Exploration of Classification in the Legal Domain

*Abstract* — In this paper, I investigate the definition of argumentation along with its importance in the legal domain. I explain the process of annotating argumentation from legal corpora in the Caselaw Access Project (Harvard Law School) and how my team and I generated an annotation guideline. I utilized a variety of machine learning classification algorithms to attribute sentences from a court case to one of the argument classes. I investigate any sources of noise or imbalance in our data and how those affected the models. In the final version of this project, I explain how I developed a multi-layer classification pipeline that has proved in achieving higher accuracies. I will also discuss any potential issues that could be raised from the continuation of this research project and effectively conclude with future work. The codebase is located [here](https://github.com/UNT-IS-REU/argument-mining).

# Introduction

In this age of digital deluge, information is pervasive and dense for a single person to consume in a short amount of time. Certain professions, such as law, fall under this condition, in terms of the tedious amount of legal research. Law professionals must parse through a multitude of cases and laws before undertaking a new case. must comprise arguments from past court cases and gain a holistic understanding from the case, to utilize that knowledge for future cases. However, time-consuming tasks like this can be aided with machines being a supporting system for these law professionals when it comes to parsing a vast amount of legal information. A law professional must be able to equip himself/herself all necessary details before taking a new case to court (even knowing the outcome of a case), to display the best case that he/she can. With the advancement of text classification being applied to numerous NLP tasks, we can finally break through applying NLP to the legal domain, a rather under-looked domain in the world of AI [6].

Before I explain the details of the research project, I must explain the definition of argumentation. An argument is a set of premises or a piece of evidence located in a corpus, offered to support a conclusion. To clarify, a conclusion is a claim made by someone who believes it to be true. So, argumentation is the process of constructing and evaluating these arguments with their connection to other arguments. Once all arguments are identified in a legal corpus, it takes the next step to analyze how each argument works in tandem to achieve a plausible conclusion, whether it is true or not. Taking a step further, there are chains of reasoning in argumentation, meaning premises and conclusions can be used to make further claims or conclusions to drive the case forward. This becomes important when we want to define the argument structure of legal corpora [3].

The data for this project is in the form of a published court report from the [Caselaw Access Project](https://case.law/), containing millions of cases from over the past centuries, in the United States. Each court report contains a summarization of how the court judges one case, which contains several reasoning components. I will be slightly modifying the definition of an argument, for this research project, to be a sentence belonging to one of the reasoning component categories. As mentioned previously, there are natural relationships between arguments and each published court report comprises a comprehensive structure of arguments. Usually, this form of argumentation can be defined as a graph structure, specifically a tree structure of arguments, where each node is an argument and each edge is a directional relationship.

The final goal of this project is to generate a reasoning structure for any published court report, based on the arguments detected and allow for easy readability and faster information retrieval. There is a total of 5 phases for this project:

1. ***classification of arguments in legal corpora***
2. ***generation of argument tree structure/database***
3. ***construction of a retrieval system for arguments***
4. ***generation of a report-like argument skeleton***
5. ***prediction of new court rulings using past data***

This paper will be mainly focusing on phase 1 (***classification of arguments in legal corpora***) out of the 5 phases and the results of these machine learning classifiers.

To train these classifiers, a labeled dataset will be required, labeling each argument (or sentence) from published court reports, under one of the 5 reasoning component classes: **Fact, Issue, Rule/Law/Holding, Analysis, and Conclusion**. My team and I were employed with the task of manually labeling these arguments under one of these 5 categories, to satisfy the requirements of a labeled dataset for the automation of classification of arguments. To label or annotate these sentences appropriately, the development of an intrinsic legal truth is required. This led us to construct an annotation guideline, to ensure consistency in annotator agreement, without resorting to naïve intuition. Evaluating the annotator agreement, using a selected agreement measure, allows us to judge the annotation quality. However, since annotation involves high-level semantics, labeling an argument under one general label can prove to be difficult at times, and can result in low annotator agreement. Discovering methods of quality control for the labeled data will be an important research task to achieve high accuracies in the classifications stage of this research project.

# related work

Compared to the numerous domains ML and NLP knowledge is applied to, the legal domain is an under-explored area where machines have the potential to play an important role in alleviating the arduous tasks committed by law professionals by a daily basis.

As argumentation is a crucial part of the overall project, [6] explains what argumentation is and how to identify it in legal corpora. They use the European Court of Human Rights (ECHR) data as legal corpora to classify argumentative and non-argumentative sentences in their argumentation structure(s). They were able to receive an 80% accuracy score in their binary classification task.

Corpus annotation is a crucial step for the annotation task, as a principle in machine learning states “garbage in, garbage out”. [4] understands to create an automated annotation system, manual annotation performed by 2 or more people is required, to ensure the quality of data is reliable for training. An important diagram in the paper states that for the highest of quality annotations, an annotator would require domain and linguistics knowledge and need to have a solid theory behind the annotation labels, to reduce ambiguity among annotators.

Potentially the most difficult phases to this research project are phases 3, 4, and 5, as they require a deeper study in the semantics and pragmatics of legal linguistics. A similar goal of implementation is thoroughly documented in [5], as their main focus is centered around “a ground truth for testing predictions about outcomes in new cases with new evidence; patterns for successful and unsuccessful argumentation; and guidance in retrieving, extracting, and organizing evidence for new arguments and new situations”. A creative approach that is taken for the construction of evidence-based arguments is a DeepQA architecture. This was the same architecture used to build IBM Watson for the game *Jeopardy*, except it is being used here to extract relevant arguments from past information using a relation extraction method.

In another part of the world, the same research project my team and I are researching is being conducted over in India by a group of Indian researchers. However, [2] is applying similar principles and goals to Indian law, instead. They used an annotator agreement system called Inter-Annotator Agreement (IAA), to judge the quality of the annotations, along with acknowledging the subjectivity existent in the labels. After the realization that hand-crafted features do not produce high results, the paper utilizes BiLSTM models, which achieves between 80-90% accuracy, for 7-class classification. It is possible to achieve similar results for U.S. law, using a similar process implemented by this paper.

# research questions and design

Before attempting the research on this project, some essential questions will need to be defined, to guide this project in the right direction. Any exclusion of these questions from the overall development of the project will stray away from the desired results. Every question is essential to every step in the project to ensure quality control of data and each phase is constructed correctly, as each incremental phase is dependent on its previous phase.

## Research Questions

As this paper is focused on mainly phase 1 of the research projects, the research questions listed below are only pertinent to phase 1 material:

1. ***Are there too many or too little labels/classes?***
   1. *Should we rename any current labels that could be better labeled than our current ones?*
2. ***What creates an effective annotation guideline?***
   1. *A simplistic guideline can offer consistency for annotator agreement but can introduce ambiguity when labeling certain sentences, without any additional rules to follow for classification*
   2. *A complex guideline can offer greater semantical understanding and reduce ambiguity in labels, however, if an annotator is not well versed in the subtleties of the domain*
3. ***How to handle imbalanced data to reduce bias?***
4. ***Effective classification algorithms in the legal domain?***
   1. *Any supporting algorithms or tools to improve accuracy models?*

All these questions were answered and thoroughly researched in the development of phase 1 of this project, which will be explained in later sections.

## Research Design

The rudimentary process of creating a labeled legal dataset and the workflow of training a classifier is shown below in the illustrated diagrams.

A screenshot of a cell phone

Description automatically generated

**Fig. 1.** Creating A Labeled Legal Dataset

****Fig. 2.** Creating A Labeled Legal Dataset

Figure 1 shows the general process of how our legal dataset was created. First, as mentioned, data needs to be retrieved from the Caselaw Access project to be annotated on (the period of the data will be consistent), then we need to use an argument (sentence) splitter to split the legal corpora into sentences. However, legal documents tend to not contain a specific structure at times and use a lot of periods when describing laws, abbreviations, references, etc. This can confuse a sentence splitter, hence not allowing a 100% sentence splitting accuracy (What happens with the invalid sentences will be explained later). Using an annotation tool, it provides ease for the annotators to quickly select the appropriate label for each sentence. After a batch of sentences (or stage) is labeled, an annotator agreement measure will be used to check if the quality of the annotations is reliable. This will give insight if everyone has differing definitions of the labels or not. And finally, through a personally selected metric, if the annotator agreement in each stage of data is at least 50%, we consider that data to have decent quality, which was can use for training. But, if it is lower than 50%, most likely the issue stems from having a poor annotation guideline that annotators are not familiar with or it is too complex or ambiguous to understand. A revision of the annotation guidelines will be deemed necessary in that case. After one pass of this workflow, this process will be repeated until enough, quality data is collected to train a classifier with promising results.

Figure 2 details the general process of what goes into pre-training, training, and post-training for classifiers (at least for our research project). As mentioned, to build our dataset, my team and I receive a batch of sentences consistently for us to label (usually about 500 sentences). These sets are called stages, which allows division in potential stages of poor annotation quality, leading to an updated annotation guideline for every consecutive stage. A stage of data will be used for training if it passes the annotator agreement metric defined. Preprocessing of the selected data will be required, as there are still potential issues, which will be discussed in the next section. From there, the basic workflow of machine learning is implemented. We utilize data augmentation to increase the number of training samples, as our current dataset does not have enough sentences to be proficiently trained on (*note: data augmentation is not implemented in the experiments, and is posted for future work)*. Next, using any form of vectorization or word embeddings transforms the data into a readable format by the machine. Once all training samples are ready to go, training the machine learning algorithm (classifier) would be our final step in generating a classification model. To ensure our model is doing well, testing for the model’s accuracy is a good metric. However, accuracy can sometimes be misleading, so other evaluation metrics would need to come into play to see how classification is occurring behind the scenes. The process of the model generation workflow repeats with another classification algorithm until we find an algorithm that correctly suits our dataset.

# data Acquisition, preparation, and modeling

## Data Acquisition

Even though this research project requires us my team and me to build our legal dataset, we must rely on another reliable dataset for quality published court reports, for us to analyze and classify labels. The [Caselaw Access Project](https://case.law/) contains over 6.7 million unique cases, collected over the last 360 years in the United States. The scope of the cases ranges from state to federal court cases, including territorial court cases. On top of the dataset being documented well, there is also freedom on what type of data do we want to train our classifiers on. However, to get high accuracy and consistent results, we have selected to take our court reports from 1960-1965. Containing our training data within a certain period ensures that legal semantics and law has not drastically changed, which could confuse the classification algorithm. Currently, our labeled dataset contains approximately 4,500 labeled sentences, with only a little over 1,000 sentences guaranteed with decent annotation quality.

## Preparation

To create a labeled legal dataset ready for classification training, several steps are required, as mentioned in the research design workflow.

We will need to divide the legal corpora obtained into sentences. We will be using a library called [LexNLP](https://github.com/LexPredict/lexpredict-lexnlp), a tool for working with legal texts, providing a variety of functions, such as sentence parsing. As all legal corpora are not constructed or written the same, it still proves difficult for even a machine to detect a sentence all the time. When using LexNLP, we find that at least 90% of the sentences are parsed correctly. This indicates that most of the legal corpora are still usable for labeling. The remaining 10% of invalid sentences will be used for another purpose.

Dividing the sentences among sub-teams will prove to be more efficient in labeling and time taken. Each sub-team receives a batch of sentences (or a stage) of 500 unique sentences weekly. Each annotator in a sub-team labels their assigned sentences with the guidance from an annotation guideline. The annotation guideline consists of definitions and examples of 7 labels, which are **Fact, Issue, Rule/Law/Holding, Analysis, Conclusion, Other, and Invalid Sentence**. The definitions of each of these labels are provided below.

* **Fact**: Any fact that is pertinent to the case.  This includes testimony, statements of record, case history, and anything else that is a fact that helps establish the foundation of the case for the court to build its analysis and judgment on. Anything that “sets the stage,” as it were, should be labeled as a fact.  Factual sentences do not include any synthesis or reasoning (that would fall under Analysis); rather, they simply state events or matters of record.
* **Issue**: Any issue or question that the court must decide.  This includes the overall issue of the case as well as any sub-issues that are raised in the case. Additionally, an issue supersedes Rule/Law/Holding; that is, if a sentence is an issue and it contains a rule, law, or holding, it should still be classified as an issue.
* **Rule/Law/Holding**: Any statement of or reference to a rule, law, and/or holding.  This includes sentences that reference a rule, law, or holding and then use it to reason through some point. Note that these sentences could be read as a fact, since whenever a law is quoted, for example, it is a fact that the law states that quotation. However, assign this label to sentences that directly or indirectly reference a rule, law, or holding except in the case of sentences that are issues.
* **Analysis**: Any sentence that synthesizes information to further the court’s reasoning.  This includes sentences that refer to the facts of the case and then uses them to push forward towards a resolution. Analysis sentences tend to move the court through the case from the facts towards the conclusion, so there is often a logical progression stringing analysis sentences together.
* **Conclusion**: Any sentence that effectively resolves an issue facing the court. Conclusions tend to be short and straight to the point, either agreeing or disagreeing with some argument.  Please note that as there are sub-issues, there are also sub-conclusions that conclude a specific sub-issue being discussed by the court.
* **Other**: Any sentence or phrase that does not fit the other labels in terms of its content.  This includes section headings. Sentences that fit this label tend to be sentences that do not add any information to the case and thus do not fall into any of the other labels.
* **Invalid Sentence**: Any sentence that is incomplete or does not have any logical flow to it. This category usually is reserved for sentences that were not split correctly using the argument splitter.

The first draft of the annotation guideline was inspired by a local resource, however, after labeling hundreds of sentences at a time, it was quickly realized that the initial annotation guideline seems to be useless or defective, in a sense. This enabled a process to review and update the annotation guideline frequently, to ensure annotator understanding and agreement when labeling.

The annotation tool used to label sentences with ease is called [Doccano](https://github.com/doccano/doccano), an open-source text annotation tool for humans. It significantly streamlined the process of uploading sentences, labeling, and downloading the sentences in a suitable data format. The benefit of using an annotation tool like this is the ability to see preceding and succeeding sentences. Understanding the surrounding context of a sentence helps an annotator make a better-informed decision of a label for a certain sentence.

After a stage of data is completed, it is evaluated for annotator agreement. The measure we use to judge the annotator agreement is kappa scores. We use Cohen’s kappa score when a sub-team contains 2 annotators and Fleiss’ kappa score when there are 3 or more annotators. A kappa score determines the agreement in labels between 2 annotators. The kappa score is applied to each pair of annotators on a sub-team in each stage of data. Through experimentation, a classifier can get good accuracy when the overall annotator agreement is at least 50% (overall in this case means that each kappa score in a sub-team is above 50%).

If all sub-teams’ kappa scores are above 50% (to calculate a sub-team’s kappa score, average all individual kappa scores within that sub-team), then that stage of data is acceptable for training. The data is then split into 3 main categories: *best\_labels, invalid\_sentences, and conflict\_labels*. The *best\_labels* category contains all the sentences that have a clear majority label on them (e.g. for a certain sentence, if the labels by the annotators are “*Fact, Fact, Analysis*”, then the majority label for that sentence is “*Fact*”). The *invalid\_*sentences category contains all invalid sentences. Separating the invalid sentences would be helpful to review any rules in those sentences that can be appended to the argument splitter, hopefully increasing the success rate of splitting sentences above 90%. The *conflict\_labels* category contains sentences that have no clear majority (e.g. “*Fact, Analysis, Issue*”or “*Fact, Fact, Issue, Issue*”). These sentences with no conclusive label cannot be used for training, however, they can be repurposed for updating the annotation guideline. Manually reviewing the conflict labels within each sub-team helps increase awareness in legal linguistics and semantics and produce a greater understanding of the legal annotation mindset. A sub-team can come to a unanimous vote on one label for a conflict label sentence. After all conflict label sentences have been discussed and voted with a majority label, an update to the annotation guideline with clearer definitions or clarifying examples, from the conflict label sentences, will be beneficial for quality control in the long run.

## Modeling

When choosing a classification algorithm, there was no one algorithm that stood out for our argument classification task. The best approach to seeing which classification algorithm would achieve the best results would be to try all the common ones. As for training and testing sets, the standard 80-20 split was done here on *best\_labels*, with random shuffle enabled (this not only randomizes the data but keeps the percentages of each argument class consistent, even when split). There were multiple iterations involved in the experimentation of various classification algorithms (*note: Iterations 1 and 2 were trained with Invalid Sentences, to prove a certain result*).

**Iteration 1**: In Iteration 1, the first 2 stages of data were completed (Stage 1 and 2), totaling 2,666 labeled sentences. Removal of any *conflict\_labels.* TFIDF vectorization was performed on all labeled sentences. The classification algorithms used were Decision Tree, kNN, Logistic Regression, MLP, Naïve Bayes, Random Forest, and SVM using the *sklearn*library. No hyperparameter adjustment. No data augmentation.

**Iteration 2:** In Iteration 2, 2 more stages of data were completed (Stage 3 and 4), totaling 1,214 labeled sentences. Removal of any *conflict\_labels*. Stages 1 and 2 were excluded from the training and testing sets. TFIDF vectorization was performed on all labeled sentences. The classification algorithms used were Decision Tree, kNN, Logistic Regression, MLP, Naïve Bayes, Random Forest, and SVM using the *sklearn*library. No hyperparameter adjustment. No data augmentation.

**Iteration 3:** In Iteration 3, stages 3 and 4 were used for training and testing sets. The *Other* and *Invalid Sentence* labels from the dataset were creating noise to the classifiers, which led to the creation of a multi-layer classification pipeline system. The first layer would perform binary classification on *Argument* (i.e. *Fact, Issue, Rule/Law/Holding, Analysis, Conclusion*) vs. *Non-Argument* (i.e. *Other, Invalid Sentence*). A new category was created called *argument\_*labels. The second layer would perform regular classification on all argument labels. TFIDF vectorization was performed on all labeled sentences. The classification algorithms used were Decision Tree, kNN, Logistic Regression, MLP, Naïve Bayes, Random Forest, and SVM using the *sklearn*library. No hyperparameter adjustment. No data augmentation.

# data analysis and result evaluation

## Data Analysis

The metric to judge the annotator agreement is kappa scores. Kappa scores tell us whether the annotator agreement in labels is good or not (defined by a 50% threshold). The table below provides all the kappa scores for each sub-team in each stage.

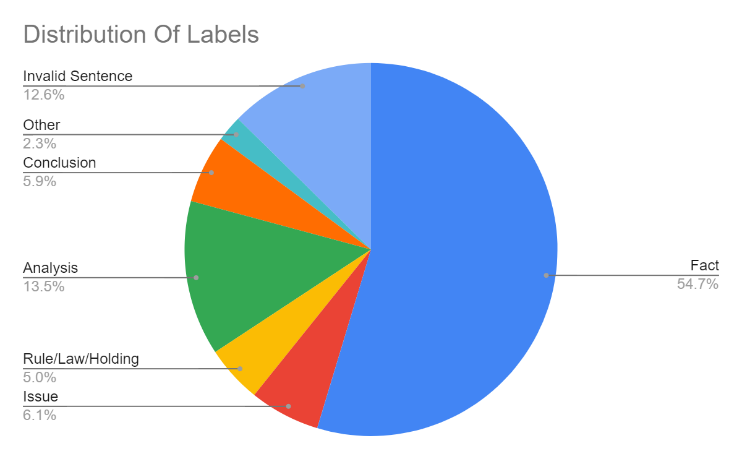
| Stage | Sub-Teams | | |
| --- | --- | --- | --- |
| Team1 | Team2 | Team3 |
| 1 | 0.1985 | 0.4369 | 0.3726 |
| 2 | N/A | 0.7265 | 0.4287 |
| 3 | 0.7189 | 0.5191 | N/A |
| 4 | 0.4936 | N/A | N/A |

**TABLE 1**. Kappa Scores For Each Sub-Team In Each Stage Of Data

In Stage 1, none of the sub-teams reached the 50% threshold. In Stage 2, Team 2 was able to receive a high kappa score, but not Team 3, which indicates there is still some misunderstanding in the annotation guidelines. After Stage 2, there was better consistency in reaching the 50% threshold, which means the continual updates of the annotation guideline was improving the overall annotation quality.

To understand what exactly needed to be updated in the annotation guideline, we had to look at what were the current underlying issues that lead to Stage 1 having sub-par kappa scores. After extensive study, we concluded with a couple of issues from the annotation guideline. First and foremost, our original inspiration for the annotation guideline was far too complex for under-experienced annotators to grasp. The definitions had legal jargon that was not familiar to all annotators and the examples provided did not correlate well with the sentences given for labeling. The initial confusion in labels can be attributed to a lack of experience in the legal domain, which leads to labeling based on naïve intuition. The second issue was there was a fair amount of ambiguity in the labels. At times, it would be hard to distinguish whether a sentence should fall under *Fact* or *Analysis*, for example. Even with the current annotation guideline, understanding the subtleties of the argument classes comes with experience in the legal domain, thus ambiguity cannot be removed completely, however, its effect can be reduced. These 2 main issues are primary factors in generating *conflict\_labels* for a multitude of sentences. After noticing these 2 prevalent issues, a major annotation guideline update was applied, with the input of the entire team, to reduce the impact of these issues, which lead to kappa scores nearing the 50% threshold or even beyond it. As there seemed more comprehension in annotating legal sentences, after the major update, Stage 3 and 4 data seemed the most stable to use for training and testing sets (even though Stage 4 data did not pass the 50% threshold, it approximately hits the threshold for decent annotation quality).

Another important point to mention (as defined in our research questions) is what to do if there is a case of an imbalance of data. After generating a distribution of data from all 4 stages, the following distribution is shown below:



**Fig. 3.** Distribution of Labels

Figure 3 shows that *Fact* is the biggest label in our labeled dataset, with a contribution of over half of the labels. This can introduce bias when a classifier is under training. However, fixing the problem of imbalance is not easy and will require further study on techniques to reduce bias, without affecting the nature of the labeled dataset. The experimentation results do not handle this issue for the moment.

## Result Evaluation

The results of each iteration training and testing are displayed in the form of tables. The leftmost column tells which classification algorithm is used and the right 2 columns provide the training and testing accuracy for each classification algorithm. Keep in mind that these are basic classifiers, as there has not been any hyperparameter tuning.

**Iteration 1**: The table below illustrates the results of Iteration 1.

| Algorithm | Accuracies | |
| --- | --- | --- |
| Training | Testing |
| Decision Tree | **0.9945** | 0.6745 |
| kNN | 0.7547 | 0.6526 |
| Logistic Regression | 0.8151 | 0.7025 |
| MLP | 0.9941 | 0.7274 |
| MultinomialNB | 0.6291 | 0.5935 |
| Random Forest | **0.9945** | 0.7243 |
| SVM | 0.9477 | **0.7430** |

**TABLE 2**. Model Accuracies Scores For Iteration 1

When it came to training accuracies, the Decision Tree and Random Forest classifiers performed the best, however, the SVM classifier performed the best when it came to testing accuracy, with its training accuracy still above 90%. The overall testing accuracies are average results but can still be greatly improved.

**Iteration 2:** The table below illustrates the results of Iteration 2.

| Algorithm | Accuracies | |
| --- | --- | --- |
| Training | Testing |
| Decision Tree | **0.9990** | 0.6705 |
| kNN | 0.7742 | 0.6899 |
| Logistic Regression | 0.7878 | 0.7054 |
| MLP | 0.9981 | **0.7287** |
| MultinomialNB | 0.6269 | 0.6357 |
| Random Forest | **0.9990** | 0.7054 |
| SVM | 0.9390 | 0.6860 |

**TABLE 3**. Model Accuracies Scores For Iteration 2

When it came to training accuracies, the Decision Tree and Random Forest classifiers performed the best (like in Iteration) however, the MLP classifier performed the best when it came to testing accuracy, with its training accuracy still above 90%. Looking at each classifier’s result, it is evident that some of their accuracies dropped and some of their increased from Iteration 1, but from a holistic perspective, one could say there is not much difference. The reason for no real improvement could be due to a variety of factors, such as not enough training samples (as compared to Iteration 1), potential noise, imbalanced dataset, too many classification classes or even simply the classifiers are not understanding the refined annotation guideline’s rules and definitions.

To increase testing accuracy, some of these mentioned issues would have to be resolved or minimalized. Data augmentation would help minimalize the lack of training samples. As for potential noise, I ran an experiment on whether if *Other* and *Invalid Sentences* are generating noise in our dataset. The prediction is if they do not correlate with argumentation, then training a machine to learn to classify them is redundant. To check if the labels are producing noise, multiple classification tests would need to run to see if the accuracy improves or worsens. Instead of using all the classification algorithms listed, I used Naïve Bayes and SVM to produce the results below. The data used to train the following classifiers is Stage 3 and 4 data and the accuracies listed are *testing accuracies*.

| Exclusion | Algorithms | |
| --- | --- | --- |
| MultinomialNB | SVM |
| No Exclusion | 0.6357 | 0.6860 |
| Excluded *Other* | 0.6289 | 0.6953 |
| Excluded *Invalid Sentence* | **0.6939** | **0.7469** |
| Excluded Both | 0.6543 | 0.6996 |

**TABLE 4**. Exclusion Accuracies of *Other* and *Invalid Sentences*

Logically, if both classes were excluded, this should increase the testing accuracy for a classifier. The reason is not explained here, but an increase in accuracy is found when *Invalid Sentences* are excluded from the training and test set. However, when *Other* and *Invalid Sentences* are excluded, the testing accuracies are lower than when it was just *Invalid Sentence* excluded. A potential reason could be the small noise created by *Other* label, helped classify the other argument labels better, which is an uncommon occurrence. Nonetheless, it is better to exclude both labels than it is to include them both in the training and testing sets. However, even if we remove *Other* and *Invalid Sentences* from the training and testing sets, these types of sentences will still exist in legal corpora. To combat this issue, a 2-layer classification pipeline would suffice to handle this issue and slightly reduce the imbalance problem, in the training and testing sets.

The 2-layer classification pipeline contains one layer of binary classification: *Argument vs. Non-Argument* and the second layer is regular argument classification (5 labels). The results of this implementation are below.

| Algorithm | Accuracies | |
| --- | --- | --- |
| Training | Testing |
| Decision Tree | **1.000** | 0.8625 |
| kNN | 0.9023 | 0.9331 |
| Logistic Regression | 0.8949 | 0.9257 |
| MLP | **1.000** | **0.9368** |
| MultinomialNB | 0.8958 | 0.9257 |
| Random Forest | **1.000** | 0.9331 |
| SVM | 0.9237 | 0.9331 |

**TABLE 5**. Model Accuracies Scores For Iteration 3 (Binary Classification)

From the results above, the MLP classifier was the strongest in the binary classification task in training and testing accuracy, however, many classifiers were able to go beyond 90%, which is a good result. The ability to separate *Other* and *Invalid Sentence* from the argument classification task will prove to be useful in the next phases of this research project, as only relevant information is being classified and being passed on. The results are listed below.

| Algorithm | Accuracies | |
| --- | --- | --- |
| Training | Testing |
| Decision Tree | **0.9990** | 0.6641 |
| kNN | 0.8140 | 0.7819 |
| Logistic Regression | 0.8285 | 0.7407 |
| MLP | **0.9990** | **0.7819** |
| MultinomialNB | 0.6674 | 0.6667 |
| Random Forest | **0.9990** | 0.7613 |
| SVM | 0.9742 | 0.7531 |

**TABLE 6**. Model Accuracies Scores For Iteration 3 (Argument Classification)

From the results above, the MLP classifier is still the strongest in training and testing accuracy, however, many classifiers improved their testing accuracies from Iteration 2. Without *Other* and *Invalid Sentences* in the training set, there is less noise and imbalance in labels, for many of the classifiers to perform better. With a classification pipeline system, it allows multi-filtering of data for whatever relevant task is at hand and has a selective process of using a variety of classifiers together, at different layers.

# discussion

With any research project, there exists a fair amount of questions and concerns in the confidence of fruition for the project, such as security, privacy, etc. However, working with the legal domain, none of those concerns practically apply, as published court reports are online public information for anyone to access, so the issue of security and privacy is mitigated. This research project is geared toward the support of law professionals, without an ulterior motive.

With legality not an issue anymore, there are other potential concerns when furthering the continuation of this research project. One concern is with the data acquisition. As aforementioned, the court reports were pulled from 1960-1965’s. In a modern-day context, law professionals would not see any use of models trained in decades-old data, as the law has changed drastically since then. To properly support law professionals, models would have to be trained on current labeled data, on top of being trained on old data. Building a comprehensive classifier would significantly ease later phases of this research project, especially for the generation of arguments using past data. However, training on data that large is computationally expensive and time-consuming, so other methods would need to be implemented to mitigate that issue. One solution could be to train a variety of models on different periods, to split up the workload of finding patterns and rules in argumentation. Further research will be required to find an efficient solution to a computational issue.

Another research question mentioned was the complexity of the annotation guideline. The method of analyzing syntax and semantics is different for a machine than it is a human. Even with an annotation guideline to follow, a classifier could still have trouble learning the rules, patterns, and subtleties between argument labels. The major update to the annotation guideline was done to conform to the lack of experience in the legal domain (in essence, the annotation guideline was made with simpler rules). However, abstraction for a machine could lead to ambiguity in its predictions. Optimizing the annotation guidelines for annotators and/or machines will require a future study into potential resources to solve this question.

# conclusion and future work

In this paper, my team and I researched argumentation and its applications in the legal domain. Knowing the world of AI has yet to still graze the legal domain, we investigate methods of text classification to support current law professionals by first developing a model for automatic detection of argumentation in legal corpora, regardless of the period. The research questions were defined early on to help guide the project towards better work ethic and results, as well as 2 research designs for creating the legal dataset and generating classifiers. To create high-quality annotations, an argument splitter that minimizes the number of invalid sentences received is used. Using an annotation tool helped streamlined the process of annotating sentences for all annotators. Those labeled sentences were then split up into categories, with each category having a purpose of its own. Out of all the iterations, Iteration 3 was by far the best iteration, with the implementation of the multi-layer classification pipeline system. *sklearn’s* MLP classifier proved to have the best results in training and test accuracies, without any hyperparameter tuning, in Iteration 3. For the binary classification task, it was able to hit 93% testing accuracy, and for the argument classification task, it was able to hit 78% testing accuracy.

While the results of the paper are not standalone, impressive breakthroughs, the basis of this research project is set up appropriately to expand and develop further. There is plenty of future progress to be made, ranging from better quality control of annotations to developing a customized neural network. Below are suggestions to implement for better model accuracies:

* Adopt legal expertise for another renewal of the annotation guideline
* Implement new types of classifiers (geared towards text): RNN, LSTM, and Transformers
* Use pre-trained language models as classifiers, like BERT and GPT-3
* Transfer learning from language models to a classification algorithm
* Psuedo-Labeling: A method to increase model accuracies, if the size of labeled data is small compared to the amount of unlabeled data
* Add another layer to the classification pipeline system (3-layer). This can potentially help with the imbalance problem of Fact being an overwhelming label in the dataset.
  + *Argument v. Non-Argument*
  + *Fact v. Non-Fact*
  + *Argument Classification of Non-Fact*
* Generate a customized complex neural network, geared towards the labeled legal dataset. If MLP was able to generate good results from a common machine learning library, a custom neural network can most likely perform better.

##### Acknowledgments

The amount of research and resources used for this project wouldn’t have been possible without the continuous support of the NSF REU program at the University of North Texas (UNT), along with the faculty mentors, research assistants, and undergraduate students who collaborated on this research project.

I would like to thank the number of research talks attended from the NSF REU program lead by other top researchers who inspired us in the field of NLP and for our research project.

##### References

1. O.-M. Sulea, M. Zampieri, S. Malmasi, M. Vela, L. P. Dinu, and J. van Genabith, “Exploring the Use of Text Classification in the Legal Domain,” arXiv:1710.09306 [cs], Oct. 2017, Accessed: Aug. 10, 2020. [Online]. Available: http://arxiv.org/abs/1710.09306.
2. P. Bhattacharya, S. Paul, K. Ghosh, S. Ghosh, and A. Wyner, “Identification of Rhetorical Roles of Sentences in Indian Legal Judgments,” arXiv:1911.05405 [cs], Nov. 2019, Accessed: Aug. 10, 2020. [Online]. Available: http://arxiv.org/abs/1911.05405.
3. R. Mochales and A. Ieven, “Creating an argumentation corpus: do theories apply to real arguments?: a case study on the legal argumentation of the ECHR,” in Proceedings of the 12th International Conference on Artificial Intelligence and Law - ICAIL ’09, Barcelona, Spain, 2009, p. 21, doi: 10.1145/1568234.1568238.
4. Lavid, Julia. (2010). Towards a science of corpus annotation: a new methodological challenge for Corpus Linguistics. International Journal of Translation. 22. 13-36.
5. K. D. Ashley and V. R. Walker, “Toward constructing evidence-based legal arguments using legal decision documents and machine learning,” in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law - ICAIL ’13, Rome, Italy, 2013, p. 176, doi: 10.1145/2514601.2514622.
6. R. M. Palau and M.-F. Moens, “Argumentation mining: the detection, classification and structure of arguments in text,” in Proceedings of the 12th International Conference on Artificial Intelligence and Law - ICAIL ’09, Barcelona, Spain, 2009, p. 98, doi: 10.1145/1568234.1568246.