# Predicting Citations from Legal Texts: An Application of Artificial Intelligence in Law

by

Christopher Mazzuca

Supervisor: Anthony Niblett
April 2020

**B.A.Sc. Thesis**

Division of Engineering Science
UNIVERSITY OF TORONTO

University of Toronto

# Predicting Citations from Legal Texts: An Application of Artificial Intelligence in Law

Christopher Mazzuca (1003150181)

Prof. Anthony Niblett

Undergraduate Thesis (ESC499)

Thursday, April 9th, 2020

**Abstract.** *This thesis seeks to show that new techniques in Natural Language Processing and Artificial Intelligence are capable of learning the language of law.*

*The central task of the research involves predicting legal citations solely from the text of a decision. Citations from the Supreme Court of Canada are predicted from the decisions of the Canadian Federal Court of Appeal.*

*This thesis proposes a new learning-based method that outperforms traditional static approaches used in the field. Much improvements can be made, but the proposed model is capable of learning the language of law and appropriately predicting relevant citations. The results from this research lay the groundwork for future endeavors in legal research, as the results show that the proposed methods can the language of law.*

**Keywords.** Citations, Law, Artificial Intelligence, Predictions.

# Acknowledgement

I want to thank my supervisor, Professor Anthony Niblett, for guiding me throughout this project, and for inspiring me to push the boundaries of exploration. Without his passion for the project, the goal of this project would not have been reached.

I want to sincerely thank the entire CanLII organization for all their help and provisioning of data. The project would not have been possible without their help.

Finally, I want to acknowledge my friends and family, specifically Rose, Serge, Matthew and Melissa. You supported me throughout the project and provided me with new perspectives as the project unfolded. I am forever grateful for your endless support, inspiration and generosity.

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

## 1.1 Background

When approaching a new case, it is vital for lawyers to conduct research to examine the legal landscape pertaining to that case, regardless of the lawyer's area of expertise or practice. The law is heavily based on precedents, or distinguished past cases within an area of law, and lawyers will look for these leading cases when conducting legal research.

Analyzing the outcomes of past precedents allows lawyers to examine how specific rules and standards were previously acted upon, enabling them to prepare for and make predictions about their current case. If a past case is used in the development of a decision, it will be cited in the final decision document of that case. By using citations, the law builds upon the legislation serving as the foundation for the legal system.

Since the practice is constantly evolving, lawyers must educate themselves on the current state of law. This implies, in an ideal world, reading through every relevant past decision to see how past decisions were made. However, it is extremely difficult if not impossible for humans to efficiently read through hundreds of relevant legal cases across many legal landscapes and to reasonably extract information from them. Improving the research process even marginally would mean significant savings in time for lawyers, thereby improving effectiveness and efficiency.

## 1.2 Research Overview

The objective of this research is to tackle the problem of retrieving relevant decisions for legal research, using only textual information. The intention is to design a method of prediction that requires no human input, so that any legal professional from any specialty within law can perform effective and efficient legal research in any area of law. By the success of these predictions, it could be argued that Artificial Intelligence (AI) is capable of understanding or learning the language of law. This result can be applied to many tasks not only limited to citation predictions.

The decisions from two Canadian courts were collected. The courts were chosen specifically so that the decisions from one of the courts often cites the other, which enabled the citation dataset to be less sparse. The Federal Court of Appeal (FCA) and the Supreme Court of Canada (SCC)

were chosen strategically because the FCA frequently cites cases from the SCC, and because data collection was possible from these courts. Once the decisions were downloaded, several preprocessing measures were taken in order to eliminate bias (i.e. removing citations and quotes from the FCA corpus), along with generating a truth table that housed all of the citation information. These processes are thoroughly described in the Methods section.

Upon completion of the data collection and preprocessing, an investigation of the dataset was conducted. It was discovered that within the FCA corpus a correlation existed between the usage of citations and the use of text in the decisions.

A baseline predictive model was built, using traditional Natural Language Processing (NLP) comparison methods. This static model was compared to the deep learning models developed, which were trained on learned Doc2Vec embeddings of the FCA and SCC corpora. The deep learning models outperformed the baseline model significantly for the top 1 and 5 predictions, an important result in this research. The baseline achieved an F1 score of 0.164 for the top 5 predictions, while the final deep learning model achieved 0.632 on the training set and 0.217 on the test set. This is a 32% improvement from the test set relative to the baseline. A deeper investigation also showed that the deep learning models are interpreting intricacies of the language used, a positive finding of this research.

The proposed models have limitations. There is a notable difference between training and test results, which can signify memorization of the data or problems with the collected data. Upon further review, it appears that the data is limiting the generalization ability of the models. This could be improved if more data is collected or through data augmentation. Nonetheless, the proposed models significantly outperform the baseline model on the test data, a static model that does not consider language context and is static. This result illustrates that the proposed AI architecture can learn the language of law, the initial hypothesis of the research.

## 1.3 Current Landscape

Many challenges impede this research and future work in this field, including learning the intricacies of the English language and its syntax, and understanding its use within legal documents. Recent advances in NLP and AI across different fields make a compelling argument for the case that AI will one day be capable of “understanding” the language of the law and the

structure of legal arguments [1]. The field has recently begun to use the newly developed mechanisms in NLP, as outlined in the literature review. However, there still exists a significant gap between the use of NLP within textual analysis in law, and the predictions of relevant legal research or citations. Bridging this gap between the field of NLP, textual analysis and citation analysis in law is the primary focus of this research project. The success of this project will lay the foundation for future endeavors between AI and law.

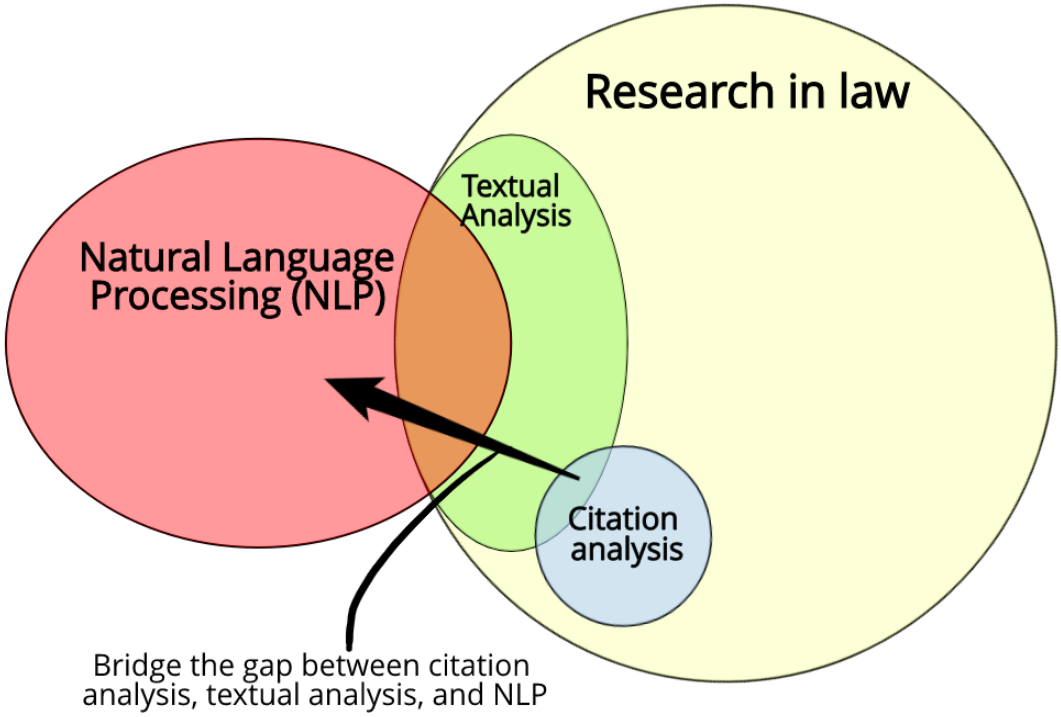
In the industry, companies like Blue J Legal [2], Westlaw [3], ROSS Intelligence [4], Bloomberg Law [5], and LexisNexis [6] are focusing on predicting relevant cases for research purposes. However, they require manual inputs to make predictions. They assume that the user has some knowledge of the case, and more importantly has knowledge of that particular field of law; enough to be able to enter an appropriate query. This paper, on the other hand, proposes that a dependence on user input is not required in order to make reasonable predictions, and it may even eliminate the bias inherent in current citation use (see literature review).

## 2 Literature Review

The landscape of research in law is quite vast. Significant work has been done to investigate the way in which language is used within legal texts, producing some sort of empirical and quantitative results. Another portion of research within law, although much less substantive, involves the investigation and use of citations, also referred to as the field of citation analysis. Nonetheless, these areas are still small in comparison to the whole research umbrella.

Upon a literature review, it appears that textual analysis within law has recently started to employ Natural Language Processing (NLP) and Artificial Intelligence (AI) learning methods. In addition, researchers are introducing supplemental information (i.e. citations) to aid the training of the text-based learning methods. Yet, there still exists a significant gap between the use of NLP in other fields and its use in law (as displayed in Figure 1 below). The purpose of this work is to take a significant step forwards in the integration of NLP with law, and to outline a roadmap in which future work can be developed.

*Figure 1: Representation of the existing gap between textual analysis in law, citation analysis, and NLP.*




## 2.1 Natural Language Processing (NLP)

The term Natural Language Processing (NLP) refers to a very broad array of linguistic studies, typically but not limited to the study of the English language [7]. Each area within NLP is concerned with extracting information from text or speech in some capacity, and in some cases making predictions based on the information extracted. This section will outline some of the many applications of NLP research, how NLP is used to represent text on computers, and how these representations are used to extract information and group similar documents.

### 2.1.1 The Wide-Reaching Applications of NLP

A significant portion of NLP research is concerned with syntax. Some examples include: *part-of-speech tagging* (determining the part of speech for each word in a sentence); *stemming* and *lemmatization* (breaking down words into their roots); and *parsing* (analyzing the grammar within each sentence). Another major component of NLP focuses on the semantics of languages. Some tasks within this include: *sentiment analysis* (examining the polarity of text); *topic*

*segmentation* (identifying the dominant topics within a piece of writing); and *distributional semantics* (learning semantic representations from text). A third component to NLP research is devoted to the study of discourse. Tasks include: *summarizing text* (reading through a piece of writing and highlighting important sentences); *coreference resolution* (determining commonalities between what texts mention); and *general discourse analysis* (looking at the relationships between sentences).

These are only some of the tasks currently being studied within NLP. This research is very extensive, and its applications reach many different fields. For instance, Bharadwaj and Shao used the text from 6,256 articles to classify whether or not that news was fake. They compared three different models (a recurrent neural network, a naïve bayes classifier, and a random forest classifier), and their best model was able to achieve an accuracy of 95.66% using only text [8]. Another interesting application of NLP is its use in predicting future stock prices. Mehtab and Sen developed a technique that exploited long short-term memory (LSTM) models to make predictions, in combination with a sentiment analysis of Twitter [9]. They found that the traditional predictive models performed better in combination with the NLP Twitter sentiment analysis. These are only some of the many different applications of NLP research, and it looks as though there will be many more in the future.

### 2.1.2 Representations of Text

The main obstacle with textual analysis is representing text appropriately. Many options currently exist, each with their own advantages and disadvantages. Textual representation is heavily application dependent, and this section will primarily focus on the leading methods used when making textual comparisons or analyzing textual similarities as this research project sets out to achieve.

A portion of textual representations fall under the umbrella of what is referred to as "bag-of-words" approaches. This refers to representations that do not consider the context in which words are used within a document, but only the frequency at which they are used [10]. In this way, all of the words collected from a document are organized into a vector, with each entry representing the *term frequency* (tf). For instance, the sentence "The quick brown fox jumps over the lazy dog" will have the tf vector shown in Figure 2.

*Figure 2: Example of a term frequency (tf) vector for the sentence "The quick brown fox jumps over the lazy dog".*

| **the** | **quick** | **brown** | **fox** | **jumps** | **over** | **lazy** | **dog** |
|---|---|---|---|---|---|---|---|
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

This is the simplest way to represent text, and is prone to many errors since it is ignorant to the context of words. For instance, these vectors would have higher frequencies of common words like "the", "as", and "it". These words are referred to as "stop words", and often introduce noise within textual analyses.

To work around this, the concept of *inverse document frequency* (idf) is used. Across a corpus of documents, each term is searched and counted, and their idf score is the inverse of this total. For words that are common within the corpus (i.e. stop words), the idf is really low, and words that are very rare will be higher. Typically, the two measures are combined logarithmically to produce a more complete form for each document in a corpus, called *term frequency inverse document frequency* (tf-idf). Although there are more advanced and effective techniques as described below, Shahmirzadi et al. have found that tf-idf can perform well under certain conditions [11].

There are newer approaches to modelling text, aside from the traditional methods discussed above. They are known as "word embeddings", and are learning-based approaches to the problem of representing text discussed above [12]. The advantage of word embeddings is that they are dense, and have been shown to be more effective than traditional methods. There are a variety of ways to generate these embeddings, including neural networks, probabilistic models, and dimensionality reduction techniques.

Word2Vec is a commonly used method that uses a two-layer feedforward linear neural network to compute word embeddings. This method was proposed by Mikolov et al. in 2013 [13]. FastText is an open source project that has pretrained models for English representations that is frequently used [14]. ELMo is a recently published technique (2017) developed by Peters et al. that focuses on the modelling the complex use of words, and how these uses vary across linguistic contexts [15]. GloVe is an unsupervised learning method developed by Pennington et al., and is also considered a staple within the field [16]. Grzegorczyk provides a more detailed summary of all of these techniques, along with a few more recent methods [17].

In short, there are a multitude of methods that are currently being employed to learn word embeddings within the field. An example encountered in this literature review applies the Word2Vec learning approach to 10-K financial filings. Using 10-K filings collected from the S.E.C. between 1993 and 2018, Sehrawat was able to learn word embeddings particular to that application [18]. Similarly, Ali et al. decided to learn word embeddings for the Sindhi language, since there had been no previous work done on the language. They showed that it was possible to learn word embeddings even on an entirely different language, using the GloVe, Skip-Gram (SG), and Word2Vec algorithms [19]. These examples show us that it is possible to successfully use and implement the word embedding algorithms discussed above.

Unfortunately, it is not clear which method performs better. Each method's performance is highly dependent on the corpus and application. In this research project, each of the methods above will be consulted and tested in different capacities.

### 2.1.3 Text Similarity using Word Embeddings

As described in the previous section, it is possible to compare documents within large corpora and group documents by similarity. This is an important task across many different fields, and there are many examples of its use across different fields.

In a recent publication, Steier used NLP to analyze various aspects of Dante Alighieri's famous work "*La Divina Commedia*", or "*The Divine Comedy*". Particularly, Steier applied textual similarity to compare "The Divine Comedy" to a variety of Shakespeare's plays, and found that the section "*Paradise*" was similar in text to Shakespeare's sonnets [20]. This example is interesting because it highlights the wide-reaching applications of these methods. In another interesting application, Saedi and Dras used similarity-based approaches to identify the author of blog posts. They used *Siamese networks*, a complex set of Convolutional Neural Networks (CNNs) along with a new word embedding approach called BERT, and confirmed that their system was able to learn a general notion of authorship simply by analyzing text [21]. Another recent example displays how this research is being applied in the medical industry. Dobrakowski et al. approached the question: "Is it true that patients with similar conditions get similar diagnoses?". By using GloVe to compute word embeddings on a corpus of about 100,000 clinical records, they were able to group and visualize terms based on the embeddings generated [22]. They also were

able to find what were the top-5 most recommended treatments by the doctors, for a certain subset of medical disciplines.

The examples discussed above show how word embeddings are being employed to measure textual similarity and extract insights in many fields. Many fields across many different domains have used these tools successfully, and highlight that when used properly, these successes can propagate to the specialty of law.

## 2.2 Textual Analysis in Law

This section will briefly discuss the history of textual analysis in legal research, and examine the increased use of NLP to encode documents, which has become a recent trend in the research.

### 2.2.1 Empirical Analyses of Legal Texts

In the infancy of textual analysis in law, researchers were primarily focused on investigating how language was used. In 1980, Danet produced "*Language in the Legal Process*" that empirically analyzed argumentation used across many cases, and reasoned that language is used in many ways, including as an argumentation style (a concept referred to as "thickening") [23]. Other publications performed similar empirical analyses of legal text, such as Goodrich's "*Law and Language: An Historical and Critical Introduction*" in 1984 [24], and Fairclough's "*Discourse and Text: Linguistic and Intertextual Analysis within Discourse Analysis*" in 1992 [25]. These were some of the early investigations into textual analysis within law.

More recently, the research conducted on legal texts has pivoted to use statistical methods to make inferences about the language within a dataset. In 2014, Fagan applied a naïve Bayes classifier to analyze legal texts, and predict if they were relevant or not. This study used 2,111 opinions, and the correct labels for each of these were generated manually by humans. With a standard Porter stemmer, removing stop words, using two-word phrases ("bigrams") and ten-fold cross validation, the classifier achieved an accuracy of 78.38% [26]. Using the bigrams, Fagan could visualize what words associated well with different labels (that were generated manually). Similarly, Macey and Mitts used co-occurrence to analyze the similarities of documents with bigrams, and used a naïve Bayes classifier to show how public policy is systematically piercing the "*corporate veil*" [27]. In this study, they analyzed the bigrams that were most associated with

each category (i.e. “Bankruptcy values”, “Undercapitalization”), and were able to explain how judicial opinions were formed from the bigrams. In 2014, Kosnik used a regression model (a “bag-of-words” approach, like the examples described above) to examine contract completeness. Specifically, hydroelectric licenses were examined from 1997-2007, and found that the word choice did not vary much over time [28]. Again, this example demonstrates how textual analysis (at the time) was mainly used to support empirical studies. As Kosnik wrote, “textual analysis in the law and economics literature is still very much in its infancy”. Often, however, statistical methods do not generalize well to new data, and the methods described above required manually generated inputs. It was not until a few years later that more advanced NLP methods infiltrated the textual analysis space in law, which will be discussed in the following section.

### 2.2.2 Using NLP for Legal Textual Analysis

Over the last few years, NLP has been more readily implemented in legal textual analysis. In 2016, Nay proposed a completely new method for encoding text with the specific application of legal texts, building upon some of the methods described in the previous section. The major change within this encoding allows for a more rigorous legal analysis, specifically because Gov2vec is able to discern meaningful differences between government branches (i.e. different Presidents or Congresses) [29]. This may have pretty important repercussions for this research project, because it allows the researcher to investigate questions like “How does Obama differ in addressing climate change, and how does it differ from leading environmental perspectives?”. In 2018, Ash and Chen applied document embeddings to law, with the specific goal of understanding judicial reasoning and how judges are related. Specifically, they use the Doc2Vec embedding, along with citations that inform their analysis, a method proposed by Le and Mikolov in 2014 [30]. The extension of NLP and citation analysis in this context is directly related to the task at hand. In this study, they found that “as with word embeddings, cases that tend to be cited together locate near each other in the embedding space” [31]. In other words, they found that word embeddings and citation similarities are positively correlated, a finding that strengthens the central hypothesis of the research project.

In a recent publication from 2018, Bommarito et al. describe an open source project dedicated to making NLP more accessible to legal texts. Specifically, the package can segment documents, identify key text (i.e. titles, headings), extract structured information, transform text

to features and word encodings, and build supervised and unsupervised models [32]. These examples show how textual analysis research has evolved over the last few years, and where the field is trending in the future. However, there is still much work to do, and the gap is still significant. As Robaldo et al. note, "recent research has highlighted the need to create a bridge between conceptual questions, such as the role of legal interpretation in mining and reasoning, as well as computational and engineering challenges, such as the handling of big legal data" [33].

## 2.3 Citation Analysis in Law

In Canada, along with many other legal landscapes, the legal system is based on a combination of legislation (or "civil law") and precedent (or "case law") [34]. Civil law describes a set of rules and legislations that focus on general principles. On the other hand, case law describes rulings that deal with the intricacies and peculiarities of different legislations. However, case law is not explicitly written as a rule or legislation. Rather, it describes the way in which past judicial decisions are used to set precedents for future decisions. Judges cite past cases that help inform a new decision, and in this way, the law evolves over time.

A small subset of legal research is devoted to the study of these citations, also known as the field of "citation analysis". This field draws directly from the concept of case law, and how the law builds upon itself through the use of citations to support or dispute an argument. Citation analysis has existed since the 1990s, and has evolved over time. For a while, citation analysis was mainly used to gather empirical results and to test central hypotheses, as Posner describes [35].

In the 2000s and afterwards, a majority of the work involving citation analysis included some form of statistical analysis to extract different quantitative insights. Clark & Lauderdale asserted that citations provide a useful source of information about an opinion's doctrinal location. To test this, they trained a standard Bayesian ideal point estimator to locate opinions across the citation space, and Markov chain Monte Carlo methods to calculate the posterior distributions [36]. The statistical nature of the point estimator lead to some fundamental flaws, particularly since the calculated prior did not generalize well to other citations. Ződi holistically analyzed the citation patterns from a corpus of 61,512 Hungarian decisions. Using a basic statistical analysis of the citations used in the corpus, Ződi found that different branches of

Hungarian courts cited cases in distinguishable patterns [37]. In addition, he found that with a simple statistical approach, precedent cases could be found without actually reading the documents. Similar statistical analyses have been applied to the study of judicial biases within citations, such as with Choi & Gulati. By analyzing a dataset over the course of a year, they were able to support three central hypotheses surrounding bias in judicial citations: judges cite from the same political party; judges are more likely to be biased in stressful or high stakes situations; and judges are more inclined to cite judges who also cite them [38]. Using basic statistics and analytical approaches, these studies were able to extract meaningful empirical and qualitative results. However, statistical methods are severely limited in their ability to adapt, and their performance is highly dependent on the corpus gathered.

Network analysis also comprises a large portion of the methods within citation analysis. Fundamentally, a corpus of cases can be abstracted as a network of citations, where cases are the edges and citations are the nodes. This metaphor is very useful when analyzing citations. By applying basic concepts of network science to citations, Smith found that within a corpus of over 5 million U.S. Supreme Court cases, only about 0.3% of them were cited 500 or more times [39]. In the paper, Smith asserted that due to the findings, it can appropriately be concluded that precedential authority relies within a very small portion of the case space. In other words, there are a few cases that are cited frequently, which is directly tied to the concept of case law. This behavior is typically seen in other networks (such as the Web or social networks), and may be important to note while experimenting with citations. Similarly, Derlén & Lindholm applied HubRank and PageRank (two well-known network science algorithms) to the study of centrality within a network of 8,891 CJEU (Court of Justice of the European Union) decisions. They found that the PageRank algorithm generally performed best for the purpose of measuring "authority score", but it is not clear when measured against other metrics [40]. A similar analysis was applied to investigate whether or not U.S. Supreme Court cases influenced U.S. Supreme Court IP (Intellectual Property) cases. Statistical methods were applied to view holistic trends, but Smith looked at the in and out degree for each case in the collected network, along with the "authority score" of each case [41]. There are many insights that can be gathered from this sort of analysis, which is why a considerable portion of research into legal citations involves some sort of network analysis.

There seems to be a recent shift in the use and analysis of citations in legal research. Over the last few years, citation analysis has started being employed primarily as a supplemental tool in other research. For instance, in 2001 Conrad and Dabney used many of the concepts described in this research project to generate citation predictions. However, they employ a rule-based logistic regression ranking model, of which one-third of the rules were manually generated from editors [42]. This important publication laid the groundwork for future research to come. In a similar manner, Livermore et al. have used a combination of citation analysis and manually-derived relevance scores to measure the efficacy of different algorithms in law search. Using 9,575 U.S. Supreme Court decisions, they used citation similarity scores and the manually-derived scores to test two different search algorithms and a reinforcement learning model [43]. They found that the reinforcement learning model performed the best using the citation similarity metrics and the manually-derived scores made by lawyers. However, Livermore et al. use a basic textual analysis techniques that takes only term frequencies into account. They concede that law search has been "left almost entirely undiscussed" and present several ways to improve their methods. In a recent work produced in 2019, Medvedeva et al. used dynamic machine learning methods to classify decisions from the European Court of Human Rights. Interestingly, they incorporated information about the judges behind each decision and the citations used into their models, and this produced the best results [44]. They, like Livermore et al., used a "bag-of-words" analysis to compare texts, and agree that more advanced Natural Language Processing methods – that consider the position and context of words – may yield better results.

One central theme is consistent in these new research studies; there exists a clear gap within the areas of legal research between textual analysis and citation analysis. As discussed above, recent research has shown that the marriage of artificial intelligence and machine learning with law and citation analysis can be successful. In fact, this freedom of human input may help produce predictions that are devoid of the inherent bias behind citation use uncovered in the literature review [37] [38] [39].

# 3 Methods

## 3.1 Data Collection and Cleaning

A significant portion of this project dealt with data collection and preprocessing. The intention of the section is to thoroughly explain each method used in the collection and preprocessing of the data, along with the reasoning behind why each court was chosen.

### 3.1.1 Court Selection

The task of predicting citations using text was quite complicated, and meant that several pieces of information had to be collected. Two courts had to be selected strategically so that one frequently cited the other (to avoid a sparse prediction space), and the decisions from each court had to be downloaded. Two Canadian courts that fit the criteria were the Federal Court of Appeal (FCA) and the Supreme Court of Canada (SCC), where the FCA cites cases from the SCC frequently and both datasets are available online. Once the textual documents were collected, citation information had to be collected as well.

### 3.1.2 Collecting Decision Data

Permission to download the data for the sole purpose of education and research falls under the Reproduction of Federal Law Order passed in 1997, so long as it is not redistributed and altered in any way [45]. Under this direction, data was manually downloaded from the FCA website as pdfs.

Fortunately, a partnership was made with the Canadian Legal Information Institute (CanLII), a private company that securely houses decisions from a wide range of Canadian courts. Under this partnership, CanLII provided all SCC decisions, as well as access to contextual information (i.e. citation information, case numbers, etc.) through an API key. Python code was written to collect case titles, citation id's, and a list of all citations to and from the SCC using the API key.

The partnership with CanLII was made under the approval from the SCC. Throughout the research project, the data was securely stored remotely, and will be deleted once the project is terminated. The same follows with the FCA data, as many precautionary steps were taken throughout the project to maintain the security of this sensitive data.

### 3.1.3 Preprocessing the Decision Data

The FCA decision corpus was downloaded in pdf format, while the SCC decision corpus was downloaded in HTML format. *Tika*, a Python library designed to extract text from many different file formats [46], was used to extract the text from each file within each court. Once the text was extracted, text files were saved (with the .txt file format). It became much easier to perform textual analysis on the downloaded decisions once they were converted to this format, and it became much easier to clean and preprocess the text of each decision as well. It is important to note that these libraries are not perfect, and could introduce external error in the analysis. However, this is a known limitation, as the original text formats could not be used.

The next significant phase of preprocessing dealt with eliminating all potential bias from the textual information of each court. This step is crucial, since analyzing the predictive power of the methods used could be directly affected by inherent bias within the dataset. Bias in this context implies anything that could be interpreted by a model that would improve performance or predictive power, which would ultimately detract from the results and conclusions reached. For instance, one direct source of bias could result from leaving citations in the texts. Depending on the method used for predictions, it is feasible that something could learn to make predictions by searching for the citations within the document. The objective of this project is to test whether or not AI can learn the language of the law and make predictions, so eliminating these citations was necessary in terms of maintaining the validity of the results.

Aside from removing citations, many other considerations were involved in the text preprocessing. This includes:

1. removing case names and titles (i.e. *Canada v. Canada*),
2. removing all numbers (including dates, page numbers, paragraph numbers),
3. removing all special characters (specifically converting everything to ASCII since some of the names were in French),
4. removing the words “SCC” and “SCR” (which refers to the Supreme Court of Canada) or any variation of these words such as “S.C.C.” or “S.C.R.”,

5. removing all punctuation (task dependent[1]), and
6. removing all quotes[2].

These are aggressive measures, but they were taken so that the results truly reflect what can be gained from different analyses. It is important to note that the citations and quotes were only removed from the FCA corpus but not from the SCC corpus, since it may have had an adverse effect by removing too much information. This could have introduced some bias, a limitation of the data preprocessing methods used. It could also be argued that too much information was removed from the texts (like punctuation and paragraph information), and these are known limitations of the preprocessing methods used.

### 3.1.4 Citation Collection

Another significant portion of the data preprocessing pipeline involved extracting the citation information, and reformatting it into a workable form. Fortunately, the API provided by CanLII allowed for a relatively easy downloading of all information for citations from and into SCC decisions. A Python script was written to make API calls as necessary, and download all of the citations to each SCC case. Using this information, all of the downloaded FCA cases were searched for, and this citation information was stored in a truth table.

The truth table was constructed with the FCA decisions as the rows, and the SCC decisions as columns. Each element $e_{i,j}$ (in row $i$ and column $j$) is represented as a 1 if case $i$ cited case $j$, and represented as a 0 otherwise. Figure 3 below is a representation of how the truth tables were constructed.

[1] Removing punctuation is necessary for some tasks in this research, but not all. In tasks where words sentence structure was important (i.e. training Doc2Vec embeddings for the dataset), sentence and paragraph structure was maintained.
[2] It was found later in the project that removing quotes had no effect on results, and these were added back into the texts.

*Figure 3: Fictional representation of what the truth table looks like (rows are cases from the downloaded FCA corpus, and the columns are various SCC citations in this example).*

| | **2009scc23** | **2018scc10** | **2006scc05** | **…** | **1980scc03** |
|---|---|---|---|---|---|
| **2013fca02** | 1 | 0 | 1 | | 1 |
| **2015fca104** | 0 | 0 | 0 | | 1 |
| **2019fca09** | 0 | 1 | 0 | | 0 |
| | | | • | | |
| | | | • | | |
| | | | • | | |
| **2018fca83** | 1 | 0 | 1 | | 1 |

The truth table was a convenient way to store the data because it offered a simple lookup for predictions. As the data needs changed throughout the project, several truth tables were constructed. These were all developed using the methods described here.

## 3.2 Measuring Citation Similarity

For a few different subtasks in the project, investigating the similarity of how cases cite was desired. Particularly, this was measured using a metric known as "cosine distance". Cosine distance considers each case as a vector in the citation space, and effectively finds the case (vector) with the smallest cosine distance, or most similar angle in the n-dimensional space. A mathematical representation of this metric is shown below[3]:

$$similarity = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

[3] A and B are vectors, which in this context represent the citation information from decisions in the corpus.

By calculating the scores for each possible pair of cases, the most similar decisions can be found for a given case. In fact, this method was also used in the baseline predictive model to gather decisions based on textual similarity (more details below).

The only disadvantage of using this metric to score vectors is that it does not consider Euclidean distance, or how far vectors actually are in the citation space. In theory, there could be two vectors that have the same cosine angle in the vector space, but are extremely far apart (i.e. (1, 1) and (1000, 1000) in $\mathbb{R}^2$). Practically however, at least in this context, this is irrelevant. There are only two acceptable values in each dimension of the citation space (0 or 1), meaning that any two vectors with the same angle will always have the same norm. Therefore, citation similarity is an acceptable metric for this purpose.

For the citation similarity analysis conducted, the top K predictions (by citation similarity) were analyzed, where K is a chosen parameter. For instance, in the fictitious truth table represented in Figure 4 below, the top K similar documents for the decision 2013fca02 is the case 2018fca83 (where K = 1 in this case).

*Figure 4: Fictional representation of a truth table built for citation analysis (the highlighted rows depict the top-K most similar decisions to 2013fca02, where K = 1).*

| | **2009scc23** | **2018scc10** | **2006scc05** | **…** | **1980scc03** |
|---|---|---|---|---|---|
| **2013fca02** | 1 | 0 | 1 | | 1 |
| **2015fca104** | 0 | 0 | 0 | | 1 |
| **2019fca09** | 0 | 1 | 0 | | 0 |
| | | | . | | |
| | | | . | | |
| | | | . | | |
| **2018fca83** | 1 | 0 | 1 | | 1 |

## 3.3 Predicting Citations

With the data properly preprocessed and cleaned, the main task of predicting citations could be addressed. This section thoroughly describes the methods used in each of the models made, including the baseline and learning-based models.

### 3.3.1 Scoring Metrics

The central task of this research is predicting legal citations using only the texts of the original decision, which is known as an "information retrieval" task within the NLP landscape [47]. Information retrieval typically deals with queries instead of full documents, for instance search engines like Google retrieve websites based on an inputted query.

The research presented here was specifically crafted to remove the human aspect associated with the retrieval, specifically in the query. However, the same scoring metrics can still be applied, to measure the true performance of developed models. There are many, but the most common ranking metrics are known as "recall" and "precision".

Recall refers to the percentage of relevant documents that were returned in the predictions, of all the relevant documents in the corpus. A mathematical representation of this metric is shown below:

$$recall = \frac{\#\ of\ correct\ predictions}{total\ \#\ of\ correct\ predictions}$$

Recall was scored for each FCA decision, and averaged across the corpus. The averages are presented in the results of each model. Also, within each FCA decision, a different number of predictions were returned. The number of predictions is referred to as K. As K increases, the number of predictions increases, which inflates the recall values, since the expected value of correct predictions increases. As a result, the expected value was documented for each prediction, and the net average recall was calculated by taking the difference between the average recall and expected value. The net average recall measure is more indicative of true performance.

Precision is a measure that aims to look at the accuracy of each prediction. Precision@K is a specific measure of precision, for K predictions. A mathematical representation of this metric is shown below:

$$precision@K = \frac{\#\ of\ correct\ predictions}{K}$$

One flaw with this measure is that for sparse datasets such as this one, the precision drastically decreases as K increases. This has little relevance, because predictions are not practical for large K. The ideal predictive model is very precise at small K, and has strong recall at small

K. However, as discussed in the data analysis, recall is low for K=1, since the average number of citations is about 4 per FCA decision. This means that for K=1, perfect recall would be about 0.25. For this reason, the precision and recall results for K=5 were more closely monitored.

Finally, since both of these metrics are important, it is also useful to gather a weighted sum of these measures. The F1 score is a weighted sum of recall and precision, and for this context was measured at each value of K. A mathematical representation is shown below:

$$F1 = 2 \cdot \frac{(average\ precision@K) \cdot (average\ recall)}{average\ precision@K + average\ recall}$$

This measure is often used in information retrieval, and is a good summary of how the model performs on the dataset for given values of K.

The main limitation with these measures is that they assume relevant documents for a given corpus are known beforehand. In this context, this research assumes that the citations are relevant documents. This assumption may pose problems for the results, since the citations within legal research have been shown to contain a significant amount of bias [37] [38] [39]. Also, it assumes that the authors of each FCA decision in the corpus read every possible SCC decision and made logical citations.

There are notable limitations of using citations as relevancy feedback, but anything else would require significant amounts of time. In an ideal world, a team of legal professionals would have to read and take notes on every case within the FCA and SCC corpora, and then choose relevant documents. The choices of each professional would then have to be cross-referenced against each other, to eliminate all bias. This would not only take an extraordinary amount of time, well beyond the scope of this project, and would not eliminate all bias because "relevancy" in this context is a vague concept. As a result, citations should be a good approximation for relevancy.

### 3.3.2 Baseline Predictive Model

The baseline model was the first model built. The motivation behind building this model was to provide a reference when evaluating other models. This was necessary since the literature review found no projects of this nature in legal research. This method had to be simple, so that any proposed model should in theory perform at least as good as the baseline.

The proposed baseline model used a static "bag-of-words" method known as "tf-idf". It uses a combination of term frequency and inverse document frequency to represent documents. It measures the frequency at which each document in a corpus uses words within the whole corpus. This method was chosen because it has been a standard in the field of Natural Language Processing (NLP) for quite some time, and since the literature review showed that this method is useful and sometimes as good as learning-based methods [11].

In addition, this method was specifically chosen because of its flaws. Representations for each document are only computed by analyzing word frequencies. This leaves significant room for error, as it does not consider the contextual meaning of words in the English language.

An example of this deficiency is show in Table 1 below. The example consists of two sentences, each of which is compared to the test sentence "The quick brown fox jumps over the lazy dog" using the cosine distance of their tf-idf representations.

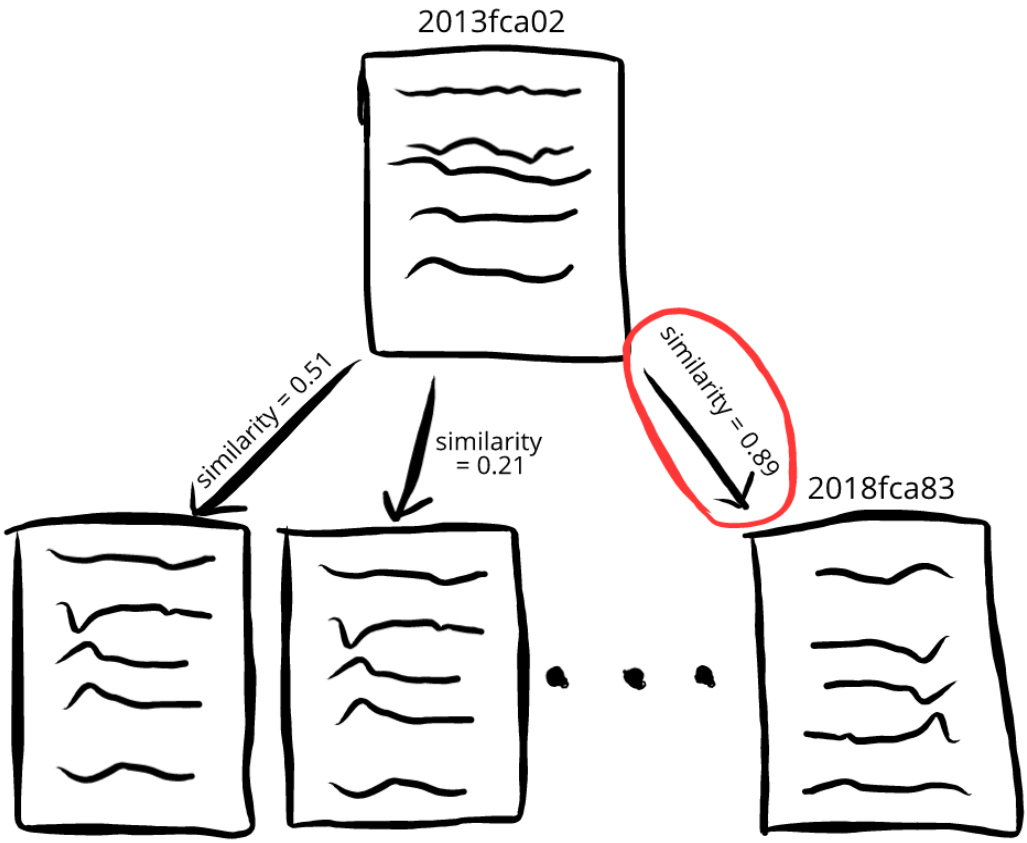
*Table 1: An example of sentence comparison using cosine distance with tf-idf document representations.*

| Sentence | Cosine similarity (using tf-idf) |
|---|---|
| "This is a fast pet" | 0 |
| "The dog is friendly" | 0.35 |

As shown above, "This is a fast pet" shares no words in common with "The quick brown fox jumps over the lazy dog", and therefore they have a similarity score of 0. "The dog is friendly" has two words in common with "The quick brown fox jumps over the lazy dog", and therefore received a distance score of 0.35. In this example, "This is a fast pet" is semantically closer to the test sentence, even though they do not share any common words. As a result, tf-idf fails to identify this, since it cannot uncover the meaning of each sentence. This is a major flaw with "bag-of-words" approaches like tf-idf, as discussed in the literature review.

To make predictions in the baseline model, a tf-idf vector was created for each document, and the similarity between each document was computed using cosine distance. In this way, similarities between two legal texts are computed and compared throughout the entirety of the corpus. Once the similarity scores are computed, the documents are ordered by this similarity score, and the top K documents are returned (shown in Figure 5, where K = 1).

*Figure 5: A representation of the textual analysis performed using tf-idf and cosine similarity scores.*




The top K most similar documents are predicted for each FCA case. Key scoring metrics are calculated, as described in the previous section. These measures are then averaged over the entire FCA corpus, and presented in the Results section. The tf-idf representations are developed from the Python package *TfidfVectorizer* from package *sklearn.feature_extraction.text* [48]. The cosine distance scores are calculated using the *cosine_similarity* Python package from the *sklearn.metrics.pairwise* library [49]. These are both open source libraries offered in Python.

### 3.3.3 Learning Document Embeddings

Tf-idf is a method learning document representations, as described above, but there are much newer techniques in the field of NLP. Word2Vec was the method that truly changed the field in

2013, that learns embeddings for a set of words [13]. One year later, Doc2Vec was proposed as an extension of Word2Vec, for learning embeddings for documents and paragraphs within a corpus. GloVe and ELMo are more recently developed techniques that built upon these foundational teehniques [15] [16]. Each of these methods, along with others, are described in detail in the literature review.

There are many techniques within the field that are currently in use, and it is not quite clear which is better as they are task dependent. A great deal of research has been done in many applications, showing that it is possible to learn embeddings effectively for different tasks such as 10-K filings [18], or for an entirely different language [19]. These examples show us that it is possible to successfully use and implement the word embedding algorithms discussed above.

When approaching this task, two main avenues were explored. One option involved using pre-trained embeddings, which is fairly common. The advantage of this is that these embeddings are thoroughly trained on a corpus much larger than the one in this research project. The disadvantage of using pre-trained embeddings is that the learned embeddings are not specific to legal texts. It was decided to train embeddings specifically for this corpus as a result. This decision was made in part from the observed success in the literature review (as described above as well), but may be a source of limitations in the results.

The Doc2Vec method was chosen to learn the embeddings for this task. This method was chosen because of the success demonstrated from the research of Ash and Chen in 2018, within the legal landscape [31]. In a corpus similar to that of this research project, they found that they "tend to be cited together locate near each other in the embedding space". The learned document embeddings were strongly correlated with the use of citations in their corpus, which was desirable in this project. Also, this method was easily implemented in Python through the *gensim* library, which also factored into the decision process [50].

Ash and Chen built the document embeddings with the distributed bag-of-words model, which samples words from a context window at random and learns the use of text without any context. This made more sense for their task, but ultimately this research chose to use the distributed memory model, since a bag-of-words approach was used in the baseline model. The distributed memory model maintains the order of each sentence, and the context of each word within a sentence is maintained. There are advantages and disadvantages to each learning model

within Doc2Vec, but the distributed memory model was chosen because of its ability to learn the semantic and contextual meanings of words within a corpus.

There are a few other parameters involved as well, including window size, embedding size, learning rate, and number of epochs (or passes through the data). Ash and Chen did not explicitly state all of the parameters they used, but mentioned an embedding size of 200 that trained for 5 epochs. To uncover which parameters to use, research conducted by Lau and Baldwin was consulted, which tested various hyper-parameters and make recommendations for them [51]. For the distributed memory mode, they used an embedding size of 300, a window size of 5, a learning rate of 0.025 with decay, and trained for up to 1000 epochs depending on the application. In accordance with their findings, the same window size and learning rate ($\alpha$) was used. Different embedding sizes and number of epochs were used however, due to the computational complexity associated. Embedding sizes of 250 and 1000 were chosen, but only trained for 100 and 10 epochs respectively. Two embedding sizes were chosen to examine the effects on learning. It should be acknowledged that this is a limitation of the learned embeddings, but still are more complex than what was used by Ash and Chen, and therefore are good enough for the task.

There are obvious limitations in the methods used to train the embeddings. Of course, pre-trained embeddings should have been consulted, and future projects should definitely explore this. Also, methods other than Doc2Vec should be explored, to examine the effects on performance. There is much room for optimization in the tuning of the hyper-parameters used, including training the embeddings for more epochs. This should all be considered when viewing the results of each model's performance.

### 3.3.4 Deep Learning Models

This research is concerned with showing that Artificial Intelligence models can learn the language of the law. Deep Learning, a subset of Artificial Intelligence, was used to learn how to predict citations given the text of a bias-free legal decision. As described in the literature review, these techniques are not currently used in the field. The models built and developed for this task are all referenced to models seen in the literature review of other fields.

Two distinct models were developed in this research, each serving a distinct need. Both models were trained using the trained Doc2Vec embeddings thoroughly described in the previous section. This process of layering models together is common in NLP research.

The first model is known as a Multilayer Perceptron (MLP), but also referred to as a "vanilla" neural network because it is the most basic design available [52]. The designed MLP consisted of 3 layers; an input layer, hidden layer, and output layer. This specific model was used because under the "Universal Approximation Theorem", an MLP with a single hidden layer and non-linear activation functions is capable of modelling any function to any arbitrary degree of accuracy [53]. This mathematical fact states that the MLP is the most basic deep learning model capable of learning any arbitrary representation of data.

As a result, this was the first design chosen to try and learn the citation data, and make predictions. The learning task was designed as a multi-class classification, which allowed for multiple predictions to be made across each of the designated SCC cases. The model and training of the model was implemented in *PyTorch*, an open source machine learning framework available in Python [54].

The model was designed to have only the FCA embeddings inputted into the model, which purposefully limited the ability of the learning process. The dataset was split into a training, validation and test sets that comprised of 70%, 20% and 10% of the full dataset respectively. These splits are done to ensure the model is learning, and not simply memorizing the training set. Two test sets (a validation and test set) are used to ensure that the model does not memorize the validation set, since the test set is kept completely outside of the training process. These splits are pretty standard in the field as it is based on the "Pareto Principle", where an additional 10% is taken from the training set and devoted to a test set [55].

The model trained on the training set, with random batches of 50 examples taken from it at a time. The input size varied, since different embedding sizes were tested in this example. The hidden layer had size 100, while the output layer was the number of SCC citations in the dataset. The Rectified Linear Unit (ReLU) activation function was used in between each layer, while the output layer was given a sigmoid activation. The loss function chosen is a Binary Cross Entropy Loss (BCELoss), and the optimizer chosen is Adam. Also, the weights from this model were initialized with Xavier initialization, a technique that avoids getting stuck in local minima at the beginning of training. The results from this model are described in detail in the next section.

Ultimately, the MLP was not capable of representing the dataset well. Several different loss functions, optimizers, and hyper-parameters (i.e. learning rate, batch size) were tested, but the

results did not improve by much. It was ultimately concluded that there was not enough data inputted to the models, since it had no information about what it was predicting.

To introduce the SCC embeddings into the learning, a new model had to be constructed for a few reasons. The challenge with crafting this model was handling the introduced memory and computational complexity. Introducing the SCC embeddings meant adding about 10,000 embeddings, each of size 250. If the same MLP model was used, the memory and computational toll would increase by a factor of at least 10x. For this reason, a CNN structure was introduced to help reduce the number of connections within the model, while not compromising on the ability to learn.

CNN's are typically used for tasks involving images. They are extremely good at feature extraction, and contain two important properties known as *invariance* and *equivariance* to changes in the inputs (meaning that the ordering of the data and perturbations in the data do not affect the output) [56]. For the given task, these were desired properties. Aside from the memory and computational savings, the CNN was chosen because it could extract key pieces of information from the SCC embeddings. A 2-layer CNN was ultimately chosen over other architectures for this reason.

This model was also implemented and trained in PyTorch. The CNN architecture was layered on top of the previous MLP, and applied over the entire embedding matrix of the SCC corpus and the current FCA embedding. The first layer had an output of size 50x250, kernel size of 15x15, padding of 7 and a stride of 1. The second convolutional layer had an output size of 1x250, kernel size of 5x5, padding of 2, and a stride of 1. The 1x250 output from the CNN was concatenated with the 1x250 representation of the current FCA decision, and inputted into the 3-layer MLP as before. The MLP had the exact same specifications as above, and the same loss and optimizer were used. After some testing, the optimal batch size was 1, so this was used for all experiments. All specifications mentioned above achieved the best results on the dataset. There is much room for experimentation in the construction of these architectures.

# 4 Results

This section thoroughly describes the results of each experiment performed in the research project. Refer to the Methods section for a detailed description of the experiments presented below.

## 4.1 Summary of the Dataset

The first conducted experiment was an investigation of the dataset. This is a vital step in the research project, as it uncovers subtleties within the data that might inform results. A holistic summary of the dataset, along with various preprocessing methods applied, is discussed below.

### 4.1.1 The Downloaded Dataset

A total of 3,360 Federal Court of Appeal (FCA) decisions were collected, as well as 11,354 Supreme Court of Canada (SCC) decisions. The downloaded FCA corpus comprised decisions from a period of 2005 to 2019, while the downloaded SCC corpus spans from as far back as 1867 to 2019. Table 2 below shows a summary of the decisions downloaded from each court.

*Table 2: Summary of the FCA and SCC decision corpora.*

| Court | Total number of downloaded decisions | Date range |
|---|---|---|
| Federal Court of Appeal (FCA) | 3,360 | 2005-2019 |
| Supreme Court of Canada (SCC) | 11,354 | 1867-2019 |

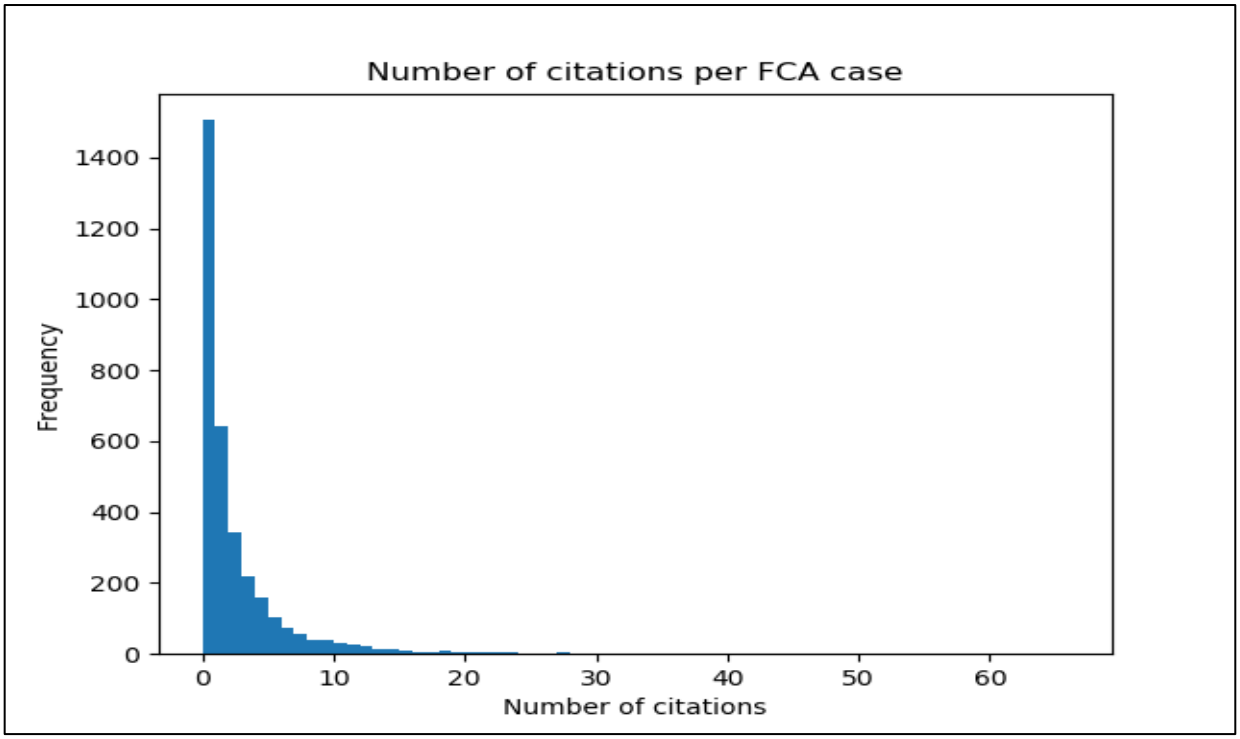
Within the downloaded FCA corpus of 3,360 decisions, only 2,724 decisions have any citations, while only 1,855 of them (about 55% of the downloaded dataset) cite to the SCC.

Of the 11,354 downloaded SCC decisions, only 9,747 cases were used. Some of these cases were dropped because they were written in French. For simplicity, the project was scoped to only English, which is common practice in NLP tasks. Fortunately, limiting the SCC dataset to 9,747 files did not have a significant effect on the citation landscape and therefore did not significantly impact the predictions.

### 4.1.2 Analyzing the FCA Corpus

As mentioned above, of the 3,360 downloaded FCA decisions, 1,855 of them had at least one citation to the 9,747 SCC decisions. The average number of citations to the SCC from the dataset is 2, where the median is 1 and the maximum is 66. A histogram of this distribution of citations is plotted in Figure 6 below.

*Figure 6: Number of SCC citations per 3,360 downloaded FCA decisions.*

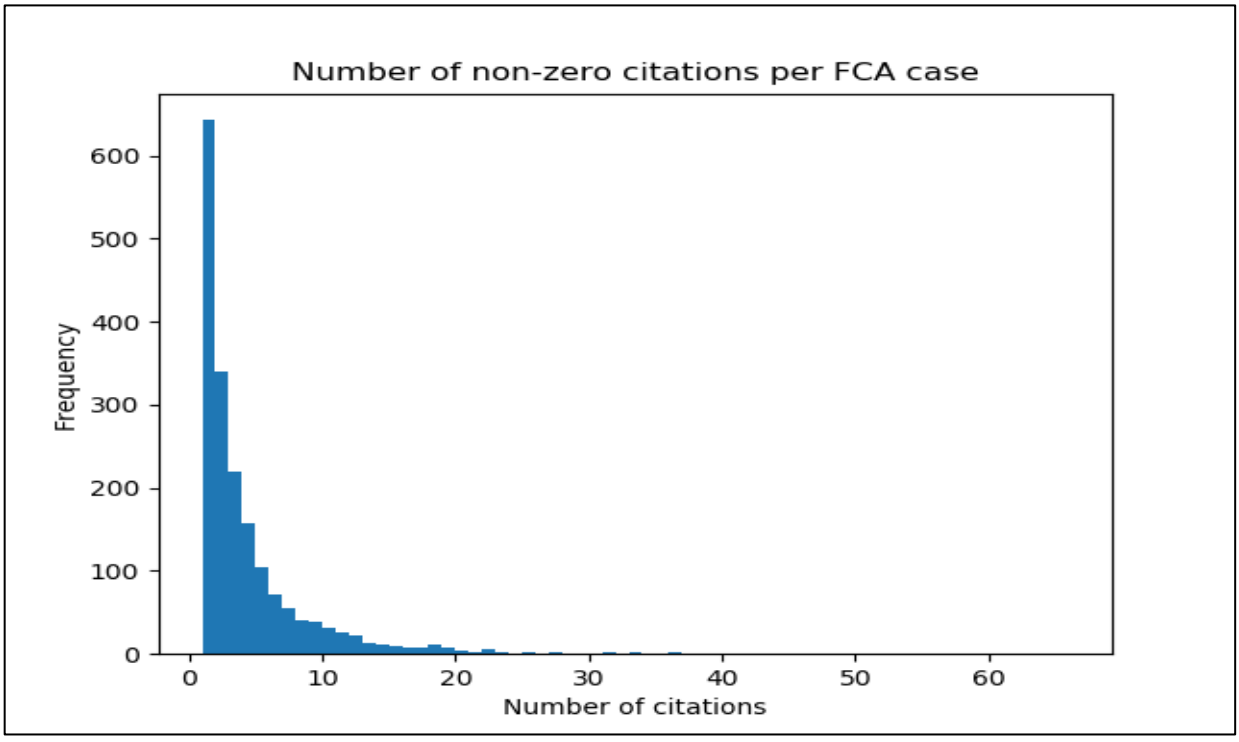



As seen in Figure 6, a majority of the downloaded FCA decisions (roughly half of the dataset) do not cite to the SCC. Although the maximum is 66, almost all of the cases have less than 10 citations to the SCC. This confirms that the dataset, and more specifically the truth table, is extremely sparse. There are 6,722 citations out of a possible 32,749,920 combinations (9,747x3,360). In other words, only about 0.02% of the citation space has valid citations. This is typically not optimal for learning predictive models, especially when the predictions are already as large as 9,747 decisions.

As a result, only the FCA decisions with at least one citation were used. This reduces the amount of decisions from 3,360 to 1,855. By eliminating about half of the downloaded FCA cases, the proportion of citations to FCA cases increased by a factor of 2, which helps enhance the prediction space. Aside from this, the FCA decisions with no citations were dropped for practical reasons as well. The direct application of this research would be to create a program that predicts which decisions a particular case should reference/cite. There would be no added value to the user (in this case a legal professional) if the program predicts that there will be no citations, and the user should not consult any decisions whatsoever.

For these reasons, only FCA decisions with at least one citation were kept in the dataset. The average number of citations to the SCC among the remaining corpus is now 4, with median 2 and max 66. A histogram of this distribution is shown in Figure 7 below.

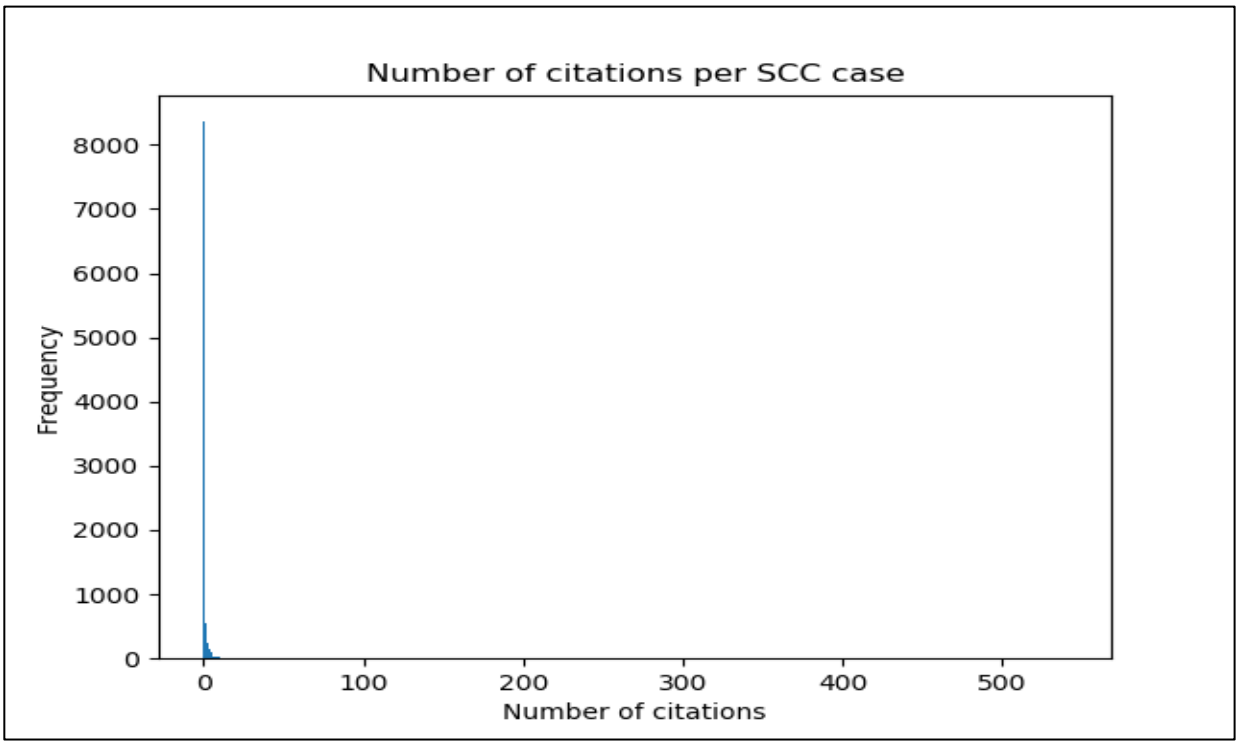
*Figure 7: Number of SCC citations per 1,855 FCA decisions with at least one citation.*



### 4.1.3 Analyzing the SCC Corpus

Of the 9,747 decisions in the SCC corpus, 8,360 (85.8%) of them are never cited by the FCA corpus. Correspondingly, the average number of citations to each SCC decision is 0, the median is 0, but the max is 541. This lines up with what has been discovered frequently in citation research [39], that a majority of cases are never cited and a select few are heavily cited (also known as "precedents"). A histogram of this distribution is shown in Figure 8 below. A majority of the decisions are shown on the left (less than 10 citations), while there are so few beyond that to the right that it can't be seen visually.

*Figure 8: Number of citations to each of the 9,747 SCC decisions.*



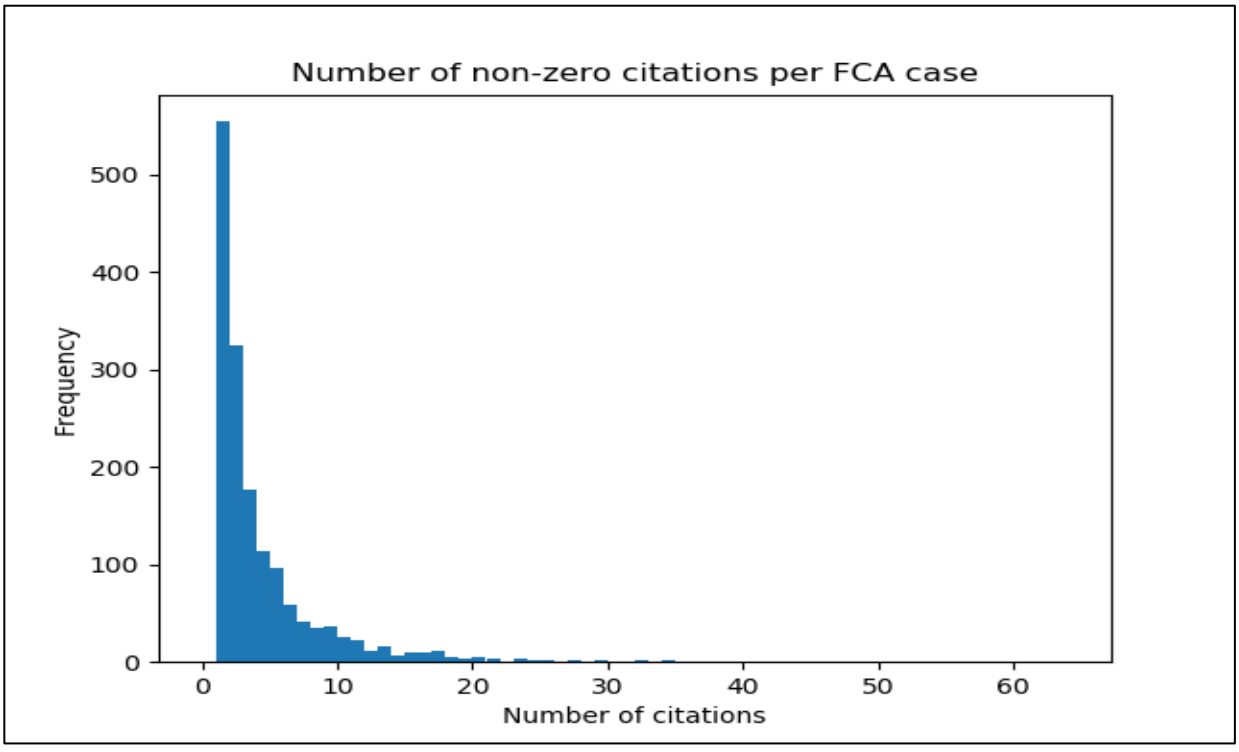

1,387 of the 9,747 SCC decisions have at least one citation. Of these decisions, the average is 5 citations to them, the median is 2, and the max is 541. When investigating a bit further, the three most-cited SCC cases (from the downloaded FCA corpus) are: "Housen v. Nikolaisen" (2002 SCC 33) with 514 citations [57]; "Dunsmuir v. New Brunswick" (2008 SCC 9) with 448 citations [58]; and "Agraira v. Canada" (2013 SCC 36) with 200 citations [59]. After these three, the next highest SCC case has 122 citations, and then it drops off from there.

These are extremely high numbers, and about 17% of all the citations from the downloaded FCA corpus to the downloaded SCC corpus belongs to these three citations. Upon further investigation, it was discovered that these three cases are known as "standard of review cases", which explains why they are so often cited in the FCA corpus. More aptly put, "the standard of review is the legal approach to analyzing the decision" [60]. So, in many decisions, standard of review cases will be cited to demonstrate the effectiveness of an argument, although it does not develop the legal arguments made. As a result, these cases can introduce noise to predictions.

When these three standard of review cases are removed from the SCC corpus, the number of FCA cases with at least one citation drops from 1,855 to 1,588. In other words, a total of 267 FCA cases only cited one of these standard of reviews, which could introduce a significant amount of bias. All of the methods presented in this thesis were tested against the dataset with and without the standard of review cases, to examine its effects.

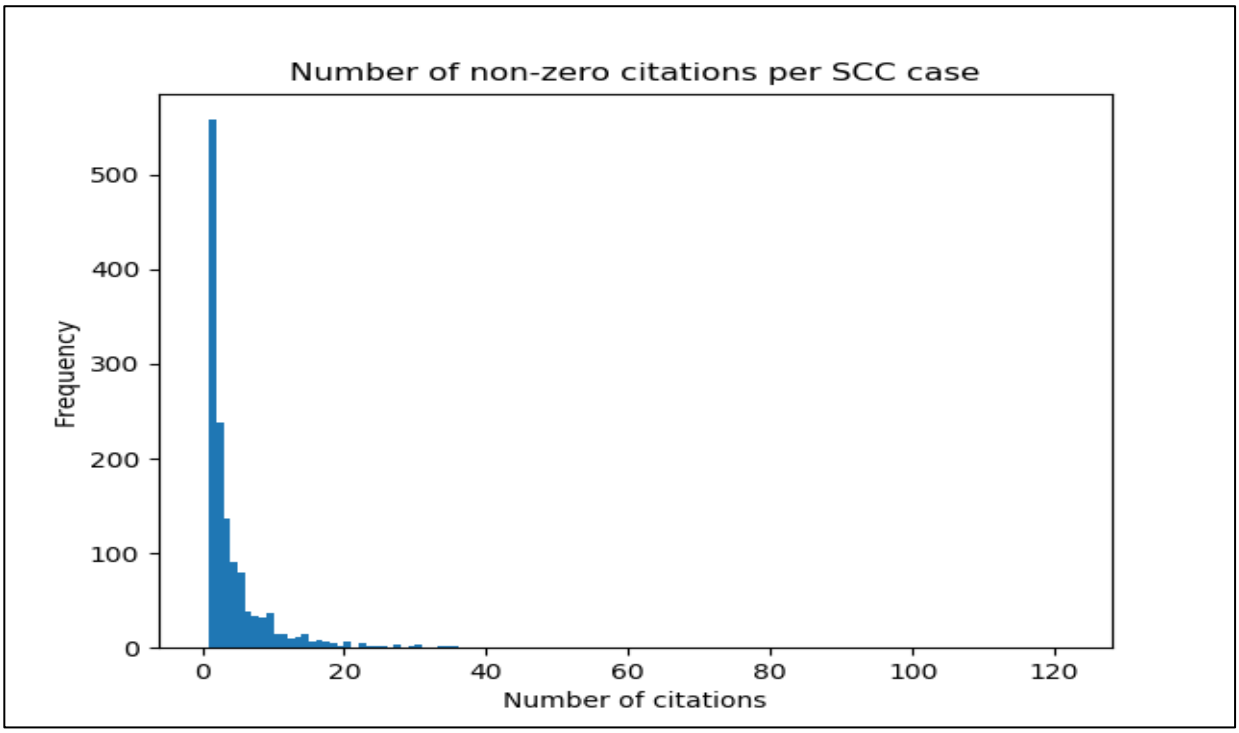
When removing these cases, as shown in Figure 9 below, the histogram of the number of citations per FCA case becomes slightly more balanced. Now, with the removal of these cases, the remaining 1,588 FCA cases cite on average 4 cases (with a median of 2 and a max of 64).

*Figure 9: Number of SCC citations per 1,588 FCA decisions with at least one citation (without standard of review cases).*



The average number of citations for each of the 9,744 remaining SCC cases is now 4 (with a median of 2 and max of 122). Both datasets (with and without the standard of review cases) are used throughout the rest of the analysis to examine the effect on predictions. Similarly, with the removal of the standard of review cases and removing the SCC cases with less than one citation, the histogram becomes much more balanced, as shown in Figure 10 below.

*Figure 10: Number of citations to the 1,384 SCC decisions with at least one citation.*




## 4.2 Analyzing the Text and Citation Usage within the FCA Corpus

To investigate the correlation between the textual similarity and citation similarity in the corpus, an initial analysis was conducted. In this analysis, textual similarity between the preprocessed (unbiased) FCA decisions was conducted using a "bag-of-words" approach known as tf-idf. Tf-idf ("term frequency inverse document frequency") is a standard method used in the field of Natural Language Processing (NLP) [61]. It represents documents within a corpus as a vector, based on the words used in the document with respect to all words used in the full corpus. The representation of each document depends on the number of times a word is used in each

document (term frequency), and how unique the word is to the corpus (inverse document frequency). So, if two decisions in the FCA corpus use a specific word such as "supercalifragilisticexpialidocious", tf-idf would pick up on that, and these two documents would appear to be more similar as a result. One of the obvious flaws with tf-idf is that it does not consider the context of the way in which words are used. Also, in this context, it may pick up on specific words that are relevant to a certain case, and may sway results accordingly. More details about this technique, along with its limitations, are presented in the Methods section.

For this analysis of the data, the textual tf-idf similarities and the citation similarities were organized by a method known as "cosine similarity". Using this analytical method, documents are scored and listed by their tf-idf scores and citation scores. In this way, it is possible to gather FCA cases that cite similarly and FCA cases that use words similarly.

For each FCA decision, the K-most similar documents were determined via textual similarity and citation similarity (where K is [1, 5, 10, 20, 50, 100, 1000]). The overlap between the predictions using textual and citation similarity were measured and documented for each value of K. This measures the overlap between the text-based predictions exist and the citation-based predictions.

When K=1 for instance, one prediction will be made using citation similarity and one prediction will be made using textual similarity. These predictions will be compared and if they are the same, the overlap is 1 (or 100%), otherwise it is 0 (or 0%). This measure was applied to all values of K listed above, and averaged across the complete dataset. The results of this analysis are shown in Table 3 below, showing the "average overlap" scores, the expected value (random probability of overlap) and the net correlation, and the median and max scores for different K values across the full FCA corpus.

*Table 3: Citation similarity and tf-idf textual similarity analysis of 1,855 FCA decisions.*

| K | Average overlap | Expected value | Net Correlation | Median | Max. |
|---|---|---|---|---|---|
| 1 | 0.034 | - | +3.34% | - | 1.00 |
| 5 | 0.060 | 0.003 | +5.75% | - | 0.80 |
| 10 | 0.073 | 0.005 | +6.77% | - | 0.80 |
| 20 | 0.089 | 0.011 | +7.78% | 0.05 | 0.65 |
| 50 | 0.113 | 0.027 | +8.62% | 0.08 | 0.72 |
| 100 | 0.144 | 0.054 | +8.96% | 0.11 | 0.78 |
| 1000 | 0.573 | 0.539 | +3.38% | 0.57 | 0.70 |

As Table 3 highlights, there is a noticeable correlation between the tf-idf textual similarity and citation similarity of the FCA decisions. The average overlap start at 3.4% when K=1, and jumps to 57.3% when K=1000. These scores seem extraordinary, but it is important to note that there are only 1,855 documents in the corpus, so K=1000 accounts for 53.9% of the corpus. The effect of random guessing is shown in the "Expected value" column. The "net correlation" column shows the difference between the average overlap and random guessing, to truly highlight the added value within the predictions. A positive score in this column means that there is a positive correlation between citation similarity and textual similarity.

The net correlation starts at +3.34% and hovers around +7-9%. The positive values help support the initial hypothesis of the project, that cases of similar language will cite the same cases. This finding suggests that there may be a correlation between the way in which language is used in decision documents, and the way in which they cite. This initial analysis ultimately justified the hypothesis of the project, and confirmed that learning the relationship between citation usage and language usage is possible.

The same test was performed without the standard of review cases. When these citations were removed, 267 FCA decisions no longer cited to any of the SCC corpus. These FCA decisions were removed for this analysis, since they would not contribute additional information in the citation similarity scores. The results are summarized in Table 4 below.

*Table 4: Citation similarity and tf-idf textual similarity analysis of 1,588 FCA decisions.*

| K | Average overlap | Expected value | Net correlation | Median | Max. |
|---|---|---|---|---|---|
| 1 | 0.050 | - | +4.97% | - | 1.00 |
| 5 | 0.087 | 0.003 | +8.35% | - | 0.80 |
| 10 | 0.104 | 0.006 | +9.74% | - | 0.80 |
| 20 | 0.123 | 0.013 | +11.0% | 0.10 | 0.85 |
| 50 | 0.152 | 0.031 | +12.1% | 0.12 | 0.74 |
| 100 | 0.175 | 0.063 | +11.2% | 0.13 | 0.78 |
| 1000 | 0.645 | 0.630 | +1.52% | 0.65 | 0.71 |

Interestingly, removing the standard of review cases had a significant effect on the results. The net correlation values improve consistently by about 1.5% across all values of K. Improvements are also noted in the median and maximum columns, and these highlight more robust predictions. Therefore, it appears as though the way in which standard of review cases are cited is not correlated with the textual similarity between documents. This is consistent with intuition, since as noted above, the standard of review cases that are mentioned do not develop the main argument. They show that the reasoning provided in the decision was developed in a sound manner. Given the increase in results, it could also be argued that the standard of review cases may even have a net negative effect on predictions. This finding was noted when designing and building new predictive models for SCC citations.

## 4.3 Predicting Citations using Tf-idf Similarity

After the initial data analysis and investigation into the FCA corpus described above, it became clear that there is a correlation between citation similarity and textual similarity between the downloaded FCA cases. This meant that predicting, or at least learning how to predict SCC citations, could be done reasonably well.

Developing a baseline predictive model became the next objective. It was believed that this would help inform how the newly designed models were performing. To score possible citations by textual similarity, tf-idf was used again. It is perfect for a baseline model, because anything built beyond this should perform at least as good as this approach, since it does not consider the

context in which words are used in documents. In the same manner as the analysis described above, each document of the corpus was represented using tf-idf, and similar documents are gathered using cosine similarity. In this way, each FCA decision in the corpus is compared to each SCC decision in the corpus, and comparisons are made based on textual similarity.

The K most similar SCC decisions for each FCA decision based on tf-idf are predicted for each value of K, and the K most similar measures are gathered via cosine similarity. For different values of K, the model's performance can be measured and examined. Three distinct measurements were taken, known as "precision@K", "recall", and the "F1 score". These are all commonly used measures for information retrieval tasks such as this, and more details are provided in the Methods section. In short, precision@K measures the accuracy of the K predictions, recall measures the percentage of all relevant documents predicted in K predictions, and F1 is a weighted combination of the two.

The first analysis involved using the raw texts from the FCA and SCC corpus (of 1,855 and 9,747 decisions respectively), and K values of [1, 5, 10, 20, 50, 100, 200, 500, 1000]. Like the previous investigation with FCA decisions described above, the expected value for recall is noted and a net score is analyzed. Also, the precision@K and recall values for each value of K were collected and averaged across the FCA dataset. These results are presented as "average precision@K" and "average recall" in Table 5 below.

*Table 5: Baseline citation predictions (using the raw 1,855 FCA and 9,747 SCC texts).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.198 | 0.069 | - | +6.9% | 0.103 |
| 5 | 0.124 | 0.193 | - | +19.3% | 0.151 |
| 10 | 0.087 | 0.255 | 0.001 | +25.4% | 0.130 |
| 20 | 0.058 | 0.321 | 0.002 | +31.9% | 0.098 |
| 50 | 0.032 | 0.429 | 0.005 | +42.4% | 0.060 |
| 100 | 0.020 | 0.515 | 0.010 | +50.5% | 0.038 |
| 200 | 0.012 | 0.617 | 0.021 | +59.6% | 0.023 |
| 500 | 0.006 | 0.738 | 0.051 | +68.7% | 0.012 |
| 1000 | 0.003 | 0.838 | 0.103 | +73.5% | 0.007 |

The average precision when K=1 is roughly 20%, which is pretty high, and the net average recall is 6.9%. This means that for about every 5 FCA cases, the top prediction is actually cited 1 of those times. As K increases, the average precision decreases, but this does not mean that the model becomes less precise. This result is due to an increase in K but a stagnant number of true citations. To illustrate this subtlety, note that the average recall and net average recall values monotonically increase as K increases, which implies that more relevant citations are being predicted. Therefore, for theoretical and practical purposes, precision@K is only a valuable measurement in this context for small K. Similarly, average and net average recall scores are not as useful for small K values, since as discovered in the data analysis, only about half of the FCA decisions have one citation. This is why the F1 score is so low for K=1, and thus K=5 is the most meaningful measurement in the dataset.

These results generally seem pretty high, at least higher than intuitively expected. It *is* possible that tf-idf is picking up on the citations within the raw FCA texts, and matching them to the correct SCC text. To be sure, the same analysis was applied to the preprocessed FCA texts with the citations removed. These results are listed in Table 6 below.

*Table 6: Baseline citation predictions (using the preprocessed 1,855 FCA and 9,747 SCC texts).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.175 | 0.061 | - | +6.1% | 0.090 |
| 5 | 0.110 | 0.171 | - | +17.1% | 0.134 |
| 10 | 0.077 | 0.222 | 0.001 | +22.1% | 0.114 |
| 20 | 0.053 | 0.294 | 0.002 | +29.2% | 0.090 |
| 50 | 0.029 | 0.383 | 0.005 | +37.8% | 0.055 |
| 100 | 0.018 | 0.467 | 0.010 | +45.7% | 0.035 |
| 200 | 0.011 | 0.557 | 0.021 | +53.6% | 0.021 |
| 500 | 0.005 | 0.688 | 0.051 | +63.7% | 0.011 |
| 1000 | 0.003 | 0.799 | 0.103 | +69.6% | 0.006 |

As expected, the average precision@K scores decrease by about 2.5% across all K values, and average recall decreases as well. Correspondingly, the F1 score at K=5 dropped from 0.151 to 0.134. This may not seem like much, but removing all forms of bias within the texts did have a considerable effect on the predictive power. This could mean that the tf-idf approach is merely

looking at particular words within the text (i.e. citations), and making predictions based on that. In theory, there is nothing wrong with this approach, and ultimately the results are incredibly effective for a static method that does not learn.

One final test was performed using this same predictive model, but now the standard of review SCC decisions were removed, along with the 267 FCA cases that did not have any citations as a result. The results are presented in Table 7 below, and interestingly the results improved.

*Table 7: Baseline citation predictions without standard of review cases (using the preprocessed 1,588 FCA and 9,744 SCC texts).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.201 | 0.081 | - | +8.1% | 0.115 |
| 5 | 0.121 | 0.214 | - | +21.4% | 0.155 |
| 10 | 0.084 | 0.276 | 0.001 | +27.5% | 0.129 |
| 20 | 0.057 | 0.355 | 0.002 | +35.3% | 0.098 |
| 50 | 0.031 | 0.456 | 0.005 | +45.1% | 0.058 |
| 100 | 0.019 | 0.536 | 0.010 | +52.6% | 0.037 |
| 200 | 0.011 | 0.619 | 0.021 | +59.8% | 0.022 |
| 500 | 0.006 | 0.730 | 0.051 | +67.9% | 0.012 |
| 1000 | 0.003 | 0.818 | 0.103 | +71.5% | 0.006 |

The average precision scores roughly deviated back to the original levels with raw texts, while the average recall improved beyond its original levels. This change is reflected in an increased F1 score of 0.155 for K=5. This confirms the suspicion that the standard of review case citations do not contribute any valuable information to the predictions, and actually introduce noise in the predictions.

Another experiment was conducted on the same dataset provided above, but with all quotes within the FCA decisions stripped. It was suspected that the tf-idf similarity measure was picking up on direct quotes used throughout the FCA corpus. The results from this experiment are not presented here, simply because they were incredibly similar to the previous experiment discussed above. There was no conclusive evidence to claim that keeping the quotes influenced predictions negatively or positively, and as a result, they were kept in the FCA corpus for future models.

Another limitation of the presented baseline model is that the predictions are independent of time. The results presented above, in theory, are extremely conservative, since it is possible for

the model to predict SCC citations that were not available to the author of the FCA decision at the time of the cases publication. In fact, this was observed in the existing model, as described in the Discussion section. A final test was conducted on the same model, but only valid predictions (in terms of time) were kept. The results are presented in Table 8 below.

*Table 8: Baseline citation predictions without standard of review cases and release dates accounted for (using the preprocessed 1,588 FCA and 9,744 SCC texts).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.241 | 0.094 | - | +9.4% | 0.136 |
| 5 | 0.129 | 0.223 | - | +22.3% | 0.164 |
| 10 | 0.088 | 0.284 | 0.001 | +28.3% | 0.135 |
| 20 | 0.060 | 0.365 | 0.002 | +36.3% | 0.102 |
| 50 | 0.032 | 0.462 | 0.005 | +45.7% | 0.060 |
| 100 | 0.019 | 0.541 | 0.010 | +53.1% | 0.037 |
| 200 | 0.011 | 0.625 | 0.021 | +60.4% | 0.022 |
| 500 | 0.006 | 0.734 | 0.051 | +68.3% | 0.011 |
| 1000 | 0.003 | 0.822 | 0.103 | +71.9% | 0.006 |

The average precision for K=1 increases by 4%, and the average recall improves by 1.3% for K=1. This is shown in the F1 score, which improves by 0.021 for K=1. The F1 score for the top 5 predictions increased as well, from 0.155 to 0.164. As intuitively predicted, removing invalid predictions helped improve the model. In this analysis, only the year of publication for each decision was used, as the specific date was too hard to extract. In theory, edge cases are possible, where the prediction was published after the FCA case but within the same year. These situations should not happen frequently, and should be inconsequential to the presented results.

As shown above, the predictive power of the tf-idf similarity is considerably good in this context. This is a similar finding to what was mentioned by Shahmirzadi et al. [11]. The qualitative analysis of this model is discussed in detail in the Discussion section, and this model will be referenced as a baseline in comparison to the newly developed predictive models discussed below.

## 4.4 Predicting Citations with Deep Learning

With the baseline results, it became possible to design new models and test their effectiveness. At this point, the literature review of NLP applied to legal texts became helpful. To get the texts into workable form, document embeddings were learned for each decision in the FCA and SCC corpora. Doc2Vec was the method used to learn these embeddings. This is a paragraph or document targeted method of learning embeddings, an extension of version of Word2Vec, proposed by Le and Mikolov in their seminal paper from 2014 [30]. More details about the algorithm are provided in the Methods section.

The Python library *gensim* was used to implement the Doc2Vec method of learning document embeddings [50]. The 'distributed memory' setting was used as opposed to the 'distributed bag-of-words' setting, since it was intended for the structure of the sentence and contexts of the words to be maintained in some capacity. Standard learning parameters were used in training, as referenced with Lau and Baldwin's exploration with large corpora [51]. With that said, there is plenty of room for experimentation and optimization within this portion of the research, as more research needs to be conducted to explore the embedding space for legal texts.

More explicitly, model parameters of $\alpha = 0.025, window\ size = 5, and\ \alpha_{min} = 0.01$ were used to train the embeddings. The size of the embedding vector was experimented with, however. Two sets of embeddings were trained on the model, one with embedding size 250 (that trained for 100 epochs) and one that had embedding size 1000 (that trained for 10 epochs). These two sizes were chosen based on what has been used in research such as Lau and Baldwin's work mentioned above, but the embedding size of 1000 trained extremely slowly and only 10 epochs ran. Relative to Lau and Baldwin who reference up to 1000 epochs, 100 epochs for the embedding size of 250 is quite small as well, and this could be improved with more computation available in a future project. Under the given time and computation constraints of this project, these were the only feasible methods of training, and could very well be improved.

With the learned representations of each decision, building learning-based models to predict SCC citations became possible. Two main model architectures were used, both of which involved neural networks. The first model, intended to be simple, was a 3-layer Multilayer Perceptron (MLP). The inputs to this model only consisted of the learned embeddings of the FCA corpus. The second model was built with the intention of introducing the learned SCC embeddings into the model as contextual information. To do so, a 2-layer Convolutional Neural Network (CNN)

was added to scan the embeddings of the SCC corpus, condense the information to save computation, and concatenate the results to the original MLP model. The specific details of the model architectures are discussed thoroughly in the Methods section, and the results of the training is listed in the next two sections.

### 4.4.1 MLP Model Results

The same model was trained on the trained embeddings of size 250 and 1000, to quantitatively evaluate which learned embeddings was more powerful. The dataset of 1,855 FCA decisions was split into training, validation and test sets (which included the standard of review cases). The training set comprised 70% of the dataset, while the validation set was given 20% and the test set 10%, a relatively standard data allocation.

Each model is measured against the same metrics used in the baseline: "average precision@K"; "average recall"; "net average recall"; and "F1" scores. Each model is evaluated against K values of [1, 5, 10, 20, 50, 100, 200, 500, 1000], and the results are split between the training, validation and test sets.

The first experiment was training the model on the embedding vectors of size 1000. Quickly after training began, it became very apparent that no learning was taking place. It quickly approached a minimum loss solution after the first epoch, and always predicted the same thing. The results of the training, validation and test sets are presented in Tables 9-11 below.

*Table 9: MLP model performance on the training dataset of 1,335 FCA examples (embedding size 1000).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.006 | 0.002 | - | +0.2% | 0.003 |
| 5 | 0.009 | 0.013 | - | +1.3% | 0.011 |
| 10 | 0.007 | 0.018 | 0.001 | +1.7% | 0.011 |
| 20 | 0.006 | 0.028 | 0.002 | +2.6% | 0.010 |
| 50 | 0.003 | 0.033 | 0.005 | +2.8% | 0.006 |
| 100 | 0.003 | 0.060 | 0.010 | +5.0% | 0.006 |
| 200 | 0.005 | 0.204 | 0.021 | +18.4% | 0.009 |
| 500 | 0.005 | 0.574 | 0.051 | +52.4% | 0.009 |
| 1000 | 0.002 | 0.613 | 0.103 | +51.2% | 0.005 |

*Table 10: MLP model performance on the validation dataset of 334 FCA examples (embedding size 1000).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.003 | - | - | - | 0.001 |
| 5 | 0.010 | 0.012 | - | +1.2% | 0.011 |
| 10 | 0.007 | 0.017 | 0.001 | +1.6% | 0.010 |
| 20 | 0.004 | 0.030 | 0.002 | +2.8% | 0.011 |
| 50 | 0.003 | 0.036 | 0.005 | +3.1% | 0.006 |
| 100 | 0.003 | 0.067 | 0.010 | +5.7% | 0.006 |
| 200 | 0.005 | 0.241 | 0.021 | +22.1% | 0.009 |
| 500 | 0.004 | 0.570 | 0.051 | +51.6% | 0.008 |
| 1000 | 0.002 | 0.608 | 0.103 | +50.7% | 0.004 |

*Table 11: MLP model performance on the test dataset of 186 FCA examples (embedding size 1000).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.005 | 0.001 | - | +0.1% | 0.001 |
| 5 | 0.004 | 0.004 | - | +0.3% | 0.004 |
| 10 | 0.004 | 0.006 | 0.001 | +0.5% | 0.005 |
| 20 | 0.004 | 0.013 | 0.002 | +1.1% | 0.006 |
| 50 | 0.003 | 0.023 | 0.005 | +1.8% | 0.005 |
| 100 | 0.002 | 0.041 | 0.010 | +3.1% | 0.005 |
| 200 | 0.004 | 0.168 | 0.021 | +14.8% | 0.008 |
| 500 | 0.004 | 0.512 | 0.051 | +46.1% | 0.008 |
| 1000 | 0.002 | 0.574 | 0.103 | +47.3% | 0.005 |

It is evident from the results that the model was not able to successfully learn the dataset. The average recall and precision scores are nearly zero for each of the data sets, for low values of K. The F1 score is at least 10x worse than the baseline results for K=5, across each of the datasets. Another bad sign is that the performance is really similar among each of the datasets, which implies that the model is not learning from the training set. This also confirmed that the model has learned to always predict the same cases, and upon further inspection that was confirmed. These findings are described in detatil in the Discussion section. A few model parameters such as learning rate and loss functions were tweaked to try and help the learning, but the problem persisted. It

appeared as though the problem lied with either the architecture, the amount of training data, or the inputted parameters.

The same model was retrained using the trained Doc2Vec embeddings of size 250 for 10 epochs. This model appeared to learn for the first 10 epochs, after which it stopped and the same predictions were being made for every FCA decision. Model parameters were tweaked, but to no avail. The results across each data set are presented in Tables 12-14 below.

*Table 12: MLP performance on the training dataset of 1,335 FCA examples (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.218 | 0.099 | - | +9.9% | 0.136 |
| 5 | 0.064 | 0.138 | - | +13.8% | 0.088 |
| 10 | 0.044 | 0.177 | 0.001 | +17.6% | 0.070 |
| 20 | 0.027 | 0.208 | 0.002 | +20.6% | 0.047 |
| 50 | 0.012 | 0.228 | 0.005 | +22.3% | 0.023 |
| 100 | 0.007 | 0.244 | 0.010 | +23.4% | 0.013 |
| 200 | 0.004 | 0.261 | 0.021 | +24.0% | 0.008 |
| 500 | 0.002 | 0.305 | 0.051 | +25.4% | 0.004 |
| 1000 | 0.001 | 0.364 | 0.103 | +26.1% | 0.002 |

*Table 13: MLP performance on the validation dataset of 334 FCA examples (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.240 | 0.110 | - | +11.0% | 0.151 |
| 5 | 0.065 | 0.142 | - | +14.2% | 0.089 |
| 10 | 0.043 | 0.174 | 0.001 | +17.3% | 0.069 |
| 20 | 0.026 | 0.194 | 0.002 | +19.2% | 0.045 |
| 50 | 0.011 | 0.206 | 0.005 | +20.1% | 0.022 |
| 100 | 0.006 | 0.224 | 0.010 | +21.4% | 0.012 |
| 200 | 0.004 | 0.239 | 0.021 | +21.8% | 0.007 |
| 500 | 0.002 | 0.287 | 0.051 | +23.6% | 0.004 |
| 1000 | 0.001 | 0.334 | 0.103 | +23.1% | 0.002 |

*Table 14: MLP performance on the test dataset of 186 FCA examples (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.226 | 0.095 | - | +9.5% | 0.133 |
| 5 | 0.061 | 0.120 | - | +12.0% | 0.081 |
| 10 | 0.038 | 0.137 | 0.001 | +13.6% | 0.060 |
| 20 | 0.024 | 0.167 | 0.002 | +16.5% | 0.043 |
| 50 | 0.011 | 0.178 | 0.005 | +17.3% | 0.021 |
| 100 | 0.006 | 0.185 | 0.010 | +17.5% | 0.012 |
| 200 | 0.003 | 0.196 | 0.021 | +17.5% | 0.007 |
| 500 | 0.002 | 0.228 | 0.051 | +17.7% | 0.003 |
| 1000 | 0.001 | 0.271 | 0.103 | +16.8% | 0.002 |

At first glance, the results seem to be much more reasonable. The average precision of the training set is 21.8%, which is relatively consistent with the baseline that achieved 24.1%. The average recall is actually higher than the baseline across each dataset, and the F1 scores are higher for K=1. However, the net average recall stagnates around +25-26% for the training set as K increases, whereas the baseline achieved up to +71.9%. This finding is reflected in the F1 score of the model as K increases. Another interesting point, as in the previous model with embedding size 1000, the results from each dataset is not significantly different. This again implies that it did not learn from the training set, and predicts the same cases each time, independent of the inputted FCA case embedding.

This experiment, although unsuccessful in beating the baseline results, is meaningful because it suggests that the embeddings of size 250, although smaller, do model the corpus better and perform better for predictions compared to the embeddings of size 1000. This is likely due to the fact that these embeddings were trained for 100 epochs on the Doc2Vec algorithm, while the embeddings of size 1000 were only trained for 10 epochs. This resulted from the limitations in computational complexity and memory for the project. As a result, the problem with the above model is more likely to exist in the architecture and/or the inputted parameters, so a new model was designed accordingly.

### 4.4.2 CNN + MLP Model Results

The new architecture was developed due to the lack of success with the simple MLP architecture. This model was designed to include the SCC embedding information, where the previous only considered FCA embeddings and was prone to overfitting or memorizing the data. This new model included a Convolutional Neural Network (CNN) architecture, of which the details are outlined in the Methods section.

The model initially trained on 1,855 FCA decisions (which included the standard of review cases). The Doc2Vec embeddings of size 250 were used, since the size of 1000 was too expensive in terms of memory and computation, and following the findings from the previous model. The same data split was applied as before, with 70% devoted to the training set, 20% to the validation set, and 10% to the test set. The model trained for 11 epochs before training was stopped. At this point, the validation loss began to stagnate while the training loss kept decreasing. The same metrics as the baseline were used to analyze this model, and are presented in Table 15 below.

*Table 15: CNN+MLP performance on the training dataset of 1,335 FCA examples (embedding size 250).*

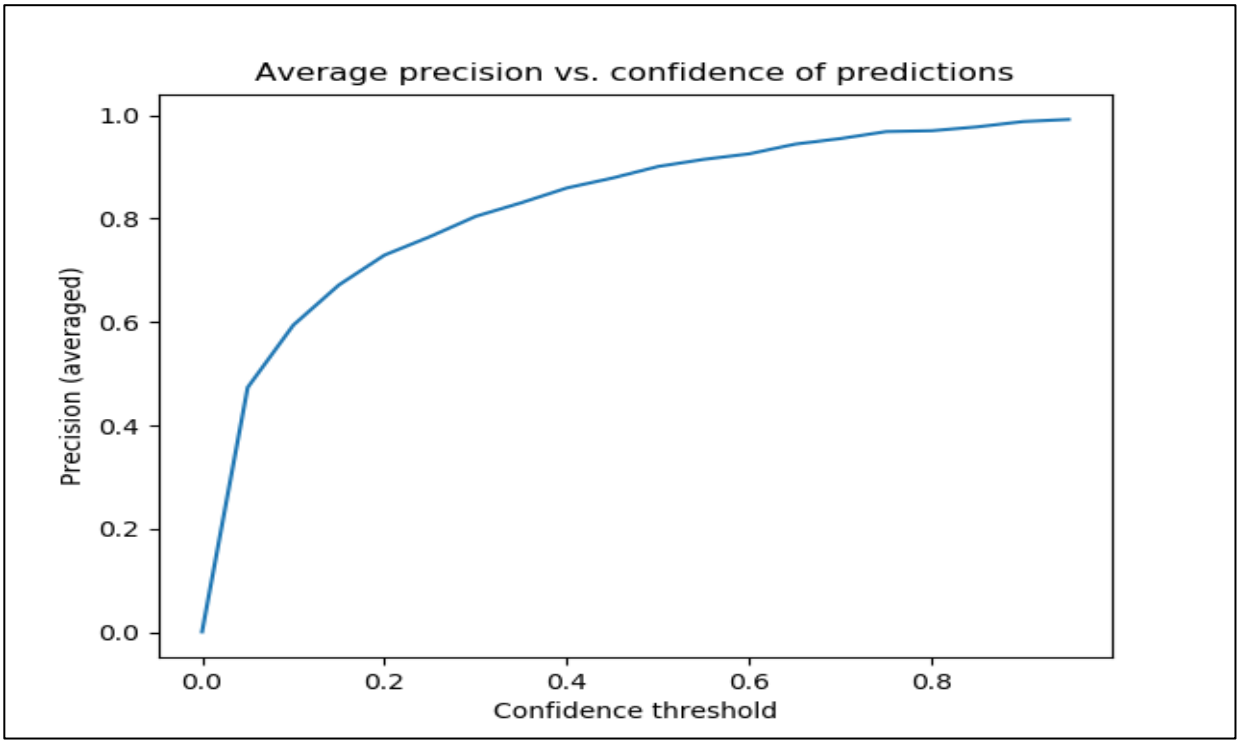
| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.866 | 0.432 | - | +43.2% | 0.576 |
| 5 | 0.481 | 0.816 | - | +82.6% | 0.605 |
| 10 | 0.316 | 0.919 | 0.001 | +91.8% | 0.470 |
| 20 | 0.186 | 0.975 | 0.002 | +97.3% | 0.312 |
| 50 | 0.081 | 0.997 | 0.005 | +99.2% | 0.150 |
| 100 | 0.041 | 1.000 | 0.010 | +99.0% | 0.079 |
| 200 | 0.020 | 1.000 | 0.021 | +97.9% | 0.039 |
| 500 | 0.008 | 1.000 | 0.051 | +94.9% | 0.016 |
| 1000 | 0.004 | 1.000 | 0.103 | +89.7% | 0.008 |

It is clear that the model is learning from the training set, and it is learning exceptionally well. The average precision when K=1 is 86.6%, and the average recall is 0.432. These are much better than the baseline. As K increases, the average recall quickly jumps to 81.6% at K=5, and 100% at K=100. At K=5, the F1score is 0.605, about 4x better than the baseline. This is a positive step, meaning that the model is not only precise, it is predicting the most relevant documents.

An advantage of this architecture is that the outputted predictions are processed through a sigmoid activation function before computing the loss (more details provided in Methods). The sigmoid activation function smooths out all possible inputs and maps them so that they are within the range of 0 and 1. As a result, if the model is trained properly, these predictions can be viewed as confidence scores. In theory, the most confident prediction will have the highest value, and vice-versa.

To evaluate how the model is learning, the average precision of the predictions could be plotted against the confidence of the predictions. This would help confirm if the model is too confident in its predictions, and potentially overfitting on the dataset. To do this, the confidence scores were divided into 20 buckets (equally of size 0.05), and the average precisions were plotted for each confidence threshold. Figure 11 below shows this plot on the training set.

*Figure 11: CNN+MLP training set average precision vs. confidence of predictions (embedding size 250).*




The plot confirms that the model has learned the training set (the average precision monotonically increases for all confidence thresholds). However, the plot is not linear, and appears

logarithmic in shape. At a confidence threshold of about 0.1, the average precision reaches about 60%. Ideally, the relationship between the average precision and confidence should be linear, meaning that the predictions are not heavily skewed towards either 0's or 1's. The implication of the above plot could mean that the training data is overfitting to training set, since for very low confidences the average precision is much better than the baseline.

To test a model's ability to generalize, the ultimate test is to evaluate the results on data it hasn't trained on, in this case the validation and test sets. These results help signify if the model has overfitted to the training set, or if it has generalized to examples beyond the training set. The same model was tested against the validation and test sets, and the results are presented in Table 16 and Table 17 below.

*Table 16: CNN+MLP performance on the validation dataset of 334 FCA examples (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.434 | 0.211 | - | +21.1% | 0.284 |
| 5 | 0.206 | 0.405 | - | +40.5% | 0.273 |
| 10 | 0.135 | 0.487 | 0.001 | +48.6% | 0.211 |
| 20 | 0.086 | 0.571 | 0.002 | +56.9% | 0.149 |
| 50 | 0.043 | 0.667 | 0.005 | +66.2% | 0.081 |
| 100 | 0.025 | 0.734 | 0.010 | +72.4% | 0.048 |
| 200 | 0.015 | 0.802 | 0.021 | +78.1% | 0.029 |
| 500 | 0.007 | 0.869 | 0.051 | +81.8% | 0.014 |
| 1000 | 0.004 | 0.906 | 0.103 | +80.3% | 0.008 |

*Table 17: CNN+MLP performance on the test dataset of 186 FCA examples (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.355 | 0.190 | - | +19.0% | 0.248 |
| 5 | 0.173 | 0.359 | - | +35.9% | 0.233 |
| 10 | 0.118 | 0.430 | 0.001 | +42.9% | 0.185 |
| 20 | 0.077 | 0.510 | 0.002 | +50.8% | 0.134 |
| 50 | 0.040 | 0.624 | 0.005 | +61.9% | 0.075 |
| 100 | 0.024 | 0.705 | 0.010 | +69.5% | 0.046 |
| 200 | 0.014 | 0.782 | 0.021 | +76.1% | 0.028 |
| 500 | 0.007 | 0.870 | 0.051 | +81.9% | 0.014 |
| 1000 | 0.004 | 0.910 | 0.103 | +80.7% | 0.008 |

The results for the validation and test sets are not as good as the training set, but are very promising. For the top prediction (K=1), the validation set achieved an average precision of 43.4% and the test set achieved 35.5%. The model is much more powerful than the baseline, as can be seen from the validation and test F1 scores of 0.273 and 0.233 for K=5. The discrepancy between these values can mean that the model began to memorize the validation set, and therefore the test results are more indicative of the true performance as a result.

The same plots for the average precision vs. the confidence of the prediction are gathered for the validation and test sets, and shown in Figure 12 and Figure 13 below respectively.

*Figure 12: CNN+MLP validation set average precision vs. confidence of predictions (embedding size 250).*

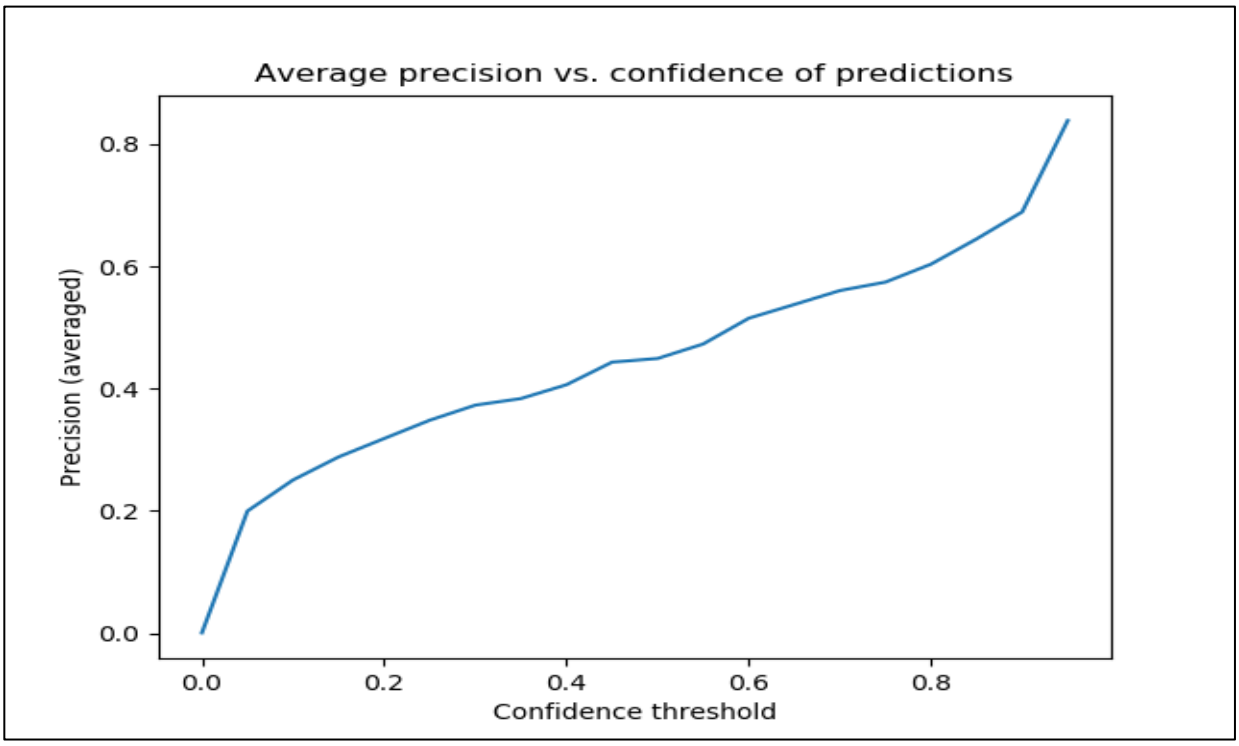
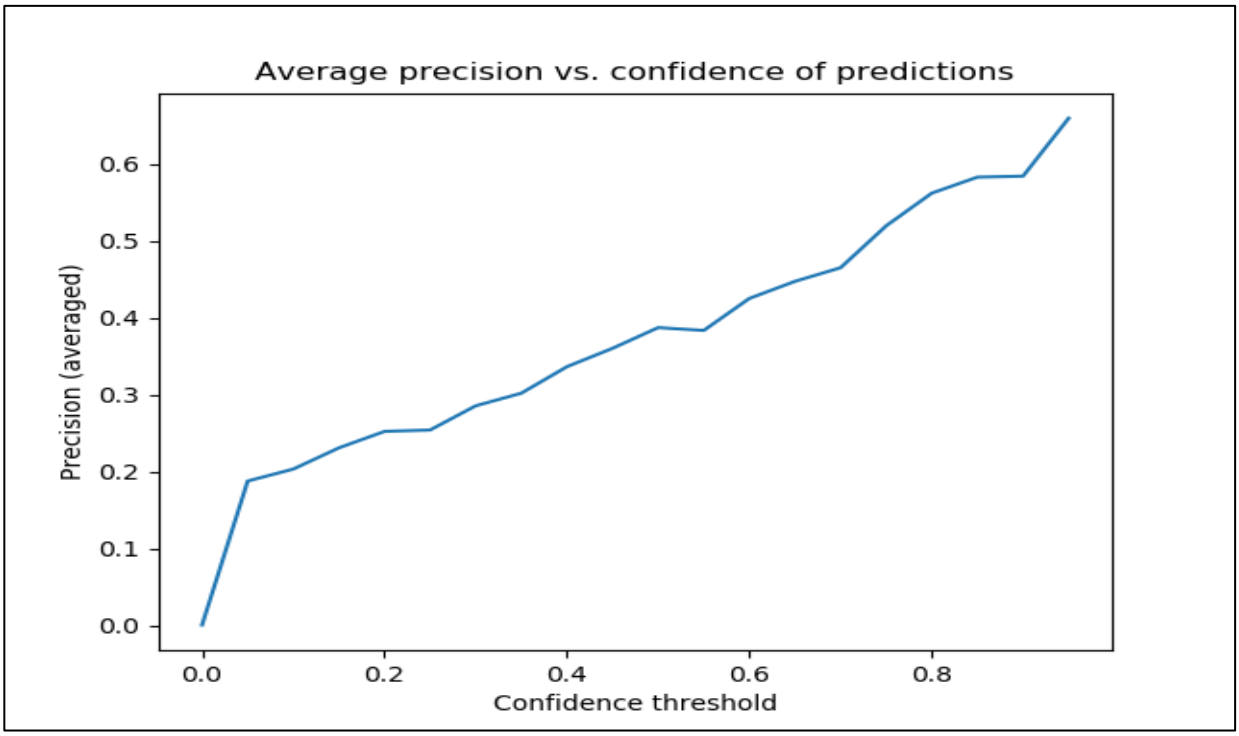


*Figure 13: CNN+MLP test set average precision vs. confidence of predictions (embedding size 250).*




The above plots display positive signs of generalization, as the relationship between the average precision and confidence is roughly proportional. It is also important to note that for confidence's over about 80%, the average test precision is roughly 60%. These are incredible results, and this precision is a significant improvement on the baseline. The precisions do start at about 20% for low confidences, and the predictions for low confidences can be improved.

The main criticism of this model lies in the discrepancy between the training results and validation and test results. The results imply that the model was overfitting and/or the learned representation of the data is not powerful enough to generalize to other examples like the test set. These results could also result from limitations in the dataset. The limited number of training examples could lead to overfitting, and since the data is so sparse, it could limit the potential of this model. The underlying assumption of this project is that the citations generated from legal professionals are correct and represent the data well, but as shown in the literature review, this may not be the case [37] [38] [39]. A detailed qualitative analysis of this model, its advantages and limitations are all described in the Discussion section.

The model was retrained on the dataset without the standard of review cases, to evaluate the effects on predictions. The results are displayed in Tables 18-20 below.

*Table 18: CNN+MLP performance on the training dataset of 1,143 FCA examples (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.886 | 0.417 | - | +41.7% | 0.567 |
| 5 | 0.515 | 0.819 | - | +81.9% | 0.632 |
| 10 | 0.340 | 0.923 | 0.001 | +92.2% | 0.497 |
| 20 | 0.200 | 0.975 | 0.002 | +97.3% | 0.332 |
| 50 | 0.086 | 0.996 | 0.005 | +99.1% | 0.158 |
| 100 | 0.044 | 1.000 | 0.010 | +99.0% | 0.084 |
| 200 | 0.022 | 1.000 | 0.021 | +97.9% | 0.043 |
| 500 | 0.008 | 1.000 | 0.051 | +94.9% | 0.016 |
| 1000 | 0.004 | 1.000 | 0.103 | +89.7% | 0.008 |

*Table 19: CNN+MLP performance on the validation dataset of 286 FCA examples (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.381 | 0.139 | - | +13.9% | 0.204 |
| 5 | 0.199 | 0.307 | - | +30.7% | 0.241 |
| 10 | 0.134 | 0.382 | 0.001 | +38.1% | 0.198 |
| 20 | 0.086 | 0.462 | 0.002 | +46.0% | 0.145 |
| 50 | 0.046 | 0.596 | 0.005 | +59.1% | 0.085 |
| 100 | 0.028 | 0.677 | 0.010 | +66.1% | 0.054 |
| 200 | 0.016 | 0.763 | 0.021 | +74.2% | 0.031 |
| 500 | 0.007 | 0.854 | 0.051 | +80.3% | 0.014 |
| 1000 | 0.004 | 0.893 | 0.103 | +79.0% | 0.008 |

*Table 20: CNN+MLP performance on the test dataset of 159 FCA examples (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.314 | 0.124 | - | +12.4% | 0.178 |
| 5 | 0.172 | 0.282 | - | +28.2% | 0.214 |
| 10 | 0.120 | 0.370 | 0.001 | +36.9% | 0.181 |
| 20 | 0.080 | 0.455 | 0.002 | +45.3% | 0.136 |
| 50 | 0.042 | 0.577 | 0.005 | +57.2% | 0.078 |
| 100 | 0.025 | 0.656 | 0.010 | +64.6% | 0.048 |
| 200 | 0.014 | 0.730 | 0.021 | +70.9% | 0.027 |
| 500 | 0.007 | 0.822 | 0.051 | +77.1% | 0.014 |
| 1000 | 0.003 | 0.851 | 0.103 | +74.8% | 0.006 |

The training set results are roughly similar, but as expected, the validation and test results are slightly worse. This confirms the suspicion that certain cases are being predicted more often than others, which directly results from the dataset. By removing the standard of review cases, the validation and test average precisions decrease by about 4.5% and 5.5% respectively, across all values of K. The average recall also falls, and this is reflected in the F1 scores.

These scores are more representative of the model's predictive capabilities, since previous investigations showed that the standard of review cases do not contribute any significant information. With that said, the model is still notably better than the baseline, with an improved F1 score of 0.214 on the test set from the baseline of 0.155 when K=5. This is promising, and means that the model did succeed in learning the intricacies of the language used in the legal texts.

It is important to note that time is not accounted for when the model is training. As shown with the baseline model, the results in theory should improve when time is factored into the predictions. This means that the model is truly more powerful than presented above.

Indeed, the test results do improve when time is factored into the predictions. The results are presented in Table 21.

*Table 21: CNN+MLP performance on the test dataset of 159 FCA examples with release year factored in (embedding size 250).*

| K | Avg. Precision@K | Avg. Recall | Expected Recall | Net Avg. Recall | F1 score |
|---|---|---|---|---|---|
| 1 | 0.327 | 0.128 | - | +12.8% | 0.184 |
| 5 | 0.174 | 0.289 | - | +28.9% | 0.217 |
| 10 | 0.116 | 0.363 | 0.001 | +36.2% | 0.176 |
| 20 | 0.080 | 0.461 | 0.002 | +45.9% | 0.137 |
| 50 | 0.042 | 0.570 | 0.005 | +56.5% | 0.078 |
| 100 | 0.025 | 0.660 | 0.010 | +65.0% | 0.048 |
| 200 | 0.014 | 0.735 | 0.021 | +72.4% | 0.028 |
| 500 | 0.006 | 0.806 | 0.051 | +75.5% | 0.013 |
| 1000 | 0.003 | 0.834 | 0.103 | +73.1% | 0.007 |

The average precision and recall both increase for K=1 and K=5. These improvements are reflected in the F1 score, which improved slightly from 0.214 to 0.217 when K=5. The findings are consistent with the improvements shown in the baseline model, when time was factored in.

# 5 Discussion

This section is devoted to qualitatively analyzing the results from all predictive models presented above, identifying key findings and limitations of each.

## 5.1 Baseline Predictor using Tf-idf Similarity

A predictive model of SCC citations was developed using tf-idf to generate document embeddings, and cosine similarity to find similar decisions. This model was developed to serve as a baseline comparator for the newly developed models, since in theory, any reasonable predictor should be at least as good as this basic one.

The baseline predictive model was applied to the raw dataset, the cleaned dataset, and the cleaned dataset with standard of review cases removed. The quantitative results of these experiments are presented in the previous section. The F1 score of the bias-free model (preprocessed texts and standard of review cases removed) is 0.115 for K=1 and 0.155 for K=2. These numbers intuitively seem high. A more thorough investigation is necessary to truly

understand why this model is performing reasonably well, and specifically what the tf-idf similarity is interpreting in each prediction.

To get a better picture of what the model is predicting, 5 correct and incorrect predictions were chosen at random. Each case was read to try to understand (holistically) where the model went wrong, and what it interpreted when the correct prediction was made. For simplicity, only the top predictions (K=1) were considered for this analysis, since in theory these are the predictions that the model is most confident about.

The model correctly predicted the citation 2005 SCC 57 for the decision 2013 FCA 168. The respondent of both cases is "The Minister of Citizenship and Immigration", which is a positive sign. When looking a bit more closely, the FCA decision refers to the SCC decision three times throughout the document. For instance, in the unformatted text for 2013 FCA 168, paragraph 5 reads:

> *The Judge found as a fact that Mr. Zhang failed to provide a credible individualized plan for mitigating the excessive demand on social services in Canada (per Hilewitz v. Canada (Minister of Citizenship and Immigration); De Jong v. Canada (Minister of Citizenship and Immigration), 2005 SCC 57, [2005] 2 S.C.R. 706).* [62]

This may have swayed the baseline model if the raw texts were used, as it may have picked up on unique words such as "Hilewitz" or "De Jong" that would appear in both the FCA and SCC decisions. In the preprocessed texts, the entire reference at the end of the above paragraph is removed, along with the other references to the citation in the rest of the document. Upon further investigation, the sentence preceding the citation is truly what is intriguing. The author of the decision did not directly quote 2005 SCC 57, but there are specific phrases common to both decisions such as "excessive demands on social services" and "individualized plan". This is an interesting finding because even though direct quotes were removed, tf-idf was able to read into this subtlety in the author's use of language. This means that reading the SCC case had a significant effect on the author, and the author used similar language in the FCA decision to really drive the main point of the argument. This could be a reason why the baseline tf-idf predictor was so effective in this context.

This same effect was observed in each of the other 4 randomly chosen correct predictions. The baseline model is able to extrapolate some of the subject matter from the decisions, even without knowing the context or semantic meaning of the phrases. For instance the model correctly

predicted that 2015 FCA 286 cited 2008 SCC 61. Similar to the above example, the two documents used similar language and both contain key phrases such as “balance of probabilities” and “self-evident to try to obtain the invention” [63], although they are not directly quoted.

The correct predictions tell an interesting story, but these account for only about one in every five predictions, as the average precision@K is 20.1% for K=1 in the bias-free model. It is more interesting to analyze the incorrect predictions, to truly understand the shortcomings of the model.

The model predicts that 2012 FCA 199 will cite 2014 SCC 29, which of course is impossible since the SCC decision did not exist at the time of the citation. So why did the model predict this citation? The facts of the cases are extremely similar. They both mention the “St. Lawrence River”, “Baie Comeau”, the “Marine Liability Act”, and the “Charts and Nautical Publications Regulations” [64] [65]. In fact, the SCC case actually cites the FCA case, as it was an appeal from a previous judgement. In some sort of consolation, the model was not wrong, but the citation occurred in the other direction. The obvious solution to this would be incorporating time within the predictions.

Similarly, the model incorrectly predicted a citation between 2015 FCA 186 and 2010 SCC 2, but the prediction seems reasonably plausible. The FCA case is a consolidated appeal about Nuclear power plant projects that were not completed properly, while the SCC case is about mining in British Columbia [66] [67]. One of the appellants in the FCA case is the “Minister of Fisheries and Oceans”, who is the respondent in the SCC case. Both cases refer to the “Canadian Environment Assessment Act”, and phrases like “energy” are used. In this way, one could argue that the subject matter is similar and the model has made a reasonable prediction. However, the law is very different, and the model failed to identify subtleties in cases like “nuclear” vs. “mining”. The model was anchored on the other terms used, as they appeared more often.

There were other examples where the model’s predictions were severely wrong as well. The model predicted that 2008 FCA 215 would cite 2001 SCC 68, two very different cases. The FCA case deals with a class action filed by the respondents who wanted refunds on immigration visas, while the SCC case is about a noise and pollution complaint about practices in the city of Toronto [68] [69]. This is a scenario demonstrates when tf-idf failed miserably. There are certain words that the predictor has noticed appears in both cases, but unfortunately, they appear in very different contexts. For instance the FCA case mentions the “Immigration and Refugee Protection Regulations” and the SCC case mentions the “Environmental Protection Act”. It is clear that

"Protection" is the common word, but the reference legislations are completely different. The only similar component between these two documents is that they both deal with a "class action", and these two words are referenced frequently in both texts.

To see if there are any larger trends in the predictions of the model, a holistic analysis of the predictions was gathered and averaged. Each prediction from the model was tracked, and the confidences (positions) of each prediction were averaged. This could be used to see what cases are most often predicted on average, and what cases are least often predicted on average. The 10 most and least common predictions by the baseline model, along with the true number of citations to each case, is presented in Table 22 below[4].

*Table 22: List of the 10 most and least common predictions by the baseline model (averaged on the dataset).*

| Most confident predictions | Number of citations | Least confident predictions | Number of citations |
|---|---|---|---|
| 1999 CanLII 665 | 12 | 1928 CanLII 41 | 0 |
| 1923 CanLII 45 | 0 | 1969 CanLII 94 | 0 |
| 1907 CanLII 104 | 0 | 1967 CanLII 5 | 0 |
| 1971 CanLII 305 | 0 | 1951 CanLII 41 | 1 |
| 1991 CanLII 73 | 0 | 1919 CanLII 37 | 0 |
| 1997 CanLII 17020 | 5 | 1967 CanLII 62 | 0 |
| 1982 CanLII 42 | 0 | 1989 CanLII 55 | 0 |
| 2017 SCC 55 | 3 | 1934 CanLII 55 | 0 |
| 1992 CanLII 45 | 0 | 1934 CanLII 49 | 0 |
| 1991 CanLII 76 | 0 | 2012 SCC 6 | 0 |

The top 10 most cited predictions do not look much different than the top 10 least cited predictions. Seven of the most cited predictions do not actually have any true citations to them. Two important notes can be made from these results. Firstly, it confirms that the model has no

[4] Note that each of the presented cases are SCC citations, some with a different naming convention for the citations. The naming provided is consistent with the naming convention of CanLII (Canadian Legal Information Institute), who graciously provided the SCC texts used in this project.

bias resulting from the citation space of the dataset, and predicts cases that it believes will have the highest likelihood of being cited. Another interesting point is that this means the model could be improved substantially. The model is not learning, and the number of citations to each citation should in theory affect predictions. For instance, assuming the dataset is representative of the way in which predictions should be cited in practice, then it would be in the best interest of the model to consider the amount of times a predicted citations is actually cited in the dataset.

So in short, the baseline predictor works reasonably well, but fails to identify the subtleties and intricacies of the English language. It is also naïve in the sense that it does not consider the use of citations in the dataset, which in theory could make the model more powerful. These factors significantly limit the capabilities of the model, suggesting that the learning-based methods developed and presented in this research should learn to take advantage of this. The trade-off, however, exists between naïve predictions and overfitting on the given data. This is discussed thoroughly in the analysis of the learning-based methods below.

## 5.2 MLP Predictor

As discussed in the Results section, the originally designed MLP models did not perform as expected. There was not a significant difference between the training and test results, which is troubling. This signifies that the model was not properly training, and thus not properly learning from the data.

As initially suspected, a deeper investigation into the results revealed that the models were almost always predicting the same citations. With the trained Doc2Vec embeddings of size 250, the model was predicting one of three cases in the test set, for K=1. These cases are 2002 SCC 33, 2008 SCC 9, and 2013 SCC 36. Interestingly, these three cases are all standard of review cases, and comprise about 17% of all citations in the initial dataset. This explains why the average recall and precision is so high for K=1, and why the results do not follow as K increases.

There are several key takeaways from the experiments with the MLP model. The first is that although they are larger and should in theory carry more information, the learned Doc2Vec embeddings of size 1000 carry less information than the size 250. This probably follows from the length of training, as the 1000 sized embeddings were only trained for 10 epochs and the 250 sized were trained for 100 epochs. This was chosen because of the resource constraints in the project,

and the larger embeddings took much longer to train per epoch. There is much room for experimentation in future research projects, but at least for now, the embeddings of size 250 were used for the rest of the project.

Another important finding is that the current architecture, although flawed in many ways, at least learned to predict the same three cases. These predictions were made without any information about the SCC corpus, which intuitively should improve the predictive capability of the model. It should be noted that the model did in fact outperform the baseline model, as naïve as it was. This was a positive experiment for these reasons, which helped inform the models that were built next.

## 5.3 CNN + MLP Predictor

As shown in the Results section, this model was much better at learning the training set as was shown in the test results. The model is capable of significantly outperforming the baseline, and another advantage of the model over the baseline is that it gives a confidence of the prediction. For high predictions, the model showed high average precision on the dataset. With time and the standard of review cases factored in, the F1 score for K=5 was 0.217, a significant improvement on the baseline of 0.164. These are all positive signs which indicate that the model is learning the intricacies of the language used in the legal texts, which the baseline model neglects.

A holistic view of the predictions is shown in the following table, which shows the most and least confident predictions (averaged over the dataset), along with the number of true citations.

*Table 23: List of the 10 most and least common predictions by the proposed model (on the test set).*

| Most confident predictions | Number of citations | Least confident predictions | Number of citations |
|---|---|---|---|
| 1999 CanLII 699 | 102 | 1933 CanLII 51 | 0 |
| 2008 SCC 51 | 19 | 1936 CanLII 16 | 0 |
| 2014 SCC 53 | 40 | 1990 CanLII 97 | 2 |
| 2005 SCC 54 | 122 | 1955 CanLII 16 | 0 |
| 2002 SCC 42 | 47 | 1992 CanLII 32 | 0 |
| 1998 CanLII 837 | 91 | 1997 CanLII 389 | 0 |
| 2013 SCC 37 | 0 | 2009 SCC 13 | 0 |
| 2011 SCC 61 | 119 | 1996 CanLII 177 | 0 |
| 2003 SCC 19 | 55 | 1923 CanLII 48 | 0 |
| 2008 SCC 10 | 0 | 1959 CanLII 78 | 0 |

The results from Table 23 are interesting for a few reasons. It is quite apparent that the model has learned which cases are cited more often than others, which the baseline model did not learn. Most of the top 10 predicted cases on average have a significant amount of true citations in the dataset, and the least predicted 10 cases are almost never cited in the dataset. Is this truly a desirable quality for a predictive model? Does this mean that the model is overfitting, and will never predict certain cases? More often than not, it will predict cases that are cited frequently in the dataset. The validity of the citation dataset is an underlying assumption of this research project, and the model is able to learn it better than the baseline. As shown in Table 23, two of the top predicted citations do not have any citations in the dataset, so the model is not overfitting. This means that the model is still considering textual information, and not learning solely from the citation information.

It also appears that the model, although not given any context of time, is consistently predicting newer cases and disregarding old cases. Some of the least predicted cases are from the 1930's, while most of the top predicted cases are newer than the 2000's. This may imply that the older cases use much different language, and that the model is interpreting this. Alternatively, this could mean that the model is learning how legal professionals read and cite past cases, where typically older cases are seldom referenced.

It should also be noted that this model was able to learn the training data extremely well, on a dual-core personal computer. Of course, there is much room for optimization in this task, but the results from this model are extremely powerful and consistent.

There is a possibility, however, that the underlying assumption is wrong. This would mean that the dataset is not representative of relevancy in citations, and the collected data is not sufficient for this learning task. As the size of the dataset increases in theory, the citations should be more representative of relevancy. For this project, the number of training examples (downloaded FCA cases) were one tenth of the size of the prediction space (downloaded SCC cases). This of course could mean that the collected data is not sufficient for learning.

To address this, data augmentation techniques can be applied to create synthetic data, more training examples can be gathered from the FCA, or predictions from the SCC can be more selectively chosen. In future research, these are all options that should be explored to ensure that the data is representative of what is being predicted. The deep learning model proposed in this paper is capable of learning these representations, better than the simple baseline model.

# 6 Conclusion

The purpose of this research is to ultimately show that AI is capable of learning the language of legal texts. To motivate this, the task of predicting citations using only legal texts was chosen. The developed AI models outperformed traditional methods that do not consider language use and context, illustrating the ultimate hypothesis that AI can learn the language of the law.

The decisions from two Canadian courts were used. Specifically, 3,360 decisions were downloaded from the Canadian Federal Court of Appeal (FCA), and 11,354 decisions were downloaded from the Supreme Court of Canada (SCC). These two courts were chosen because the FCA frequently cites decisions from the SCC, a desirable trait for this task. This meant that the citation space would allow for more robust predictions.

The downloaded predictions were cleaned and preprocessed, and ultimately the final dataset comprised of 1,588 FCA decisions and 9,744 SCC decisions. The preprocessing involved reformatting the originally downloaded data, removing all possible forms of bias without removing too much information, and building a truth table that housed all citation information.

This project proposes a new architecture for citation prediction tasks using texts. The proposed method uses Doc2Vec embeddings of the FCA and SCC corpora as inputs. The proposed deep learning model architecture involves 2 convolutional layers, and 3 linear layers. As discussed above, the proposed model significantly outperforms the baseline model. The proposed model is extremely efficient, and can be trained on most personal computers.

These simple architectures were able to outperform the traditional baseline model, but they are not perfect. There is room for much more optimization and experimentation with the architectures and learning techniques. Also, the research project involves many assumptions that may have inadvertently affected the results of this experiment. For instance, by using citation data as a proxy for relevant decisions, there is an underlying assumption that legal professionals cite cases optimally. This implies reading through hundreds if not thousands of past decisions, and citing the most relevant ones. Also, as discussed in the Literature Review, research has uncovered many different biases within the use of citations [37] [38] [39], which may have impacted the integrity of the truth table, and impacted the results of this set of experiments.

Ultimately, the results from this research help demonstrate the initial hypothesis of the research project, that AI can learn the language of law. This research will mark an important first step in the development of legal research, and lays the groundwork for much more experimentation

in the field. In addition, it confirmed that there are various real applications to this research, including developing a recommendation system for citations that does not require any input or query from the user.

# References


[1] B. Alarie, A. Niblett and A. Yoon, "Regulation by Machine," 1 December 2016. [Online]. Available: https://ssrn.com/abstract=2878950. [Accessed 1 October 2019].

[2] Blue J Legal, "Tax Foresight," [Online]. Available: https://www.bluejlegal.com/tax-usforesight-ca.

[3] WestlawNext Canada, "KeyCite Canada," [Online]. Available: https://www.westlawnextcanada.com/westlaw-advantages/keycite-canada/.

[4] ROSS Intelligence, "Features," [Online]. Available: https://www.rossintelligence.com/features.html.

[5] Bloomberg Law, "Essential, AI-powered litigation resources.," [Online]. Available: https://pro.bloomberglaw.com/ai-analytics/.

[6] NexisLexis, "Lexis Advance® Quicklaw®," [Online]. Available: https://www.lexisnexis.ca/en-ca/products/lexis-advance-quicklaw-overview.page.

[7] Wikipedia, "Natural language processing," [Online]. Available: https://en.wikipedia.org/wiki/Natural_language_processing.

[8] P. Bharadwaj and Z. Shao, "Fake News Detection with Semantic Features and Text Mining," International Journal on Natural Language Computing (IJNLC) Vol.8, No.3, June 2019, 24 July 2019. [Online]. Available: https://ssrn.com/abstract=3425828. [Accessed 29 November 2019].

[9] S. Mehtab and J. Sen, "A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing," 12 December 2019. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3502624. [Accessed 6 January 2020].

[10] Manning, Raghavan and Schütze, "Scoring, term weighting and the vector space model," in *Introduction to Information Retrieval*, © 2008 Cambridge University Press, 2009.

[11] O. Shahmirzadi, A. Lugowski and K. Younge, "Text Similarity in Vector Space Models: A Comparative Study," 15 September 2018. [Online]. Available: https://ssrn.com/abstract=3259971. [Accessed 7 January 2020].

[12] TensorFlow, "Word Embeddings," [Online]. Available: https://www.tensorflow.org/tutorials/text/word_embeddings.

[13] T. Mikolov, K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space," [Online]. Available: https://arxiv.org/abs/1301.3781v3.

[14] FastText, "Library for efficient text classification and representation learning," [Online]. Available: https://fasttext.cc/.

[15] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettlemoyer, "Deep contextualized word representations," 27 October 2017. [Online]. Available: https://arxiv.org/abs/1802.05365. [Accessed 6 January 2020].

[16] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," October 2014. [Online]. Available: https://nlp.stanford.edu/pubs/glove.pdf. [Accessed 6 January 2020].

[17] K. Grzegorczyk, "Vector representations of text data in deep learning," Doctoral Dissertation, 4 July 2018. [Online]. Available: https://arxiv.org/abs/1901.01695. [Accessed 6 January 2020].

[18] S. Sehrawat, "Learning Word Embeddings from 10-K Filings Using PyTorch," 5 September 2019. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3480902. [Accessed 6 January 2020].

[19] W. Ali, J. Kumar, J. Lu and Z. Xu, "A New Corpus for Low-Resourced Sindhi Language with Word Embeddings," 28 November 2019. [Online]. Available: https://arxiv.org/abs/1911.12579. [Accessed 6 January 2020].

[20] J. Steier, "A Characterization of Dante Alighieri: An NLP approach to The Divine Comedy," 17 August 2019. [Online]. Available: https://ssrn.com/abstract=3438785. [Accessed 6 January 2020].

[21] C. Saedi and M. Dras, "Siamese Networks for Large-Scale Author Identification," 23 December 2019. [Online]. Available: https://arxiv.org/abs/1912.10616. [Accessed 6 January 2020].

[22] A. G. Dobrakowski, A. Mykowiecka, M. Marciniak, W. Jaworski and P. Biecek, "Clustering of Medical Free-Text Records Based on Word Embeddings," 3 July 2019. [Online]. Available: https://arxiv.org/abs/1907.04152. [Accessed 6 January 2020].

[23] B. Danet, "Language in the Legal Process," Law & Society Review Vol. 14, No. 3, Contemporary Issues in Law and Social Science (Spring, 1980), pp. 445-564 (120 pages), 1980. [Online]. Available: https://www-jstor-org.myaccess.library.utoronto.ca/stable/3053192. [Accessed 12 December 2019].

[24] P. Goodrich, "Law and Language: An Historical and Critical Introduction," Journal of Law and Society Vol. 11, No. 2 (Summer, 1984), pp. 173-206 (34 pages), 1984. [Online]. Available: https://www-jstor-org.myaccess.library.utoronto.ca/stable/1410039. [Accessed 13 December 2019].

[25] N. Fairclough, "Discourse and text: linguistic and intertextual analysis within discourse analysis," Discourse & Society Vol. 3, No. 2 (1992), pp. 193-217 (25 pages), 1992. [Online]. Available: https://www-jstor-org.myaccess.library.utoronto.ca/stable/42887786. [Accessed 13 December 2019].

[26] F. Fagan, "From Policy Confusion to Doctrinal Clarity: Successor Liability from the Prospective of Big Data," Virginia Law & Business Review, Vol. 9, No. 3, p. 391, 2015,

18 November 2014. [Online]. Available: https://ssrn.com/abstract=2488819. [Accessed 5 January 2020].

[27] J. R. Macey and J. Mitts, "Finding Order in the Morass: The Three Real Justifications for Piercing the Corporate Veil," Cornell Law Review, Forthcoming; Yale Law & Economics Research Paper No. 488, 18 February 2014. [Online]. Available: https://ssrn.com/abstract=2398033. [Accessed 6 January 2020].

[28] L.-R. D. Kosnik, "Determinants of contract completeness: An environmental regulatory application," 2014. [Online]. Available: https://www.semanticscholar.org/paper/Determinants-of-contract-completeness%3A-An-Kosnik/bc78f7d5ccabf81621135f1c33f1190ec29d03fe. [Accessed 5 January 2020].

[29] J. Nay, "Gov2Vec: Learning Distributed Representations of Institutions and Their Legal Text," Proceedings of 2016 Empirical Methods in Natural Language Processing Workshop on NLP and Computational Social Science, 49–54, Association for Computational Linguistics., 5 November 2016. [Online]. Available: https://ssrn.com/abstract=3087278. [Accessed 7 January 2020].

[30] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," 16 May 2014. [Online]. Available: https://arxiv.org/abs/1405.4053. [Accessed 22 March 2020].

[31] E. Ash and D. L. Chen, "Case Vectors: Spatial Representations of the Law Using Document Embeddings," 8 June 2018. [Online]. Available: https://ssrn.com/abstract=3204926. [Accessed 6 January 2020].

[32] M. J. Bommarito, D. M. Katz and E. M. Detterman, "LexNLP: Natural Language Processing and Information Extraction For Legal and Regulatory Texts," 16 June 2018. [Online]. Available: https://ssrn.com/abstract=3192101. [Accessed 6 January 2020].

[33] L. Robaldo, S. Villata, A. Wyner and M. Grabmair, "Introduction for artificial intelligence and law: special issue “natural language processing for legal texts”," Artificial Intelligence and Law, 13 April 2019. [Online]. Available: https://link.springer.com/article/10.1007/s10506-019-09251-2. [Accessed 6 January 2020].

[34] Government of Canada, "Where our legal system comes from," 16 October 2017. [Online]. Available: https://www.justice.gc.ca/eng/csj-sjc/just/03.html.

[35] R. A. Posner, "The Theory and Practice of Citations Analysis, with Special Reference to Law and Economics," University of Chicago Law School, John M. Olin Law & Economics Working Paper No. 83, September 1999. [Online]. Available: https://ssrn.com/abstract=179655. [Accessed 26 November 2019].

[36] T. S. Clark and B. Lauderdale, "Locating Supreme Court Opinions in Doctrine Space," CELS 2009 4th Annual Conference on Empirical Legal Studies Paper, 4 August 2009. [Online]. Available: https://ssrn.com/abstract=1444031. [Accessed 27 November 2019].

[37] Z. Ződi, "Analysis of Citation Patterns of Hungarian Judicial Decisions: Is Hungarian Legal System Really Converging to Case Laws? Results of a Computer Based Citation

Analysis of Hungarian Judicial Decisions," 24 February 2014. [Online]. Available: https://ssrn.com/abstract=2410070. [Accessed 17 November 2019].

[38] S. J. Choi and G. M. Gulati, "Bias in Judicial Citations: A New Window into the Behavior of Judges?," 2 July 2006. [Online]. Available: https://ssrn.com/abstract=913663. [Accessed 19 December 2019].

[39] T. A. Smith, "The Web of Law," San Diego Legal Studies Research Paper No. 06-11, 5 January 2005. [Online]. Available: https://ssrn.com/abstract=642863. [Accessed 27 November 2019].

[40] M. Derlén and J. Lindholm, "Measuring Centrality in Legal Citation Networks – A Case Study of the HITS and PageRank Algorithms," 7 January 2017. [Online]. Available: https://ssrn.com/abstract=2910926. [Accessed 26 November 2019].

[41] J. S. Miller, "Which Supreme Court Cases Influenced Recent Supreme Court IP Decisions? A Citation Study," UCLA Journal of Law & Technology, Fall 2017; University of Georgia School of Law Legal Studies Research Paper No. 2017-22, 7 August 2017. [Online]. Available: https://ssrn.com/abstract=3012262. [Accessed 26 November 2019].

[42] J. G. Conrad and D. P. Dabney, "Automatic Recognition of Distinguishing Negative Indirect History Language in Judicial Opinions," Proceedings of the tenth international conference on Information and knowledge management - CIKM01, 2001. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.156.1571. [Accessed 6 January 2020].

[43] M. A. Livermore, F. Dadgostari, M. Guim, P. Beling and D. Rockmore, "Law Search as Prediction," 5 November 2018. [Online]. Available: https://ssrn.com/abstract=3278398. [Accessed 26 November 2019].

[44] M. Medvedeva, M. Vols and M. Wieling, "Using machine learning to predict decisions of the European Court of Human Rights," Artificial Intelligence and Law, 26 June 2019. [Online]. Available: https://doi.org/10.1007/s10506-019-09255-y. [Accessed 26 November 2019].

[45] Government of Canada, "Reproduction of Federal Law Order," 08 01 1997. [Online]. Available: https://laws-lois.justice.gc.ca/eng/regulations/si-97-5/page-1.html. [Accessed 23 March 2020].

[46] "tika-python," [Online]. Available: https://github.com/chrismattmann/tika-python.

[47] C. D. Manning, P. Raghavan and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.

[48] "sklearn.feature_extraction.text.TfidfVectorizer¶," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html. [Accessed 22 March 2020].

[49] "sklearn.metrics.pairwise.cosine_similarity," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.html. [Accessed 22 March 2020].

[50] gensim, "Doc2vec paragraph embeddings," [Online]. Available: https://radimrehurek.com/gensim/models/doc2vec.html. [Accessed 22 March 2020].

[51] J. H. Lau and T. Baldwin, "An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation," August 2016. [Online]. Available: https://www.aclweb.org/anthology/W16-1609/. [Accessed 22 March 2020].

[52] "Multilayer perceptron," [Online]. Available: https://en.wikipedia.org/wiki/Multilayer_perceptron. [Accessed 22 March 2020].

[53] "Universal Approximation Theorem," [Online]. Available: https://en.wikipedia.org/wiki/Universal_approximation_theorem. [Accessed 22 March 2020].

[54] "PyTorch," [Online]. Available: https://pytorch.org/. [Accessed 22 March 2020].

[55] "Pareto Principle," [Online]. Available: https://en.wikipedia.org/wiki/Pareto_principle. [Accessed 22 March 2020].

[56] "Convolutional neural network," [Online]. Available: https://en.wikipedia.org/wiki/Convolutional_neural_network. [Accessed 22 March 2020].

[57] "Housen v. Nikolaisen, 2002 SCC 33 (CanLII), [2002] 2 SCR 235," 28 March 2002. [Online]. Available: http://canlii.ca/t/51tl. [Accessed 22 March 2020].

[58] "Dunsmuir v. New Brunswick, 2008 SCC 9 (CanLII), [2008] 1 SCR 190," 7 March 2008. [Online]. Available: http://canlii.ca/t/1vxsm. [Accessed 22 March 2020].

[59] "Agraira v. Canada (Public Safety and Emergency Preparedness), 2013 SCC 36 (CanLII), [2013] 2 SCR 559," 20 June 2013. [Online]. Available: http://canlii.ca/t/fz8c4. [Accessed 22 March 2020].

[60] "The Standard of Review," [Online]. Available: https://www.scc-csc.ca/case-dossier/cb/2019/37748-37896-37897-eng.pdf. [Accessed 22 March 2020].

[61] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for IDF," *Journal of Documentation,* vol. 60, no. 5, pp. 503-520, 2004.

[62] "Zhang v. Canada (Citizenship and Immigration), 2013 FCA 168 (CanLII)," 27 June 2013. [Online]. Available: http://canlii.ca/t/fzhvl. [Accessed 22 March 2020].

[63] "Eli Lilly Canada Inc. v. Mylan Pharmaceuticals ULC, 2015 FCA 286 (CanLII)," 15 December 2015. [Online]. Available: http://canlii.ca/t/gmlct. [Accessed 22 March 2020].

[64] "Peracomo Inc. v. Société Telus Communications, 2012 FCA 199 (CanLII)," 29 June 2012. [Online]. Available: http://canlii.ca/t/frx79. [Accessed 22 March 2020].

[65] "Peracomo Inc. v. TELUS Communications Co., 2014 SCC 29 (CanLII), [2014] 1 SCR 621," 23 April 2014. [Online]. Available: http://canlii.ca/t/g6ldv. [Accessed 22 March 2020].

[66] "Ontario Power Generation Inc. v. Greenpeace Canada, 2015 FCA 186 (CanLII)," 10 September 2015. [Online]. Available: http://canlii.ca/t/gl4hl. [Accessed 22 March 2020].

[67] "MiningWatch Canada v. Canada (Fisheries and Oceans), 2010 SCC 2 (CanLII), [2010] 1 SCR 6," 21 January 2010. [Online]. Available: http://canlii.ca/t/27jmr. [Accessed 22 March 2020].

[68] "Hinton v. Canada (Minister of Citizenship and Immigration), 2008 FCA 215 (CanLII), [2009] 1 FCR 476," 13 June 2008. [Online]. Available: http://canlii.ca/t/1z1cd. [Accessed 22 March 2020].

[69] "Hollick v. Toronto (City), 2001 SCC 68 (CanLII), [2001] 3 SCR 158," 18 October 2001. [Online]. Available: http://canlii.ca/t/51zq. [Accessed 22 March 2020].