DEPARTMENT OF COMPUTER SCIENCE

# Natural Language Processing in Intersecting Fields of Law

Oliver Ryan-George

---

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Bachelor of Science in the Faculty of Engineering.

---

Wednesday 15$^{th}$ May, 2019

# Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of BSc in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Oliver Ryan-George, Wednesday 15$^{th}$ May, 2019

# Acknowledgements

I would like to thank Dr. Miranda Mowbray for her guidance on both the computer science and law aspects of the project. I would also like to thank Dr. Megan Rae Blakely for sharing her knowledge of the law domain and for her feedback on the project.

# Executive Summary

This project is a contribution to Dr. Megan Rae Blakely's research into the growing relationship between intellectual property law and human rights law; in particular, the extent to which intellectual property laws involve human rights considerations and their balance between consideration of creators and users. Currently, the relationship between intellectual property and human rights is analysed on a case-by-case basis due to a lack of systematic means of analysis. The project shows that a systematic method in natural language processing can be appropriate in this domain and therefore allow a more comprehensive argument to be made about the relationship.

The project shows the technology is appropriate for the domain through the use of three computer science skills: natural language processing, data visualisation and usability consideration. A support vector machine is used to analyse a set of journal articles to establish a model of what the language of human rights and intellectual property consists of. It is then used to test a further set of journal articles to see how the set compares to the model over time. A rules-based method is used to determine whether articles' tone suggest the legal context is benefiting the user or the creator. This data is then visualised in order to make the results of the natural language processing pallatable. This is done via an iterative process in order to find the most appropriate visualisation for the domain. A user interface is built in order for the intended user to be able to make the most of this functionality by adding new articles while not needing to know any of the intricacies of natural language processing.

The natural language processing methods scored well in cross-validation using balanced accuracy scores and the final feedback from Dr. Blakely on the visualisation and user interface was positive. This was acknowledged at the British and Irish Law Education and Technology conference.

The tool will go on to be extended based on my recommended future work. The project will lead to further understanding of how past events have impacted the intellectual property field with respect to the human rights field and vice versa. Further, the project's success may encourage further use of technology in the domain.

# Contents

# Chapter 1

# Introduction

## 1.1 Chapter Overview

To open the thesis, this chapter gives a brief background of why the project is being pursued in the law domain. It continues by explaining the project's aims with regards to the given background and then finishes with how the aims intended to add value to the law domain once they had been met.

## 1.2 Project Context

Intellectual Property law is a term typically used to describe the areas of law which establish property protection over intangibles such as ideas, signs and information. This protection is in order to make the advancement of ideas profitable which incentivises this act[1].

There is, therefore, a balance to be struck between limited exclusive rights and benefits to the public. While limiting exclusive rights may facilitate progress, benefiting the public, the overprotection of the exclusive property may restrict their access[2]. One example of this is the expansion of copyright terms such as the controversial Copyright Term Extension Act of 1998 which was heavily lobbied for by Disney just years before Mickey Mouse's copyright ran out[3]. The trend in the extension of copyright terms, illustrated by Figure 1.1, illustrates the appearance that the balance is tipping towards exclusivity rights.

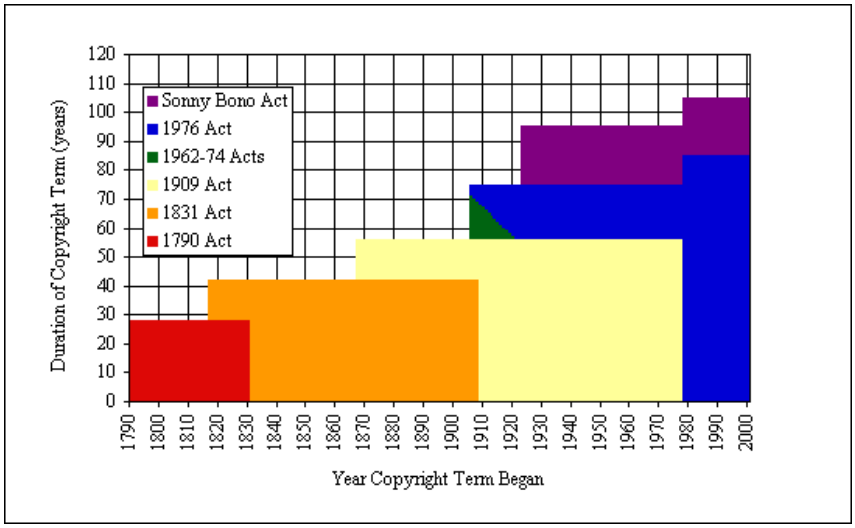



Figure 1.1: Expansion of U.S. copyright term lengths[4].

This implementation of intellectual property law brings in the question of human rights of access to culture, education and other social and economic rights, but traditionally this has not been included in the discussion of intellectual property law. However, recent scholarship and legislation has progressively begun to incorporate each other's linguistics[5].

Dr. Megan Rae Blakely of Lancaster University is looking to persuade other academics that elements of human rights, especially cultural heritage, should be discussed in the intellectual property field and

vice versa. She has determined that showing a strong and growing intersection between the fields will help to achieve these aims. For this reason, she is looking for assistance in analysing journals and legal instruments for overlap in their languages.

As well as the human rights and intellectual property, Dr. Blakely speculated that it may be interesting to see how the language in these fields benefit the user and the creator of intellectual property[6] particularly because of the development of technology in the last few decades which has created ambiguity over who the user is. This will also help in the investigation over whether copyright is functioning as an incentive for creators to disseminate their works to the public.

In this analysis, I will map the dynamics of this change in legal and social perspective, to make evident moments of significant shifts in language. Previously, analysis of the intersection between human rights and intellectual property law, such as Helfer's[7], has been limited to more manual case-by-case methods. Supplying a more systematic method using natural language processing that can cope with large amounts of data could give concrete evidence of the relationship between human rights and intellectual property.

## 1.3 Ground Truths

Dr. Blakely supplied a set of ground truths that she wished to analyse the language of. This consisted of the following:

- Four journals, shown in Table A.1: two that are on the topic of human rights and two that are on the topic of intellectual property;
- Four international treaties, shown in Table A.2: two that are on the topic of human rights and two that are on the topic of intellectual property;
- Four lists of words, shown in Table A.3: one that Dr. Blakely expects to see in documents about human rights, one that Dr. Blakely expects to see in documents about intellectual property, one that Dr. Blakely thinks indicates that a segment indicates benefits to the user and one that Dr. Blakely thinks indicates that a segment indicates benefits to the creator.

## 1.4 Requirements

In the early stages of the project, I discussed Dr. Blakely's aims for the project. Her key objective was to prove that natural language processing could be used by law academics via a tool to gather evidence on the differing languages in law fields. Bearing this in mind, I established a set of requirements which Dr. Blakely then approved. These requirements can be split into three categories: natural language processing, visualisation, and usability.

Each of the following subsections discusses the requirements of a different category. Each subsection starts by specifying the deliverable elements for the category, followed by high-level criteria of how they will be evaluated. All goals needed to be completed by 16th April 2019, when Dr. Blakely presents to the British and Irish Law Education and Technology Association (BILETA) Conference.

### 1.4.1 Natural Language Processing

The natural language processing in the project will take the form of two models. Each model will take the same corpus of training PDF documents as input but will classify different characteristics of the corpus.

- The first model will use topic classification to identify whether a document is related more to human rights or intellectual property;
- The second model will use sentiment analysis to identify whether the document indicates that the current legal climate more strongly benefits the user of intellectual property of the creator of it.

It is notable that Dr. Blakely expressed that the former is the more important model in terms of the project's success as there is more scholarly work done on this relationship. The latter models' relationship is less widely covered and was requested by Dr. Blakely as a personal preference.

These models will be successful if they meet the following criteria:

- The model accurately represents a large proportion of the documents in the corpus with regards to its classification characteristics;

- The model deduces any trends in the corpus with regards to its classification characteristics if those trends exist;
- The model must only consider the language of the content of the article.

### 1.4.2 Visualisation

The previously mentioned models will then have to be visualised. Dr. Blakely showed interest in exploring Eboch's tone graph[8], a modification of which is shown in Figure 1.2, to start with. This would involve an x-axis representing time, a y-axis representing human rights-intellectual property and a z-axis representing user-creator.

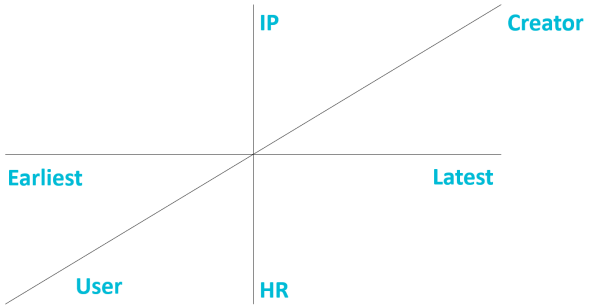



Figure 1.2: An adaptation of Eboch's tone graph for this domain.

The visualisation will be successful if it meets the following criteria:

- The visualisation and its axes' meanings are self-explanatory;
- It is clear where a point lies on the visualisation's axes;
- The visualisation has all the possible information that Dr. Blakely requires, on it;
- The visualisation clearly indicates any trends deduced by the models;
- The visualisation is aesthetically pleasing.

### 1.4.3 Usability

There will be two key types of stakeholders in this project. Both their circumstances will have to be considered in order to deliver a suitable final product.

The first type of stakeholder are the users of the project's end product. This will primarily be Dr. Blakely but is likely to extend to other law academics. It is important to note that these will likely be people with standard information technology skills and no computer science skills. Therefore, the end product will need to be in the form of a tool that is accessible to people of this calibre.

The tool must have the following features:

- Allow for input of new documents;
- Display visualisations based on models with given documents as input;
- Display information about the success of the model.

The tool will be successful if it meets the following criteria:

- The tool does not require any prior Computer Science knowledge to setup or use;
- The tool has a graphical user interface which is intuitive;
- The tool allows for maximum time to be spent on analysis of results.

The second type of stakeholder is the future developers furthering Dr. Blakely's research. These developers will be experts in natural language processing and computer science and so will be able to both use and change the tool to their liking. However, changes to the tool will only be easily changeable if the codebase for it is well written.

The codebase will be successful if it meets the following criteria:

- It is easy to understand the purpose of each section of code;
- It is easy to expand upon and adapt the overall codebase.

## 1.5 Project Challenges

The key challenges posed by achieving the requirements set out in Section 1.4 are presented in the following list:

- Managing priorities in a project with significant breadth so all aims have sufficient attention paid to them;
- Working in the law domain which I have had no education in meant a lot of research needed to be done to understand the purpose of the project to motivate my decisions in the project;
- Working in the law domain which has minimal similar established work meant that there was little precedent for my work and that more creativity was needed to solve the problems;
- Understanding the preferences of a client and working around both our busy schedules;
- Explaining computer science concepts in a paletable way to those with no computer science experience.

## 1.6 Added Value

This project is original in its application of natural language processing methods to find the relationship between intellectual property and human rights language. The individual computing methods used were not original, but the added value appears through the iterative process which found the most appropriate visualisation and tool for the domain. The outcome of the project will help add value to the domain by illustrating how historical changes in technology have impacted the tone of intellectual property law. Blakely points out that this, in turn, will allow legal professionals to consider how future technological changes will impact their work and adapt accordingly.

# Chapter 2

# Literature Review

## 2.1 Chapter Overview

This chapter details the background research that was carried out in order to make informed decisions during the project.

## 2.2 Natural Language Processing

### 2.2.1 Evaluation of Binary Classification Algorithms

A binary classification model is an algorithm that sorts a set into two categories. One set is often referred to as 'positive' and the other 'negative'. It is important to validate a model in order to see whether its results can be trusted. Upon validating, if an item in the set has been correctly classified, it is considered a 'true' classification, otherwise, it is considered a 'false' classification.

The holdout method of validation involves training the model on a large proportion of the data, known as training data, and then classifying the rest of the data, known as test data, and giving the model a score based on how the test data was classified based on their actual class[9]. This is limited as the test data set may be overfit to the model, making the model perform better for that test data than it would for new incoming data. Two common methods of solving this problem are cross-validation and bootstrapping.

Cross-validation is where the dataset is split into $k$ segments. Each segment is treated a training data $k-1$ times and treated as test data once. This will give $k$ different scores for the model. Meanwhile, bootstrapping is where $n$ items in the dataset are picked uniformly at random. Each item is put back into the dataset once picked. This can be done several times to produce different scores for the model. In both cases, if the scores have a high average and low variance, the model is successful and not overfit to the data[9].

Kohavi compares these methods and recommends the use of cross-validation because bootstrap can be prone to large biases[9]. Kohavi found that the optimum number of folds depends on the stability of the dataset. He also notes that cross-validation's evaluation is most accurate when each fold is stratified to have a similar number of each class in it.

The simplest score that can be used for a model is accuracy. Accuracy is the ratio of correctly classified test documents to the total number of test documents. This is insufficient for imbalanced datasets[10].

A common solution to this in topic classification is to use precision, which is the ratio of true positives to the total number of positive classifications shown in Equation 2.1, and recall, which is the ratio of true positives to the total number of classifications made in total, shown in Equation 2.2. Precision and recall can be combined in two ways. The F1 score finds the harmonious mean between the two as shown in Equation 2.4. The precision/recall breakeven point is obtained by modifying the classifying threshold until the precision and recall values are equal[11]. These methods are most appropriate for information retrieval when it is important to evaluate true positives and true negatives are in excess[12].

$$precision = \frac{true\text{-}positives}{true\text{-}positives + false\text{-}positives} \tag{2.1}$$

$$recall = \frac{true\text{-}positives}{true\text{-}positives + false\text{-}negatives} \tag{2.2}$$

| type | score |
|---|---|
| accuracy | 0.95 |
| $f_1$ score | 0 |
| BEP | 0 |
| BACC | 0.48 |

Table 2.1: Scores with a ground truth of 5 positives and 95 negatives and a prediction of 0 positives and 100 negatives.

| type | score |
|---|---|
| accuracy | 0.95 |
| $f_1$ score | 0.97 |
| BEP | 0.95 |
| BACC | 0.48 |

Table 2.2: Scores with a ground truth of 95 positives and 5 negatives and a prediction of 100 positives and 0 negatives.

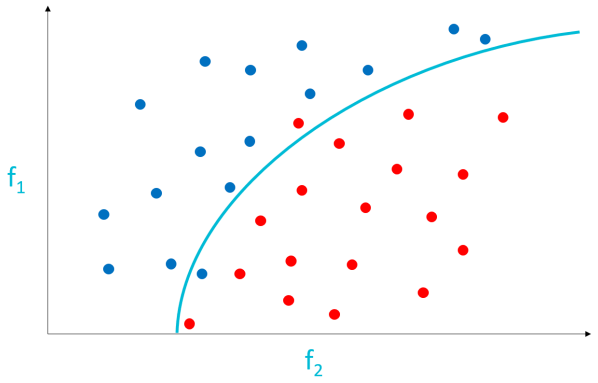



Figure 2.1: A simplified example of how Support Vector Machines work.

$$true\text{-}negative\text{-}rate = \frac{true\text{-}negatives}{true\text{-}negatives + false\text{-}positives} \tag{2.3}$$

Velez suggests balanced accuracy, shown in Equation 2.6, as the solution. This is the average of the true positive rate, which is another name for recall shown in Equation 2.2, and the true negative rate, which is the ratio of true negatives to the total number of classifications made shown in Equation 2.3. This method is more appropriate for cases where it is important to evaluate both the true positives and true negatives.

$$f_1 = 2 \times \frac{precision \cdot recall}{precision + recall} \tag{2.4}$$

$$break\text{-}even\text{-}point = precision = recall \tag{2.5}$$

$$balanced\text{-}accuracy = \frac{recall + true\text{-}negative\text{-}rate}{2} \tag{2.6}$$

Tables 2.1 and 2.2 show how the different scoring systems rate the performance of imbalanced datasets and algorithms. The algorithms both classify a given test item into a predetermined class, regardless of the item's contents. Balanced accuracy is the only scoring system to rank both algorithms poorly.

### 2.2.2 Topic Classification

Document topic classification is the automated assignment of natural language texts to predefined categories based on their content[11].

Support vector machines are a common feature-based approach to classification. They work by considering features, which are to be discussed in Subsection 2.2.4, and plotting each document in the feature space by the number of times each feature occurs in it. It then plots a hyperplane that best splits each document class, as exemplified by Figure 2.1. This led to an 0.86 precision/recall breakeven point when Joachims classified the Reuters 'acq' dataset[13].

In contrast, Naive Bayes is a probabilistic approach based on Bayes' rule. Bayes' rule, shown in Equation 2.7, is a formula for finding the probability of class $C_k$, given a set of features $x$. There are many variations of this type of classification where the likelihood, representing the probability of a document given a class, diers. The Bernoulli multivariate model only considers whether a word occurs in a document while the multinomial model considers how many times a word appears in each document. Predictably, McCallum and Nigam found that the multinomial model performs consistently stronger over

any significant vocabulary size. This led to a 0.89 precision/recall breakeven point when classifying the Reuters 'acq' dataset, slightly better than Support Vector Machines 0.86.

$$p(C_k|\mathbf{x}) = \frac{p(C_k)p(\mathbf{x}|C_k)}{p(\mathbf{x})} \tag{2.7}$$

Wang and Manning improved the performance of both naive Bayes and support vector machines, for datasets with average wordcounts over 100, by adding elements of naive Bayes to support vector machines[14].

All these methods assume that each document can only be a member of one class. This is often not the case. Godbole and Sarawagi use the example that if a document is classed as being about wheat, it is also likely to be about grain[15]. This relationship between topics can be captured in a topic-topic distribution and used to enhance results. Godbole and Sarawagi implement a few variations of this, all based around the standard support vector machines with one added feature per class representing how similar the given class is to all other classes. It leads to a slight improvement of F1-score across two datasets.

Established support vector machine and naive Bayes algorithms are available as part of Python's 'sklearn' library.

### 2.2.3 Sentiment Analysis

The same classification techniques used in topic classification in Subsection 2.2.2 can be used for sentiment analysis as demonstrated by Moraes et al who experimented with support vector machines, naive Bayes and artificial neural networks. They classified reviews as positive or negative with a 0.87 accuracy score using neural networks on a movie reviews dataset[16]. Using these machine learning techniques, however, needs significant data to train on which is not always available.

An alternative method for sentiment analysis is a lexicon based approach. As a preliminary investigation, Pang et al obtained a list of positive keywords and a list of negative keywords for a different set based on analysis of the term frequencies of a set of training data[17]. They then classified the test data based on whether they had more positive or negative keywords. This achieved an accuracy of 0.64, performing significantly worse than Moares et al at the same task.

Mandal and Gupta propose a more advanced lexicon based approach[18]. This involves a detecting comparative and superlative words and assigning a higher weighting when these precede a keyword. Most notably, it involves negating rating should negation words such as 'not' appear before a keyword. This increased Mandal and Gupta's accuracy from 0.87 to 0.97 on their manually created online review dataset.

### 2.2.4 Language Features

In natural language processing, each document is represented as a set of features where each feature is given a value. The simplest set of features a document can have is the set of unique words that appear in the document. This is known as the bag of words model. It has the benefit of having dense feature spaces, meaning that there is a low number of features which reoccur many times in the document. Dense feature spaces are more computationally efficient than sparse feature spaces. However, bag of words is limited in that these features are less capable of conveying the document's meaning. For example, Wallach points out that the phrase "the department chair couches offer" has a different topic to "the chair department offers couches", yet a bag of words model will represent them as the same[19]. This sort of ambiguity could give inaccurate results so it is worth using a feature a bit more resource heavy that provides more accuracy.

Another common type of feature used is an n-gram model. This is where $n$ words are taken as the feature, giving each feature context. For example, a 5-gram would have the capability to distinguish between the above examples. However, the higher n is the less computationally efficient the processing will be and Tan et al found that the pay-off in computational efficiency was not worth it for n=3 but was for n=2[20]. There are methods to improve computational efficiency, such as filtering out all features that do not meet some threshold. Tan et al's found that filtering out all features that appeared in less than three documents or less than 0.5% of documents helped sufficiently. Implementing bigrams and combining them with the documents' unigrams, resulted in an average F1-score improvement of 0.03 on the Reuters datasets[20].

Alongside changing the type of features, the way that features are counted can be changed. The simplest form of this is counting the features as binary, representing a feature as 1 if it occurs in a

document and 0 if it does not. Another simple count is to represent a feature by the number of times it occurs in a document. However, method gives more influence to topics discussed in longer documents. Term frequency is shown in Equation 2.8 where $f_{t,d}$ is the number of times term $t$ occurs in document $d$. This solves the problem of weighting towards large documents as taking the ratio of term appearance removes the relevance of the document's size. Inverse document frequency is shown in Equation 2.9 where $N$ is the total number of documents and $n_t$ is the number of documents where term $t$ occurs. This gives a higher weight to words that appear in fewer documents. Term frequency-inverse document frequency (tf-idf) is a combination of its namesake methods, shown in Equation 2.10. Lan et al investigate these methods and variants of them and find that term frequency performs better than tf-idf frequency for the Reuters dataset despite tf-idf being the standard[21].

$$tf = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{2.8}$$

$$idf = \log \frac{N}{n_j} \tag{2.9}$$

$$tf\text{-}idf = tf * idf \tag{2.10}$$

Liu and Yang proposed a simple improvement to tf-idf named term frequency-inverse document frequency-class frequency (tf-idf-cf) which is shown in Equation 2.11 where $n_{ct}$ is the number of documents of class $c$ that term $t$ occurs in and $N_c$ is the total number of documents in class $c$[22]. This adds weight to a term if it commonly occurs in the document's class. This gave a 0.08 accuracy improvement over tf-idf on the Reuters dataset.

$$tf\text{-}idf\text{-}cf = tfidf * \frac{n_{ct}}{N_c} \tag{2.11}$$

### 2.2.5 Detecting Trends

Regression analysis is the construction of mathematical models which explain relationships between variables. Ordinary least squares is the simplest regression technique intuitively. A line $f(x, \beta)$ is constructed from the parameters $\beta$ that minimise the total squared distance, $S(\beta)$, of the line to each observation$y_i$[23].

$$f(x, \beta) = \sum \beta_j \phi_j(x) \tag{2.12}$$

$$r_i(\beta) = y_i - f(x_i, \beta_\mathbf{i}) \tag{2.13}$$

$$S(\beta) = \sum r_i^2(\beta) \tag{2.14}$$

The least squares method can produce lines that are not straight using polynomial linear regression where $\beta$ remains linear but the $x$ inputs are polynomials. Alternatively, a nonlinear method can be used which involves the $\beta$ parameters taking the form of functions[24].

Ordinary least squares works on the assumption that all distances are errors that are independent and identically distributed normally. However, a single grossly out-lying observation can spoil the estimate[25]. The solution to this is to modify least squares in order to make the estimate less sensitive to extreme values of residuals. This is called robust regression.

Algorithms that perform all mentioned variants of ordinary least squares are available as part of Python's 'statsmodels' library.

### 2.2.6 Statistical Significance

If a linear regression line has a gradient not equal to zero, there is a relationship between the variables. It is very likely that this will be the case but it is not necessarily the case that the gradient will be similar every time. Statistical significance is the idea that a claimed trend has not happened by chance and will consistently occur with other data. Fisher arbitrarily dictated that a trend is statistically significant if its p-value is less than 0.05[26].

A p-value is the probability that a statistical summary of the data would be equal to or more extreme than its observed value[27]. It is worked out based on the gradient's t-value. A t-value is the ratio of the difference between the estimated value of a parameter, from its hypothesised value and the standard error. For finding the statistical significance of a gradient, the estimated value of the parameter is the gradient worked out in linear regression and the hypothesised value is zero.

However, the importance of p-values has come into question since many scientific claims have turned out to be false despite being backed by p-values. Benjamin et al recommend reducing the statistical significance threshold to 0.005 to ensure it is very likely to be statistically significant before making any bold claims[28]. The American Statistical Association, meanwhile, just warn that statistical significance should not be taken as gospel and any decisions came to based on p-values should be assisted by thorough human analysis[27]. As well as this, Boos and Stefanski question the reproducibility of p-values and encourage cross-validation-like methods to ensure that a claim is statistically significant[29].

Algorithms that calculate the p-value of a regression line are available as part of Python's 'statsmodels' library.

### 2.2.7 Pre-processing

Pre-processing text before classification is where text is formatted and changed before classification. It primarily reduces the feature set which improves the efficiency of classification but it can also improve classification accuracy. Three of the main pre-processing methods are stemming, removing stopwords, lowercasing.

It is easiest to process plaintext but most documents are available in PDF or HTML format. There are algorithms which go through the source of these formats and extract the text out, removing all of the formatting which is unnecessary for the natural language processing. There is no library considered as the standard that performs these tasks but 'pdfminer' and 'PyPDF2' are commonly recommended to extract text from PDF files and 'html2text' and 'BeautifulSoup' are commonly recommended to extract text from HTML files.

Stemming is where the suffix of a word is removed in order to collate all variants of the same word. This would change the words 'property' and 'properties' to 'properti'. Lemmatising is a similar method performing same the task which removes the suffix of a word but then replaces it with a standard suffix so it is still a valid word[30]. This would change both 'property' and 'properties' to 'property'. Toman et al found that both stemming and lemmatising either made no significant improvement on classification or gave significantly worse classification results than using neither method[30]. Meanwhile, Uysal and Gunal found that whether there was performance improvement with stemming was dependent on the domain but the proportion of words that were stemmed were up to 0.45[31], therefore, increasing the efficiency of the classification. Stemming and lemmatisation can be implemented using Python's 'nltk' library.

Stopwords are words that are so common in all classes that they make no difference to classification and therefore, there may be no point in including them. This includes words such as 'the', 'and' and 'i'. Interestingly, Toman et al found that stopwords should be taken out of text as that improves classification but Uysal and Gunal found that stopwords should be left in as they can inadvertently indicate class. This reaffirms Uysal and Gunal's hypothesis that there is no set of pre-processing methods that can be applied to all domains and they should instead be experimented with.

Lowercasing is replacing all capital letters with lowercase letters so a normal word is treated as the same word even when it is at the start of the sentence. Uysal and Gunal found this made a significant improvement to classification in all domains.

## 2.3 Visualisation

### 2.3.1 Evaluation

Likert scales are a common format for rating visualisations[32]. This involves ranking the visualisation on its quality from low to high from at least five options. Since this is an ordinal measurement, continuous analysis of results such as mean and variance should not be used. Median and range are more suitable.

An alternative to a Likert scale is a continuous scale. This also involves ranking the visualisation on its quality from low to high but instead of selecting a level of quality from the given options, the user can drag a pointer between low to high to put it exactly where they want.

### 2.3.2 Displaying Trends

Gestalt set out principles of visual perception. These are widely used in information visualisation because they aid understanding of how people perceive patterns[33]. The following list is a subset of Gestalt principles that are most useful for emphasising trends and anomalies:

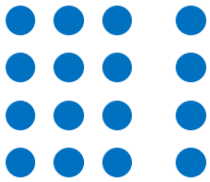


Figure 2.2: An illustration of proximity.

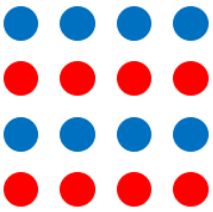


Figure 2.3: An illustration of similarity.

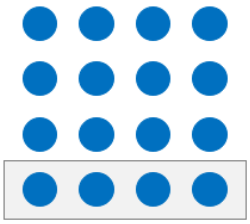


Figure 2.4: An illustration of enclosure.

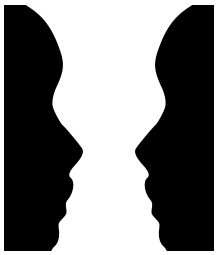


Figure 2.5: An illustration of figure & ground.

- Proximity, illustrated in Figure 2.2, is the idea that groups of objects that are close together are quickly identified as a group;
- Similarity, illustrated in Figure 2.3, is the idea that objects that have similar attributes, such as colour or shape, are quickly identified as a group;
- Enclosure, illustrated in Figure 2.4, is the idea that objects that are encapsulated by another object are quickly identified as a group;
- Figure & ground, illustrated in Figure 2.5, is the idea that a perceived foreground and background are quickly identified as two separate groups.

Healey extends upon this by investigating preattentive features of objects[34]. Preattentive features are features which are identified immediately as an image is first scanned. These included length, width, volume, orientation, number, density, curvature, closure, terminators, colour, luminance and more. The idea is that a human can identify any inconsistencies in these features without having to think, such as if there is one red dot among many blue dots. It is important to note that some features are asymmetric, for example, a curved line among vertical lines can be identified preattentively but a vertical line among curved lines cannot be. Healey also investigated how visual features compete for attention. The purpose of this is so the most important information can be encoded to be processed earliest. He identified the order of processing of visual features as follows:

1. Determine the 3D layout of a scene;
2. Determine surface structures and volumes;
3. Establish object movement;
4. Interpret luminance gradients across surfaces;
5. Use colour to fine tune these interpretations.

Kosslyn lays out eight principles of graph design which emphasise the importance of the graph's context[35]. The following is a list of the principles that do not overlap with Gestalt principles:

- Relevance - communication is most effective when the appropriate amount of information is displayed. All decisions should be made based on whether the outcome more effectively conveys the desired message;
- Appropriate knowledge - communication requires prior knowledge of relevant concepts, jargon and symbols. The amount of explanation behind the graph should be tailored to the prior knowledge of the audience.
- Salience - attention is drawn to large perceptible differences. It is important to highlight the significant information in the graph with striking changes in the visualisation;
- Compatibility - a message is easiest to understand if its form is compatible with its meaning. This can be fundamental such as using line graphs for a continuous x-axis or it can be used to illustrate a message as a persuasive device, such as using red in a graph about murder;

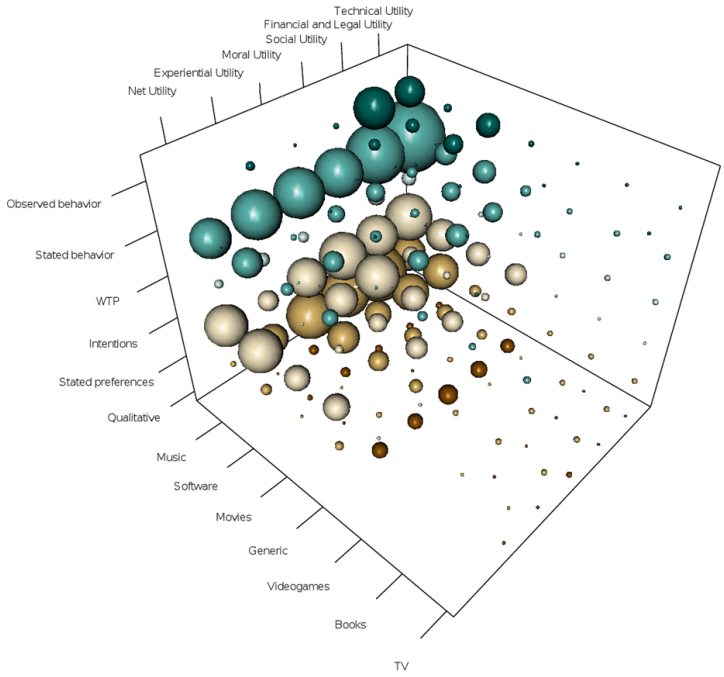



Figure 2.6: A natural languageprocessingvisualisation for topics in unlawful file sharing[36].

- Informative changes - a change in properties should carry a change in information. Any changes in the properties of the visualisation must have a reason behind it, otherwise it may confuse the viewer;
- Capacity limitations - people have a limited capacity to retain and process information so too much information can cause important information to be lost. This is why it's important either to limit information in a visualisation or to prioritise the most important information according to Healey's processing order, for example.

### 2.3.3 Use in Law

A particularly relevant example of visualisations used in law is Watson et al's visualisation of their summary of the nature of research into unlawful file sharing[36]. As shown in Figure 2.6, they used three dimensions with each dimension representing a list of topics. The size of the bubble shows how much research was done relating to this combination of topics. One of the many conclusions that can be seen from the visualisation is that discussion on unlawful file sharing mostly surrounds music, software and movies. It is a good example of how 3D visualisations can be used clearly and effectively to make a point.

## 2.4 Usability

### 2.4.1 User Interface

Nielsen collated his heuristics from existing sets of heuristics with the aim of creating a general set which is as good as possible at explaining the usability problems that occur in real systems[37]. These have been highly regarded since he finalised them in 1994. They are as follows:

- The user must be kept informed of the system status;
- The system must match the real world with concepts familiar to the user;
- The user must have control over the system, especially with an easy emergency exit;
- The system should be consistent with its looks and actions;
- The system should do everything within its power to prevent errors;

- The system should make the user's options visible so they do not have to memorise them;
- The system should be flexible so novice users can use with ease but expert users can use hidden accelerators for more efficient use;
- The design should be aesthetic and minimalist since every piece of information is competing for the user's attention;
- The system should give error messages that are expressed in plain language, that precisely indicate the problem and that constructively suggest a solution;
- The system should provide help and documentation for the user in case anything about the system is unclear.

Nielsen also presents how to measure the severity of usability problems. The severity of a problem is split into three categories: the frequency with which a problem occurs which is how commonly it occurs; the impact of when it occurs which is how easy it is to work around; and the persistence of the problem which is how frequently a user will be bothered by the problem.

### 2.4.2 Code Structure

Spinellis outlines the best practices for readable code[38]. Notably, he explains that the 'Don't Repeat Yourself' principle means to write what the code is doing and not how in this context as the code itself shows how an action is done. He also highlights how code should be somewhat self-explanatory and therefore, bad code cannot be rectified with lengthy documentation and the code should be rewritten instead. There are many textbooks on the principles of writing readable and refactorable code. Most suggest using object-oriented techniques for this[39].

Python documentation is standardised by the Python Enhancement Proposals. PEP 8 and PEP 257 are useful for the readability of Python code as they supply a style guide for the code and documentation respectively[40].

# Chapter 3

# Implementation

## 3.1 Chapter Overview

The following chapter describes the process of achieving the requirements specified in Section 1.4. Each section details the exploration of a different category and goes through its subsections in a logical order, rather than chronologically as many subsections were being completed simultaneously. When results are used as evidence for a subsection, it can be assumed that elements of other subsections that are used in testing are at their specified optimum states.

## 3.2 Natural Language Processing

### 3.2.1 Evaluation

#### Motivation

As established in Section 1.2, the project is a new application of technology in this domain. Any new use of technology in a domain brings scepticism and so to convince law academics of the validity of any results in the tool, there needs to be a simple, logical and consistent validation process. The validation process needs to be simple so the user can easily understand how the results came about and agree that this is a logical way to evaluate results. Furthermore, if the validation process is not consistent, they will not trust that it can identify whether a model is accurate.

As well as providing confidence to the user of the natural language methods, the evaluation techniques offer a guide to development as development decisions are made based on what yields the best evaluation results.

#### Classification

I chose to implement balanced accuracy as the score type for my classification. This is because, in this domain, 'positives' and 'negatives' are interchangeable because they represent two different equally valuable categories. To treat either category differently to another would be to misrepresent a classification algorithm. This was discussed in detail in Subsection 2.2.1.

I arbitrarily set the intellectual property and the creator class as positive and the human rights and the user class as negative. I implemented an algorithm that calculates how many true positives, true negatives, false positives and false negatives there are when given a set of predicted outs and a set of actual outputs. I then implemented an algorithm that takes these values as input and works out the balanced accuracy.

In order to ensure the evaluation method was consistent, I then implemented a cross-validation algorithm. The cross-validation algorithm involves randomly generating $n$ sets of training and testing documents where each document is a test document once and a training document $n - 1$ times. The model then trains and tests on each set and returns $n$ scores. The higher $n$ is, the more consistent the average of the results are. The average of the results gives a reliable idea of how successful the model was and a high variance may suggest more training data is needed.

In the rest of the document, whenever the word 'score' is used it refers to the average balanced accuracy score across a four fold cross-validation.

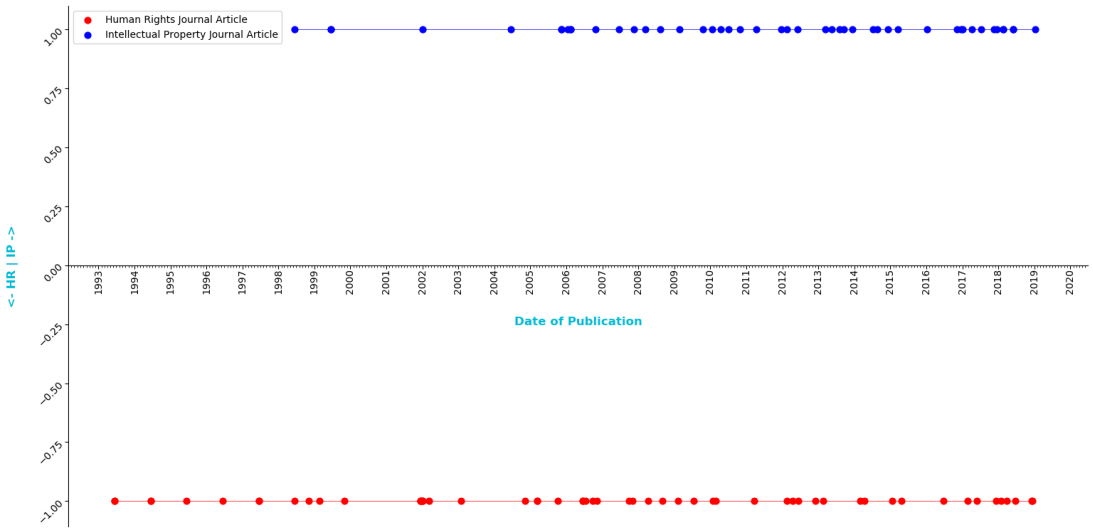



Figure 3.1: How test documents were classified when no pre-processing was done to the text.

#### Trends

To evaluate trends I used statsmodel's p-value attribute in their linear regression classes. I chose the statistical significance threshold to be 0.05, a standard discussed in Subsection 2.2.6.

Contrary to the spirit of statistical significance, in the early stages of the project, p-values were found to be wildly variable depending on what was used as test data. Some trends were found to be statistically significant once in every five times which in reality, means that the trend was probably not statistically significant. Because of this, I implemented cross-validation for trends as well. This algorithm extended the classification cross-validation specified above by generating trends for each set of training and testing data. This is particularly helpful in ensuring that no false conclusions are drawn because of one convenient set of data.

### 3.2.2 Pre-processing

#### Motivation

The aim of pre-processing is to get the dataset into an appropriate form for the model. Additionally, natural language processing should improve classification results by cleaning the text to make sure features that have effectively the same meaning are considered that way. They should also make the model more efficient by removing features that are not important enough to alter classification results.

#### From Web to Text

In order to create a classification model, a dataset of documents is needed. I downloaded a set of 411 journal articles from the journals specified in Table A.1, and the four treaties specified in Table A.2. The justification behind the number of journal articles downloaded and how they were picked is put forward in Subsection 3.2.4. All documents were downloaded manually from the internet in PDF format. It is worth noting for replication purposes that many of these journal articles have restricted access but are available to most academic institutions including the University of Bristol. The PDF files were then converted to plaintext using the pdfminer.

Further pre-processing is done at this stage to automate the extraction of metadata. This is discussed fully in Subsection 3.5.1 as it helps achieve a usability requirement.

#### From Text to Features

A classification with no more pre-processing yielded results that were plainly too good to be true. Cross-validation achieved an accuracy of 1.0 every time and the probability of a document being in its respective class was 1.0 without fail. These results would suggest that there is next to no change in the language in the fields towards each other in any of the documents as shown by Figure 3.1.

After analysing the text, I realised that on each article there were keywords which either overtly, by stating the journal of origin, or covertly, by including metadata of a certain format, revealed the article's parent class. It appeared that the model was classifying based on the format of the document as these were the most distinguishing features, contrary to the requirement to analyse trends in language.

I analysed each different document format and identified any phrases that of this nature. They were then identified by regular expressions and removed from the text. An example of this metadata being identified were documents from the Journal of Intellectual Property Law all had some variant of "Downloaded from http://www.academic.oup.com at the University of Bristol Library on 18 January 2019" vertically on the right-hand side. Before pre-processing each letter was a different feature but when new lines were removed in pre-processing, they each became whole words. This meant that the word University once per page would make it clear that there was a very high chance that this document is an intellectual property document. I identified the phrase using the regular expression:

$$\backslash nl\backslash n\backslash nD\backslash no\backslash nw\backslash nn[a]^{+}?[\backslash n]^{7}$$

and removed it. This was done with 17 expressions in total.

After removing these expressions, the score of in cross-validation averaged 0.58. This suggests that the model is not much better than chance. To improve this, I performed pre-processing techniques that commonly performed well: cleaning the text of punctuation and new lines, removing features that include integers, and changing all letters to lowercase. After this, I observed that many words were being split in two, commonly if they had certain combinations of letters in them such as 'ff' or 'fi'. I discovered that this was to do with letters being processed by PDFs into Unicode blocks called ligatures. These could not be interpreted by pdfminer. Therefore, I identified a list of all these ligatures which included the discovery that some ligatures were converted into a code, such as '(cid:123)'. I then replaced them, and the space following them, with the individual letters' characters which they represented.

After these steps, the results were drastically improved with a cross-validation score of 0.99. This result more than suffices in terms of accuracy but I carried on to see if removing stopwords and lemmatisation would maintain the same accuracy while reducing the feature count. Lemmatisation was performed using an nltk library function. I chose lemmatisation over stemming so the origin word is more obvious should analysis be done using the processed words. nltk also has a function for removing stopwords from text but it does not let you append your own stopwords, for this reason, I implemented my own algorithm to do this. I added words to the list such as 'abstract' and 'references', which again could have given away what journal an article was from based on the required structure of the journal article.

As can be seen in Table 3.1, all results after the first set of processing methods are similarly high. There is a slight decrease when stopwords are removed, possibly because of the words indicating the journal's structure. Performing both these methods reduces the number of features by more than a factor of three. I am choosing to use all the processing methods because it has the highest score that involves removing possibly indicative but irrelevant stopwords and it has the fewest number of features.

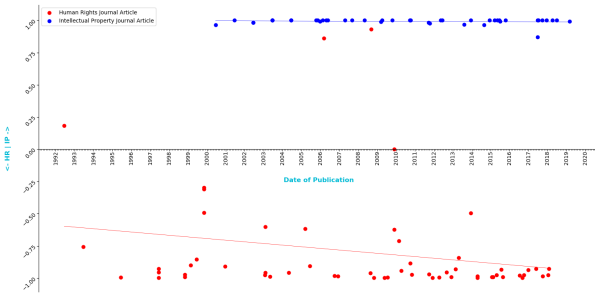
| Level of Pre-processing | Number of Features | Accuracy Score |
|---|---|---|
| 000 | 257604 | 0.579 |
| 100 | 77669 | 0.991 |
| 110 | 77356 | 0.981 |
| 101 | 71464 | 0.992 |
| 111 | 71363 | 0.984 |

Table 3.1: The results for defferent levels of pre-processing where the three bits in the 'Level of Pre-processing' column represent what pre-processing methods were performed with the most significant bit representing removing punctuation, new lines, integers and glyphs; the middle bit representing removing stopwords; and the least significant bit representing lemmatisation.

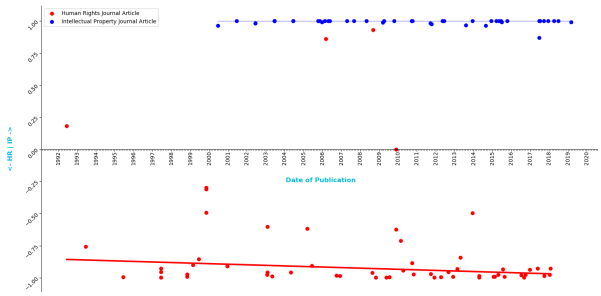
### 3.2.3 Trends

#### Method

Since the two classes are polarised, doing a line of best fit for all the data would be useless, giving a line approximately in the middle without slope. Instead, I chose to perform linear regression on each class separately. This will show how the language in each set of journals has changed over time.




Trends illustrating standard least squares regression (a)




Trends illustrating robust least squares regression (b)

Figure 3.2: The same dataset with different types of linear regression performed on it.

The method of generating trends needs to be robust. I initially used statsmodel's ordinary least squares class but found that some extreme outliers forced this to exaggerate trends. I then switched to a robust version of ordinary least squares which appeared to be more resistant to the outliers as shown by Figure 3.2. It also improved the average p-value from 0.556 and 0.679 to 0.377 and 0.567.

### 3.2.4 Human Rights-Intellectual Property Classification

#### Motivation

Here, I will look to model what a typical human rights and intellectual property document looks like respectively. I will test its accuracy on a different set of test documents and detect whether any significant trends have appeared based on these trends. In classification, we are looking for the highest possible level of accuracy to show that the model works and the lowest possible p-value in order to find a trend in the data.

#### Training Model

In my implementation, I chose to use scikitlearn's support vector machine class as it is simple and often outperforms naive Bayes. This class has an attribute that gives the probability that each document belongs to a topic, $p(hr)$ and $p(ip)$. To get a value which I will later plot, I will use a simple equation, Equation 3.1.

$$hr\text{-}ip\ score = p(ip) - p(hr) \tag{3.1}$$

#### Training Documents

Dr. Blakely thought that it may be interesting to use the treaties specified in Table A.2 as the training documents as they are the legal basis for the fields. However, four documents turned out to be not enough to train a model. The results are fairly meaningless with a consistent score of 0.5 and no meaningful trends shown by Figure 3.3. Instead, I chose to use three-quarters of the articles in the dataset I downloaded in Subsection 3.2.2. This is a high enough proportion of the documents to give an accurate classification of the test documents but low enough to give the opportunity to find some statistically significant trends.

#### Feature Count

I used a bag of words model for features. This involved separating each word in an article and including each unique word as a feature.

I started counting features by simply implementing an algorithm to count the number of occurrences of each term in a document. This gave a suitably high result of 0.97. However, I deemed that occurrence count might not be solely considering the language of the article. It is likely that each journal has word limits for article submission and this could be reflected in the feature count and affect classification. This can be seen by the fact that 27% of intellectual property articles are between 4,000 and 5,000 words while only 15% of human rights documents are. Although, the total word count is not a feature in itself, the fact that the average feature will probably lie in a certain space means that a document between these word counts would be more likely to be classified as an intellectual property document. This is another

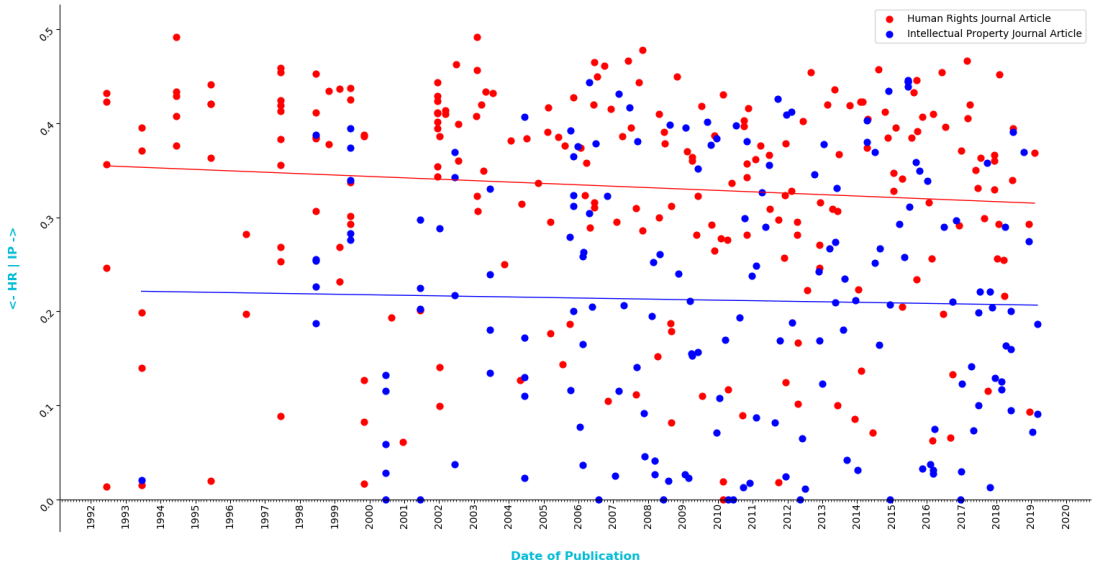



Figure 3.3: The results of testing when using four treaty documents to train.

case of an attribute other than the article's language influencing classification which goes against the requirements. Using term frequency in the feature count removed this possible bias.

I implemented algorithms for four types of feature counts based on their equations given in Subsection 2.2.4. Their results are shown in Table 3.2. It shows that tf-idf-cf has the highest accuracy and the lowest p-values.

| Feature Type | Average Accuracy Score | Average HR p-value | Average IP p-value |
|---|---|---|---|
| word occurrence | 0.973 | 0.222 | 0.602 |
| tf | 0.962 | 0.588 | 0.570 |
| idf | 0.940 | 0.191 | 0.370 |
| tf-idf | 0.946 | 0.213 | 0.693 |
| tf-idf-cf | 0.984 | 0.227 | 0.157 |

Table 3.2: The results for different types of features.

#### Trends Found

On the rare occasion a statistically significant trend was detected, human rights documents began to include more typical human rights language. Cross-validation showed that this was particularly infrequent, happening once in the four folds at most, and therefore, not worth taking seriously.

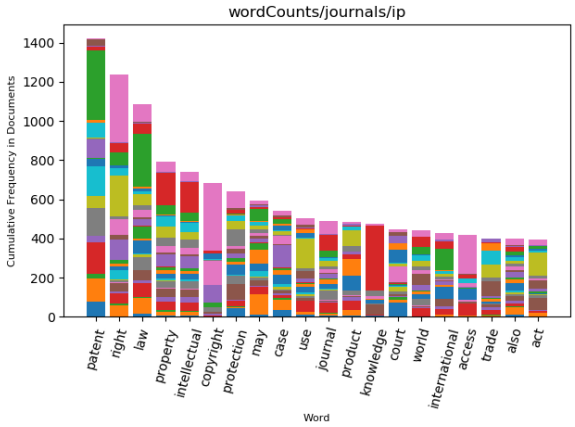
## 3.3 User-Creator Classification

### 3.3.1 Motivation

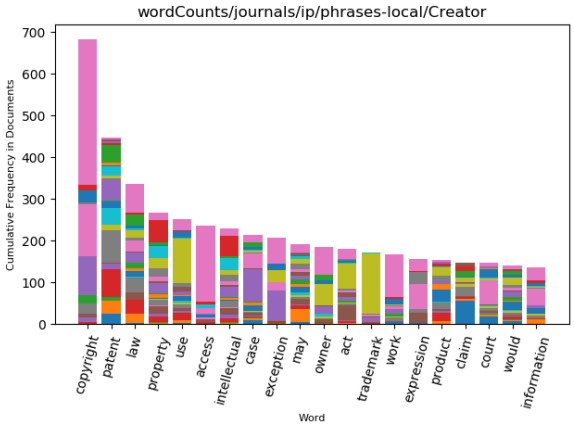
Here, I will look to model what a typical document which portrays the legal climate as user-benefiting as opposed to creator-benefiting and vice-versa looks like. Again, the aim of this classification is to get the highest accuracy possible with the lowest p-values to find a trend.

### 3.3.2 Ground Truth

Unlike the human rights-intellectual property model, there is no extensive dataset for indicating whether an article is user-benefiting or creator-benefiting. Instead, Dr. Blakely supplied a list of keywords that suggested each. These are given in Table A.3. During my preliminary investigations, I prompted Dr. Blakely to extend her keyword lists by visualising the word occurrence count in each class of journal articles and the word occurrence count in each class of journal articles in sentences where a keyword




Total word counts in intellectual property articles (a)




Word counts in intellectual property articles in sentences that continaed creator keywords (b)

Figure 3.4: The visualisation of word counts as part of preliminary investigations.

occurs. These visualisations for intellectual property journals and creator keywords are shown in Figure 3.4. Dr. Blakely stated she did not wish to add more words to the list.

In order to evaluate my model, I assigned my own document document-by-document ground truth to 20 randomly selected documents since Dr. Blakely was not available. This must be taken with quite the pinch of salt since my experience in this domain does not stretch beyond this project. Also, since the model does not train on documents itself, cross-validation was not used for evaluation for this model, the same 20 documents are repeatedly checked to see if their prediction matches ground truth. In reality, the small test set and that curated by someone who knows little about the fields, indicates that the evaluations that took place in this section were unreliable.

### 3.3.3 Method

I attempted to use a lexicon-based system to give each document a score on how it scales based on a creator score and a user score as shown in Equation 3.2, where positive scores are classified as creator-benefiting and negative scores as user-benefiting.

$$user\text{-}creator\ score = creatorscore - userscore \tag{3.2}$$

I began this by considering every sentence as indicating an independent sentiment on whether the legal climate was user-benefiting or creator-benefiting. I then assigned *creatorscore* as the proportion of sentences that included a creator keyword and treated *userscore*. This resulted in a balanced accuracy score of 0.75. Because the keywords chosen for creator-benefiting were more common than the keywords for user-benefiting, this resulted in a large proportion of documents being classified as creator-benefiting. The average p-value upon cross-validation was 0.196.

This model's main weakness was that it did not really capture the sentiment since it ignored negators which completely changed the meaning of the sentence. For example, the clauses this means that the user will have access and this means that the user will not have access are both scored as benefiting the user when the latter clause clearly does not benefit the user. My next approach, therefore, was to extend the model to consider the sentiment of each sentence. This included creating five lists based on Mandal and Gupta's[18]: comparative positive, comparative negative, superlative positive, superlative negative and negators. I implemented an algorithm that assigned a *creatorscore* of 1 if the sentence contained a word from the creator keyword list. The algorithm then checks for any words in the above lists. The current score is added to if a positive word comes up and then subtracted from if a negative word comes up. The scalar used is 0.65 for a superlative and 0.35 for a comparative. The score is multiplied by -1 if a negator word comes up. So, for example, the sentence "What we can do is carefully study the development of patent pooling regulations in other countries and implement the most successful models for protecting IP rights and promoting fair competition" now receives a *creatorscore* of 1.65 because of the word patent which is enhanced by the word most. I then divided by the number of sentences in the article. The equivalent is done with *userscore*.

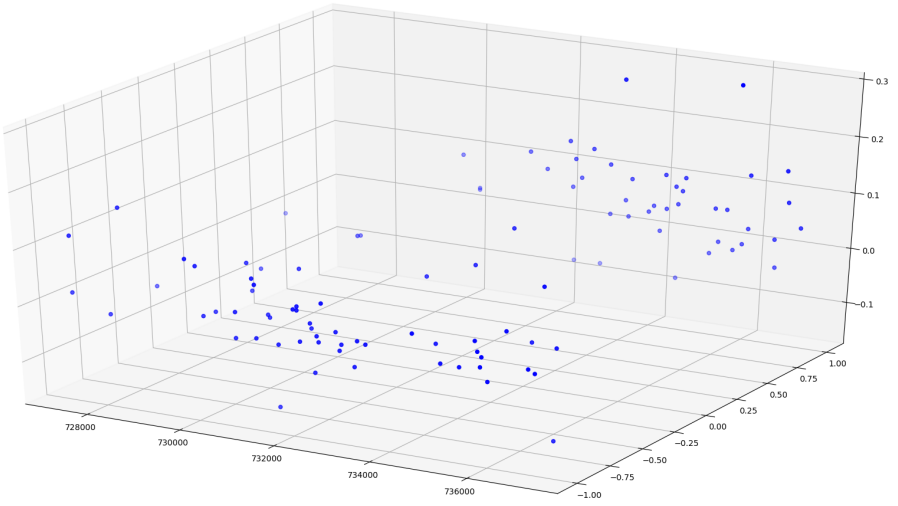



Figure 3.5: Documents plotted with no further configuration.

This method did not actually cause any classification changes as the lists were not extensive enough. It also raised the average p-value in cross-validation to 0.465, making statistically significant trends very rare. Since this method was not developed enough, it produced no trends over cross-validation of p-values and there was no way to be sure this model was accurate since my evaluation method was poor, I decided to not use this option. I chose to use the simpler option as it's clearer what it represents.

Both these methods were weak in that the keyword lists contained some of the same words, leading to sentences containing keywords from both lists cancelled out when they are likely to have higher importance than usual. This can be

#### Trends Found

Trends were found approximately half the time with the chosen model. As this model had not been properly evaluated for accuracy, however, I fully explained how the model worked to Dr. Blakely and explained that proper evaluations would need to be done before any conclusions were drawn.

## 3.4 Visualisation

### 3.4.1 Evaluation

#### Motivation

The result of the natural language processing is a collection of probabilities, scores and gradients. This would mean very little to the average law academic without explanation. The purpose of the visualisation is to explain these numbers so anyone can understand them with as little comprehension as possible. Simply plotting each test document's date of publication on the x-axis against the results of its human rights-intellectual property classifications on the y-axis against the results of its user-creator classifications on the z-axis with no further configuration of the visualisation gives Figure 3.5. This was plotted using Python's 'matplotlib' library. The 'Axes3D' class allows the user to change the angle of the visualisation using their mouse.

A successful evaluation method will convert Figure 3.5 into the perfect visualisation for the target audience in minimal time. This will be possible if the evaluation method allows for informative communication and allows visualisations to be compared. It is worth noting that the visualisations in this section are not shown in chronological order and are simply placed in the order which best shows the effect of the change being discussed. It is also worth noting that each visualisation does not necessarily show the same data plotted. It should be apparent that the changes discussed apply to the task whatever combination of data is plotted.

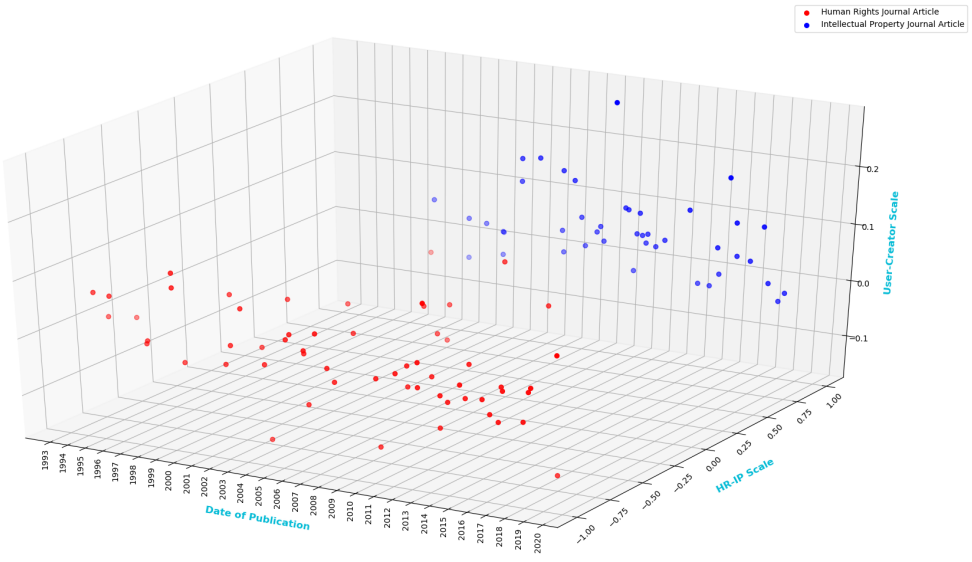



Figure 3.6: A visualisation advanced from Figure 3.5.

#### Criteria

Visualisations were evaluated based on the following criteria:

1. The visualisation and its axes' meanings are self-explanatory;
2. It is clear where a point lies on the visualisation's axes;
3. The visualisation has all the possible information that is desired by the audience;
4. The visualisation clearly indicates any trends deduced by the models;
5. The visualisation is aesthetically pleasing.

For ease of notation, a criteria will be referred to as 'Criteria $n$' where $n$ is the number it has been given above.

I evaluated the visualisation iteratively with Dr. Blakely and my dissertation tutor, Dr. Mowbray throughout to guide the process of constructing the visualisation. A Likert scale was be used to quantify results so they can be compared. I chose this as opposed to a continuous scale as a continuous scale may not have been defined enough and may have left inconsistencies in answers. The Likert scale was in regards to each of the five statements above and asked the user to select a statement from one to five where one represented "Strongly Disagree" and five represented "Strongly Agree". This allowed me to compare criteria so I knew where improvement should be prioritised. I also requested comments after each of the statements for an explanation of why the selected answer was given so I would know where to improve on each criteria.

### 3.4.2 Point Location

Figure 3.6 shows a lone 3D visualisation, which is what Dr. Blakely suggested. To represent a 3D space on a 2D screen, matplotlib uses luminance to portray depth. The fainter a point is, the further away it is. Dr. Mowbray disagreed with Criteria 2 as she thought this was not immediately clear enough.

A 3D solution to this problem is particularly hard to find because we are limited to 2D screens. This led to the idea that there could be additional 2D axes to compare each of the three axes individually. This would allow for clearer point location and trend identification as can be seen in Figure 3.7. The other two combinations are shown by Figure 3.16 and Figure 3.10. These additions prompted Dr. Mowbray to strongly agree with Criteria 2.

### 3.4.3 Identifying Classes

In order to see trends in each different category, it must be immediately obvious which documents belong to which journal category. I chose the preattentive feature, colour, to distinguish between human rights

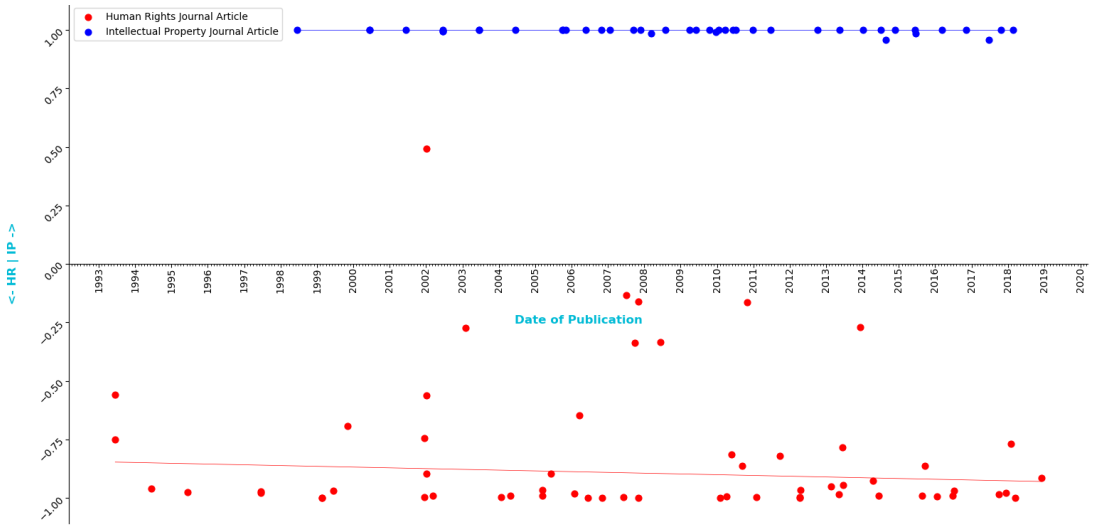



Figure 3.7: A 2D visualisation of human rights-intellectual property plotted against time.

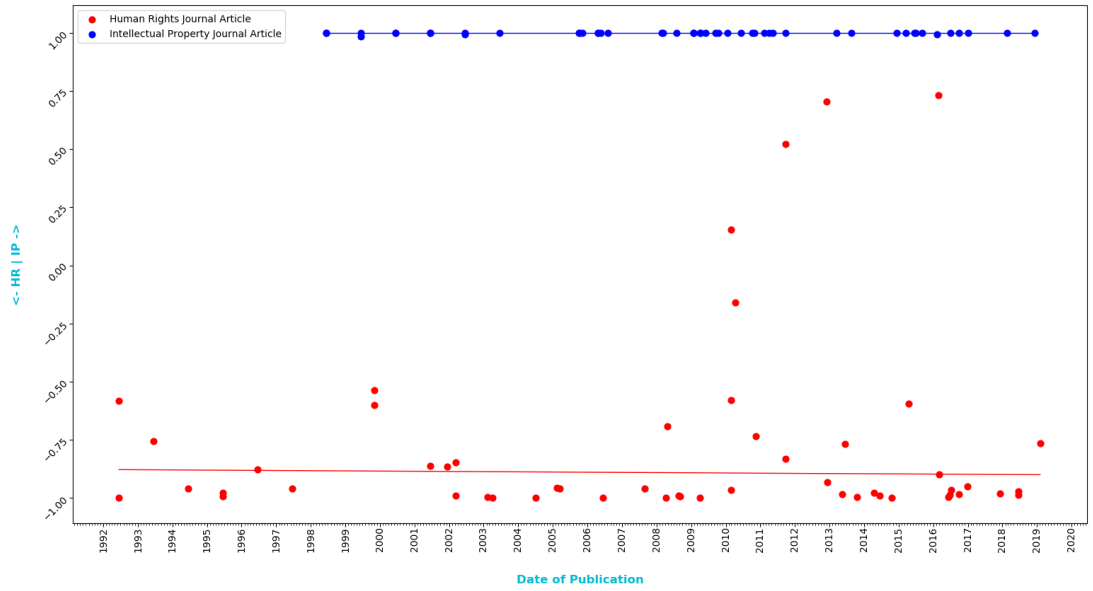



Figure 3.8: A 2D visualisation with its axes at the left and bottom, the default positions.

journals, coloured in red, and intellectual property journals, coloured in blue, as shown in Figure 3.6. As well as making it easy to identify trends, anomalies become particularly obvious due to the combination of preattentive features of colour and proximity. If a red point is a large distance from other points then it is obvious this is an anomaly.

Initially, the 2D visualisations had their axes placed left and bottom as default as shown in Figure 3.8. Where possible, I moved the axes to the zero point in the other axes. This made the most of the Gestalt principle of enclosure as the axes now enclose all points based on their predicted classification. In Figure 3.7, it can be seen that one document has been classified incorrectly as it is below the x-axis like the rest of the red points.

### 3.4.4 Labelling

Initially, Dr. Mowbray disagreed with Criteria 1. She commented that the visualisation was not self-explanatory because it was not clear what the points represented as the legend labelled blue points with 'Intellectual Property' and red points with 'Human Rights' which could refer to a number of things. To rectify this, I changed the labels to 'ntellectual Property Journal Article' and 'Human Rights Journal Article' respectively. Dr. Mowbray also commented that it was not clear what the x-axis represented as

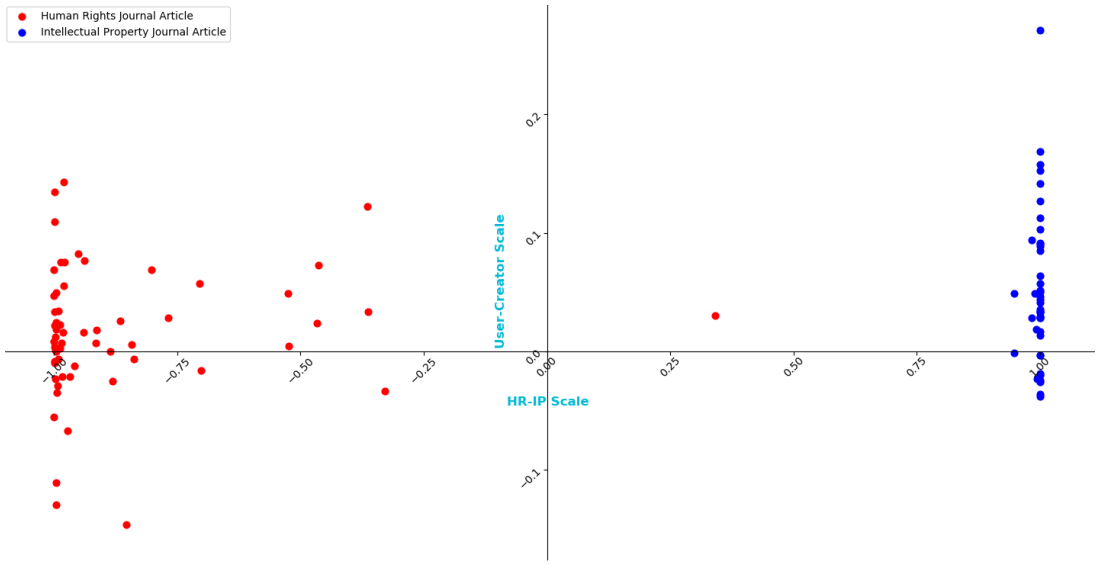



Figure 3.9: A 2D visualisation with basic axes labels.

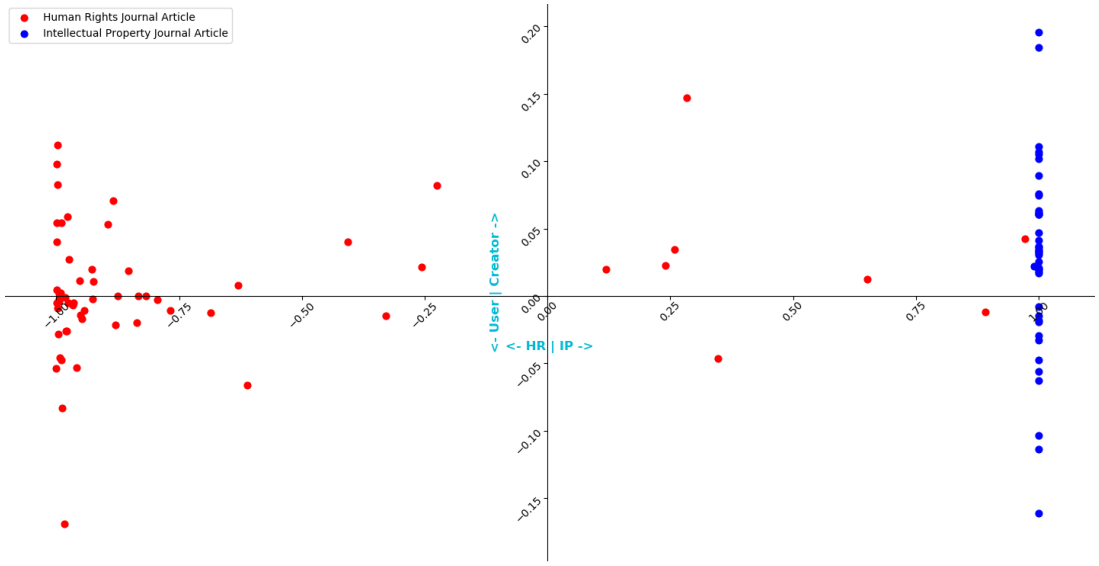



Figure 3.10: A 2D visualisation with arrows in axes labels.

the x-axis was solely labelled with 'Time' and this was not clear enough as it could refer to any number of things regarding a document. I rectified this by changing the label to 'Date of Publication'. Upon these changes, Dr. Mowbray neither agreed nor disagreed with Criteria 1.

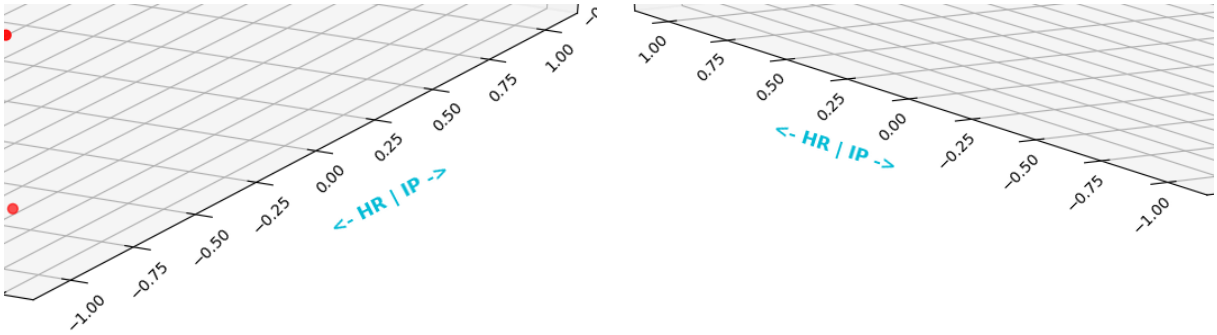
Dr. Mowbray still did not agree with Criteria 1 because it was not immediately clear which end of the y and z-axes represented which class as seen in Figure 3.9.

To solve this, I changed the axes, 'HR-IP Scale' and 'ser-Creator Scale' to `'<- HR | IP ->'` and `'<- User | Creator ->'` respectively so the arrows were pointing in the direction that their class was represented by. This is shown in Figure 3.10.

I then attempted to apply the same label change to the 3D visualisation. This was flawed because the label started off pointing the right way, shown by Figure 3.11a but when the axes flip on the user's command, the labels do not flip causing the 'HR' arrow to be pointing at the intellectual property side of the visualisation and vice versa, shown by Figure 3.11b. This could have caused false conclusions being made based on the visualisations so had to be avoided.

As a solution, I annotated the graph at the beginning of the x-axis to indicate at which sides of the graph indicate which classes, as shown in Figure 3.12. This made it clear which end of the axes meant which class and also slightly improves the user's ability to locate where points are.

In the interest of consistency, I attempted the same solution for 2D graphs but this was considerably




Axes label pointing the right way (a) Axes label pointing the wrong way (b)

Figure 3.11: The same axes labels as Figure 3.10 on 3D axes.

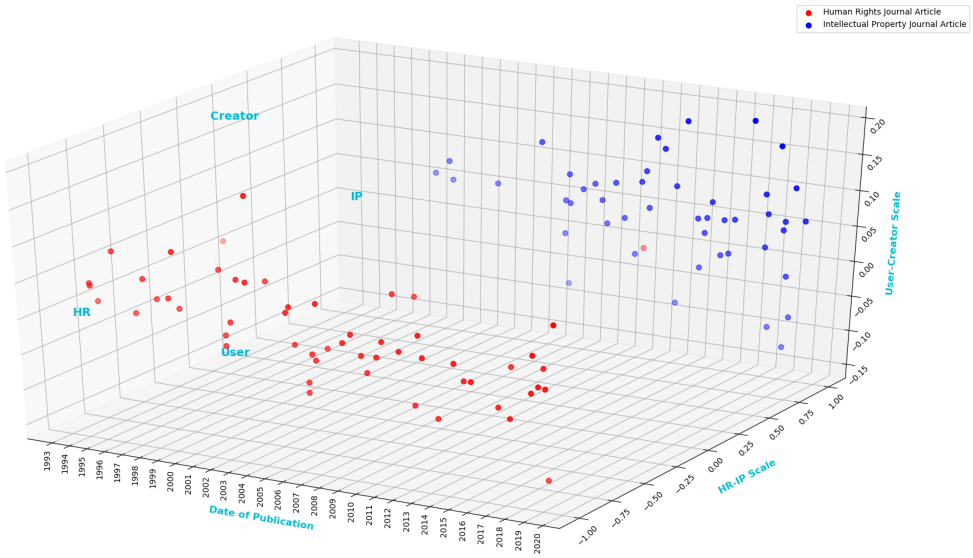



Figure 3.12: The 3D visuaisation from Figure 3.6 with annotations to make axis polarity clear.

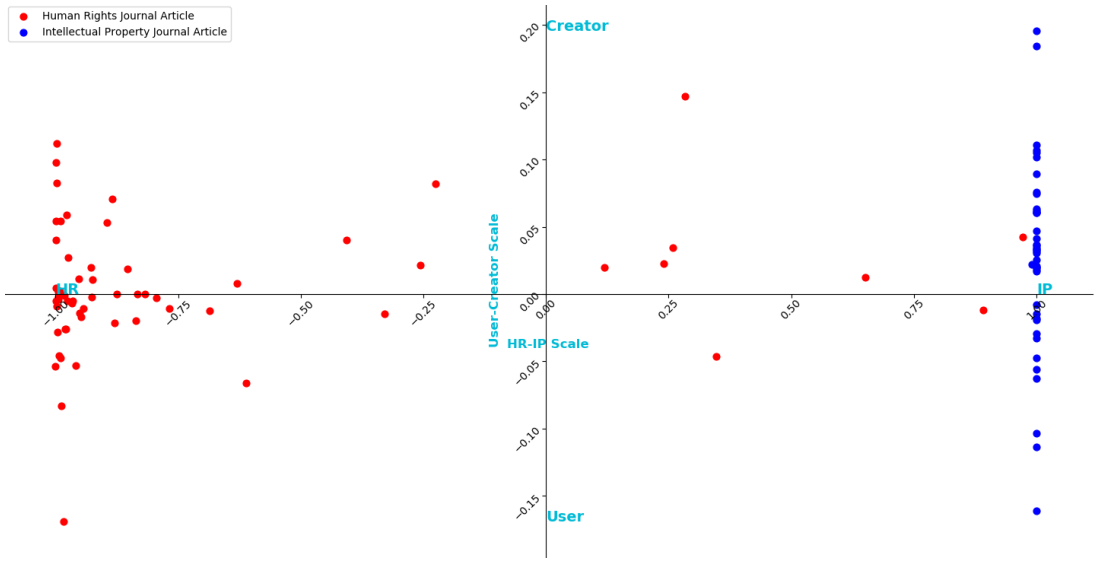



Figure 3.13: 2D visualisation with axis polarity labelled by annotations.

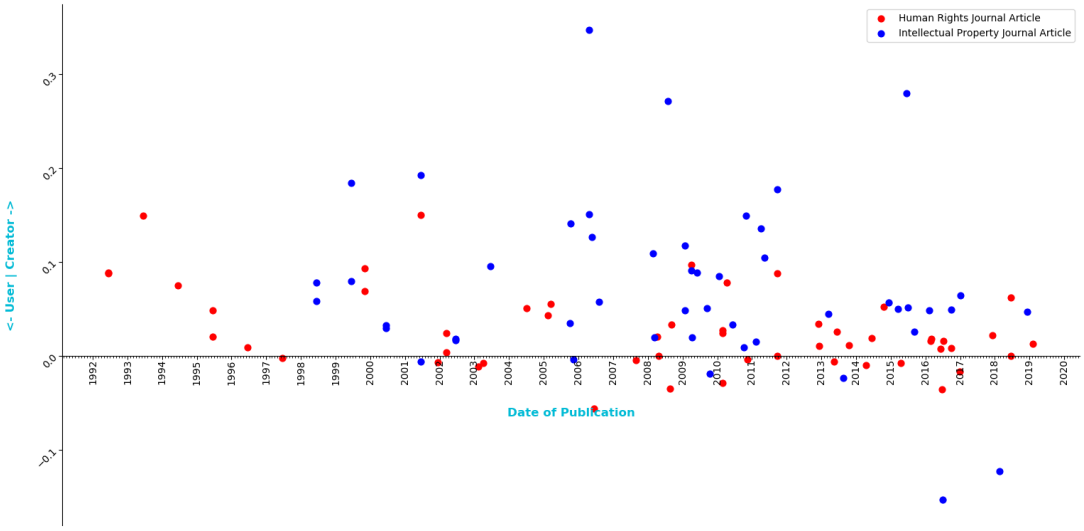



Figure 3.14: A 2D visualisation without any trend lines.

worse than the solution found in Figure 3.13 because the labels were covered up by points due to the polarised nature of the dataset. Therefore, I chose to stick to the original solution.

### 3.4.5 Trends

Initially, trends had to be deduced by eye. This could be particularly difficult when you are trying to deduce the trend for a subset of the data points and the other data points are interspersed within the subset as in Figure 3.14. In order to make trends more obvious, I plotted the results of the linear regression from Subsection 3.2.3 by finding the y-values for a set of x-values based on the result's gradient and y-intercept as in Figure 3.15. This visualisation could be deceiving to users as a line that is statistically significant is presented as significant as one that is not. For this reason, I made lines that are found to be statistically significant, six times thicker than ones that are not as in Figure 3.16. I chose to keep the line that represented the non-statistically significant trend but make it half as thick as it originally was. The thinness shows its lack of significance but its existence allows the user to investigate further if they find a detail, such as the direction of the trend, interesting.

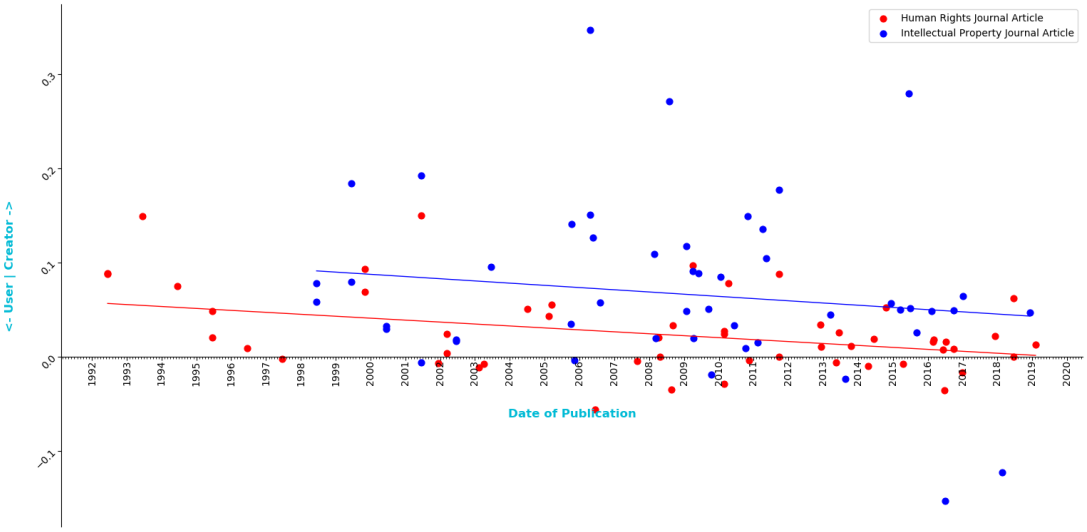



Figure 3.15: A 2D visualisation with standard trend lines.

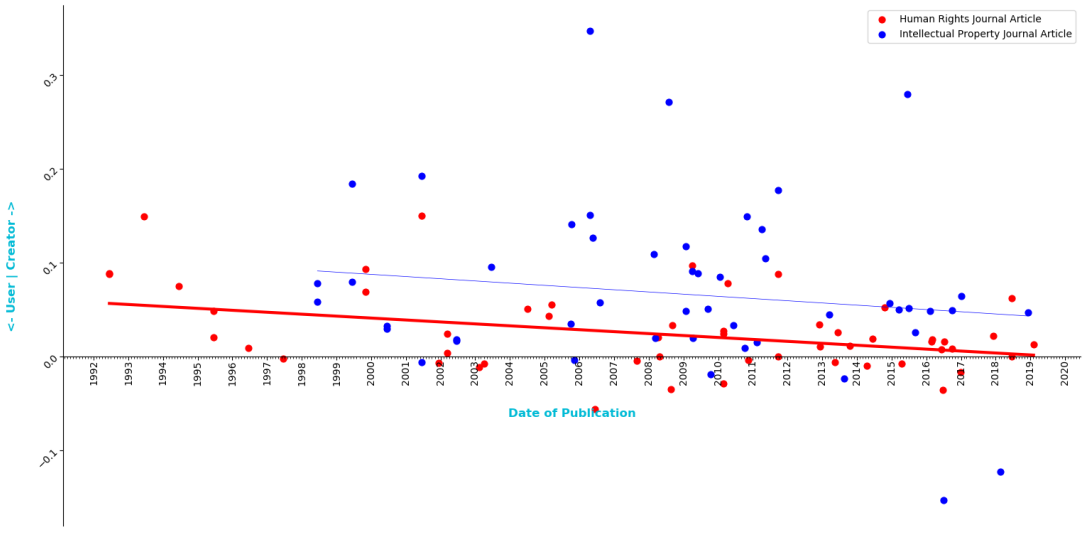



Figure 3.16: A 2D visualisation with bold trend lines if statistically significant, thin if not.

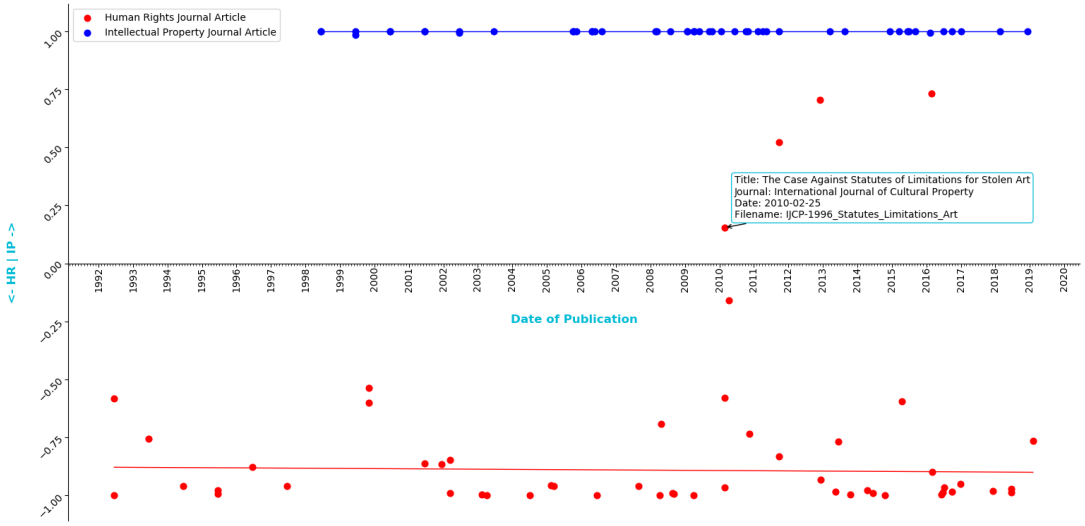



Figure 3.17: An annotation prompted by clicking a point.

### 3.4.6 Investigating Documents

Anomalies are important as they show where the two fields intersect most. The fact there is an anomaly is not too interesting without being able to understand why. Initially, there was no way of finding out about which point represented which document. I changed this by adding an event listener to listen out for clicks. If a click occurred somewhere a point was, the details of that point appear as in Figure 3.17. Now any anomalies can be investigated to further understanding of where the fields intersect most.

### 3.4.7 3D Enclosure

Since Dr. Blakely originally stated her request for a 3D visualisation, I attempted for that to carry some of the same advantages that the 2D visualisations have. I particularly wanted the same use of enclosure by the axes to indicate the predicted class that was used for the 2D visualisations. However, it is not possible to move a set of 3Daxes to the inside of a visualisation. As an alternative, I plotted two surfaces. One surface extended across the plane where $y$ equals zero; this split the documents that were predicted to be classed as user benefiting as opposed to creator benefiting. The other surface extended across the plane where $x$ equals zero; this split the documents that were predicted to be classed as human rights based as opposed to intellectual property based.

I then proposed that the surfaces could be a cleaner way of solving the problem of axes polarity discussed in Subsection 3.4.4 by colouring the surface with a colour map and attaching a colour bar to explain what the colours mean. This is shown in Figure 3.18 where the z-axis varies by colour. I did not manage to get both axes to vary with colour so a mixture of solutions is used here. Having used colour for the predicted classifications, I now deemed it to be a cause of confusion to also use colour for the true classifications. I instead used another preattentive feature, shape, to distinguish documents that came from human rights and intellectual property journals. In this case, circles that were above the orange plane were wrongly classified. I decided not to opt with this visualisation because there is too much going on to process easily.

## 3.5 Usability

### 3.5.1 User Interface

#### Motivation

Much of the testing and developing was performed on a Linux terminal. This is unsuitable for the final product as most law academics do not have Linux and are not used to a terminal-based environment. This is why a graphical user interface must be made which can be booted from a desktop. The user interface must maximise the amount of time spent on analysis of results by allowing intuitive access to all features.

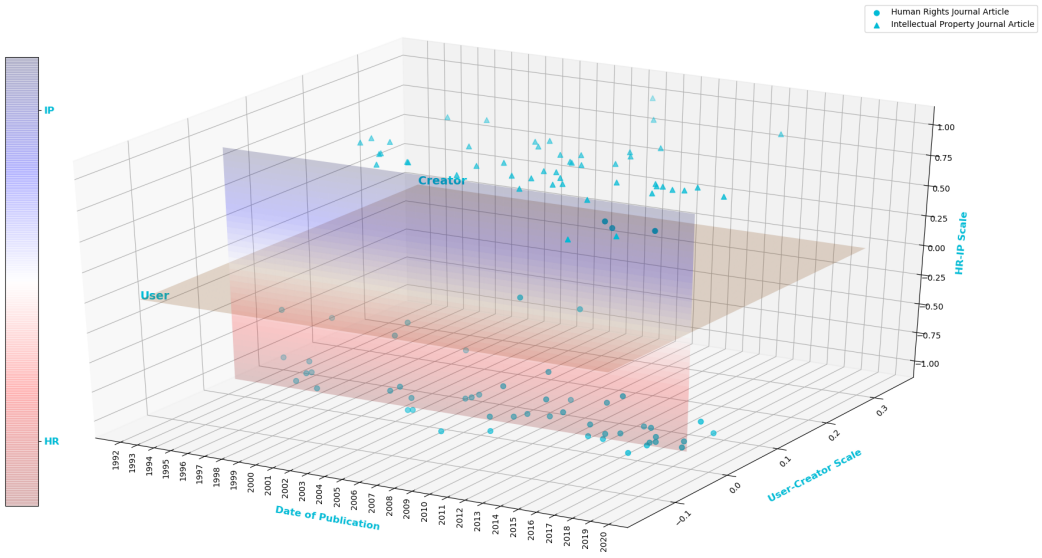



Figure 3.18: A 3D visualisation with planes dividing predicted classifications.

### Executable

As the intended users are not used to a terminal environment, nor will they have Python installed, I used the 'PyInstaller' library to create an executable for the tool meaning that they do not have to install anything extra to run the program, just the data files and the executable. This saves significant time as the user does not have to learn any complicated computer science techniques. I prioritised Windows for the executable as every academic should have some access to this operating system through their university.

### Functionality

The user interface allows the user to view visualisations and their results with ease. It also allows them to cross-validate results via a button. The most useful functionality is regarding the documents. The user can edit the metadata in the user interface, including the title, journal and date, in case automatic extraction was not correct. This means that they do not have to go through lengthy data files for this. They can also add new documents and remove documents as well as selecting which documents are test documents. This gives them control over the input of the model so they can test article of their own interest and see where they fall on the axes. For further ease of use, there is a pop-up which allows for more options on test data and allows the user to deselect all documents as test data, select all or select a random 25% of the documents. The user can also open PDFs directly from the user interface which is particularly useful as when the user reads the information about an anomaly in the visualisation, they do not have to leave the user interface to open the PDF and analyse the language for themselves.

### Design

The user interface was programmed using Python's 'Tkinter' library as it is known as the de facto standard for Python user interfaces. I started off by drawing wireframes, shown in Figure 3.19, for Dr. Blakely's approval so I did not waste time creating a complicated user interface for it not to be suitable. I used Nielsen's heuristic of using real-world concepts by using tabs, separated by 'Visualisation', 'Results' and 'Documents. These are clearly defined and separate sections. Dr. Blakely approved of the wireframes so I moved onto implementing them in Python.

Tabs were implemented using tkinter's 'Notebook' class, the buttons' functionalities were implemented in call-back functions and the scroll bar was implemented by putting one of tkinter's 'Scrollbar' objects into a 'Canvas' object. The three tabs can be seen in Figures 3.20, 3.21 and 3.22. One notable feature is how the results for a statistically significant trend line will turn bold. This is using visualisation principles that important information should stick out via preattentive features as there is a lot of text on the screen so the important one needs attention drawn to it.

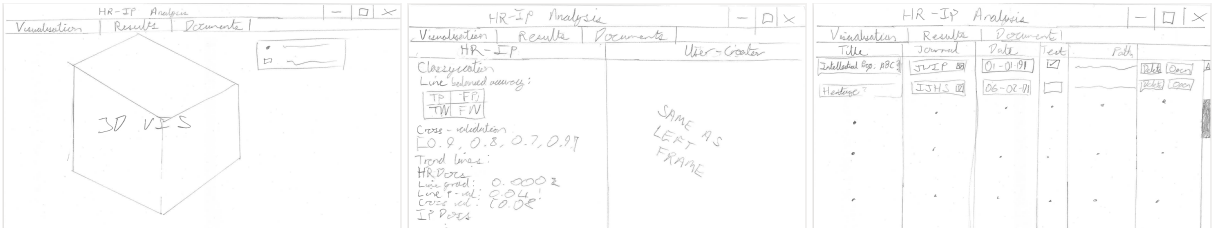



Figure 3.19: Wireframes drawn for user interface.

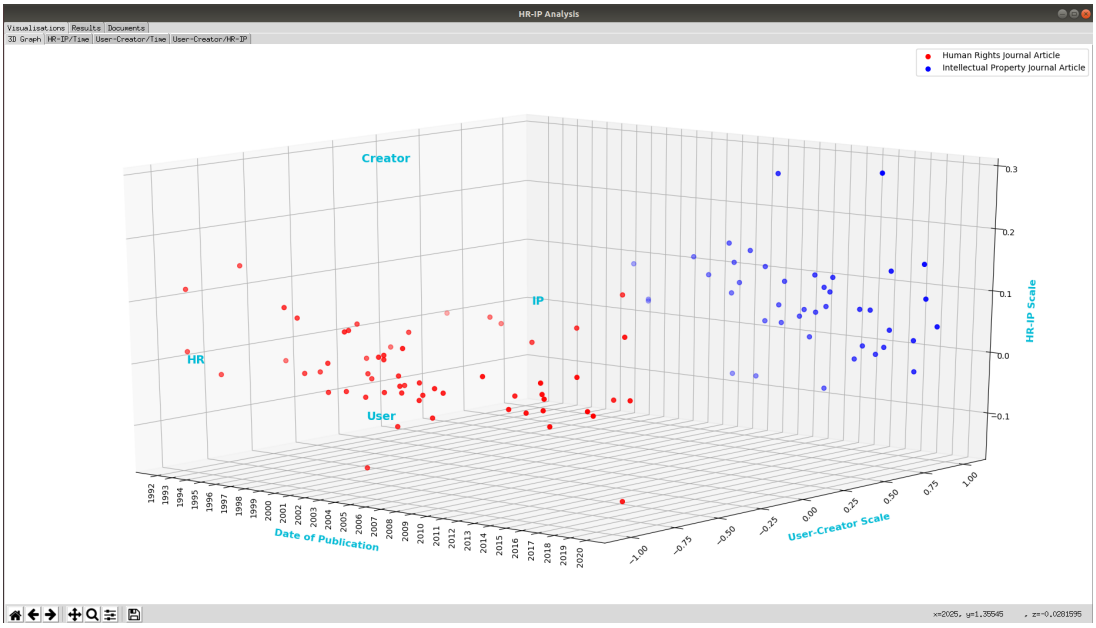



Figure 3.20: The tab of the user interface displaying the 3D visualisation.

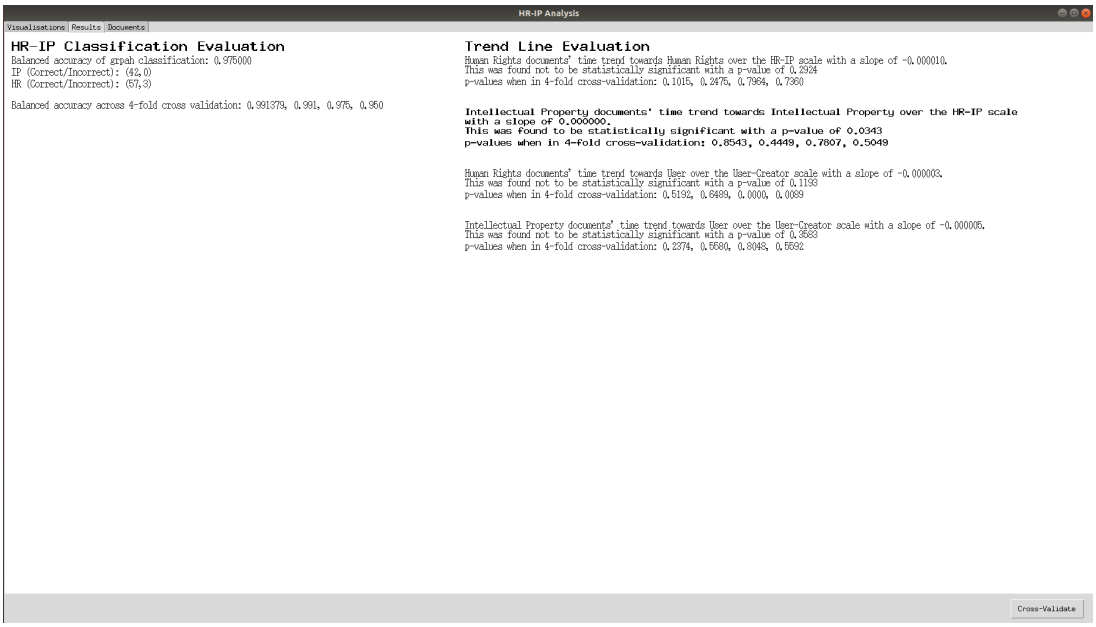



Figure 3.21: The tab of the user interface displaying the evaluation data.

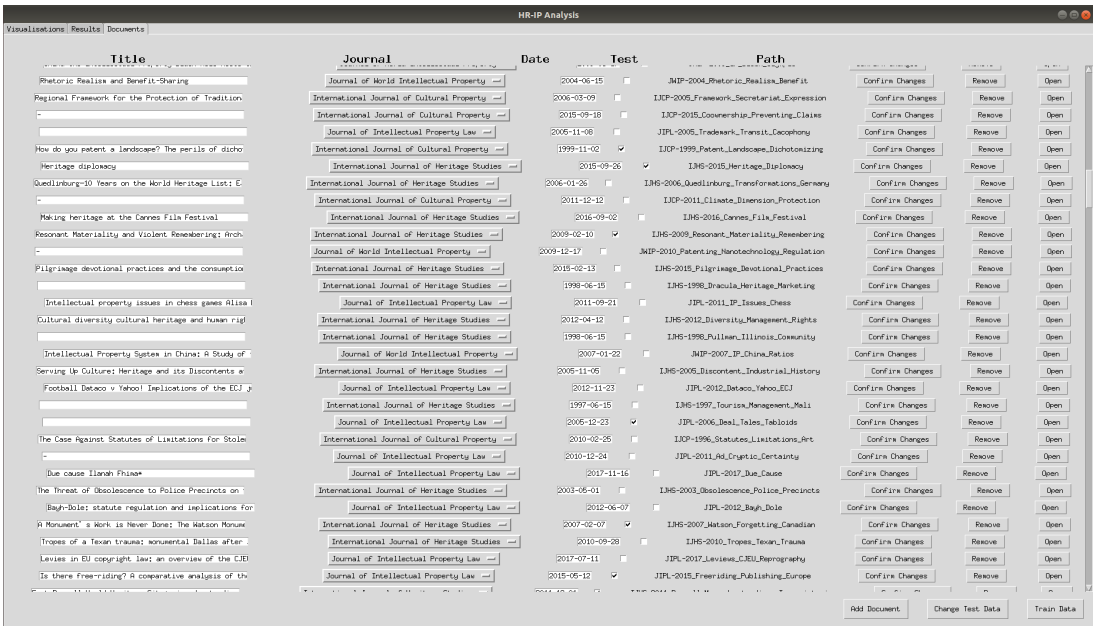



Figure 3.22: The tab of the user interface displaying the list of documents used.

### 3.5.2 Speed

#### Automating Metadata Assignment

As previously mentioned, the date an article was published, the journal it was published in and the title of the article is needed. Since a large number of documents are likely to be uploaded at once, it would be laborious to manually enter this data. For this reason, I implemented an automatic metadata extraction algorithm at the pre-processing stage. This proved to be fairly challenging due to the mixed origins of the documents.

The algorithm takes the PDF of a journal article as input and outputs the article's title, journal of origin and date of first publication. A '-' as a field's output is used to indicate that the algorithm is unsure of what the field should be. This is preferred to the field being wrong since if the algorithm is correct on the first few instances, the user may assume that the algorithm is correct in all instances and not check the rest. A number of wrong fields could lead to the contamination of the training data with an incorrect ground truth assignment or the regression lines with an incorrect date assignment.

There were two means of determining the fields. The first of which was to use the PDF's metadata. This was simple to implement by using the 'pdfminer' library to assign the PDF's 'Creation Date' property as the date of publication and its 'Title' property as its title.

There are a few limitations to this method. There are no means to get the origin journal from the PDF's metadata so ground truth would have to be downloaded manually. The method is also very inconsistent. Titles are often just 'No Job Name' or 'untitled' and the 'Creation Date' is often a wildly inaccurate representation of when the document was first published. This is because documents published before a certain date in each journal were not created for PDF and were scanned into PDF format later meaning their 'Creation Date' was far later than their actual first publication date. If a field was empty it was marked with a '-'. The result of the algorithm over 100 samples is shown in Table 3.3.

The second means of determining the fields was to extract the information from the text. This was done by identifying different formats of documents in the dataset and identifying where each of the fields is found in each of the different formats. Regular expressions are then used, first to identify the journal and then to identify the document format. Then if the document format indicates that the PDF metadata will be correct, then the PDF metadata is used, otherwise, more regular expressions are used to identify the date and title. For example, to identify the date of publication of a document in the International Journal of Heritage Studies between 1999 and 2019, the following regular expression was used:

$$[a] * \text{To cite this article}[a]^{*}\text{`}[0-9]^{4}\text{'})$$

where the single speech marks indicate the target string.

| | correct | incorrect | '-' |
|---|---|---|---|
| journal | 0 | 0 | 100 |
| title | 25 | 37 | 38 |
| date | 63 | 27 | 0 |

Table 3.3: Success over 100 samples of determining fields using PDF method.

| | correct | incorret | '-' |
|---|---|---|---|
| journal | 99 | 0 | 1 |
| title | 85 | 1 | 14 |
| date | 77 | 0 | 23 |

Table 3.4: Success over 100 samples of determining fields using text extraction method.

The result of this algorithm over 100 samples is shown in Table 3.4. This shows that 87% of fields are automatically filled out for the user, significantly reducing the amount of time they have to put in when inputting large numbers of documents and therefore, freeing their time to focus on analysing the results of the classification.

#### Storing Computations

A short way through the project, I decided to output the results of computations because the computations are repeated every time the tool is started. For example, converting all 411 PDFs to text and then counting their features takes approximately ten hours. The text only changes if the code for their conversion or cleaning changes which is rare so this does not warrant being performed every time. I found that cross-validation takes eight minutes which is too long to add to the startup time when the user may not even want to do this. They may just want to see the results of the last cross-validation. This is why the last cross-validation is stored and then loaded up on startup, rather than recomputing. This is similar to retraining and testing the data which takes approximately six minutes when it will produce the same visualisation that was previously computed. This is why all values that are plotted on the graph are stored. Considering that the loading of documents takes approximately four minutes, storing cross-validation and the plotting values rather than recomputing them means startup goes from taking 18 minutes, down to 4 minutes. Increasing the speed of startup by 78%. This means the user does not have to remember to start the program an excessive length of time before they want to use it.

### 3.5.3 Codebase

#### Object Orientation

The codebase is made refactorable by my use of object orientation. The classes are all separated clearly by their functionality. Using objects makes programming on the frontend far easier as the backend can be passed in via a single object and then navigated through via its child objects, giving it structure. An example of this is how a new tab is automatically created in the frontend for every Graph object contained in DocumentList. This means that tabs do not have to be hardcoded for every visualisation. Encapsulation is also used to avoid so it is clear where data can be manipulated from.

#### Commenting

Backend classes are commented to make the function of every class and function clear to the next developer. Some are yet to be commented but will be before handover. An example of this can be seen below:

```
"""
    Generates the train and test indexes for each fold of the
    cross-validation.

    Arguments:
    split              (int)
        -- the number of folds in the cross-validation
    docsLen            (int)
        -- the total number of documents to be trained and tested
           on

    Returns:
    trainIndexesList   ([[int]])
        -- a list of lists with each list representing a different
```

```
            train and test and each integer representing the integer
            of a Document object to be trained on
    testIndexesList   ([[int]])
        -- a list of lists with each list representing a different
            train and test and each integer representing the integer
            of a Document object to be tested on
"""
```

# Chapter 4

# Conclusion

## 4.1 Chapter Overview

The following chapter evaluates the project as a whole. The final state of the project in each requirement category set out in Section 1.4, is presented followed by reflection on where the requirements were met well and how they were failed to be met. Then, recommendations on how the project can be taken forward are set out before the project is concluded with final remarks.

## 4.2 Final State

### 4.2.1 Natural Language Processing

Two models were created in this category to be compared against three aims. Both models go through the same pre-processing process. This involves the following steps:

1. Removing metadata;
2. Removing glyphs;
3. Removing new lines;
4. Removing punctuation;
5. Changing all letters to lower case;
6. Removing all integers;
7. Lemmatising words;
8. Removing stopwords

The first model classifies documents in terms of whether they cover the topic of human rights or intellectual property. It takes each unique word as a feature and calculates the value of each feature using tf-idf-cf, specified in Equation 2.11, and classifies using a support vector machine. The support vector machine assigns probabilities to each document on how likely it is to be from each topic. A score is then assigned to each document using Equation 3.1 and the documents are classified as human rights if the score is less than zero or intellectual property if the score is greater than or equal to zero. Over four-fold cross-validation using 411 journal articles, this achieves an average balanced accuracy score of 0.984 and average p-values of 0.222 and 0.157 for human rights and intellectual property documents respectively.

The second model classifies documents in terms of whether their tone suggests that the current legal environment benefits the user or the creator. It assigns each document a score based on Equation 3.2. $hrscore$ and $ipscore$ are calculated given the proportion of sentences in a document that contains at least one word from their respective keyword list in Table A.3. The evaluation was not reliable enough in this section to be presented as a genuine result.

### 4.2.2 Visualisation

These results are visualised using four different visualisations. All visualisations plot articles that originate from human rights journals in red and intellectual property journals in blue.

The first visualisation is in 3D and is shown in Figure 3.12. It plotted the date that an article was published as the x-axis, hr-ip-scores as the y-axis, and user-creator scores as the z-axis. Luminance is used to represent the distance from the surface as it is difficult to see where points are but annotations are also visible on the edge of the visualisation to make this slightly clearer.

The second visualisation is shown in Figure 3.7. It plotted the date an article was published as the x-axis and the hr-ip scores as the y-axis. The third visualisation is shown in Figure 3.16. It plotted the date an article was published as the x-axis and the user-creator scores as the y-axis. The fourth visualisation is shown in Figure 3.10. It plotted the hr-ip scores as the x-axis and the user-creator scores as the y-axis.

All these 2D visualisations have similar features. They have two trend lines to accentuate the trend in human rights and intellectual property documents respectively. This trend line is boldened if the trend is statistically significant. Also, if a point is clicked on, that point shows an annotation of the title of the document, what journal it came from, the date it was published and its filename, as shown in Figure 3.17.

### 4.2.3 Usability

There is a user interface which can be opened from a Windows desktop via an executable. The user interface consists of three tabs.

The first tab is shown in Figure 3.20 and displays the visualisations in four further tabs. The second tab is shown in Figure 3.21 and displays the evaluation data of the two models. There is also a button that causes the tool to re-cross-validate the data. The third tab is shown in Figure 3.22 and displays the documents that the model is using. This contains all the functionality relating to the documents. Document metadata can be changed and documents can be added, removed and opened. This is also where test data is selected and the user commands the tool to train and test the documents.

There is also backend functionality to make the tool more usable. Automatic metadata extraction means the user does not have to manually input an article's title, journal or date. The storage of data once computed means that it does not have to be recomputed, saving the user considerable time.

The code for the tool is organised into an object-oriented structure with common coding principles followed, making it easy to adapt. The functionality of all classes and functions are made clear by comments.

## 4.3 Requirement Evaluation

### 4.3.1 Natural Language Processing

**The model accurately represents a large proportion of the documents in the corpus with regards to its classification characteristics**

This goal was a clear success in the first model with a balanced accuracy score of 0.98. However, it was less so in the user-creator model which potentially had a balanced accuracy of 0.75 over 20 documents. In reality, this is not enough documents to assess whether the model is accurate. Because of this lack of confidence in the evaluation process, I chose the simplest model possible. The fact that negators are ignored casts doubt on whether the model accurately represents whether documents benefit the user or creator since sentences with the opposite meaning are classified in the same way. However, because I chose a simple model, it is easy to explain what the model does accurately represent. The model represents the net number of sentences that contain keywords based around the creator compared to the number of sentences that contain keywords based around the user.

Had I determined ground truths for a sizable number of documents, I would have been able to evaluate properly and develop a model with confidence. This will be where recommended future work is focused.

**The model deduces any trends in the corpus with regards to its classification characteristics if those trends exist**

For HR-IP over time, the model very rarely detected statistically significant trends. In fact, human rights and intellectual property documents appeared very polarised. Dr. Blakely stated in her final evaluation

that this supports her hypothesis because "the two fields operate in a siloed way and don't 'speak' to each other, despite having a substantive effect on the same areas (creative/cultural production)".

For User-Creator over time, the model rarely detected statistically significant trends for intellectual property articles but detected them for human rights articles towards user-benefiting language approximately half the time. Dr. Blakely stated that she did not have any expectations for these trends but that trend was not entirely unexpected. One benefit of having a very simple model is that it is very clear what the trend is indicating: there seems to be an increasing amount of user-oriented language compared to creator-oriented language in human rights articles over time.

I have confidence in the trend detection itself. The future work will be focused on improving the accuracy of the models and seeing whether trends happen to turn up.

#### The model must only consider the language of the content of the article

The idea of this aim was to make sure that classifications were correct only because of the legal language used in the article as opposed to other irrelevant elements in it, such as metadata in the footnote. For this reason, a reduced score actually suggests that this aim has been achieved. This was observed when regular expressions were used to remove metadata and brought the accuracy score down from 1 to 0.58. Other methods to meet this aim was adding words that indicated the document's structure to the stopwords list and using term frequency in the feature count because it may have alluded to the word limit of a journal.

After a thorough read of the text being classified for a few documents of different formats, I deemed that there is a negligible amount of text included that is not part of the actual content of the article. I will not recommend any further work for the aim to be met.

### 4.3.2 Visualisation

These aims were met as a whole with most future work needing to be done to the user-creator model.

#### The visualisation and its axes' meanings are self-explanatory

Dr. Blakely agreed that this aim was met. This was done using clear labels with arrows where necessary. It occurred to me after finishing the project that a better way of indicating which side of an axis represented which topic could have been done using the axes' major ticks. This would have been a cleaner way to solve this problem.

#### It is clear where a point lies on the visualisation's axes

Dr. Blakely agreed that this aim was met. This was done most effectively via the addition of the 2D visualisations which made it obvious where each point was on the axes. However, Dr. Blakely pointed out that she still wished for points on the 3D visualisation to be clearer. Unfortunately, my desired solution to this to move the axes to the middle of the visualisation was not possible and my solution to add surfaces where I wanted the axes made the visualisation too convoluted.

#### The visualisation has all the possible information that a user may require on it

Dr. Blakely agreed that this aim was met. This was done by adding annotations to the 2D visualisations that appear when a point is clicked on. However, Dr. Blakely pointed out that she wished for the same functionality on the 3D visualisation. This was a low priority for me as the annotations would have been difficult to make look presentable and the pay-off would have been low in my opinion because the functionality was already available elsewhere.

#### The visualisation clearly indicates any trends deduced by the models

Dr. Blakely agreed that this aim was met. This was done by adding lines representing the linear regression calculations to the 2D visualisations. However, Dr. Blakely again pointed out that she wished for the same functionality on the 3D visualisation.

#### The visualisation is aesthetically pleasing

Dr. Blakely agreed that this aim was met. This was done by keeping the visualisation minimalist and adding a colour scheme which applied to the labels.

#### Overall

This category of aims has been achieved well overall as Dr. Blakely agreed that each individual aim had been met. The main limiting factor that prevented her from strongly agreeing that the aims had been met was that she wanted the 3D visualisation to have the same extra functionalities as the 2D visualisations. This was due to two main faults in my process. Had I been in more regular contact with Dr. Blakely, I would have known to prioritise functionality in the 3D graph more. This was difficult because of busy schedules on both sides. It was also difficult because of my choice of matplotlib which I found restrictive in 3D. Had I done more research early on in the project, I may have been more willing to experiment with other visualisation libraries which have better 3D functionality.

### 4.3.3 Usability

#### The tool does not require any prior Computer Science knowledge to setup or use

This aim was met well. All functionality established in the requirements is available via buttons in the graphical user interface which is easily accessible on a Windows desktop. Dr. Blakely was able to use the system without this knowledge.

#### The tool has a graphical user interface which is intuitive

This aim was partially met as all features are accessible in logical places in the user interface. This is due to the following of some of Nielsen's heuristics. The design was kept minimalist meaning the user was never distracted from important features by fancy design. The system matched real-world concepts familiar to the user such as tabs to go to new sections and elevated buttons which sunk when pressed. The system was consistent with looks and actions meaning nothing unexpected. For these reasons, Dr. Blakely had no issues navigating the user interface. However, she was familiar with some of the process that was taking place before use.

A user completely new to the project should be able to pick the tool up with ease. I don't believe this is the case. A new user would be unaware of how many test documents are needed, for example, but the user interface does not come with a clear enough set of instructions or data validation which would indicate this. If not enough documents are set as test data, the tool will attempt to perform its tasks with the incorrect data and there will be an error which will not be explained by the user interface. This can be solved by someone who has knowledge of Python and natural language processing but not the new user of the tool. This and other Nielsen's heuristics will be recommended as future work.

#### The tool allows for maximum time to be spent on analysis of results

This aim was met well. The intuitive layout of the design means that the functionality is obvious and time does not have to be wasted finding out how things work. The time saved with the automatic data extraction and data storage was significant. However, Dr. Blakely suggested that future users may not want to bother downloading the large dataset and would rather a web app link that encapsulated all functionality. This is discussed in the future work section.

#### It is easy to understand the purpose of each section of code

This aim was met via the comments on classes and functions and code that was written intuitively.

#### It is easy to expand upon and adapt the overall codebase

This aim was partially met. The object-oriented structure of the code makes it well refactorable and reusable but the lack of automated testing of functionality limits the achievement in this aim. The developer needs to have confidence that the codebase all functions as claimed. My claim that I have thoroughly tested the tool manually is not enough to supply this confidence meaning that the next developer would need to do their own manual tests or write automatic tests as future work.

#### Overall

Overall, these aims were met to a reasonable standard considering that natural language processing and visualisation were given higher priority because they were more important in proving the computer science is appropriate for this application.

## 4.4 Future Work

### 4.4.1 Natural Language Processing

Most of the future work here must be focused on the user-creator model. A dataset with a few hundred articles where ground truth has been established would allow for a comprehensive evaluation of any models. This, importantly, will allow for the comparison of models. Should this be a rule-based system, experimentation can take place by changing the threshold of classification so there is a more balanced prediction between creator-benefiting and user-benefiting or the keyword lists can be extended to more comprehensively cover the classes. However, I recommend that a machine learning approach similar to that of the human rights-intellectual property model is taken up because it is known to perform better and needs less manual analysis.

Although the human rights-intellectual property model is very accurate, no consistent statistically significant trends were found. In order to be sure that no trends were missed, n-grams can be used as a type of feature over bag of words. This will give more context to the words and therefore provide a better representation of the documents. This will lead to a considerably larger and sparser feature space so I also recommend that the feature space is refined using rules based on how many documents a feature appears in.

To further test the human rights-intellectual property model, the dataset should be extended to include articles that are not as polarised as the journals used in this project. This will test the model more and may also show more interesting trends.

### 4.4.2 Visualisation

The next work to be done on visualisations is to bring the 3D visualisation up to standard with the 2D visualisations as Dr. Blakely believes one visualisation with full functionality is preferable to a range of visualisations. This most importantly includes improving the user's ability to locate points on the axes. This can be done by experimenting with more pre-attentive features such as colour maps or size. I recommend exploring other libraries that have more freedom with 3D plotting, such as 'gnuplot', in combination with these. Further, 3D visualisations can have added functionality such as annotations for when points are clicked and 3D trend lines.

### 4.4.3 Usability

The next focus on future work on this area should be on making sure the user is fully informed of what is going on in the system. There must be data validation and error messages in order for them to know what problem has occurred and what they need to solve it. Despite the user interface's intuitiveness, there should be full documentation on how to use the tool in case anything is unclear from person to person. There also should be the addition of informing the user of the system status as there is a lot of long wait times for processing and the user needs to be assured that the tool is still processing and an error has not occurred. Another useful feature that should be added is a search feature in the document tab since documents can take a while to find if there are a lot of them.

For Dr. Blakely's BILETA presentation, I prepared a PowerPoint presentation to simply explain the natural language processing techniques behind the tool. A similar guide could be added as another tab in the tool to help the user understand how the results came about.

Dr. Blakely points out that she cannot see other law academics wanting to put in the time to download large amounts of data and a desktop app. The alternative to downloading data is hosting the data online. This could vary in complexity significantly depending on Dr. Blakely's requirements for this. My recommendation would be to host a web app with shared data between law academics who could alter the dataset. This would not be too much of a jump in terms of infrastructure and would mean that no files would have to be downloaded.

The key issue for the further development of the codebase in this project is the lack of automated testing. This should be added by a future developer to give them confidence that everything works as stated and then kept up to date so developers after them will have confidence in their work.

### 4.4.4 Other Fields

Since the tool is easily refactorable, it would be easy to apply to other domains. After discussing with law academics at BILETA, I established that this could be applied to more specific variants of these fields

such as patent law and right to health care.

## 4.5 Final Remarks

The project's aims have been met to a good standard given the time frame. The project has met Dr. Blakley's aim showing that natural language processing can be usefully applied to intellectual property and human rights by tailoring skills from natural language processing, data visualisation and usability to suit the domain and then synthesising them into a tool. This was acknowledged at BILETA and the results and future work were enthusiastically discussed by the audience.

# References


[1] Lionel Bently, Bred Sherman, *Intellectual Property Law*, 4th ed. OUP Oxford, 2014, ISBN: 978-0199645558.

[2] Christophe Geiger, *Research Handbook on Human Rights and Intellectual Property*. Edward Elgar Publishing, 2016, ISBN: 978-1786433411.

[3] Victoria A. Grzelak, "Mickey Mouse & Sonny Bono Go To Court: The Copyright Term Extension Act and Its Effect On Current and Future Rights," *John Marshall Review of Intellectual Property Law*, vol. 2, no. 1, pp. 95–115, 2002.

[4] Tom Bell, *Trend of Maximum U.S. General Copyright Term*, tomwbell.com, CC-BY-SA 3.0, 2008.

[5] Laurence R. Helfer, Graeme W. Austin, *Mapping the Interface Between Human Rights and Intellectual Property*. Cambridge University Press, 2011, ISBN: 978-0521711258.

[6] Megan R. Blakely, *BILETA 2019 Abstract*, Not published - available upon request, 2019.

[7] Laurence R. Helfer, "Human rights and intellectual property: Conflict or coexistence?" *Minnesota Intellectual Property Review*, vol. 5, no. 1, pp. 47–62, 2003.

[8] Doug Eboch, *Some Thoughts About Tone*, letsschmooze.blogspot.com, 2015.

[9] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *International Joint Conference on Artificial Intelligence*, vol. 14, 1995, pp. 1137–1145.

[10] Digna R. Velez, Bll C. WHite, Alison A. Motsinger et al, "A Balanced Accuracy Function for Epistasis Modelling in Imabalanced Datasets Using Multifactor Dimensionality Reduction," *Genetic Epidemiology*, vol. 31, no. 4, pp. 306–315, 2007.

[11] Febrizio Sebastiani, "Machine learning in automated text categorization," *ACM Computing Surveys*, vol. 34, no. 1, pp. 1–47, 2002.

[12] David M. W. Powers, "Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness & Correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[13] Thorsten Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European Conference on Machine Learning*, 1998.

[14] Sida Wang, Christopher D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Annual Meeting of the Association of Computational Linguistics*, ser. 50, vol. 2, 2012, pp. 90–94.

[15] Shantanu Godbole, Sunita Sarawagi, "Discriminative methods for multi-labeled classification," in *Advances in Knowledge Discovery and Data Mining*, ser. 8, vol. 1, 2004, pp. 22–30.

[16] Rodrigo Moares, Joao F. Valiati, Wilsion P.G. Neto, "Document-level sentiment classification: An empirical comparison between svm and ann," *Expert Systems with Applications*, vol. 40, no. 2, pp. 621–633, 2013.

[17] Bo Pang, Lillian Lee, Shivakumar Vaithyanathan, "Thumbs up?: Sentiment classification using machine learning techniques," in *Empirical Methods in Natural Language Processing*, vol. 10, 2002, pp. 79–86.

[18] Santanu Mandal, Sumit Gupta, "A lexicon-based text classification model to analyse and predict sentiments from online reviews," in *International Conference on Computer, Electrical & Communication Engineering*, 2016.

[19] Hannah M. Wallach, "Topic Modelling: Beyond Bag-of-Words," in *Journal for Language Technology and Computational Linguistics*, vol. 23, 2006, pp. 977–984.

[20] Chade-Meng Tan, Yuan-Fang Wang, Chan-Do Lee, "The use of bigrams to enhance text categorisation," *Information Processing & Management*, vol. 38, no. 4, pp. 529–546, 2002.

[21] Man Lan, Sam-Yuan Sung, Hwee-Boon Low, Chew-Lim Tan, "A Comparative Study on Term Weighting Schemes For Text Categorization," in *IEEE International Joint Conference on Neural Networks*, 2005, pp. 546–551.

[22] Mingyoung, Jiangag Yang, "An Improvement of TFIDF Weighting in Text Categorization," in *International Proceedings of Computer Science and Information Technology*, vol. 1, 2012, pp. 44–47.

[23] George A.F. Seber, Alan J. Lee, *Linear Regression Analysis*. John Wiley & Sons, 2012, ISBN: 978-0471415404.

[24] Harvey J. Motulsky, Lennart A. Ransnas, "Fitting Curves to Data Using Nonlinear Regression: A Practical and Nonmathematical Review," *The Federation of American Societies for Experimental Biology Journal*, vol. 1, no. 5, 1987.

[25] Peter J. Huber, "Robust Regression: Asymptotics, Conjenctures and Monte Carlo," *The Annals of Statistics*, vol. 1, no. 5, pp. 799–821, 1973.

[26] Ronald A. Fisher, *Statistical Methods for Research*, 14th ed. Oliver & Boyd, 1970, ISBN: 978-0050021705.

[27] Ronald L. Wasserstein, Nicole A. Lazar, "The ASA's Statement on p-values, Conext, Process, and Purpose," *The American Statistician*, vol. 70, no. 2, pp. 129–133, 2016.

[28] Daniel J. Benjamin et al, "Redefine Statistical Significance," *Nature Human Behaviour*, vol. 2, pp. 6–10, 2018.

[29] Daniel D. Boos, Leonard A. Stefanski, "P-value precision and preproducibility," *The American Statistician*, vol. 65, no. 4, pp. 213–221, 2011.

[30] Michal Toman, Roman Tesar, Karel Jezek, "Influence of Word Normalization on Text Classification," in *The Association for Information Science and Technology*, 2006.

[31] Alper K. Uysal, Serkan Gunal, "The Impact of Preprocessing on Text Classification," *Information Processing and Management*, vol. 50, pp. 104–112, 2012.

[32] Elaine I. Allen, Christopher A. Seaman, "Likert scales and data analyses," *Quality Progress*, vol. 40, no. 7, pp. 64–65, 2007.

[33] Stephen G. Kobourov, Tamara Mchedlidze, Laura Vonessen, "Gestalt Principles in Graph Drawing," in *Graph Drawing and Network Visualization*, vol. 9411, 2015, pp. 558–560.

[34] Christopher G. Healey, James T. Enns, "Attention and Visual Memory in Visualization and Computer Graphics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 7, 2012.

[35] Stephen M. Kosslyn, *Graph Design for the Eye and Mind*. OUP USA, 2006, ISBN: 978-0195311846.

[36] Stephen J. Watson, Daniel J. Zizzo, Piers Fleming, "Determinants and welfare implications of unlawful file sharing: A scoping review," CREATe, Tech. Rep.

[37] Jakob Nielsen, "Enhancing the Explanatory Power of Usability Heuristics," in *SIGCHI Conference on Human Factors in Computing Systems*, vol. 1, 1994, pp. 152–158.

[38] Diomidis Spinellis, "Code documentation," *IEEE Software*, vol. 27, no. 4, pp. 18–19, 2010.

[39] Robert C. Martin, *Clean Code: A Handbook of Agile Software Craftsmanship*, 1st ed. Prentice Hall, 2008, ISBN: 978-0132350884.

[40] Guido van Rossum et al, *Index of Python Enhancement Proposals*, python.org/dev/peps, 2019.

# Appendix A

# Ground Truths

| name | class |
|---|---|
| The International Journal of Cultural Property | Human Rights |
| The International Journal of Heritage Studies | Human Rights |
| The Journal of Intellectual Property Law | Intellectual Property |
| The Journal of World Intellectual Property | Intellectual Property |

Table A.1: A list of journals for each class supplied by Dr. Megan Rae Blakely.

| name | year of publication | class |
|---|---|---|
| Basic Texts of the 1972 World Heritage Convention | 1972 | Human Rights |
| Convention for the Safeguaring of the Intangible Cultural Heritage | 2003 | Human Rights |
| The Agreement on Trade-Related Aspects of Intellectual Property Rights (TRIPS) | 1995 | Intellectual Property |
| Berne Convention for the Protection of Literary and Artistic Works | 1979 | Intellectual Property |

Table A.2: A list of journals for each class supplied by Dr. Megan Rae Blakely.

| human rights | intellectual property | user | creator |
|---|---|---|---|
| heitage | protection | access | right |
| cultural | property | open | protection |
| intangible | author | right | control |
| safeguarding | licence | progress | exclusive |
| natural | intellectual | compulsory | |
| protection | trademark | licence | licence |
| international | authority | free | royalty |
| property | patent | collaborative | creative |
| educational | monopoly | reuse | control |
| conservation | scientific | acknowledgement | exclusionary |
| universal | artistic | credit | limited |
| identity | literary | noncommercial | author |
| generation | intangible | fan fiction | individual |
| inherent | idea | adaptation | literary |
| generation | expression | parody | artistic |
| transmission | discovery | research | invention |
| | copyright | exception | economic |
| | useful | limitation | moral |
| | derivative | education | compensation |
| | transformative | personal | incentive |
| | limited | | |
| | infringe | | |
| | fixed | | |
| | incentive | | |

Table A.3: A list of keyword for each class supplied by Dr. Megan Rae Blakely.