British Columbia Negligence Decisions Analysis

UBC Master of Data Science - Computational Linguistics

By: Ilana Zimmerman, Niki Hajmoshir, Ravi Gill

Capstone project supervisor: Julian Brooke

Client: Peter A. Allard School of Law, Lachlan Caunt

Acknowledgments

To our supervisor—Dr. Julian Brooke—we would have not been able to do this without you.

Thank you for your patient guidance, enthusiastic encouragement, and useful critiques of this research work.

To our capstone client — Lachlan Caunt — who brought this amazing project to us and did not hesitate to help and give directions.

Your willingness to give your time so generously has been very much appreciated.

British Columbia Judicial Decisions Analysis Directory

Summary

In recent years, some members of the legal community specializing in negligence cases have begun to suspect that the overall damage payouts have been steadily increasing in British Columbia. Negligence can be described as "an area of tort law that deals with the breach of duty to take care and it involves harm caused by carelessness, not intentional harm."[1]. Negligence cases may involve damage payments to the plaintiff broken down into specific sub-categories based on the evidence presented and the precedence set for similar cases that have occurred in the past. They also may find the plaintiff partially liable for the damages which are known as contributory negligence. The goal of this project is to analyze a sample of negligence cases in BC since the year 2000 in order to determine how damages and contributory negligence have been changing over time. Currently, there does not exist any research into whether damages or contributory negligence decisions have been changing over time or if they have remained relatively steady.

In order to complete a meaningful analysis, important pieces of information such as the damages paid and the percentage that the plaintiff is found liable must be extracted from case reports. This is the most challenging aspect of our analysis and is referred to as information extraction. With enough help, this information can be extracted manually with high accuracy. However, the problem with this is the amount of manpower, time, and cost required. Developing a computer system to perform this information extraction is a tradeoff between having lower accuracy and being able to process much larger amounts of cases very quickly. Another benefit is that it can also be applied to new cases in the future with no additional costs. Our team focused on two specific methods to develop a system that pulls out relevant information from case reports. The first method is commonly referred to as a rule based classifier approach. This method primarily involves manually laying out specific patterns that a computer will use to search through large amounts of text data and return the text which matched a pattern. Many case reports will use similar language when stating the final amount of liability or damages awarded which makes the rule based classifier a good starting point. The second method used is known as a statistical classification approach. The idea behind a statistical classifier is to get a computer to inspect potentially relevant pieces of information and determine the group or class the information belongs to. In the context of a negligence case, the class could be the different types of damages that can be awarded such as special, punitive, or aggravated damages. The relevant information could be the words used directly before or after a numeric value. This approach still requires some cases to be manually inspected and classified by humans in order for the system to learn to distinguish classes. This is a time consuming process but will typically produce results at least as good as a rule based approach.

For evaluation, we compared both the statistical and rule based classifier to a baseline. The baseline used was a dummy classifier. Dummy classifiers are classifiers with very simple rules that can be used as a simple baseline. Both methods we used performed better than the baseline which proves our classifiers are extracting information in a somewhat intelligent way. Both methods used gave similar results for overall accuracy. However, there are many cases where the case does not have any damages or liability apportioned which means the overall accuracy is a little bit misleading. The statistical classifier performs better than the rule based classifier when an actual

[1] Negligence Definition

damage or liability percentage is present. This is more important for our analysis because we are not interested in cases where there is no information for us to extract.

The next step is to statistically analyze the results of the information we were able to extract from the case reports. It was found that, on average, negligence cases in BC have been seeing a gradual increase in damage payouts, adjusted for inflation, since the year 2000. It was also found that judges have generally become more thorough in case reports as the average decision length of cases has also been steadily increasing over the same timeframe. Contributory negligence showed a mean of 38% reduction which was stable over the years. The analysis has methodically confirmed the overall suspicion in the legal community about the changes in how judges have been treating negligence cases over the past 20 years. It can be inferred by data from the past 20 years that the overall upward trend will continue to rise in the near future.

Background Knowledge

In order to fully understand the language, concepts, and methodology used in this paper some terminology and concepts need to be introduced.

Damage Types

Tort law is divided into three categories of “Intentional tort”, “Negligence”, and “Strict liability”. In this paper, we will only analyze negligence cases in BC since the year 2000. The damages awarded in negligence cases branch into 3 main categories: “Non Pecuniary”, “Pecuniary”, and “Other”. Non Pecuniary damages are awarded to compensate the plaintiff for things that have no bill or real cost but have a real impact on the plaintiff's well being, an example of this is pain and suffering. Pecuniary damages are damages awarded to the plaintiff where a bill exists or there is a reasonable estimable cost for them. Pecuniary damages can be further separated into:

- General Damages: Any compensation decided by the court for any future costs such as future loss of earning
- Special Damages: Damages where there is a physical bill that has already been produced or previous costs can be calculated. Examples include physiotherapy bills or wage loss due to missed work.

The “Other” category of damages involves:

- Aggravated Damages: Damages to compensate the plaintiff for loss of dignity, and
- Punitive Damages: Damages to punish the defendant for conduct that was malicious, high handed, or particularly offensive.

A full flow diagram of damage types is attached in Appendix 1.

Technical Knowledge

In this paper two main approaches were used to extract desired outputs as stated in the summary. In the rule based classification method, one tries to assign a set of rules to find patterns and characters that are repeated in the text which may be a marker of the desired information. For example, the text may mention *“I award the plaintiff $20,000 in special damages”,* the rule can be if the word “special damages” occurred in the text then the number before that is referring to the

amount awarded for special damages. To perform pattern matching a standard technique known as regular expressions (regex) was used (more information in appendix 2). The problem with this approach is that the language used in legal cases has a lot of variation in it making it difficult to engineer patterns that do not over or under generalize the information that needs to be extracted.

The second approach used in this paper is the statistical classification approach. Statistical classification is the process of predicting the class or category of given data. Using this approach a model has to be built, the model learns from examples to categorize and classify text. An example of an algorithm that is used could be a decision tree[2], where a "tree" is drawn upside down with its root at the top which represents a condition, called nodes, based on which the tree splits into branches, called edges until it can't split into any more categories. Each node will contain a condition that can be used to differentiate different classes.

To make the model we feed a classification algorithm a set of labeled data, called training data, which is data where we already know what the amount of damages and what amount of liability was apportioned to the plaintiff, if any. In other words, the training data comes with an answer key for the classifier to learn from. The classifier "fits" the training data, meaning it learns the patterns that differentiate the different classes. After fitting, the classifier is now ready to classify text where it does not know the answers.

Before finalizing the classifier one must be sure that they have achieved optimal performance with it. Fine-tuning is done by changing parameters or features and seeing how it impacts the performance. To help with fine-tuning we use a technique known as cross validation. Instead of fitting or training the model once we will fit the model many times using most of the training data to train and a small amount of it to evaluate performance. For example, if our training data had 10 examples in it we would train the model using 8 of them and make a prediction on the remaining 2. Then we would select 2 new examples to test against and train with the remaining 8 until every datapoint has had a chance to be included in the test set. The alternative to cross validation is to dedicate a certain portion of the training set to always be used as the test set. The downside to having a dedicated test set is that the size of the training set is reduced and you don't test on as many data points.

The main classification algorithms that are used in this paper are Random forest, XGBoost, Logistic regression[3]. To understand the terminology in more detail refer to Appendix 2.

Data

Data Source

Our primary data source for legal documents was LexisNexis. LexisNexis is a corporation that specializes in computer-assisted legal research[4]. They make large amounts of legal cases easily accessible electronically. All cases that were pulled from LexisNexus were meant to be negligence cases in BC and were queried from the website using the following search parameters.

---

[2] Decision Tree
[3] Random forest, XGBoost, Logistic regression
[4] Lexis Nexis - About Us

- Limit to cases identified as torts
- Limit to cases between January 1st, 2000 and April 19th, 2020
- Limit to cases where the word “negligence” appears

From the initial query, we pulled a total of 4,118 cases from the database. The data came formatted as 85 individual Microsoft Word .DOCX formatted files. The data was converted from the original .DOCX format into a more lightweight .TXT format. It is unnecessary to keep additional information from the .DOCX format as we are primarily interested in parsing through the actual text of each document to pull out relevant information. One major limitation in our approach is that we are unable to make any use of information that is stored in images, such as a table of damages inserted into the .DOCX file as an image rather than text. In general, a majority of cases followed the same structured format; beginning with information such as the case title, entities involved, hearing date, judge, decision length, case summary, and location of the hearing. Because the beginning of every case began in the same format it was easy to extract the listed items by making use of patterns and location of the text.

Data Filtering & Preprocessing

Not every case in our collection of 4,118 was relevant to our analysis. Time had to be spent filtering out cases that would not award any damages to improve the reliability of our analysis. The full list of cases filtered out of our analysis is found below.

Criteria	Description	Number of Cases
“R. v. *Defendant*” in case title	“R. v.” signifies a crown case involving criminal law	43
Any case without “British Columbia Judgements”	If the originating reporter is not B.C.J. we exclude the case to boost the reliability of our study of cases exclusively in BC	190
“(Re)” in case title	Any case involving “(Re)” will not share the common plaintiff-defendant dynamic and will fall under interlocutory matters. In other words, they will not have a final decision made by the judge.	15
Client & Solicitor cases	Does not follow the common plaintiff-defendant dynamic.	17
“In the matter of” instead of “Between plaintiff & defendant”	Does not follow the common plaintiff-defendant dynamic.	12
Third Party Procedure	Deals with the third party of a case. Typically will involve an argument for removing or adding a third party. Does not award damages.	60
Disposition without Trial	Deals with the defendant asking the judge to dismiss a case based on a lack of evidence or proper documentation from the plaintiff.	73
Applications and Motions	Deals with applications or motions brought up with the judge.	20

	Will not award any final damages.	
Right to a Jury	Deals with applications requesting case to be heard only by a judge without a jury	10
Adding or Substituting Parties	Similar to third party procedure. Usually deals with applications requesting to add or remove a defendant	20

Table 1 - Case Filtering

The table above sums up to 460 cases, 15 of which overlap, causing us to remove a total of 445 cases out of 4,118. This resulted in having 3,673 cases in our final dataset.

Case Format

Each case pulled from LexisNexus follows a very similar structure. The beginning of each case includes information like the case title, judge name, registry location, decision length, case summary, and the plaintiffs, defendants, and third parties involved in the case. Within the case summary, the report may include the result of the case in the form of "HELD: Application Dismissed" which allows for filtering of cases where the plaintiff lost. Sometimes the summary may include damages awarded or liability apportioned which we are attempting to extract. The case report will follow the case summary where each line begins with a paragraph number unless it is a section header which is used to organize the report into an easier to read format. Section headers are not standard across cases but some common headers will be used such as "Conclusion", "Summary", or "Damages". When converting the cases into a TXT format, the headers were surrounded by XML tags which are used in our classifier based approach. Surrounding the header with a specific tag will allow the classifier to be able to understand which section it is currently inspecting; headers are useful for both humans and machines because they give additional context about the text. Typically, the information we are trying to extract will appear either in the case summary or near the end of the case in the conclusion. Finally, each case will end with an "End of Document" marker. An example of a typical case that we need to extract information from is included in appendix 3.

Annotations

In our methods section, we discuss the use of annotated training data in the statistical classification based approach. In general, training data can be thought of as data that a machine learning algorithm uses to learn from. The algorithm uses knowledge from examples it has already seen and applies it to new examples that it has not seen before. The idea is that we can manually create a training set from a small sub-sample of cases that the algorithm can use to provide useful information on the entire set of cases. The information that we wish to feed our classifier would be any numeric value found in a case and the result we want from the classifier is what type of damage, if any, is it. Therefore, in our training data we must annotate every numerical value and assign a class or category to it. One method of doing so is to insert tags around the values directly in the .TXT file which can be easily read by a computer. An example sentence we wish to annotate may be,

"I award the plaintiff $20,000 in special damages"

This would be annotated manually to,

"I award the plaintiff <damage type = 'special'>$20,000</damage> in special damages"

We would annotate liability (contributory negligence) in a similar fashion and use the word "percentage" rather than "damage" because liability is apportioned as a percentage fault towards the plaintiff or defendant. We manually went through 298 cases and annotated each numeric dollar value or percent value. The possible classes we could have assigned to a value is included below.

Tag Name	Tag Type
Damage	"Other" - Any value that is not a final damage (e.g. plaintiff is seeking this value or the plaintiff is referencing a damage payout from a previous case) "Non Pecuniary" "Special" "In Trust" "Past Wage Loss" "Future Wage Loss" "General" "Punitive" "Aggravated" "Future Care" "Reduction by" - Another value is reduced by this amount "Reduction to" - Another value is reduced to this amount "Total" - Total damages awarded in a case "Total After" - Total damages awarded after contributory negligence
Percentage	"Other" - Any percentage that is not contributory negligence "CNP" - Contributory negligence percent for the plaintiff "CND" - Contributory negligence percent for the defendant

Table 2 - Annotation tags

Overall, the tags presented above were able to cover the different types of information we needed to extract for our analysis. However, our biggest challenge is when a case has damages broken down into an itemized list. For example, the court may award a plaintiff money for past wage loss at different jobs and may not sum the values. In this case, we would prepend the term "sub-" in front of the tag, leading to the tag becoming "sub-past wage loss", to indicate that the value is not the final value but is the partial value. The intuition behind this is that our classifier based approach may be able to learn the difference between certain damages being broken down into individual items compared to the final sum of a damage type. An example of a case annotation can be seen in Appendix 4.

Data Statistics

As stated earlier, our data range from the years 2000 to 2020. The distribution of cases per year is relatively even with the exception of 2020 due to the fact that we pulled the case information part way through the year. Therefore, 2020 was excluded from any analysis.

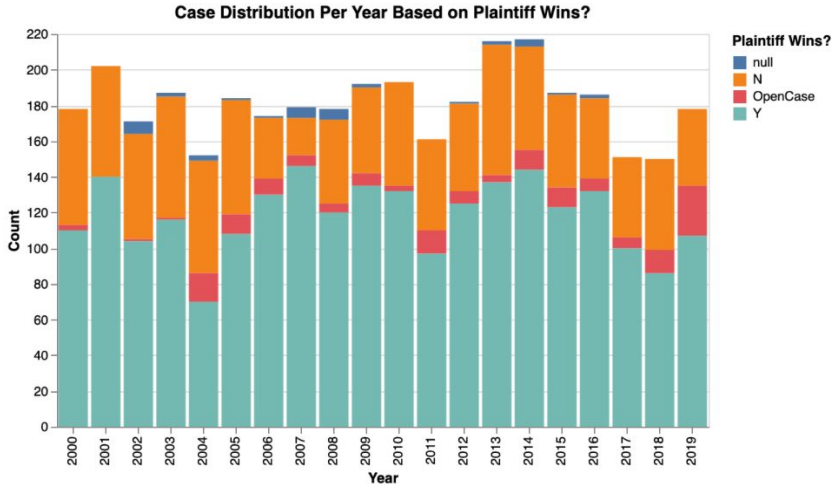


Figure 1 - Case Distribution per Year Based on plaintiff winning or Not
Y (green) suggests plaintiff winning, N (orange) indicates plaintiff losing, OpenCase (Red) indicate undecided cases and null (blue) represents the cases where the classifier could not decide whether the plaintiff won or not

We tagged 5,168 individual numbers as damages and 997 individual numbers as percentages. Of the 5,168 numbers tagged as damages, there were 3,006 (58.2%) with the tag type of "other". This shows that the majority of values mentioned in a case will not be the amount that is awarded. Similarly, of the 997 numbers tagged as percentages, there were 695 (69.7%) with the tag type other. One different reason for percentage having so many "other" tags is because we are only tagging percentages that involve contributory negligence. Of the remaining 302 percentages tagged, about half were percentages assigned to the plaintiff and half were percentages assigned to the defendant. Below we show the overall distribution of each damage tag type excluding the "other" tag.

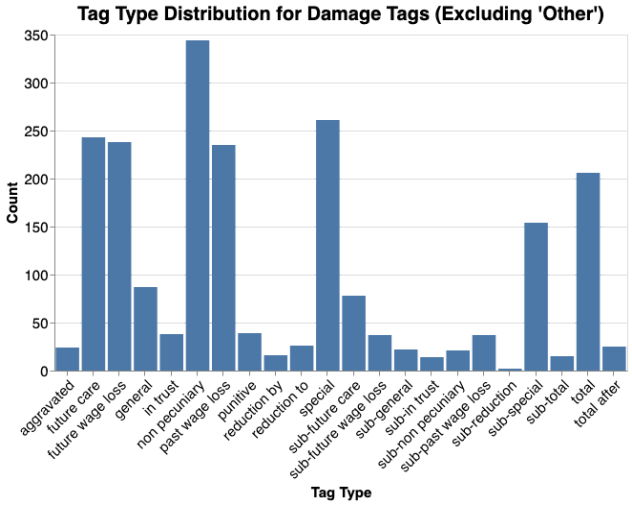


Figure 2 - Damage Tag Type Distribution

After tagging each case we place the relevant information into a .CSV format file. This file stores information about the case such as the case title, judge name, location, damage awards and contributory negligence percentages. The damage awards are always broken down into subcategories. This .CSV file can later be used to evaluate how well we are doing with our rule based classifier approach and statistical classifier approach. For each of our approaches, the goal is

to extract all relevant information from unannotated data and export the information into a .CSV file for further analysis.

Research Questions

The main goal of this paper is to answer the following questions:

- *How have negligence payouts changed between the years 2000 and 2019 in BC?*
- *What percentage of cases involve contributory negligence and what percentage succeeded in proving contributory negligence?*

To answer these questions the following types of information were extracted from the text files.

String type	Numeric type	Boolean type
Case Name	Total Damage	Written Decision
Year	Total Pecuniary	Plaintiff Wins
Judge Name	Non Pecuniary	Multiple Defendants
Registry	General	Contributory Negligence Raised
	Special	Contributory Negligence Successful
	Punitive	
	Aggravated	
	Future Care	
	Decision Length	
	Percent Reduction due to Contributory Negligence	

In addition to our main research questions, other observations were made using statistics such as if there is a correlation between a specific variable and the total damage payouts (per year, per judge, per location, and if multiple defendants are present). See the analysis section for more information.

Methods

Preprocessing Word Documents

To approach this problem, we first converted all .DOCX files queried from LexisNexis to .txt files for ease of manipulation in Python. We used a Python library called *docx* to iteratively extract all the text and any tabulated information found in the original .DOCX files, and write the extracted text to a blank .txt file. We found a few cases that had images embedded in the original document. These cases were rare and we chose to skip any images found.

Rule-Based Metadata Extraction

After spending some time understanding the common structure of British Columbia Judgements (BCJ) negligence cases from LexisNexis, we decided to make a first pass by extracting desired metadata per case using regular expressions (regex). The beginning of each case starts with the case name, such as "*Mawani v. Pitcairn, [2012] B.C.J. No. 1819"*, and the end of each case is marked by the line "*End of Document*". This made it easy to split documents by case. Many of the desired meta fields such as Case Name, Plaintiff Wins, Judge Name, Multiple Defendants, Decision Length, and Registry, could also be encountered within the first 20 lines of each case. Due to the structured format of these meta fields, we decided there is no reason to include these fields in our statistical classifier. We also filtered for cases that are identified as British Columbia Judgements using the first few lines of each case.

Rule-Based Damage Extraction

The rules used to extract the damages awarded to each case required reading through many cases to understand common patterns. The most deciding patterns we encountered include the location in the document where final values are mentioned and the language used to describe each value. As one would expect, final values awarded by the Judge are typically mentioned near the end of each case and reiterated at the beginning of the text under a header called *Case Summary*. Although some cases separate final decisions under a header with the word *Conclusion* or *Damages*, this was not consistent enough to use as a rule.

As shown in Appendix 1, some damage types are the sum of other damage types. For example, General damages are made up of Future Wage Loss, and Future Care. Special Damages include In-Trust awards, Housekeeping, and Past Wage Loss. Although some cases include a cumulative value per category (eg. Total, Non-Pecuniary, General, Special, etc), many do not. This added another layer of complexity to our rule-based methods for extracting damages.

Ultimately, the rules used to extract damages include the context around dollar amounts (up to 10 tokens), the ratio of paragraph number where the value was found to the decision length, and vocabulary specific to each damage type. The general steps include:

1. Only consider cases in which the plaintiff wins
2. Use a regex pattern to split the case into numbered paragraphs
3. Search each paragraph for a dollar amount
    a. If the paragraph location ratio is greater than 0.9 or the value was found in the *Case Summar*y section, we continue to apply rules to categorize values, otherwise, we disregard the dollar value match
4. If the value was located in the last 10% of paragraphs or in the summary: Compare the context surrounding the value to the common language specific to each category to determine which damage type
5. Assign each value to its corresponding category - choose the last value found per category

a. If none of the values in the text were identified as a Total Damage value, sum Non-Pecuniary and Pecuniary damages to get a total

Rule-Based Contributory Negligence Extraction

The methods used to extract the percent reduction [of damages] due to contributory negligence are similar in nature to those used to extract damages. Before seeking individual percentages mentioned in the case, we filtered for cases in which the plaintiff won and the phrase *contributory negligence* was mentioned somewhere in the text. After filtering, again, we started by iterating through numbered paragraphs in the case and assigning a score to each paragraph based on where it occurred in the text. If the paragraph score was greater than 0.6, we continued to apply conditions to determine if the percentage was related to contributory negligence, otherwise, we continued to the next paragraph. The paragraph score of 0.6 is a ratio that we tuned to ensure the rule-based method was not over or under assigning the percentage as contributory negligence.

1. For any percentages found whose location scored above 0.6, we searched up to 8 tokens on either side to check for specific vocabulary related to contributory negligence
    a. An example of some of this vocabulary includes the following words: contributory, liability, apportion(ed), fault, against, recover, and responsible
    b. Depending on the language, the percent fault is either assigned to the plaintiff or assigned to the defendant.
2. Another rule was utilized to determine whether the word *plaintiff* or the plaintiff's name (extracted from the case name) was found in the context or the opposite was true for the defendant
3. If a percent was found located in either the last 40 percent of paragraphs with the appropriate language in the context, a final check was made to ensure either the word *plaintiff*, *defendant,* or *damages* was also found in the context
    a. If so, *Contributory Negligence Successful?* was set to True, otherwise, it was deemed False
    b. If contributory negligence was not found to be successful after searching through all paragraphs in the case, repeat steps 1-3 for percentages in the Case Summary
4. Select the last percentage found in the text if the above is satisfied

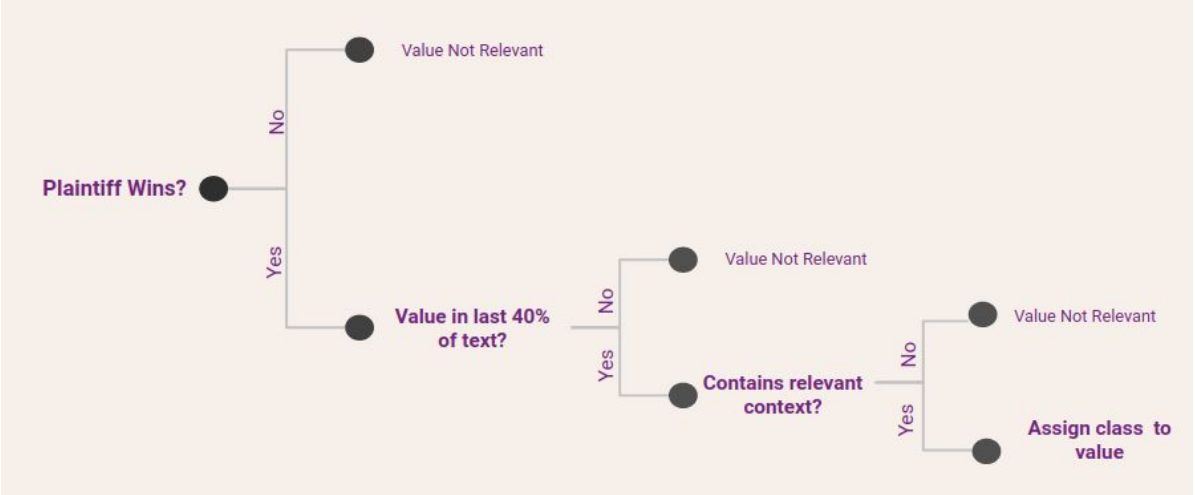


Figure 3: Rule-Based Classification Pipeline

Rule-Based Evaluation

To test how well our rule-based information extraction was performing, we created a function that computes the accuracy of empty fields, filled fields, and overall accuracy across each attribute we extracted. These accuracies all compare the values extracted by our rule-based methods to the true values annotated manually. Empty accuracy is defined as the number of times our rule-based extraction correctly determined a field was empty, out of the total number of empty values for that attribute. Filled accuracy is the ratio of correctly identified values in a certain field out of the total number of filled values for the same field. For example, many cases do not have damages awarded - these cases would have an empty entry for all damage quantities. If our rule-based method correctly identifies that the case does not have any damages, this is a correct empty field. Table 5 in the *Evaluation* section demonstrates rule-based accuracies.

As shown in Table 5, our rule-based information extraction does quite well for both filled and empty fields for the fields such as Registry, Plaintiff Wins, Multiple Defendants, Decision Length, and Case Name. As anticipated, these values are stored in a more consistent format, and therefore results are upwards of 85% accuracy for both filled and empty fields. Accuracies for damages awarded are quite good when there are not any damages awarded (empty fields), but does not do so well for filled values. There is such variation in the structure for how these values are stored, as well as many irrelevant dollar values mentioned, that it is hard to make enough rules to correctly capture the values. The performance of *Contributory Negligence Successful?* And *% Reduction* performed a lot better than anticipated after fine-tuning the rules and optimal paragraph location ratio cut-off.

To improve results for damages extracted, we built a statistical classifier to categorize dollar values as described in the Annotations section above.

Statistical Classification

Damages Statistical Classifier

Given the class imbalance between tagged damage types as described in the Data Statistics section above, the low volume of annotated data, the need to differentiate between the 23 classes being predicted, and large feature set inherent to text classification, we decided to compare results across 3 classifiers: Logistic Regression, Random Forest, and XGBoost. Both Logistic Regression and Random Forest were implemented using the *sklearn* packages, where XGBoost uses an open-source library[5]. First, we preprocessed the text in each case by lowercasing all words and removing stop words included in nltk.stopwords[6] (*and*, *the, if, of, an, to, as, but, their* etc). The pipeline for building the training set for these classifiers is as follows:

1) For each annotated case, using a regex iterator, loop through each tagged damage
2) For each tagged damage:
    a) extract the quantity tagged (damage value) and tag type (true label)
    b) extract context around the value and build list of dictionaries, where each dictionary is a feature, based on word counts in the context (Bag-of-Words) among other features
3) Use the sklearn DictVectorizer[7]to turn the list of dictionaries into a matrix of shape number of tagged examples by number of features

As enumerated in the pipeline above, to implement these classifiers, we built two functions that iterate through all annotated cases, use a regular expression to find tagged damage values, and extracts several fine-tuned features into a list of dictionaries for each tagged value. The context length around each value was a parameter than we manually optimized for. The best results for the Damages Statistical Classifier used a context length of 6 tokens on either side within the same sentence. The feature engineering process is described in detail below.

Feature Engineering

To start, we chose many of the same features that were used in the rule-based method for damage extraction. The primary features include a Bag-of-Words (BOW) representation of the context surrounding each value. BOW maps each word in the context to the number of times it appears. This allows the classifier to associate certain words with each category. In addition to the BOW counts of each word in the text, we added a BOW feature that has word counts for where the word occurs relative to the value, before or after. Another feature adopted from the rule-based method is what we call *start_idx_ratio* which represents the location of the value relative to the start of the text. Values near the end will have a ratio close to one and values near the start have a ratio close to zero. In addition to the location and BOW features, we included a range in which the value itself falls. These bucket ranges include: values less than $1,000, values between $1,000 and

[5] Open Source Library
[6] nltk.stopwords
[7] sklearn DictVectorizer

$25,000, values from $25,000 - 100,000, $100,000-500,000, and values above $500,000. For example, a value of $5,000 falls in the bucketed range of $1,000 and $25,000. Another feature we added is based on the header above the dollar value of interest. Often, cases have headings such as *Future Care Awards, Wage Loss, Summary of Damages,* etc. An example of a feature set for a single value is shown below:

{'case@Heading': 1,
'summary@Heading': 1,
'start_idx_ratio': 0.04429809982826436,
'range': '100000 - 500000',
'prev word': 'services,',
'next word': 'accommodations',
'prev bigram': 'therapies services,',
'next bigram': 'accommodations renovations,',
'prev trigram': 'equipment, therapies services,',
'next trigram': 'accommodations renovations, $175,000',
'age@Before': 1, '19,@Before': 1,
'$950,000@Before': 1,'Equipment,@Before': 1,
'therapies@Before': 1,'services,@Before': 1,
'age': 1, '19,': 1, '$950,000': 1,
'equipment,': 1, 'therapies': 1,
'services,': 1, 'accommodations': 1,
'renovations,': 1, '$175,000': 1,
'in-trust': 1, 'past': 1, 'future': 1}

Ablation Study

To optimize our set of features, we used a process called ablation which involves removing one feature at a time and comparing precision and F-score results before and after. In Table 10 in the *Evaluation* section below, the ablation results are shown for the optimized Logistic Regression classifier. In our final feature set, we used the least number of features resulting in the highest precision performance for our best classifier, shown in the last row. The row in bold shows the best performance after removing the *BOW after* feature.

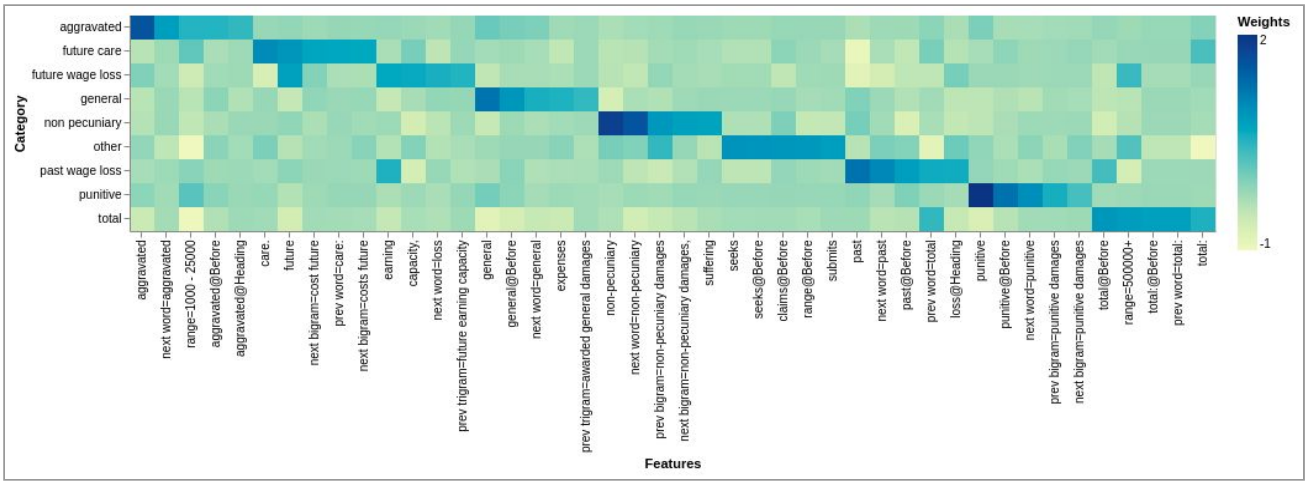


Figure 4: Top 5 Features Per Category: Feature Weights

As shown in Figure 4 above, our model was able to pick out sensible features for predicting damage types. For example, features for the category of *future care* include the words *care, future,* bigram *costs future,* and previous token equal to *care.* The feature *future* has shared importance for the other damage type *future wage loss*. One of the best examples that our classifier is working well, are the top five features for the category *other*. Namely: *seeks, seeks@Before, claims@Before, range@Before,* and *submits*. Often the plaintiff "seeks" a damage award of some value, but it is ultimately not what we want to pick out of the text.

Hyper-Parameter Optimization

Once we determined the best combination of features across all classifiers, we ran a cross-validated grid search using the *sklearn* package GridSearchCV[8] to optimize for the best subset of classifier hyperparameters. *GridSearchCV* compares the performance of the classifier specified using all combinations of parameters specified. While Random Forest and XGBoost have few possible hyperparameters to tune, Logistic Regression has several. To control for randomness, we set the *random_state* parameter in each classifier to 42 throughout our research. For Logistic Regression, we tested for different combinations of the following hyperparameters:

Parameter	Values
penalty	l1, **l2**
C	0.1, **1**, 10, 100
solver	liblinear, **newton-cg**
class_weight	**None**, balanced

Table 3: Hyper-parameter optimization - best results in bold font

The best results for Logistic Regression to classify damages are bolded in the table above. The two tree-based classifiers performed best using default values.

Damages Statistical Classifier Evaluation

After feature engineering and hyperparameter tuning, the final performance across the three damage classifiers tested can be found below in Table 8 in the *Evaluation* section.

All three classifiers performed similarly, but we wanted a model that gave us a high precision for the Total Damage category specifically. This is because our main research question about changing damages overtime can be best answered by analyzing the Total Damage category. We decided to use the Logistic Regression model for final analysis both because of its high precision

[8] GridSearchCV

performance, but also because of its *predict_proba*[9] method which provides a probability distribution across all categories for each value. Although many sklearn classifiers include a *predict_proba* method, Logistic Regression derives the probability distribution as part of its decision boundary output and performs better than most. This probability distribution is useful for selecting high-confidence results for unseen data. The overall performance of the optimized Logistic Regression model is shown in Tables 4 and 9 in the *Evaluation* section below.

Applying Damages Logistic Regression Classifier to Unseen Data

Once we determined the optimal feature set and the Logistic Regression model, we implemented two functions to apply the model to unseen data: one that predicts dollar values found in the text, and another that assigns a final prediction for each damage type to each unseen case. Given an unseen negligence case, we then use the optimized model fit on all annotated data, and the *sklearn* vectorizer used to transform the annotated features to predict the damage category. The *predict* function follows this general procedure:

1. Iterate over all dollar amounts found in the case using a regex iterator
2. Extract features for the value found and append to a list
3. Store the float value of the dollar amount found
4. Once steps 1-3 are repeated for all values, transform the features using the fit-transformed vectorizer
5. Use the fit classifier to predict the damage tag and associated probability distribution

Using the results obtained from our *predict* function, we then pass results into the *assign_classification_damages* function which follows this pipeline:

1. Iterate over all predictions and associated probabilities in a single negligence case
2. If the maximum probability (between 0 and 1) associated with the predicted tag is above the threshold we set (0.5), add the tag to a dictionary mapping the predicted tag to a list of possible high-scoring values
    a. {general:[5000, 5000, 3000, 5000], special: [10000, 7000], non pecuniary: [80000, 80000], ...}
3. Once all predictions have been assigned to the dictionary of possible values per category:
    a. If the tag is of type *other*, continue
    b. If the keyword argument *high_precision_model* is:
        i. *True*, skip all 'sub' categories and choose the last value present in each list of possible values
        ii. Otherwise, if the category is of type 'sub' (eg *sub-special* or *sub-general*) and there are no non-sub versions of the same tag (eg *special* or *general*), then we add the values in the list
    c. If no values were found in either a *sub* category or a non-sub category, the final value is set to *None*
    d. Assign the final value per damage type to a dictionary
4. Return the final damage assignments

[9] Predict_proba

We modified our original parsing function that uses our rule-based methods as default, to allow for use of a statistical classifier to assign values for damages and contributory negligence percentages. The output of this parsing function contains a list of each case along with the extracted values for that particular case. Similar to the damages statistica classification process, the optimal context length around each value was optimized and found to be 2 tokens on either side of the value.

Contributory Negligence Percent Reduction Statistical Classifier

Feature Engineering

The feature engineering process for our percent reduction classifier was quite similar to that used for the damages classifier. We started using a BOW representation and differentiated word counts for words that appear before and after the percent value of interest. We also added a float equivalent of the tagged string percentage as well as the start index ratio feature telling where the value lies relative to the end of the case. We also made our own contributory negligence-related lexicons containing words such as *fault, against, apportion, responsible, recover,* and *contributorily*. The one feature that we used for percent reduction that we did not need to apply for damage quantities was the presence of the word *plaintiff* or *defendant* and their respective names. As described in the rule-based methods for percent reduction, we used a regex to extract the two names from the case title. This feature improved accuracy by a few percent as shown in Table 11 in the *Evaluation* section below. The table shows results of our ablation study comparing the effects of each feature on model performance. Removing the *start_idx_ratio,* and *lemmatization* proved to have the best impact of classification performance.

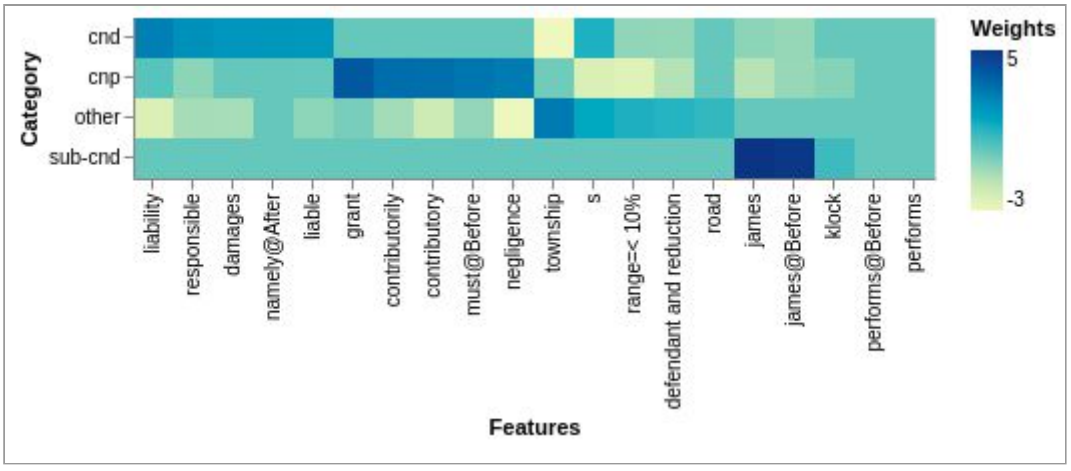


Figure 5: Top 5 Most Important Features, Percent Reduction Classifier

As shown in Figure 5 above, the most important features used to predict our contributory negligence percentages against the defendant (*cnd*) include words such as *liability, responsible,* and *damages*. Features used to predict contributory negligence percentages against the plaintiff (*cnp*) include: *grant, contributorily, contributory,* and *negligence*. You can tell by the feature weights that our model is less sure about predicting the *other* category of percentages, though the feature *range<10%* is a good indication; our team has not seen a percent reduction value less than 10% mentioned in a case during the annotation process. Finally, the *sub-cnd* category is quite certain

about its predictions (dark blue) given the features *james*, and *james@Before*. This is indicative of overfitting. We only have 6 examples of annotated *sub-cnd* features from a single case, in which James is one of the defendants. For this reason, we did not use the *sub-cnd* category to assign percent reductions for evaluation.

Hyper-Parameter Optimization

The same methods were used to optimize our model parameters for the contributory negligence percent classifier as those for the damages classifier. For more details, refer to that section above.

Applying Contributory Negligence Logistic Regression Classifier to Unseen Data

Similar to applying the damages classifier to unseen negligence cases, once we determined the optimal feature set and optimized the Logistic Regression model, we implemented two functions to apply the model to unseen data: one that predicts percentages found in the text, and another that assigns a final prediction for each percentage type to each unseen case. As described in the *Annotations* section above, our percent reduction classifier distinguishes between 3 main classes: *CNP* which is percent reduction against the plaintiff, *CND* which is percent reduction against the defendant, and *Other* for percent values which are unrelated to contributory negligence. Given an unseen negligence case, we then use the optimized model fit on all annotated data, and the *sklearn* vectorizer used to transform the annotated features to predict the percent category. The *predict* function follows this general procedure:

6. Iterate over all percentage amounts found in the case using a regex iterator
7. Extract features for the percent found and append to a list
8. Store the float equivalent of the percent found
9. Once steps 1-3 are repeated for all values, transform the features using the fit-transformed vectorizer
10. Use the fit classifier to predict the damage tag and associated probability distribution

Note that because the *predict* function using the damages classifier is almost identical to that for the percent reduction classifier, we added a few keyword arguments to use the same function for both.

Using the results obtained from our *predict* function (with the appropriate keyword argument telling it which regex pattern to use), we then pass results into the *assign_classification_CN* function which follows this pipeline:

5. Iterate over all predictions and associated probabilities in a single negligence case
6. If the maximum probability (between 0 and 1) associated with the predicted tag is above the threshold we set (0.7), add the tag to a dictionary mapping the predicted tag to a list of tuples of possible high-scoring values, along with their associated probability. For an example:
    a. {CNP:[(0.5, 0.88),(0.3, 0.97), (0.5, 0.71)], CND: [(0.7, 0.88)]}
7. Once all predictions have been assigned to the dictionary of possible values per category:

a. If the tag is of type *other*, continue
b. If the category is of type *CND*, we need to subtract the float from 1 because the percent is against the defendant, and we are interested in the equivalent value against the plaintiff
c. Otherwise, choose the most probable value from the list of tuples, (value, probability)
d. If no values were found in either a *CND* category or *CNP* category, the final value is set to None

8. Return the final, most probable percentage for the case

Evaluation

Below are the results for the procedures described in our *Method*s section above.

Rule-Based Classification vs Logistic Regression Classification

Columns	Rule-Based Overall accuracy (%)	Logistic Regression Overall accuracy (%)	Baseline Overall accuracy
$ Damages total	50	48.57	32.5
$ Pecuniary Damages Total	26.07	23.93	9.29
$ Non-Pecuniary Damages	82.14	90.36	48.21
$ General Damages	77.14	80	50
$ Special damages	80.71	78.57	43.57
$ Punitive Damages	98.21	98.57	84.64
$Aggravated Damages	98.92	98.21	89.29
$Future Care Costs	87.86	91.07	52.5
% Reduction	93.93	93.57	21.48

Table 4: Overall Accuracy - Rule-Based Classifier vs Logistic Regression Classifier

Note: Dummy Classifier was used to generate baseline scores - randomly guesses based on category distribution

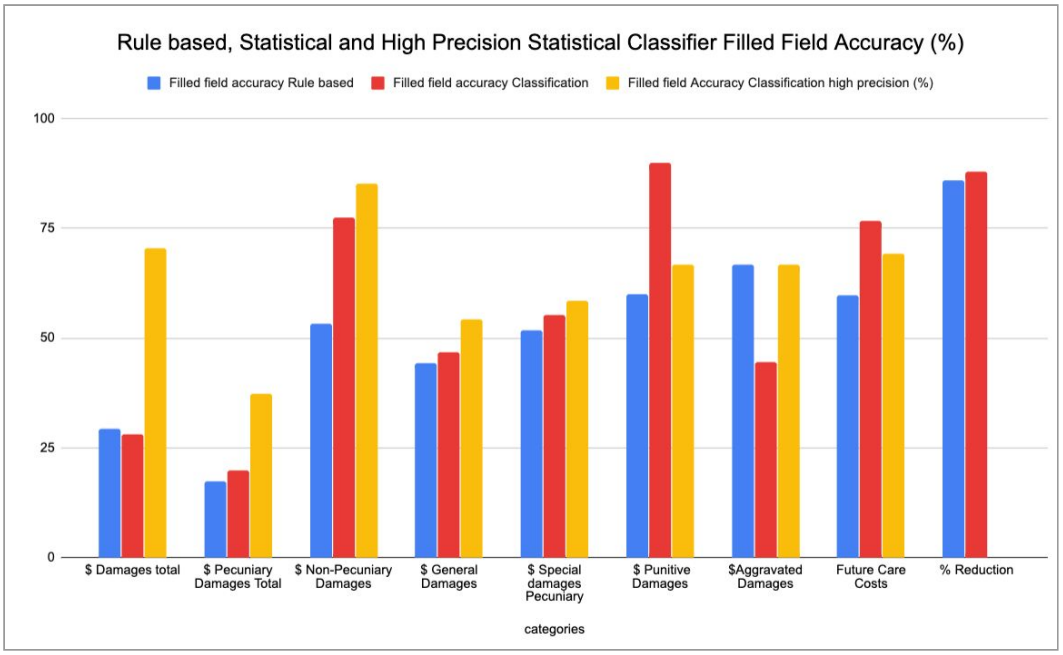


Figure 6: Rule-Based ,Logistic Regression, High Precision Logistic Regression Filled Field Accuracy Results

To evaluate the results of our classifiers we compared three different accuracies. Overall accuracy is the ratio of correctly predicted values (empty or filled), empty accuracy is the ratio of correctly identified empty fields (e.g. cases without damages awarded), and filled accuracy is the ratio of correctly identified filled values to the total number of filled values (e.g. cases with dollar amounts or percentages reduced in the text). As shown in Table 4 and Figure 6 above, our statistical classifier using Logistic Regression performs better than our rule-based classifier. Figure 6 also includes the accuracy of logistic regression when we look at the cases where our statistical classifier made some prediction (high precision). When optimizing our model we found that logistic regression gives very precise results for the total damage column. We can only take advantage of this fact if we only consider cases where our model actually makes a prediction for the total damage. This is expected given the statistical model is able to detect patterns than we could not manually define using the rule-based techniques. We also compared accuracies for a simple baseline model. The baseline uses an sklearn package called *DummyClassifier* which classifies values randomly given the ratio of each class in the train data. For example, roughly 60% of our annotated damages were of type Other, so the classifier will randomly label 60% of values as Other. We wanted to compare our classifiers to this baseline to see if our results are better than chance. You can see in Table 4 that our results are much better than chance across the board.

Bootstrap Analysis: Annotated vs Predicted Damages

Bootstrapping is the process of random sampling from a subset of data, many times, to create a distribution of values for a certain statistic. In our case, the bootstrapped statistic of interest is the median damages for a certain category. This is because we are interested in comparing median damages over time. In figures 7 and 8 we compare the results of bootstrapped damages (Total Damage and Non Pecuniary) between our annotated cases and our predictions. To generate the bootstrapped median distributions, we compare subsets of cases in which plaintiff wins and damages were awarded. For each subset, we subtract the predicted median damages from the gold (annotated) median damages. In Figures 7 above, we can see that the median error when computing median Total Damage from our predicted data is around $7,500 less than the gold

data with a 95% confidence interval of $-49,831 and $61,840. However, when comparing medians for Non Pecuniary damages, the error is significantly reduced. The bootstrapped Non Pecuniary median damages have a 95% confidence interval between $0 and $7000, with a median difference of $0. The reason error is higher when calculating Total Damage is because Total Damage is made up of a combination of Pecuniary and Non Pecuniary damages. Adding all the damage sub-categories causes error to compound. Non Pecuniary damages on the other hand are predicted with 90.36% accuracy and therefore closely matches the annotated damages for that category.

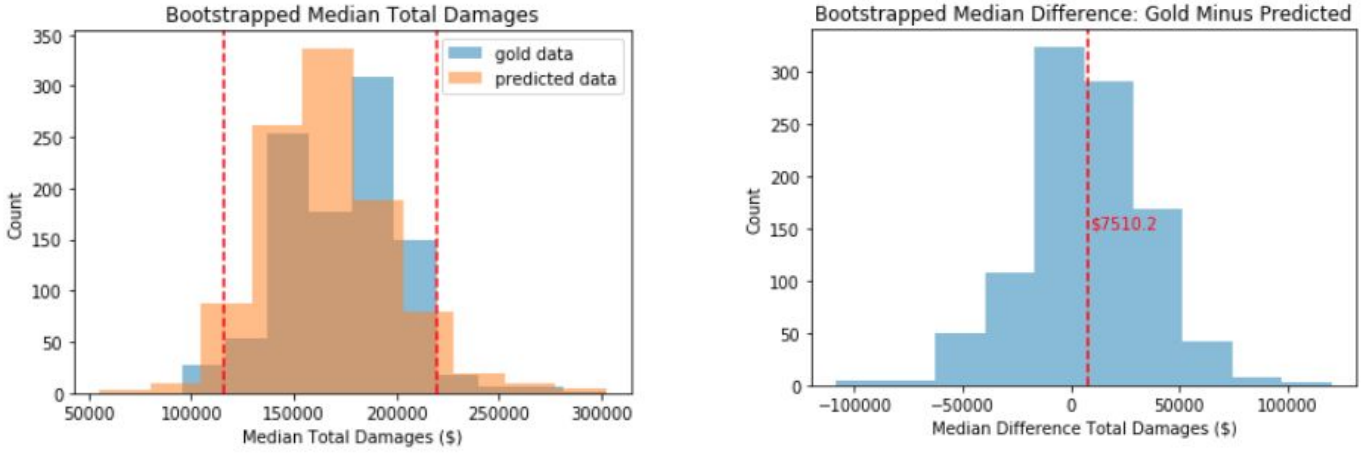


Figure 7: Bootstrap Median Total Damages - Gold vs Predicted

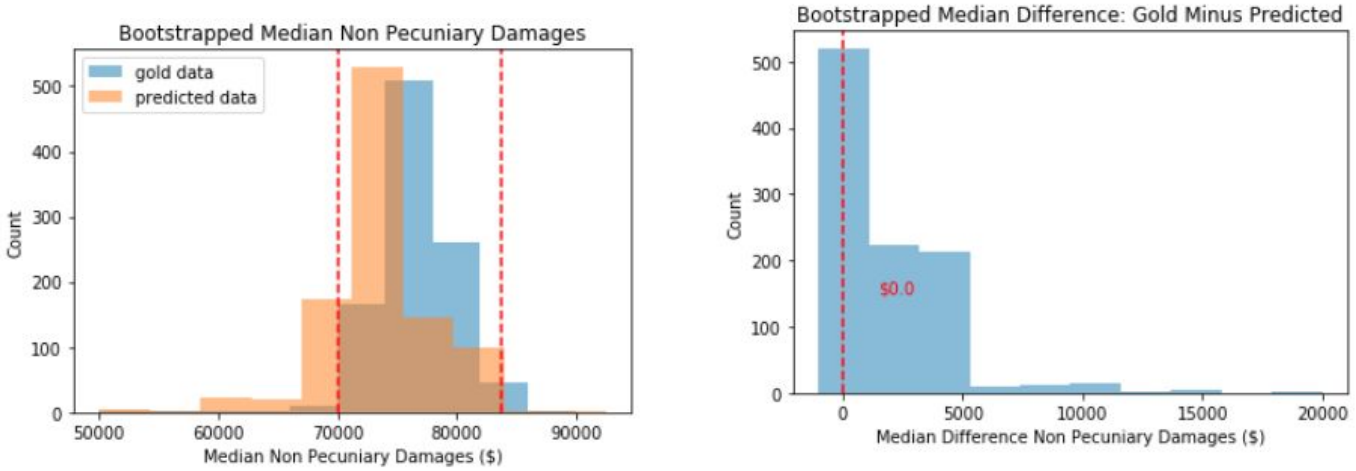


Figure 8: Bootstrap Median Non Pecuniary Damages - Gold vs Predicted

Overall Rule-Based Classification Results

Columns	Overall accuracy (%)	Filled field accuracy	Empty field accuracy
Case Name	100	100	N/A
$ Damages total	50.35	29.38	97.67
$ Pecuniary Damages Total	26.07	17.28	83.78
$ Non-Pecuniary Damages	82.14	53.27	100
$ General Damages	77.14	44.21	94.05

$ Special damages	80.71	51.81	99.41
$ Punitive Damages	98.21	60	99.62
$Aggravated Damages	98.92	66.66	100
$Future Care Costs	87.85	59.74	98.52
Decision Length	99.28	99.28	N/A
Multiple Defendants?	96.07	96.07	N/A
plaintiff Wins?	86.42	86.42	N/A
Registry	97.28	97.85	N/A
Cont. Neg. Success?	93.57	93.57	N/A
% Reduction	93.57	84.09	95.33

Table 5: Rule-Based Information Extraction Accuracies

Overall Statistical Classification Evaluation

To perform an overall analysis of our classification results, similar to the rule-based evaluation, we compared filled and empty value accuracies between the final assigned predictions for both damages and percentages, against the gold annotated data set.

Columns	Overall accuracy (%)	Filled field accuracy	Empty field accuracy
$ Damages total	43.75	28.16	98.0
$ Pecuniary Damages Total	26.34	19.8	74.07
$ Non-Pecuniary Damages	89.29	77.57	96.5
$ General Damages	76.34	46.67	96.27
$ Special damages	76.79	55.24	95.8
$ Punitive Damages	99.11	90.0	99.53
$Aggravated Damages	100	44.44	97.77
$Future Care Costs	90.18	76.71	96.67
% Reduction	92.86	87.8	93.99

Table 6: Overall Accuracies Using Logistic Regression

In Tables 5 and 6 immediately above we have filled, empty, and overall accuracy results for our two classification methods - rule-based and statistical.

High Precision Statistical Classification Results - Damages

Columns	Overall Accuracy (%)	Filled Accuracy (%)	Empty Accuracy (%)
$ Damages total	69.44	70.4	0
$ Pecuniary Damages	35.04	37.27	0
$ Non-Pecuniary Damages	87.5	85.11	92

$ General Damages	69.44	54.35	96.15
$ Special damages	68.06	58.49	94.74
$ Punitive Damages	98.61	66.66	100
$Aggravated Damages	98.611	66.66	100
$Future Care Costs	77.77	69.23	87.87

Table 7: High Precision Damages

As mentioned in the *Applying Damages Classifier to Unseen Data* sections above, we set a probability threshold that the predicted damage must meet in order to be assigned. In order to achieve high precision results, we added a keyword argument called *high_precision_mode*. If *high_precision_mode* is *True*, we only use predicted tags found in the text and do not add any sub-categories together. Adding sub-categories has shown to propagate errors leading to lower filled accuracies. For example, if a *total* damage value is not found in the text, instead of adding the *pecuniary* and *non-pecuniary* damage results to get a total, we ignore the *total* field (set it to *None*). As shown in Table 9 above, this is shown to result in more accurate filled values. That said, the more strict our probability threshold, the less values we use from our classifier. We modified the threshold to maximize accuracy as well as the number of values being predicted in each category, specifically the *total damages* category.

The motivation for creating a *high_precision_mode* option for assigning values is to verify how well our model performs without adding sub-categories. High precision results allows us to more confidently analyse trends over time. In the *Analysis* section below however, we did not wind up using the high precision filter because it limited the amount of data available for analysis.

Comparing Damages Statistical Classifier Models - Damages

Classifier	Overall Precision (weighted)	Overall F-Score (weighted)
Logistic Regression	0.68959	0.6483
Random Forest	0.72247	0.5888
XGBoost	0.69038	0.6116

Table 8: Final Results for comparing classifiers

Logistic Regression Damages Classifier Results

category	precision	recall	f1-score	support
aggravated	0.93	0.62	0.74	21
future care	0.74	0.78	0.76	241
future wage loss	0.71	0.72	0.71	237
general	0.69	0.53	0.6	87
intrust	0.6	0.32	0.41	38
non pecuniary	0.8	0.85	0.83	343
other	0.87	0.95	0.91	2980
past wage loss	0.73	0.73	0.73	231
punitive	0.9	0.9	0.9	39

special	0.74	0.8	0.77	256
sub-future care	0.53	0.13	0.21	78
sub-future wage loss	0.5	0.19	0.27	37
sub-general	0.33	0.09	0.14	23
sub-intrust	0.83	0.36	0.5	14
sub-non pecuniary	0.73	0.38	0.5	21
sub-past wage loss	0.00	0.00	0.00	37
sub-special	0.46	0.2	0.28	153
sub-total	0.86	0.4	0.55	15
total	0.67	0.66	0.66	203
total after	0.71	0.2	0.31	25
accuracy	NA	NA	0.81	5123
macro avg	0.65	0.46	0.51	5123
weighted avg	0.79	0.81	0.8	5123

Table 9: Classification Report - Best Results, Damage Classifier

In Table 9 above, the performance of our Logistic Regression classifier using all low-level tagged damages is shown, broken down by category. Precision is the ratio of correctly predicted values of the predictions made for each category. Precision describes how well the classifier is performing for when predicting a certain tag. Recall tells us how many values our classifier is correctly tagging given all tags in a specific category. Recall answers the question: What ratio of tags are we guessing in each category given the total number of tags that should be guessed? F1-score is a weighted ratio between precision and recall. Support is the total number of tags in a specific category. Above, you can see we have a total of 5,123 tags where 21 are of type Aggravated, 241 are of type Future Care, etc. This table tells us how well our model is performing on using all low-level annotated tags. Of the 5,123 tags used in analysis above, they are only representing 205 annotated cases with damages awarded.

Ablation Results - Damages Statistical Classifier

Feature Removed	**Overall Precision (weighted)**	**Macro F-Score(weighted)**
ALL Features	0.68959	0.6483
Float Bins	0.66183	0.6303
start_idx_ratio	0.68491	0.6471
prev/next word, bigram and trigram	0.61969	0.5918
BOW before	0.68646	0.6423
BOW after	**0.68959**	**0.6521**
BOW (unordered)	0.68958	0.6383
headers	0.67706	0.6289

Table 10: Feature Ablation - Damages

Note: The following features were ignored in the weighted calculation - *other, total after,* all *reduction* categories. Best results in bold font

Ablation Results - Percent Reduction Statistical Classifier

Feature Removed	**Overall Precision (weighted)**	**Overall F-Score (weighted)**	**Overall Accuracy**
ALL Features	0.62191	0.6281	0.81
Float	0.60723	0.61337	0.81
Float and Bins	0.56792	0.56924	0.8
Bins	0.60716	0.60845	0.81
start_idx_ratio	0.62680	0.63294	0.82
plaintiff and reduction	0.59736	0.59865	0.8
defendant and reduction	0.61208	0.60845	0.81
reduce lexicon	0.62683	0.61828	0.81
defendant mentioned	0.61211	0.61825	0.81
plaintiff mentioned	0.62191	0.62805	0.81
bow (unordered)	0.54921	0.55630	0.77
bow after	0.62191	0.63789	0.82
bow before	0.60719	0.62317	0.81
lemmatize	0.62686	0.63300	0.82
lemmatize, start_idx_ratio	**0.64155**	**0.64769**	**0.83**

Table 11: Feature Ablation Results, Not Including 'Other' category. Best results in bold font

Contributory Negligence Classifier Evaluation

After feature engineering and hyperparameter tuning, the final performance across the three percent reduction statistical classifiers tested can be found below in Table 6 :

Classifier	Overall Precision (weighted)	Overall F-Score (weighted)	Overall Accuracy
Logistic Regression	0.64155	0.64769	0.83
Random Forest	0.62766	0.58195	0.8
XGBoost	0.65710	0.56617	0.81

Table 12: Comparing Performance Across Percent Reduction Classifiers

For the same reasons discussed in the *Damages Classifier Evaluation* section, we decided to move forward with the Logistic Regression classifier for our analysis. Below in Table 7 is a more detailed chart of the optimized Logistic Regression results using the low-level annotations evaluation process as mentioned above.

Contributory Negligence Logistic Regression Results

category	precision	recall	f1-score
cnd	0.69	0.72	0.71
cnp	0.59	0.56	0.58
other	0.9	0.9	0.9
sub-cnd	0.71	0.83	0.77
accuracy	NA	NA	0.83
macro avg	0.72	0.76	0.74
weighted avg	0.82	0.83	0.82

Table 13: Classification Report - Percent Reduction Classifier

Analysis

One of the main goals of this project was to analyze BC's negligence cases and see whether the trend of damages awarded was increasing or not since the year 2000. Our evaluation shows that the statistical classification method is the most accurate and precise approach (Table 8). Note that our analysis is performed only on cases where the plaintiff won. Meaning when we calculate values such as the median we are only considered cases where damages are actually awarded or liability is apportioned. Out of the different classification algorithms used to predict damages such as Decision Tree, Random Forest, XGboost, LGBM, SVC, and Logistic Regression, the latter generated the most promising results (specifically for Total Damage category) with an overall accuracy of 68% and F-score of 64% as shown in Table 6.

The predicted results with the Logistic Regression classifier shows that there is an increasing trend in the median of Total Damages awarded per year after being adjusted for inflation (Figure 9). The smoothed red line in Figure 9 is after applying a smoothing function to the data. We applied a sliding window technique where we replace the individual median values with an average of itself and its neighbours. The window size was set to 7. We see the smoothed median amount start at approximately $150,000 CAD (adjusted to currency value in 2020) in the year 2000 and end at around $225,000 CAD in the year 2019. There is a slight dip in the median amount between the years 2005 and 2008 where the smoothed value hovered around $125,000 CAD.

From visually inspecting the trend of the median of Total Damage awarded per year, it can be seen from Figure 9 that the Total Damages awarded have been increasing over the past 20 years (mean of total damage awarded per year in appendix 5). To confirm that the results are not due to chance, we performed statistical analysis, namely calculating Pearson correlation and Regression analysis to test the validity of the predicted results.

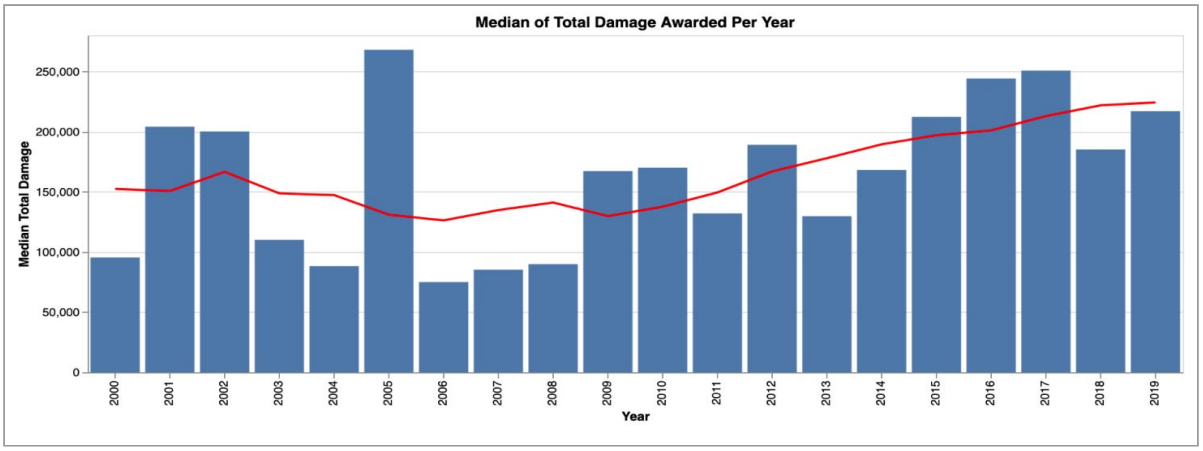


Figure 9: Illustrates the median of total damages over years adjusted for inflation, layered with smoothed line plot (red, smoothing window size 7) of the median of Total damages per year.

The Pearson correlation coefficient measures the linear correlation between two variables X and Y. The coefficient has a value between +1 and −1, where 1 is total positive linear correlation, 0 is no linear correlation, and −1 is total negative linear correlation. The Pearson correlation coefficient between the year and Total Damage awarded is 0.73 which indicates a strong positive correlation between year and damage awarded. In addition, we performed linear regression and calculated an associated p-value, which is a type of inferential statistics[10]. Inferential statistics focuses on the strength of the relationship between two or more variables. Linear Regression analysis[11] assumes independence between independent variables and a dependent variable and tries to prove otherwise. In this case, the null hypothesis for regression analysis would be that the year and Total Damage are independent of one another therefore, the slope of Total Damages over the years would be zero, or a flat line. Regression analysis outputs a p-value[12] which is "the probability of obtaining a test statistic just as extreme or more extreme than the observed test statistic assuming the null hypothesis is true". The test statistic is what the change in the median would be if we assumed that total damages have not changed over the years which would be 0. The observed test statistic would be the actual change in total damages over the same years which would be above 0. The p-value obtained was 0.00014, which is smaller than the alpha value[13] of 0.05 meaning the result is significant. Significant results suggest that the null hypothesis is rejected and the slope between the year and Total Damage awarded is not 0 which confirms that there is a dependence between the year and Total Damage. Due to the regression analysis, the predicted Total Damages awarded using the Logistic Regression classifier show that the trend of Total damage awarded in BC negligence cases has increased from the year 2000 to 2019.

The median of the damages awarded for other subcategories such as Total Pecuniary, Non Pecuniary, Special, General, and Future Care were also analyzed and the results showed that most types show an increasing trend over years (Figure 10).

[10] Inferential Statistics
[11] Linear Regression analysis
[12] p-value
[13] alpha value

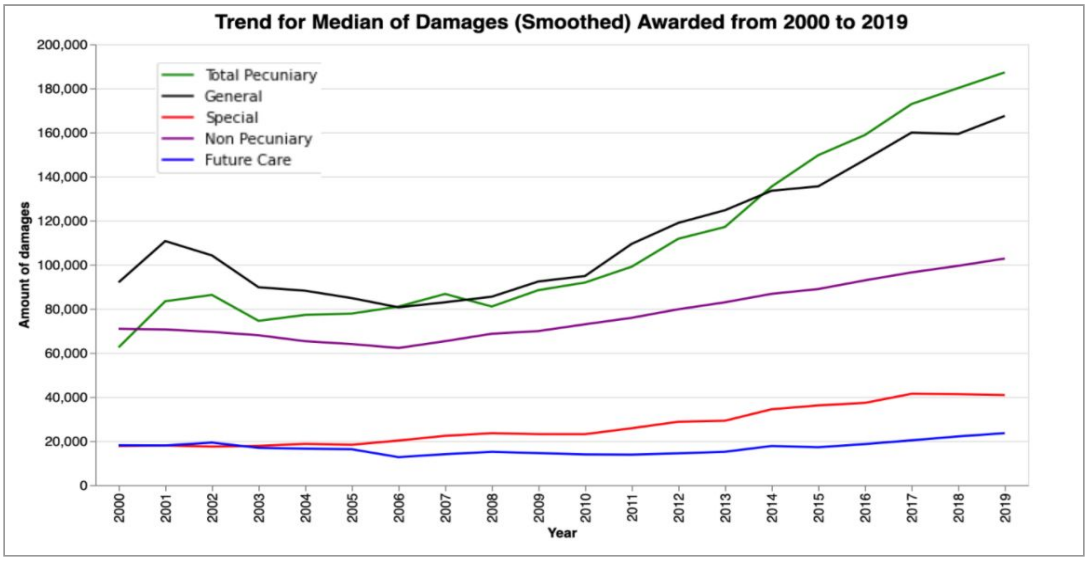


Figure 10**:** Shows a line plot for a median of General, Non Pecuniary, Total Pecuniary, Special, and Future Care damages from the year 2000 to 2019, smoothed with a window of 7.

The trend of the lines suggests that the median damages awarded are increasing per year for Non Pecuniary, Special, and General damages. Special and Future Care damage categories have a less obvious increasing trend, therefore, statistical analysis was performed to see if the results had significance. The results are shown in Table 14.

Damage Type	Correlation	P-value	Significant?
Non Pecuniary	0.76	5.82×10^{-5}	YES
Total Pecuniary	0.79	1.72×10^{-5}	YES
General	0.59	0.0045	YES
Special	0.63	0.001	YES
Future Care	0.37	0.097	NO

Table 14: Displays correlation analysis and linear regression statistical inference p-value for damage types Non Pecuniary, Total Pecuniary, Future Care, General, and Special.

Non Pecuniary damage awarded has a correlation coefficient of 0.76 with the year and the p-value suggests that its slope is greater than zero. We can see from both the mean and median of Non Pecuniary damages per year that there is an increasing trend per year (Figure 11). It is less obvious with the mean because it is heavily influenced by outliers. We also observe the same dip in values from about 2005 to 2009.

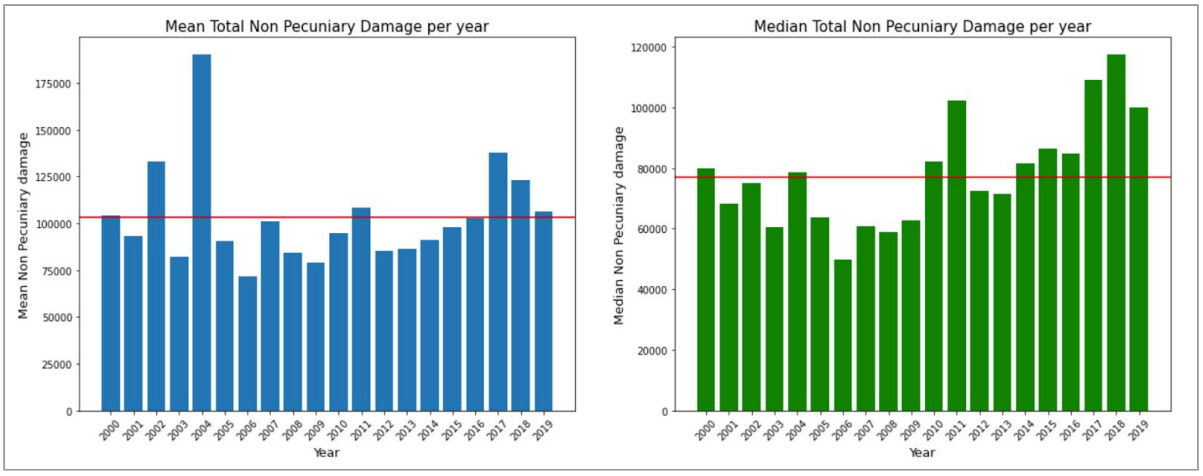


Figure 11: The mean (left), and median (right) of Non Pecuniary damages awarded from the year 2000 to 2019. Year on the x-axis and damage amount on the y_axis. Red line indicates the mean (left) and median (right) of damages over the years.

Total Pecuniary damage awarded has a correlation of 0.79 with the year and the significant p-value suggests that Total Pecuniary damage's slope is not zero. As we can see from the predictions as well, the median of Total Pecuniary damages has an increasing trend per year (Figure 12). Both Figure 11 and Figure 12 and the figures mentioned later, medians (right-hand sides) show a better visualization of the trend of damages than the mean since the median neutralizes the effect of outliers and gives a more clear understanding of the trends.

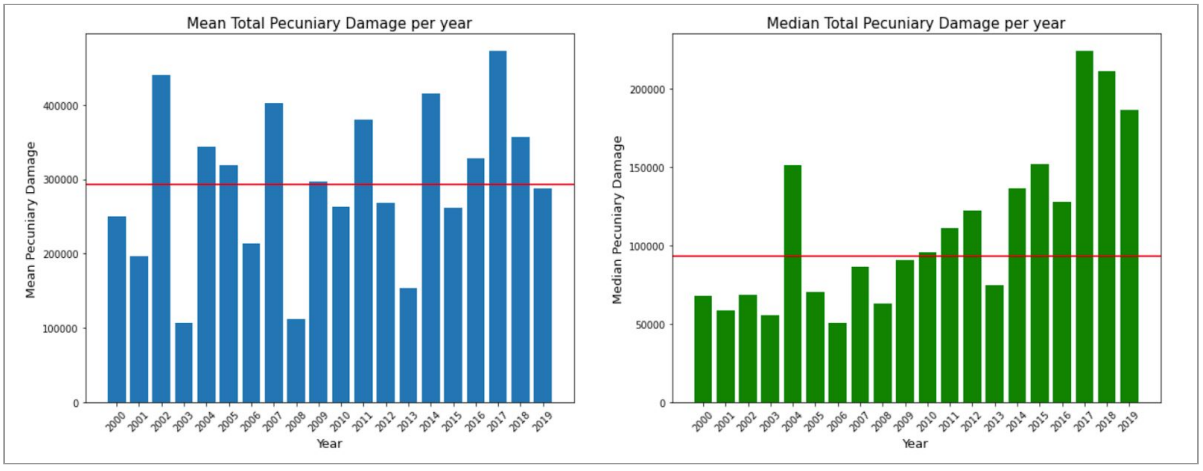


Figure 12: The mean (on the left), and median (right) of Total Pecuniary damage from year 2000 to 2019. Year on the x-axis and damage amount on the y_axis. Red line indicates the mean (left) and median (right) of damages over the years.

General damages have a positive correlation of 59% with the year and significant p-value suggests that damage's slope is non zero. As we can see from the predictions as well, the median of General damage has an increasing trend per year (Figure 13).

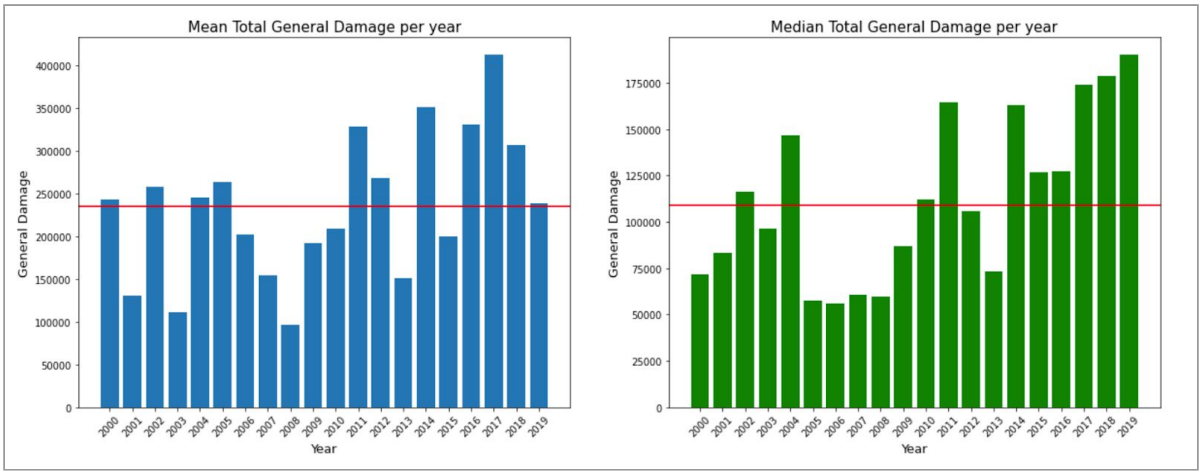


Figure 13: The mean (on the left), and median (right) of General damage from year 2000 to 2019. Year on the x-axis and damage amount on the y_axis. Red line indicates the mean (left) and median (right) of damages over the years.

Figure 10 clearly showed that Future Care and Special damages do not have steep slopes like Total Pecuniary and Non Pecuniary, however, there is a slight increase in the slope per year for both damage types. The p-value of 0.001 for the Special damage type suggests that the results are not due to chance and the trend is in fact increasing per year. As we can see from the predictions as well, Figure 14, the median Special damages awarded does have an increasing trend per year.

On the other hand, the p-value of 0.097 for Future Care Damages is bigger than the alpha value, which suggests that the results are not significant and the null hypothesis that slope = 0 can't be rejected. The slight increase in trend of Future Care damage that was evident in Figure 10 is only due to chance. As we can see from the predictions as well, Figure 15, the median of Future Care damage does not have an increasing trend per year and seems to remain steady. The reason that Future Care damage is not increasing over the years could be due to the fact that it is under the subcategory of general damages and is mostly awarded for accidental cases and depending on the injury, the damage amount can vary a lot.

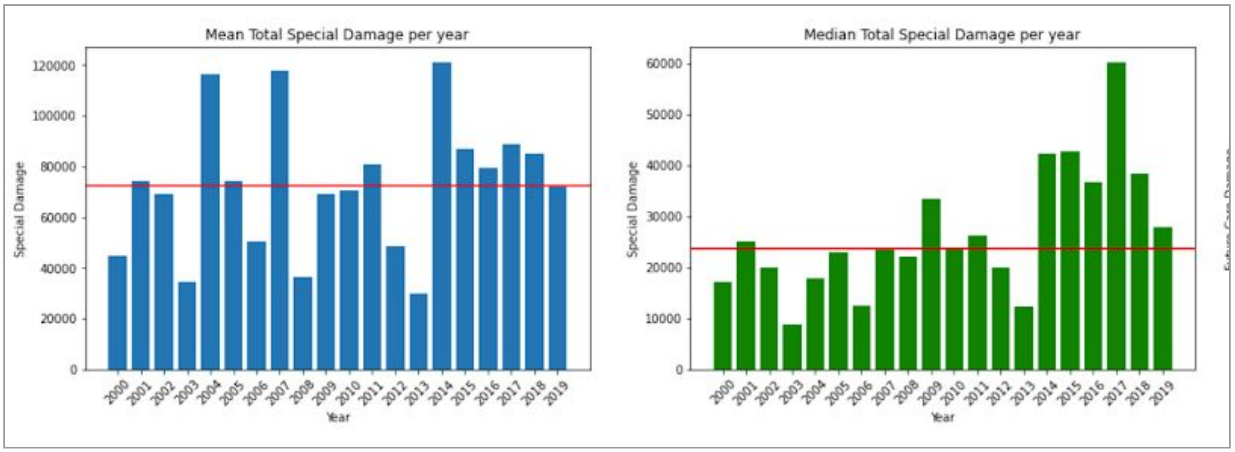


Figure 14: The mean (on the left), and median (right) of Special damage from year 2000 to 2019. Year on the x-axis and damage amount on the y_axis. Red line indicates the mean (left) and median (right) of damages over the years.

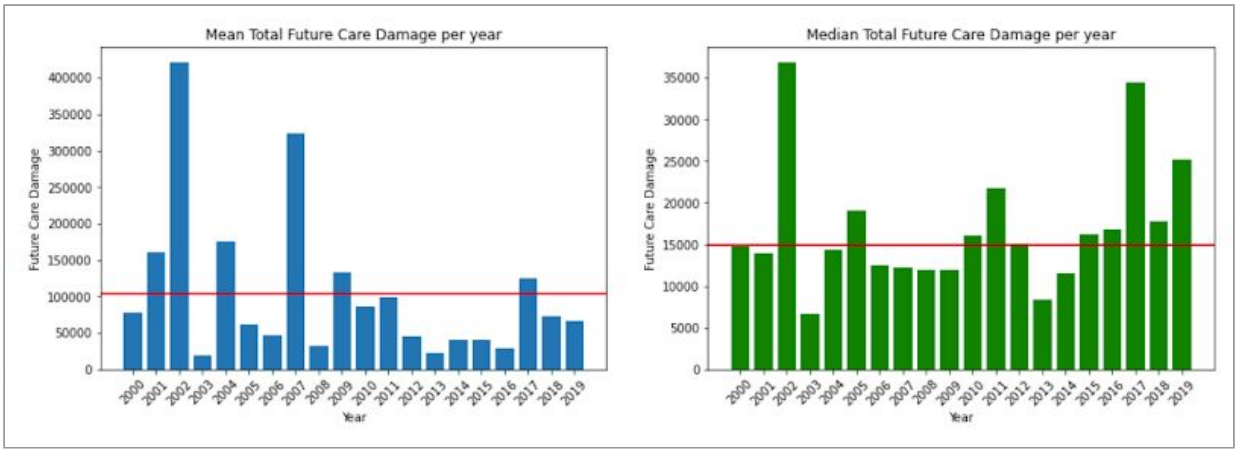


Figure 15: The mean (on the left), and median (right) of Future care damage from year 2000 to 2019. Year on the x-axis and damage amount on the y_axis. Red line indicates the mean (left) and median (right) of damages over the years.

Analysis of more rare types of subcategories of damages namely Aggravated and Punitive damages were done. Due to the rare nature of Aggravated and Punitive damages, there were few data points for each therefore, the result did not show any increase or decrease of trends over time. As we can see from the predictions as well (Appendix 6) there is no consistent trend for either of the subcategories.

Further analysis was done on the amount of damages predicted with the statistical classifier and the metadata extracted, such as the median of Total damage awarded for judges with at least 20 cases in the dataset (Figure 16).

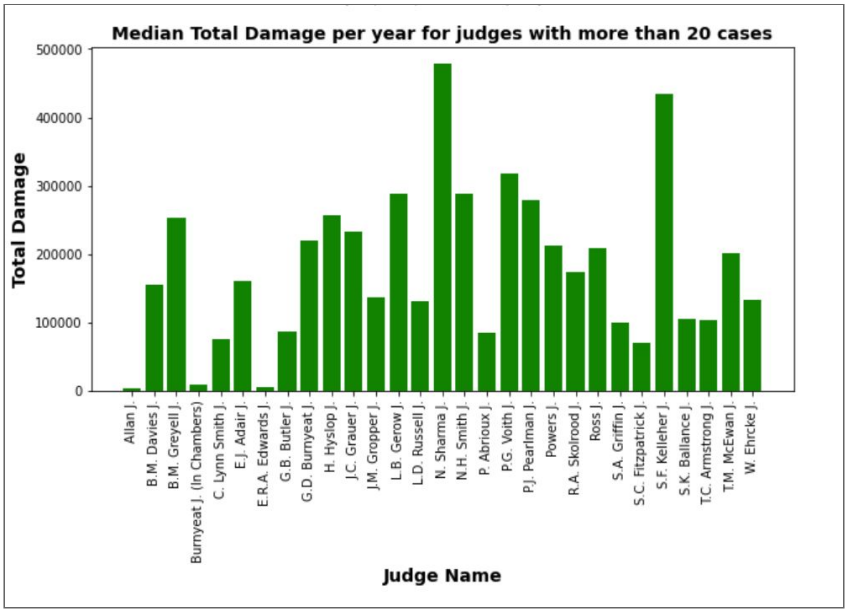


Figure 16: Median of Total damage awarded for judges with at least 20 cases in the dataset.

Figure 16 shows a high variance of the median of Total damage awarded across judges which might be due to the fact that some judges specialize in specialized areas and are assigned to specific domains. For example, one judge might be specialized in the accidental cases where mostly

future care damages awarded for a long period of time to compensate for the plaintiff's loss of ability which can increase the median of damage awarded for that judge.

An analysis of the mean and median of Total damage awarded for registries with more than 30 cases are shown in figure 17.

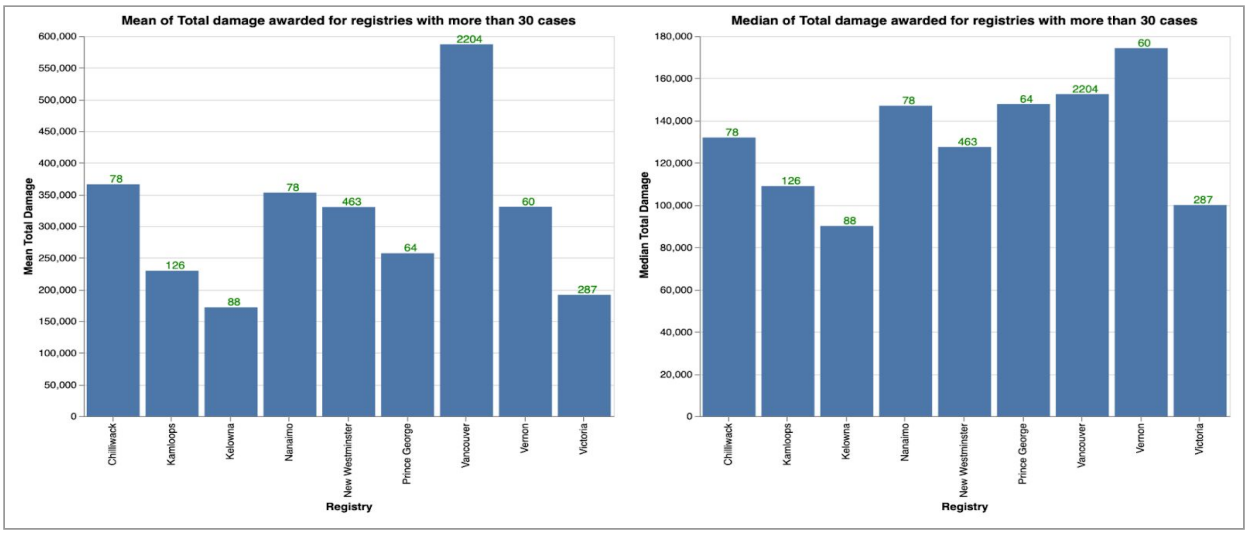


Figure 17: Mean(left) and median(right) of total damage awarded in registries with more than 30 cases. Numbers on top of the bars are counts of cases in each registry.

As illustrated in Figure 17, Vancouver has the highest mean of total damage awarded among registries with more than 30 cases and Vernon has the highest median of total damage awarded. This change of behavior between mean and median is due to the fact that the median eliminates the effect of outliers. Vancouver registry had few cases where a large amount of damage was awarded and caused such a drastic difference in the mean.

Cases in the dataset are labeled with a number of paragraphs which represent the case's length. Decision lengths over the years were analyzed and it showed an increasing slope (Figure 18).

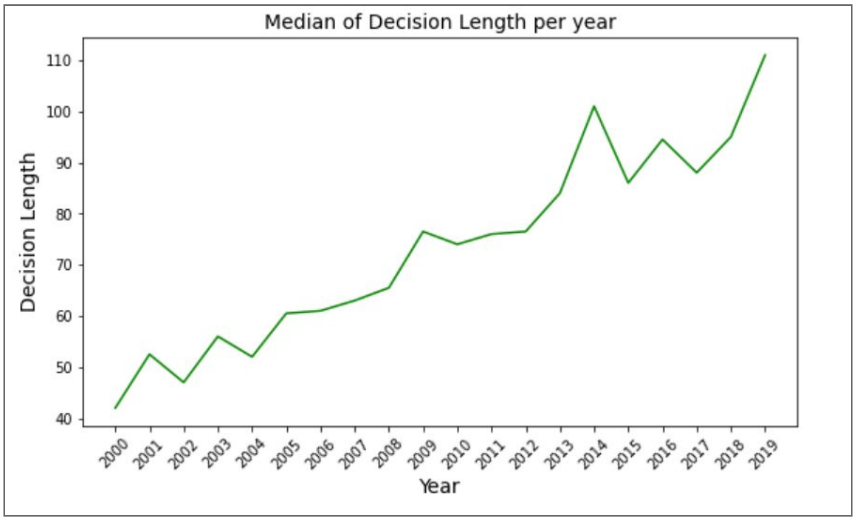


Figure 18: Median of decision length over the years. Median decision length on y-axis and year on x-axis.

This means that the cases have become more lengthy and explicit over the years and it seems to be increasing in the near future.

We also looked at the median of decision lengths for judges with 15 cases and more (Figure 19).

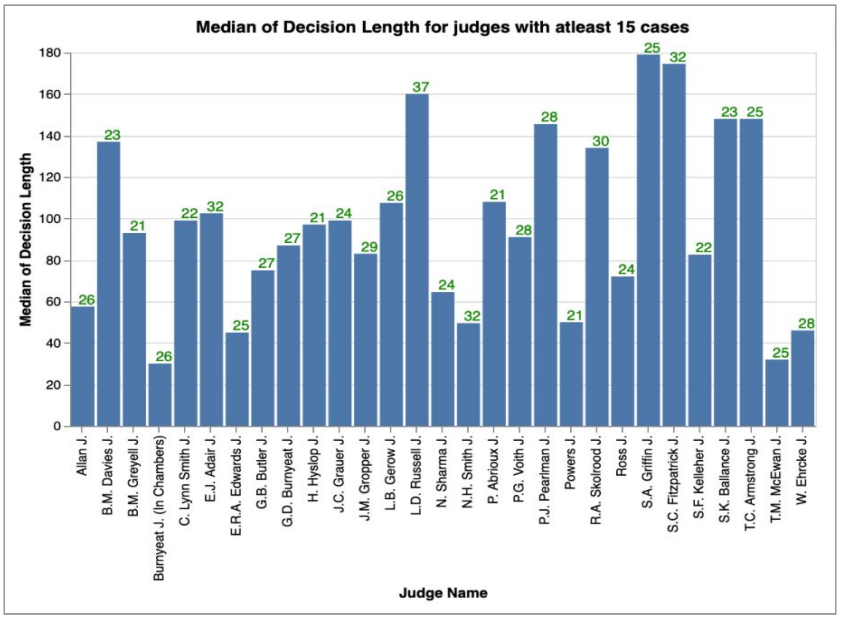


Figure 19: Mean decision length for judges with 15 cases or more. Bars are annotated with the number of cases for each judge in green number.

This high variance of decision lengths across judges, as explained before is due to the fact that some judges specialize in some specific domains and therefore might require more verbose explanation than others.

In negligence cases, if the plaintiff is at fault for some portion or all the damage caused, it is said that the plaintiff was found contributorily negligent. Contributory negligence is raised in a court case, when the Judge is considering whether or not the plaintiff is at fault. Once it's proven that the plaintiff is contributorily negligent some percentage of damage awarded is reduced to compensate for plaintiff's negligence. In this project predicted contributory negligence was analyzed to answer the research question "What percentage of cases have contributory negligence raised in them and what percentage were successful in contributory negligence?" (Figure 20 and 21)

Figure 20 illustrates the portion of cases where contributory negligence was raised (orange bars) over all the cases in the dataset (blue bars). Contributory is raised in a case when the plaintiff is susceptible to being at fault for some or all portion of the damage. On average 23% of cases have contributory negligence raised in them. Next, we looked at what percentage of the cases where contributory negligence was raised (the 23%) were successfully charged with contributory negligence.

Figure 21 displays a count of cases being successfully charged with contributory negligence (red bars) per year when contributory negligence was raised (orange bars). On average 34% of cases when contributory negligence was raised were successful with charging the plaintiff with contributory negligence.

The trend of the percentage of contributory negligence raised and successful in cases seems to be steady over time and do not show an increase or decrease over the years.

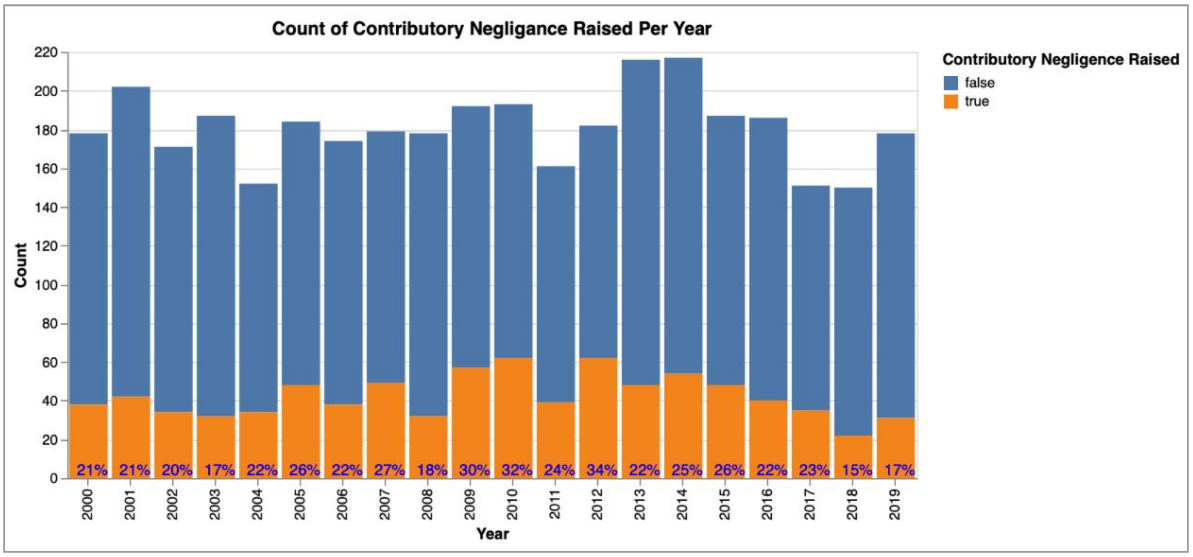


Figure 20: Count of contributory negligence raised per year (Orange) layed on top of count of contributory negligence not raised (blue) per year. The percentage of contributory negligence raised each year is marked in blue percentages on top of graphs.

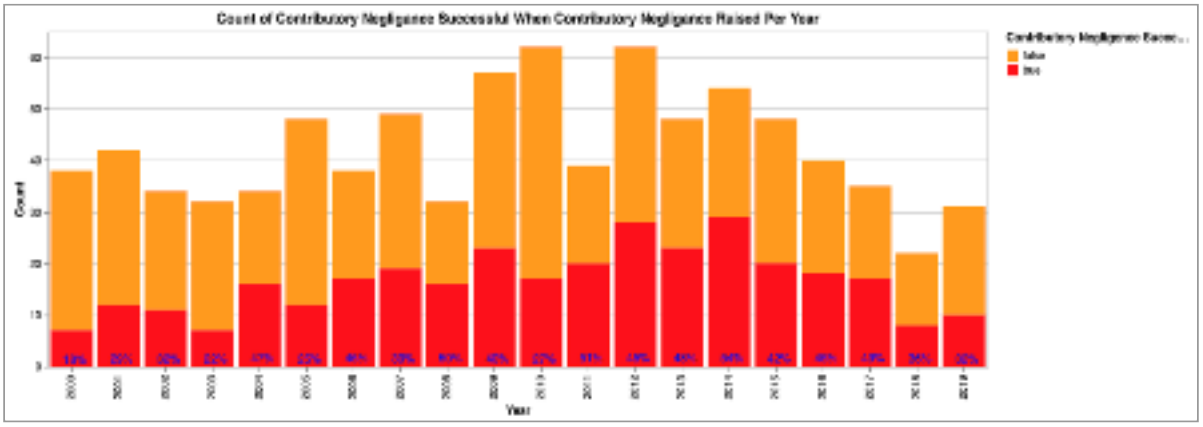


Figure 21: Count of cases where contributory negligence was successful when it was raised (orange) and contributory negligence was unsuccessful when it was raised (red). Percentages indicate percentage of cases where contributory negligence was successful when it was raised.

The mean percentage reduction when contributory negligence is successful was analyzed in Figure 22 and shows a ~35% mean percentage reduction over the years. This means on average when the plaintiff was charged with contributory negligence, a 35% reduction was applied to the total damage amount. To check whether there might be a decreasing trend in the mean of percent reduction over the years, linear regression analysis was performed. The 0.16 p-value suggests that the result is not significant and the mean percent reduction remains steady over the years.

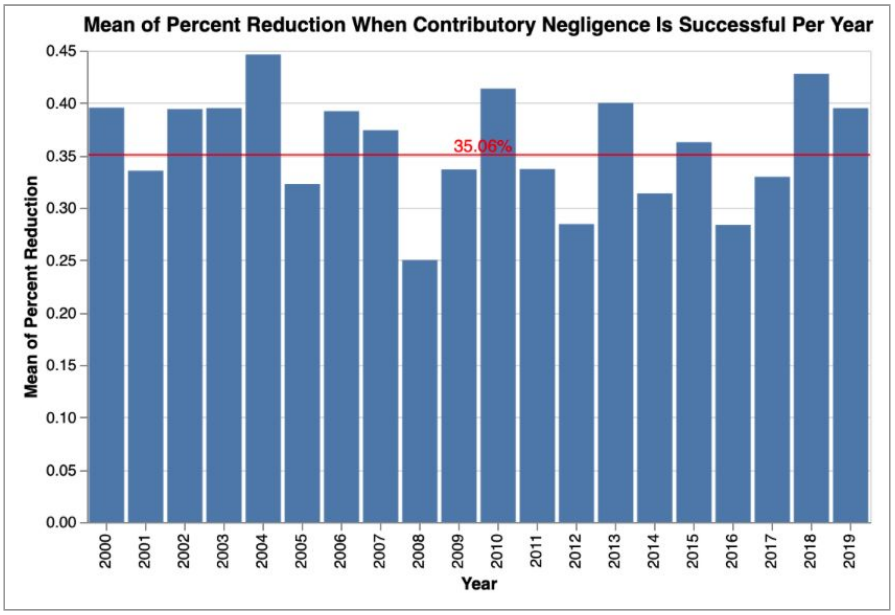


Figure 22: Mean of percentage reduction due to contributory negligence from year 2000 to 2019.

To conclude, the total damages awarded to plaintiffs has been increasing from 2000 to 2019. Sub category damages of Non Pecuniary, Total Pecuniary, General, and Special damages are increasing over time. The only sub category that did not increase over time is Future Care, which may be due to the fact that it is a subcategory of General Damages and it's only awarded in specific situations. On average 23% of cases have contributory negligence raised in them of which 34% were successful in contributory negligence. The trend of contributory negligence is steady and does not show and increase or decrease over the years. On average, 35% reduction was applied to cases where contributory negligence was successful and the percent reduction does not increase or decrease over time.

Future Work

Due to the time constraints related to the project we were unable to try some other approaches. This section is dedicated to listing other potential approaches to explore that may result in better model performance or more interesting results. Increased confidence in the model would result in increased confidence in the analysis stage.

Data

In the data section of this paper, the rules for filtering out types of cases that would not award any damages were listed. There may be other types of cases that fall into this category and could be removed from the overall dataset given enough time and resources. The amount of these cases will be low considering most of them have already been identified and filtered. Due to time constraints, there was no justification to continue searching for these case types due to the relatively low impact it would have on the analysis.

LexisNexis is our only source of case data for this paper. Another legal research resource where cases may be retrieved from is WestLaw. Given more time, our plan was to include cases from both resources which could help improve our model results and help with overall analysis. There may be some overlap between the two resources but we are simply looking to increase the number of unique negligence cases that are available digitally.

Another improvement that can be made is the order in which cases were annotated. Cases were selected manually, rather than randomly, to be annotated. Completing annotations took up approximately three to five days for each group member. This is due to the annotator needing to fully understand each case that they are annotating which can take extended periods of time depending on the case length and complexity. Since cases were selected from LexisNexis using a query there may be some ordering involved which was not taken into consideration. This is problematic because we may have annotated cases that are not fully representative of negligence cases in general. If there was any ordering involved it is always preferable to select cases at random because the annotations should be representative of the entire dataset. In the future, it would be preferable to randomize the assignment of cases to each annotator to get a more unbiased set of annotations. Furthermore, it would be preferable to annotate more than the 298 cases that were done in this paper. There were a total of 3,673 cases available to us which means that we had only 8.1% of the data annotated to train and test our machine learning model. It would have been preferable to increase this percentage to be between 15% and 20%.

Given more time, it would also be preferable for each team member to annotate the exact same cases. The benefit of this is that you can measure the disagreement between team members on the annotations. Another benefit is that the

Research Question

Our research questions solely focussed on cases in British Columbia. It would be interesting to expand this question into other jurisdictions to see if the trends found in British Columbia hold true across Canada. An analysis of cases across the country would allow for comparison between individual provinces.

Classifier

The methods section explains the use of a classifier based approach to perform information extraction. After optimizing our best classifier, the next step would be to try using an artificial neural network as our classifier. The idea behind an artificial neural network is to encode an input into a matrix of values and pass the values through a “network” which will apply different mathematical operations at each layer of the network. Each layer will update how it interacts with the input values as it is being trained with more and more data which will eventually result in a more reliable network.

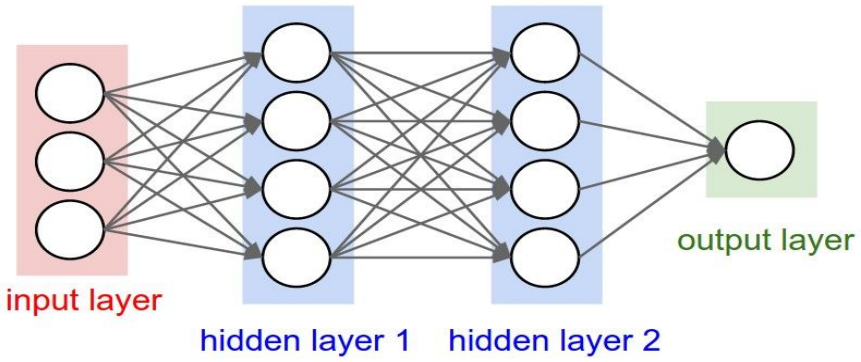


Figure 23 - Example Neural Network (image source)

Artificial neural networks perform best with large sets of data. The overall approach would remain the same as our current method, the goal would be to feed the network a sentence or paragraph into the input layer containing a monetary value or percentage we are trying to identify, which would be predicted in the output layer. We most likely do not have enough data to be able to harness the power of neural networks for this project. The main prerequisite to using neural networks would be to collect a lot more data.

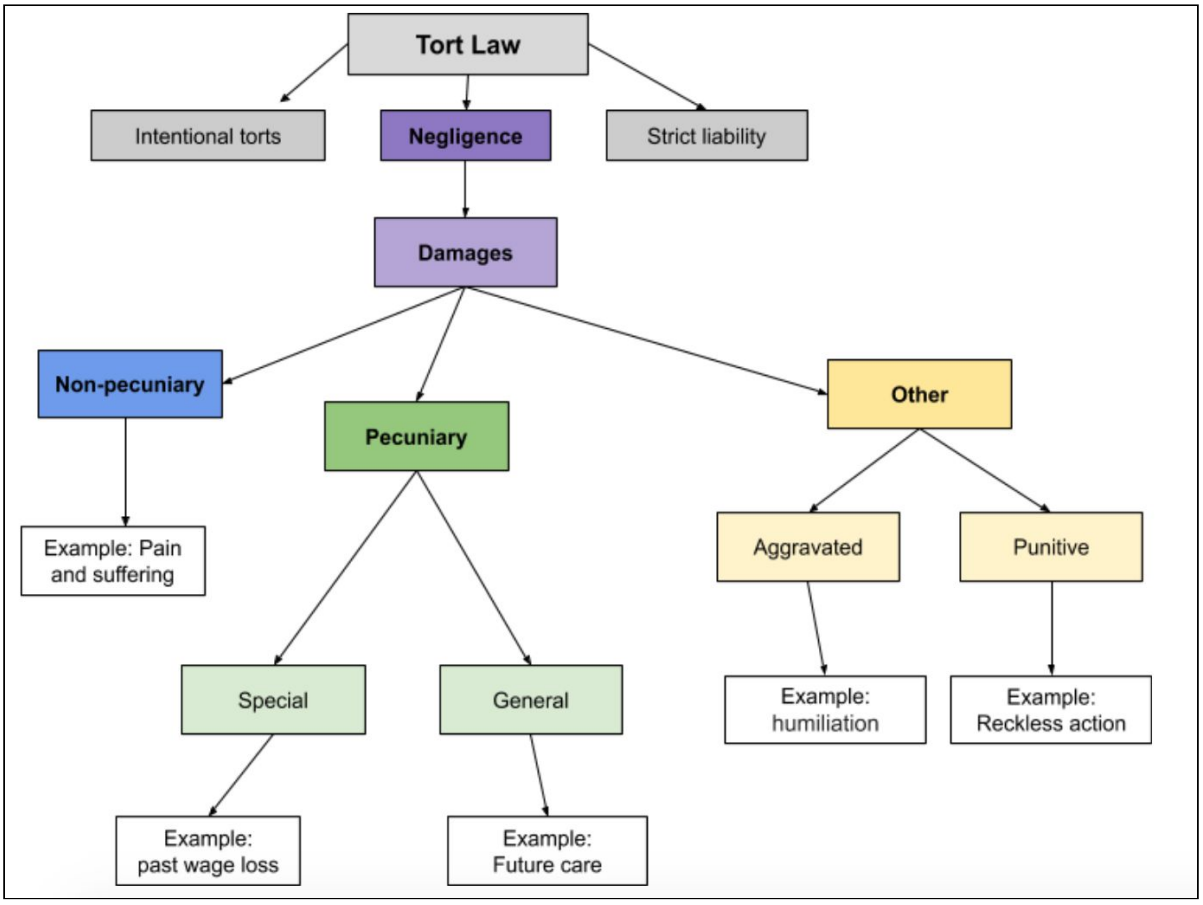
Timeline

At the beginning of the project, our team laid down a 5-week schedule for the tasks that require completion in order to deliver a successful outcome for the project. For the most part, we were able to follow our weekly schedule but we were ahead of schedule for some tasks and behind for others. In Appendix 6 we have included our updated timeline which highlights the main tasks completed by each group member on a weekly basis. Most experimental tasks or time spent researching was not included in the table for readability purposes.

Appendix

Appendix 1:

The flow chart illustrating the sub categories of tort law and damage types of negligence category.



Flow chart of branches of BC law and the damages for tort law division.

Appendix 2:

Terminology:

Bag-of-words (BOW): A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things: A vocabulary of known words. A measure of the presence of known words.

Contributory Negligence*:* "Failure of an injured plaintiff to act prudently, considered to be a contributory factor in the injury suffered, and sometimes reducing the amount recovered from the defendant."

Defendants: A defendant is a person accused of committing a crime in criminal prosecution or a person against whom some type of civil relief is being sought in a civil case.

P-value*:* "The probability of obtaining test results at least as extreme as the results actually observed, assuming that the null hypothesis is correct."

Bootstrapping*:* Bootstrapping is the process of random sampling from a subset of data, many times, to create a distribution of values for a certain statistic.

95% Confidence Interval*:* A range of values that captures a certain statistic for a dataset 95 percent of the time.

plaintiff: A plaintiff is a party who initiates a lawsuit.

Regular expressions: "Regex is a string of text that allows you to create patterns that help match, locate, and manage text."[14] For example, if we want to identify the amount in the previous example this regex "\$[0-9]+?,[0-9]{3,}" will pull out $20,000.

Statistical Inference: "The theory, methods, and practice of forming judgments about the parameters of a population and the reliability of statistical relationships, typically on the basis of random sampling."

Train set: is the material through which the computer learns how to process information

Development set[15] : "A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's hyperparameters."

Test set: data set that is used to test a machine learning program after it has been trained on an initial training data set

[14] Regex Definition
[15] Development set

Appendix 3:

Case Format Example

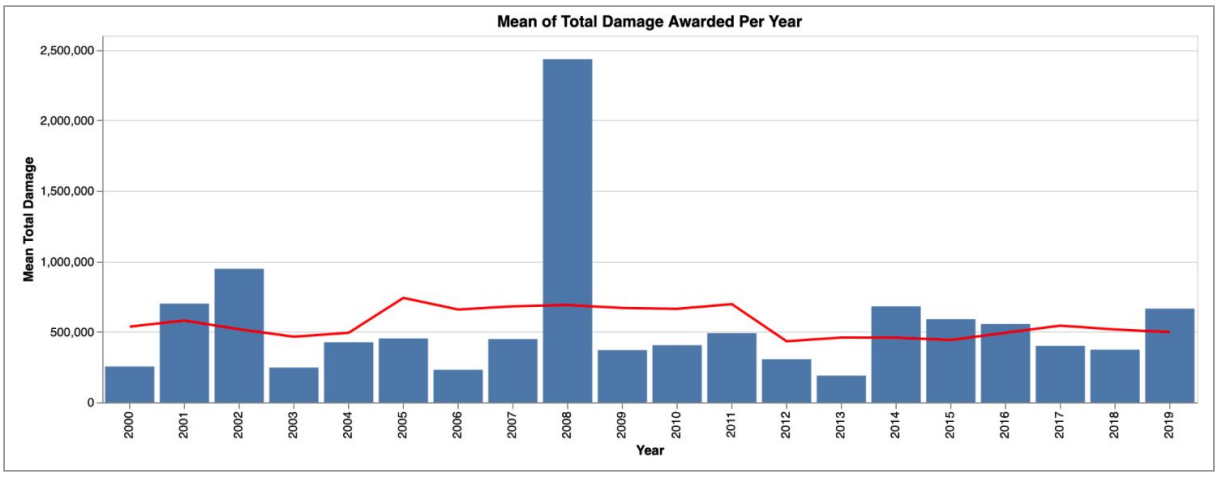
Mclaren v. Rice, [2009] B.C.J. No. 2108
British Columbia Judgments
British Columbia Supreme Court
Vancouver, British Columbia
T.R. Brooke J.
Heard: November 24-28 and December 11, 2008.
Judgment: October 26, 2009.
Docket: M063770
Registry: Vancouver
[2009] B.C.J. No. 2108 | | |
Between Matthew Robert Joseph Mclaren, Plaintiff, and Jacob John Rice, Michael John Rice and Hilltop Gardens Farm Ltd., Defendants
(46 paras.)
Case Summary
...
...
HELD: The defendants were jointly and severally liable.
There was no evidence of any brake marks on the road...
...
...
Reasons for Judgment
T.R. BROOKE J.
1 This action arises out of single-vehicle accident, which occurred on the morning of February 26, 2005, when a three-quarter ton Ford truck owned by the defendants, Michael John Rice and Hilltop Gardens Farm Ltd., and driven by the defendant, Jacob John Rice, went off the left side of the highway, struck a wooden utility pole and rolled at least two times before coming to rest on its wheels. The issues of liability and quantum were severed and the only issue before the court is that of the defendants' liability, if any, and the contributory liability of the plaintiff.
2 ...
...
...
45 I am satisfied that the defendants have failed to establish on a balance of probabilities that the plaintiff was not wearing a seatbelt or that the seatbelt available to him was in working order, or if it had been in working order, that it would have prevented or reduced his injuries.
46 In the result, I find the defendants jointly and severally liable to the plaintiff for such loss and damages he sustained as a result of the accident. Given this result, I do not consider that it is necessary for me to remain seized of the assessment of damages. The plaintiff will have his costs at Scale B.
T.R. BROOKE J.
End of Document

Appendix 4:

An example of case annotation at high level:

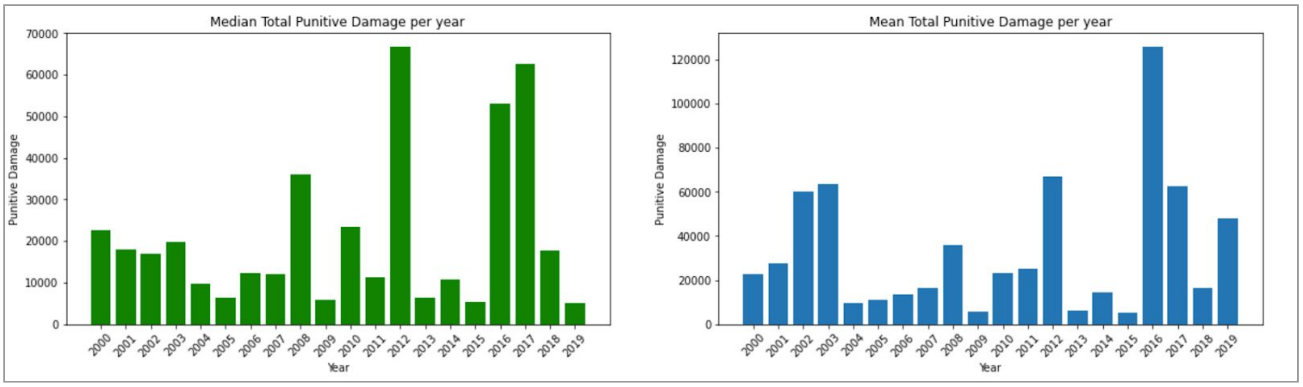
Case Name	Graham v. Lee, [2004] B.C.J. No. 2052
Written Decision?	Y
plaintiff Wins?	Y
Multiple defendants?	N
Judge Name	C. Lynn Smith J
Decision Length: paragraphs)	56
Registry	Vancouver
$ Damages total before contributory negligence	215000
$ Non-Pecuniary Damages	90000
$ Pecuniary Damages Total	125000
$ Special damages Pecuniary (ie. any expenses already incurred)	0
Future Care Costs (General Damages)	0.00
$ General Damages	125000
$ Punitive Damages	0
$Aggravated Damages	0
Contributory Negligence Raised?	Y
Contributory Negligence Successful?	Y
% Reduction as a result of contributory negligence	50
$ Reduction as a result of contributory negligence	43000
$ Final Award after contributory negligence	172000

Appendix 5:

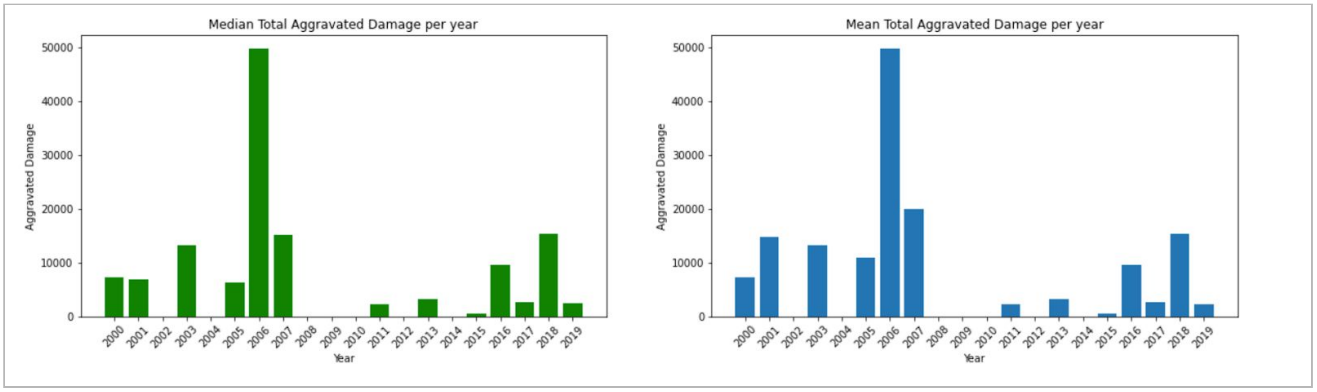


Mean of total damage awarded per year, layered with smoothed mean of total damage awarded per year with window 7

Appendix 6:



The mean (on the left), and median (right) of Punitive damage from year 2000 to 2019. Year on x-axis and damage amount on y_axis.



The mean (on the left), and median (right) of Aggravated damage from year 2000 to 2019. Year on x-axis and damage amount on y_axis.

Appendix 7:

Timeline of Events

	Ravi	Ilana	Niki
Week 1 (May 4 - 8)	Completed expected deliverables section of project plan. Completed conversion of .DOCX to .TXT Began work on determining if there are multiple defendants in a case (rule based) Extracted some static/non-moving information (judge, registry location)	Completed methods section of project plan. Began work on extracting extracting contributory negligence percentages (rule based) Extracted some static/non-moving information (case title, year)	Completed description of problem, dataset, and schedule sections of project plan. Began work on extracting if plaintiff wins and whether the case has a written decision (rule based)
Week 2 (May 11 - 15)	Completed extracting multiple defendants (rule based) Completed first pass of damage extraction (rule based) Created evaluation function for future use	Completed first pass of contributory negligence extraction (rule based)	Completed extracting plaintiff wins (rule based) Added 100 additional annotations for our final .CSV format
Week 3 (May 18 - 22)	Annotated damage tags in ~43 cases First Pass Feature Engineering Tree based classification for damages (XGBoost)	Annotated damage tags in ~43 cases First Pass Feature Engineering Linear classification for damages (Logistic Regression & SVC)	Annotated damage tags in ~43 cases First Pass Feature Engineering Tree based classification for damages (Decision Tree & Random Forest)
Week 4 (May 25 - 29)	Updates to evaluation methods to include more fine-grained info Convert our data into .CSV that matches external partners format First pass classification for contributory negligence	Bag of words feature engineering Further work with tuning damage classifier with Logistic Regression (best model)	Annotated percentage tags in 129 cases Begin work on generating visualizations based on results so far
Week 5 (June 1 - 5)	Filtering & excluding newly discovered irrelevant cases	Fine tuning contributory negligence classifier using Logistic Regression Moving code from Jupyter Notebook into .PY file	Further work on generating visualizations Created presentation for Lachlan

Week 6 **(June 8 - 12)**	Begin on summary, data, and timeline sections of final report Create some visualizations	Begin on methods section of final report Proving statistical significance & plotting distributions	Begin on analysis & evaluation & background section of final report More work on generating visualizations for final report
Week 7 **(June 15 - 19)**	Added final features for slight improvements to classifier Writing future work section of report Some proof-reading & minor edits of report	Ran ablation study on features	Final report review Creating presentation slides
Week 8 **(June 22 - 26)**	Report, presentation, and finalizing data product	Report, presentation, and finalizing data product	Report, presentation, and finalizing data product