UNIVERSITY OF LIVERPOOL

Predicting decisions of the European Patent Office's Boards of Appeal using Machine Learning

Author: David Bareham (ID: 201060865)
Project Supervisors: Prof. Katie Atkinson (School of EEECS) and Mr. Jeremy Marshall (School of Law)

Abstract

This work aims to assess the feasibility of applying data science and artificial intelligence methods to the problem of case outcome prediction for appeals from the European Patent Office's Boards of Appeal, concerning the grant of a patent application. The task is conceptualised as a binary classification task in which an appeal can 'affirm' or 'reverse' the prior judgement. Using a range of machine learning classifiers and textual representations, including custom-trained word and document embeddings, two experiments were conducted on appeal cases from the Examining Division of the European Patent Office. The first using randomly-sampled data and the second with year-stratified data to engage in future prediction. The first experiment achieved 85% accuracy and the second an average of 86%. The results demonstrate the viability of applying machine learning techniques to appeals concerning the patent grant procedure, showing that patents as a legal domain may be promising for future case outcome prediction research.

Acknowledgments

I would like to thank my supervisors Prof. Katie Atkinson and Mr. Jeremy Marshall for their support and guidance in completing this dissertation, especially through the ups and downs of getting access to suitable data. I'd also like to thank my fiancée Marta for her endless support each and every day.

Contents

List of Figures

List of Tables

1 Introduction

In this chapter, I will be introducing the choice of domain for this work as well as outlining the main aims and objectives to be realised.

1.1 Overview

The task of case outcome prediction is the branch of Artificial Intelligence (AI) and Law referring to automatically predicting the outcome of a court decision given some input relevant to the case. Most work focuses on the European Court of Human Rights [1, 2], the Supreme Court of the United States [3, 4] and the Chinese Legal System [5, 6]. Comparatively, little research has been performed in case outcome prediction for the legal domain of Intellectual Property Law, encompassing sub-domains such as trademarks and patents. This lack of research motivated the selection of the European Patent Office's (EPO) Boards of Appeal as the focus domain for this work. While some prior literature has discussed predicting whether patents may be granted upon their submission [7, 8, 9], to the best of my knowledge no prior work has computationally analysed the appeals process when a patent submission is refused.

The benefit of drawing upon case outcome prediction techniques is that it can help us to better understand the appeal decision-making process and unlock new insights into the EPO appeals.

1.2 Aims and Objectives

The aims and objectives of this project are:

Aims

- To assess the feasibility of applying Data Science/AI techniques to EPO appeal decisions and to understand whether this previously unstudied data source may be fruitful for further research involving data analytics.

Objectives

- Experiments testing a range of different classification models and hyper-parameters across both randomly sampled data and year-stratified data
- Evaluating the best classifiers across various metrics such as F1-score, Accuracy and MCC on both train and test data, as well as the model's interpretability
- To pre-train custom Word2Vec and Doc2Vec models with data from patents and appeal judgements to see if a domain specific embedding would result in a performance boost over generalised embeddings and traditional word representation techniques i.e. Bag-of-Words

1.3 The European Patent Office

A patent can be defined as "a legal title granting its holder the right – in a particular country and for a certain period of time – to prevent third parties from exploiting an invention for commercial purposes without authorisation." [10]. Effectively it acts as a mechanism granting a limited commercial monopoly on an invention in exchange for technical disclosure of such an invention for a period of 20 years [11]. The most common justification given for this system is that it act as an incentive to individuals or organizations to disclose information that might otherwise have remained secret, whilst also fostering a beneficial economic incentive for new inventions [11].

Filing patent applications can be costly and time consuming but provides crucial legal protection for a company or individual's inventions, so to mitigate this within many European countries the European Patent Convention 1973 (EPC)[1] created a mechanism for the grant of multiple national patents within a single application [11]. Litigation and infringement are still dealt with by the respective national legal systems but the grant is handled by the EPO whose role is to administer the EPC.

For a European patent application to be granted, the Examining Division of the EPO will assess the substantive content of the application according to the following, non-exhaustive, criteria [10]:

- Invention: It must be an invention. This is not explicitly defined but things that are not inventions include those whose commercial exploitation would be contrary to public order/morality i.e. cloning human beings.
- Novelty: The invention must be new and not considered to be part of the state-of-the-art.
- Inventive Step: An invention involves an inventive step if it is not obvious to the skilled person in light of the state of the art.

After being granted by the Examining Division, there is a period of 9 months in which a third party, i.e. a commercial competitor, may object to the granting of the patent, which is heard before the Opposition Division. Any party who has been adversely affected at any stage, by the Examining Division or Opposition Division, may file an appeal against the decision. The Technical Boards of Appeal are responsible for appeals concerning refusal of a patent application from the Examining Division or appeals against decisions of the Opposition Division. There are other boards such as the Legal Board which deal with different matters. The decisions granted in appeal proceedings are generally delivered at the oral proceedings and have the force of *res judicata*, meaning that the decisions made

[1] `https://www.epo.org/en/legal/epc-1973/2006/convention.html`

cannot be subject to further legal action [12]. Figure 1.1 shows a simplified representation of the grant and appeal process within the EPO.

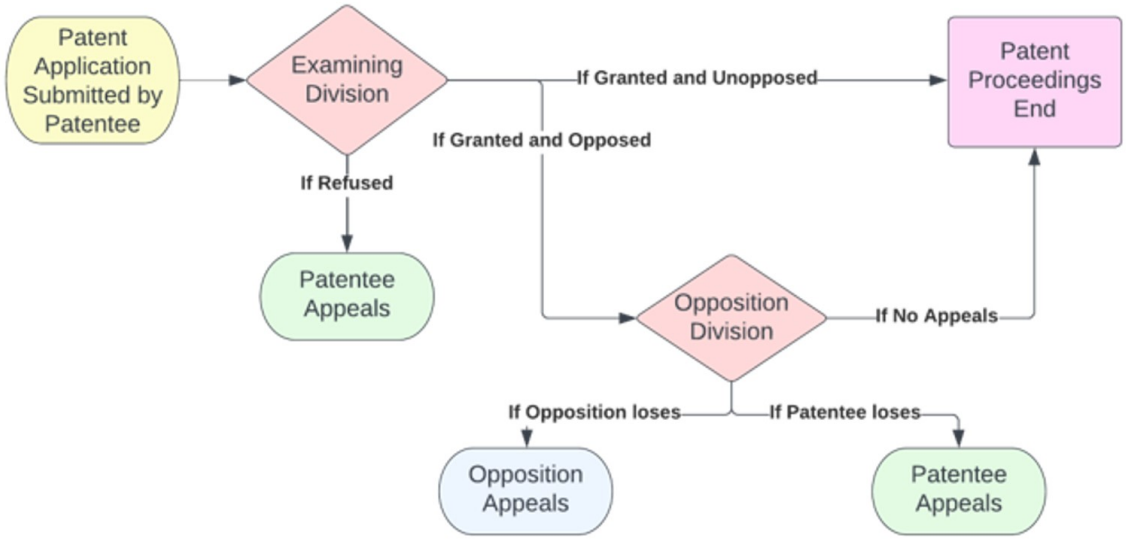


Figure 1.1: Patent Application Grant and Appeal Process

1.4 Data Access & Ethics

All work performed in this analysis was in accordance with the terms of use for the data set out by the EPO and in line with the responsible innovation practices outlined by the UKRI[2]. The data is publicly available through the EPO's proprietary platform[3] for bulk data access. The data itself is free to access, but a subscription charge is necessary to use the bulk download service which was funded by the Data Analytics and Society CDT[4].

[2] https://www.ukri.org/manage-your-award/good-research-resource-hub/responsible-innovation/

[3] https://shop.epo.org/en/Data-and-services/c/subscriptions

[4] https://datacdt.org/

2 | Literature Review

In this chapter I will outline some of the key approaches which have been taken within the AI & Law literature to case outcome prediction. The chapter will proceed by giving a brief overview of the history of case outcome prediction before exploring the more recent trends of applying techniques from machine learning (ML) and natural language processing (NLP). Finally, work concerning the application of ML to patents will be discussed.

2.1 Symbolic Approaches to Prediction

Since at least the mid-twentieth century there has been an interest in modelling legal reasoning and legal knowledge in a computational manner. Initially symbolic approaches were dominant such as rule-based approaches (modelling statutes and legislation directly), and case-based approaches (CBR) (using judicial precedent), which were set up as competitors but subsequently became recognised as complimentary [13]. One of the more prevalent rules-based approaches consisted of so-called 'expert systems' which are rules-based reasoning systems typically created in conjunction with legal experts, who can help to facilitate the process of decomposing the legal rules and knowledge into a structure which a computer can process [14]. A common problem with expert systems is what happens when the "rules run out" [15], as such systems don't adequately account for the role of judicial precedent in common-law legal systems.

To account for judicial precedent directly, CBR systems focus on the process of

comparing and citing legal cases, in an attempt to mimic the real-world application of precedent. Early systems such as HYPO [16] and CATO [17] use dimensions or legal factors, stereotypical fact patterns, to provide the basis for deriving higher order legal factors which are known to strengthen or weaken a side's argument [18] in order to distinguish between cases based upon their similarities and differences. These factors can be structured hierarchically to generate legal arguments from precedent cases but such systems were not designed for case outcome prediction.

With the introduction of IBP [19], case outcome prediction became possible from CBR systems with the incorporation a logical rules-based layer on top of CATO's factor hierarchy. Further developments of CATO, include ANGELIC [20] which incorporates an Abstract Dialectical Framework [21] corresponding to CATO's factor hierarchies, allowing case outcome prediction. CBR methods designed for prediction are capable of achieving tremendous accuracy scores, for example 96.8% in predicting the outcome of US Trade Secrets Misappropriation cases [20], alongside generating human comprehensible explanations. However, they often lack the scalability required to deal with large amounts of varied data, because creating the systems requires a significant amount of human knowledge-intensive work.

2.2 Machine Learning Approaches to Prediction

Over the last 20 years data-driven techniques have become more prevalent [22, 23] in case outcome prediction due to a greater abundance of legally relevant data which has now been digitised in many jurisdictions: for instance, HUDOC [1] for the European Court of Human Rights (ECtHR). Unlike CBR and expert systems, ML encompasses a variety of techniques which seek to automatically learn patterns and meaningful relationships from a given set of training examples and use these patterns to generate predictions for new heretofore unseen data [24]. Within the literature concerning case outcome prediction, we can distinguish between two

[1] `https://hudoc.echr.coe.int/`

families of techniques that are frequently used: feature-based methods and neural methods.

Neural-based methods are part of deep learning (DL) and leverage the advancements made with artificial neural networks, which have increasingly deep architectures and enhanced learning capacities [25]. The key difference in the context of NLP between feature-based and neural-based approaches is that the former uses explicitly hand-engineered features such as topics or Bag-of-Words, acting as a shallow representation of the text, whereas the latter typically uses word embeddings, e.g. Word2Vec [26], as inputs and creates a deeper representation of the text by automatically learning features at different levels of abstraction.

Feature-based approaches themselves consist of a suite of different ML algorithms with Support Vector Machines (SVM) being the most prevalent. Aletras et al's [1] paper on predicting judicial decisions of the ECtHR, the first work to rely solely on textual content to predict whether a given case has violated an individual Article of the European Convention of Human Rights (ECHR), used SVMs as their model of choice achieving a 79% accuracy in predicting the outcome. Other researchers have attempted to replicate or expand upon the results found in [40] using similar ECHR data. [27] compared various ML methods such as k-NN, logistic regression (LR), random forests (RF) and SVMs on the same ECHR dataset and Articles used in [1] to provide a comparison in this domain between different methods. The results found that SVM methods outperformed the others. However, other research in the ECtHR literature, such as [28], compared a variety of classical ML models and found Gradient Boosting to be dominant, with SVMs not achieving the best performance for predicting the outcome of any individual ECHR Article.

However, neural-based approaches using DL usually out-perform feature-based approaches in a variety of legal domains. For the ECtHR prediction task, Convolutional Neural Networks (CNN) [29] have achieved 82% train accuracy and a variety of BERT, called Hierarchical-BERT [30], achieved 82% F1-score. Both are

higher than any score a feature-based model in the same domain has achieved. But despite the increase in performance possible with these methods they are far more computationally expensive and data-intensive than feature-based approaches.

A key issue with many DL methods is their lack of interpretability. [29] using CNNs does not offer any attempt at explainability, likely due to the difficulty of extracting any relevant information from a complex algorithm such as a CNN. Similarly, BERT models alone, cannot be used to extract the factors which drove their decisions due to the complexity of their architectures [31]. Despite feature-based methods being more inherently interpretable, due to the ability to determine feature importance or interpret the size of the coefficients, many works using feature-based approaches do not even mention the predictors at all [4, 28, 32].

2.3 Patent Analytics

Over the last two decades there has been an increased interest in applying computational analysis techniques to the field of Intellectual Property law, and in particular, for our purposes, patents. The term 'Patinformatics' coined in [33] refers to this process of patent data mining and using automated tools to extract insights and intelligence from patents [34]. A survey by Aristodemou and Tietze [35] identified 4 key areas of active research in Intellectual Property analytics:

- Knowledge Management: Focuses on evaluating the quality of patent document and tools for better managing large quantities of documents
- Technology Management: Includes the identification of emerging technology and technological trends
- Economic Value: For example, identifying the impact of different factors on patent value
- Information Extraction: A diverse category including name-entity recognition from patents, patent landscaping, and technological classification i.e.

categorising patents into technology areas

From [35] one may note the lack of work concerned directly with patent applications, the grant procedure or litigation including infringement. The only work relevant to these topics, that is presented in [35] is [9], within the knowledge management section. [9] created a patent application prediction system with a 'patentability' metric using word-age to determine if a patent would be accepted or rejected within Japan, achieving a rather low score of 60% accuracy. Compared to other areas in Patinformatics such as technological forecasting [34, 36] or technology area classification [37, 38], problems concerning the patent at the application stage have received far less attention in the literature.

Some more recent work has attempted to address this. [8] builds upon the work in [9] to develop a method for predicting the outcome of US patent applications and attempting to identify the reasons for rejection using a CNN + Long Short-Term Memory Network (LSTM) model. They far surpassed [9] achieving 87.7% accuracy but this is limited by using a small sample of 2,539 patent applications from 2013 to 2020 only relating to a single topic of 'electric vehicle'. Similarly, they claimed to be able to predict the reasons for rejection with accuracies ranging from the high 80s to the high 90s but this range was across isolated individual models for each rejection reason, with no performance given for how the models may work when combined, i.e. in ensemble, to generate a single, clear reason for a given case. More recent work in grant prediction [7] for Chinese patents achieved 77% F1-score using XGBoost for over 400,000 applications in 2011 with a variety of primarily patent-level features including numerical, categorical and textual (using TF-IDF). They found the most significant features to be the number of pages in an application, the application success rate of the previous year and the number of inventors for a patent.

Work in litigation prediction has focused on either predicting whether a patent is likely to be litigated or not, or understanding factors which contribute to patent litigation taking place, but neither strand attempts case outcome prediction for

patents which are litigated, such as in an infringement case. For example, [39] found that patents which are litigated have markedly different characteristics to those which aren't especially in regard to their acquired characteristics i.e. transaction history. Whilst [40] used XGBoost to achieve a true positive rate of 75%, with a false positive rate of 25%, for predicting which US patents would be litigated of those granted between 2002-05. However, due to the extremely small percentage of patents which are ever litigated, around 1.1% [40], even these results only amount to showing that 2.9% of the total positives identified by the model will actually be litigated. This is an increase on the unconditional probability of 1.1% but demonstrates the difficulty inherent in rare event prediction.

2.4 Summary

Within this chapter, I have identified a trade-off between scalability and explainability within the case outcome prediction literature present in the differences between ML and CBR approaches. Due to the size of the EPO data an ML approach was chosen to maximise scalability. Whilst the lack of interpretability, and computational cost of neural-based methods, motivated the choice of feature-based methods in the analysis. Furthermore, I have demonstrated that within the literature on AI & Law and patent analytics, case outcome prediction for court related outcomes for patent cases, such as appeals relating to a patent application's grant has not yet been explored.

3 Methods

In this chapter, I will outline the methodology used for the experiments conducted within this work. Figure 3.1 shows a high-level overview of the methodology, proceeding from initial data extraction to evaluation of the models created. The chapter will proceed by describing each step in greater detail. All code described for this, and the following chapter, used Python 3.10.10.

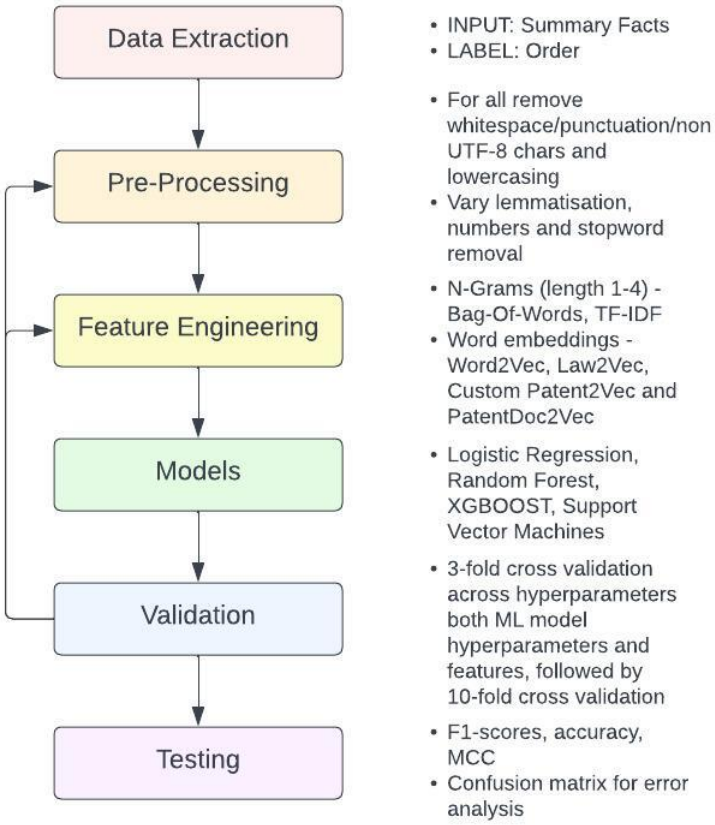


Figure 3.1: Methodology Overview

3.1 Data and Pre-Processing

There are two distinct datasets from the EPO which form the basis of this work: Decisions of the EPO Boards of Appeal[1] and European Patent Full-Text Data for Text Analytics[2] . This section will provide a description of both datasets, the pre-processing steps used and their usage within the work presented.

3.1.1 Decisions of the EPO Boards of Appeal

The dataset consists of the complete set of textual decisions from all subsidiary courts of the EPO Boards of Appeal from 1978-2022, with more than 40,000 decisions. The data is available in machine-readable XML format, which I used alongside the Python standard library XML to parse the relevant parts of the data into a pandas [41] dataframe table, for ease of processing. The data captured ranged from metadata such as court type, language, appeal number and the board of appeal, to the text of the decision, split by sub-heading.

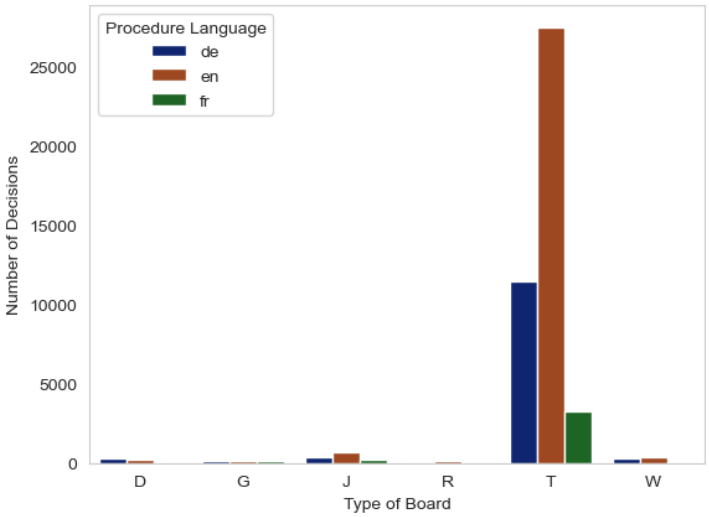
Figure 3.2a shows the distribution of the different Boards of Appeal, motivating the selection of the Technical Board as the focus of this work, since the vast majority of decisions fall within its remit, similarly, we can see that the majority of decisions are published in English. Examining the distribution over time of the dataset, Figure 3.2b, it demonstrates a steady increase in the abundance of decisions made until a peak in 2019. A sharp drop-off can be observed in 2020 and 2022, with 2021 resuming the trend observed prior to 2020. We can hypothesise that the drop in decisions in 2020 was due to the effects of the COVID-19 pandemic, whilst the dip in 2022 is due to the publication date of the dataset being prior to the end of 2022.

It is desirable to constrain the time period of the decisions used for the analysis as the law changes over time and so does the nature of patented inventions. Consequently,

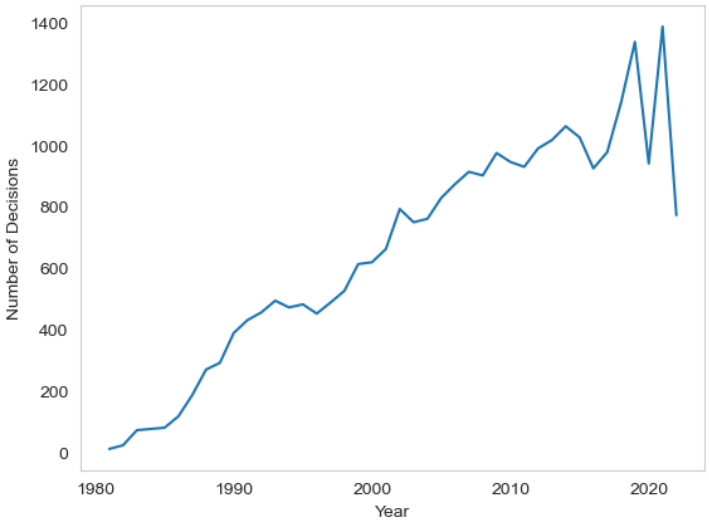
[1] `https://www.epo.org/searching-for-patents/data/bulk-data-sets/boards-of-appeal-decisions.html`

[2] `https://m.epo.org/searching-for-patents/data/bulk-data-sets/text-analytics.html`

the decision was taken to constrain the time period used to decisions rendered after, and including, the year 2000. Constraining the year, as well as the language and the board of appeal, while also removing duplicate cases, gives a complete dataset of 21,426 unique decisions.



(a) Board Type



(b) Decision Year

Figure 3.2: Appeal Data Distributions

Extraction

For any supervised classification task in ML, an input and a target label are required to train the model. As this dataset was not published specifically for the application of ML, the target label (the decision outcome) has not been explicitly provided and

required extracting before training could begin. Similarly, as stated in Sec 1.3, within the Technical Boards of Appeal there are two distinct types of appeals, those from cases previously heard before the Examining Division and the Opposition Division. The decision was taken to separate the data from these two types of appeals, and to use the cases stemming from the Examining Division as the basis of this work due to the additional complexities of Opposition Division appeals in determining who is bringing the appeal within the raw data.

Using simple keyword matching in the SpaCy [42] library for NLP [3] within the Summary of Facts section of the textual content, I created and tested a series of patterns to identify the type of appeal. The types of appeal are Opposition Division appeal, Examining Division appeal, Admissibility and Other. Admissibility corresponds to cases which solely concern admissibility rather than patent validity, and Other corresponds to cases which cannot be classified from the existing patterns.

The patterns were created by manually analysing a set of 50 randomly sampled appeals and splitting them into types based on the keywords used. The patterns were as follows[4]:

- Opposition Division = ["LOWER": 'opposition', "LOWER": 'division']
- Admissibility = ["LOWER":'restricted',"OP":'*',"OP":'*',"LOWER": "FUZZY": "admissibility"]
- Examining Division = ["LEMMA":'refuse',"OP":'*',"LOWER": 'european',"OP":'*',"LOWER":'patent','OP':'*',"LOWER":'application']

The patterns were initially tweaked until 100% accuracy was achieved on the original 50 appeals. To test their generalisability, another random sample of 50 appeals were selected with an initial success rate of 47/50 classified correctly. The 3 misclassified

[3]SpaCy was also used for most of the textual pre-processing in this work

[4]LOWER = lowercase; OP = optional; * = wildcard token; FUZZY = alternate spellings are acceptable; LEMMA = any acceptable lemmatisation of the word

appeals were Examining Division appeals using previously uncaptured keyword patterns. After alteration, the patterns achieved 50/50 and I was satisfied with the accuracy. Furthermore, I checked a small sample of cases in the 'Other' category for glaring omissions in the final dataset but none were observed. After separating out the Opposition Division cases, and excluding Admissibility and Other I was left with 8,121 Examining Division appeal cases.

The final extraction task was to extract the target label using the Order section of the decision which provides the board's outcome. The phrasing of the outcomes are relatively homogeneous and four different types of outcome were observed: the appeal was dismissed, the appeal was rejected for being inadmissible, the decision under appeal was set aside and Other outcomes. These first two outcomes were treated the same as they both result in the original decision by the Examining Division being maintained, thus they were labelled as 'Affirmed', whereas the previous decision being set aside reverses the prior outcome so was labelled as 'Reversed'. The Other outcomes refer to unique or infrequent outcome decisions such as referrals to the Enlarged Board of Appeal, which the patterns could not detect and thus were excluded from the analysis. The patterns were as follows:

- Dismissed = ["LOWER":"FUZZY": "appeal",'OP':'*',"LOWER": "dismissed"]
- Rejected = ["LOWER":"FUZZY": "appeal",'OP':'*',"LOWER": "rejected"]
- Set Aside = ["LOWER":"FUZZY": "appeal",'OP':'*',"LOWER": "set","LOWER": "aside"]

I created the patterns on the same manually analysed random sample of 50 appeals, used to identify case type, before sampling 50 more appeals to test generalisability. For both sets of appeals the pattern achieved 100% accuracy providing confidence in its labelling abilities on this dataset. The distribution of outcomes identified can be found in Figure 3.3.

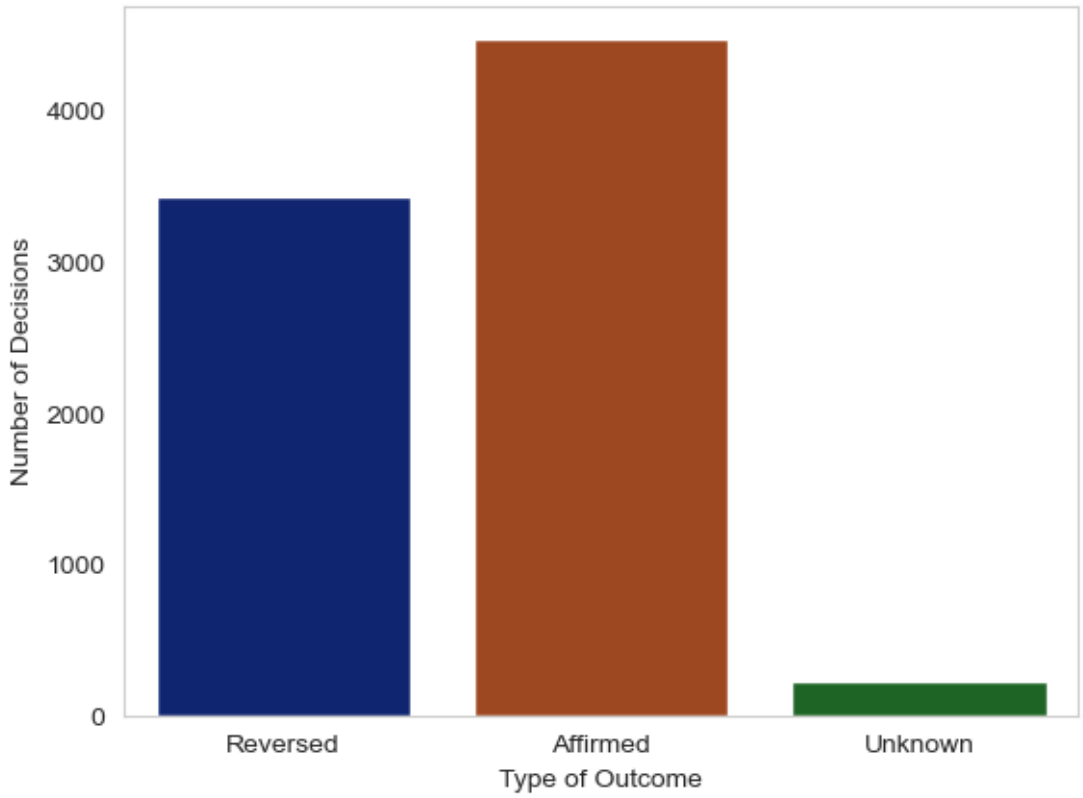


Figure 3.3: Outcome Distributions

Pre-Processing

An EPO appeal decision consists of three main parts: Summary of Facts, Reasons for Decision and Order. The Summary of Facts outlines the facts of what happened in the prior decision, the core arguments the appeal is based on and the desired outcome for the appellants and/or opponents. The Reasons for Decision summarise the rationale from the board for coming to a particular outcome, which is given in the Order section. To predict the outcome of appeal cases *ex ante* we must use only data which was available before the verdict was given. For EPO appeals the only data currently available concerns decisions which have already been rendered, thus to test the possibility of predicting the outcome *ex ante* we must make the same assumption as [1]. that there is enough similarity between parts of the text of the published judgements and the information available prior to the proceedings. To justify this assumption, we exclude the Reasons for Decision section as this is written in hindsight to justify an already decided appeal, but use the Summary of Facts, as that

only summarises information available until the time of the appeal proceedings.

Before the Summary of Facts section is ready to become the input of our ML models, a number of pre-processing steps must be performed to reduce noise and ensure consistency across all appeals. These steps are as follows:

1. Remove: whitespace, punctuation, XML tags, HTTP links, non-alpha characters, individual letters other than 'i' and 'a' as not valid words, the first 35 characters from each case as they are boilerplate and not case-specific
2. Lowercase all text
3. Outcome is labelled 1 for Affirmed and 0 for Reversed
4. Vary the inclusion of numerical characters, stopwords and lemmatisation as pre-processing hyperparameters

3.1.2 European Patent Full-Text Data for Text Analytics

This dataset consists of XML-tagged titles, abstracts, descriptions, claims and search reports of European patent publications from 1978 onwards. The data is split into 40 different files, each averaging around 5-6GB in size and covering patent publications associated with 100,000 publication numbers. This data is used to train the embedding models, more detail given in Section 3.2.2. Due to the size of the total dataset, a subset of 5 files was chosen from this dataset to train the embedding models to ensure the models would train quickly whilst still providing sufficient text from 500,000 publication numbers.

As an individual file from this dataset is quite large, a batch streaming approach was used to load the data in increments of 10,000 publications at a time for training the embedding models. From this a pandas dataframe was created to filter only English entries and exclude HTTP links, to the original documents as PDFs, before a number of pre-processing steps were undertaken:

1. All data including titles, abstracts, claims, descriptions and amendments were

used

2. Step 1 from the EPO Decisions dataset pre-processing was repeated
3. Stopwords were not excluded to provide more accurate context for embeddings i.e. 'not' is a stopword but is contextually important. Similarly, no lemmatisation was performed

3.2 Feature Engineering

ML algorithms rely on numerical feature vectors for their input, so in the case of textual input, words need to be represented numerically. This section will give a brief overview of the approaches used within the work.

3.2.1 Bag-of-Words and TF-IDF

For feature-based approaches, one of the most common representations is a bag-of-words (BOW) approach using n-grams, which consist of n number of tokenized words encoded as a numerical vector. One issue with BOW approaches is that they do not account for the frequency of words which occur in a document, only whether they occur or not. A popular method accounting for frequency, normalizes the word frequency using the term frequency-inverse document frequency measure (TF-IDF). This measure assumes that less frequent n-grams may be more informative than common ones as they will be more characteristic of the specific content of a given document and calculated as[5]:

$$TF-IDF\,(t = term,\, d = document) = TF(t,d) * IDF(t)\; and\; IDF(t) = \log\left(\frac{n}{df(t)+1}\right)$$

[5] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html

3.2.2 Word2Vec and Doc2Vec

Both BOW and TF-IDF suffer a number of drawbacks, including a linear increase in the length of the vector with the number of unique words in the document, increasing model training times as well as failing to capture contextual or semantic meaning within the numerical vector assigned to a word. Word embeddings aim to remedy this under the assumption that a word's meaning can be defined by the words which appear in close proximity to it.

In this work Word2Vec [26] will be used, which uses a feed-forward neural network to train the embeddings using the Skip-gram method of predicting the context words for a given focus word. Word2Vec models can be used out-of-the-box which use news or Wikipedia data to achieve a strong general performance, or they can be pre-trained on custom data to account for the language used in specialist domains. One example in the legal domain is Law2Vec [43] which was pre-trained on 492M words of legal documentation.

The challenge with Word2Vec embeddings for classification is how we allow a collection of embeddings for individual words to represent an entire document. Two ways are used in this work, the first is to average the embedding values for each word to create a document representation, the second is to experiment with a Doc2Vec embedding [44]. Doc2Vevc is similar to Word2Vec but instead of training an embedding for each word, it trains an embedding for each document.

3.3 ML Models and Performance Metrics

This section will provide a brief overview of the 4 types of ML model, and the baseline, which are used for the work, as well as defining the performance metrics used to evaluate them. All models are implemented using the Sci-Kit Learn library [45], other than XGBoost which uses [46].

3.3.1 Logistic Regression (LR)

LR [47] is a relatively simple method, popular in inferential statistics, that can explain or predict a binary outcome using a set of predictors or covariates to find a separating hyperplane.

3.3.2 Support Vector Machines (SVM)

A SVM is similar to LR but it works by finding the optimal hyperplane maximising the distance between the support vectors from each class [48]. The support vectors are the points which fall closest to the line, while the separating hyperplane is always 1-dimension less than the input vector. To prevent overfitting in situations in which the number of features vastly outstrips the number of training examples, an issue common in legal text classification, the so-called 'kernel trick' is used when training to help prevent overfitting and improve performance by mapping the data to a higher-dimensional feature space using a pairwise similarity matrix between all example patterns [48].

3.3.3 Random Forests (RF)

RFs are an extension of decision trees using bootstrap aggregation, also known as bagging, which fits multiple models simultaneously by sampling the dataset repeatedly, with replacement, creating a number of independent models. RFs extend bagging by determining, at random, the features considered to construct the tree with each figure representing a smaller part of the feature space, increasing the diversity of the answers given to better represent the outcome [49].

3.3.4 XGBoost (XGB)

XGB is another extension of decision trees, but rather than bagging it uses boosting [50] in which a decision tree is trained in a repeated procedure with each subsequent iteration giving a higher weighting to the data mis-classified in the previous step,

forcing the model to pay greater attention to the most difficult to classify data points. XGB is a computationally efficient execution of the idea of boosting which has led to significant success, for example in online ML contests on the website Kaggle [51].

3.4 Performance Metrics

To understand whether the ML models are performing better at classification than simple random guessing, we can initiate a baseline classifier using a very simple strategy. For our purposes, the baseline always guesses the majority class present in the dataset.

The three evaluation metrics used within this work are Accuracy, F1-score and Matthew's Correlation Coefficient (MCC). Accuracy is defined as the number of correct classifications divided by the total number of examples. Whilst easy to interpret, it is limited as it fails to account for the cost associated with false positives and false negatives[6]. Consequently, we will also be using the F1-score, defined as the harmonic mean between precision and recall as an additional metric to better account for different error types:

$$Precision = \frac{TP}{TP + FP}$$

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

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

However, recent work in AI & Law [52] has begun to adopt MCC as an even more robust measure for binary classification tasks. This metric, ranging from -1 (worst) to 1 (best), is only high if there are high true positive and negatives, and low false positives and negatives, accounting for all 4 values present in a confusion matrix.

[6]https://deepai.org/machine-learning-glossary-and-terms/f-score

This is more useful in regard to unbalanced datasets, whereas the datasets in this work are balanced, regardless this metric is reported to follow best practice. This can be formally defined as follows [53]:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}$$

3.5 Experimental Setup

This section will outline the setup for the two main experiments forming the basis of the results, as well as the training of the custom word embeddings. The experiments in this work were undertaken on Barkla, part of the High-Performance Computing facilities at the University of Liverpool, UK.

3.5.1 Patent2Vec and PatentDoc2Vec

Custom word and document embeddings, using Word2Vec and Doc2Vec, were created for this project to compare against out-the-box word embeddings to understand whether using patent related data would provide a performance increase. The GenSim library [54] was used for all training. The baseline word embeddings are a pre-trained Word2Vec [26] and Law2Vec [43]. The corpus used for my pre-training was 5 files from the European Patent Full-Text data, as well as a subsection of the EPO Decisions data, namely the training set, further described in Experiment 1[7]. The rationale for also incorporating EPO Decision data is due to the lack of legal language used within the patent document data, so to avoid many key terms in the decisions not having a vector representation, the decision data was used to supplement the training. The corpus totalled 4.15B non-unique words (tokens) of which 3.06B words were actually used for training after ignoring unknown words

[7]Despite only using the training set from experiment 1 for training the embeddings, which are the same embeddings used in experiment 2 which has a different train-test split, I do not anticipate data leakage to be a concern. This is due to the nature of Word2Vec simply learning semantic similarity rather than associating the learnt words with any kind of target objective, such as the outcome.

and trimming sentence length.

The parameters used for the custom Word2Vec model, hereafter referred to as Patent2Vec, followed some of the same hyperparameters used for Law2Vec in [43], such as a threshold of 10-word minimum occurrences to have a trained embedding, a 5-word context window and it is trained for 3 iterations. However, unlike Law2Vec which creates 200-d embeddings, Patent2Vec uses 300-d embeddings due to the abundance of data used compared to Law2Vec (which uses 492M tokens). Patent2Vec is trained using sentences as the input. The Doc2Vec implementation, hereafter referred to as PatentDoc2Vec, uses the same hyperparameters but uses the full document as the input.

3.5.2 Experiment 1

The aim of the first experiment is to understand what level of predictive performance is possible with this dataset using random sampling of the train and test sets. The first step is to balance the dataset before performing a stratified split into the training set (90%) and the test set (10%). A balanced training set helps to ensure the dataset does not have a bias towards the majority class. However, if we leave the test set balanced, when the real data is imbalanced, this could cause us to overestimate our predictive performance on an unrealistic test set. Following work such as [28, 30], I have chosen to create a realistic test set mimicking the original outcome distribution observed in the data, this can be seen in Table 3.1.

The first experiment conducts a 3-fold cross-validated[8] randomized grid search for each combination of hyperparameters, except the final 3 listed in Table 3.2, for each combination of non-embedding input (BOW, TF-IDF) and model (LR, SVM, RF, XGB). For the final 3 hyperparameters (stopwords, numbers and lemmatisation) all 8 possible combinations are attempted for each input + model combination, resulting in 800 models being trained for each input + model combination. For the embedding

[8]In each of the 3 iterations the training set is split into thirds with 66.6% used for training and 33.3% used for validating. The validated performance is then reported.

Table 3.1: Train and Test set distributions

	Training Data			Test Data		
	Affirmed	Reversed	**Total**	Affirmed	Reversed	**Total**
Experiment 1	3086	3086	**6172**	443	343	**786**
Experiment 2: 2019-20	2877	2877	**5754**	538	317	**855**
Experiment 2: 2021-22	2877	2877	**5754**	416	235	**651**
Experiment 2: 2019-22	2877	2877	**5754**	954	552	**1506**

inputs (Word2Vec, Patent2Vec, Law2Vec, PatentDoc2Vec) a grid search is used, testing all possible model hyperparameter combinations, as many of the pre-processing hyperparameters are omitted since they are only relevant for n-grams. Similarly, of the final 3 listed hyperparameters only stopwords are varied.

The first stage of training uses 3-fold cross-validation to search over a wide hyperparameter space in a more computationally efficient manner, but this does not provide robust enough results to be confident in the generalisation ability of the models in order to select the best model. To mitigate this a second step is performed in which 10-fold stratified cross-validation is repeated 10 times for each of the best hyperparameter combinations for each input + model combination.

3.5.3 Experiment 2

The aim of the second experiment is to test the prediction of only future cases, following [2], by creating more realistic train and test sets. The issue with experiment 1 is that a case from 2019 may form part of the training set, and be used to predict the outcome, in the test or validation set, of a case from 2001, which fails to mirror the time series nature of the application of legal process. To mitigate against this issue, the first step is to divide the training set and test sets into cases from different years. The training set is from 2000-2018, and the test set from 2019-2022, with further divisions also created in the test set to monitor whether there is an observable degradation in performance the further away the test set is in time from the training set, as observed in [2]. These further splits are 2019-20 and 2021-22, as can be seen in Table 3.1. Unlike experiment 1, the test set distributions are unaltered to mimic the

real case distributions present in those years and understand how successful the model would have been had it been deployed before the test set began.

The hyperparameter search process is the same as experiment 1, including the 10-fold cross validation step. The crucial difference is that rather than using a cross validation function which randomly splits the data, TimeSeriesSplit() in Sci-Kit Learn[9] is used, which splits the data according to the order they are given to the model (ascending date order), to ensure that the validation stage tests for the models which deliver the best future prediction capabilities. In addition, for the 10-fold cross validation step, the procedure is not repeated 10 times, as in experiment 1, since the time series split function always uses the same splits of the data unlike the randomly stratified cross-validation in experiment 1.

[9] `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.TimeSeriesSplit.html`

Table 3.2: Hyperparameters

Name	Value	Description
N-Gram Parameters		
ngram_range	(1,1),(1,2),(1,3),(1,4),(2,2), (2,3),(2,4),(3,3),(3,4),(4,4)	Length of the n-grams
norm	None, 'L2'	Normalisation term for vectors
min_df	2, 5, 10	Minimum document frequency the terms must appear in to be included
use_idf	True, False	Use Inverse Document Frequency weighting (TF-IDF)
Model Parameters		
C	0.1, 1, 10, 100	SVM and LR: Regularisation strength
solver	'lbfgs', 'sag'	LR: Algorithm to use in the optimisation problem
penalty	None, 'L2'	LR: The norm of the penalty parameter
max_iter	100, 250, 500	LR: Maximum iterations for the solver to converge
n_estimators	100, 200, 300	RF and XGB: Number of trees in the forest
max_features	'sqrt', 'log2'	RF: Number of features to consider when looking for the best split
max_depth	10, 50, 100, None	RF: Maximum depth of the tree
num_boost_round	100, 200, 300	XGB: Number of boosting rounds
learning_rate	0.01, 0.02, 0.05	XGB: Step size shrinkage used in update to prevent overfitting
gamma	0.0, 0.1, 0.2	XGB: Minimum loss reduction to make a further partition on a leaf node of the tree
Pre-Processing Parameters		
stopwords	True, False	Include or exclude stopwords
lemmatisation	True, False	Perform lemmatisation or not
numbers	True, False	Include or exclude numerical tokens

4 | Results and Discussion

In this chapter, I will describe the results achieved across the different experiments conducted. First, I will proceed by analysing the custom word and document embeddings trained, before discussing the results of experiments 1 & 2, and finally interpret these results in regard to performance and explainability.

4.1 Word Embeddings

The trained custom word and document embeddings, Patent2Vec and PatentDoc2Vec, have a final vocabulary size of 387,919 unique words, compared to the vocabulary of 169,439 words in Law2Vec [43]. No formal procedure for evaluating the quality of word embeddings exists, rather qualitative approaches are often used to provide a sense-check that the embeddings learned are sensible for the domain. One such approach to qualitatively evaluate embeddings is using T-SNE [55], a dimensionality reduction technique that can project a nonlinearly separable high dimensionality space into a two-dimensional representation. Using this we can visualise words which are similar to each other within the embeddings to better understand the representations.

Figure 4.1 shows an example of this using five domain appropriate words (patent, law, appeal, outcome, appellant), which appear frequently within the classification task's training data. Some of the chosen words are represented as one would expect i.e. patent is most similar to words such as 'provisional' or 'application', while others have more unexpected similarities i.e. law is most similar to 'henry' or 'boyle', and

outcome is most similar to words such as ‘chemotherapy’ or ‘prognoses. The word ‘law’ appears to be associated far more with laws as defined within natural science such as Henry’s law and Boyle’s law, than the legal system. Whereas, ‘outcome’ is associated with medical and clinical outcomes rather than legal decisions. This is unsurprising given the training data consisted of a vast number of patent applications, with many inventions clearly having medical applications or referencing aspects of scientific theory informing the design or innovation.

This selection of similar words is by no means representative of the large vocabulary, but the similarities generated are plausible given the context of the data but may fail to capture the meaning of words which have a legal meaning distinct from that in other contexts.

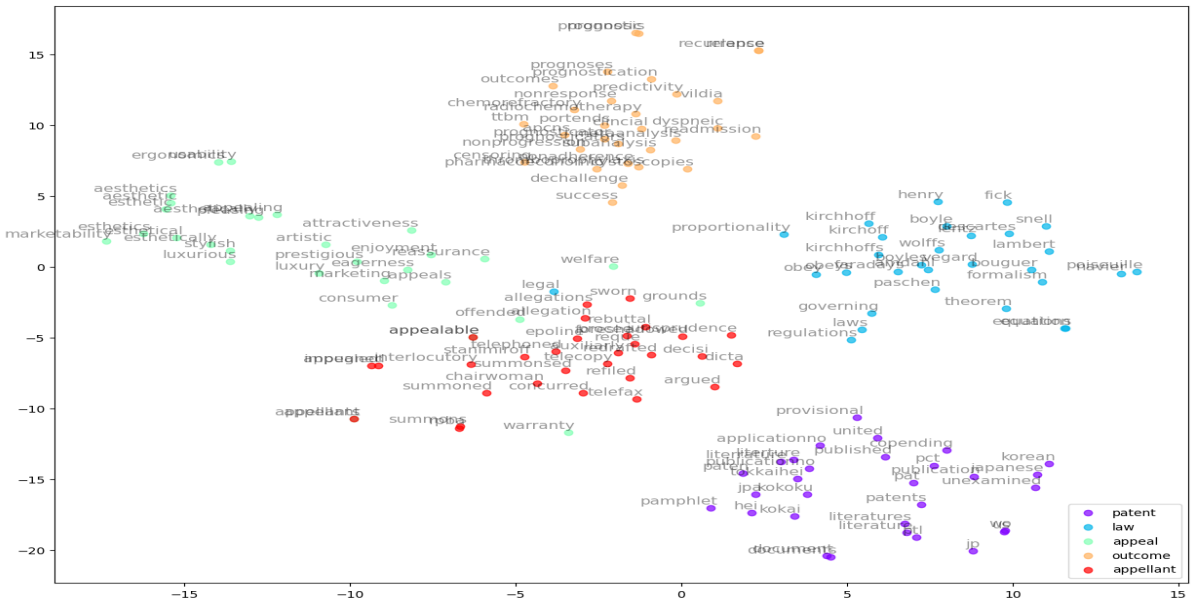


Figure 4.1: T-SNE

4.2 Experiments

The number of models trained for each experiment is 7,076 (3 CV) and 2,400 (10 CV), for a total of 18,952 models between both experiments. Tables 6.1 and 6.2 show the 10-fold cross validated results of both experiments on the training set, reporting the mean scores of the performance metrics stated in Section 3.4, although experiment 2

reports a weighted-average of these metrics. The weighted-average is used in experiment 2 due to the nature of the time series split in the cross-validation procedure. To preserve the time series the initial splits are smaller than the later splits as each split permits a greater number of training cases, across a larger period of time. The weighted average is calculated for n weights and x splits:

$$Weights : w_i = \frac{i}{n} \; for \; i = 1, ..., n.$$

$$Weighted Average : \sum_{i=1}^{n} w_i * x_i$$

Table 4.1: Patent Refusal: Experiment 1 (2 d.p.) - Mean and Standard Deviation of 10-fold Cross-Validation

	N-Grams			**TF-IDF**			**Word2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	85.87 ±1.38	85.78 ±1.39	71.77 ±2.75	86.08 ±1.38	85.90 ±1.39	72.21 ±2.76	**71.40** ±1.46	**71.31** ±1.63	**42.84** ±2.93
LR	85.73 ±1.28	85.51 ±1.34	71.50 ±2.56	85.55 ±1.36	85.51 ±1.35	71.14 ±2.72	71.38 ±1.59	71.25 ±1.72	42.78 ±3.17
RF	85.07 ±1.32	85.06 ±1.34	70.16 ±2.66	85.02 ±1.38	84.79 ±1.46	70.10 ±2.76	67.61 ±1.78	65.60 ±2.04	35.48 ±3.57
XGB	**86.47** ±1.40	**86.37** ±1.45	**72.99** ±2.81	**86.58** ±1.19	**86.47** ±1.22	**73.19** ±2.38	70.76 ±1.56	70.18 ±1.74	41.58 ±3.12

	Patent2Vec			**Law2Vec**			**PatentDoc2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	**73.16** ±1.46	**73.17** ±1.59	**46.35** ±2.92	**70.25** ±1.72	**70.15** ±1.89	40.54 ±3.43	72.24 ±1.60	71.76 ±1.79	44.55 ±3.20
LR	73.11 ±1.57	73.11 ±1.69	46.26 ±3.15	70.08 ±1.57	69.88 ±1.73	40.19 ±3.16	72.42 ±1.50	72.04 ±1.66	44.90 ±3.00
RF	69.51 ±1.71	67.85 ±1.99	39.25 ±3.42	67.58 ±1.85	65.62 ±2.19	35.43 ±3.71	70.78 ±1.75	69.88 ±1.95	41.65 ±3.49
XGB	72.47 ±1.57	71.97 ±1.78	45.00 ±3.15	70.24 ±1.72	69.58 ±1.95	**40.56** ±3.44	**73.34** ±1.66	**72.87** ±1.91	**46.73** ±3.30

Table 4.2: Patent Refusal: Experiment 2 (2 d.p.) - Weighted Average and Standard Deviation of 10-fold Time Series Split Cross-Validation

	N-Grams			**TF-IDF**			**Word2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	84.23 ±3.07	85.14 ±5.56	67.76 ±6.05	84.17 ±3.47	85.13 ±5.88	67.69 ±6.67	68.17 ±2.06	70.11 ±4.69	35.53 ±3.82
LR	84.50 ±2.99	85.43 ±5.38	68.31 ±5.82	**84.42** ±3.22	85.27 ±5.56	**68.19** ±6.25	**68.80** ±2.07	**70.65** ±4.19	**36.78** ±3.77
RF	82.27 ±3.93	83.03 ±7.92	63.82 ±7.45	82.53 ±3.95	82.99 ±7.48	64.59 ±7.08	64.33 ±2.21	62.15 ±7.97	29.85 ±3.98
XGB	**84.80** ±2.57	**85.76** ±4.61	**68.95** ±4.94	84.32 ±2.83	**85.32** ±5.19	67.89 ±5.63	67.81 ±2.54	68.45 ±6.30	35.09 ±4.74

	Patent2Vec			**Law2Vec**			**PatentDoc2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	**71.20** ±1.78	**73.54** ±4.94	**41.07** ±3.13	67.29 ±1.74	69.64 ±5.33	33.15 ±2.72	69.72 ±2.02	71.41 ±5.01	38.73 ±4.13
LR	71.10 ±1.68	73.43 ±4.80	40.87 ±2.99	67.03 ±1.55	69.55 ±4.85	32.74 ±3.06	**70.39** ±2.12	**72.11** ±4.79	**39.97** ±4.15
RF	65.80 ±2.52	64.31 ±8.27	32.28 ±4.47	64.06 ±2.76	61.26 ±8.76	29.82 ±5.02	67.42 ±2.46	66.66 ±9.48	34.87 ±4.05
XGB	69.45 ±2.22	70.19 ±6.84	38.17 ±5.32	**67.45** ±2.22	**67.87** ±7.23	**34.33** ±3.17	69.22 ±2.66	69.76 ±7.09	38.00±4.67

Both experiments demonstrate strong results in the binary classification task with experiment 1 producing higher scores overall (86.47% F1) than experiment 2 (85.76% F1). We can compare this to a majority baseline, which achieves 56%, meaning that the models are outperforming the baseline by a large margin. Experiment 1 outperforming experiment 2 is unsurprising as experiment 1 is trained on a higher volume of decisions, and the time series split procedure in experiment 2 means some splits are being trained with a very low number of cases, so despite mitigating this with a weighted average, a drop-in performance for the cross-validation stage was expected[1].

An interesting pattern emergent across both experiments is the dominance of BOW text representation over the word embeddings. For example, in experiment 1 the best BOW approach achieves 85.76 F1-score, compared to the best word embedding (Patent2Vec) which achieves 73.17% F1-score, a 12% performance difference. This may have occurred because word embeddings only represent individual words, thus to represent an entire document you have to combine the individual word embeddings into a single representation. The choice to average the word embeddings, could, across a large document lose crucial information in representing the semantic space, causing a decrease in performance.

However, this issue should not plague PatentDoc2Vec, which is trained to create document representations from scratch, yet PatentDoc2Vec (72.11% F1 in experiment 1) actually performs worse than Patent2Vec (73.54% F1 in experiment 1) across both experiments. PatentDoc2Vec's lack of performance might be due to the difference between the training data for PatentDoc2Vec and the actual data represented in the experiments. The vast majority of documents fed to PatentDoc2Vec were sections of patent applications, such as the claims, with only a small minority actually being the 'summary of facts' sections from decision documents. Despite the nature of the majority of the input in Patent2Vec and PatentDoc2Vec being substantively different

[1]The nature of the time series split is also why the standard deviation for experiment 2's results is far greater than experiment 1.

to the training data, a statistically significant boost in performance over the other word embeddings (Word2Vec and Law2Vec) can be observed from the confidence intervals of the results across both experiments in Fig 6.6. While the difference in performance is small between the custom and off-the-shelf embeddings, this result still supports the importance of training domain specific embeddings.

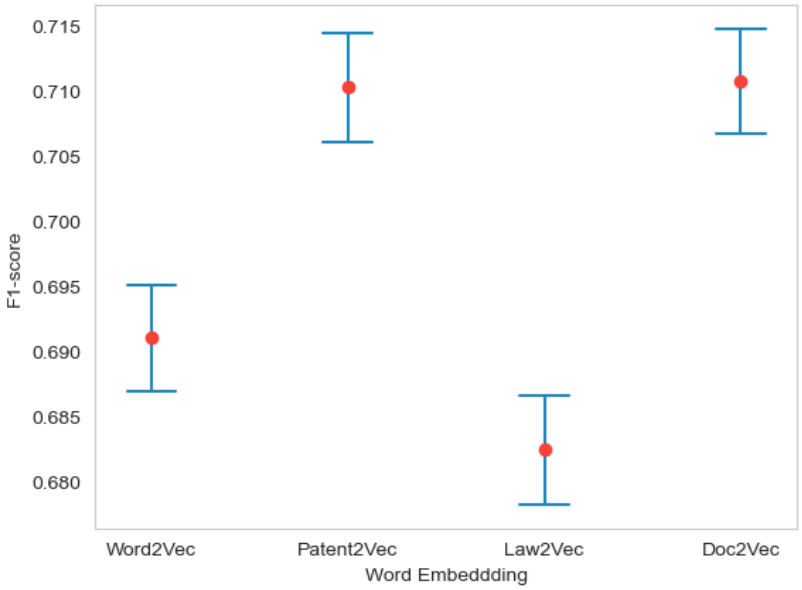


Figure 4.2: Word Embeddings Confidence Intervals

Another emergent pattern is that XGBoost tends to achieve the best results across the different inputs, and for both experiments is the classifier for the best performing model. However, the second best performing classifier is the SVM, achieving the best F1-scores in experiment 1 for all embedding methods other than PatentDoc2Vec, with LR also performing well in experiment 2. Whereas the RFs are consistently the worst performing classifier across all input/model combinations. Though it may be noted that overall all models achieve similar results. The best models and their associated hyperparameters can be observed in Table 6.5.

Table 4.3: Best models and Their Selected Hyperparameters

Experiment	Model	Model Hyperparameters	Input	Input Hyperparameters	Stopwords	Lemma	Numbers
1	XGBoost	num_boost_round: 200 n_estimators: 300 learning_rate: 0.05 gamma: 0.1	TF-IDF	use_idf: True norm: L2 ngram_range: (1,4) min_df: 5	False	True	False
2	XGBoost	num_boost_round: 300 n_estimators: 300 learining_rate: 0.05 gamma: 0.2	Bag of Words	use_idf: False norm: None ngram_range: (1,4) min_df: 10	True	True	True

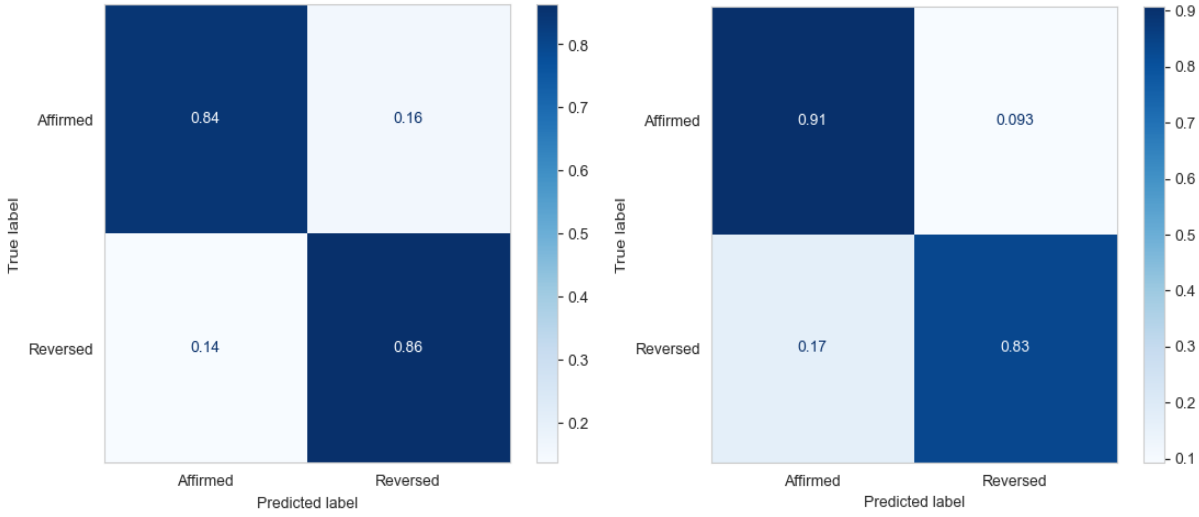
Table 4.4: Test Set Results

	Patent Refusal			
	Acc	**F1**	**MCC**	**Baseline**
Experiment 1	85.24	85.09	70.33	56.36
Experiment 2: 2019-2020	**87.95**	**87.07**	**74.13**	62.92
Experiment 2: 2021-2022	84.49	83.41	66.91	**63.90**
Experiment 2: 2019-2022	86.45	85.48	70.97	63.43

Table 4.4 shows the results of applying the best models, from Table 6.5, to the test data. Initially we can observe that the baselines of each experiment are clearly outperformed by the models and that the scores of the models align closely to the cross-validated training data results, demonstrating the ability of these models to generalise to unseen data. In regard to the robustness of the models we can see that in addition to high accuracy, they achieve high F1 and MCC scores. Furthermore, we can check the robustness by examining the confusion matrices for each experiment's test data to understand whether the models are bias towards misclassifying either affirmed or reversed outcome decisions more frequently. Examining Fig 4.3 we observe that for experiment 1 there is a slight bias towards misclassifying affirmed cases as reversed cases, whereas in all parts of experiment 2 the opposite holds, in which reversed cases are more frequently misclassified than affirmed cases. Generally, the difference is not too great between the misclassification of both outcomes, and the discrepancies that do exist, especially in experiment 2, may be due to the imbalance of the test sets.

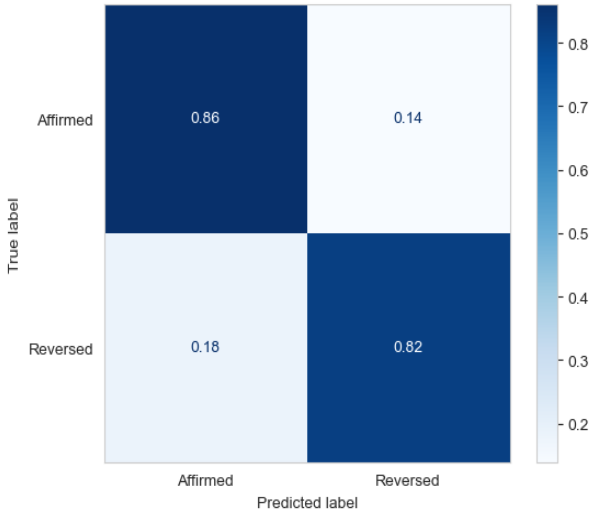
We can also observe that the results of the variants of experiment 2, other than 2021-22, outperform experiment 1. These results are contrary to the observed degradation in performance with the future prediction task compared to the random sampling approach, as is the case in [2]. One reason the future prediction results may be so strong is due to the use of the time series split for cross validation, meaning we are only picking models which are good at predicting future outcomes. Another may be the stability of a domain like patents and the criteria for their grant by the EPO Boards of Appeal, which have received little substantive change since its inception.

However, a small degradation is observed between the 2019-20 test set and the 2021-22 test set. This may support the observation in [2] that the further away the test set is from the training data the more performance decreases, or it could be an artefact of the smaller test set, as 2019-20 has 200 less cases than 2021-22.

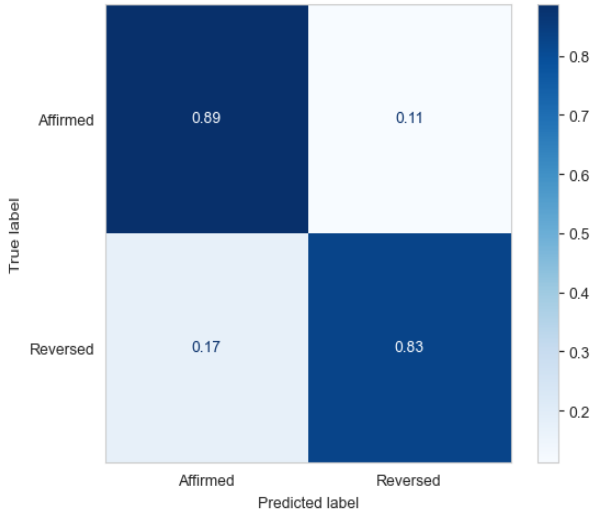


(a) Experiment 1

(b) Experiment 2: 2019-2020



(c) Experiment 2: 2021-2022



(d) Experiment 2: 2019-2022

Figure 4.3: Confusion Matrices of Test Data Results

4.3 Interpretation

In order to better understand the results of the models we can examine the most important features in deciding the outcome of an appeal. We will apply this to the results of experiment 1 to understand the trends and key factors across the entire

timespan of the dataset. Examining Fig 6.3, we can observe that many of the top features are administrative in nature such as 'the notification of', 'proceeding be appoint in' or 'the present application', rather than relating to more substantive legal questions about whether the given invention had, for example an inventive step. There are also a number of outliers to this trend with phrases such as 'five new' (rated as the most important feature) and 'or organ', terms whose significance is difficult to understand without wider context. A limitation of XGBoost's feature importance is that it does not give an indication of which outcome these features favoured.

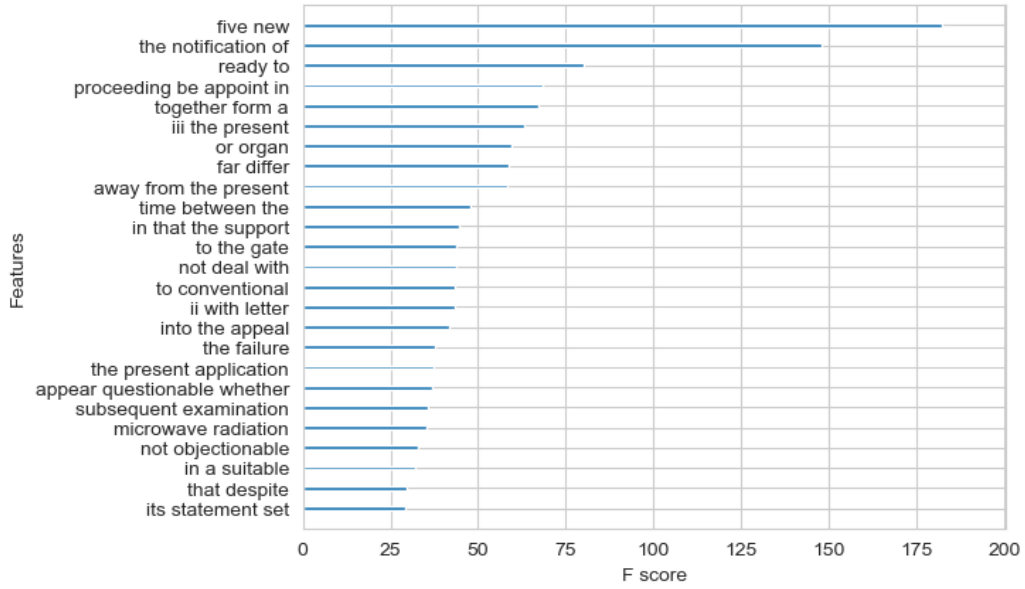


Figure 4.4: XGBoost Feature Importance

To observe the factors which were most significant for a particular outcome we can look at the coefficients of the SVM model in which the highest positive factors correspond to the affirmed outcome and the lowest negative factors to the reversed outcome. Fig 6.4 shows a similar trend to the XGBoost feature importance, in that administrative factors are the most prevalent for both outcomes, but there is a greater correlation with legal factors present. Significant to the affirmed outcome are phrases such as 'the claims do not', 'to recite the claims' and 'an inventive step on'; and for the reversed outcome there is 'since it was obvious' and 'skilled person has no'. For the affirmed outcome the consideration of 'claims' seems sensible as the 'claims' section of a patent is a crucial section in assessing its validity, and the 'inventive step'

a part of the criteria outlined in Section 1.3. The more surprising features are those for the reversed outcome, in which the patent is granted after appeal, as phrases such as 'since it was obvious' and 'skilled person has no' appear to relate to the part of the criteria concerning whether the invention would be obvious to someone skilled in the art and if it was obvious, the patent ought to be refused. It is unsurprising these phrases would appear in the summary of facts since it may describe why the patent was initially rejected, but the model using the 'obviousness' of a patent to decide that it should be granted could be a potential misapplication of the patent grant procedure.

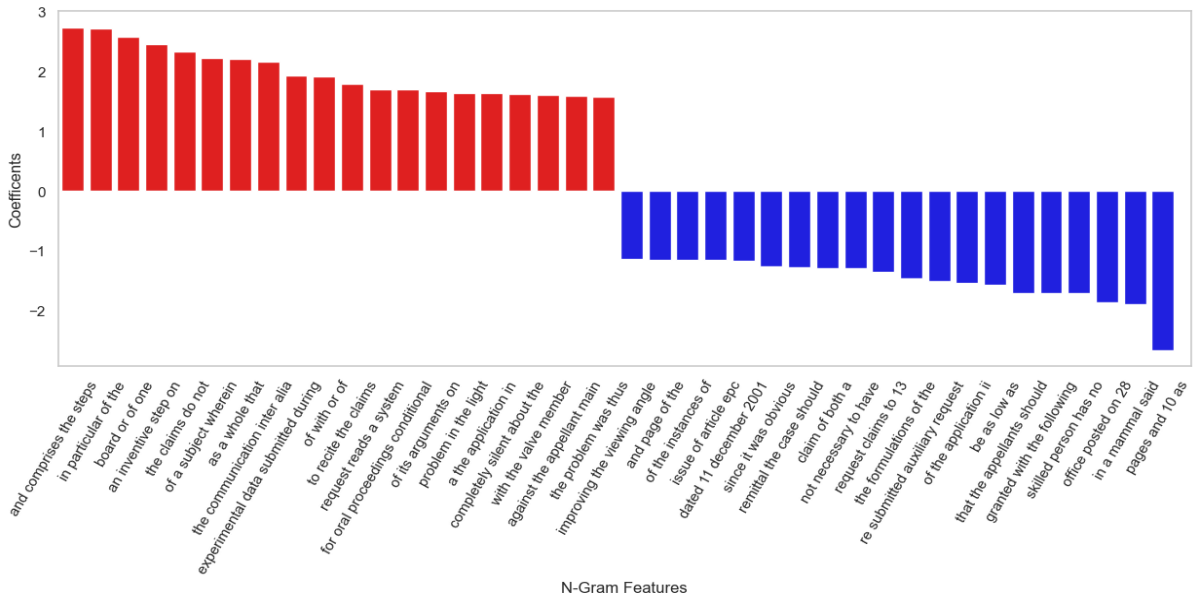


Figure 4.5: SVM Feature Importance

From this analysis, we can conclude that despite the strong performance of the models in regard to performance, the factors the models uses to decide do not sufficiently correspond to the real legal factors, or the correct process of legal reasoning within the domain. This is a problem which has previously been identified for NLP approaches in case outcome prediction [56].

5 Conclusion

In this chapter I will summarise the experiments and results achieved, before outlining some limitations and directions for future work concerning this topic and dataset.

5.1 Summary

The aim of this project was to assess the feasibility of applying AI techniques to decisions from the European Patent Office's Boards of Appeal relating to the Examining Division. Two main experiments were conducted to achieve this testing the performance when randomly sampling the data, compared to year-stratified data for future prediction, using a variety of models, inputs (including custom word embeddings) and a large hyperparameter search to find the best models. The project was successful in showing the viability of this domain and dataset for further study, achieving strong predictive performance of 85% accuracy in experiment 1 and 86% in experiment 2, far higher than the majority baseline of ≈60%. Further analysis in regard to interpretability did demonstrate a lack of correlation between legally relevant factors and the most important features, highlighting the need to ensure that decision-making systems conform to legal reasoning.

5.2 Limitations

A few of the limitations of this work are:

- The 'summary of facts' is written after the appeal hearing takes place, so despite this section attempting to only explain what happened before the hearing, it is inevitably susceptible to bias, framing the summary around the outcome of the appeal
- The BOW approaches (N-grams and TF-IDF) were very slow to train due to the large size of the vectors, even with varying the minimum frequency within the corpus for an n-gram to be included. Dimensionality reduction techniques, such as Principal Component Analysis, were considered but would have an adverse effect on the interpretability of the bag-of-words models so their inclusion was rejected for this project
- This work does not consider appeals from outcomes of the opposition division, which would be required to give a full analysis into the feasibility of applying ML to appeals relating to a patent's validity

5.3 Future Work

In regard to further work, it would be important to perform a more in-depth interpretability analysis using explainable AI methods such as SHAP (SHapley Additive exPlanations) [57], to help understand why certain outcomes may have been predicted by the model for specific cases. Special attention should also be paid to the correspondence between explicit legally relevant factors and the reasoning undertaken by proposed ML models. Furthermore, using more advanced state-of-the-art methods such as BERT, Hierarchical Attention Networks or Convolutional Neural Networks, may be beneficial in increasing the predictive power achievable.

6 APPENDIX

Table 6.1: Patent Refusal: Experiment 1 (2 d.p.) - Mean and std dev of 10-fold cross-validation

	N-Grams			**TF-IDF**			**Word2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	85.34 ±1.14	85.21 ±0.98	70.72 ±2.27	85.63 ±1.26	85.45 ±1.18	71.31 ±2.52	71.63 ±1.44	**71.53** ±1.77	43.29±2.88
LR	85.29 ±1.57	85.11 ±1.56	70.61 ±3.14	85.34 ±1.28	85.15 ±1.19	71.72 ±2.57	**71.66** ±1.26	71.49 ±1.57	**43.35** ±2.51
RF	84.79 ±0.83	84.74 ±0.95	69.60 ±1.66	85.00 ±1.31	84.85 ±1.31	70.03 ±2.61	67.22 ±0.99	65.10 ±1.20	34.71 ±2.00
XGB	**86.68** ±1.09	**86.58** ±1.16	**73.39** ±2.18	**86.88** ±0.85	**86.81** ±0.79	**73.78** ±1.72	70.06 ±1.31	69.60 ±1.61	40.15 ±2.60

	Patent2Vec			**Law2Vec**			**PatentDoc2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	**73.16** ±1.46	**73.17** ±1.59	**46.35** ±2.92	**70.25** ±1.72	**70.15** ±1.89	40.54 ±3.43	72.24 ±1.60	71.76 ±1.79	44.55 ±3.20
LR	73.07 ±1.82	73.03 ±1.99	46.18 ±3.63	69.75 ±1.34	69.54 ±1.45	39.52 ±2.68	73.14 ±1.71	72.90 ±1.70	46.30 ±3.42
RF	69.30 ±1.05	67.51 ±1.13	38.83 ±2.11	67.43±1.33	65.54 ±1.27	35.09 ±2.72	70.14 ±1.77	69.13 ±1.93	40.39 ±3.55
XGB	72.23 ±1.46	71.78 ±1.51	44.49 ±2.92	69.88 ±1.13	69.28 ±1.30	**49.83** ±2.30	**73.27** ±1.51	**72.91** ±1.61	**46.56** ±3.02

Table 6.2: Patent Refusal: Experiment 2 (2 d.p.) - Weighted Average 10-fold TimeSeriesSplit

	N-Grams			**TF-IDF**			**Word2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	83.66 ±3.55	84.76 ±5.30	66.61 ±6.97	82.88 ±4.04	84.08 ±5.52	65.02 ±7.77	68.46 ±2.13	**70.56** ±4.51	36.07 ±4.04
LR	83.78 ±3.61	84.90 ±5.12	66.88 ±7.12	83.02 ±3.88	84.23 ±5.51	**65.38** ±7.53	**68.52** ±2.27	70.45 ±4.64	**36.20** ±4.37
RF	82.21 ±3.51	82.81 ±7.99	63.86 ±6.43	81.99 ±3.80	82.57 ±8.05	63.35 ±6.84	64.49 ±2.67	61.07 ±8.42	28.33 ±4.23
XGB	**85.34** ±3.83	**86.11** ±6.41	**70.02** ±7.65	**84.15** ±3.20	**85.07** ±5.69	67.61 ±6.25	67.63 ±2.18	68.23 ±6.81	34.68 ±4.16

	Patent2Vec			**Law2Vec**			**PatentDoc2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	70.94 ±2.42	**73.45** ±5.31	40.55 ±4.41	64.73 ±3.14	67.29 ±5.57	28.19 ±5.34	**69.79** ±1.88	**71.66** ±4.68	**38.88** ±3.89
LR	**70.96** ±1.72	73.34 ±4.75	**40.57** ±2.90	67.08 ±1.60	**69.47** ±4.91	32.80 ±2.84	69.46 ±1.89	71.45 ±4.19	37.88 ±3.74
RF	65.01 ±2.07	63.49 ±7.97	30.74 ±3.57	63.63 ±2.51	60.72 ±9.11	28.97 ±4.11	66.92 ±2.27	65.84 ±9.62	34.02 ±3.53
XGB	68.83 ±2.37	69.82 ±6.23	36.88 ±4.67	**67.73** ±2.66	67.86 ±7.93	**35.10** ±4.45	69.72 ±2.38	70.29 ±7.40	38.72±4.32

Table 6.3: Opposition Division: Experiment 1 (2 d.p.) - Mean and std dev of 10-fold cross-validation

	N-Grams			**TF-IDF**			**Word2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	70.74 ±1.08	71.27 ±1.11	41.51 ±2.17	70.29 ±1.11	70.87 ±1.11	40.63 ±2.20	62.60 ±1.34	62.15 ±1.45	25.22 ±2.70
LR	70.86 ±1.68	71.30 ±1.67	41.74 ±3.36	70.55 ±0.91	71.05 ±1.01	41.13 ±1.82	62.90 ±1.55	62.43 ±1.94	25.82 ±3.10
RF	76.56 ±1.51	77.00 ±1.56	53.18 ±3.03	76.83 ±0.96	77.28 ±0.79	53.72 ±1.89	65.49 ±1.51	66.61 ±1.84	31.08 ±3.06
XGB	**79.90** ±0.94	**80.66** ±0.83	**60.01** ±1.84	**79.14** ±1.15	**79.77** ±1.14	**58.42** ±2.29	**66.50** ±1.41	**67.78** ±1.45	**33.13** ±2.85

	Patent2Vec			**Law2Vec**			**PatentDoc2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	62.76 ±2.25	62.28 ±2.68	25.54 ±4.48	61.90 ±1.25	61.14 ±1.39	23.81 ±2.49	62.77 ±1.78	63.07 ±1.53	25.57 ±3.55
LR	62.81 ±2.26	62.45 ±2.43	25.64 ±4.52	61.94 ±1.73	61.23 ±2.10	23.89 ±3.46	62.76 ±1.70	62.91 ±1.47	25.54 ±3.40
RF	66.21 ±1.57	66.55 ±1.57	32.44 ±3.14	**65.11** ±1.91	**65.90** ±2.03	**30.28** ±3.83	**69.25** ±1.32	71.54 ±0.98	39.02 ±2.54
XGB	**67.38** ±2.02	**68.07** ±2.16	**34.82** ±4.05	65.26 ±1.63	66.44 ±1.37	30.61 ±3.22	69.01 ±1.09	**70.89** ±0.74	**38.34** ±2.07

Table 6.4: Opposition Division: Experiment 2 (2 d.p.) - Weighted Average 10-fold TimeSeriesSplit

	N-Grams			**TF-IDF**			**Word2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	68.88 ±1.68	69.39 ±2.11	37.94 ±3.30	68.17 ±1.89	68.05 ±2.79	36.50 ±3.72	62.16 ±3.11	61.63 ±3.18	24.39 ±6.24
LR	68.79 ±1.98	69.10 ±2.25	37.71 ±3.93	67.96 ±1.96	68.01 ±2.26	36.01 ±3.91	62.46 ±2.60	61.77 ±3.00	25.01 ±5.20
RF	73.17 ±2.60	72.12 ±2.93	46.55 ±5.30	73.53 ±3.00	72.66 ±3.37	47.19 ±6.02	64.02 ±3.34	61.45 ±3.95	28.40 ±6.91
XGB	**76.74** ±3.29	**77.48** ±3.09	**53.65** ±6.53	**75.91** ±2.79	**76.72** ±2.99	**51.94** ±5.56	**65.10** ±3.45	**64.48** ±4.22	**30.30** ±6.95

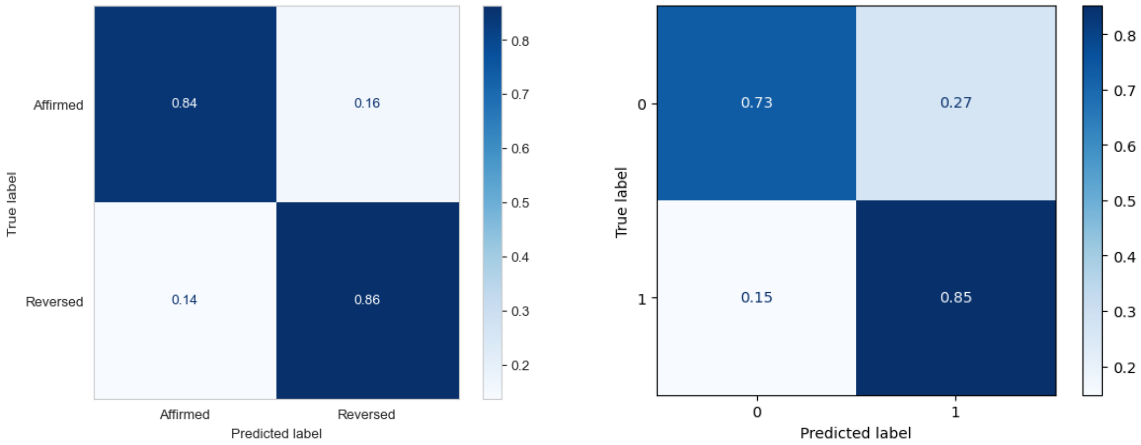
	Patent2Vec			**Law2Vec**			**PatentDoc2Vec**		
	Acc	F1	MCC	Acc	F1	MCC	Acc	F1	MCC
SVM	62.01 ±3.14	61.90 ±3.55	24.06 ±6.25	59.26 ±2.92	58.04 ±2.72	18.60 ±5.92	61.27 ±3.04	61.35 ±3.35	22.55 ±6.08
LR	62.24 ±2.46	62.17 ±2.77	24.55 ±4.89	60.95 ±1.95	59.80 ±2.33	22.02 ±3.99	61.28 ±2.84	61.27 ±3.15	22.58 ±5.66
RF	64.04 ±3.92	62.11 ±5.09	28.30 ±7.81	62.57 ±3.33	60.37 ±4.35	25.38 ±6.79	67.35 ±3.29	66.14 ±4.55	34.82 ±6.57
XGB	**65.64** ±3.69	**65.65** ±4.52	**31.30** ±7.34	**63.22** ±3.06	**62.66** ±3.37	**26.52** ±6.10	**69.21** ±3.02	**69.30** ±3.49	**38.48** ±6.03

Table 6.5: Best models and their selected parameters

Board of Appeal	Experiment	Model	Model Hyperparameters	Input	Input Hyperparameters	Stopwords	Lemma	Numbers
Patent Refusal	1	XGBoost	n_estimators: 300 learning_rate: 0.05 gamma: 0.0	TF-IDF	use_idf: True norm: L2 ngram_range: (1,4) min_df: 2	False	False	False
Patent Refusal	2	XGBoost	n_estimators: 300 learining_rate: 0.05 gamma: 0.2	Bag of Words	use_idf: False norm: L2 ngram_range: (1,4) min_df: 2	False	False	False
Opposition Division	1	XGBoost	n_estimators: 300 learning_rate: 0.02 gamma: 0.0	Bag of Words	use_idf: False norm: None ngram_range: (1,2) min_df: 5	True	True	True
Opposition Division	2	XGBoost	n_estimators: 300 learning_rate: 0.05 gamma: 0.0	TF-IDF	use_idf: True norm: L2 ngram_range: (1,4) min_df: 10	True	False	False

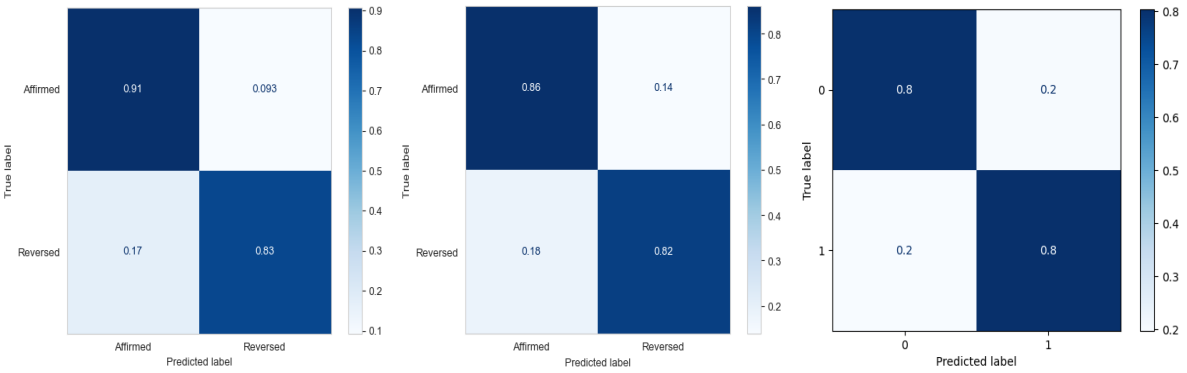
Table 6.6: Test data results for Experiment 1 and 2

	Patent Refusal				**Opposition Division**				**Average**			
	Acc	**F1**	**MCC**	**Baseline**	**Acc**	**F1**	**MCC**	**Baseline**	**Acc**	**F1**	**MCC**	**Baseline**
Experiment 1	85.24	85.09	70.33	**43.64**	78.67	78.66	58.29	54.33	81.96	81.89	64.31	**48.99**
2019-2020	**87.95**	**87.07**	**74.13**	37.08	79.29	79.16	58.57	56.01	**83.62**	**83.12**	**66.35**	46.55
2021-2022	84.49	83.41	66.91	36.1	**80.83**	**80.46**	**60.10**	**57.82**	82.66	81.94	63.51	46.96
2019-2022	86.45	85.48	36.65	36.65	80.08	79.84	59.83	56.94	83.27	82.66	48.24	46.795
Overall									82.88	82.40	60.60	47.32



(a) Patent Refusal (b) Opposition Division

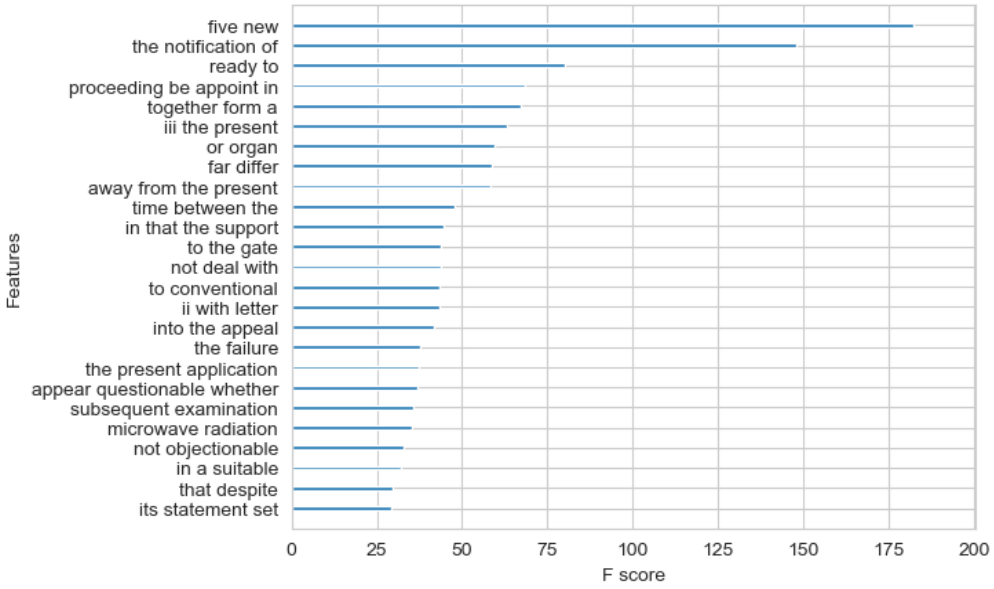
Figure 6.1: Experiment 1: Confusion Matrices Test Results



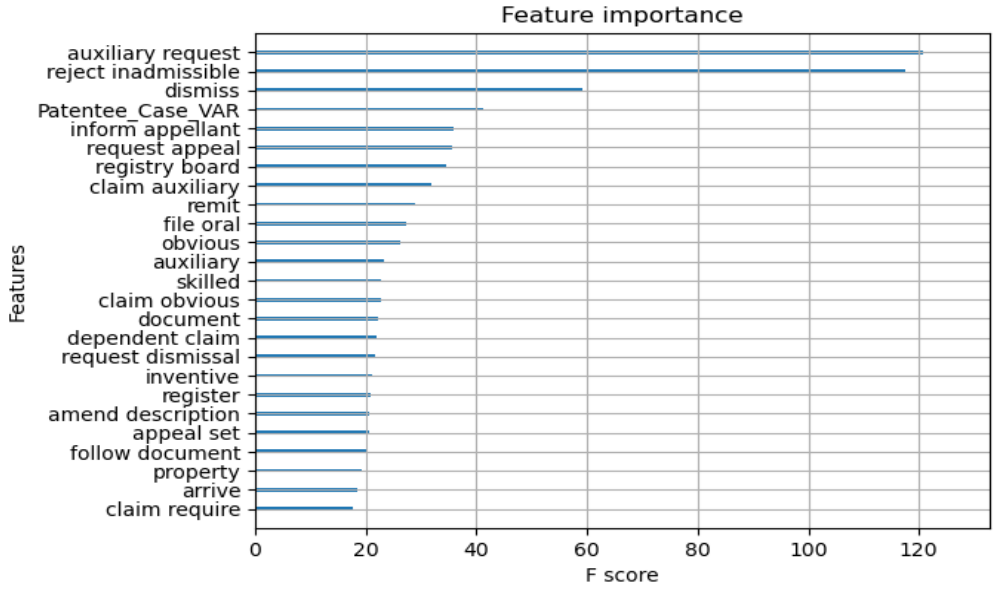
(a) Patent Refusal 2019-2020 (b) Patent Refusal 2021-2022 (c) Patent Refusal 2019-2022

(d) Opposition Division 2019-2020 (e) Opposition Division 2021-2022 (f) Opposition Division 2019-2022

Figure 6.2: Experiment 2: Confusion Matrices Test Results

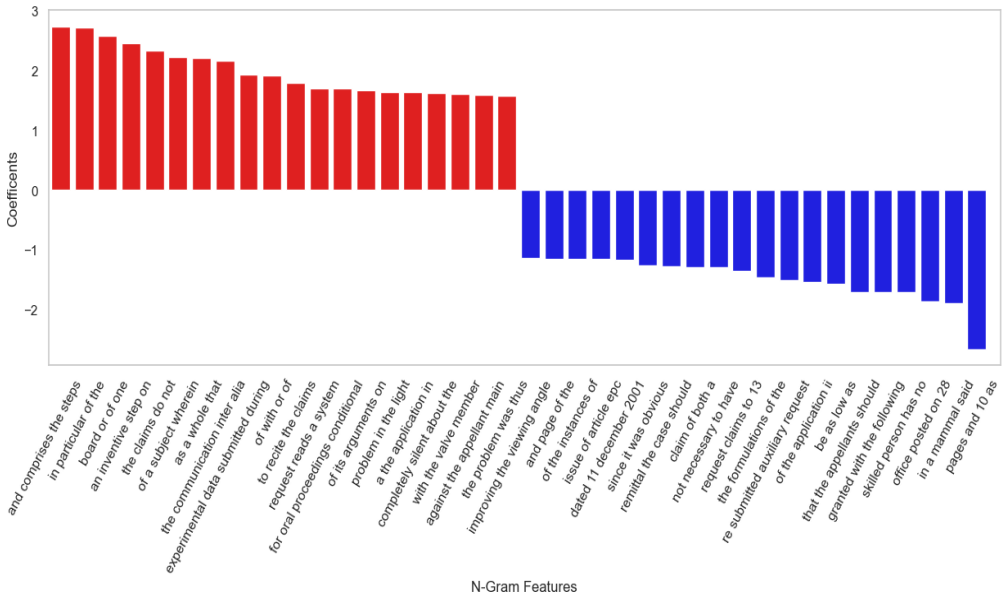


(a) Patent Refusal

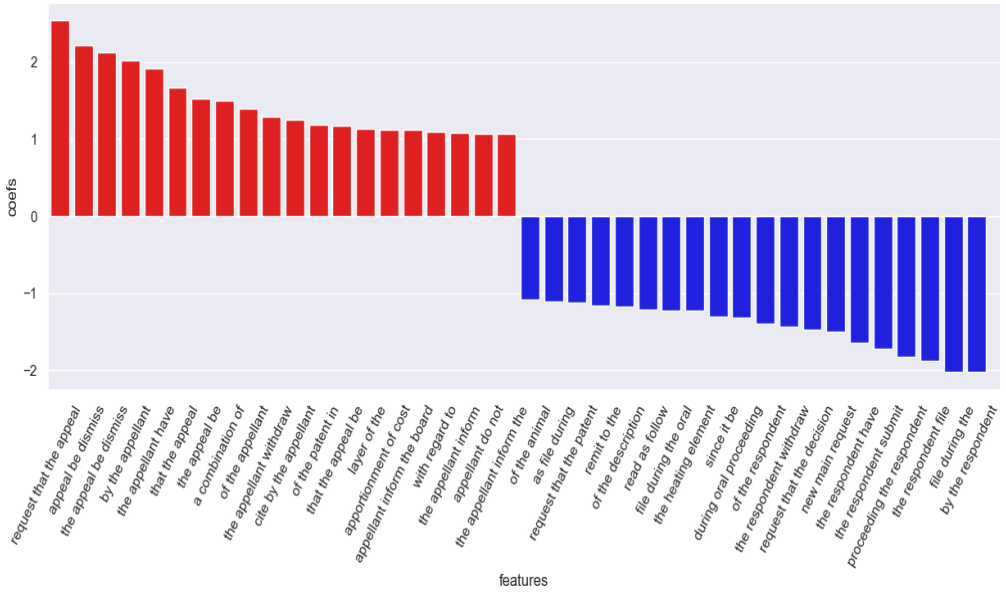


(b) Opposition Division

Figure 6.3: XGBoost Feature Importance



(a) Patent Refusal



(b) Opposition Division

Figure 6.4: SVM Feature Importance

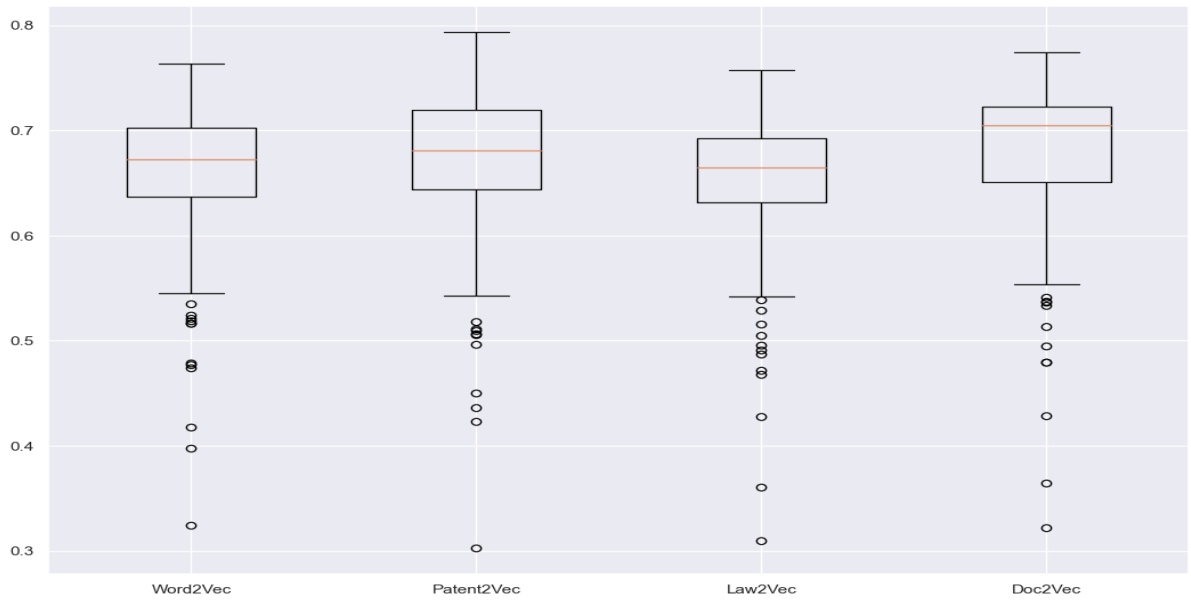


Figure 6.5: Word Embeddings boxplot

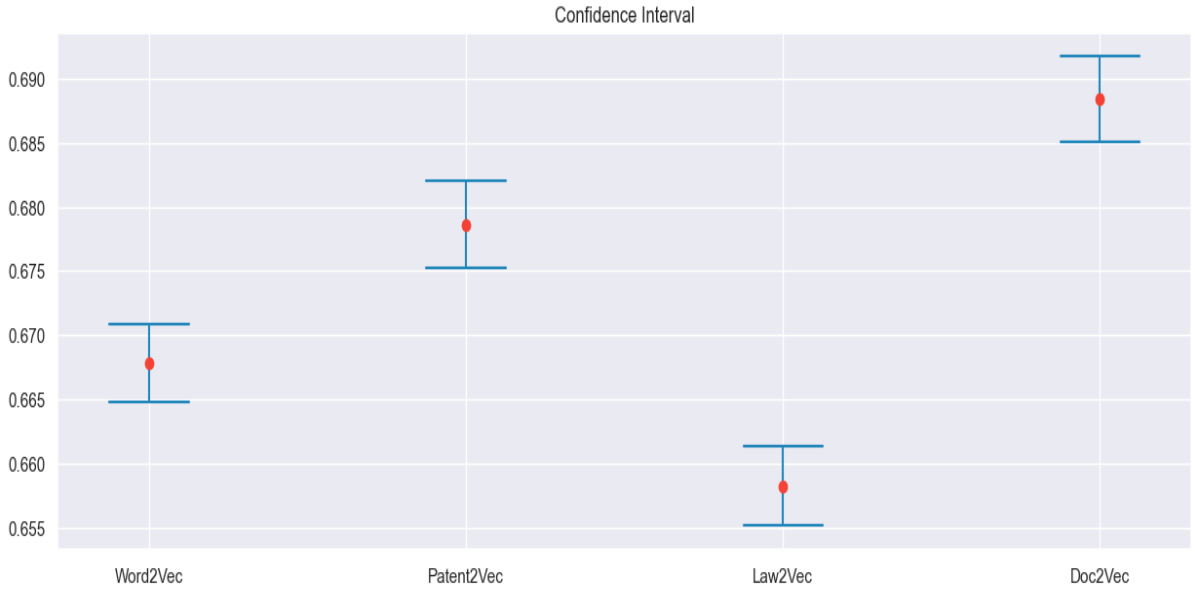


Figure 6.6: Word Embeddings CI

Bibliography

[1] N. Aletras, D. Tsarapatsanis, D. Preoţiuc-Pietro, and V. Lampos, "Predicting judicial decisions of the european court of human rights: A natural language processing perspective," *PeerJ Computer Science*, vol. 2016, 2016.

[2] M. Medvedeva, M. Vols, and M. Wieling, "Using machine learning to predict decisions of the european court of human rights," *Artif. Intell. Law*, vol. 28, no. 2, p. 237–266, jun 2020. [Online]. Available: https://doi.org/10.1007/s10506-019-09255-y

[3] D. M. Katz, M. J. Bommarito, II, and J. Blackman, "A general approach for predicting the behavior of the supreme court of the united states," *PLOS ONE*, vol. 12, no. 4, pp. 1–18, 04 2017. [Online]. Available: https://doi.org/10.1371/journal.pone.0174698

[4] A. R. Kaufman, P. Kraft, and M. Sen, "Improving supreme court forecasting using boosted decision trees," *Political Analysis*, vol. 27, no. 3, p. 381–387, 2019.

[5] H. Zhong, Z. Guo, C. Tu, C. Xiao, Z. Liu, and M. Sun, "Legal judgment prediction via topological learning," pp. 3540–3549. [Online]. Available: https://github.

[6] W. Yang, W. Jia, X. Zhou, and Y. Luo, "Legal judgment prediction via multi-perspective bi-feedback network," 2019.

[7] L. Yao and H. Ni, "Prediction of patent grant and interpreting the key

determinants: an application of interpretable machine learning approach," *Scientometrics*, 2023.

[8] Y.-C. Chi and H.-C. Wang, "Establish a patent risk prediction model for emerging technologies using deep learning and data augmentation," *Adv. Eng. Inform.*, vol. 52, no. C, apr 2022. [Online]. Available: https://doi.org/10.1016/j.aei.2021.101509

[9] S. Hido, S. Suzuki, R. Nishiyama, T. Imamichi, R. Takahashi, T. Nasukawa, T. Idé, Y. Kanehira, R. Yohda, T. Ueno, A. Tajima, and T. Watanabe, "Modeling patent quality: A system for large-scale patentability analysis using text mining," *Information and Media Technologies*, vol. 7, pp. 655–666, 2012.

[10] E. P. Office, "European patent guide: How to get a european patent," 2022. [Online]. Available: https://link.epo.org/web/how_to_get_a_european_patent_2022_en.pdf

[11] L. Bently, B. Sherman, D. Gangjee, and P. Johnson, *Intellectual Property Law*, 5th ed. Oxford University Press, 2018,
Find notes in IP.docx

Find at <u>Kortext | EPUB Reader</u>.

[12] V. Andrea, "Appeal procedure before the european patent office," in *Patent Law and Theory*, T. Takenka, Ed. Edward Elgar Publishing, 2009.

[13] G. Governatori, T. Bench-Capon, B. Verheij, M. Araszkiewicz, E. Francesconi, and M. Grabmair, "Thirty years of artificial intelligence and law: the first decade," *Artificial Intelligence and Law*, vol. 30, pp. 481–519, 12 2022. [Online]. Available: https://link.springer.com/article/10.1007/s10506-022-09329-4

[14] H. Surden, "Artificial intelligence and law: An overview recommended citation," *U. L. Rev*, 2019. [Online]. Available: https://readingroom.law.gsu.edu/gsulrAvailableat:https://readingroom.law.gsu.edu/gsulr/vol35/iss4/8Electroniccopyavailableat:https://ssrn.com/abstract=3411869

[15] A. von der Lieth Gardner, *An Artificial Intelligence Approach to Legal Reasoning*. MIT Press, 1987.

[16] K. D. Ashley, "Modelling legal argument: Reasoning with cases and hypotheticals," Ph.D. dissertation, University of Massachusetts, 1988, order No: GAX88-13198.

[17] V. Aleven, "Using background knowledge in case-based legal reasoning: A computational model and an intelligent learning environment," *Artificial Intelligence*, vol. 150, pp. 183–237, 11 2003.

[18] K. D. Ashley, *Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age*. Cambridge University Press, 2017.

[19] S. Bruninghaus and K. D. Ashley, "Predicting outcomes of case based legal arguments," in *Proceedings of the 9th International Conference on Artificial Intelligence and Law*, ser. ICAIL '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 233–242. [Online]. Available: https://doi.org/10.1145/1047788.1047838

[20] L. Al-Abdulkarim, K. Atkinson, and T. Bench-Capon, "A methodology for designing systems to reason with legal cases using abstract dialectical frameworks," *Artificial Intelligence and Law*, vol. 24, pp. 1–49, 2016.

[21] G. Brewka, H. Strass, S. Ellmauthaler, J. P. Wallner, and S. Woltran, "Abstract dialectical frameworks revisited," in *International Joint Conference on Artificial Intelligence*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:14236262

[22] G. Sartor, M. Araszkiewicz, K. Atkinson, F. Bex, T. van Engers, E. Francesconi, H. Prakken, G. Sileno, F. Schilder, A. Wyner, and T. Bench-Capon, "Thirty years of artificial intelligence and law: the second decade," *Artificial Intelligence and Law*, vol. 30, pp. 521–557, 12 2022. [Online]. Available: https://link.springer.com/article/10.1007/s10506-022-09326-7

[23] S. Villata, M. Araszkiewicz, K. Ashley, T. Bench-Capon, L. K. Branting, J. G. Conrad, and A. Wyner, "Thirty years of artificial intelligence and law: the third decade," *Artificial Intelligence and Law*, vol. 30, pp. 561–591, 12 2022. [Online]. Available: https://link.springer.com/article/10.1007/s10506-022-09327-6

[24] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

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

[28] C. O'Sullivan and J. Beel, "Predicting the outcome of judicial decisions made by the european court of human rights," 2019.

[29] A. Kaur and B. Bozic, "Convolutional neural network-based automatic prediction of judgments of the european court of human rights," in *Proceedings for the 27th AIAI Irish Conference on Artificial Intelligence and Cognitive Science, Galway, Ireland, December 5-6, 2019*, ser. CEUR Workshop Proceedings, E. Curry, M. T. Keane, A. Ojo, and D. Salwala, Eds., vol. 2563. CEUR-WS.org, 2019, pp. 458–469. [Online]. Available: https://ceur-ws.org/Vol-2563/aics_42.pdf

[30] I. Chalkidis, I. Androutsopoulos, and N. Aletras, "Neural legal judgment prediction in english," *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 4317–4323, 2020.

[31] M. Medvedeva, X. Xu, M. Wieling, and M. Vols, "Juri says: An automatic judgement prediction system for the european court of human rights," in *Legal Knowledge and Information Systems*, ser. Frontiers in Artificial Intelligence and Applications, S. Villata, J. Harašta, and P. Křemen, Eds. IOS Press, Dec. 2020,

pp. 277–280, publisher Copyright: © 2020 The Authors, Faculty of Law, Masaryk University and IOS Press.

[32] O.-M. Şulea, M. Zampieri, M. Vela, and J. van Genabith, "Predicting the law area and decisions of French Supreme Court cases," in *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. Varna, Bulgaria: INCOMA Ltd., Sep. 2017, pp. 716–722. [Online]. Available: https://doi.org/10.26615/978-954-452-049-6_092

[33] A. J. Trippe, "Patinformatics: Tasks to tools," *World Patent Information*, vol. 25, no. 3, pp. 211–221, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0172219003000796

[34] J. M. de Rezende, I. M. da Costa Rodrigues, L. C. Resendo, and K. S. Komati, "Combining natural language processing techniques and algorithms lsa, word2vec and wmd for technological forecasting and similarity analysis in patent documents," *Technology Analysis & Strategic Management*, vol. 0, no. 0, pp. 1–22, 2022.

[35] L. Aristodemou and F. Tietze, "The state-of-the-art on intellectual property analytics (ipa): A literature review on artificial intelligence, machine learning and deep learning methods for analysing intellectual property (ip) data," *World Patent Information*, vol. 55, pp. 37–51, 12 2018, find notes in IP.docx.

[36] K. Makovi, Z. Somogyvari, K. Strandburg, J. Tobochnik, P. Volf, L. Zalanyi, D'Agostino, and M. E. Greenberg, "Prediction of emerging technologies based on analysis of the us patent citation network," *Scientometrics*, vol. 95, pp. 225–242, 2013. [Online]. Available: http://www.nsf.gov/statistics/seind10/c4/c4s5.htm

[37] J. Yun and Y. Geum, "Automated classification of patents: A topic modeling approach," *Computers & Industrial Engineering*, vol. 147, p. 106636, 2020.

[Online]. Available:
https://www.sciencedirect.com/science/article/pii/S0360835220303703

[38] M. F. Grawe, C. A. Martins, and A. G. Bonfante, "Automated patent classification using word embedding," *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*, vol. 2017-December, pp. 408–411, 2017.

[39] C. V. Chien, "Predicting patent litigation," *Law & Society: Private Law eJournal*, p. 48 pages, 2011. [Online]. Available: http://tind.wipo.int/record/42089

[40] S. Juranek and H. Otneim, "Using machine learning to predict patent lawsuits," *SSRN Electronic Journal*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238832316

[41] W. McKinney, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, S. van der Walt and J. Millman, Eds., 2010, pp. 51 – 56.

[42] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[43] I. Chalkidis and D. Kampas, "Deep learning in law: early adaptation and legal word embeddings trained on large corpora," vol. 27, pp. 171–198, 2019. [Online]. Available: https://doi.org/10.1007/s10506-018-9238-9

[44] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:18593743

[45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine

learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[46] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785

[47] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958.

[48] D. A. Pisner and D. M. Schnyer, "Support vector machine," *Machine Learning: Methods and Applications to Brain Disorders*, pp. 101–121, 1 2019.

[49] L. Breiman, "Statistical modeling: The two cultures," pp. 199–231, 2001.

[50] J. H. Friedman, "Greedy function approximation: A gradient boosting machine." *Annals of Statistics*, vol. 29, pp. 1189–1232, 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:39450643

[51] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2016. [Online]. Available: https://doi.org/10.1145%2F2939672.2939785

[52] C. Steging, S. Renooij, and B. Verheij, "Taking the law more seriously by investigating design choices in machine learning prediction research," 2023. [Online]. Available: http://ceur-ws.org

[53] D. Chicco, M. J. Warrens, and G. Jurman, "The matthews correlation coefficient (mcc) is more informative than cohen's kappa and brier score in binary classification assessment," *IEEE Access*, vol. 9, pp. 78 368–78 381, 2021.

[54] R. Rehurek and P. Sojka, "Gensim–python framework for vector space modelling," *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, vol. 3, no. 2, 2011.

[55] L. V. D. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[56] H. Zhong, C. Xiao, C. Tu, T. Zhang, Z. Liu, and M. Sun, "How does nlp benefit legal system: A summary of legal artificial intelligence," 2020. [Online]. Available: https://github.com/thunlp/LegalPapers

[57] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf