UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO
DEPARTAMENTO DE COMPUTAÇÃO E MATEMÁTICA

VITHOR GOMES FERREIRA BERTALAN

Usando métodos de processamento de linguagem natural para prever resultados judiciais (Using natural language processing methods to predict judicial outcomes)

Ribeirão Preto - SP

2020

VITHOR GOMES FERREIRA BERTALAN

Usando métodos de processamento de linguagem natural para prever resultados judiciais (Using natural language processing methods to predict judicial outcomes)

Dissertation submitted to Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) from the Universidade de São Paulo (USP), in partial fulfillment of the requirements to hold the Master of Science degree.

Field of Study: Applied Computing.

Supervisor: Evandro Eduardo Seron Ruiz

Ribeirão Preto - SP

2020

Vithor Gomes Ferreira Bertalan
Usando métodos de processamento de linguagem natural para prever resultados judiciais (Using natural language processing methods to predict judicial outcomes). Ribeirão Preto - SP, 2020.
112p. : il.; 30 cm.

Dissertation submitted to the Faculty of Sciences and Letters
at Ribeirão Preto (FFCLRP) from the University of São
Paulo (USP), in partial fulfillment of the requirements to
hold the Master of Science degree.
Field of study: Applied Computing.
Supervisor: Evandro Eduardo Seron Ruiz

1. Predição Jurídica (Legal Prediction). 2. Processamento de Linguagem Natural (Natural Language Processing). 3. Classificador Jurídico (Legal Classifier).

Este trabalho é dedicado à minha mãe, Márcia,
por sempre estimular seu filho rato de livros.

Acknowledgments

My first thank you goes to my Master's Supervisor, for his guidance as a supervisor was essential for the execution of this study, and my fellow colleagues of the Graduate Program, whose advice helped me to work through the challenges I encountered along the way.

I also thank my partner, who was a pillar of patience and encouragement at every stage of this degree.

Finally, I offer my deepest gratitude to my mother, for her continuous love and understanding.

“All things that are,
Are with more spirit chased than enjoy’d.”
(William Shakespeare, The Merchant of Venice)

Abstract

Natural Language Processing (NLP) and Artificial Intelligence (AI) for the field of Law is a growing area, with the potential of radically changing the daily routine of legal professionals. The amount of text generated by those professionals is outstanding, and to this point, it is a knowledge area to be more explored by Computer Science. One of the most acclaimed fields for the combined area of NLP, AI, and Law is Legal Prediction, in which intelligent systems try to predict specific judicial characteristics, such as the judicial outcome or the judicial class or a given case. This research creates classifiers to predict judicial outcomes in the Brazilian legal system. For this purpose, we developed a text crawler to extract data from the official Brazilian electronic legal systems. Afterward, we developed a dataset of Second Degree Murder and Active Corruption cases, and different classifiers, such as Support Vector Machines and Neural Networks, were used to predict judicial outcomes by analyzing textual features. As a final goal, we used the findings of one of the algorithms, Hierarchical Attention Networks, to find a sample of the most important words used to absolve or convict defendants.

Keywords: Legal Prediction; Natural Language Processing; Legal Classifier

Resumo

Processamento de Linguagem Natural (PLN) e Inteligência Artificial (IA) para a Área Jurídica é uma área em crescimento, com o potencial de mudar radicalmente a rotina diária dos profissionais jurídicos. A quantidade de texto gerada por estes profissionais é imensa, e até o momento inexplorada pela Ciência da Computação. Uma das áreas mais aclamadas é a Predição Jurídica, onde sistemas inteligentes tentam predizer certas características jurídicas, como os pareceres ou a classe jurídica de um dado caso. Esta pesquisa cria classificadores para predizer pareceres jurídicos no sistema legal brasileiro. Para atingir este objetivo, desenvolvemos um rastreador de texto para retirar dados dos sistemas eletrônicos legais do Brasil. Depois, criamos um conjunto de dados composto por casos de Homicídio Simples e Corrupção Ativa, e diferentes classificadores, como máquinas de vetores suporte e redes neurais, foram utilizados com o objetivo de predizer os pareceres através da observação das características textuais. Como um objetivo final, utilizamos os resultados de um dos algoritmos, as Hierarchical Attention Networks, para achar exemplos das palavras que foram mais importantes para absolver ou condenar réus.

Palavras-chave: Predição Jurídica; Processamento de Linguagem Natural; Classificador Jurídico.

List of figures

List of tables

List of abbreviations and acronyms

AI	Artificial Intelligence
eSAJ	Eletronic System of Automation of Justice (in Brazilian portuguese, *Sistema Eletrônico de Automação da Justiça*)
GRU	Gated Recurring Unit
HAN	Hierarchical Attention Network
MLP	Multilayer Perceptron
LSTM	Long Short Term Memory
NLP	Natural Language Processing
RNN	Recurrent Neural Network

List of symbols

α	Greek letter Alpha
β	Greek letter Beta
φ	Greek letter Phi
$\sum$	Greek letter Sigma

Summary

Introduction

Computer Science has been revolutionizing many different fields of expertise. Subfields of Computer Science, like Natural Language Processing (also known as NLP), have steadily improved a myriad of professional and scientific activities. NLP helps researchers understand how to read and understand different types of text, how to extract words, sentences, and their meanings. Even simple NLP mechanisms, such as dictionaries and word counts, with effective processing, can identify interesting underlying facts.

One of the human areas of knowledge that are the most dependent on text is Law. Millions of papers, legislation, court decisions, and appeals are produced daily, and many different specializations, such as lawyers, judges, defendants, and plaintiffs, have different necessities that could be supplied by intelligent systems.

Therefore, it is a reasonable thought to consider that AI could also be used to optimize the daily routines of Law professionals. As this research intends to show, recent studies in the field of AI and Law have been growing for the last years, opening new research areas and market applications. Even though Law methods are long-established, AI for Law is still a developing field. Also, when analyzing the whole state of the art, we could see that the overwhelming amount of research covers the English language only. Analyses in this area for Brazilian Portuguese are still infant, presenting an exciting challenge for new applications.

Primary Objective

As a primary objective, this research intends to develop a framework to predict judicial outcomes in the São Paulo Justice Court[1]. The Sao Paulo Justice Court is the largest judicial court on the planet, considering the number of legal processes [2]. A computational prediction model that offers a satisfactory result for this large judicial court could be of great use and, maybe after fine-tuning, it could also attend any other court. Over the last years, researchers have been dedicated to trying to predict outcomes of judicial

[1] Tribunal de Justiça do Estado de São Paulo, Brasil, TJSP.

[2] https://www.tjsp.jus.br/QuemSomos (in Brazilian Portuguese)

cases, thought the application of NLP and Machine Learning on the texts of those cases. However, to the best of our knowledge, we have no research with this intention in Brazilian Portuguese or for Brazilian courts, as of 2020.

Secondary Objectives

Firstly, we built a text crawler to gather data from the judicial outcomes. To create this model, we combined NLP tools to extract characteristics from the text, selecting what are the main information that can lead to useful predictions. After this step, we inserted this information into machine learning frameworks, such as neural networks and support vector machines, so as to detect what are the best tools to provide better results to the judicial outcomes of the court.

Related Works

This section focuses on a set of articles similar to the work shown in this dissertation. The readers will read other academic references not so strictly related to this study in the next chapter.

In the field of legal prediction, our main research goal, recent advancements have significantly improved the state-of-the-art. Considering a very significant work related to this project, Aletras et al. (2016) have used a dataset of cases from the European Court of Human Rights, containing cases that violate three articles of their Convention. These are:

- Article 3 - *Prohibits torture and inhuman and degrading treatment*;
- Article 6 - *Protects the right to a fair trial*; and
- Article 8 - *Provides a right to respect for one's private and family life, his home, and his correspondence.*

The authors then selected an equal number of cases that violate (+1) and do not violate (-1) each of the Articles. After using regular expressions and pre-processing tools to extract the texts, the authors obtained N-gram features for the Procedure, Circumstances, Facts, Relevant Law, Law, and the Full case. After the extraction of N-grams, they formed groups using vector-space models to find the main topics of each article. They used Support Vector Machines (SVM) to achieve a 0.78 accuracy on predicting Topics for Article 3, 0.84 on predicting Topics and Circumstances for Article 6, and 0.78 on predicting Topics and Circumstances for Article 8.

After the prediction step, the authors also studied the weights of their SVMs so as to find the main words that impacted each of the violations. For Article 3, for example, words such as "*injury, protection, ordered, damage, civil, caused, failed, claim, course, connection, region, effective, quashed, claimed, suffered, suspended, carry, compensation, pecuniary, ukraine*" impacted positively towards a violation. On the other hand, "*sentence, year, life, circumstance, imprisonment, release, set, president, administration, sentenced, term, constitutional, federal, appealed, twenty, convicted, continued, regime, subject, responsible*" impacted negatively towards a violation. They repeated the same process for the other two Articles.

Katz, II and Blackman (2017) have used Random Forests to predict the behavior of the Supreme Court of the United States. The authors rely on a United States Supreme Court dataset with 240 variables, such as chronological variables, case background variables, justice-specific variables, and outcome variables. The authors then add a disposition coding for each one of the cases, labeling them as 'Reversed', 'Affirmed', or 'Other'. They converted all other categorical variables to binary or indicator variables. For the prediction effort, the authors used different methods, such as Support Vector Machines and Random Forests, which they have found to be the best tool to work in their dataset. With Random Forests, the authors reached as high as 0.77 of recall value for the 'Affirmed' class. In the binary labeling, they had a 0.78 of recall value for the 'Not Reversed' cases.

In another influential research, Sulea et al. (2017a) have done a similar investigation, predicting the law area (like Criminal, Social, or Commercial Law) and decisions of the French Supreme Court by using lexical features and SVMs. The authors used a diachronic collection of rulings from the French Supreme Court (*Court de Cassation*, in European French). The complete collection contained 131,830 documents, each consisting of a unique ruling and metadata. Standard metadata available in most documents included law area, timestamp, case ruling (e.g., cassation, rejet, non-lieu, etc.), case description, and cited laws.

After pre-processing, their dataset contained 126,865 different court rulings, each containing a case description and four different types of labels: a) a law area; b) the date of decision; c) the case ruling itself, and; d) a list of articles and laws cited within the description. Features were then selected using hierarchical clustering, and SVMs were used to classify the dataset. The authors reached as high as 90.2% of accuracy to classify the law area; 96.9% of accuracy, using a 6-class SVM to classify the court ruling; and 74.3% of accuracy, using a 7-class SVM to estimate the date of the case and ruling.

All those works indicate that the field of AI and NLP applied to Law is a growing field, offering new and promising research opportunities to the future. Law is a field that covers a wide range of applications, and usually produces a significant amount of data, especially text. Consequently, researchers can produce substantial outcomes, with noteworthy applications.

1

Theoretical Framework

Our theoretical framework analyzed the applications of Natural Language Processing and Artificial Intelligence with the field of Law and characteristics of its work, also approaching novelties of the area and potential issues.

1.1 Natural Language Processing, Artificial Intelligence, and Law

Natural Language Processing, according to Manning and Schütze (1999), is the utilization of quantitative and probabilistic approaches to the automatic processing of texts and speech. Natural Language Processing, unlike image and audio processing, traditionally treat words as discrete atomic symbols, with arbitrary encodings. (PHUOC et al., 2017)

As to Machine Learning, Russell and Norvig (2003) state that it is a subfield of Computer Science (CS) and Artificial Intelligence concerned with computer programs that can learn from experience and thus improve their performance over time. Machine Learning is a promising trend in the field of CS, and it opens new possibilities of research in many areas of knowledge. This technique relies on the hypothesis that large groups of data have, in general, hidden patterns that can be statistically inferred. Therefore, in order to develop a robust machine learning strategy, the most crucial requisite is finding a mathematical model that represents the entry data as trustworthy as possible.

Those concepts, over the last decades, have intertwined with fields that do not necessarily belong to the natural sciences. Branting et al. (2017) cite that automation of legal reasoning and problem-solving has been a goal of Computer Science research from its earliest days. However, according to the author, broad adoption of legal computer systems never occurred, and Computer Science and law remained a niche research area with little practical impact.

Firstly, in order to understand how NLP and AI and interact with Law, we must first understand what this field is. Law, as defined by Le et al. (2015), is the system of rules that guarantees peace, personal freedom, and social justice by regulating human behaviors in all aspects of life. Legal documents, according to the authors, are documents that state some contractual relationship or grant some rights.

Since NLP works well with documents, the application of NLP in Law may hence seem like a perfect match. Hyman et al. (2015) point out the Zubulake versus UBS Warburg case, a series of trials and decisions dealing with what data a litigant has to preserve and under what circumstances the parties must pay for search and production costs, as a seminal case for AI in Law. According to the authors, this case became a landmark for the practical applications of the research field because of the inability of the defendant to recover hundreds to thousands of emails that were claimed by the plaintiff to be relevant to the main issue in the lawsuit.

However, regardless of all the recent advancements in technology, Surden (2014) argues that modern AI algorithms have been unable to replicate most human intellectual abilities, falling far short in advanced cognitive processes, such as analogical reasoning, that are basic to legal practices. This phenomenon is due, according to Branting et al. (2017), to the difficulties of scaling the logic-based approach to the dimensions of complex, dynamic, real-world legal systems. Two main challenges have been written by the authors: the problem of efficiently and verifiably representing legal texts in the form of local expressions, and the difficulty of evaluating legal predicates from facts expressed in the language of ordinary discourse.

As reported by Ashley and Brüninghaus (2009), two long-time goals in AI and Law research are classifying case texts automatically and predicting case outcomes in a manner that is clear so that attorneys can understand them. Sulea et al. (2017a) argue that law professionals would greatly benefit from the type of automation provided by machine learning. According to the authors, artificial intelligence ”systems could act as a decision support system or at least a sanity check for law professionals.” Sulea et al. (2017a, pp. 1). This finding is of utmost importance to overcome one of the main arguments that opponents to the adoption of AI in the Law field have: that the intelligent systems would eventually replace law professionals. We can infer, by looking at the state-of-the-art in the area, that present researchers seek to help law professionals with the most demanding and repetitive tasks, leaving them free to perform better in other activities where computers can not help.

This task, however, has a high level of difficulty. As posited by Surden (2014), many of the tasks performed by attorneys do appear to require the type of higher-order intellectual skills that are beyond the capability of current AI techniques. However, as written by Branting et al. (2017), recent advances in both human language technology and techniques for large-scale data analysis have vastly increased capabilities for automated interpretation of a legal text. Possible hope for AI, therefore, becomes the development of Big Data technologies.

1.2 Areas of research of Artificial Intelligence and Law

According to Branting et al. (2017), there are three main areas of data-centric research in AI and Law:

- **Case-oriented research**, in which the researchers focus on the significant characteristics of cases considered as a whole. Examples of practical applications of this area are litigation assistance and tax recommendation systems.
- **Document-oriented research**, in which researchers focus on the analysis of individual documents. Examples of applications are information extraction for Law, automated summarization, and form completion.
- **Corpus-oriented research**, in which researchers focus on the proprieties of entire collections of legal texts, including network structures, temporal and sequential characteristics. Possible applications are argumentation mining and judicial dataset analysis.

Branting et al. (2017) goes on specifying possible legal tasks amenable to each research area. Those tasks are essential not only on a research level but also to inform potential future research pathways for the field of AI and Law.
These possible tasks are:

1. Legal analysis;
2. Information retrieval;
3. Legal prediction;
4. Argument generation;
5. Dialectical argumentation;
6. Document drafting;
7. Legal planning;

8. Legislative drafting;

9. Trend analysis;

10. Adjudication; and

11. Legal document auditing and quality control.

In the effort of this research, we began by adopting document-oriented research. We have found this option as the most viable one. After all, case-oriented researches are difficult to conduct in Brazil, because some of the documents are heard in private, with legal confidentiality[1]. Therefore, one would not have complete access to the case files. On the other side, we do not have access to a full corpus of labeled documents, so a corpus-oriented research is also not the primary goal. As each of the judicial outcomes goes as a single document in the Brazilian law system, document-oriented research was the leading choice to deal with the present problem, to perform legal prediction.

1.3 Nature of legal activity and practice

Two different schools of thinking dictate the outcome of judicial cases: Legal Formalism and Legal Realism. As mentioned by Liu and Chen (2017), Legal Formalism postulate that judicial decision-making is rationally determinate, in which the procedure of judges' decision making can be modeled either deductively, using formal rules, or with more complex reasoning paradigms. On the opposite side, Legal Realism dictates that formal legal rules are rationally indeterminate on many occasions, insisting that judges decide appellate cases primarily by responding to the stimulus of the facts of the case.

As stated by Surden (2014), a lawyer might employ a combination of judgment, experience, and knowledge of the law to make reasoned predictions about the likelihood of outcomes on particular legal issues or overall issue of liability, often in contexts of considerable legal and factual uncertainty. Zeng et al. (2007) write that lawyers frequently make arguments by analyzing, assessing, abstracting, and interpreting the significance of similarities and differences between cases. Successful arguments critically depend on whether the cited precedents can convince the judge or court.

This finding is confirmed by the works of Barraud (2017). They argue that judges are people that depend strongly on their knowledge of rules, syllogisms, and logic, while judging with their intuitions and sensibility. Judges also have very particular ways of writing, as mentioned by Alarie, Niblett and Yoon (2017), frequently developing very peculiar writing skills in order to individualize the way they present information. Le et

[1] In Brazilian Portuguese, *segredo de justiça*.

al. (2015) also remembers that legal documents, in general, are very formal, and their structure is also very important, maybe as much it's readability. Branting et al. (2017) write that even if legal rules appear with perfect fidelity, the terms in the rules are typically impossible for a layperson to interpret.

Those problems of interpretation can be summarized as written by Boella et al. (2016).

- **Terms with different meanings than ordinary.** Some words have acquired meaning from statutory definitions and scholarly of judicial interpretations that differ from their purpose in standard language;
- **Terms can vary in different contexts and jurisdictions.** Also called polysemy, names can have multiple meanings, according to the legal field of expertise;
- **Intentional vagueness.** Legislation can also be intentionally vague sometimes in order to allow for social and technological changes; and
- **General problems of language.** Imprecise use of language, opening possibilities of interpretation.

Those precedents, as written by Conrad and Al-Kofahi (2017), have patterns that repeat themselves, benefiting practitioners by seeing such patterns comprised of facts, claims, counter-claims, legal principles applied, analysis and decisions. However, as stated by Alarie, Niblett and Yoon (2017), those patterns are challenging to find, because of the highly contextualized nature of legal writing.

This task increases in difficulty due to the fact, as reported by Katz, II and Blackman (2017), that courts have to deal with many different types of juridical analysis, such as tax law, freedom of speech, patent law, administrative law, equal protection, and environmental law. Because of this broad range of judicial specializations, the amount of data gathered is also growing in exponential level. Therefore, as regarded by Zeng et al. (2007), the need to manage legal knowledge effectively for lawyers and judges to locate knowledge and information becomes urgent, due to the rapidly growing volume of the landmark cases.

1.4 Characteristics and structure of legal texts

Aletras et al. (2016) have discovered that formal facts of a case are the most important predictive factor. This factor is of utmost importance since legal texts also have specific characteristics that make them different from other kinds of narratives. For instance, mentions in legal texts have specific structures, which are different from mentions in the public domain (TRAN et al., 2014). Aletras et al. (2016) have written that the textual content and the different parts of a case are essential factors that influence the outcomes reached by judicial courts. Even way before the technological era, as Alarie, Niblett and Yoon (2017) state, judicial decisions were placed in published volumes, as were legislative acts, regulations, and academic and practice-based commentaries in order to re-utilize legal texts in different cases.

Surden (2014) writes that the combination of human intelligence and computer-based analytics will likely prove superior to that of social analysis alone. Kingston (2017) also countersign this idea by saying that AI technologies ought to be able to assist by providing best advice, asking all and only the relevant questions, monitoring activities, and carrying out assessments.

When studying legal precedents via AI, the task of matching specific case facts may prove to be complicated. As mentioned by Zeng et al. (2007), many issues may arise from many different perspectives, such as what the relevant law is, how one could interpret the applicable law in the context, how one could apply the law and what the facts of the case are. This problem is also observed by Sannier et al. (2017), mentioning that when identifying and elaborating legal requirements, analysts need to follow the cross-references in legal texts and consider the additional information in the cited provisions. Tran et al. (2014) also realize this issue, mentioning that, at the discourse level, legal texts contain a lot of reference phenomena.

1.5 Text-based analysis of judicial texts

Aletras et al. (2016) raise the hypothesis that published judgments could be used to test the possibility of a text-based analysis for ex-ante forecasting of outcomes. This idea is corroborated by Surden (2014), arguing that entities concerned with legal outcomes could, in principle, leverage data from past client scenarios and other relevant public and private data to build machine learning predictive models about likely future results on particular legal issues that could complement legal counseling.

Aletras et al. (2016) mention that the judgments of judicial courts have distinctive structures, which makes them particularly suitable for a text-based analysis. Ashley and Brüninghaus (2009) have also followed this way of thinking. The authors have developed a model to extract information from the textual descriptions of the facts of decided cases and apply that information to predict the outcomes of the issues. However, analysts may not forget that, as cited by Sannier et al. (2017), a critical complexity that arises in the analysis of legal texts is that legal provisions are typically interrelated and spread over different texts that cannot be considered in isolation of one another.

Liu and Chen (2017) write that an effective way to explain past decisions of judges and to predict future ones is to study the empirical variables that reflect the non-legal facts of cases, rather than the pure legal deductive arguments. In the same line of thinking, Zeng et al. (2007) write that past legal cases are frequently used to support ideas and judicial opinions, even in classical methods, not using artificial intelligence or any mathematical or statistical models. According to the authors, past cases are called *precedents* in the Common Law system and can be followed, analogized, distinguished or overruled. Sulea et al. (2017b) also wrote that general NLP methods have played an essential role in the intersection between artificial intelligence and law.

It is important to highlight that the main core of Common Law, the use of precedents, is not a common tenet of the Civil Law, the law model used in the Brazilian judicial system. Even though the use of precedents has recently been more frequent after Brazil's Constitution of 1988[2], Brazil still focuses on laws and codes instead of analyzing past cases.

1.6 Practical applications

As cited by Aletras et al. (2016), recent advances in Natural Language Processing and Machine Learning have provided us with many different tools to build predictive models that can be used to infer the patterns driving judicial decisions. This possibility is also suggested by Surden (2014), writing that attorneys could potentially use machine learning to highlight useful unknown information that exists within their current data but disappears due to complexity.

Liu and Chen (2017) also write that NLP and ML have augmented possibilities based on the exploration of the semantic of law and case texts. And according to Barraud (2017), NLP and AI have expanded the limits of justice, transforming it into a predictive, quantitative, statistic, and simulative justice. As stated by Alarie, Niblett and Yoon (2017), intelligent judicial systems can not only help lawyers with timely and objective assessments of their claims but also governments, by using legal classifiers to help evaluate claims and manage litigation risks.

Another practical use of NLP in the legal field is discourse analysis. Discourse analysis is a widespread tool utilized in social research. Moreover, researchers use it in addition to ML methods in Natural Language Processing fields, such as opinion mining and sentiment analysis.

[2] http://www.brazil.gov.br/about-brazil/news/2018/11/civil-law-tradition-guides-rights-in-brazil-but-common-law-is-also-present

A very effective way to compare different thoughts on the judiciary process is by studying the language used by other people. As mentioned by Fairclough (2003), social agents are not entirely free, being socially constrained by the language they use. Comparing different writing styles can be a way to infer what are the social constraints of each region of the world.

Law texts are usually dealing with different positions between the agents. Those differences are directly expressed by their writing styles. Fairclough (2003) argue that differences in the style of writing can be summarized into five different scenarios:

- an openness to, acceptance of, recognition of difference; an exploration of difference, as in ‘dialogue’ in the richest sense of the term
- an accentuation of difference, conflict, polemic, a struggle over meaning, norms, power
- an attempt to resolve or overcome difference
- a bracketing of difference, a focus on commonality, solidarity
- consensus, a normalization and acceptance of differences of power which brackets or suppresses differences of meaning and norms

The author also mentions that those writing styles are not disconnected, and only a single discourse can share more than one level of writing style. In this work, we presume that those levels of stylistic difference can be more effectively measured by NLP. As Fairclough (2003) mentions, one way of capturing those differences “is through looking at collocations, patterns of co-occurrence of words in texts, simply looking at which other words most frequently precede and follow any word which is in focus, either immediately or two, three and so on words away.”.

NLP can adequately address this issue by using, for example, the concept of n-grams. In a similar way of thinking, Brown et al. (1992) also propose the utilization of n-grams to identify similarities between expressions. For them, we can presume two histories are equivalent if they end in the same $n-1$ words.

1.7 Recent advancements in Artificial Intelligence and Law

Artificial Intelligence methods have been used effectively in diverse areas of the law. Mcshane et al. (2012), for example, have used AI and NLP to build a Hierarchical Bayesian model to predict fraud settlements in American federal securities class action lawsuits, using data from risk metrics, identifying predictors of settlement incidence and settlement amount. Talley and O'Kane (2011) used regular expressions and Latent Semantic Analysis to analyze force measures clauses in mergers and acquisitions agreements, by replicating, correcting, and extending the reach of the hand-coded data.

Zeng et al. (2007) have used case-based reasoning to solve new judicial problems by remembering and adopting previous similar situations, developing a new set of subelements for legal case representation. In case-based logic, as mentioned in the previous sections, each judicial case usually consists of three parts: description of the problem or situation, the solution to that problem or situation, and the outcome of that solution. By introducing new contextual features, the authors have managed to help retrieval in the domain of accident compensation.

Gokhale and Fasli (2017) have developed a co-training algorithm to classify human rights abuses, using SVM and Logistic Regression on a domain ontology created for the domain of human rights as background knowledge, so as to extract the initial terms for generating the labeled data to train the classifier. Branting et al. (2017) have used Hierarchical Attention Networks, SVMs and Maximum Entropy classifications for decision support in administrative adjudication, such and routine licensing, permitting, immigration, and benefits decisions of the Board of Veterans Appeals (BVA) and World Intellectual Property Organization (WIPO) domain name dispute decisions. SVMs are also used by Fornaciari and Poesio (2013) to automatically detect deception, such as slander and false testimony, by analyzing the results obtained by using stylometric techniques to identify deceptive statements in a corpus of hearings collected in Italian courts.

Remmits (2017) have used Latent Dirichlet Allocation to discover the main topics of discussion in judicial outcomes of the United States Supreme Court. In this work, the author also compares whether or not legal experts and people with a non-legal background agree in their judgments, discovering that domain experts and non-domain experts might evaluate topics differently. Mochales and Moens (2011) have used argumentation mining to structure better legal arguments, capturing main issues and evidence of a given corpus, also stating that the method needs further research to automatically acquire the necessary background knowledge and more specifically common sense and world knowledge.

Le et al. (2015) have used index extraction using structural information of sentences for Japanese legal documents, assigning each token with a weight, which is a statistical score to indicate its importance. El Jelali, Fersini and Messina (2015) also uses information retrieval to support adjudication in Italian court decisions, by adopting machine learning and natural language processing techniques to better match disputant case descriptions (informal and concise) with court decisions (formal and verbose).

Also, using information retrieval techniques, Hyman et al. (2015) have developed a process model for knowledge discovery, using judicial cases as examples of their model's applicability, finding the constructs of uncertainty, context and relevance while retrieving information. Boella et al. (2016) have developed a system to create ontologies destined to give the relevant law on any given topic, using NLP tools to semi-automate the lower-skill tasks. With a similar goal, Francesconi and Peruginelli (2009) have created a system to search and retrieve Italian legal literature, by creating a centralized index of legal resources, using OAI and machine learning approaches. Aires et al. (2017) have used deontic logic to identify potential norm conflicts in contracts.

Some recent research have also been explicitly conducted in Brazilian Portuguese. As an example, Araujo, Rigo and Barbosa (2017) have used ontology-based algorithms adopted by using a domain ontology of legal events and a set of linguistic rules integrated through inference mechanism to classify legal documents in Brazilian Supreme Court judicial outcomes.

1.8 Potential issues in Artificial Intelligence and Law

Regarding the technical issues, as stated by Surden (2014), there are some well-known limitations to the application of AI in Law. The first one is that a model will only be useful to the extent that the class of future cases has pertinent features in common with the prior analyzed topics in the training set. Therefore, the model will not contemplate subtle changes in judicial thinking over time, only if those changes arise to represent a considerable size of the training data.

The authors also present an example: not every law firm will have a stream of cases that are sufficiently similar to one another such that the past case has elements that are useful to predict future outcomes. Hence, one may infer that only the largest law firms will have the necessary financial and technological assets to develop such models.

As to the social issues, another possible problem, as stated by Surden (2014), is an overgeneralization, also known in machine learning as *overfitting*. The model in intrinsically based in the cases provided for the training set. Thus, if the training set has cases that are so finely tuned to the idiosyncrasies of a few judicial matters, it will not be able to have the necessary adaptability to cases of different judicial natures. For that reason, the past case data upon which a machine learning algorithm is trained may be systematically biased in a way that leads to inaccurate results in future legal cases.

Katz, II and Blackman (2017) suggest that qualitatively-oriented legal experts tend to suggest model improvements based on anecdote or their untested mental model, instead of reliable and factual data. The authors write that, to support a case from a model's future applicability, it should consistently outperform a baseline comparison. This requisite is not necessary only for scientific purposes but also to gain attorneys' trust in the model.

Another problem, not entirely dependent on Artificial Intelligence, is the problem of accessibility of Law. As written by Boella et al. (2016), difficulties of accessibility arise because of the following reasons.

- Law is increasing in scope, volume, and complexity. This problem is also noted by Francesconi and Peruginelli (2009), which points out the size of legal literature as one of the factors of paramount importance for consideration in future researches;
- The myriad of areas of expertise of law, frequently not classified intuitively on official legislative portals. Again, Francesconi and Peruginelli (2009) indicate this problem to be of fundamental importance, stressing out the importance of delimitating the legal domain;
- Legal norms are coming from different sources, such as regional, national, or supranational authorities. Hyman et al. (2015) use the Philip Morris litigation case as an example of this item, saying that, at one point on the issue, more than 30 million pages of documents were available, coming from many federal government agencies; and
- New legislation modifying or overriding existing norms but not explicitly saying so.

Therefore, expert consultation still proves itself to be fundamental to the making of a practical AI system for Law. Because of this limitation, Kingston (2017) argue that AI commercial systems should rely on the Pareto Principle: an 80-20 rule, where the AI system should cover 80% of the questions, and leave the remaining 20% to the decision of the legal experts using the application. That remaining 20% can also be used to refine the AI model, turning the algorithms gradually more useful to predict new outcomes.

2
Methodology

In this chapter, we describe the domain characterization, the necessary steps to conclude the research. We also describe the evaluation measures used to assess the effectiveness of the proposed model.

2.1 Domain characterization

As mentioned in the previous sections, we collected a corpus of judicial outcomes from the eSAJ[1], the electronic system of the TJSP, Tribunal de Justiça de São Paulo. We selected a few previously defined judicial subjects to restrict the documents captured. We chose only judicial subjects with very well defined outcomes. Namely, second-degree murder (*Homicídio simples*), from now on called homicide, and active corruption (*Corrupção Ativa*), from now on called corruption. We then selected those judicial outcomes with the condemnation or absolution of the defendant. Many different judicial subjects do not have explicit terms for condemnation or absolution. Therefore, it is of utmost importance to find those judicial subjects with clear and well-established results.

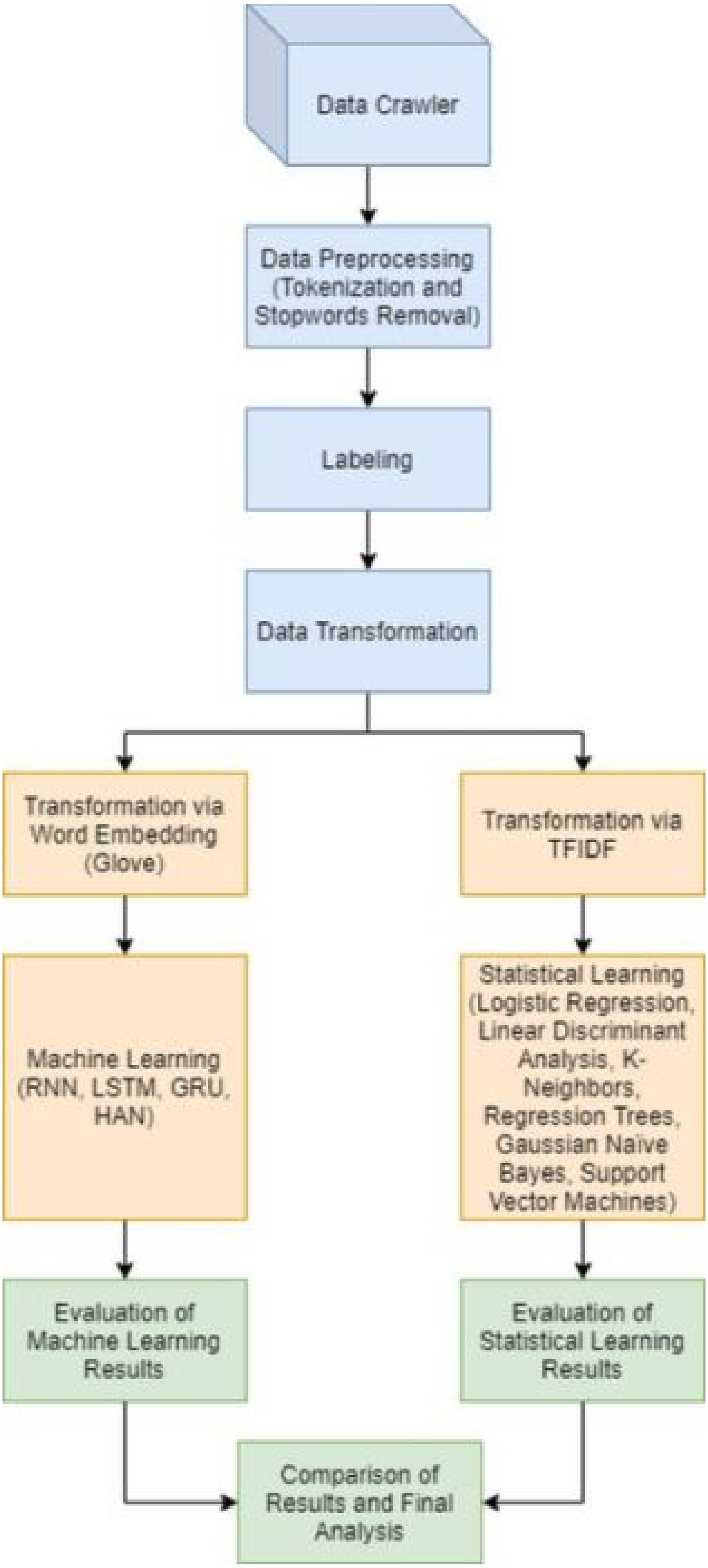
2.2 Data collection

We have implemented a web text crawler to capture the data from eSAJ, São Paulo Justice Court judicial electronic system. As the user can select from many different fields to exhibit judicial opinions, such as classes, subjects, judges, and process numbers, the crawler was able to choose the appropriate texts. The crawler saved the documents retrieved from the queries to files. We describe the pseudocode used in Algorithm 1. The complete code for the text crawler is available in the author's GitHub page [2].

[1] http://esaj.tjsp.jus.br/cjpg/

[2] https://github.com/vbertalan

Figure 1 – Methodological phases of this project.


Data Crawler
Data Preprocessing (Tokenization and Stopwords Removal)
Labeling
Data Transformation
Transformation via Word Embedding (Glove)
Transformation via TFIDF
Machine Learning (RNN, LSTM, GRU, HAN)
Statistical Learning (Logistic Regression, Linear Discriminant Analysis, K-Neighbors, Regression Trees, Gaussian Naive Bayes, Support Vector Machines)
Evaluation of Machine Learning Results
Evaluation of Statistical Learning Results
Comparison of Results and Final Analysis

As the eSAJ data fields are fundamental to the comprehension of the texts captured, it is necessary to explain each of the areas. They are[3]:

1. *Judicial Class* (in BrPT, *Classe*)
2. *Judicial Subject* (*assunto*)
3. *Magistrate* (*Magistrado*)
4. *County* (*Comarca*)
5. *Judicial Forum* (*Fórum*)
6. *Judicial Court* (*Vara*)
7. *Date of availability* (*Data de disponibilização*)
8. *Text* (*Texto*).

'Classes' are types of judicial documents. For example, repeals and termination of contracts would be judicial classes under Brazilian law. Subjects are the type of judicial case being conducted, such as drug trafficking or feminicide. The magistrate is the state judge responsible for judging the case. The county, in the Brazilian judicial system, works differently from the Common Law system. In the Executive and Legislative branches, the geographic divisions of the country are called cities. As for counties, in Brazil, these are the geographic divisions made by the Judiciary. The judicial forum is the sector responsible for evaluating the case, in which a magistrate stands. The clerks that support the magistrates in conducting the subjects work at a judicial court. The date in which the judicial order is presented to the public and the interested parties is the date of availability. 'Text' stands for the full content of the judicial dispatches.

Therefore, the data collection poses a compelling challenge, as the amount of data displayed on TJSP's website is indeed very significant. As of June 1st, 2018, a simple search for the Brazilian Portuguese word corresponding to 'rape' (*estupro*) returned 5,658 hits. Another search for the Brazilian Portuguese term for drug (*droga*) returned 138,956 hits. Each result is a judicial opinion of its own, containing many sentences and text topics. As each judicial class under the Brazilian law system has different text topics, e.g., the issues in a text from the class *divorce papers* (in Brazilian Portuguese, *documentos de divórcio*) differ substantially from contents from the class release permits (in Brazilian Portuguese, *alvará de soltura*).

As judicial texts can extend to the length of many pages, the data may have a significant size. Hence the crawler must be able to gather the data in a predefined date

[3] Also, the original term in Brazilian Portuguese, BrPT.

Algorithm 1 Text crawler to capture data from the eSAJ system

```
procedure TextCrawler($class, subject, magistrate, initialDate, finalDate$)    ▷ Captures each section into a CSV column
    $text \leftarrow NULL$
    $esaj \leftarrow searchResults$    ▷ eSAJ Search Results, as raw text
    $section \leftarrow esaj.firstSection$
    while $section \neq NULL$ do    ▷ Searches the document until it finishes
        $newSection \leftarrow section.decode('utf8')$
        $text \leftarrow text + newSection.encode('iso - 8859 - 1')$    ▷ Populates new column
        $section \leftarrow text.nextSection$
    end while
    return $text$    ▷ The CSV is given as a result
end procedure
```

interval, to keep the number of selected cases doable. So, the initial date of availability and a final date were requested from the user to capture a restricted corpus.

For this research, using the text crawler developed, we have collected **2467** cases in total, only selecting homicide and corruption subjects, resulting in 1681 homicide cases and 786 corruption cases. The crawler was used to gather documents from different periods. The total distribution, including absolutions and condemnations, is shown in Table 1. Related crimes were not considered for this research.

Table 1 – Documents collected for the research.

Judicial subject	**Homicide**	%	**Corruption**	%
Number of Absolutions	844	50.2	197	25.0
Number of Condemnations	837	49.7	589	75.0
Total Cases	1,681	100	786	100

2.3 Data pre-processing

We pre-processed the data retrieved to remove unnecessary information. We began this step by tokenizing the text and eliminating stopwords using the Natural Language Processing Toolkit in Python, called NTLK, as developed by Bird, Klein and Loper (2009). As NLTK can deal with different languages other than English, such as Brazilian Portuguese, the framework was the selected tool to handle this task.

2.3.1 Tokenization

The exact definition of a token is not very precise and very liable to change, according to Manning and Schütze (1999). Different linguists have different explanations to the term, according to the domain of study and the application of the corpora. One of the most accepted definitions of tokenization is defined by Maverick (1969) as a chain of alphanumeric sequential characters with spaces on both sides, possibly including hyphens and apostrophes, but not punctuation signs.

As stated by Manning and Schütze (1999), many different challenges are faced when one has to tokenize a corpus. The first one is in its composition. In most corpora, in the majority of times, words are directly followed by punctuation signs, without a blank space, even if it represents a different token.

A second difficult challenge is a phenomenon called *haplology*, where the same punctuation sign has different meanings, according to its use in other sentences. As an example, the character '.', the dot. It may represent a period if put at the end of a sentence, but it may also mean suspension points if put in groups of three. It may also symbolize the separation between exact numbers and its fractions, as in USD 100.00, for instance. Another good example is the character '-,' the hyphen, that may represent a compound word, such as "long-term." This utilization of '-' to aggregate different terms into a single one is widespread in Brazilian Portuguese. The very same character may also symbolize syllable separation, again very popular in Brazilian Portuguese, but not very used in English. Moreover, in Brazilian Portuguese, hyphens are also used to join object pronouns to the nouns they refer. For example, the expression 'give me' can be written as the Portuguese *dá-me*.

In order to counter those issues, we used the NLTK Sentence Tokenizer package, which has internal algorithms to find patterns such as parenthesized expressions and division of substrings. It uses by default the Punk algorithm, described in Kiss and Strunk (2006), using context-independent criteria to tokenize words.

2.3.2 Stopwords Removal

Stopwords are defined, as mentioned by Khosrow-Pour (2008), as words that have no significant semantic relation to the context in which they exist, frequently occurring in most documents in a given collection. Those words are ubiquitous terms that would appear to be of little value in helping select records that match our needs. Hence, we used this step to remove from the corpus words that are not interesting to the analysis of the text, that is, words such as articles, linking words, and prepositions.

As suggested by Moens (2001), we have used stopword removal in this work, intending to remove terms that are not relevant for the classification of legal texts. Once more, we used the NLTK framework in this step. NLTK has a standard set of Brazilian Portuguese stopwords, making it easier to identify those words in a given corpus.

Another great advantage of using stopwords is to remove words that do not interest the final lexicon. This strategy might increase in performance of the algorithms. According to Manning and Schütze (1999), using a simplified list of words may reduce the size of the inverted index by half, a phenomenon explained by a mathematical and statistical empirical precedent called Zipf's Law, also very relevant to the field of Linguistics, mentioned in Zipf (2013).

We show an example of an application of stopword removal in sentences in the legal field in Tables 2 and 3.

Table 2 – Sentences before and after stopword removal - in English.

Original Sentences	**Altered Sentences**
The defendant was found guilty of drug trafficking.	The defendant guilty drug trafficking
The judge has requested new documents to the attorney.	The judge requested documents attorney
After a thorough trial, the lawyer has a new plea to ask.	After thorough trial, lawyer plea ask

Table 3 – Sentences before and after stopword removal - in Brazilian Portuguese.

Original Sentences	**Altered Sentences**
O réu foi condenado pelo crime de tráfego de drogas.	Réu condenado crime tráfego drogas
O juiz solicitou novos documentos ao advogado.	Juiz solicitou novos documentos advogado
Após um julgamento exaustivo, o advogado tem um novo apelo a apresentar.	Após julgamento exaustivo advogado novo apelo apresentar

2.4 Labeling

As we were to manipulate the data gathered in machine learning and NLP algorithms, it was necessary to label the dataset in order to have a guide to the algorithms of supervised learning.

We have read each one of the judicial outcomes, and we have classified them between condemnation (+1) and absolution (-1). We have used the professional guidance of Brazilian lawyers, with the purpose of better understanding the texts. The language adopted worldwide in the field of Law is notoriously obscure. Therefore, we decided that professional consulting was necessary in order to understand the outcome of each of the judicial cases fully.

As mentioned before, only a few selected judicial subjects were selected. The main criterion of selection were subjects that had a clear definition of condemnation or absolution. We show the full amount of condemnations and absolutions in Table 1.

2.5 Data transformation

After the data pre-processing, we have transformed the resulting data, which in our case is composed of words, sentences, and documents, into a mathematical sequence that can be passed through machine learning or statistical algorithms, such as an SVM or a neural network. Many different methods are available in this step, and we have chosen two of them: TFIDF and Word Embeddings.

2.5.1 TFIDF

TFIDF is an acronym that stands for 'Term Frequency– Inverse Document Frequency,' is a method in which uncommon words, including hapaxes (words that occur only once in a context), are ranked with more importance than common terms, such as 'the.' Therefore, terms are quantified in an inverse function of the number of documents that they occur.

To get the TFIDF value of a term, we calculate the product of two sub-equations: term frequency, in which we obtain the number of times that term t occurs in document d; and inverse document frequency, in which we get the measure of how much information the word provides, finding its rarity across all documents, applying the logarithmically scaled inverse fraction of the documents that contain the term. We show the equation regarding TFIDF in Equation 2.1, with $tf_{t,d}$ standing for the frequency of t in d, N standing for the total number of documents, and df_t standing for the number of documents containing t.

$$\text{tf-idf}_{t,d} = (1 + \log \text{tf}_{t,d}) \cdot \log \frac{N}{\text{df}_t} \tag{2.1}$$

To understand the model, we can use as an example a document containing 100 words wherein the word 'jury' appears three times. The term frequency (i.e., tf) for jury is then $tf = (3/100) = 0.03$. Now, assuming we have 10 million documents, and the word jury appears in one thousand of these. Then, the inverse document frequency (i.e., idf) is calculated as $log(10,000,000/1,000) = 4$. Thus, the tf-idf weight is the product of these quantities: $tfidf = 0.03 * 4 = 0.12$.

2.5.2 Word Embeddings

We can use different methods with the intention of transforming data. One of the most popular is word embedding. Word embeddings, as defined by Turian, Ratinov and Bengio (2010), are vectors composed by real numbers distributed over an inter dimensional space, induced by semi-supervised learning. Word embeddings have become a very effective alternative to transform the pure text into mathematical values, making it easier to manipulate data using machine learning algorithms. The algorithms calculate the similarities between two vectors by using cosine similarity. Each dimension of the vector represents a characteristic, intending to capture semantic, synthetic or morphological proprieties of a word in a distributed way.

In this research, we have used GloVe, a method created by Pennington, Socher and Manning (2014). GloVe, a reduction for Global Vectors, is a word embedding method developed to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Rather than using a window to define local context, GloVe constructs an explicit word-context or word co-occurrence matrix using statistics across the whole text corpus. It has the ability to combine local and global representations of a term by mixing the features of two model families, namely the global matrix factorization and local context window methods. We have used the pre-trained GloVe corpora developed by Hartmann et al. (2017) for Brazilian Portuguese.

2.6 Classification: statistical and machine learning algorithms

The problem of classification of different entities is present in various fields of the humanities, including Law. In order to achieve this goal, many strategies can be tested. There are many different algorithms destined to solve the problem of classification. Depending on the method of representation and the technique utilized, these algorithms can be divided into many fields of concentration, such as symbolic methods, statistical methods, neural networks, connectionist models, and hybrid models.

According to Manning, Raghavan and Schütze (2008), ML is a field of Artificial Intelligence dedicated to building systems to optimize a classifying function based on a set of examples or past experiences.

Data classification has three main strategies for learning: supervised learning, unsupervised learning, and reinforcement learning. As stated by Mohri, Rostamizadeh and Talwalkar (2012), in the first strategy, *supervised learning*, the learner receives a set of labeled examples as training data and makes predictions for all unseen points. In this approach, the first step required is having a training sample, with entry data and labeling information for each entry, from which the algorithm can learn. This process is also known as model training. With the resulting trained model, it is possible to infer the classification of unseen data. In the context of NLP in our research, we used the labeled text as the training data.

Also described by Mohri, Rostamizadeh and Talwalkar (2012), in the second strategy, *unsupervised learning*, the algorithm exclusively receives unlabeled training data and makes predictions for all unseen points. This strategy also demands a training sample, but labeling each entry with its class is not required. The goal is to find hidden patterns within this sample, such as repetition patterns that happen more frequently than others. Clustering is an example of a prevalent unsupervised learning method.

The last strategy is called *reinforcement learning.* As specified by Mohri, Rostamizadeh and Talwalkar (2012), this method works by intermixing the training and testing phases. To collect information, the algorithm actively interacts with the environment and, in some cases, affects this environment, receiving an immediate reward for each action. The algorithm applies a step by step learning based on the observation of the domain. Each action taken by the algorithm has a specific effect on the environment. This effect is then reapplied into the algorithm, learning from each previous activity. In this strategy, the sequence of steps taken are of utmost importance to the learning process.

In this research, we have adopted the supervised learning strategy, using the labeled samples of the judicial outcomes as the classifying information. Hence, the labeled samples served as the basis for the algorithm to predict new cases not included in the dataset.

Over our bibliographic review for this dissertation, we came across different NLP algorithms, used in many purposes. We selected algorithms that had good performances in similar classification works so that we could try their accuracy in our datasets. We show the full list of chosen algorithms and a few examples of their applications in Table 4.

Table 4 – Chosen algorithms and practical NLP applications.

Algorithm	**Practical NLP Applications**
Logistic Regression	Liu and Chen (2017), for prediction of circumstances and topics of law cases Pelle, Alcântara and Moreira (2018), for offensive text detection
Linear Discriminant Analysis	Krestel, Fankhauser and Nejdl (2009), for tag recommendation in search websites Pavlinek and Podgorelec (2017), for text classification in newsgroups datasets
K Nearest Neighbors	Chantar and Corne (2011), for document categorization in the Arabic language Desmet and Hoste (2014), for automatic recognition of suicidal messages in social media
Classification and Regression Trees	Kanakaraj and Guddeti (2015), for measuring sentiment analysis on Twitter Rios-Figueroa (2011), for predicting judicial independence in Latin American courts
Naive Bayes	Harcourt and Harcourt (2015), for feature selection and vectorization in legal documents Rios-Figueroa (2011), for measuring sentiment analysis on Facebook statuses
Support Vector Machines	Do et al. (2017), for legal question answering and ranking Sulea et al. (2017b), for predicting law area and decisions of French Supreme Court cases
Multilayer Perceptron	Rao and Spasojevic (2016), for political text classification Sa, Santos and Moura (2017), for defining the author reputation of product comments
Recurrent Neural Networks	Alschner and Skougarevskiy (2017), for automated production of legal texts Kim et al. (2017), for demographic inference on Twitter
Long Short Term Memory	Li et al. (2017), for political ideology analysis Xie, Liu and Dajun Zeng (2017), for mining product adverse events in social media
Gated Recurring Unit	Luo et al. (2017), for predicting charges for criminal cases Zhang, Robinson and Tepper (2018), for detecting hate speech on Twitter
Hierarchical Attention Networks	Branting et al. (2017), for predicting models for decision support in administrative adjudication Gao et al. (2018), for information extraction from cancer pathology reports

2.6.1 Logistic Regression

Logistic regression is a model used to predict a categorical variable, usually binary, from a series of explanatory continuous or binary variables. Like all regression analyses, logistic regression is a predictive analysis. It is used to predict the occurrence of an event directly. The algorithm takes a weighted combination of the input features (in our research, the result of the TFIDF transformation). It passes it through a sigmoid function, which outputs any real number to a number between 0 and 1.

As cited by Bishop (2006), it works as a statistical method destined to find an equation that predicts an outcome for a binary variable, from one or more response variables. The response variables can be categorical or continuous, as the model does not strictly require continuous data. Logistic regression uses the log odds ratio rather than probabilities to predict group membership, and an iterative maximum likelihood method rather than the least squares to fit the final model.

Compared to other dependence techniques, logistic regression gathers categorical variables more easily. It is also an excellent approach to problems that involve probability estimation since it categorizes each event on a scale from 0 to 1. The main goals of a logistic regression are to determine the effect of a subset of independent variables in the overall probability, considering the isolated impact of variables as well, and having the highest possible predictive accuracy, given a subset of predictors.

2.6.2 Linear Discriminant Analysis

Linear Discriminant Analysis is a technique from multivariate statistics used to discriminate and classify objects. It ranks each sample in one of many populations, using a p number of characteristics, seeking to minimize the probability of a wrong classification. In order to do this, the algorithm uses a combination of linear features that present a higher capability of classification between populations. This combination is called a discriminant function.

Balakrishnama and Ganapathiraju (1998) argue that Linear Discriminant Analysis handles the case where the within-class frequencies are unequal, and their performances has been examined on randomly generated test data. This method maximizes the ratio of between-class variance to the within-class variance in any particular data set, thereby guaranteeing maximal separability. The use of Linear Discriminant Analysis for data classification is quite often applied to NLP classification problems, such as sentiment analysis or speech recognition. The authors mention that the algorithm tries to provide more class separability and draw a decision region between the given classes in a given field.

The discriminant function seeks to verify which variables from the subset are essential to classify that subset among the populations. Since our research aims to rank the subset among two groups, condemnation and absolution, we have used the Linear Discriminant Analysis technique. This approach uses the discriminant function to classify each value between two populations, by selecting the minimum ratio of the difference between a pair of group multivariate means to the multivariate variance within the two groups.

2.6.3 K-Nearest Neighbors - KNN

The algorithm K-Nearest Neighbors (KNN) is a supervised learning algorithm that intends to find the k labeled examples closest to a non-classified example, by getting the label of those most comparable examples. The algorithms in the KNN family do not demand a significant computational effort during the training phase. However, the computational cost to label a new instance is considerably high, since, in the worst case, this example will have to be compared will all the other instances present in the training dataset.

As cited by Bishop (2006), in regions of high data density, models may lead to over-smoothing and washing out of a structure that might otherwise be extracted from the data. However, simplifying the model may lead to noisy estimates elsewhere in data space where the density is smaller. Thus the optimal choice for the model may be dependent on location within the data space. Nearest-neighbor methods for density estimation address this issue.

In our research, KNN uses the output of TFIDF as the input matrix. It gets the label of condemnation and absolution for each row in the dataset. The algorithm classifies each document in the Euclidean space as a point. Afterward, it uses the Euclidean distance to classify each of the subsets.

2.6.4 Regression Trees

Classification and regression trees (CART) are machine-learning methods for constructing prediction models from data. As stated by Loh (2011), the models are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. Classification trees work for dependent variables that take a finite number of unordered values, with prediction error measured in terms of misclassification cost. Regression trees are for dependent variables that take continuous or ordered discrete values, with prediction error typically measured by the squared difference between the observed and predicted values.

In this research, we the CART version available in the NLTK Framework. The method of analysis uses classification rules made by decision trees. The tree begins with a root node with all the text characteristics. The following nodes contain subsets and subdivisions of the data. Each division results in precisely two nodes. This method allows the identification of homogeneous groups of data by systematically comparing their characteristics, intending to establish a relationship between explanatory variables and a single answer variable - in our case, the label of condemnation or absolution. The model is adjusted through successive divisions in the dataset, to make the subsets every time more homogeneous towards the answer variable. The division process is repeated until none of the variables selected show significant influence in the division, or if the subset is too small to be divided again.

Using CART, the criterion to exclude variables from the model is a measure called *improvement*, which is responsible for classifying the variables excluded from the model with the addition of new variables. The higher the improvement value, the greater the importance of a variable in the classification, and the more homogeneous will be new nodes be. In our research, we have used a minimum improvement of 0,01 as a stop criterion.

2.6.5 Naïve Bayes

Naïve Bayes classifiers are a family of probabilistic classifies that draw on the Bayes's Theorem to generate models with high independence assumptions between the features. As stated by Manning, Raghavan and Schütze (2008), Bayesian classifiers are the ones in which an object x is assigned to a class, C_k, based on the probability that x belongs to C_k. We show an example of the formula in Equation 2.2.

$$\mathrm{P}(C_k|x) = \frac{\mathrm{P}(C_k)\mathrm{P}(x|C_k)}{\mathrm{P}(x)} \tag{2.2}$$

Where:

- $P(C_k|x)$ is the probability of hypothesis C_k given the data x. This value is called the posterior probability.
- $P(C_k)$ is the probability of hypothesis C_k being true (regardless of the data). This value is called the prior probability of C_k.
- $P(x|C_k)$ is the probability of data x given that the hypothesis C_k was true.
- $P(x)$ is the probability of data x (regardless of the hypothesis).

Bayesian classifiers are fast, producing results with shorter processing times than other classification methods. They work both for binary and multinomial classification.

2.6.6 Support Vector Machines

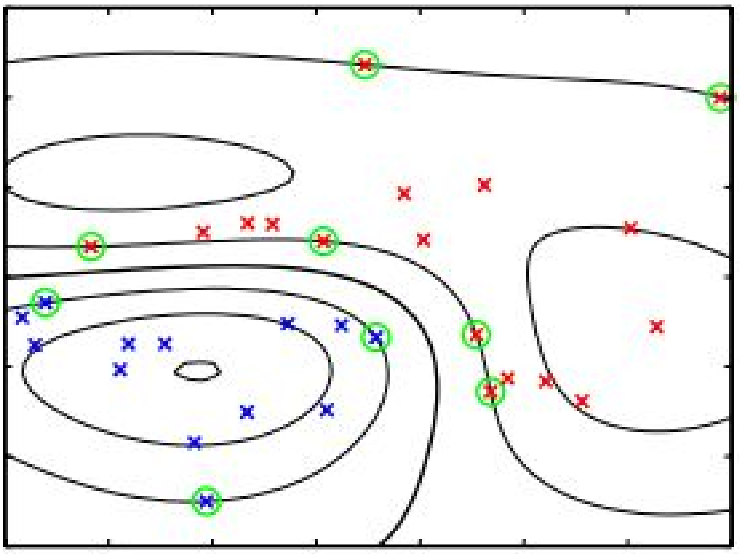
Support Vector Machines are an algorithm destined to binary classification by plotting the elements of a dataset and trying to separate it by defining a separation function. The most effective separation function is the one that shows the best classification by offering the largest margin between the two given classes.

In this model, the support vectors are the dots from both classes that are closest to the separation function. This separation function is also called a hyperplane. The algorithm plots the new element in the same space to predict new features, and verifies in which group the new element has fallen into.

As stated by Bishop (2006), if there are multiple solutions, all of which classify the training data set precisely, then we should try to find the one that will give the smallest generalization error. According to the author, the support vector machine approaches this problem through the concept of the margin. This concept is defined to be the smallest distance between the decision boundary and any of the samples. In Support Vector Machines the decision boundary is chosen to be the one that maximizes the margin.

We show an example in Figure 2. The separation function tries to separate the red from the blue crosses. A new example can have its class predicted by placing it in the hyperplane.

Figure 2 – Example of synthetic data from two classes in two dimensions showing contours of constant $y(x)$ obtained from a Support Vector Machine having a Gaussian kernel function. Also shown are the decision boundary, the margin boundaries, and the support vectors. Source: Bishop (2006).



2.6.7 Multilayer Perceptrons - MLP

Multilayer Perceptrons are algorithms that extract characteristics from a dataset, composed by units called *Perceptron neurons*, which are interconnected. Those neurons are units responsible for controlling the error that arises from each prediction made by the algorithm.

A MLP works by following Equation 2.3.:

$$y = \varphi(\sum_{i=1}^{n} w_i x_i + b) = \varphi(w^T x + b) \tag{2.3}$$

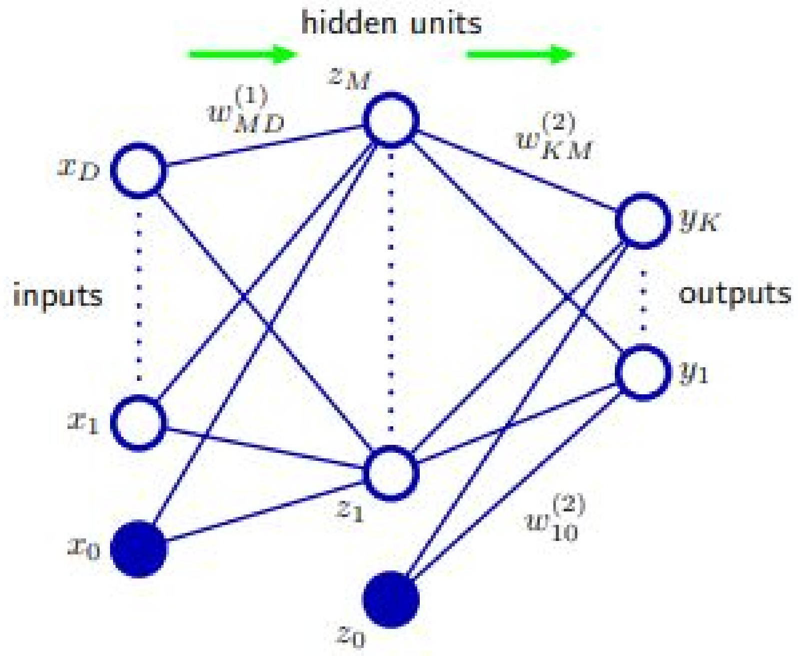
In which n stands for the number of inputs of the Perceptron neuron, w_i represents the weight of the connection referring to the input i, x_i is the value of input i, b is a bias, acting just like a weight (also called *synapses*) in a connection of a unit whose activation is always 1, and φ is the neuron activation function, such as the sigmoid function, destined to process the signal generated by the the linear combination of the inputs and weights of the synapses to generate the output signal of the neuron.

The MLP architecture works in layers. The first layer functions as a sensory receptor, receiving an input signal (data). The last layer is known as the exit layer, in which we can see the answer from the MLP to the input signal. Between the first layer and the exit layer, we can have one or more hidden layers. The hidden layers and the exit layer are composed of Perceptron neurons, receiving the input signal, processing the signal through an activation function, and passing it to the next layer.

The weights (synapses) have to be calibrated after every input so that the network learns the most important characteristics from the dataset. The calibration of those synapses works through an algorithm destined to minimize the error produced by the MLP, known as *backpropagation*.

We show an example of an MLP in Figure 3. The architecture has three layers: one input layer, one hidden layer, and one output layer. The neurons $x_0, ..., x_m$ receive the input signals, and pass them to the weights w_{ij} in an activation function. The hidden layer receives the intermediate signals in neurons $y_0, ..., y_{n-1}$. Those signals are sent again to the weights v_{ij}, passed through an activation function, and outputted as $z_0, ..., z_{p-1}$. If $p < m$, the MLP also performs a dimensionality reduction.

Figure 3 – Network diagram for an MLP with 3 layers. The nodes represent the input, hidden, and output variables. The links between the nodes define the weight parameters, and the links coming from additional input and hidden variables x_0 and z_0 denote the bias parameters. Arrows indicate the direction of information flow through the network during forwarding propagation. Source: Bishop (2006).



2.6.8 Recurrent Neural Networks - RNN

MLPs work for many applications. However, if we have an input that behaves like a time series when a value intrinsically depends on the previous output, this signal is going to affect the next input directly. Values tend to get lost inside MLP architectures, since all the inputs receive a singular output, without dependencies between them.

Recurrent Neural Networks are an architecture that handles variable-length sequential input by way of a recurrent, shared hidden state. An RNN is a machine learning model in which M inputs are entirely connected to N units. We show an example with $N = 3$ in Figure 4. Since the meaning of a word in a text is entirely dependent on the previous and posterior terms, we can also consider text a time series, being strong candidates for RNNs.

In an RNN, the output of a unit in step $n + 1$ does not only depend on the external exits of the network in the previous step $(u(n - k), k = 0, ..., M - 1)$, but also of the previous outputs of units $y_k(n), k = 1, ..., N$, having feedback inputs in the recurring layer, allowing the network to keep information in memory through the epochs.

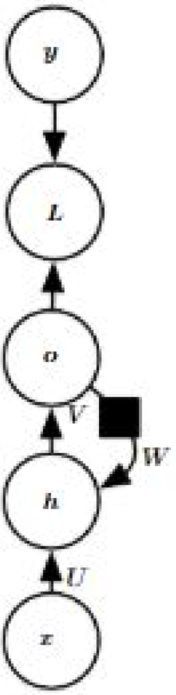
An RNN works by following the next two Equations 2.4 and 2.5:

$$v_k(n+1) = \sum_{m=1}^{N} w_{km}(n) y_m(n) + \sum_{m=N+1}^{N+M} w_{km}(n) u(n-m+N+1) \tag{2.4}$$

$$y_k(n+1) = \varphi(v_k(n+1)) \tag{2.5}$$

In which $w_{km}(n)$ stands for the weight of the connection between units m and k in step n, and φ is the activation function, such as sigmoid, tanh or ReLU, given by Equation 2.4.

Figure 4 – An RNN whose only recurrence is the feedback connection from the output to the hidden layer. At each time step t, the input is x_t, the hidden layer activations are $h^{(t)}$, the outputs are $o^{(t)}$, the targets are $y^{(t)}$, and the loss is $L^{(t)}$. Source: Goodfellow, Bengio and Courville (2016).



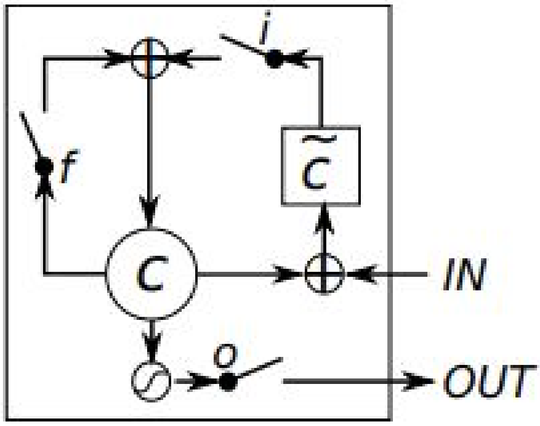
2.6.9 Long Short Term Memory Networks – LSTM

After running an RNN for many epochs, a known problem that may occur is called the *vanishing gradient problem*, in which the gradient of the loss function decays exponentially with time, effectively preventing the weight from changing its value. In this scenario, the RNN becomes stalled and does not offer any valuable prediction capabilities.

A Long Short Term Memory Network (LSTM) is a different type of RNN architecture destined to avoid the vanishing gradient problem. LSTM networks use select units in addition to standard units. Those units add *memory cells* that can incorporate data inside memory for more extended periods than an ordinary RNN could.

A set of *gates*, namely the *input gate*, the *output gate*, and the *forget gate* is used to control whether and when data enters the memory, the emission of an output, and when to forget the previous data from processing in the next epochs, respectively. This RNN architecture allows the network to keep only valuable information to the subsequent epochs, effectively learning dependencies for a longer time (*Long Term*), and discarding information that does not add to the model (*Short Term*).

Figure 5 – Illustration of a LSTM, in which i, f, and o are the input, forget, and output gates, respectively, while c and $\tilde{c}$ denote the memory cell and the new memory cell content. Source: Chung et al. (2014a).



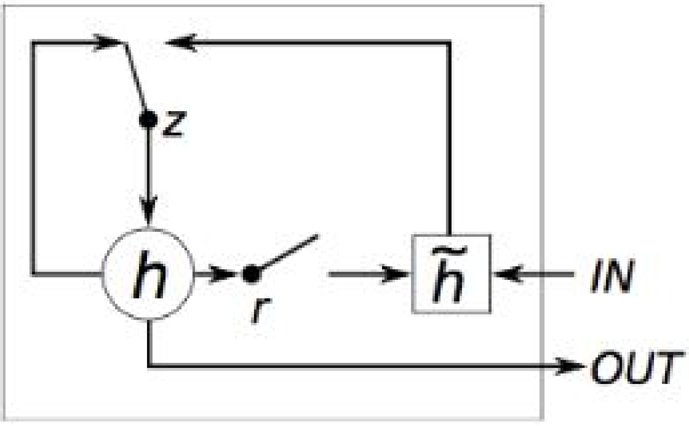
An example of LSTM can be seen in Figure 5. After it receives x_t, an activation function gives the value to an element-wise multiplication, which is given to the cell C. The output is the fed to input i_t, output o_t and forget f_t gates, at time t. The exit arrows going from C_t to the gates are destined to keep the current state to the next input, effectively working as an $t-1$ input.

2.6.10 Gated Recurring Unit Networks - GRU

Gated Recurring Unit networks are based on the LSTM architecture, with a few notable differences, like the absence of memory cells and the absence of an output gate. To compensate those losses, GRUs work by operating a *reset gate* and an *update gate*. The reset gate works by measuring the previous activation with the next candidate activation, in order to discard (forget) the previous state, while the update gate works by deciding whether it will use the candidate activation to update the cell state.

An example of GRU can be seen in Figure 6. In the model, r is the reset gate, z is the update gate, h is the current activation, and $\tilde{h}$ is the candidate activation. While LSTMs restricts the cell state through the control of its gates, GRUs expose the memory content to other units in the architecture. Without the restrictions, and with a simpler model, GRUs are usually faster to train than LSTMs.

Figure 6 – Illustration of a GRU, in which r and z are the reset and update gates, and h and $\tilde{h}$ are the activation and the candidate activation, respectively. Source: Chung et al. (2014a).



2.6.11 Hierarchical Attention Networks – HAN

Hierarchical Attention Networks, as described by Yang et al. (2016), are a neural network architecture that highlights the importance of individual words or sentences in the construction of the representation of a document. Since not all terms are equally important to the classification of a text, and sentences do not all represent the same meaning, this model stresses the most important sequences that affect the document's class.

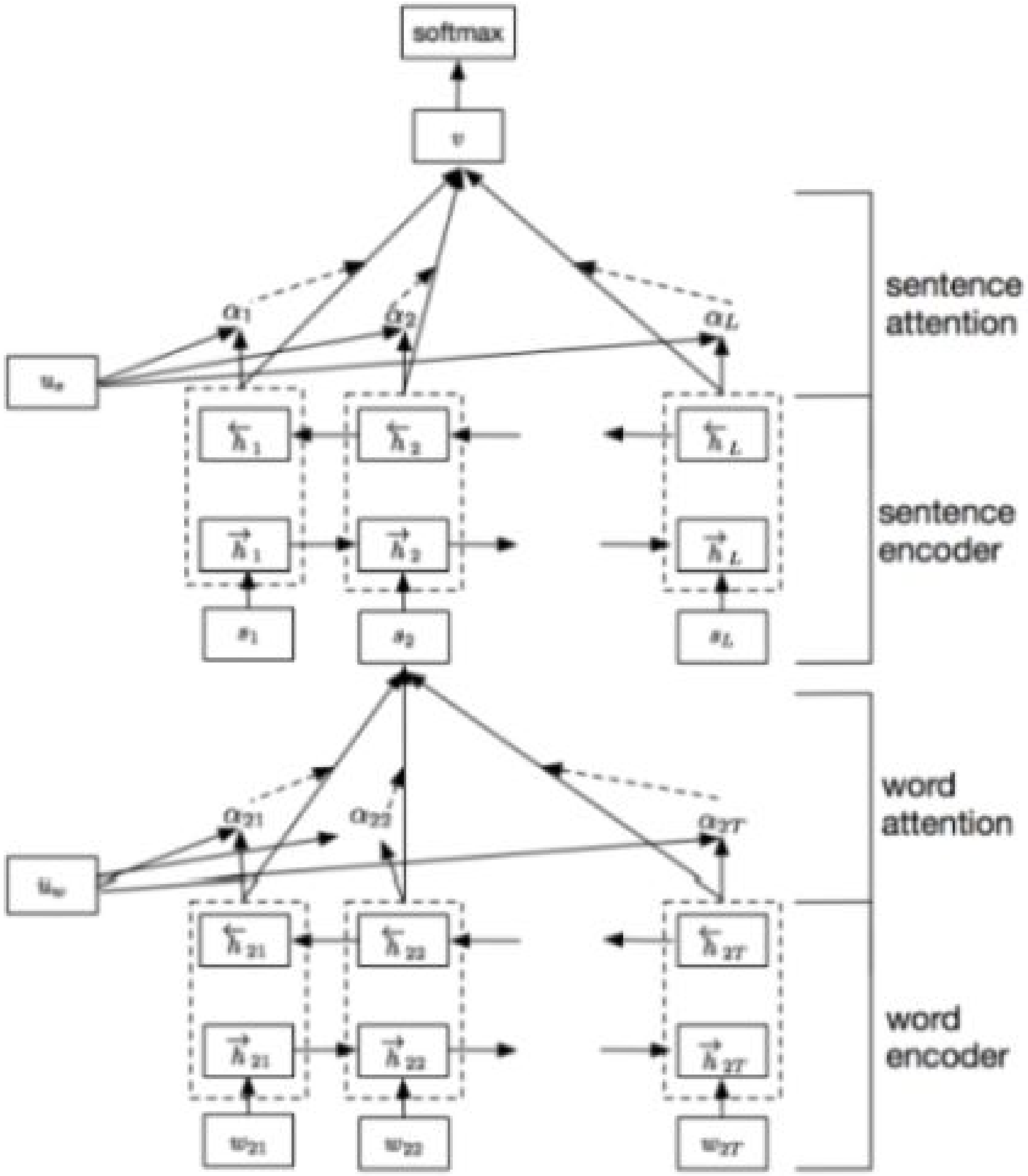
HANs are usually composed by 6 layers:

- An *embedding layer*, responsible for creating a matrix with the characteristics (size of the vocabulary, the maximum length of sentences) of the documents being processed

- A *word sequence encoder*, a word-level bi-directional GRU to get a rich representation of words

- A *word attention layer*, a layer to get important information in a sentence

Figure 7 – Diagrammatic example of a HAN. Source: Yang et al. (2016).



- A *sentence encoder*, a sentence level bi-directional GRU to get a rich representation of words
- A *sentence attention layer*, a layer to get important information in a sentence
- A *final layer*, destined to fully connect all the previous outputs and apply a softmax activation function.

HANs work by queuing a substructure called word encoder, followed by another substructure called sentence encoder. The first applies attention to each of the words inputted in order to form sentence representations. The second applied attention to each of the sentences received before to create document representations. An example of a HAN can be seen in Figure 7.

For the **word encoder**, the following structure was built:

- An **input layer**, to receive the output generated by GloVe, the tokens w_{it}, representing word i per sentence t, in a matrix of None per N, with N representing the maximum words in a sentence in the dataset. It is essential to mention that the term *None* is used by Keras, the Python framework that we have used, to represent any scalar number so that we can use this model to infer on an arbitrarily long input.

- To make the model understand sequences of characters, we have an **embedding layer**, destined to process strings. This layer assigns multidimensional vectors $W_e w_{ij}$ to each token. Therefore, words are represented numerically as x_{it}, as a projection of the term in a continuous vector space. There are many embedding methods available. For this research, we have used the GloVe framework. This layer will output a matrix with None per N per the number of dimensions of the word embedding training file, in our case, 600.

- The third layer contains an **encoding layer**, in our case, a Bidirectional GRU, to encode the data. The bidirectionality works by reading the sentence from the first to the last word, and reversing the order afterward, in order to understand the connections between words in the left and the right. As an example, in the sentence *The black car is beautiful*, the term *black* relates directly to the word *car*, as it gives a character to the word, and the word *is* also represents a strong correlation to *car*, indicating that the following word will describe its nature. The context annotations outputted are represented by h_{it}.

- The following **dense layer** works by applying the activation function (in our case, ReLU, to counter the vanishing gradient problem) to return the output of the neural network.

- Afterward, the result is processed in the **word attention layer**, which is an MLP destined to learn the importance of the words through training with randomly initialized weights (W), biases (b), and the outputs of the encoding layer, as in the Equation 2.6:

$$u_{it} = tanh(W_w h_{it} + b_w) \tag{2.6}$$

After that step, the result u_{it} is then multiplied by a trainable context vector u_w, and normalized to an importance weight per word α_{it} by a softmax function, described in Equation 2.7. The word context vector u_w is randomly initialized and jointly learned during the training process.

$$\alpha_{it} = \frac{\exp(u_{it}^T u_w)}{\sum_t \exp(u_{it}^T u_w)} \tag{2.7}$$

Finally, those importance weights α_{it} are multiplied by the context annotations h_{it}, being called sentence vectors, and are inputted into the sentence encoder. This operation is described in Equation 2.8.

$$s_i = \sum_t \alpha_{it} h_{it} \tag{2.8}$$

As for the **sentence encoder**, the following structure was built:

- The **input layer** receives the result from the last layer of the word attention, with a matrix of None per M per N, with M being the maximum number of sentences in one document, and N being the maximum words in a sentence in the dataset.
- The second layer represents the **time distributed model**, responsible for wrapping every input it receives as a dense layer, applying to all word-level layers on each sentence. In contrast, a regular dense layer would compute all the inputs as single N units.
- As in the word encoder, an **encoding layer** is used, in our case, a Bidirectional GRU. As previously mentioned, the GRU is used to understand the semantic relations between the sentences.
- After, a **dense layer** is stacked, with ReLu activation, to retrieve the outputs h_i.
- Finally, the result is then inputted into the **sentence attention layer**. It works in a similar way to the word attention layer, but the final output is a document vector v, which can be used as features for document classification. Trainable weights and biases are again randomly initialized and jointly learned during the training process. The operation is described in Equations 2.9, 2.10, and 2.11.

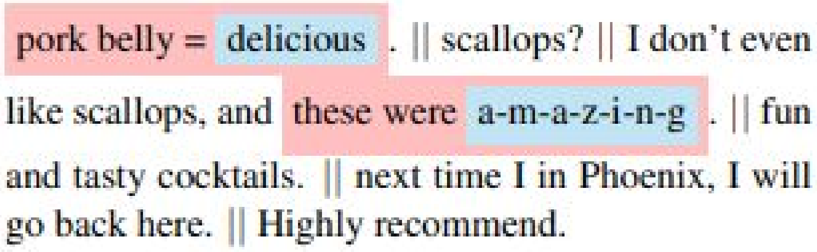
$$u_{it} = tanh(W_s h_i + b_s) \tag{2.9}$$

$$\alpha_{it} = \frac{\exp(u_i^T u_s)}{\sum_t \exp(u_i^T u_s)} \tag{2.10}$$

$$v = \sum_i \alpha_i h_i \tag{2.11}$$

After the processing of the HAN networks, every word gets an attention coefficient, indicating the importance of that word in its sentence. An example from Yang et al. (2016) is shown in Figure 8. The can see that sentence 1 (*"pork belly = delicious"*) and the final words of sentence 3 (*"these were a-m-a-z-i-n-g"*) are marked in pink. This highlight happens because the HAN has implied that those two sentences are among the most important of that text - that is, they are among the highest sentence attention weights. Inside those sentences, two words are marked in blue. That represents that those two words carry the most important terms in those sentences, also having the highest word attention weights.

Figure 8 – Example of attention generated by the HAN. Source: Yang et al. (2016).



pork belly = delicious . || scallops? || I don't even like scallops, and these were a-m-a-z-i-n-g . || fun and tasty cocktails. || next time I in Phoenix, I will go back here. || Highly recommend.

HANs are currently one of the most popular neural network algorithms being adopted in Computer Science. In 2020, the year of publication of this research, we have seen their applications in many different fields, such as Finance, to predict stock market values - Huang et al. (2020); Linguistics, in the classification of historical documents - Kim et al. (2020); Technology, in mobile app recommendations - Liang et al. (2020); and Medicine, with detection models for Atrial Fibrillation - Mousavi, Afghah and Acharya (2020). We believe this flexibility shows the strength of the Attention model to handle different texts across many different vocabularies.

2.7 Evaluation measures

As it is known, researches in the fields of NLP, including automatic classification of texts and data recovery, should be consistently evaluated so as to compare the efficacy of the results. It is common to use standard quality evaluation measures to achieve this comparison, such as accuracy, precision, recall, and F-measure.

2.7.1 Accuracy

The most standard measure of the efficiency of any experiment is *accuracy*. The accuracy shows the division between the number of correct predictions over the total number of documents, as shown below in Equation 2.12. In this formula, we sum the number of True Positives (*tp*) with the number of True Negatives (*tn*), and divide that sum by the total sum of all possible results (True Positives, True Negatives, False Positives, False Negatives).

$$\text{Accuracy} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \tag{2.12}$$

2.7.2 Precision and recall

Besides accuracy, we can also evaluate an algorithm by selecting a specific class from all the classes being studied. If one class is more critical to the research being conducted than the others, perhaps accuracy will not be a useful measure, since the algorithm may be having a high accuracy because it is classifying all the texts in the wrong class.

Alternative measures may prove to be more effective to address this problem. Two of the most used are *precision* and *recall.* In order to better explain those concepts, it is useful to use a contingency table, which shows the semantics of possible combinations of text classification, matching actual classes with predicted classes.

Table 5 – Contingency Table

		Actual Class	
		Yes	**No**
Predicted Class	**Yes**	True Positive (tp)	False Positive (fp)
	No	False Negative (fn)	True Negative (tn)

From the contingency table, we can infer the equations for precision and recall. According to Manning and Schütze (1999), precision is defined as the relevant fraction of recovered documents, while recall is the recovered fraction of relevant documents. The formula for precision can be seen in Equation 2.13, and the formula for recall can be seen in Equation 2.14.

$$\text{Precision} = \frac{\text{tp}}{\text{tp+fp}} \tag{2.13}$$

$$\text{Recall} = \frac{\text{tp}}{\text{tp} + \text{fn}} \tag{2.14}$$

Along this line of thinking, accuracy can be understood as the fraction of correct hits among the total amount of documents, as shown in Equation 2.15.

$$\text{Accuracy} = \frac{\text{correct hits}}{\text{total number of documents}} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}} \tag{2.15}$$

2.7.3 F-measure

With all the multitude of different evaluation measures, a new problem arises: the existence of a unified measure of comparison. Depending on the nature of the research, a small loss of precision with an improvement in the recall of the results is acceptable. Even though the balance of those measures strongly depends on the context of the research, it is possible to express this balance by using an equation called *F-measure.*

F-measure is defined by Manning, Raghavan and Schütze (2008) as the harmonic mean between precision (P) and recall (C), and can be expressed by Equation 2.16.

$$\text{F-measure} = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PC}{\beta^2 P + C} = \frac{(tp}{tp + \frac{1}{2}(fp + fn)} \tag{2.16}$$

Where α in this equation represents the weight between precision and recall, which relates to β by Equation 2.17.

$$\beta^2 = \frac{1-\alpha}{\alpha} \tag{2.17}$$

In the cases where precision and recall have the same importance and weight, F-measure is called F_1, since, in this case, $\beta = 1$ ($\alpha = 0.5$). The equation can be simplified if used in this manner, as we can see in Equation 2.18.

$$F_{\beta\,=\,1} = \frac{2PC}{P+C} \tag{2.18}$$

2.7.4 K-Fold cross-validation

In K-Fold Cross-Validation, the original dataset is randomly sliced in k disjoint parts, of approximately equal size. Those partitions are called *folds.* The classifier is then trained from a dataset composed of $k-1$ folds, and the remaining fold is used as a validation set.

In this method, it is essential to split the dataset d in K parts of equal size m_k, in which $\sum_{k=1}^{K} m_k = n$. The whole process has to have k iterations, and in each iteration, the testing set will be given by d_k, with $k = 1, 2, ..., K$, and the training set for the algorithm will be the sum of the other $K-1$ parts, i.e., $d_{(-k)} = d_1, d_2, ..., d_{k-1}, d_{k+1}, ..., d_K$. Therefore, at the end of the k iterations, we will have used all the data available in both training and testing steps.

3

Experiments and Results

In this chapter, we present the experiments and results found after the datasets were processsed for each of the algorithms chosen, both neural networks and non-neural networks. We also list the word attention weights for each of the datasets. Lastly, we show the top-ranked words in all scenarios.

We highlight that several different combinations of hyperparameters were tried before the optimal match was found. In every case that we do not mention the hyperparameters used, the standard set offered by the framework was the best choice.

3.1 Non-Neural Networks

We have used K-Fold Cross-Validation, as mentioned in the previous chapter, with 10 Folds. The algorithms selected were:

- Logistic Regression (LR);
- Linear Discriminant Analysis (LDA);
- K-Nearest Neighbors (KNN);
- Classification and Regression Trees (CART);
- Naïve Bayes (NB); and
- Support Vector Machines (SVM)

The same methodology was applied to the homicides and the corruption datasets.

3.1.1 Homicides Dataset

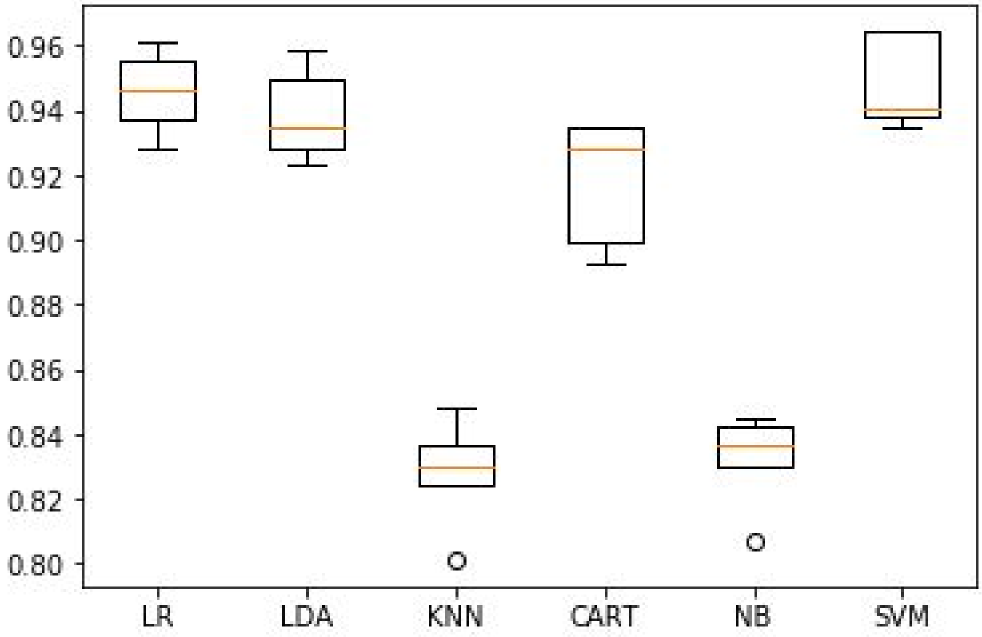
The metrics for the homicides dataset, after the first experiment, are shown in Table 6. We can see that Logistic Regression, Linear Discriminant Analysis and Support Vector Machines showed the highest performance, with high values of precision, recall, f-score and accuracy. Support Vector Machines showed the hiuhest performance in 3 of the 4 metrics. Regression Trees showed the highest value in recall.

Table 6 – Metrics for the algorithms, in the homicides dataset, with mean values after 10-Fold cross-validation.

Algorithm	**Precision**	**Recall**	**F-Score**	**Accuracy**
Logistic Regression	0.948895	0.934847	0.939733	0.941117
Linear Discriminant Analysis	0.921595	0.928063	0.922120	0.923271
K-Neighbors	0.779389	0.820864	0.795847	0.795330
Regression Trees	0.888953	**0.954632**	0.888741	0.892924
Naive Bayes	0.651370	0.894028	0.769831	0.723989
Support Vector Machines	**0.951587**	0.933694	**0.940827**	**0.952380**

We can see the variation, for the 10 executions, on the accuracy of each algorithm, by analyzing the boxplot graphic shown in Figure 9. We can see the SVM shows the best accuracy overall, keeping a relatively average standard deviation, compared to the other algorithms.

Figure 9 – Boxplot of accuracies for 10 executions of the algorithms on the homicides dataset



3.1.2 Corruption dataset

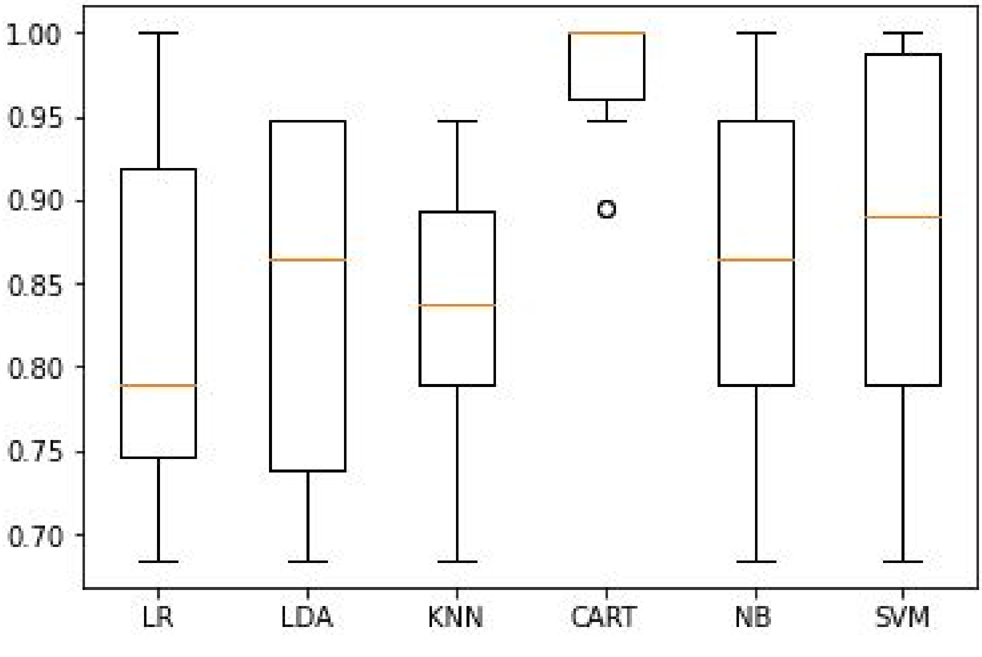
The metrics for the corruption dataset, after the first experiment, are shown in Table 7. We can see that Regression Trees were the best algorithm, scoring the highest value in all of the four metrics.

Table 7 – Metrics for the algorithms, in the corruption dataset, with mean values after 10-Fold cross-validation.

Algorithm	Precision	Recall	F-Score	Accuracy
Logistic Regression	0.814269	0.827583	0.870421	0.824269
Linear Discriminant Analysis	0.859783	0.828997	0.807658	0.839766
K-Neighbors	0.851981	0.828655	0.897243	0.824854
Regression Trees	**0.967917**	**0.968705**	**0.973648**	**0.968421**
Naive Bayes	0.876901	0.899474	0.925169	0.866374
Support Vector Machines	0.876901	0.872200	0.930724	0.876901

We can see the variation, for the 10 executions, on the accuracy of each algorithm, by analyzing the boxplot graphic shown in Figure 10.

Figure 10 – Boxplot of accuracies for 10 executions of the algorithms on the corruption dataset



After we did our testing, we could infer that Regression Trees are the method which showed highest accuracy in both data sets chosen, even though the other algorithms showed varying results. SVM, as an example, showed a good performance in the homicides dataset but did not match the results Regression Trees showed in the corruption dataset. Regression Trees have always kept good predicting outcomes. Those results match other results found by other researches in the legal area in many different countries, such as the one conducted by Kastellec (2010), who obtained good outcomes by using Regression Trees in the American legal system. The author mentions that Regression Trees have the capability of studying intrinsic conceptions of Law, revealing patterns that other methods cannot emulate as effectively.

Other researchers also used the same method, such as Rios-Figueroa (2011), who used Regression Trees to analyze the concept of judicial independence and corruption among Supreme Courts in Latin America, Antonucci, Crocetta and D'Ovidio (2014), who adopted Regression Trees to measure the efficiency of Italian courts, and Kufandirimbwa and Kuranga (2012), who used the same algorithm to predict outcomes in Zimbabwe.

Those researches show that, even though legal systems are exceedingly different around the world and throughout other languages and countries, such as Brazil, the United States, Italy, and Zimbabwe, they do have similar characteristics that can be effectively measured by the correct algorithms. In that way, we can see that legal texts might have intrinsic factors that remain even when languages change.

3.2 Neural Networks

For the experiments involving neural networks, we have applied the following algorithms:

We have used K-Fold Cross Validation, as mentioned in the previous chapter, with 10 Folds. The algorithms selected were:

- Multilayer Perceptron (MLP);
- Recurrent Neural Networks (RNN);
- Long Short Term Memory (LSTM);
- Gated Recurring Unit Networks (GRU); and
- Hierarchical Attention Networks (HAN)

The same methodology was applied to the homicides and the corruption datasets. All tests were run using a GloVe file destined for Brazilian Portuguese, with 600 embedding dimensions.

The convergence criteria were defined by modeling the learning rate and loss functions. For the learning rate, it would be lowered by 0.2 every 3 epochs that the loss function would not lower. For the loss function itself, the algorithm stops if, after 5 epochs, it does not decrease at least 0.001.

3.2.1 Multilayer Perceptron

For the tests using Multilayer Perceptrons, an architecture of 3 hidden layers was used, with 512, 512 and 250 neurons, respectively, running on 25 epochs. This architecture was proven to be the most cost-effective, offering the best results with the computational resources available. The results are shown below, in Table 8.

Table 8 – Metrics for the MLP, with mean values after 10-Fold cross-validation.

Dataset	**Precision**	**Recall**	**F-Score**	**Accuracy**
Homicides Dataset	0.981267	0.986228	0.983620	0.984562
Corruption Dataset	0.749448	0.987332	0.855894	0.749448

The confusion matrices for both datasets are shown in Table 10 and Table 9. The matrices were obtained after the 10th execution for the 10th fold of the algorithms. We can see that the MLP showed an excellent result for the homicides dataset, but failed to predict the absolutions for the corruption dataset correctly. This characteristic may have caused the low accuracy shown in Table 8.

Since the MLP does not consider the recurrency of the words, and our corruption dataset was smaller than the homicides dataset (786 vs. 1681 cases), we presume that the MLP did not manage to effectively learn how to predict outcomes for corruptions, due to the volume of tokens processed. This result is also shown in the confusion matrices, where we can see a perfect score for the homicides, but a failure for the corruption texts, where it wrongly predicted all the 78 test cases as condemnations. In this point of the research, we decided to use recurrent networks, to see if they would be able to capture the relation between words. As we can see in the following sections, we did not have notable improvements using RNNs, but the other recurrent networks showed higher accuracies.

Table 9 – Confusion matrix for the MLP, using the homicides dataset, after 10-fold.

n = 1681	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	833	11
Actual Condemnations	15	822

Table 10 – Confusion matrix for the MLP, using the corruption dataset, after 10-fold.

n = 780	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	2	158
Actual Condemnations	5	615

3.2.2 Recurrent Neural Networks

For the tests using Recurrent Neural Networks, an architecture of 1 hidden layer with 128 units was used, with a 0.5 probability of dropout for the hidden layer, and a 0.2 dropout for the inputs. We used sigmoid as the activation function, and binary cross-entropy as the loss function, running on 25 epochs. The results are shown below, in Table 11.

Table 11 – Metrics for the RNN, with mean values after 10-Fold validation

Dataset	**Precision**	**Recall**	**F-Score**	**Accuracy**
Homicides Dataset	0.864426	0.882808	0.866621	0.853257
Corruption Dataset	0.797958	0.989252	0.882732	0.804203

The confusion matrices for both datasets are shown in Table 12 and Table 13. The matrices were obtained after the 10th execution for the 10th fold of the algorithms. This method showed better results than the MLP but still failed to predict all the absolutions in the corruption dataset. Also, comparing to the MLP, we now have lower results for the homicides dataset, mispredicting some of the absolutions, and showing a lower accuracy.

We can see that RNNs increased the accuracy of the corruptions, but diminished the previous accuracy found in the MLPs for the homicide cases. As shown in Table 12, it mistakenly predicted some absolutions as condemnations. The same phenomenon happened for the corruptions. Further experiments could be made, increasing the volume of cases in both datasets, to see if the accuracies of both MLP and RNN could show significant improvements.

Another interesting point to notice is that, in both MLP and RNN, the mistakes shown in the confusions matrices all involve condemnations, which could indicate a sign of bias towards condemnations in both networks. What we would expect to see is the mistakes evenly distributed in both outcomes, especially in the homicides dataset, since the numbers of absolutions and condemnations were also evenly collected - 50.2% and 49.7%, respectively. However, this is not the case. We presume that this happens due to a larger number of unique word tokens used for condemnations, in both datasets, as shown in Sections 3.2.8 and 3.2.9. Even though the number of cases for each outcome is evenly distributed in the homicides dataset, for example, judges tend to write more in condemnations. With more words, the network could show a bias towards the outcome with more tokens, showing a limitation of both networks.

Table 12 – Confusion matrix for the RNN, using the homicides dataset

n = 1680	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	706	84
Actual Condemnations	7	883

Table 13 – Confusion matrix for the RNN, using the corruption dataset

n = 780	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	63	147
Actual Condemnations	0	570

3.2.3 Long Short-Term Memory Networks

For the tests using Long Short-Term Memory Networks, an architecture of 1 hidden layer with 128 units was used, with a 0.2 probability of dropout for the hidden layer, and a 0.2 dropout for the inputs. We used sigmoid as the activation function, and binary cross-entropy as the loss function, running on 25 epochs. The results are shown below, in Table 14.

Table 14 – Metrics for the LSTM, with mean values after 10-Fold validation

Dataset	**Precision**	**Recall**	**F-Score**	**Accuracy**
Homicides Dataset	0.986512	0.985387	0.985890	0.986355
Corruption Dataset	0.940756	0.986466	0.962574	0.942762

The confusion matrices for both datasets are shown in Table 15 and Table 16. The matrices were obtained after the 10th execution for the 10th fold of the algorithms. The failure rates were lowered for the LSTM, compared to the RNN. The overall accuracy was also increased for both datasets.

We presume that the increased accuracies could be explained by the additional controlling knobs, the forget and output gates, that compose the LSTM model, when compared to the RNN, offering more flexibility in handling the outputs and allowing for better control over the gradient flow and enabling better preservation of long-range dependencies. As judicial texts can have significantly extense sentences, learning how to keep essential tokens highlighted among many words could be the key to effectively predict the outcome of those texts.

We also see a paramount difference from the two previous neural networks: mistakes shown in confusion matrices are now evenly distributed, virtually eliminating the “condemnation bias” shown in Sections 3.2.1 and 3.2.2. In LSTMs, long sentences are not treated just as many unique word tokens, but as expressions of meaning that may carry elaborate explanations.

Table 15 – Confusion matrix for the LSTM, using the homicides dataset.

n = 1680	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	775	15
Actual Condemnations	32	858

Table 16 – Confusion matrix for the LSTM, using the corruption dataset

n = 780	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	85	55
Actual Condemnations	11	629

3.2.4 Gated Recurring Units

For the tests using Gated Recurring Units Networks, an architecture of 1 hidden layer with 128 units was used, with a 0.2 probability of dropout for the hidden layer, and a 0.2 dropout for the inputs. We used sigmoid as the activation function, and binary cross-entropy as the loss function, running on 25 epochs. The results are shown below, in Table 17.

Table 17 – Metrics for the GRU, with mean values after 10-Fold validation

Dataset	**Precision**	**Recall**	**F-Score**	**Accuracy**
Homicides Dataset	0.992026	0.993282	0.992625	0.992275
Corruption Dataset	0.996551	0.998245	0.997391	0.996551

The confusion matrices for both datasets are shown in Table 18 and Table 19. The matrices were obtained after the 10^{th} execution for the 10^{th} fold of the algorithms. GRU revealed high accuracy for both datasets. As for the confusion matrices, it had a perfect score for the corruption dataset, and a nearly perfect score for the homicides dataset, missing only one condemnation, wrongly labeled as an absolution.

We can see that the accuracy gains shown for the LSTMs are kept for the GRUs, and the mistakes are decreased, even with a simplified structure, with only two gates (the reset and the update gate). One important point to mention is that, throughout our experiments, the GRU was faster to train than the LSTM. Therefore, we have higher accuracy values combined with faster training times, which would point to the GRUs as good choices to handle judicial texts.

Our results match the ones found by Chung et al. (2014b), who have shown that the GRU is faster than the LSTM, but with comparable accuracies. The authors also mention that the choice of the type of network between LSTMs and GRUs may depend heavily on the dataset and corresponding task. Our research shows that, for our datasets, GRU is the best choice.

Table 18 – Confusion matrix for the GRU, using the homicides dataset

n = 1680	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	903	7
Actual Condemnations	23	747

Table 19 – Confusion matrix for the GRU, using the corruption dataset

n = 780	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	178	3
Actual Condemnations	2	597

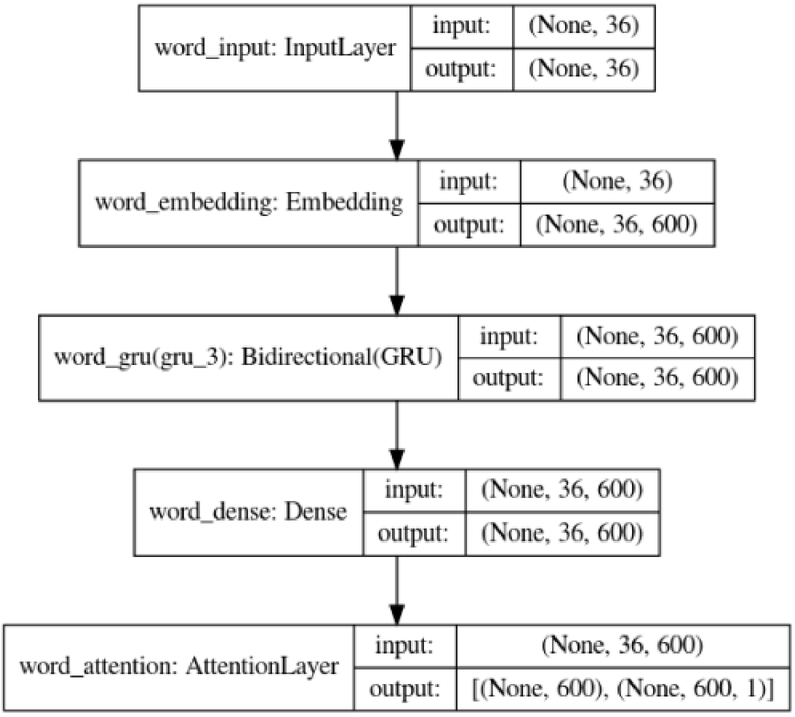
3.2.5 Hierarchical Attention Networks

For the tests using Hierarchical Attention Networks, a standard structure was built, in order to process both datasets. The values changed according to the characteristics of the dataset. After running the HAN on both datasets (homicides and corruption), we were able to highlight the key terms to condemn or absolve the defendant for each of the crimes.

3.2.6 Hierarchical Attention Networks - Corruption Dataset

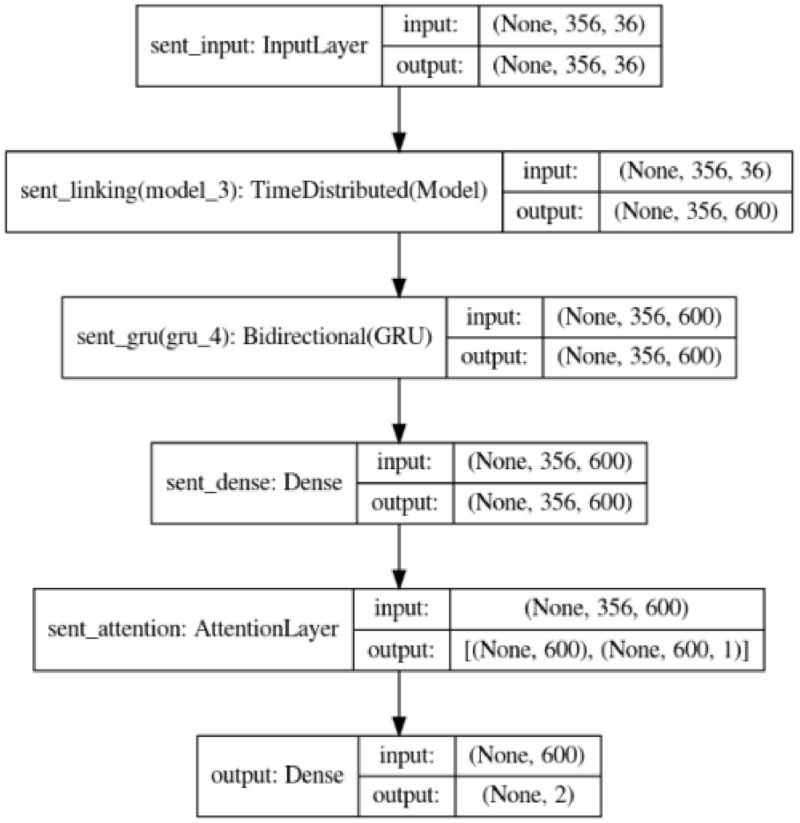
For the word encoder, the model adopted is described in Figure 11.

Figure 11 – Model for the HAN word encoder, using the corruption dataset



As for the sentence encoder, the model adopted is described in Figure 12.

Figure 12 – Model for the HAN sentence encoder, using the corruption dataset



The metrics for the corruption dataset are shown in Table 20, and the confusion matrix is shown in Table 21. The HAN showed the highest accuracy for the corruption dataset among all methods adopted, and a perfect score in the confusion matrix.

Our results match similar works that compare HAN against other neural network models found in the academia, such as the one by Gao et al. (2017), who used HANs to extract information from cancer pathology reports. Using F-Scores as the main comparison metrics, the authors found that micro and macro F-scores for the HAN with pretraining were (0.852, 0.708), compared to Naive Bayes (0.518, 0.213), Logistic Regression (0.682, 0.453), Support Vector Machines (0.634, 0.434), Random Forests (0.698, 0.508), Extreme Gradient Boosting (0.696, 0.522), RNNs (0.505, 0.301), and Convolutional Neural Networks (0.714, 0.460). In another research, Ma et al. (2019) compared HANs to Convolutional Neural Networks, SVMs, LSTMs and DeClarE to predict the outcome of claim verifications. The authors also found the highest values of precision, recall and F-Scores in the HAN networks. Tarnpradab, Liu and Hua (2018) also use HANs, to summarize online forum discussions, outperforming SVM and Logistic Regression models.

Table 20 – Metrics for the HAN in the corruption dataset, with mean values after 10-Fold validation

Dataset	**Precision**	**Recall**	**F-Score**	**Accuracy**
Corruption Dataset	0.985882	0.993251	0.985540	0.997853

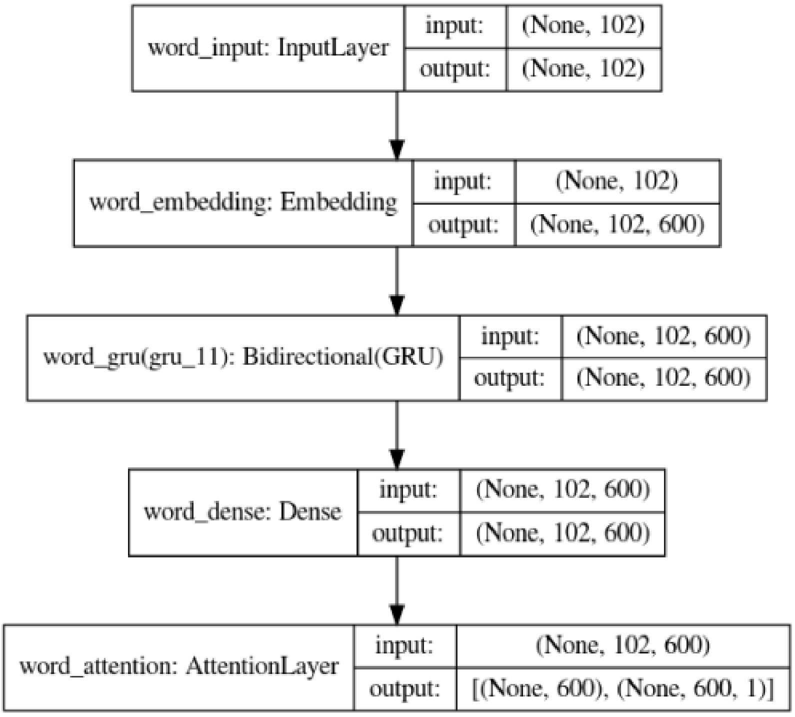
Table 21 – Confusion matrix for the HAN in the corruption dataset, after a 10-Fold validation

n = 790	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	268	2
Actual Condemnations	2	518

3.2.7 Hierarchical Attention Networks - Homicides Dataset

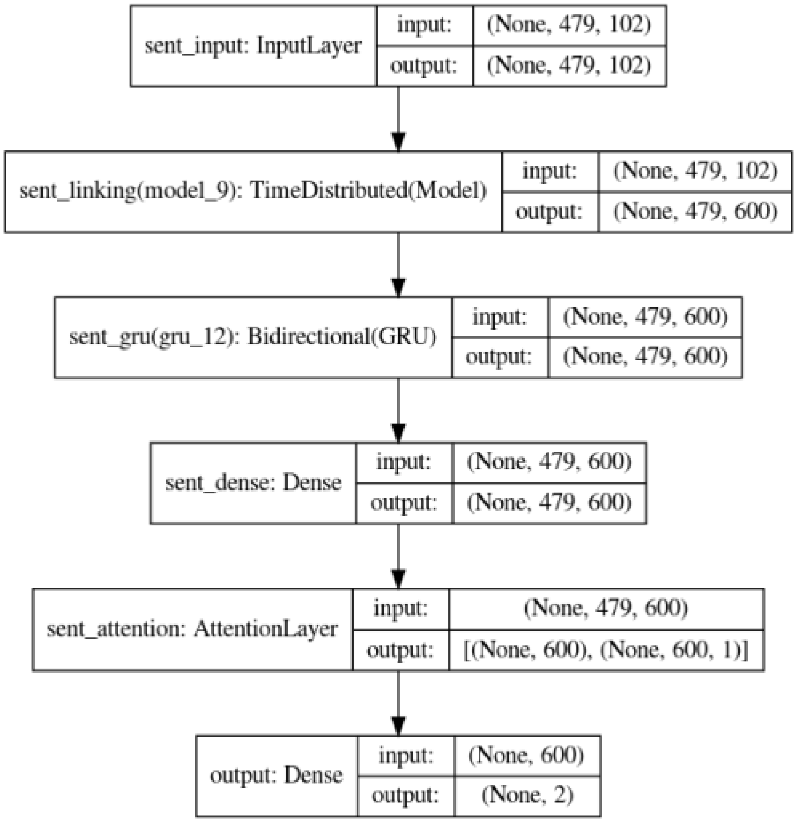
For the word encoder, the model adopted is described in Figure13.

Figure 13 – Model for the HAN word encoder, using the homicides dataset



As for the sentence encoder, the model adopted is described in Figure 14.

Figure 14 – Model for the HAN sentence encoder, using the homicides dataset



The metrics for the homicides dataset are shown in Table 22, and the confusion matrix is shown in Table 23. The HAN showed promising results for the homicides dataset, with an accuracy second best only to the GRU algorithm. The confusion matrix also showed a good outcome, missing only 2 cases among 168 overall.

As the HAN showed good results, comparable to the ones found by the GRU, but has a slower training time, it is indispensable to mention that both algorithms have proven to be choices with high accuracy for our datasets. GRUs have high accuracies but do not implement the Attention model, not giving Attention Weights to every word token. If the prediction of the outcomes is the sole interest of a research, GRUs could be the choice with the highest accuracy. If the analysis of the word tokens is necessary, HANs can be adopted without a significant loss of the overall accuracy.

Table 22 – Metrics for the HAN in the homicides dataset, with mean values after 10-Fold validation

Dataset	**Precision**	**Recall**	**F-Score**	**Accuracy**
Homicides Dataset	0.966543	0.986666	0.976097	0.986666

Table 23 – Confusion matrix for the HAN in the homicides dataset, after 10-Fold validation

n = 1680	Predicted Absolutions	Predicted Condemnations
Actual Absolutions	752	18
Actual Condemnations	12	898

3.2.8 Hierarchical Attention Networks - Attention Weights for the Homicides Dataset

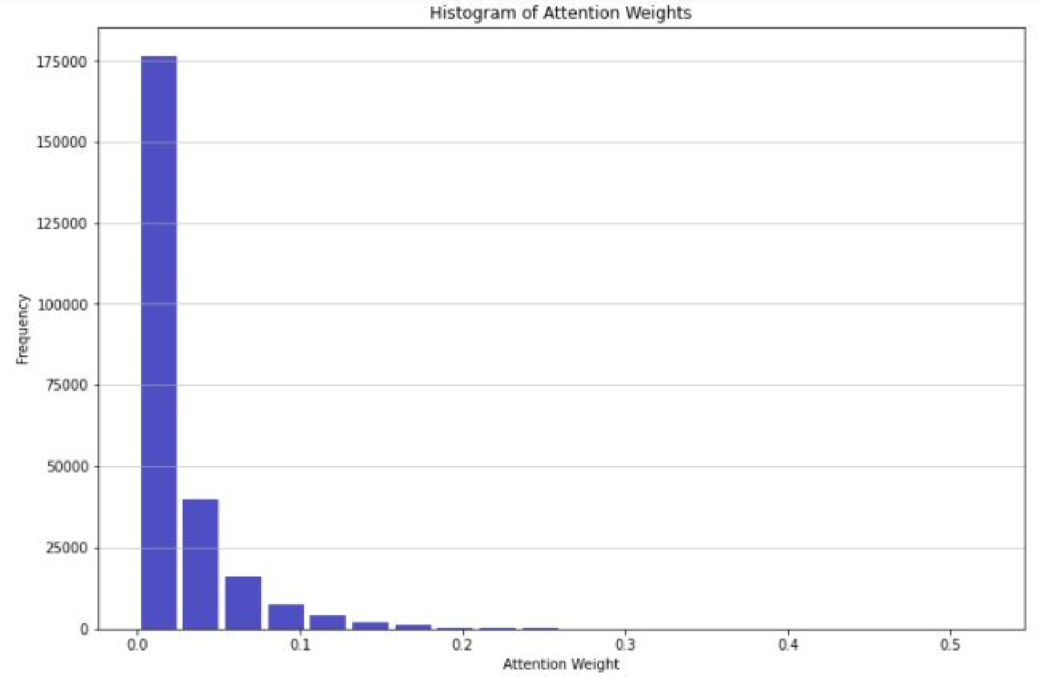
We have computed all the attention weights for every word in the datasets. Since words have different semantic meanings in distinct words, that have appeared in our attention weights dataset multiple times.

As the absolution and condemnation are different document classifications, words also might have different attention weights for both cases. This difference is the reason why we have mapped each of the datasets twice, for each outcome, in order to map all the possible words that affect the importance of each sentence.

3.2.8.1 Word Attention Weights for Homicide Absolutions

In total, we have found **248460** unique word tokens in the texts representing homicide absolutions. For each of the tokens, their attention weight was calculated. The histogram of attention weights is shown in Figure 15.

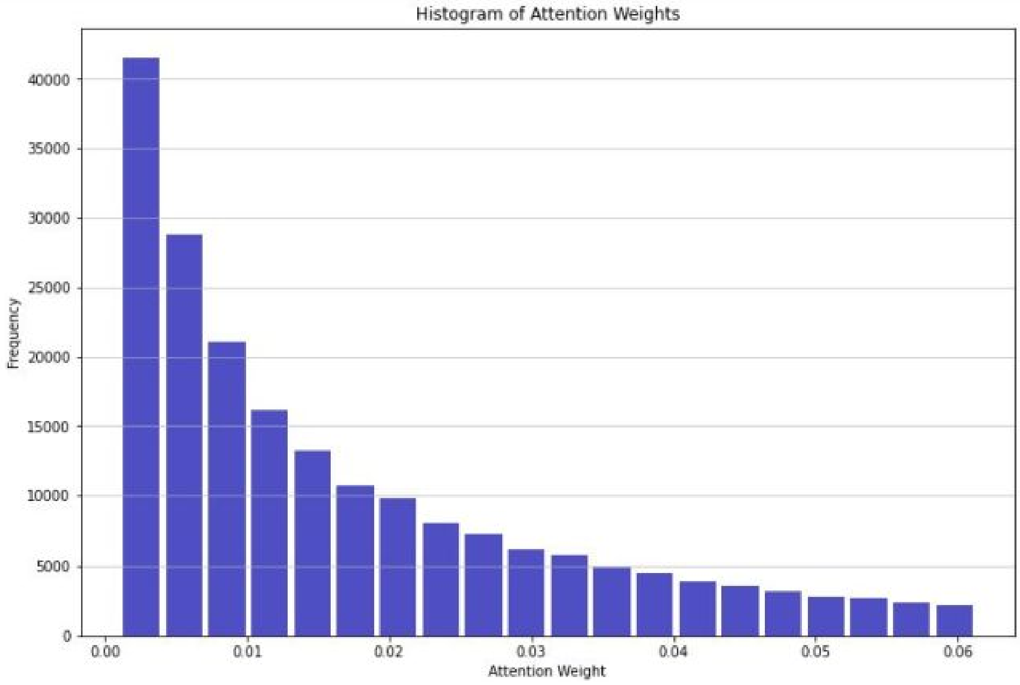
Figure 15 – Histogram of word attention weights for the absolutions in the homicides dataset



We can explain this graphic by applying Zipf's Law, which infers that the frequency of any word is inversely proportional to its rank in the relevance metric. Therefore, the relevant terms account for a small proportion of our dataset.

If we exclude the least and the most relevant words, we have a graphic according to Figure 16. In this figure, words in the top 10% and the low 10% were removed, with only the 80% middle words remaining. We can see that Zipf's Law continues to explain the behavior of the terms.

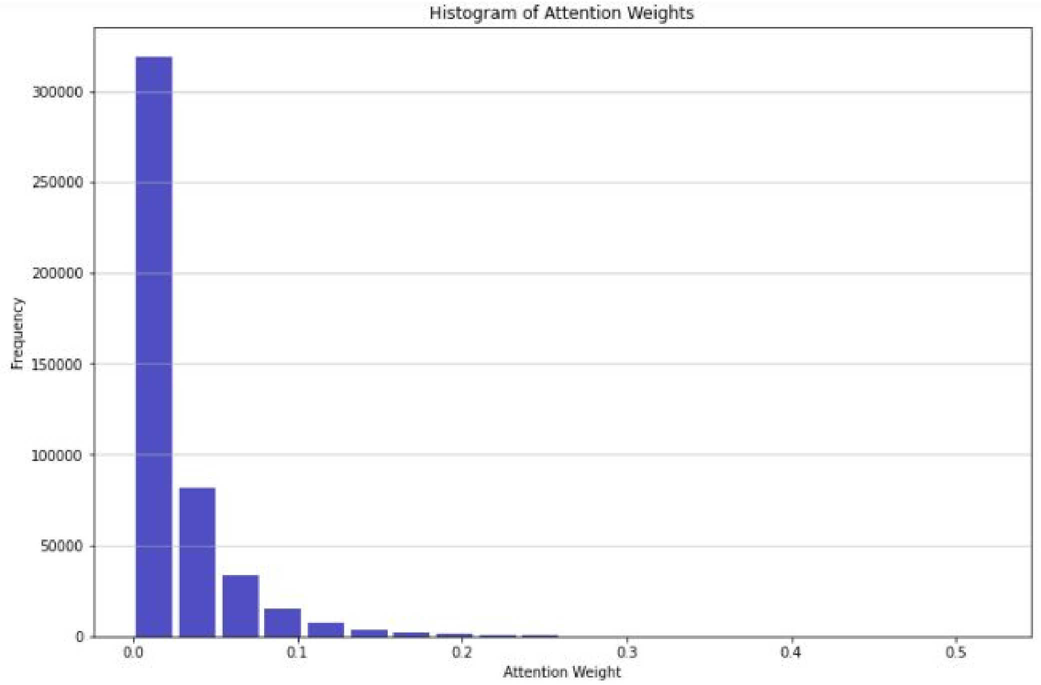
Figure 16 – Histogram of 80% word attention weights on the middle interval for the absolutions in the homicides dataset



3.2.8.2 Word Attention Weights for Homicide Condemnations

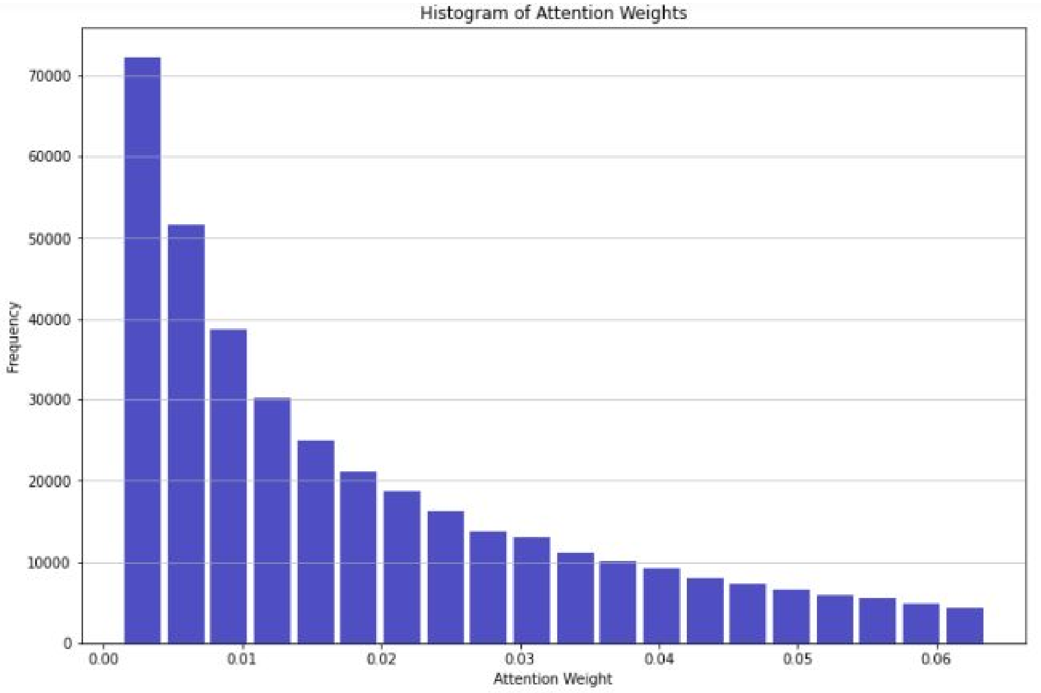
In total, we have found **466,461** unique word tokens in the texts representing homicide condemnations, the highest number of all datasets. For each of the tokens, their attention weight was calculated. The histogram of attention weights is shown in Figure 17.

Figure 17 – Histogram of word attention weights for the condemnations in the homicides dataset



Again, we can explain this graphic by applying Zipf's Law. The same pattern is repeated if we also select only the middle 80% words, as we can see in Figure 18.

Figure 18 – Histogram of 80% word attention weights on the middle interval for the condemnations in the homicides dataset



3.2.9 Hierarchical Attention Networks - Attention Weights for the Corruption Dataset

The same word attention weight analysis was also performed to the corruption dataset, in both outcomes (absolutions and condemnations).

3.2.9.1 Word Attention Weights for Corruption Absolutions

In total, we have found **66929** unique word tokens in the texts representing corruption absolutions. For each of the tokens, their attention weight was calculated. The histogram of attention weights is shown in Figure 19, and the same histogram with the middle 80% tokens is shown in Figure 20.

Figure 19 – Histogram of word attention weights for the absolutions in the corruption dataset

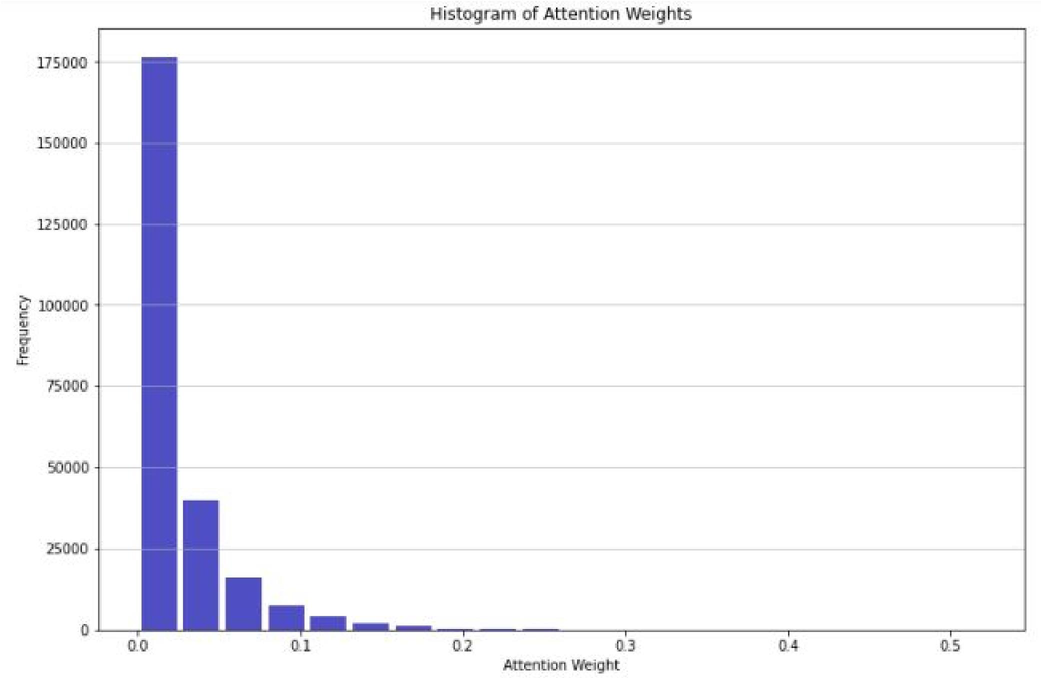
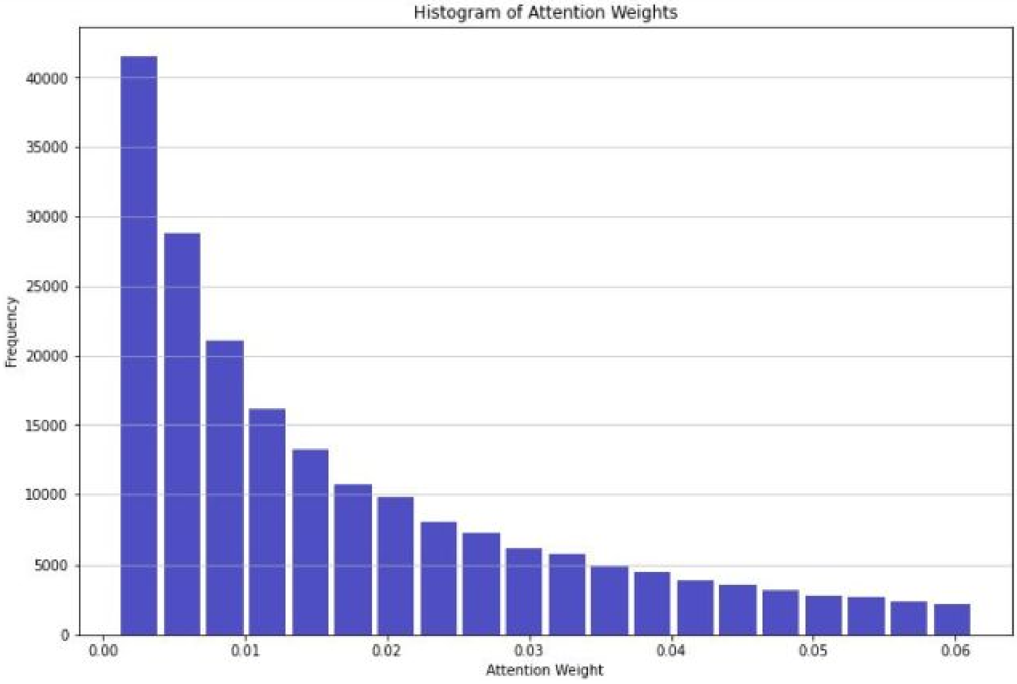


Figure 20 – Histogram of 80% word attention weights on the middle interval for the absolutions in the corruption dataset



3.2.9.2 Word Attention Weights for Corruption Condemnations

In total, we have found **252620** unique word tokens in the texts representing corruption condemnations. For each of the tokens, their attention weight was calculated. The histogram of attention weights is shown in Figure 21, and the same histogram with the middle 80% tokens is shown in Figure 22.

Figure 21 – Histogram of word attention weights for the condemnations in the corruption dataset

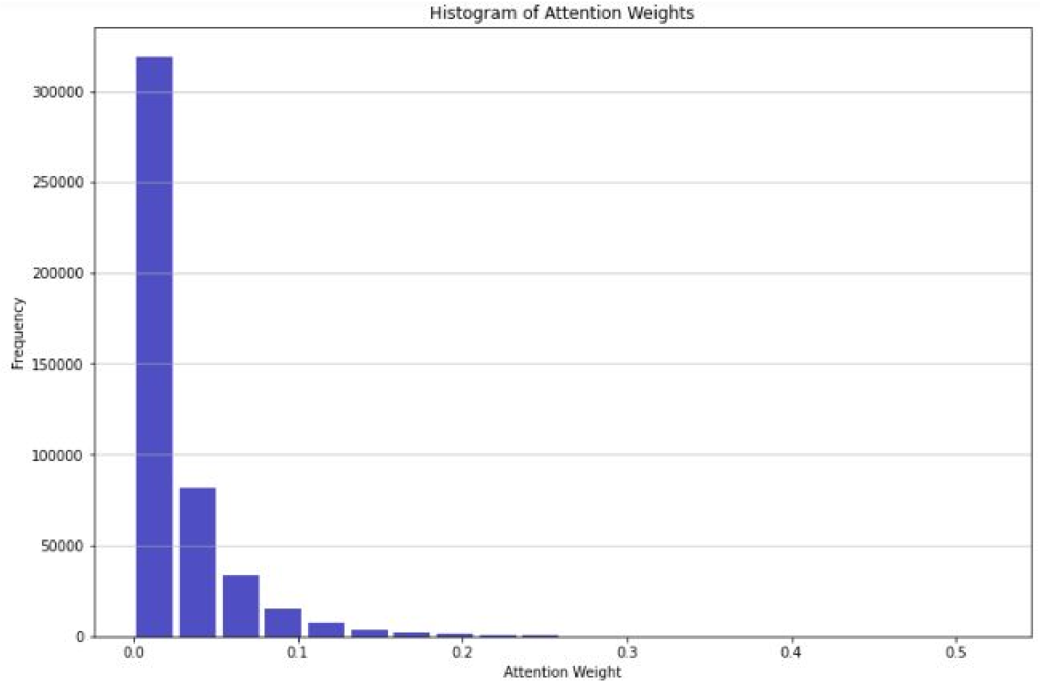
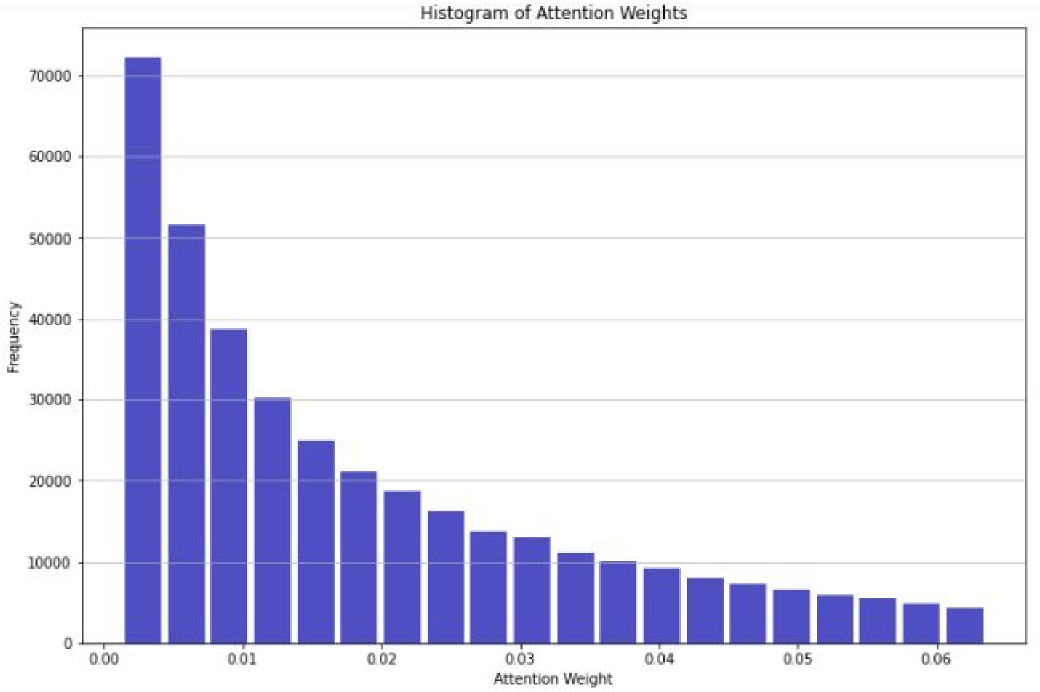


Figure 22 – Histogram of 80% word attention weights on the middle interval for the condemnations in the corruption dataset



3.2.10 Top Words for the Corruption Dataset

After the classification had been performed, we sought to order all the words in each of the datasets by their attention weights. Therefore, each word will have a unique value, ranging from 0 (where the word would have no importance in the classification of the document) to 1 (where the word would have maximum importance in the classification of the document).

It is useful to mention, as noted in Section 2, that a word might have different attention weights in distinct sentences. As a short example, the sentence *The defendant robbed a bank* and the sentence *The defendant did not participate in the robbery, because it was going to a blood bank* both have the word *bank*, but in very different contexts. In the first sentence, the word would be a vital contributor to the condemnation, while it would contribute to the absolution in the second sentence.

Therefore, words have appeared more than once in our final calculations, with different attention weights. The list with the top 50 words for each of the four outcomes is listed below in Table 24.

Table 24 – Word attention weights for the corruption dataset

Corruption Absolutions			**Corruption Condemnations**		
Position	**Word**	**Att. Weight**	**Position**	**Word**	**Att. Weight**
1	real	0.98	1	assunto	0.99
2	irregular	0.97	2	supra	0.99
3	nenhuma	0.96	3	autos	0.98
4	àquela	0.95	4	apresentou	0.98
5	ofereceu	0.94	5	decisão	0.98
6	reconheceu	0.94	6	cpp	0.98
7	apenas	0.94	7	decisão	0.97
8	pública	0.93	8	público	0.97
9	levado	0.93	9	regime	0.97
10	parcialmente	0.93	10	valdefran	0.97
11	memoriais	0.92	11	contou	0.96
12	resta	0.92	12	demonstrada	0.96
13	multa	0.92	13	ativa	0.96
14	localidade	0.91	14	exame	0.96
15	originário	0.91	15	ministério	0.96
16	inicial	0.90	16	começou	0.96
17	segue	0.89	17	quantia	0.96
18	oferecendo	0.89	18	propina	0.96
19	polícia	0.88	19	polícia	0.96
20	pretensão	0.88	20	peculato	0.96

3.2.11 Top Words for the Homicides Dataset

Just as done in the corruption dataset, the words also had their attention weights calculated for the homicides dataset. For the homicide dataset, the result in shown in Table 25.

One crucial factor that we noticed, while comparing the most important words of both datasets, is that words in the corruption dataset have a more significant weight. As we can see in Table 24, words could reach as high as 0.98 of attention weights values in the corruption dataset. Those values could either indicate that corruption texts make more frequent use of unique words, offering a deeper meaning to each one of them, or that the set of words used for absolutions are more different than the ones used for condemnations in corruption cases.

Table 25 – Word attention weights for the homicide dataset

Homicide Absolutions			**Homicide Condemnations**		
Position	**Word**	**Att. Weight**	**Position**	**Word**	**Att. Weight**
1	bo	0.52	1	bo	0.52
2	mogi	0.42	2	cristina	0.49
3	santos	0.41	3	horário	0.49
4	justiça	0.41	4	infração	0.47
5	sala	0.40	5	penal	0.46
6	competência	0.40	6	cf	0.46
7	volta	0.40	7	regime	0.45
8	origem	0.39	8	homicídio	0.45
9	infância	0.39	9	qualificado	0.44
10	social	0.38	10	disparos	0.44
11	júri	0.38	11	exposto	0.44
12	júri	0.38	12	provisório	0.44
13	júri	0.37	13	sentença	0.44
14	altura	0.36	14	juízo	0.44
15	permitido	0.36	15	golpes	0.43
16	júri	0.36	16	justiça	0.43
17	soubessem	0.36	17	acusação	0.43
18	ordinário	0.36	18	socos	0.43
19	central	0.36	19	análise	0.42
20	júri	0.36	20	lesões	0.42

We can see that, even though some words are repeated in both scenarios, others have a noteworthy significance towards absolving or condemning a defendant. Those words can be used as a key to predict the outcome of a legal document effectively.

4 Conclusions

As a first achievement, our work has been able to produce a labeled corpus of judicial cases, with examples of homicide and corruption subjects. When our research began, we found that no labeled corpus was available for the Brazilian judicial system, even though some studies on Law have been conducted in the last years. Since the intersection of AI and Law is a novelty in Brazilian science, we think that our corpus may help future researches to build new prediction strategies. Our corpus also has characteristics that we did not explore in this work. Still, it may be of great help to future analysts, such as the gender of the judge that analyzed the case, and whether the county of the matter is located in a capital or country city.

As a second achievement, we demonstrated that algorithms could predict the outcome of judicial cases, given the text that was written on their court decisions. Whether using non-neural networks, such as SVM and CART, or neural networks, such as LSTM, GRU, and HAN, we had results that exceeded 95% accuracy for most cases. Besides having a high accuracy rate for some algorithms, our work has also proven that other methods are not as effective, such as K-Neighbors and RNNs.

For the non-neural network models, as mentioned in Section 3.1, we have found that Regression Trees are the method that showed the highest accuracy to predict results in both data sets being analyzed, frequently outperforming the other methods studied. Support Vector Machines, as an example, showed a good performance in the homicides dataset but did not match the results Regression Trees showed in the corruption dataset. In both datasets, Regression Trees have always kept good predicting outcomes.

Also, as mentioned in Section 3.1, our results match other results found by other researches in the legal area in many different countries, such as the one conducted by Kastellec (2010), who obtained good outcomes by using Regression Trees in the American legal system. The thesis presented by the author, mentioning that Regression Trees have the capability of studying legal conceptions of Law, revealing patterns that other methods cannot emulate as effectively, can also be seen in our research. Kastellec (2010) also writes that classification trees could also work to increase understanding of legal rules and legal doctrine by capturing many aspects of the relationship between case facts and case outcomes, a statement that could open new research possibilities in Law. Other researches have also confirmed the efficacy of Regression Trees, such as Rios-Figueroa (2011), who used the method to analyze the concept of judicial independence and corruption among Supreme Courts in Latin America, Antonucci, Crocetta and D'Ovidio (2014), who adopted Regression Trees to measure the efficiency of Italian courts, and Kufandirimbwa and Kuranga (2012), who used the same algorithm to predict outcomes in Zimbabwe.

As previously written, those researches show that, even though legal systems are significantly different among distinct languages and countries, such as Brazil, the United States, Italy and Zimbabwe, they do have similar characteristics that can be effectively measured by the correct algorithms. In that way, we can see that legal texts might have intrinsic features that remain even when languages change.

As for the neural networks, described in Section 3.2, we have seen that the MLP and RNN showed average results. For the MLP, we infer that, since the network does not consider the recurrency of the words, and our corruption dataset was smaller than the homicides dataset, the algorithm did not manage to learn how to predict outcomes for corruptions effectively. For the RNN, even though recurrency is considered in the model, it does not contemplate many advancements shown in newer models. Both networks also showed a condemnation bias, for they were prone to predict a condemnation even for absolutions cases, but did not do the opposite.

The results shown for the LSTMs and GRUs effectively eliminate this condemnation bias, increasing the overall accuracy, proving that those last algorithms perform better than the two previous ones for our datasets. The accuracies shown by the GRUs are also slightly higher than the results shown by the Regression Trees, indicating that neural networks are in fact effective prediction methods. However, it should also be considered that Regression Trees are computationally faster than the GRUs. Therefore, there is not a single method that could be selected as the best choice for every case presented.

Nonetheless, Hierarchical Attention Networks showed the highest accuracy overall for the corruption dataset and the second-best for the homicides dataset. Our results keep pace with similar works that compare the effectiveness of HAN against other methods, such as the one made by Gao et al. (2017), who used HANs compared to Naive Bayes, Logistic Regression, Support Vector Machines, Random Forests, Extreme Gradient Boosting, RNNs, and Convolutional Neural Networks. Other researches have also found that HANs outperform traditional methods. For instance Ma et al. (2019), which compared HANs to Convolutional Neural Networks, SVMs, LSTMs, and DeClarE, and Tarnpradab, Liu and Hua (2018), who used HANs against SVM and Logistic Regression models. Since those papers did not include GRUs as one of the possible methods, we cannot also conclude that this method would also be a top pick among the algorithms chosen. Furthermore, HANs embrace the Attention method, which offers an interesting analysis on the word and sentence attention weights of each dataset.

The adoption of HANs in the legal field is a novel approach but has been positively adopted worldwide, in the works such as Chalkidis et al. (2019), who used HANs to classify the field of legal texts, and the previously mentioned Ma et al. (2019), who used the method to predict the outcome of claim verifications. Variations of the HAN model can be seen in Liu et al. (2019), who created a variation of the HAN model to determine the charges in criminal cases or types of disputes in civil cases according to the fact descriptions, and in Wenguan CHEN Yunwen (2019), who adopted an improved version of the algorithm for crime prediction, legal article recommendation, and sentence prediction from judicial documents. Our research is the first, to the best of our knowledge, to predict judicial outcomes in Brazilian Portuguese, with results that emulate the ones found in different languages the fields of the Law.

Moreover, our research shows that the best choices for our datasets, among all the methods compared, are Regression Trees, GRUs, and HANs. However, there is not a single choice between those three methods, since they carry particular advantages and flaws.

4.1 Contributions

Our work during the Master's has engendered two publications. During our research to identify the main topics of a given corpus, the first one was the conference article *Using Topic Modeling to Find Main Discussion Topics in Brazilian Political Websites*, annexed at Annex A, published at the Brazilian Symposium of Multimedia and Web Systems 2019 (Webmedia 2019), located in Rio de Janeiro, Brazil.

The second article is directly connected to this research, called *Predicting Judicial Outcomes in the Brazilian Legal System Using Textual Features*, annexed at Annex B, published at the Workshop on Digital Humanities and Natural Language Processing, co-located with International Conference on the Computational Processing of Portuguese 7(DHandNLP-PROPOR 2020), located in Evora, Portugal.

4.2 Future Work

As a first future work path, we intend to research why CART works better than other methods for judicial texts, as shown in Section 3.1.2. Languages have many different characteristics, especially languages with distinct family trees. Finding that CART works better than other methods for Portuguese, Italian, and English is a remarkable discovery, one that deserves more profound studies.

As a second possibility, we plan to increase our dataset of judicial texts, increasing both the number of cases gathered for homicide and corruption, as well as approaching other judicial subjects. Adding judicial matters that are very different than the ones that were adopted, such as Family Law or Arbitration, can considerably improve the prediction methods that were developed in this research.

As a final future work direction, we aspire to adopt other algorithms, to expand on the research possibilities. Since NLP and AI are growing rapidly, the rate of novelties to further ameliorate our methods is strikingly high. New algorithms and techniques can be employed to achieve even better results.

References

AIRES, J. P. et al. Norm conflict identification in contracts. *Artificial Intelligence and Law*, v. 25, n. 4, p. 397–428, 2017. ISSN 15728382.

ALARIE, B.; NIBLETT, A.; YOON, A. How Artificial Intelligence Will Affect the Practice of Law. In: *Artificial Intelligence, Technology and the Future of Law*. [s.n.], 2017. p. 1–16. Disponível em: <https://papers.ssrn.com/sol3/papers.cfm?abstract{_}id=3066>.

ALETRAS, N. et al. Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. *PeerJ Computer Science*, v. 2, n. Ml, p. e93, 2016. ISSN 2376-5992. Disponível em: <https://peerj.com/articles/cs-93>.

ALSCHNER, W.; SKOUGAREVSKIY, D. Towards an automated production of legal texts using recurrent neural networks. *International Conference on Artificial Intelligence and Law*, v. 5, n. July, 2017.

ANTONUCCI, L.; CROCETTA, C.; D'OVIDIO, F. D. Evaluation of Italian Judicial System. *Procedia Economics and Finance*, Elsevier B.V., v. 17, n. September 2015, p. 121–130, 2014. ISSN 22125671. Disponível em: <http://dx.doi.org/10.1016/S2212-5671(14)00886-7>.

ARAUJO, D. A. de; RIGO, S. J.; BARBOSA, J. L. V. Ontology-based information extraction for juridical events with case studies in Brazilian legal realm. *Artificial Intelligence and Law*, v. 25, n. 4, p. 379–396, 2017. ISSN 15728382.

ASHLEY, K. D.; BRÜNINGHAUS, S. Automatically classifying case texts and predicting outcomes. *Artificial Intelligence and Law*, v. 17, n. 2, p. 125–165, 2009. ISSN 09248463.

BALAKRISHNAMA, S.; GANAPATHIRAJU, A. Linear discriminant analysis-a brief tutorial. In: *Institute for Signal and information Processing*. [S.l.: s.n.], 1998. v. 18, n. 1998, p. 1–8.

BARRAUD, B. Un algorithme capable de prédire les décisions des juges: vers une robotisation de la justice? *Les Cahiers de la justice*, p. 121–139, 2017.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2009. ISBN 0596516495.

BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738.

BOELLA, G. et al. Eunomos, a legal document and knowledge management system for the Web to provide relevant, reliable and up-to-date information on the law. *Artificial Intelligence and Law*, Springer Netherlands, v. 24, n. 3, p. 245–283, 2016. ISSN 15728382.

BRANTING, L. K. et al. Inducing Predictive Models for Decision Support in Administrative Adjudication. 2017.

BROWN, P. F. et al. Class-based n-gram models of natural language. *Computational linguistics*, MIT Press, v. 18, n. 4, p. 467–479, 1992.

CHALKIDIS, I. et al. Extreme multi-label legal text classification: a case study in eu legislation. *arXiv preprint arXiv:1905.10892*, 2019.

CHANTAR, H. K.; CORNE, D. W. Feature subset selection for Arabic document categorization using BPSO-KNN. *Proceedings of the 2011 3rd World Congress on Nature and Biologically Inspired Computing, NaBIC 2011*, p. 546–551, 2011.

CHUNG, J. et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. Disponível em: <http://arxiv.org/abs/1412.3555>.

CHUNG, J. et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

CONRAD, J. G.; AL-KOFAHI, K. Scenario Analytics Analyzing Jury Verdicts to Evaluate Legal Case Outcomes. *Proceedings of the 16th international conference on Artificial intelligence and law*, v. 10, 2017.

DESMET, B.; HOSTE, V. Recognising suicidal messages in Dutch social media. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, p. 830–835, 2014.

DO, P.-K. et al. Legal Question Answering using Ranking SVM and Deep Convolutional Neural Network. *Tenth International Workshop on Juris-informatics (JURISIN 2016) associated with JSAI International Symposia on AI 2016 (IsAI-2016)*, 2017. Disponível em: <http://arxiv.org/abs/1703.05320>.

El Jelali, S.; FERSINI, E.; MESSINA, E. Legal retrieval as support to eMediation: matching disputant's case and court decisions. *Artificial Intelligence and Law*, v. 23, n. 1, p. 1–22, 2015. ISSN 15728382.

FAIRCLOUGH, N. *Analysing discourse: Textual analysis for social research.* [S.l.]: Psychology Press, 2003.

FORNACIARI, T.; POESIO, M. Automatic deception detection in Italian court cases. *Artificial Intelligence and Law*, v. 21, n. 3, p. 303–340, 2013. ISSN 09248463.

FRANCESCONI, E.; PERUGINELLI, G. Integrated access to legal literature through automated semantic classification. *Artificial Intelligence and Law*, v. 17, n. 1, p. 31–49, 2009. ISSN 09248463.

GAO, S. et al. Hierarchical attention networks for information extraction from cancer pathology reports. *Journal of the American Medical Informatics Association*, v. 25, n. 3, p. 321–330, 11 2017. ISSN 1527-974X. Disponível em: <https://doi.org/10.1093/jamia/ocx131>.

GAO, S. et al. Hierarchical attention networks for information extraction from cancer pathology reports. *Journal of the American Medical Informatics Association*, Oxford University Press, v. 25, n. 3, p. 321–330, 2018.

GOKHALE, R.; FASLI, M. Deploying A Co-training Algorithm to Classify Human-Rights Abuses. In: *2017 International Conference on the Frontiers and Advances in Data Science (FADS)*. [S.l.: s.n.], 2017. p. 108–113. ISBN 9781538631485.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

HARCOURT, P.; HARCOURT, P. Feature Selection And Vectorization In Legal Case Documents Using Chi-Square Statistical Analysis And Naïve Bayes Approaches Feature Selection And Vectorization In Legal Case Documents Using Chi-Square Statistical Analysis And Naïve Bayes Approaches. *IOSR Journal of Computer Engineering (IOSR-JCE)*, v. 17, n. APRIL, p. 42–50, 2015.

HARTMANN, N. et al. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. 08 2017.

HUANG, L. et al. Hierarchical attention network in stock prediction. In: SPRINGER. *China Conference on Information Retrieval*. [S.l.], 2020. p. 124–136.

HYMAN, H. et al. A process model for information retrieval context learning and knowledge discovery. *Artificial Intelligence and Law*, v. 23, n. 2, p. 103–132, 2015. ISSN 15728382.

Kanakaraj, M.; Guddeti, R. M. R. Performance analysis of ensemble methods on twitter sentiment analysis using nlp techniques. In: *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*. [S.l.: s.n.], 2015. p. 169–170.

KASTELLEC, J. P. The Statistical Analysis of Judicial Decisions and Legal Rules with Classification Trees. *Journal of Empirical Legal Studies*, v. 7, n. 2, p. 202–230, 2010. ISSN 1740-1461.

KATZ, D. M.; II, M. J. B.; BLACKMAN, J. A general approach for predicting the behavior of the supreme court of the united states. *PloS one*, Public Library of Science, v. 12, n. 4, p. e0174698, 2017.

KHOSROW-POUR, M. *Encyclopedia of Information Science and Technology*. 2. ed. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2008. ISBN 1605660264.

KIM, D.-K. et al. Multi-label classification of historical documents by using hierarchical attention networks. *Journal of the Korean Physical Society*, Springer, v. 76, n. 5, p. 368–377, 2020.

KIM, S. M. et al. Demographic Inference on Twitter using Recursive Neural Networks. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, p. 471–477, 2017. Disponível em: <http://aclweb.org/anthology/P17-2075>.

KINGSTON, J. Using artificial intelligence to support compliance with the general data protection regulation. *Artificial Intelligence and Law*, Springer Netherlands, v. 25, n. 4, p. 429–443, 2017. ISSN 15728382.

KISS, T.; STRUNK, J. Unsupervised multilingual sentence boundary detection. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 32, n. 4, p. 485–525, dez. 2006. ISSN 0891-2017. Disponível em: <https://doi.org/10.1162/coli.2006.32.4.485>.

KRESTEL, R.; FANKHAUSER, P.; NEJDL, W. Latent dirichlet allocation for tag recommendation. *Proceedings of the third ACM conference on Recommender systems - RecSys '09*, p. 61, 2009. ISSN 00283932. Disponível em: <http://portal.acm.org/citation.cfm?doid=1639714.1639726>.

KUFANDIRIMBWA, O.; KURANGA, C. Towards Judicial Data Mining : Arguing for Adoption in the Judicial System. v. 1, n. 2, p. 15–21, 2012.

LE, T. T. N. et al. Extracting indices from Japanese legal documents. *Artificial Intelligence and Law*, Springer Netherlands, v. 23, n. 4, p. 315–344, 2015. ISSN 15728382.

LI, X. et al. Target-specific convolutional bi-directional LSTM neural network for political ideology analysis. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [S.l.: s.n.], 2017. v. 10367 LNCS, p. 64–72. ISBN 9783319635637. ISSN 16113349.

LIANG, T. et al. Multi-view factorization machines for mobile app recommendation based on hierarchical attention. *Knowledge-Based Systems*, Elsevier, v. 187, p. 104821, 2020.

LIU, Z.; CHEN, H. A Predictive Performance Comparison of Machine Learning Models for Judicial Cases. In: *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.: s.n.], 2017. ISBN 9781538627266.

LIU, Z. et al. Legal cause prediction with inner descriptions and outer hierarchies. In: SUN, M. et al. (Ed.). *Chinese Computational Linguistics*. Cham: Springer International Publishing, 2019. p. 573–586. ISBN 978-3-030-32381-3.

LOH, W.-Y. Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, v. 1, n. 1, p. 14–23, 2011. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.8>.

LUO, B. et al. Learning to Predict Charges for Criminal Cases with Legal Basis. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. [s.n.], 2017. p. 2727–2736. Disponível em: <http://arxiv.org/abs/1707.09168>.

MA, J. et al. Sentence-level evidence embedding for claim verification with hierarchical attention networks. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, 2019. p. 2561–2571. Disponível em: <https://www.aclweb.org/anthology/P19-1244>.

MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008. ISBN 0521865719, 9780521865715.

MANNING, C. D.; SCHüTZE, H. *Foundations of Statistical Natural Language Processing.* Cambridge, MA, USA: MIT Press, 1999. ISBN 0-262-13360-1.

MAVERICK, G. V. Computational Analysis of Present-Day American English. Henry Kučera , W. Nelson Francis. *International Journal of American Linguistics*, v. 35, n. 1, p. 71–75, 1969. Disponível em: <https://doi.org/10.1086/465045>.

MCSHANE, B. B. et al. Predicting Securities Fraud Settlements and Amounts: A Hierarchical Bayesian Model of Federal Securities Class Action Lawsuits. *Journal of Empirical Legal Studies*, v. 9, n. 3, p. 482–510, 2012. ISSN 17401453.

MOCHALES, R.; MOENS, M. F. Argumentation mining. *Artificial Intelligence and Law*, v. 19, n. 1, p. 1–22, 2011. ISSN 09248463.

MOENS, M. F. Innovative techniques for legal text retrieval. *Artificial Intelligence and Law*, v. 9, n. 1, p. 29–57, 2001. ISSN 09248463.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning.* [S.l.]: The MIT Press, 2012. ISBN 026201825X.

MOUSAVI, S.; AFGHAH, F.; ACHARYA, U. R. Han-ecg: An interpretable atrial fibrillation detection model using hierarchical attention networks. *arXiv preprint arXiv:2002.05262*, 2020.

PAVLINEK, M.; PODGORELEC, V. Text classification method based on self-training and LDA topic models. *Expert Systems with Applications*, Elsevier Ltd, v. 80, p. 83–93, 2017. ISSN 09574174.

PELLE, R.; ALCÂNTARA, C.; MOREIRA, V. P. A Classifier Ensemble for Offensive Text Detection. p. 237–243, 2018.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Empirical Methods in Natural Language Processing (EMNLP).* [s.n.], 2014. p. 1532–1543. Disponível em: <http://www.aclweb.org/anthology/D14-1162>.

PHUOC, N. et al. Experimenting Word Embeddings in Assisting Legal Review. *ICAIL 2017 The 16th International Conference on Artificial Intelligence and Law, King's College London, London, UK, 12 - 16 June 2017*, n. October, 2017.

RAO, A.; SPASOJEVIC, N. Actionable and Political Text Classification using Word Embeddings and LSTM. 2016.

REMMITS, Y. *Finding the Topics of Case Law : Latent Dirichlet Allocation on Supreme Court Decisions.* 1–31 p. Tese (Doutorado), 2017.

RIOS-FIGUEROA, J. Judicial Independence and Corruption: An Analysis of Latin America. *SSRN Electronic Journal*, n. 212, 2011.

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach.* 2. ed. [S.l.]: Pearson Education, 2003. ISBN 0137903952.

SA, C.; SANTOS, R.; MOURA, R. An approach for defining the author reputation of comments on products. In: . [S.l.: s.n.], 2017. p. 326–331. ISBN 978-3-319-59568-9.

SANNIER, N. et al. An automated framework for detection and resolution of cross references in legal texts. *Requirements Engineering*, Springer London, v. 22, n. 2, p. 215–237, 2017. ISSN 1432010X.

SULEA, O.-M. et al. Exploring the Use of Text Classification in the Legal Domain. In: *Proceedings of the 2nd Workshop on Automated Semantic Analysis of Information in Legal Texts (ASAIL)*. [s.n.], 2017. Disponível em: <http://arxiv.org/abs/1710.09306>.

SULEA, O.-M. et al. Predicting the Law Area and Decisions of French Supreme Court Cases. In: *Recent Advances in Natural Language Processing*. [s.n.], 2017. Disponível em: <http://arxiv.org/abs/1708.01681>.

SURDEN, H. Machine Learning and Law. *Washington Law Review*, v. 646, p. 87–115, 2014. ISSN 00430617. Disponível em: <http://papers.ssrn.com/sol3/Papers.cfm?abstract{_}id=2417>.

TALLEY, E. L.; O'KANE, D. The measure of a mac: A quasi-experimental protocol for tokenizing force majeure clauses in m&a agreements. 2011.

TARNPRADAB, S.; LIU, F.; HUA, K. A. Toward extractive summarization of online forum discussions via hierarchical attention networks. *arXiv preprint arXiv:1805.10390*, 2018.

TRAN, O. T. et al. Automated reference resolution in legal texts. *Artificial Intelligence and Law*, v. 22, n. 1, p. 29–60, 2014. ISSN 09248463.

TURIAN, J.; RATINOV, L.; BENGIO, Y. Word representations: a simple and general method for semi-supervised learning. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 48th annual meeting of the association for computational linguistics*. [S.l.], 2010. p. 384–394.

WENGUAN CHEN YUNWEN, C. H. Z. Y. Y. H. W. Judicial document intellectual processing using hybrid deep neural networks. *Journal of Tsinghua University(Science and Technology)*, Journal of Tsinghua University(Science and Technology), v. 59, n. 7, p. 505, 2019. Disponível em: <http://jst.tsinghuajournals.com/EN/abstract/article_153411.shtml>.

XIE, J.; LIU, X.; Dajun Zeng, D. Mining e-cigarette adverse events in social media using Bi-LSTM recurrent neural network with word embedding representation. *Journal of the American Medical Informatics Association*, v. 0, n. 0, p. 1–9, 2017. ISSN 1067-5027. Disponível em: <https://academic.oup.com/jamia/article-lookup/doi/10.1093/jamia/ocx045>.

YANG, Z. et al. Hierarchical attention networks for document classification. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, 2016. p. 1480–1489. Disponível em: <https://www.aclweb.org/anthology/N16-1174>.

ZENG, Y. et al. A knowledge representation model for the intelligent retrieval of legal cases. *International Journal of Law and Information Technology*, v. 15, n. 3, p. 299–319, 2007. ISSN 09670769.

ZHANG, Z.; ROBINSON, D.; TEPPER, J. Detecting hate speech on twitter using a convolution-gru based deep neural network. In: SPRINGER. *European semantic web conference*. [S.l.], 2018. p. 745–760.

ZIPF, G. K. *Selected Studies of the Principle of Relative Frequency in Language*. Cambridge (MA): Harvard University Press, 2013. ISBN 978-0-674-43492-9. Disponível em: <https://hup.degruyter.com/view/title/321953>.

Annex

A

Using Topic Modeling to Find Main Discussion Subjects in Brazilian Political Websites

Using topic modeling to find main discussion topics in brazilian political websites

Vithor Gomes Bertalan
vbertalan@usp.br
PPG-CA, DCM, FFCLRP, University São Paulo (USP)
Ribeirão Preto, SP

Evandro Eduardo Seron Ruiz
evandro@usp.br
Department of Computing and Mathematics, FFCLRP,
University of São Paulo (USP)
Ribeirão Preto, SP

ABSTRACT

Knowing the main discussion topics debated by the general public is a valuable asset to politicians and professionals involved with politics. Lately, alternative media websites became popular venues in which political ideas are debated without the influence of mainstream media. In this article, we propose the construction of a topic modeling framework, using LSI, LDA, and HDP, to identify main discussion issues in political websites. Experiments show that these models presented results similar to state of the art, offering a viable solution to track political discourse in left-wing and right-wing websites.

CCS CONCEPTS

• **Applied computing** → **Law, social and behavioral sciences**; • **Information systems** → *Document topic models.*

KEYWORDS

latent semantic indexing, hierarchial dirichlet process, latent dirichlet allocation, topic modeling, topic coherence, political texts

ACM Reference Format:
Vithor Gomes Bertalan and Evandro Eduardo Seron Ruiz. 2019. Using topic modeling to find main discussion topics in brazilian political websites. In *Brazilian Symposium on Multimedia and the Web (WebMedia '19), October 29-November 1, 2019, Rio de Janeiro , Brazil.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3323503.3360644

1 INTRODUCTION

Over the last couple of years, the online political environment throughout the world has been in a crescendo. Both, professional politicians and ordinary citizens have taken social network websites as a new mean to express their ideas and to support their political ideas. In all sides of the political spectrum, people have begun not only to consume political news and texts but also to produce their own material. The main consequence of this new media model is the ascending decentralization of content production. Mainstream media networks are no longer the only institutions with popular credibility. Alternative websites have become increasingly popular over the last few years, offering interpretations of facts that quite often contradict the interpretations given by the more traditional means.

The research question addressed in this paper is: *How to develop a model that precisely identifies the main topics being discussed by alternative political media in Brazil?* To answer this question we decided to adopt Natural Language Processing (NLP) techniques, gathering data from three left-wing and three right-wing alternative media websites in Brazil, to find what both sides of the political spectra are revealing.

As this paper shows in Section 3, NLP techniques to study political topics have a lengthy background in state of the art. However, on a bibliographical review, no researches dealing with the Brazilian alternative political media have been found. As Brazil is one of the largest democracies in the world, understanding the needs and subjects discussed by voters is of utmost importance not only for politicians and professionals that work with politics but to the ordinary citizens as well.

The approach taken in this article is also novel, using a modern statistical method, the statistic coherence, to be discussed in Section 4 and 5, to analyze political texts. Coherence has been utilized to evaluate topic modeling of political texts, but no research to this date has been done in Brazilian Portuguese using this methodology.

As to identify the most efficient topic modeling techniques, we applied several different models, such as Latent Semantic Indexing, Latent Dirichlet Allocation, and Hierarchical Dirichlet Process. The results were also classified by their statistical coherence, and we also found topics which more precisely represent what the texts were referring to.

2 THEORETICAL FRAMEWORK

The Latent Dirichlet Allocation (LDA) statistical model was proposed by David Blei [5] as an improved topic model, defining a Dirichlet probabilistic generative process for document-topic distribution. For each document, we built a multinomial topic distribution, and we also chose a latent aspect, controlled by a Dirichlet prior variable α. Afterward, given the previously selected latent aspect, a new word was selected according to its proper multinomial distribution, controlled by another Dirichlet prior variable β. The central concept of LDA is that each document in a data set is composed of many latent topics, and each resulting topic is composed of connecting words.

The Latent Semantic Indexing (LSI) statistical model was proposed by Deerwester *et al.* [25]. LSI is an automatic retrieval and indexing model used to identify higher-order structures that associate terms with documents. Within this association, a linear

WebMedia '19, October 29-November 1, 2019, Rio de Janeiro , Brazil

ACM ISBN 978-1-4503-6763-9/19/10…$15.00
https://doi.org/10.1145/3323503.3360644

algebra technique called Singular Value Decomposition is used to identify statistical patterns between words and concepts in a text. With this technique, LSI tries to capture the many-to-many mapping between terms and concepts, outranking conventional vector-based models [20]. Therefore, while LDA is a generative probabilistic model based on the Dirichlet distribution, LSI behaves as an indexing method of a document-term matrix, being faster to train, but traditionally offering a lower accuracy [28].

The Hierarchical Dirichlet Process (HDP) was proposed by Whye and coworkers [28], and it provides a non-parametric topic model where texts are viewed as groups of observed words, topics are distributions over terms, and each document exhibits its topics with different proportions. In this manner, the Dirichlet process provides a non-parametric prior distribution for the number of mixture components within each group. Unlike the previous methods, HDP infers the number of topics from the data.

3 RELATED WORK

Many different types of research have used Natural Language Processing techniques to study political data over the last years. A common approach is to identify the main discussion ideas on the web. One of the leading research trends is focused on finding political ideas on Twitter [1, 29]. Twitter[1] offers an API that enables the download of multiple tweets at once, making it a favorite microblog medium in NLP research.

Over the years, many different authors have sought to understand the underlying differences between different political spectra. On an international level, Imbeau, Petry and Lamari [15] studied the differences between left-wing and right-wing government policies. Kitschelt and Hellemans [18] studied the evolution of the concept of 'left' and 'right' over the decades. Also, Graham, Nosek, and Haidt [13] studied the exaggerations committed in the construction of political stereotypes of liberal and conservative ideas.

Another popular goal is to identify political leanings. In Conover, Goncalves, Ratkiewicz, Flammini, and Menczer [6], the authors use network clustering algorithms to identify liberal or conservative users based on their tweets. In Kim and Lee [17], the authors analyze user behaviors to find left-wing and right-wing leanings, but considering their retweet patterns alongside with their tweet texts.

Researches have also considered political blogs as a valuable source of information, like in Jiang and Argamon [16]. In Hassanali and Hatzivassiloglou [14], the authors attempt to categorize political blogs with tags generated by named entity recognition. In Dehghani, Azarbonyad, Marx, and Kamps [8], the authors make an effort to index new political texts based on their vocabulary and ontology. Most of the research conducted captures data in English, like the works of [3, 11]. However, there are papers in other languages, such as Indonesian in Alfina, Sigmawaty, Nurhidayati, and Hidayanto [1], Korean in Kim and Lee [17], Portuguese in Amorim, Alves, Oliveira, and Baptista [7] and Spanish in Pla and Hurtado [21].

Using more massive data sets, some papers began to study political blogs and forums. The works of [16] use Support Vector Machines (SVM) to categorize blog posts between liberal and conservative leaning. Hassanali and Hatzivassiloglou [?] propose a similar approach by combining SVM with named entity recognition. In Demartini and Siersdorfer [9], sentiment analysis techniques are adopted to extract political trends from blog posts. In Godbole and Srinivasaiah [12], network analysis is used to score relevant entities positively or negatively. In Durant and Smith [10], a Naïve Bayes classifier is adopted to predict sentiments of blog posts. Feature reduction is the technique adopted by Evrim and Awwal [11] to classify political affiliations in blog posts. As mentioned in Section 1, to this date, to the best of our knowledge, no work has been done by analyzing alternative political sites in Brazil, in any possible NLP technique, like sentiment analysis or topic modeling.

Table 1: Quantity of texts crawled, by website.

Website	# of Posts Collected
Brasil247	11,581
Diário do Centro do Mundo	10,717
Pragmatismo Político	1,028
Instituto Liberal	4,933
Reaçonaria	3,091
Senso Incomum	664

4 METHODOLOGY

Firstly, we needed to extract data from political websites. A Python text crawler was used to extract data from mainstream media websites. Websites considered to be part of the Brazilian printing press were not considered for this research. We searched for popular Brazilian websites representative of left-wing and right-wing ideas. The left-wing websites chosen were *Brasil247*[2], *Pragmatismo Politico*[3] and *Diario do Centro do Mundo*[4]. The right-wing websites chosen were *Instituto Liberal*[5], *Reaconaria*[6] and *Senso Incomum*[7]. Those websites were chosen by their Alexa[8] rankings of website traffic, making them the most accessed alternative politics websites to each side of the political spectrum by November 8h, 2017. We collected all the data used in this research on this same date. The total amount of texts collected is shown in Table 1.

After pre-processing the text, the two data sets (left-wing and right-wing websites) were copied into two versions: a stemmed version and a non-stemmed version. This action was defined to test stemming efficacy towards the data sets used. Afterward, we extracted the vocabulary for each text, and we also built a bag-of-words (BOW) model for each one, a requirement to run the statistical models mentioned. Lastly, we used Python to run LSI, LDA (with 20 topics each), and HDP algorithms. The Gensim Library [22] was the framework used to run and tune the algorithms. All algorithms run with their standard Gensim parameters.

[1] https://twitter.com

[2] https://www.brasil247.com/

[3] https://www.pragmatismopolitico.com.br/

[4] https://www.diariodocentrodomundo.com.br/

[5] https://www.institutoliberal.org.br/

[6] http://www.reaconaria.org

[7] http://www.sensoincomum.org/

[8] https://www.alexa.com/

5 RESULTS

To compare the results, we used the statistical concept of topic coherence. As cited by Newman and Stevens [19, 26], topic coherence scores a single topic by measuring the degree of semantic similarity between high scoring words in the topic. Statistical topic coherence is a novel approach that has begun to be applied in topic modeling, as seen in [4, 23]. Topic coherence was computed by calculating the sum of pairwise distributional similarity scores over the set of topic words V.

$$\text{coherence}(V) = \sum_{v_i, v_j \epsilon V} \text{score}(v_i, v_j, \varepsilon) \qquad (1)$$

There are many varieties of coherence measures. In this work, we used the CV coherence measure, available in the Gensim Python package. As mentioned by Roder and Syed [24, 27], the CV coherence measure works by indirect cosine measure and the Boolean sliding windows, while segmenting the data into word pairs, calculates how strongly those word pairs support one another. Those support levels are then inserted into a comparable overall coherence score.

After running the algorithms, the topic coherence level for each of the scenarios is shown in Table 2. As the figures show, the models have reached coherence levels up to 0.5, sometimes reaching levels between 0.5 and 0.6. In an extensive research by Syed and Spruit [27], with different subsets, the highest coherence level reached by these authors was 0.594, which places some of the models developed among the best-performing ones.

Table 2: Topic coherence scores.

Political Leaning	LSI	HDP	LDA
Right-wing	0.502	0.209	0.490
Right-wing with stemming	0.416	0.207	0.408
Left-wing	0.380	0.209	0.448
Left-wing with stemming	0.327	0.210	0.429

One notable discovery was that HDP was outperformed in every scenario. In both left-wing and right-wing websites. With or without stemming included, HDP showed the lowest coherence levels, usually presenting results between 0.2 and 0.3. As shown by Röder *et al.* [24], different data sets perform better with different algorithms, and not one topic modeling technique will always perform better, depending on the variables included (Boolean sliding windows, number of passes, number of topic, etc.). Another discovery was that LSI performs better to model right-wing topics, and LDA performs better to model left-wing topics. This was a surprising discovery, since usually the LSI model is described to be faster than LDA, but not as effective [20, 25]. Lastly, we found the stemming is not always the best choice, and its application has to be studied case by case. In all the four models, stemming has caused a slight decrease in coherence values. As the other variables considered, stemming should be evaluated before it is applied in the models being studied.

6 DISCUSSING RIGHT-WING RESULTS

The results, here presented in English, in a free translation form, were different for non-stemmed and stemmed right-wing data sets. For the non-stemmed, as we can see, the algorithms show a higher coherence score in discovering words that are related to places. Words such as *house*, *square*, *prefecture*, *river*, *downtown*, and *avenue* were all selected as a single topic, with a high coherence value. Another group is related to recent presidents of Brazil. As of November 2017, the most recent presidents were *Luis Inacio Lula da Silva* (represented by the word *Lula*) and *Dilma Rousseff* (represented by the word *Dilma*). The last topic chosen represents words related to economy, like *market*, *economy*, freedom, and *society*.

For the stemmed data set, the algorithm showed again a strong relationship with words related to economic subjects, like 'work', 'company', 'enterprise', and 'government'. As a topic with medium coherence, we can see words related to public goods and rights, like 'public', 'freedom', 'country', 'service', and 'work'. Also, for the right-wing selected topics, we can see a topic that includes words related to the Brazilian contractor company Odebrecht. Words like 'Lula', 'people', 'economy', 'politics', and the word 'Odebrecht' itself, indicate that the company is quite often cited in relationship with Brazilian politics.

As a general discussion, we can see that the Brazilian right-wing alternative media is concerned about the economy, citing quite often economic freedom regarding companies and enterprises, as well as public goods and places, discussing work and economy markets with high intensity.

7 DISCUSSING LEFT-WING RESULTS

The results were also different for non-stemmed and stemmed left-wing data sets. For the non-stemmed, as we can see, the algorithm show a high precision rate towards words related to the Operation Car Wash [9], like the former Brazilian President *Lula*, the federal judge Sergio *Moro*, and the words *justice*, *car*, *wash*, *judiciary*, *federal*, *process*, *prison*, *judges* and *corruption*. The second selected topic is composed by words related to the 2014 World Cup, hosted in Brazil. A few words that compose this topic are: *soccer*, *cup*, *brazil*, *world*, *team*, *game*, *player*, *players*, *supporter*, *final*, *ball* and *field*. The third selected topic in this data set relates words related to Catholic subjects, like *pope*, *francis*, *Vatican* and *catholic*. Theere are also some aspect words that are still taboo to some religions, like *abortion*, *woman*, *rock* and *music*.

For the stemmed data set, the algorithm began by showing a strong correlation between words related to economic development, like *brazil*, *government*, *enterprise*, *economy*, *work*, *public*, *finance*, *bank*, *investment* and *growth*. A second popular topic was composed by terms related to the 2016 impeachment process in Brazil, which deposed the former president Dilma Rousseff. Words composing this topic are: *impeachment*, *dilma*, *president*, and *coup*. The name of a politician co-responsible for the impeachment process, Congressman Eduardo Cunha, also appears in two words: *eduardo* and *cunha*, as well as his political party, *PMDB*. Ending the topics selected, the last one is mainly composed by words relating to the Brazilian mainstream media: *Globo* and *Folha*. Other words like *newspaper*, *public* and *media* also appear in this topic.

[9] https://en.wikipedia.org/wiki/Operation_Car_Wash

As a general discussion, we can see that the left-wing alternative media is actively concerned with Operation Car Wash, the operation that led to the arrest of former president Luiz Inácio Lula da Silva, a left-wing historical figure in Brazil. The impeachment process is also seen by left-wing media as a coup, as we could see in the topic mentioned in the last paragraph. As Brazil was selected as the host country, as well as hosted the 2014 World Cup under the Workers' Party, this is also a popular subject of discussion. Other economic and social topics like religious dogmas or economic development issues are also frequent in Brazilian left-wing websites.

As an overall conclusion, LDA behaved better with left-wing websites, and LSI showed better results for right-wing texts. We hypothesize that this is due to a higher degree of homogeneity of discourse in texts from the left, since LSI offers better results with more analogous texts [2].

8 LIMITATIONS AND FUTURE RESEARCH

The first limitation of this research is the small number of websites gathered. As a future possibility, this work can be expanded by selecting new political websites to increase the number of sources monitored. The second limitation is the presence of alternative websites only. Even though the scope of this research was to gather alternative media exclusively, the comparison of alternative media with mainstream media websites would allow us to see see the differences of approach between different press corporations.

Tthis research can be used to track the political discourse on both sides of the political spectrum. Knowing the main topics of discussion in both left-wing and right-wing voters can be a valuable asset to predict how the public is going to react to critical events. Future possibilities of improvement to this research include the creation of new statistical models and the utilization of new NLP techniques to refine the data sets collected. Experiments to increase the effectiveness of the results between left-wing and right-wing websites has the potential of offering new insights on how governments, enterprises and the general public could react to the recent political turmoil over the world.

REFERENCES

[1] Ika Alfina, Dinda Sigmawaty, Fitriasari Nurhidayati, and Achmad Nizar Hidayanto. 2017. Utilizing Hashtags for Sentiment Analysis of Tweets in The Political Domain. *Proceedings of the 9th International Conference on Machine Learning and Computing* (2017), 43–47. https://doi.org/10.1145/3055635.3056631

[2] Leticia H Anaya. 2011. *Comparing Latent Dirichlet Allocation and Latent Semantic Analysis as Classifiers.* ERIC.

[3] Akshat Bakliwal, Jennifer Foster, Jennifer van der Puil, Ron O'Brien, Lamia Tounsi, and Mark Hughes. 2013. Sentiment Analysis of Political Tweets: Towards an Accurate Classifier. *Proceedings of the Workshop on Language Analysis in Social Media* Lasm (2013), 49–58. http://www.aclweb.org/anthology/W13-1106

[4] Paulo Bicalho, Marcelo Pita, Gabriel Pedrosa, Anisio Lacerda, and Gisele L. Pappa. 2017. A general framework to expand short text for topic modeling. *Information Sciences* 393 (2017), 66–81. https://doi.org/10.1016/j.ins.2017.02.007

[5] David M Blei, Andrew Young, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), 993–1022. https://doi.org/10.1162/jmlr.2003.3.4-5.993 arXiv:1111.6189v1

[6] Michael D. Conover, Bruno Goncalves, Jacob Ratkiewicz, Alessandro Flammini, and Filippo Menczer. 2011. Predicting the Political Allignment of Twitter Users. *Proceedings of IEEE International Conference on Privacy, Security, Risk, and Trust, and IEEE International Conference on Social Computing* (2011). https://doi.org/10.1109/PASSAT/SocialCom.2011.34

[7] Brunna de Sousa Pereira Amorim, André Luiz Firmino Alves, Maxwell Guimarães de Oliveira, and Cláudio de Souza Baptista. 2018. Using Supervised Classification to Detect Political Tweets with Political Content. (2018), 245–252. https://doi.org/10.1145/3243082.3243113

[8] Mostafa Dehghani, Hosein Azarbonyad, Maarten Marx, and Jaap Kamps. 2015. Sources of Evidence for Automatic Indexing of Political Texts. *Advances in Information Retrieval SE* 9022 (2015), 568–573. https://doi.org/10.1007/978-3-319-16354-3_63

[9] Gianluca Demartini and Stefan Siersdorfer. 2011. Analyzing Political Trends in the Blogosphere. *Association for the Advancement of Artificial Intelligence* (2011), 466–469. http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewPDFInterstitial/2838/3244

[10] Kathleen T. Durant and Michael D. Smith. 2007. Predicting the political sentiment of web log posts using supervised machine learning techniques coupled with feature selection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4811 LNAI (2007), 187–206. https://doi.org/10.1007/978-3-540-77485-3_11

[11] Vesile Evrim and Aliyu Awwal. 2015. Classification of Political Affiliations by Reduced Number of Features. *International Journal of Social, Behavioral, Educational, Economic and Management Engineering* 9, 6 (2015), 1863–1870.

[12] N Godbole and M Srinivasaiah. 2007. Large-scale sentiment analysis for news and blogs. *Conference on Weblogs and Social Media (ICWSM 2007)* (2007), 219–222. https://doi.org/10.1177/014617907037004050l arXiv:cond-mat/0112110

[13] Jesse Graham, Brian A Nosek, and Jonathan Haidt. 2012. The Moral Stereotypes of Liberals and Conservatives: Exaggeration of Differences across the Political Spectrum. *PLOS ONE* 7, 12 (2012), 1–13. https://doi.org/10.1371/journal.pone.0050092

[14] Khairun-nisa Hassanali and Vasileios Hatzivassiloglou. 2010. Automatic detection of tags for political blogs. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media.* 21–22.

[15] Louis M Imbeau, François Pétry, and Moktar Lamari. 2001. Left–right party ideology and government policies: A meta–analysis. *European Journal of Political Research* 40, 1 (2001), 1–29. https://doi.org/10.1111/1475-6765.00587

[16] Maojin Jiang and S Argamon. 2008. Finding political blogs and their political leanings. *Text Mining 2008, Workshop at the SIAM* (2008). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.216.2857{&}rep=rep1{&}type=pdf

[17] Jun-Gil KIM and Kyung-Soon LEE. 2014. Predicting Political Orientation of News Articles Based on User Behavior Analysis in Social Network. *IEICE Transactions on Information and Systems* E97.D, 4 (2014), 685–693. https://doi.org/10.1587/transinf.E97.D.685

[18] Herbert Kitschelt and Staf Hellemans. 1990. The Left-Right Semantics and the New Politics Cleavage. *Comparative Political Studies* 23, 2 (1990), 210–238. https://doi.org/10.1177/0010414090023002003

[19] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. 2010. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics, 100–108.

[20] CH Papadimitriou, H Tamaki, and Santosh Vempala. 1998. Latent semantic indexing: A probabilistic analysis. *J. of Computer and System Sciences* 61, 2 (1998), 217–235. http://dl.acm.org/citation.cfm?id=275505

[21] Ferran Pla and Lluís-F. Hurtado. 2014. Political Tendency Identification in Twitter using Sentiment Analysis Techniques. *Proceedings of the 25th International Conference on Computational Linguistics, COLING* (2014), 183–192. http://www.aclweb.org/anthology/C/C14/C14-1019.pdf{%}5Cnhttp://www.aclweb.org/anthology/C14-1019

[22] Radim Rehurek and Petr Sojka. 2011. Gensim—statistical semantics in python. *statistical semantics; gensim; Python; LDA; SVD* (2011).

[23] Ylja Remmits. 2017. *Finding the Topics of Case Law : Latent Dirichlet Allocation on Supreme Court Decisions.* Ph.D. Dissertation.

[24] Michael Röder, Andreas Both, and Alexander Hinneburg. 2015. Exploring the Space of Topic Coherence Measures. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15* (2015), 399–408. https://doi.org/10.1145/2684822.2685324

[25] Scott Deerwester, Richard Harshman, Susan T, George W, and Thomas K. 1990. Indexing by Latent Semantic Analysis. *Journal Of THe American Society For Information Science* 41, 6 (1990), 391–407. https://doi.org/10.1017/CBO9781107415324.004 arXiv:arXiv:1011.1669v3

[26] Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. *Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL '12* July (2012), 952–961.

[27] Shaheen Syed and Marco Spruit. 2018. Full-Text or abstract? Examining topic coherence scores using latent dirichlet allocation. *Proceedings - 2017 International Conference on Data Science and Advanced Analytics, DSAA 2017* 2018-January (2018), 165–174. https://doi.org/10.1109/DSAA.2017.61

[28] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *J. Amer. Statist. Assoc.* 101, 476 (2006), 1566–1581. https://doi.org/10.1198/016214506000000302

[29] Elham Vaziripour, Christophe Giraud-Carrier, and Daniel Zappala. 2016. Analyzing the Political Sentiment of Tweets in Farsi. *Tenth International AAAI Conference on Web and Social Media* Icwsm (2016), 699–702.

B

Predicting Judicial Outcomes in the Brazilian Legal System Using Textual Features

Predicting Judicial Outcomes in the Brazilian Legal System Using Textual Features

Vithor Gomes Ferreira Bertalan[1][0000−0002−1585−7694] and
Evandro Eduardo Seron Ruiz[2][0000−0002−7434−897X]

[1] PPG-CA, DCM, FFCLRP, University of São Paulo (USP), vbertalan@usp.br
[2] Department of Computing and Mathematics, FFCLRP, University of São Paulo,
evandro@usp.br

Abstract. The combination of Natural Language Processing and Artificial Intelligence for the field of Law is a growing area, with the potential of radically changing the daily routine of legal professionals. The amount of text generated by those professionals is outstanding, and to this point, still unexplored by Computer Science. One of the most acclaimed research field covering both knowledge areas is Legal Prediction, in which intelligent systems try to predict specific judicial characteristics, such as the judicial outcome or the judicial class or a given case. This research intends to create a classifier to predict judicial outcomes in the Brazilian legal system. At first, we developed a text crawler to retrieve judicial outcomes from the official Brazilian electronic legal systems. Afterward, a few judicial subjects were selected, and some of their features were extracted. Later, a set of different classifiers was applied to predict the legal considering these textual features.

Keywords: Legal prediction · Digital humanities · Artificial Intelligence and Law

1 Introduction and Research Objectives

1.1 Introduction

The combination of Natural Language Processing and Artificial Computer Science has been revolutionizing many different fields of expertise. Subfields of Computer Science, like Natural Language Processing (also known as NLP), have steadily improved a myriad of professional and scientific activities. NLP helps researchers to understand how computers can process and analyze large amounts of natural language data and their meanings. Even simple NLP mechanisms, such as dictionaries and word counts, can offer interesting underlying facts that cannot always be noticeable without effective processing.

Law is one of the knowledge areas that are the most dependent on text data. Millions of legislation workpapers, court decisions, and appeals are produced daily, and many different job specializations, such as lawyers, judges, defendants, and plaintiffs, have various necessities that could be supplied by intelligent systems.

Over the last years, researchers have been dedicated to predicting judicial case outcomes using NLP application software and Machine Learning methods over those textual cases. See Section 1.3 below. However, no research with this intention has been done in Brazilian Portuguese for Brazilian courts, as of 2019.

Branting et al. [7] cites that automation of legal reasoning and problem-solving has been a goal of Computer Science research from its earliest days. However, according to the author, broad adoption of legal computer systems never occurred, and Computer Science and law remained a niche research area with little practical impact.

Hyman et al. [10] point out the Zubulake v. UBS Warburg case, a series of trials and decisions dealing with what data a litigant must preserve and under what circumstances the parties must pay for search and production costs, as a seminal case for AI in Law. According to the authors, this case became a landmark for the practical applications of the research. This case is mainly about the inability of the defendant to retrieve hundreds to thousands of emails that were claimed by the plaintiff to be relevant to the main issue in the lawsuit.

1.2 Research Objectives

As a primary objective, this research intends to develop a framework to predict judicial outcomes in the Brazilian state of São Paulo Justice Court. The São Paulo Justice Court is the most significant legal court on the planet, considering the number of cases per year. We believe that designing a predicting model that offers consistent results for this judicial court this model could be transferred, after fine-tuning, to any other court. Firstly, we will develop a text crawler to retrieve data from the legal outcomes. To pre-process these data, we combine NLP tools to extract characteristics from the text, selecting what is believed to be the primary information that can lead to valid predictions. After this step, we will insert these pre-processed data into machine learning frameworks. Finally, we evaluate all the methods and their respective results against the real judicial outcomes of the court.

1.3 Related Works

Recent advancements have significantly improved the state-of-the-art in the field of legal prediction. In the most influential work, Aletras et al. [3] have used a dataset of cases from the European Court of Human Rights, containing cases that violate Article 3 (*Prohibits torture and inhuman and degrading treatment*), Article 6 (*Protects the right to a fair trial*), and Article 8 (*Provides a right to respect for one's private and family life, his home and his correspondence*) of the

Convention. Katz et al. [12] have used random forests to predict the behavior of the Supreme Court of the United States.

In another influential research, Sulea et al. [18] have done a similar investigation, predicting the law area and decisions of the French Supreme Court using lexical features and support vector machine, SVM. The authors have used a diachronic collection of rulings from the French Supreme Court (*Court de Cassation*, in European French).

Recent researches have used machine learning successfully to improve Law decisions. Gokhale and Fasli [9] have developed a co-training algorithm to classify human rights abuses, using SVM and Logistic Regression. Branting et al. [7] have used hierar-chical attention networks, SVMs, and maximum entropy classifications for decision support in administrative adjudication, such and routine licensing, permitting, immigration, and benefits decisions. SVMs are also used by Fornaciari and Poesio [8] to automatically detect deception, such as defamation and false testimony, in Italian court cases. Remnits [16] has used Latent Dirichlet Allocation to discover the main topics of discussion in judicial outcomes of the United States Supreme Court. Mochales [15] has used argumentation mining to structure better legal arguments, capturing main issues and evidence of a given corpus.

Not directly related to the scope of legal prediction used in this paper, some recent research has also been conducted in Brazilian Portuguese. Aires et al. [1] have used deontic logic to identify norm conflicts in contracts. In contrast, Araujo, Rigo and Barbosa [5] have used ontology-based algorithms to classify legal documents in Brazilian judicial outcomes.

Liu and Chen [14] also write that natural language processing and machine learning have augmented possibilities based on the exploration of semantic of law and case texts. Also, according to Barraud [6], NLP and AI have expanded the limits of justice, transforming it into a predictive, quantitative, statistic, and simulative justice. As stated by Alarie, Niblett and Yoon [2], intelligent judicial systems can not only help lawyers with timely and objective assessments of their claims but also governments, by using legal classifiers to help evaluate claims and manage litigation risks.

2 Methodology

2.1 Domain Characterization

A corpus of judicial sentences, along with their outcomes, was collected from eSAJ, the electronic system of the São Paulo Justice Court (TJSP). To restrict the number of documents retrieved, a few previously defined judicial subjects were selected, which are: second-degree murder (in Brazilian Portuguese, *homicídio simples*), and active corruption (in Brazilian Portuguese, *corrupção ativa*).

Only subjects with very well defined outcomes will be selected. As well defined outcomes, we intend to choose those judicial outcomes with the condemnation or absolution of the defendant. Many different legal subjects, such as divorce papers, do not have explicit terms for condemnation or absolution. Therefore, it is of utmost importance to find those judicial subjects with clear and well-established results. We also applied numerical labels to features as condemnations and to the absolutions. These binary labels are used to develop a mathematical/statistical model that one may predict the conclusion of the cases mentioned above based on their textual structure.

2.2 Data Collection

We have implemented a web text crawler to retrieve data from eSAJ, using Python. As the user can select from many different fields to exhibit the judicial opinions, such as classes, subjects, judges, and processes numbers, the crawler must be able to choose from those different query choices to compose a raw text file. The following fields were collected: judicial class, judicial subject, judge, county, release date, and full text of the judicial sentence.

In addition to the data collected from the text, additional fields were derived from the information collected: gender of the judge, and a boolean field indicating whether the county was the capital city of São Paulo, or a state city.

To classify each of the judicial outcomes, we have created the binary labels for condemnation (+1) and absolution (-1). We have also used the professional guidance of Brazilian lawyers, with the purpose of better understanding the texts. The language adopted in the field of Law worldwide might be notoriously obscure for a layman.

The data collection poses a compelling challenge, as the amount of data displayed on the website of the TJSP is indeed substantial. As of June 1st, 2018, a simple search for the Brazilian Portuguese correspondent of rape (*estupro*), for example, returns 5,658 hits. Another search for the Brazilian Portuguese term for drug (*droga*) returns 138,956 hits. Each hit is a judicial opinion of its own, containing many sentences and text topics. As each judicial class under the Brazilian law system contains different text topics, e.g. the topics in a text from the class *divorce papers* (in Brazilian Portuguese, *documentos de divórcio*) differ substantially from texts from the class release permits (in Brazilian Portuguese, *alvará de soltura*).

Table 1 illustrates the number of documents retrieved from the eSAJ system.

Table 1. Number of texts collected by judicial subject

Judicial Subject	**Number of Cases**	**Absolutions**	**Convictions**
Second-degree Murder	591	255	336
Active Corruption	191	31	158

2.3 Data Preprocessing

The data retrieved was pre-processed to remove unnecessary information. We began by tokenizing the text and eliminating stopwords. For this task, we used the Natural Language Processing Toolkit in Python, called NTLK. NLTK can deal with different languages other than English, such as Brazilian Portuguese, which makes this framework a strong candidate for this research. Unusual characters, such as hyphens or parentheses, were also removed. After those steps, we have proceeded with the stemming of the resulting text. For the testing rounds, we used a 10-fold cross validation.

2.4 Data Transformation

In this step, we transformed the data into a mathematical sequence that can be passed through machine learning algorithms. We used TFIDF (term frequency–inverse document frequency) to transform each of the sentences into numbers that are processed through various machine learning (ML) algorithms.

3 Preliminary Results

3.1 Homicides Data Set

We have already tested the two different datasets with various ML methods. The methods selected were: Logistic Regression (LR), Linear Discriminant Analysis (LDA), Gaussian Naïve Bayes (GNB), K-Neighbor (KN), Support Vector Machines (SVM) and, Regression Trees (RT). All the tests were initially run by splitting the training set with 75% of the whole data set, and the test set with the remaing 25% .

For the first data set, the homicides legal texts, we found the calibration plot in Fig. 1. In calibration plots using Platt Scaling, the closest to the perfect calibrated curve, the better. Therefore, from this figure, we can infer that Regression Trees show better results. Support Vector Machines presented the lowest performance, predicting values very differently to those that would be expected based on the reviewed literature.

At last, we run a k-fold cross validation, with 10 epochs, for each of the algorithms. The results are shown in Table 2 and Fig. 2. LDA and Linear Regression, in these tests, were the best choices among all the algorithms.

Table 2. Results (by mean values) on the homicides database, after 10 epochs k-fold

Algorithm	**Acccuracy**	**Accuracy σ**	**Recall**	**Precision**	**F1 Score**
Logistic Regression	**0.893390**	0.040169	0.912864	0.896453	**0.903230**
LDA	**0.908588**	0.035791	0.909055	0.922748	**0.914758**
K-Neighbors	0.815621	0.045490	0.873097	0.813255	0.839270
Regression Trees	0.881610	0.019635	0.887465	0.907709	0.895757
Gaussian Naïve Bayes	0.700678	0.085623	0.968779	0.661592	0.783059
Support Vector Machines	0.568644	0.080258	1.000000	0.568644	0.900421

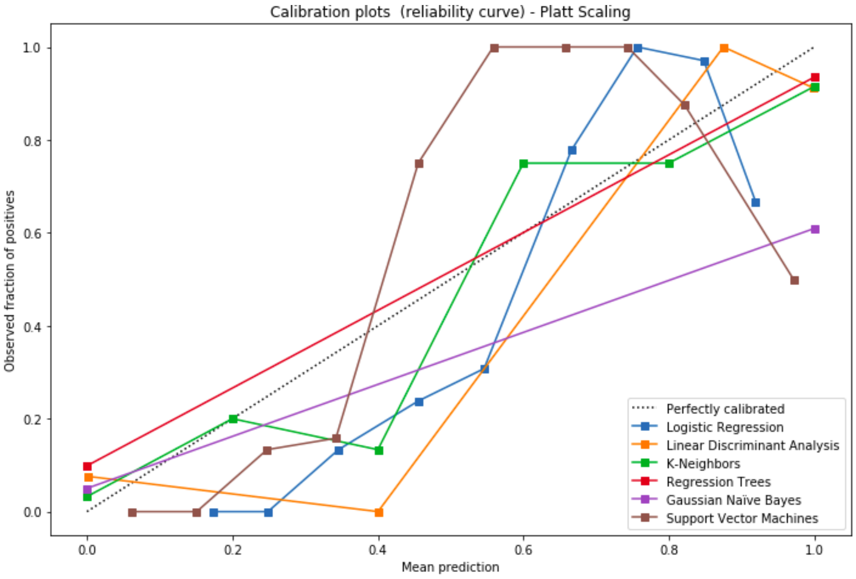


Fig. 1. Calibration plot for the homicides database.

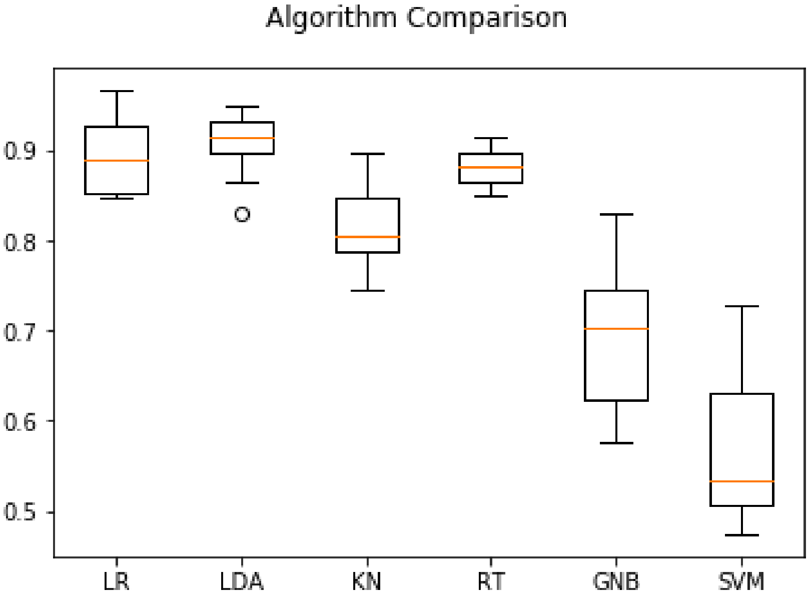


Fig. 2. Candlesticks for the algorithms with 10 k-fold

3.2 Corruption Data Set

For the second data set, with the corruption legal texts, we found the calibration plot shown in Fig. 3. In Platt Scaling, the closest to the perfect calibrated curve, the better. As the previous results with the Homicides data set, we can infer that Regression Trees are the best choice. K-Neighbors showed the lowest performance, predicting values very differently to those that would be expected. All the tests were initially run by splitting the training set considering 75% for the training set and 25% for the test set.

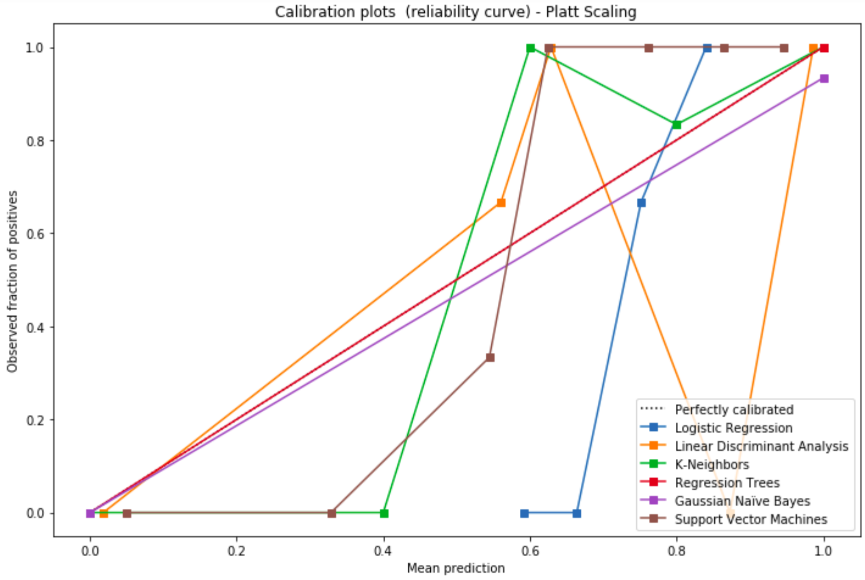


Fig. 3. Calibration plot for the corruption database.

Those results show that, at least for the two databases being evaluated, Regression Trees are the best choice for predicting the outcomes of legal texts. In both data sets, it showed to the the best choice, or being among the best choices available.

4 Conclusions and Future Steps

After the tests done, we can infer that Regression Trees are the best method to predict results in both data sets being analyzed. Even though the other algorithms showed varying results. SVM, as an example, showed a good performance in the corruption database, but the lowest value in the homicides database. Regression Trees have always kept good predicting outcomes. Those results match

Table 3. Results (by mean values) on the corruption database, after 10 epochs k-fold.

Algorithm	**Acccuracy**	**Accuracy (σ)**	**Recall**	**Precision**	**F1 Score**
Logistic Regression	0.824269	0.100378	1.000000	0.824269	0.900421
LDA	0.840351	0.108212	0.963918	0.864229	0.907444
K-Neighbors	0.824854	0.096681	0.963918	0.846981	0.897678
Regression Trees	**0.973684**	0.042433	0.988070	0.974167	**0.980524**
Gaussian Naïve Bayes	0.866374	0.101778	0.989474	0.876901	0.925169
Support Vector Machines	0.824269	0.100378	1.000000	0.824269	0.900421

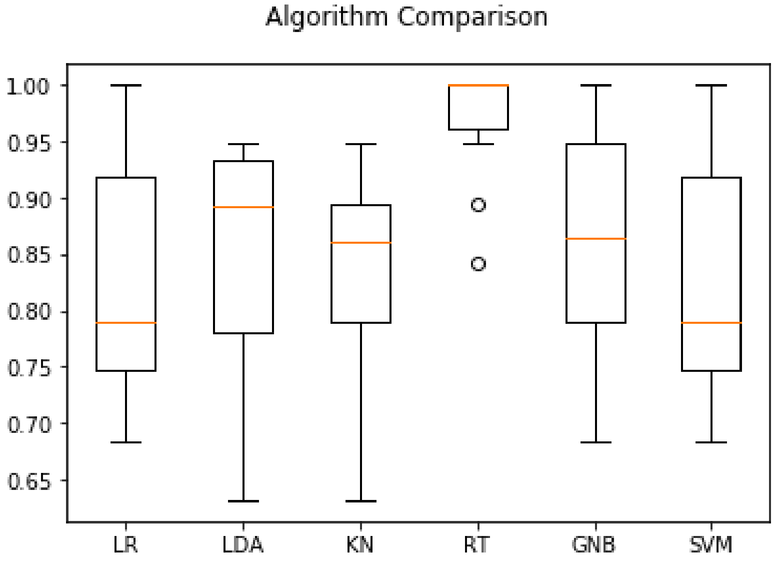


Fig. 4. Candlesticks for the algorithms with 10 k-fold

other results found by other researches in the legal area, in many different countries, such as the one conducted by Kastellec [11], who obtained good outcomes by using Regression Trees in the American legal system. The author mentions that Regression Trees have the capability of studying legal conceptions of Law, revealing patterns that other methods cannot emulate as effectively.

Other researchers also used the same method, such as Rios-Figueroa [17], who used Regression Trees to analyze the concept of judicial independence and corruption among Supreme Courts in Latin America, Antonucci et.al. [4], who adopted Regression Trees to measure the efficiency of Italian courts, and Kufandirimbwa and Kuranga [13], who used the same method to predict outcomes in Zimbabwe.

Those researches show that, even though legal systems are extremely different around the world and throughout different languages and countries, such as Brazil, USA, Italy and Zimbabwe, they do have similar characteristics that can be effectively measured by the correct algorithms.

As future steps, we plan to adopt different methods of converting word embeddings into whole texts, so that we can also utilize methods, such as neural networks, and, eventually, compare these with the ones mentioned in this work.

References

1. Aires, J.P., Pinheiro, D., De Lima, V.S., Meneguzzi, F.: Norm conflict identification in contracts. Artificial Intelligence and Law **25**(4), 397–428 (2017)
2. Alarie, B., Niblett, A., Yoon, A.H.: How artificial intelligence will affect the practice of law. University of Toronto Law Journal **68**(supplement 1), 106–124 (2018)
3. Aletras, N., Tsarapatsanis, D., Preoţiuc-Pietro, D., Lampos, V.: Predicting judicial decisions of the european court of human rights: A natural language processing perspective. PeerJ Computer Science **2**, e93 (2016)
4. Antonucci, L., Crocetta, C., D'Ovidio, F.D.: Evaluation of Italian Judicial System. Procedia Economics and Finance **17**(September 2015), 121–130 (2014). https://doi.org/10.1016/s2212-5671(14)00886-7, http://dx.doi.org/10.1016/S2212-5671(14)00886-7
5. de Araujo, D.A., Rigo, S.J., Barbosa, J.L.V.: Ontology-based information extraction for juridical events with case studies in brazilian legal realm. Artificial Intelligence and Law **25**(4), 379–396 (2017)
6. Barraud, B.: An algorithm that can predict judges' decisions: Toward a robotization of justice? Les Cahiers de la Justice (1), 121–139 (2017)
7. Branting, L., Yeh, A., Weiss, B., Merkhofer, E., Brown, B.: Inducing predictive models for decision support in administrative adjudication. In: AI Approaches to the Complexity of Legal Systems: AICOL International Workshops 2015-2017: AICOL-VI@ JURIX 2015, AICOL-VII@ EKAW 2016, AICOL-VIII@ JURIX 2016, AICOL-IX@ ICAIL 2017, and AICOL-X@ JURIX 2017, Revised Selected Papers. vol. 10791, p. 465. Springer (2018)
8. Fornaciari, T., Poesio, M.: Automatic deception detection in italian court cases. Artificial intelligence and law **21**(3), 303–340 (2013)
9. Gokhale, R., Fasli, M.: Deploying a co-training algorithm to classify human-rights abuses. In: 2017 International Conference on the Frontiers and Advances in Data Science (FADS). pp. 108–113. IEEE (2017)
10. Hyman, H., Sincich, T., Will, R., Agrawal, M., Padmanabhan, B., Fridy, W.: A process model for information retrieval context learning and knowledge discovery. Artificial Intelligence and Law **23**(2), 103–132 (2015)
11. Kastellec, J.P.: The Statistical Analysis of Judicial Decisions and Legal Rules with Classification Trees. Journal of Empirical Legal Studies **7**(2), 202–230 (2010). https://doi.org/10.1111/j.1740-1461.2010.01176.x
12. Katz, D.M., Bommarito, M.J., II, J.B.: A general approach for predicting the behavior of the supreme court of the united states. PloS one **12**(4) (2017)
13. Kufandirimbwa, O., Kuranga, C.: Towards Judicial Data Mining : Arguing for Adoption in the Judicial System **1**(2), 15–21 (2012)
14. Liu, Z., Chen, H.: A predictive performance comparison of machine learning models for judicial cases. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–6. IEEE (2017)
15. Mochales, R., Moens, M.F.: Argumentation mining. Artificial Intelligence and Law **19**(1), 1–22 (2011)
16. Remmits, Y.L.J.A.: Finding the topics of case law: Latent dirichlet allocation on supreme court decisions. B.Sc., Faculteit der Sociale Wetenschappen (7 2017), supervisors: Kachergis, G.E. ; Kuppevelt, D. van ; Dijck, G. van
17. Rios-Figueroa, J.: Judicial Independence and Corruption: An Analysis of Latin America. SSRN Electronic Journal (212) (2011). https://doi.org/10.2139/ssrn.912924

18. Şulea, O.M., Zampieri, M., Vela, M., van Genabith, J.: Predicting the law area and decisions of french supreme court cases. In: Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017. pp. 716–722 (2017)