feature preprocessing change sthe dimensionality of the features through methods such as PCA (Feurer et al. 2015). These aspects of the package are not used in this study as they have already been covered during the feature engineering process.

A downside to using the auto-sklearn package is that the individual cross-validation fold accuracies are only calculated internally. Only the average accuracies across all the folds are given as output. Hence, confidence intervals for the cross-validation accuracy cannot be calculated. This means that we cannot conclude statistically whether the difference in cross-validation accuracies of the two models are statistically significantly. Ultimately, we can only compare models base on the cross-validation accuracies.

4.3.2 Heuristic

The models' results are also compared to a simple heuristic. That is, the heuristic always predicts the outcome of the Judgment to be the outcome that was the most common in the past. So, in the case of Article 6, we saw there were more violations a non-violations. Each Judgment in the test set for Article 6, will consequentially be predicted as a violation by the heuristic. It is important to include such a baseline as it puts the model results in perspective. If the models are outperformed by a simple heuristic they are likely not going to be useful.

4.3.3 Metrics

Prediction results will be analysed using accuracy, precision and recall. The confusion matrix in Table 4.5 is used to explain how these metrics are calculated. Here, the values for false negative (FN) and false positive (FP) give, for a given Article, the number of Judgments that are incorrectly predicted as non-violation and violation respectfully. Similarly, the values for true negative (TN) and true positive (TP) give the number of Judgments correctly predicted as non-violations and violations respectfully.

		Prediction	
		Non-violation	Violation
Actual	Non-violation	True Negative (TN)	False Postive (FP)
	Violation	False Negative (FN)	True Positive (TP)

Table 4.5: Confusion Matrix

Equation 4.2 gives the formula for the accuracy metric. For a given Article, this gives the proportion of Judgment outcomes that were correctly predicted. Accuracy is calculated for each Article and for both the models and heuristic. To compare the overall accuracy, the weighted average, in Equation 4.3, is calculated for both the models and heuristic. Here $Accuracy_i$ is the accuracy achieved for Article i and w_i is the number of Judgments in the test set for Article i. The weighted average is used as it has a clearer interpretation than the average. The test sets have the same Article balance as past Judgments. The weighted average, therefore, gives the estimate of how many correct predictions the model (or heuristic) would make in the future. This is assuming future Judgments have the same Article balance as the past.

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \tag{4.2}$$

$$Weighted\ Average = \frac{\sum_{i \in Articles} Accuracy_i * w_i}{\sum_{i \in Articles} w_i} \tag{4.3}$$

The formula for precision and recall are given by Equations 4.4 and 4.5 respectfully. For a given Article, precision gives the proportion of all violation predictions that were correct. Recall, on the other hand, gives the proportion of actual violations that were correctly predicted. Precision and recall are only calculated for the models and not the heuristic. This is because not much insight can be gained for the heuristic. Due to the nature of the heuristic, if violations were most common for a given Article, precision will equal the accuracy and recall will equal 1. If non-violations are most common then both the precision and recall will be 0.

$$Pecision = \frac{TP}{TP + FP} \tag{4.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.5}$$

As mentioned in the Related Work, none of the previous studies has used a realistic test set. Consequentially, the test set accuracies achieved in this study cannot be compared to previous studies. Hence, the cross-validation accuracies for each of the Articles will also be given. The cross-validation accuracies will subsequently be compared keeping in mind the differences between the studies.

Chapter 5

Results and Discussion

5.1 Model Results

The hyper-parameters and classification algorithm that achieved the highest cross-validation accuracy for each Article can be seen in Table 5.1. The "Feature Type", "Dimension", "Section" and "Stopwords" parameters are discussed in the modelling section of the Methodology chapter. The "Classifier" is the classification algorithm that was selected by the auto-sklearn package. We can see that the classifier used varied for each Article. Quadratic Discriminant Analysis (QDA) and Random Forest were the two most common algorithms. They were each chosen for 3 of the Articles. Additionally, Gradient Boosting, Stochastic Gradient Descent (SGD), Decision Tree and AdaBoost classifiers were used. We can see that the "Feature Type", "Dimension" and "Part" also varied by Article. These hyper-parameters are analysed in more depth in sections 5.2 and 5.3.

In the Related Work chapter, we saw that a linear SVM classifier produced the highest cross-validation accuracy for each Article. Either the researchers only tested linear SVMs (Aletras et al. 2016*b*, Medvedeva et al. 2018*b*) or a linear SVM showed to produce the highest accuracy for all algorithms tested (Liu & Chen 2017). Looking at Table 5.1, we can see that, in this study, a linear SVM did not produce the highest cross-validation accuracy for any of the Articles. This is important as it suggests that, to improve accuracy, it was necessary to test additional classification algorithms. However, the results in Table 5.1 do give any indication of the statistical significance of the difference between the cross-validation accuracies of the algorithms tested. For example, for Article 3 we found that a Random Forest classifier produced the highest cross-validation accuracy. If compared to the accuracy achieved by a linear SVM we may find that difference is not statistically significant. In other words, it is not possible to say with certainty that the random forest classifier produced the highest accuracy.

Article	**Feature Type**	**Dimension**	**Section**	**Stopwords**	**Classifier**
Article 2	law2vec	100	circumstances	Yes	Gradient Boosting
Article 3	GloVe	200	procedure+facts	No	Random Forest
Article 5	GloVe	200	relevant	Yes	Gradient Boosting
Article 6	echr2vec	100	procedure+facts	Yes	SGD
Article 7	GloVe	200	circumstances	No	Decision Tree
Article 8	echr2vec	100	procedure+facts	Yes	Random Forest
Article 9	n-gram	2000	circumstances	Yes	AdaBoost
Article 10	echr2vec	200	procedure+facts	No	QDA
Article 11	GloVe	200	procedure+facts	Yes	SGD
Article 13	GloVe	100	procedure+facts	Yes	QDA
Article 14	echr2vec	200	procedure+facts	No	QDA
Article 18	echr2vec	200	procedure+facts	No	Random Forest

Table 5.1: Model Hyper-parameters

The accuracy of the models and the heuristic on the test set can be seen in Figure 5.1. The accuracy for each Article as well as the weighted average across all the Articles are shown. As mentioned in the Methodology, the weights are given by the number of Judgments in the test set for each Article. For the models, the weighted average is 0.6883. This is the best estimation of how well the models will perform in a realistic scenario on new cases. That is, it estimated that 68.83% of the predictions made by the model would be correct. All of the Articles, except for Article 10, had a test accuracy above 0.5. Article 10 had a test accuracy of 0.4576 which is 33.52% lower than the average. The highest test accuracy was 0.803 for Article 2 which is 16.64% higher than the average.

The test accuracies in Figure 5.1 can be compared to the heuristic accuracies. For all Articles, excepting 7, 14 and 18, the accuracy of the heuristic on the test set was higher. The greatest differences was for Article 10, where the heuristic accuracy was 66.7% greater than the test accuracy. This is followed by Articles 6 and 5 where the heuristic accuracy was 34.25% and 33.3% greater, respectfully. The weighted average, for the heuristic, was 0.8668 which is 29.7% higher than the weighted average for the models on the test set. Hence, in general, the heuristic has outperformed the models.

The higher accuracy of the heuristic can be partly explained by the balance of violation to non-violations in the test sets. Take for instance Article 6 where, in the past, 91% of the complaints about this Article resulted in violations. As the test sets had the same balance as past judgements, the heuristic correctly predicted the outcome of Article 6 judgements with 91%. We saw in the methodology that this imbalance is common among the Articles contributing to the higher accuracy of the heuristic. The recall and precision of the models further explain why the heuristic outperformed the models.

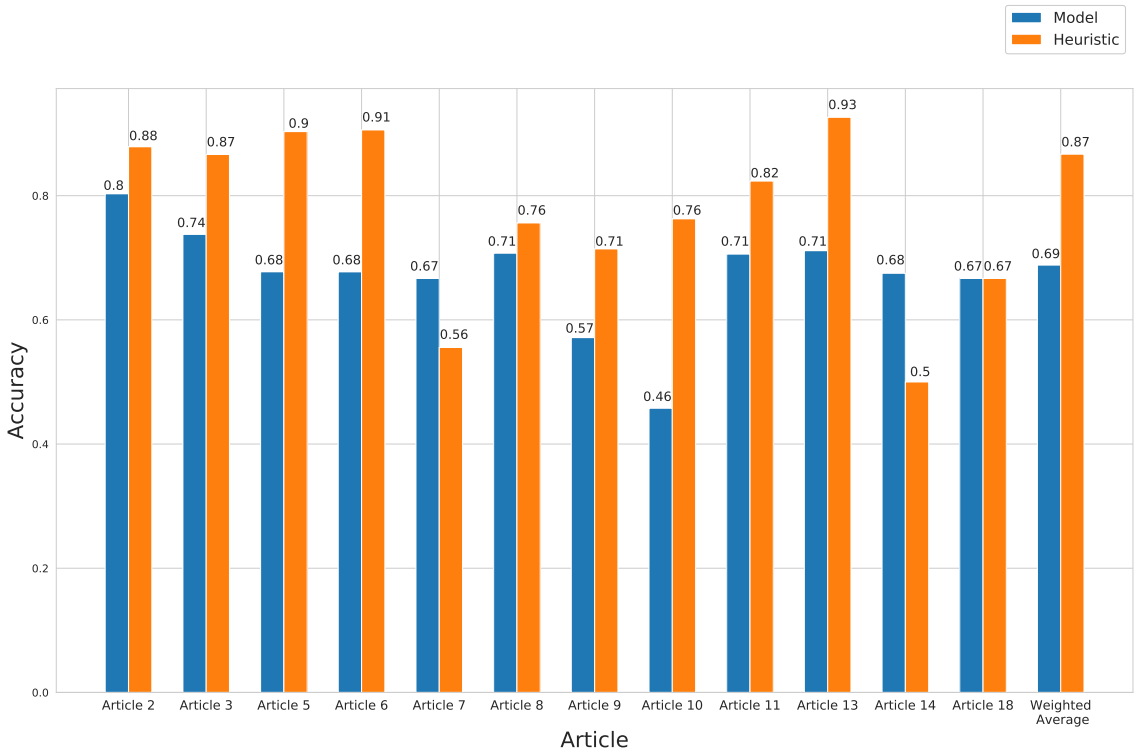

Figure 5.1: Model and Heuristic Accuracy on Test Set

Figure 5.2, shows the precision and recall, on the test sets, of the models. 7 of the Articles had a precision above 0.9 and 9 of the Articles had a precision above 0.8. Article 2 and Article 9 both had a precision of 1. Article 18 had the lowest precision of 0.5. The average precision, across all the Articles, is 0.8491. In general, the high precisions mean that models tend not to miss-classify non-violations as violations. For instance, take Article 6. Its model had a precision of 0.9677 which means 96.8% of the Judgments predicted to be violations were violations. In comparison, lower recall values are observed. For 9 of the Articles, the precision was higher than the recall. The highest recall, 1, was for Article 18 and the highest recall, for the Articles above the data threshold, was 0.7759 for Article 2. The average recall was 0.6906. The lower recall, means the models tend to miss-classify violation cases as non-violation cases. In other words, incorrect predictions are mainly due to false negatives. This is made clearer by the confusion matrix for Article 6 in Table 5.2. Here we see that the number of false negatives is 180. This is 15 times

higher than the number of false positives, 12.

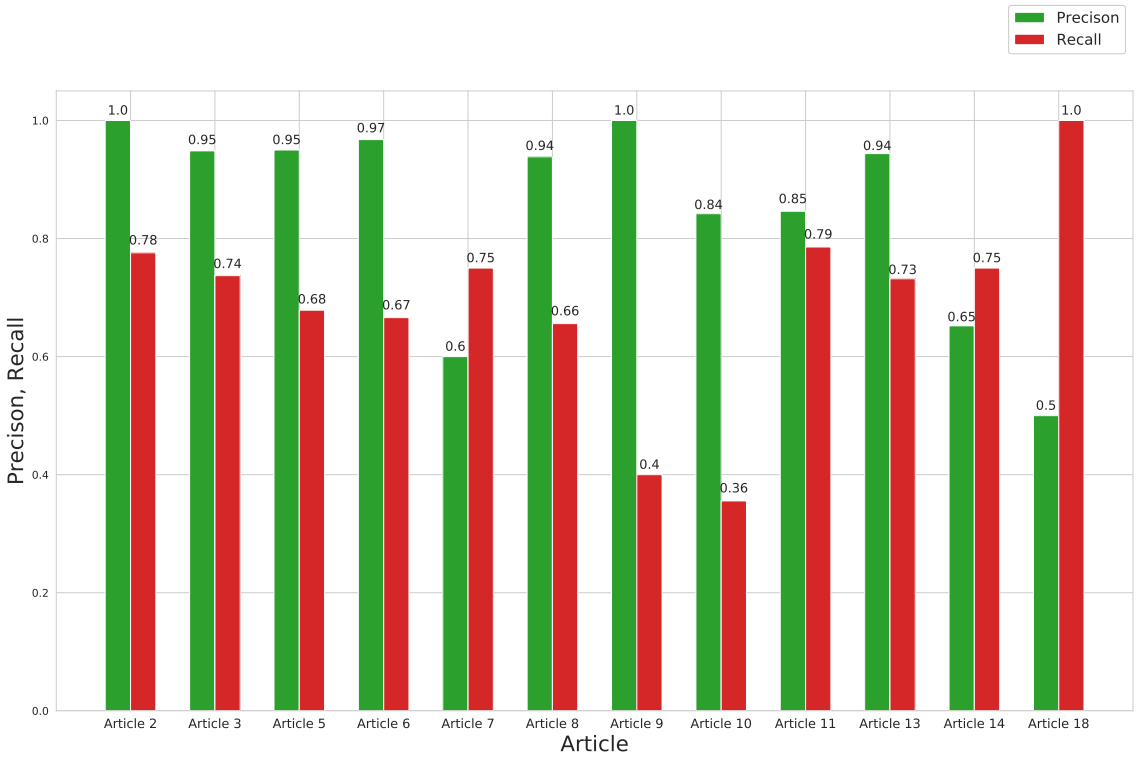

Figure 5.2: Accuracy, Precision and Recall on Test Set

		Prediction	
		Non-violation	Violation
Actual	Non-violation	44	12
	Violation	180	359

Table 5.2: Confusion Matrix for Article 6

Ultimately, it is not likely that either the heuristic or model would be used by the ECHR to make judgements. Using the heuristic would mean that all judgements would always lead to a violation (or non-violation for some Articles) regardless of the evidence provided. This would not be acceptable as there would be no due process. Similarly, it is not likely that the models would be used. This is because the decisions made by the Court are very important and an accuracy of below 70% would be unacceptable. It would mean that more than 30% of application would be given an incorrect judgement regardless of the evidence. The consequences of this could be severe considering that the Court was set up to protect human rights. This does not mean that the models would be useless to the Court.

The models could still be used to prioritise cases by identifying which cases are more likely to lead to violations. The heuristic does not provide any benefit in terms of prioritising cases. As the predictions for each Article would be the same, all complaints would be given the same priority. In this sense, the models may be more useful. As discussed above, the tendency to have a high precision means there are relatively few false positives. This means the cases identified as violations and subsequently prioritised, will tend to be violations. The downside is that those judgements, misclassified as non-violations, would be given equal priority to the remaining non-violation cases. Nonetheless, overall the models would put the cases in a better order as more violation cases would be heard sooner.

5.2 Cross-validation Accuracy by Section

It was mentioned that, from a Judgment, 5 sections are considered. There are three individual sections (procedure, circumstances and relevant) and 2 combinations of these sections. The facts consist of the circumstances and relevant individual sections and procedure+facts consists of all the individual sections. The "Section" column in Table 5.1, gives the Judgment section that gave the highest cross-validation accuracy for each Article. The most common section was procedure+facts which was used for 8 out of the 12 Articles. This is followed by circumstances and relevant which were used for 3 and 1 of the Articles respectfully. Both the procedure and facts section were not used for any of the Articles.

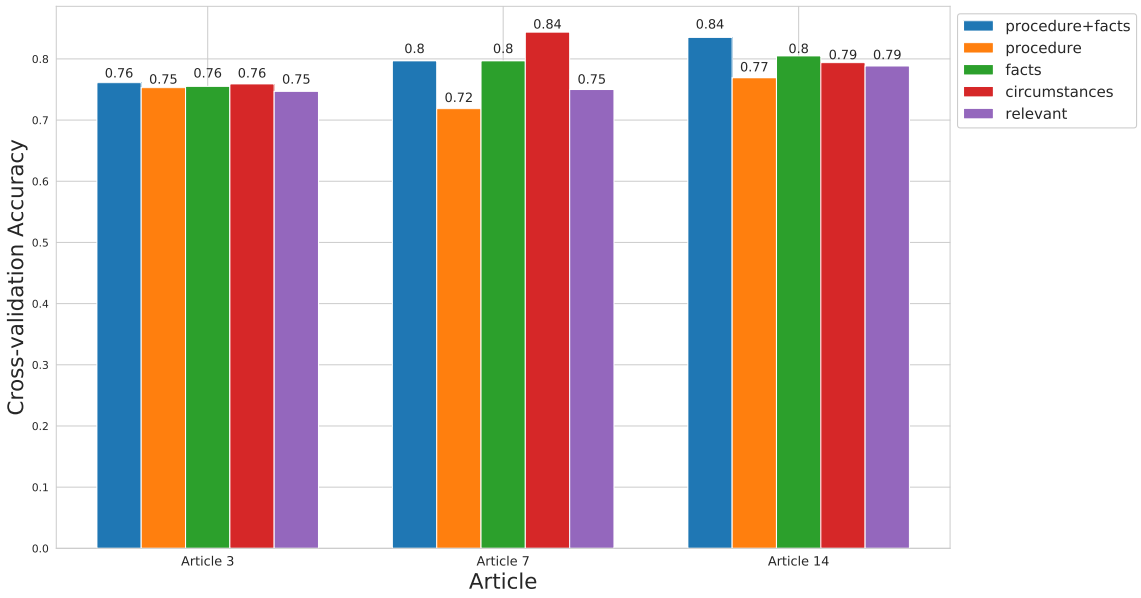

Figure 5.3: Highest Cross-validation Accuracy by Section

Using Articles 3, 7 and 14 as examples, Figure 5.3 shows how the cross-validation ac-

curacy differs by section. For each of the Articles, the highest cross-validation accuracy for each section is shown. Article 3 was chosen as it had the lowest difference between any two sections. That is the difference between the procedure+facts and relevant section was 0.0143. Article 7 provided the greatest difference and Article 14 provided the greatest difference for those articles above the data threshold. Figure 5.4, summaries the differences in cross-validation accuracies for each section. To create this Figure, the highest cross-validation accuracy for each section for each Article was taken. For instance, we see those highest accuracies for Articles 3, 7 and 14 in Figure 5.3. The average across the Articles was then calculated for each part. This gives the average if the section hyper-parameters were held constant for each Article. The procedure+facts section achieved the highest average of 0.8129. The individual section with the highest average was circumstances with 0.8027. This is 1.3% lower than the average for the procedure+facts section.

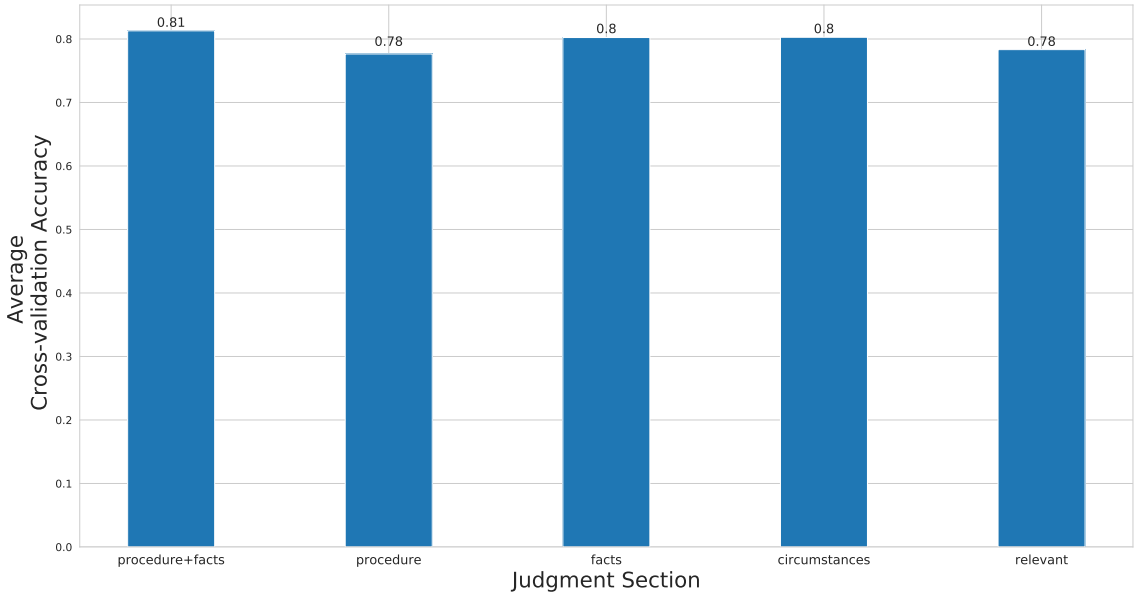

Figure 5.4: Average of Models with Highest Cross-validation Accuracy by Section

Before predictions can be made for new Judgments, the sections like those in previous documents would have to be drafted. The cross-validation accuracy results suggest that to maximise accuracy different sections should be drafted for different Articles. However, practically, it may make more sense to use the same section for each Article. It may increase the administrative burden if, for every Article, a different section had to be drafted. We see that the facts+procedure section had the greatest overall accuracy and so this section could be used. However, practically again, this may not make sense. It would mean that, for every Article, every individual section had to be drafted. As an alternative, the circumstances section could be used for every Article. This section produced the highest overall accuracy amongst the individual sections. Using models trained on this section would mean that neither the relevant or procedure sections would

have to be drafted to make predictions. In terms of addressing the application backlog faced by the ECHR, this may be a better option. This is because it would take less time to prepare the necessary sections for the models and predictions could be made sooner. Ultimately, the decision would have to be made between increasing accuracy and decreasing the time it takes to draft sections for the models.

5.3 Cross-Validation Accuracy by Feature Type

Looking at Table 5.1, we see that the feature type that produced the highest cross-validation accuracy differed for each Article. For all Articles, except Article 9, an average embedding feature type produced the highest cross-validation accuracy. The embedding model used for these Articles varied. This could be due to the nature of the Articles. For instance, the Article 6 model used the echr2vec embedding model which was created using legal documents. This Article protects the right to a fair trial. A fair trial could be considered a legal concept and so the accuracy for this Article may have been improved due to the fact that the embeddings were constructed using legal documents. In comparison, the GloVe embeddings were used for Article 3. This Article prohibits torture and inhuman and degrading treatment. In comparison to a fair trial, these concepts could be more common in general, non-legal documents. Hence, an embedding trained on more general documents was appropriate. However, this hypothesis does not hold for all of the Articles. For instance, Article 7 prohibits punishment without law, an arguably legal concept, and used the GloVe embeddings.

Figure 5.5, gives the average cross-validation accuracy by Feature Type and Dimension. Similar to Figure 5.4, Figure 5.5 is created by taking the highest cross-validation accuracy for each Feature Type and Dimension combination for each Article. Then the average across the Articles is calculated. The echr2vec with dimension 200 had the highest average of 0.8066. This is 0.4% higher than the GloVe with dimension 200 which had the second-highest average of 0.8031. The doc2vec with dimension 100 had the lowest average of 0.7298 which is 9.5% lower than that of the echr2vec with 200 dimension. For each of the word embedding and doc2vec features, the 200 dimension version had a higher accuracy than the 100 dimension version. This difference was 0.0102, 0.0078, 0.0096 and 0,001 for the GloVe, law2vec and echr2vec and doc2vec features respectfully.

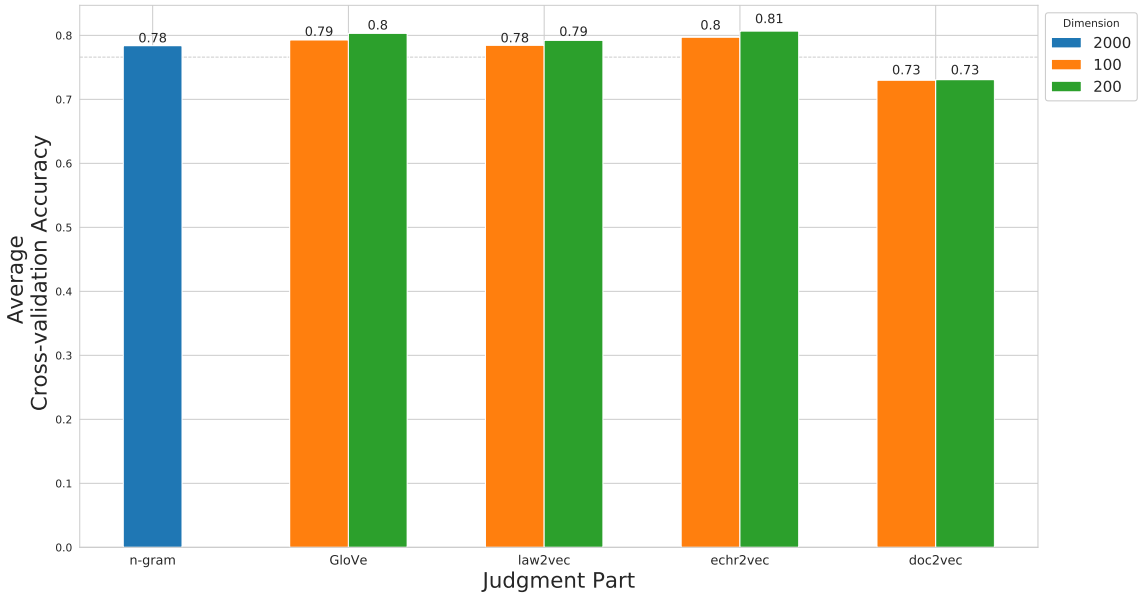

Figure 5.5: Weighted Average of Highest Cross-validation Accuracy by Feature Type and Dimenison.

In general, the echr2vec embeddings produced the highest cross-validation accuracy. This suggests that using embeddings trained on legal documents can improve model accuracy over those that use embeddings trained on general documents. In other words, legal embeddings may capture the semantics of legal documents better than general embeddings. However, we also saw that the law2vec embeddings performed worse, in general, than the GloVe embeddings. This suggests that the type of legal documents used to train the embeddings is important. The law2vec embeddings may not have captured the semantics of words used in the Judgments as well as the echr2vec embeddings. This could be because the law2vec embeddings were trained on legal documents from other countries and areas of law to that of the ECHR.

Lastly, in general, the doc2vec features had the lowest average accuracy. They also did not produce the highest accuracy for any of the Articles. This could mean that the word order of the Judgments is not important. In other words, capturing the semantics of the entire document does not provide any benefit over capturing the average semantics of the words in the document. It could also be because a variety of different types of documents were used to train the doc2vec models. Particularly, decisions made up 40% of the documents used to train the models. The semantics of these decision documents may differ from that of Judgments. Using the doc2vec models to infer vectors from the Judgement documents may not produce an adequate representation of the Judgement.

5.4 Comparison of Cross-validation Accuracy with Previous Studies

The test accuracy achieved by this study cannot be compared to that of another study. This is because, as mentioned in the Related Work, no other study has tested their models on a realistic test set. We can, however, compare the cross-validation accuracies presented in the related work section. Figure 5.6 shows the highest cross-validation accuracy for each Article achieved in this study. These are the accuracies that correspond to those hyper-parameters and classifiers shown in Table 5.1. The average accuracy is 0.8225. Articles 18 and 11 had the highest accuracies of 1 and 0.9348 respectfully. These are 21.6% and 13.7% higher than the average accuracy. The lowest accuracy, 0.7319, was for Article 5 which is 11.0% lower than the average.

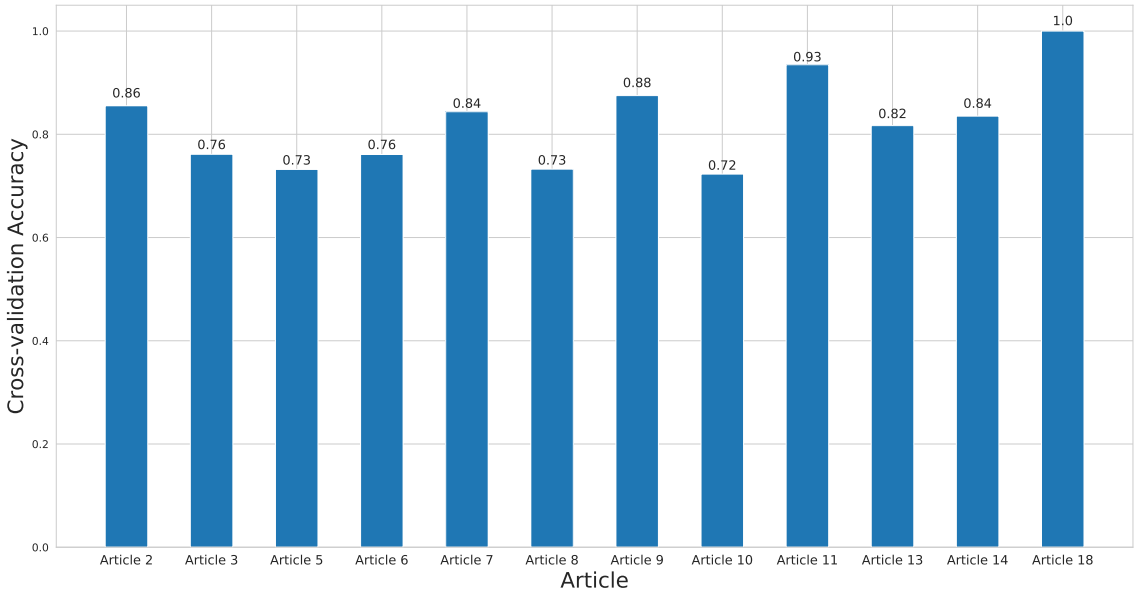

Figure 5.6: Highest Cross-validation Accuracy for each Article

When comparing cross-validation accuracies, we should consider that different studies have construct models for different Articles. Specifically, Aletras et al. (2016*b*) and Liu & Chen (2017) constructed models for Articles 3, 6 and 8. They achieved an average cross-validation accuracy of 80.1% and 79.5% respectfully. Medvedeva et al. (2018*b*) constructed models for Articles 2, 3, 5, 6, 8, 10, 11, 13 and 14. Their average across Articles 3, 6 and 8 were 74.3% and their average across all nine Articles was 75.3%. In comparison, across Articles 3, 6 and 8 this study achieved an average cross-validation accuracy of 75.1%. This is 6.2% less than the highest accuracy and 1.1% higher than the lowest accuracy previously achieved for these Articles. Across the 9 Articles, this study achieved an average accuracy of 77.7% which is 3.2% higher than that achieved by Medvedeva et al. (2018*b*).

Looking at the 9 Articles it appears as if the approach taken in this study has improved on the accuracy. However, looking at the 3 Articles it appears as if the approach has reduced the accuracy. It is difficult to determine whether these differences would result in a corresponding difference in accuracy on a test set. One reason for this is that balanced sets were used to train models and a realistic test set would be unbalanced. Suppose, hypothetically, a model could classify all non-violations correctly but misclassified 50% of violations. This model would achieve an accuracy of 0.75 on a balanced set. However, on an unbalanced test set the accuracy would be different. Particularly, if there were more violations than non-violations the test accuracy would be lower than 0.75. Additionally, over-fitting may result in high cross-validation accuracies. Looking at Figure 5.6, the cross-validation accuracies for Article 18 and 11 are particularly high. Specifically, an accuracy of 1, for Article 18, is very unlikely in reality. This suggests over-fitting has occurred and is discussed in more depth in the next section.

5.5 Cross-validation Accuracy and Over-fitting

Figure 5.7, shows the relationship between the training set size and cross-validation accuracy. These are the same cross-validation accuracies as in Figure 5.6. Each point is tagged with its associated Article and the dotted black line gives the threshold of 100 Judgments. The 4 Articles, that fall below the threshold, have 4 of the 5 highest accuracies. These 4 Articles have an average accuracy of 0.9134. The average accuracy of the remaining 8 Articles is 0.7771 which is 14.92% lower. Small training sets, of below 100 Judgments, are associated with high cross-validation accuracies. Potentially, small training set sizes are leading to over-fitting resulting in high cross-validation accuracies.

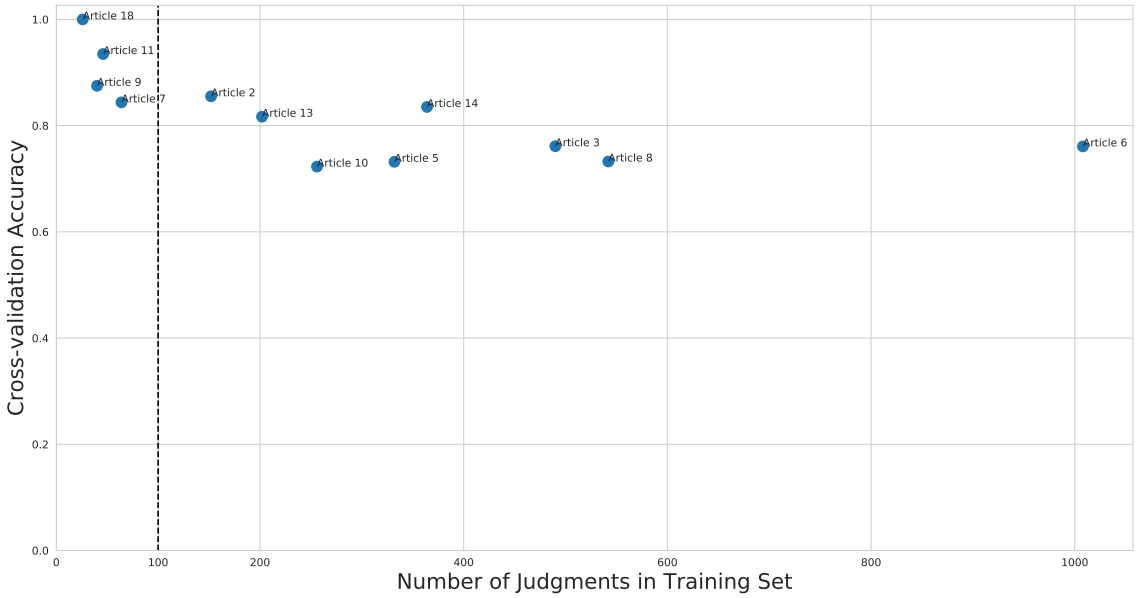

Figure 5.7: Relationship between Cross-validation Accuracy and the Number of Judgments in Training Set

To gain more insight, the difference between cross-validation and test accuracy for each Article is considered. Figure 5.8, shows the relationship between these differences and the training set size. The average difference for the Articles below the data threshold is 0.2607. These Articles had 4 of the 5 highest differences. Specifically, Article 18 and 9 had the greatest differences of 0.3333 and 0.3036 respectfully. For the Articles above the threshold, the average difference was 0.0962. Hence, small training set sizes are associated with high differences in cross-validation and test accuracy. In other words, small dataset sizes are associated with over-fitting.

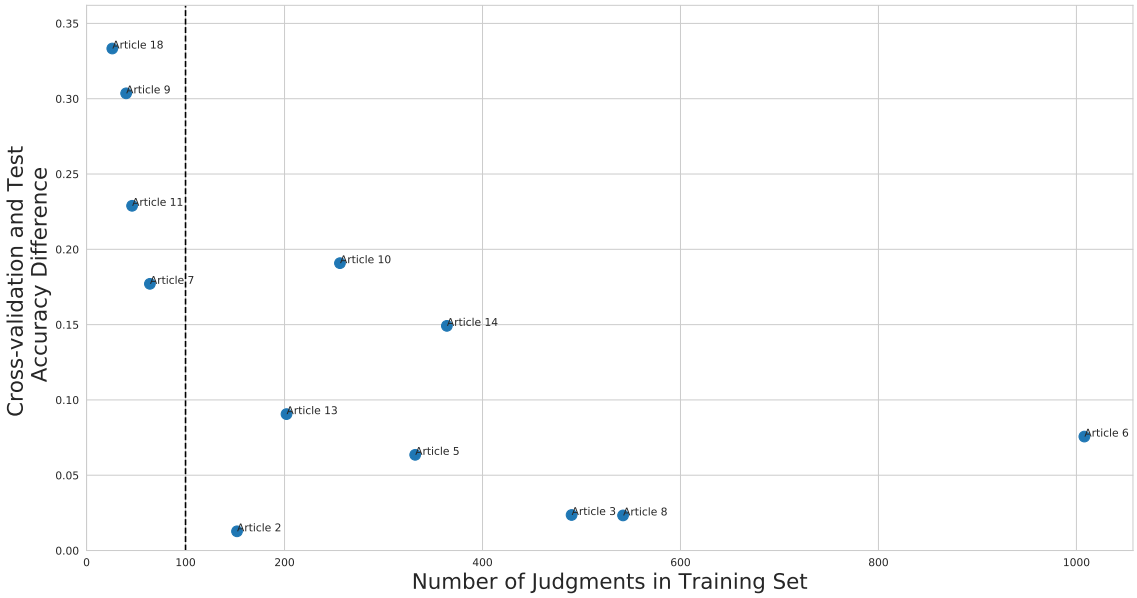

Figure 5.8: Relationship between Training Set Size and the Difference between Training and Test Accuracy

Other aspects of the methodology could also explain the over-fitting. Looking again at Figure 5.8, Articles 10, 14 and 13 had the three highest differences for those Articles above the threshold. A potential reason for this is that they all have the same classification algorithm, QDA. None of the remaining Articles, above the threshold, use this classifier and they have a smaller difference in comparison. Hence, the properties of a QDA, such as that it uses a quadratic decision boundary, may be contributing to the over-fitting for these Articles (Pedregosa et al. 2011).

Chapter 6

Conclusion

Given the results of the models, it is unlikely that the ECHR would use the models to make judgements. Using a realistic data set, the models achieved a weighted average of 68.83%. This is the best estimation of how well the models would predict the outcome of potential violations. Hence, it is estimated that if the models are used by the ECHR over 30% of rulings on human rights violations would be incorrect. Looking at the models' recall showed use that the incorrect predictions were mostly due to false negatives or, in other words, violations being predicted as non-violations. Hence, if the models were used by the Court many human rights violations would go unpunished. Additionally, in some cases, States would be falsely accused of violations. This would have negative consequences for the Court's reputation.

As discussed, this does not mean the models cannot still be a useful tool. The models could provide an indication of which applications in the backlog should be prioritised. In this sense, the study has contributed towards solving the problem of a large application backlog as the applications would be addressed in a more efficient order. Unfortunately, this does not reduce the burden on the Court in terms of how many judgements it would have to make. This is because judgements would have to be made for all applications regardless of their priority. In other words, it would take the same amount of time to address the application backlog regardless of what order the applications are in. Ultimately, the research conducted is not enough to solve the research problem. Nonetheless, the study has made some contributions towards this area of research.

6.1 Contributions

As far as we could tell, the first realistic test set has been used to determine the accuracy of the models. This provided the first realistic estimate of how well machine learning algorithms can predict the outcome of judgments made by the ECHR. This is an important baseline that the results of future work can be compared to. An additional baseline, the

accuracy achieved by a heuristic, was also provided. The heuristic was to always predict the outcome that was the most common outcome in the past. Again, this is the first instance of such a baseline being provided. It is important to include as it puts model results in perspective.

More data for the ECHR was obtained than in previous studies. This includes both Judgments as well as other documents available from the Court. This was the first study to incorporate the additional documents, besides Judgments, into the machine learning models. This was done by creating the echr2vec embedding. Textual features created using this embedding produced the highest cross-validation accuracy for 5 of the 12 Articles. Additionally, the same textual features produced the highest cross-validation accuracy on average across all 12 Articles. Hence, within the context of this study, the echr2vec embedding seems to have improved the accuracy of the models. Additionally, such an embedding could be used in other applications that would benefit the ECHR. For example, it could be used to aid legal search.

Lastly, a new tool, auto-sklearn, was used for classification model and hyper-parameter selection. This allowed for a wider range of models and parameters to be tested than tested in previous papers. Using this package, the algorithms found to maximise cross-validation accuracy differed from previous studies. In other words, using the package has likely increased cross-validation accuracy. This suggests that it was necessary to apply different algorithms to the problem, than previously applied, to improve accuracy. Ultimately, researches can use the algorithms found in this study as a basis for algorithm selection in future work.

Chapter 7

Limitations and Future Work

The over-fitting present in the results was discussed in depth in the Results and Discussion chapter. Factors that potentially contributed to the over-fitting were discussed. These include the small datasets for certain Articles and the type of classification algorithm chosen by the auto-sklearn package. It is difficult to determine the precise cause and additional experiments could be performed to gain a better understanding. For example, for a given Article, the about of data used to train models could be varied and the resulting differences between cross-validation and test accuracy could be measured. A negative relationship would provide more evidence that small datasets lead to over-fitting. Ultimately, the effect of over-fitting should be considered in future work. It also highlights the necessity for test sets as cross-validation accuracy can potentially be overstating the accuracy of the models.

Another limitation is that only the final cross-validation accuracies were used to compare classification algorithms. This is opposed to considering the accuracy on individual folds which could be used to create confidence intervals around the cross-validation accuracies. This would give an indication of whether the difference in cross-validation accuracy for each classifier was statistically significant. In other words, by only comparing cross-validation accuracy we do not take into account the variation in accuracy on the individual folds. The same can be said about the comparison of the hyper-parameters such as the Feature Type. This limitation follows from a limitation of the auto-sklearn package where the fold accuracies are only calculated internally. Hence, in future work, it may be better to use grid search to test different hyper-parameters and algorithms. This way the cross-validation fold accuracies would be available. This comes at the cost of having to select classification algorithms and associated hyper-parameters to be tested beforehand.

On a similar note, only the algorithms that produced the best cross-validation accuracy were presented. For instance, we saw that a random forest classifier produced the highest accuracy for Article 3. This is because of the way the auto-sklearn package was used. The package was requested to return the results from only the model with the

greatest cross-validation accuracy. In hindsight, this was not the best way to use the package. In future work, it would be better to return the results for all the algorithms tested. Firstly, this would have allowed us to determine how much the cross-validation accuracy for the best algorithm compared to the rest. Secondly, it would have also been possible to determine how well each algorithm performed across all the Articles. In other words, it could have been possible determined how well each algorithm performed in general. This would have provided valuable insight for future work.

A final drawback relating to auto-sklearn is that the package does not allow you to test all possible classification algorithms. It is confined to the 15 algorithms selected by Feurer et al. (2015). In the future, additional algorithms could be tested. Specifically, a CNN has been shown to perform well on other text classification tasks. These included sentiment analysis of movie reviews and classifying questions by their type (Kim 2014). This could be done using the Keras implementation of a 1D CNN network with an embedding layer (Chollet et al. 2015). Additionally, this would provide an alternative approach to using the different word embeddings that does not involve averaging the embeddings. The average word embeddings showed some promising results but information, such as word order, is lost through averaging. Ultimately, by using an embedding layer the individual word embeddings are considered which could improve the model's accuracy. That being said, the accuracy is not the only aspect of the models that could be improved.

The models constructed in this study provided only the final predictions for each Judgment. They did not provide any indication of how predictions are made. In reality, Judges have to justify their decisions and so they would not be able to rely on a model that gives only a final prediction. In future work, this is an aspect of the models that should be considered. Models that provide information on how predictions are made would likely be more useful to judges. The judge would be able to use the prediction was after confirming that the reasons for the prediction are correct. Additionally, even if the model is not used to make the final judgement, it could still be used to aid the judges decision making. For example, suppose the model could determine the laws that apply and have led to a verdict. The judge could use this information in constructing their own arguments for the verdict. Such a model would undoubtedly be more complex than the current approach. A simpler initial step could be to analyse the weights of the models. For instance, for the models that used N-gram features, the weights for the N-gram features can be analysed. N-grams with positive weights are associated with violations and those with negative weights are associated with non-violations. The greater the weight the greater effect the particular N-gram has on the prediction. Hence, by analysing the weights we can understand which N-grams are important in making predictions.

The next two limitations consider the documents used to construct machine learning models. Firstly, only Judgments of a specific structure are used. That is only Judgments that had both the circumstances and relevant sections. This means the models were not

constructed using all available Judgments. Training the models on fewer data could have a negative effect on the model's accuracy. An additional downside is that the subset of Judgments, used to train models, is not necessarily representative of all Judgments. This is not something considered in previous studies but it should be considered in future work. It may be better to only consider the facts section as a whole and not the two individual subsections. This is because, we saw in the Methodology, that a larger proportion of the Judgments would be considered.

Another consideration is that the Judgment documents are only produced after the final judgement has been made. Consequently, they would not be available to make predictions before a Judgment has been made. The sections of the Judgment, needed for the models, would have to be drafted before a prediction is made. A better alternative would be to construct a model using the documents provided by the individual or State when making an application. Firstly, these documents are available before the judgement is made. Consequently, there will be no issue of data leakage when constructing models. Additionally, would be less of an administrative burden on the Court as the documents could be taken as is without any processing. However, these documents are not publicly available. Hence, in future work, if these documents were to be used the researcher would likely have to collaborate with the Court.

The final limitation is that not all judicial decisions made by the ECHR are being predicted by the models. Only judgements are being predicted. The procedure of the Court was covered in the Background chapter. There we saw that the application must be deemed admissible before a judgement can be made. If an application is deemed inadmissible it is said to be have been decided by "decision". As these models were trained to predict the outcome of Judgments they can only be used in the final stage of the application process. In the Background section, we saw that the majority of the applications do not make it to this stage. In 2018, 94% of all applications were decided by decision. This suggests that, to address the backlog of applications, it may be better to construct models that predict the outcome of decisions. At this point, no study could be found that aimed to predict the decisions of the ECHR and so it is potentially an important area for future work.

Bibliography

Agrawal, S., Ash, E., Chen, D., Gill, S. S., Singh, A. & Venkatesan, K. (2017), Affirm or reverse? using machine learning to help judges write opinions, Technical report, Working Paper.

Alarie, B., Niblett, A. & Yoon, A. H. (2016), 'Law in the future', *University of Toronto Law Journal* **66**(4), 423–428.

Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D. & Lampos, V. (2016*a*), 'echr dataset', `https://figshare.com/s/6f7d9e7c375ff0822564`.

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

Bala, J., Kellar, M. & Ramberg, F. (2017), Predictive analytics for litigation case management, *in* 'Big Data (Big Data), 2017 IEEE International Conference on', IEEE, pp. 3826–3830.

Beck, Copp and Bazeley v. The United Kingdom (2002), `http://hudoc.echr.coe.int/eng?i=001-60697`. Accessed: 2019-06-30.

Belziuk v. Poland (1998), `http://hudoc.echr.coe.int/eng?i=001-58145`. Accessed: 2019-06-30.

Chalkidis, I. & Kampas, D. (2019), 'Deep learning in law: early adaptation and legal word embeddings trained on large corpora', *Artificial Intelligence and Law* **27**(2), 171–198.

Chollet, F. et al. (2015), 'Keras', `https://keras.io`.

Council of Europe (1950), 'European convention for the protection of human rights and fundamental freedoms, as amended by protocols nos. 11 and 14', `https://www.refworld.org/docid/3ae6b3b04.html`. Accessed: 2019-06-30.

Council of Europe (2014), 'European court of human rights: The echr in 50 questions', `https://www.echr.coe.int/Documents/50Questions_ENG.pdf`. Accessed: 2019-06-30.

Council of Europe (2016), 'European court of human rights: Questions and answers', https://www.echr.coe.int/Documents/Questions_Answers_ENG.pdf. Accessed: 2019-06-30.

Council of Europe (2017), 'European court of human rights: Hudoc user manual', https://www.echr.coe.int/Documents/HUDOC_Manual_ENG.PDF. Accessed: 2019-06-30.

Council of Europe (2018), 'Your application to the echr: How to apply and how your application is processed', https://www.echr.coe.int/Documents/Your_Application_ENG.pdf. Accessed: 2019-07-03.

Council of Europe (2019*a*), 'European court of human rights: Analysis of statistics 2018', https://www.echr.coe.int/Documents/Stats_analysis_2018_ENG.pdf. Accessed: 2019-06-30.

Council of Europe (2019*b*), 'European court of human rights: Rules of court', https://www.echr.coe.int/Documents/Rules_Court_ENG.pdf. Accessed: 2019-07-03.

Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M. & Hutter, F. (2015), Efficient and robust automated machine learning, *in* C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama & R. Garnett, eds, 'Advances in Neural Information Processing Systems 28', Curran Associates, Inc., pp. 2962–2970.
URL: *http://papers.nips.cc/paper/5872-efficient-and-robust-automated-machine-learning.pdf*

Ge, L. & Moh, T.-S. (2017), Improving text classification with word embedding, *in* '2017 IEEE International Conference on Big Data (Big Data)', IEEE, pp. 1796–1805.

Guimerà, R. & Sales-Pardo, M. (2011), 'Justice blocks and predictability of us supreme court votes', *PloS one* **6**(11), e27188.

HUDOC database (2018), https://echr.coe.int/Pages/home.aspx?p=caselaw/HUDOC&c=. Accessed: 2018-11-18.

Joachims, T. (1998), Text categorization with support vector machines: Learning with many relevant features, *in* 'European conference on machine learning', Springer, pp. 137–142.

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

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

Kim, Y. (2014), 'Convolutional neural networks for sentence classification', *arXiv preprint arXiv:1408.5882* .

Kuhn, M. & Johnson, K. (2013), *Applied Predictive Modeling*, Springer, Springer New York Heidelberg Dordrecht London.

Lau, J. H. & Baldwin, T. (2016), 'An empirical evaluation of doc2vec with practical insights into document embedding generation', *arXiv preprint arXiv:1607.05368* .

Le, Q. & Mikolov, T. (2014), Distributed representations of sentences and documents, *in* 'International conference on machine learning', pp. 1188–1196.

Li, Y. & Yang, T. (2018), Word embedding for understanding natural language: a survey, *in* 'Guide to Big Data Applications', Springer, pp. 83–104.

Lilleberg, J., Zhu, Y. & Zhang, Y. (2015), Support vector machines and word2vec for text classification with semantic features, *in* '2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)', IEEE, pp. 136–140.

Liu, Z. & Chen, H. (2017), A predictive performance comparison of machine learning models for judicial cases, *in* 'Computational Intelligence (SSCI), 2017 IEEE Symposium Series on', IEEE, pp. 1–6.

Liu, Z., Lv, X., Liu, K. & Shi, S. (2010), Study on svm compared with the other text classification methods, *in* '2010 Second International Workshop on Education Technology and Computer Science', Vol. 1, IEEE, pp. 219–222.

Loper, E. & Bird, S. (2002), 'Nltk: the natural language toolkit', *arXiv preprint cs/0205028* .

McGinnis, J. O. & Pearce, R. G. (2013), 'The great disruption: How machine intelligence will transform the role of lawyers in the delivery of legal services', *Fordham L. Rev.* **82**, 3041.

Medvedeva, M., Vols, M. & Wieling, M. (2018*a*), 'Dataset: crystal_ball_data.tar.gz', `https://www.dropbox.com/s/lxpvvqdwby30157/crystal_ball_data.tar.gz`. Accessed: 2019-06-30.

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

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. & Dean, J. (2013), Distributed representations of words and phrases and their compositionality, *in* 'Advances in neural information processing systems', pp. 3111–3119.

Nisbet, R., Elder, J. & Miner, G. (2009), *Handbook of statistical analysis and data mining applications*, Academic Press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011), 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research* **12**, 2825–2830.

Pennington, J., Socher, R. & Manning, C. (2014), Glove: Global vectors for word representation, *in* 'Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)', pp. 1532–1543.

Quemy, A. (2018), 'European court of human right open data project', *arXiv preprint arXiv:1810.03115* .

Řehůřek, R. & Sojka, P. (2010), Software Framework for Topic Modelling with Large Corpora, *in* 'Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks', ELRA, Valletta, Malta, pp. 45–50. `http://is.muni.cz/publication/884893/en`.

Rook v. Germany (2019), `http://hudoc.echr.coe.int/eng?i=001-194614`. Accessed: 2019-06-30.

Ruger, T. W., Kim, P. T., Martin, A. D. & Quinn, K. M. (2004), 'The supreme court forecasting project: Legal and political science approaches to predicting supreme court decisionmaking', *Columbia Law Review* pp. 1150–1210.

Spaeth, H. J., Epstein, L., Martin, A. D., Segal, J. A., Ruger, T. J. & Benesh, S. C. (2018), 'Supreme court database, version 2018 release 2.', `http://Supremecourtdatabase.org`. Accessed: 2019-02-24.

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

Surden, H. (2014), 'Machine learning and law', *Wash. L. Rev.* **89**, 87.

Szerdahelyi v. Hungary (2012), `http://hudoc.echr.coe.int/eng?i=001-108588`. Accessed: 2019-06-30.

van der Heijden, R. & Kapfer, M. (2016), 'echr-scraping', `https://github.com/UlmApi/echr-scraping`.

Virtucio, M. B. L., Aborot, J. A., Abonita, J. K. C., Aviñante, R. S., Copino, R. J. B., Neverida, M. P., Osiana, V. O., Peramo, E. C., Syjuco, J. G. & 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)', Vol. 02, pp. 130–135.

vizlegal (2019), 'Features', `https://www.vizlegal.com`.

Von Luxburg, U. (2007), 'A tutorial on spectral clustering', *Statistics and computing* **17**(4), 395–416.

Yoon, A. H. (2016), 'The post-modern lawyer: Technology and the democratization of legal representation', *University of Toronto Law Journal* **66**(4), 456–471.

Appendix

Additional Tables

Article	**Training Accuracy**	**Test Accuracy**	**Test Precison**	**Test Recall**	**Heuristic Accuracy**
Article 2	0.8553	0.803	1.0	0.7759	0.8788
Article 3	0.7612	0.7376	0.9485	0.7371	0.8663
Article 5	0.7319	0.6774	0.95	0.6786	0.9032
Article 6	0.7609	0.6773	0.9677	0.666	0.9059
Article 7	0.8438	0.6667	0.6	0.75	0.5556
Article 8	0.7325	0.7073	0.9385	0.6559	0.7561
Article 9	0.875	0.5714	1.0	0.4	0.7143
Article 10	0.7229	0.4576	0.8421	0.3556	0.7627
Article 11	0.9348	0.7059	0.8462	0.7857	0.8235
Article 13	0.8168	0.7114	0.9439	0.7319	0.9262
Article 14	0.8352	0.675	0.6522	0.75	0.5
Article 18	1.0	0.6667	0.5	1.0	0.6667

Table 1: Model Results for Each Article