UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO E CIENTÍFICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Thiago Raulino Dal Pont

Representation, classification and regression techniques applied to legal judgments about immaterial damage due to failures in air transport services

Florianópolis
2021

Thiago Raulino Dal Pont

Representation, classification and regression techniques applied to legal judgments about immaterial damage due to failures in air transport services

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarinapara a obtenção do título de mestre em Engenharia de Automação e Sistemas.
Supervisor:: Prof. Jomi Fred Hübner, PhD.
Co-supervisor:: Prof. Aires José Rover, PhD.

Florianópolis
2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Dal Pont, Thiago Raulino
Representation, classification and regression techniques applied to legal judgments about immaterial damage due to failures in air transport services / Thiago Raulino Dal Pont ; orientador, Jomi Fred Hübner, coorientador, Aires José Rover, 2021.
120 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2021.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Aprendizado de Máquina. 3. Mineração de Textos. 4. Juizado Especial Cível. 5. Transporte aéreo. I. Hübner, Jomi Fred. II. Rover, Aires José. III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Thiago Raulino Dal Pont

Representation, classification and regression techniques applied to legal judgments about immaterial damage due to failures in air transport services

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Fabiano Hartmann Peixoto, Dr.
Universidade de Brasilia

Prof. Eric Aislan Antonelo, Dr.
Universidade Federal de Santa Catarina

Prof. Marcelo Ricardo Stemmer, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia de Automação e Sistemas.

Prof. Werner Kraus Junior, Dr.
Coordenador do curso de Pós-Graduação
em Engenharia de Automação e Sistemas

Prof. Jomi Fred Hübner, PhD.
Orientador

Florianópolis, 2021.

To God, my family and friends.

ACKNOWLEDGEMENTS

I thank God for everything, but mainly for allowing me to work on this research amidst so many tribulations and for having completed this step.

I also thank my family, especially my parents and my brother, Gilmar, Rosane, and Gabriel, for all their love, support, and patience during the Master's period.

I thank my advisor, Professor Jomi Fred Hübner, PhD., for all his commitment, dedication, and willingness to guide me during the Master's period and, mainly, for showing me the way to grow as a researcher.

I would like to thank my co-supervisor, Professor Aires José Rover, PhD., also for his commitment, dedication, and willingness during the Master's period, for showing the research from another angle, and also for allowing the use of the EGOV's high-performance computer, without which this work would not be possible.

I need to thank Isabela, Msc., a doctoral student in law, my right-hand man during this research, whether in times of bad results from experiments, in the data collection, in the paper writing, or in times of joy when receiving their acceptance.

I must thank my colleagues in the EGOV group, for the moments of relaxation and the shared experiences, knowledge, and techniques for scientific research.

I thank the professors, colleagues, and servants of the Automation and Systems Department at UFSC, for all the shared knowledge and assistance provided.

I thank the Judge Vânia Petermann, from the Special Civel Court at the Federal University of Santa Catarina for allowing the research to be conducted by providing access to the court's data.

I thank everyone who in some way provided assistance during the research period but were not explicitly named.

Finally, I would like to thank CAPES and Petrobras for allowing this work to be carried out through the granting of a master's scholarship.

"Fortitude is the disposition of soul
which enables us to despise all inconveniences
and the loss of things not in our power"
(St. Augustine, 354-430)

RESUMO

De acordo com o último relatório Justiça em Números, publicado anualmente pelo Conselho Nacional de Justiça, 77,1 milhões de processos aguardavam solução no Judiciário brasileiro, sendo 5,2 milhões nos Juizados Especiais Civeis (JECs). Esses números vêm crescendo ano a ano, o que indica a necessidade de criação de mecanismos para auxiliar o Judiciário brasileiro a ser mais célere e eficaz. Assim, este trabalho visa contribuir nesse cenário aplicando as técnicas de Aprendizado de Máquina (ML) e Mineração de Textos (TM) para a previsão dos resultados dos julgamentos do JEC localizado na Universidade Federal de Santa Catarina, relacionados a falhas nos serviços de transporte aéreo. Para isso, dividimos o problema em três partes: representação, classificação e regressão. Na primeira parte, avaliamos a possível influência do tamanho e a especificidade dos *corpora* usados para treinar *word embeddings* sobre o desempenho da classificação de textos. Desse modo, *word embeddings* foram treinados com base em julgamentos em português. Como resultado, descobrimos que tamanho e especificidade importam, porém o tamanho influencia os resultados até certo ponto. Na segunda parte, avaliamos se as técnicas de Aprendizado Profundo (DL) apresentam melhor desempenho do que as técnicas de ML Clássico na predição dos resultados dos julgamentos do JEC. Assim, treinamos várias técnicas de DL e ML Clássico usando dois tipos de conjunto de dados, um contendo o texto completo dos julgamentos e outro com a parte do dispositivo removida. No primeiro caso, as técnicas de DL apresentam melhor desempenho, o que significa que podem assimilar melhor as partes dos textos que explicitamente indicam o resultado. Neste último, as técnicas clássicas tiveram melhor desempenho, indicando que sem a parte do resultado estar explícita, essas técnicas podem melhor aprender como classificar a partir das demais partes do processo. Na terceira parte, nos concentramos em prever o valor da indenização por danos morais, usando técnicas de regressão. Com base em vários pipelines e na avaliação de um especialista em direito, percebemos que a qualidade de predição alcançada em tal tarefa é aceitável e pode ser útil no domínio jurídico. Assim, concluímos que foi possível prever com precisão os resultados dos julgamentos do JEC e o valor da indenização por danos morais utilizando os *pipelines* propostos.

Palavras-chave: Aprendizado de Máquina. Classificação de Texto. Representação de Texto. Regressão de Texto. Juizado Especial Cível. Decisões Judiciais.

RESUMO EXPANDIDO

INTRODUÇÃO

De acordo com o último relatório *Justiça em Números*, publicado anualmente pelo Conselho Nacional de Justiça (CNJ), ao final de 2019 havia cerca de 77,1 milhões de processos em andamento aguardando solução no Judiciário brasileiro. No total, em 2019, foram 30,2 milhões de ações ajuizadas em todo o Judiciário, um aumento de 6,8% em relação a 2018. Dessas ações, cerca de 5,2 milhões foram ajuizadas nos Juizados Especiais Cíveis (JEC) (CNJ, 2020a). Para fazer face aos desafios enfrentados nos sistemas judiciários no Brasil e no mundo (SADIKU, 2020), é cada vez maior o interesse da literatura em aplicar técnicas baseadas em aprendizado de máquina e mineração de textos na área jurídica. Os artigos publicados buscam solucionar diversos problemas jurídicos, como extração de informações de contratos (HASSAN; LE, T., 2020) e previsão de decisões em tribunais inferiores e superiores (ŞULEA et al., 2017; VIRTUCIO et al., 2018). O aprendizado de máquina se concentra na construção de sistemas de computador que aprendem por meio da experiência (MITCHELL, 1997), enquanto a mineração de textos está relacionada à ideia de descobrir e analisar padrões, como tendências e outliers, a partir de dados textuais. Mineração de textos também se concentra em ajudar os usuários a analisar e digerir informações para uma melhor tomada de decisão (AGGARWAL; ZHAI, 2012). Considerando o alto nível de judicialização no Brasil, menos casos são decididos por meio de acordos entre as partes e, por isso, essas têm de esperar pelo julgamento de um juiz, o que pode levar de dias a anos (CURY et al., 2019; MANCUSO, 2020). Se as partes possuíssem mais informações sobre os possíveis resultados de seus processos, baseadas em julgamentos anteriores do juíz, elas poderiam encerram o caso ainda na conciliação. Além disso, nos JECs, as decisões ainda são manuais, sem nenhum tipo de automação. O mesmo vale para decisões envolvendo danos morais, onde não há critérios claros para definição dos valores, dependendo muito da interpretação do juiz a cerca do caso. Portanto, este trabalho pretende contribuir para o estado da arte em aprendizado de máquina e em mineração de textos através da investigação de soluções para predição dos possíveis resultados de processos. Tais processos são provenientes do JEC localizado na Universidade Federal de Santa Catarina e envolvem falhas em serviços de transporte aéreo. No entanto, para tal predição ocorra, três desafios são colocados: a representação numérica de textos jurídicos, necessária para a aplicação destes em algoritmos de aprendizado de máquina, levando em conta sua complexidade e vocabulário próprios. Porém, não há trabalhos explorando a representação de textos jurídicos em português a partir de técnicas como o *word embeddings*. O segundo desafio se refere à predição do resultado do processo, através da tarefa de classificação, onde pretende-se treinar um modelo capaz de mapear um conjunto de dados para um conjunto de finito de rótulos. O terceiro se refere ao uso de regressão para predição do valor do dano moral a partir do texto do processo. No entanto, a partir de uma revisão da literatura, pode se afirmar que regressão ainda não foram aplicadas a textos jurídicos.

OBJETIVOS

Com base na problemática de predição do resultado dos processos do JEC na UFSC, o

objetivo do trabalho é avaliar se é possível prever o julgamento final do caso a partir do seu texto e prever o valor de indenização por danos morais usando técnicas de aprendizado de máquina e mineração de textos. E, para levantar possíveis soluções para os três desafios, três objetivos específicos precisam ser executados. O primeiro deles se refere à representação, onde busca avaliar se o tamanho e a especifidade dos *corpora* usados para treinar modelos de *word embeddings* impacta na performance na tarefa classificação de textos. O segundo a partir dos resultados do primeiro busca avaliar se técnicas de aprendizado profundo superam técnicas clássicas de aprendizado de máquina na predição do resultado dos julgamentos do JEC. E, por fim, o terceiro objetivo se refere à regressão, onde busca-se avaliar se a predição de indenização por danos morais pode ser precisa e útil no ambiente jurídico a partir do uso de regressão em textos.

MÉTODO DE PESQUISA

Para cumprir os objetivos, o pesquisador seguiu um conjunto de passos aos quais são similares para cada objetivo específico, com pequenas distinções. Anteriormente à definição do problema, pergunta de pesquisa e objetivo de cada uma das três partes, construiu-se três Revisões Sistemáticas da Literatura (RSL) para identificar o estado arte em relação a aplicação de aprendizado de máquina e mineração de textos no direito e o uso de técnicas de representação e regressão no direito. O passo seguinte consistiu na coleta das bases textuais necessárias para cumprir os objetivos. Tais conjuntos de dados incluiram bases textuais de acórdãos de tribunais superiores bem como julgamento do JEC. Por fim, com o ajuda da especialista em direito, foram extraídos atributos a cerca dos processos. O terceito passo se refere à construção de experimentos de aprendizado de máquina e mineração de texto para cada objetivo. O quarto passos se refere à avaliação dos resultados obtidos e discussão.

RESULTADOS E DISCUSSÃO

Os três conjuntos de experimentos realizados buscaram atingir os três objetivos propostos. Em relação à representação, discutiu-se o impacto do tamanho e especificade dos *corpora* usado para treinar *word embeddings* na performance na classificação de textos. Percebeu-se que tanto tamanho e especificidade importam. No entanto, em termos de tamanho, percebeu-se que este impacta até um certo ponto de tal forma que adicionar mais texto não traz impactos significativos. Em relação à classificação, buscou-se comparar como técnicas de aprendizado profundo e clássicas se desempenhavam na predição dos resultados dos processos do JEC e se a primeira traria resultados superiores em relação à segunda. Percebeu-se que técnicas de aprendizado clássico se saem melhor quando o texto do julgamento não apresenta a seção de dispositivo, enquanto técnicas de aprendizado profundo performaram melhor quanto tal seção estava presente. Portanto, a aplicação real da predição no JEC usaria técnicas clássicas. Em relação aos experimentos com regressão, percebeu-se que, a partir de aprimoramentos na pipeline para predição do valor de indenização, pode-se obter resultados com qualidade aceitável para o ambiente jurídico, uma vez que o erro médio absoluto atingido foi de menos de mil reais. Portanto, com base nos resultados alcançados, é pode se afirmar que a predição do resultado e da indenização por danos morais em julgamentos do JEC é possível a partir de técnicas aprendizado de máquina e de mineração de textos.

CONSIDERAÇÕES FINAIS

A cada ano, o número de processos judiciais aguardando decisão final tende a aumentar. Portanto, é fundamental encontrar soluções para agilizar o Judiciário brasileiro em suas diversas instâncias, desde os Juizados Especiais Cíveis até o Supremo Tribunal Federal. Neste trabalho, é proposta a aplicação das técnicas de mineração de textos e aprendizado de máquina em julgamentos do Juizado Especial Civel da UFSC para prever os possíveis resultados bem como o valor da indenização por danos morais. Assim, esta pesquisa visa contribuir para o aumento de acordos em audiências de conciliação nos juizados especiais. O objetivo e a pergunta de pesquisa se refereriram à possibilidade de se prever o resultado dos julgamentos do JEC a partir do seu conteúdo, bem como o valor de indenização por danos morais. Para tanto, três desafios foram colocados para que se atingisse o objetivo geral, sendo eles a representação de textos jurídicos, a predição do resultado do julgamento, a partir de classificação e a predição da indenização por dano moral, a partir de regressão. Em relação à representação discutiu-se o impacto do tamanho e especificidade dos *corpora* para treinamento de *word embeddings* na performance classificação e percebeu-se que havia influência de ambos, no entanto o tamanho apenas até certo ponto. Em relação à classificação, percebeu-se que a predição de julgamentos do JEC foi melhor executada por técnicas clássicas de aprendizado de máquina em relação ao aprendizado profundo. Já em relação à predição de indenização por danos morais atingiu resultados aceitáveis dentro do ambiente jurídico a partir de aprimoramentos como N-Grams e atributos extraídos pela especialista em direito. Em relação ao objetivo geral, pode-se concluir que é possível prever o resultado do julgamento a partir do texto, uma vez que a melhor acurácia foi de 82,2%. Também é possível atingir resultados acuráveis na predição de indenização por danos morais a partir de regressão com o uso dos aprimoramento sugeridos. Como trabalhos futuros, almeja-se verificar como técnicas de representação mais recentes influenciam nos resultados de classificação e regressão. Além disso, pretende-se verificar se é possível prever o resultado e o valor de indenização a partir do uso de textos de petições iniciais. Por fim, pretende-se aprimorar os modelos criados para que as predições sejam *explicáveis* e incluir mecanismos para mitigar de eventuais vieses e preconceitos que possam estar presentes na base de dados.

Palavras-chave: Aprendizado de Máquina. Classificação de Texto. Representação de Texto. Regressão de Texto. Juizado Especial Cível. Decisões Judiciais.

ABSTRACT

According to the last report *Justiça em Números*, annually published by the National Council of Justice, 77.1 million processes were waiting for a solution in the Brazilian judiciary, of which 5.2 million in the Special Civel Courts (JECs). Those numbers have been increasing year after year, indicating the need of creating mechanisms toto help the Brazilian Judiciary to be faster and more effective. Thus, this work aims to contribute in this scenario by applying Machine Learning (ML) and Text Mining (TM) techniques to the prediction of the results of the legal judgments from the JEC located at the Federal University of Santa Catarina, which relate to failures in air transport services. To do so, we divided the problem into three parts: representation, classification, and regression. In the first part, we evaluate whether the size and specificity of the corpora used to train word embeddings, impact the performance of text classification. We, thus, trained embeddings based on judgments in Portuguese. As a result, we discovered that size and specificity matter, however size influences the results until a certain point. In the second part, we evaluate whether Deep Learning (DL) techniques perform better than Classical ML techniques in the classification of the judgments' results from JEC. Thus, we trained several DL and Classical ML techniques using two types of the dataset, one containing the full text of the judgments and another with the result part removed. In the former, the DL techniques performed better, implying that they can better assimilate the parts of the texts that explicitly indicate the result. In the latter, classical techniques performed better, indicating that without the explicit result part, those techniques can better learn from the other parts. In the third part, we focus on predicting the compensation value for immaterial damage, using regression techniques. Based on several pipelines and on a legal expert's evaluation, we noticed that the prediction quality achieved in such a task is acceptable and it can be helpful in the legal domain. Thus, we concluded that it was possible to accurately predict the results of the judgments from JEC and the compensation value for immaterial damage using the proposed pipelines.

Keywords: Machine Learning. Text Classification. Text Representation. Text Regression. Special Civel Court. Legal Judgments.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AB	AdaBoost
AELE	Attributes Extracted by the Legal Expert
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
BG	Bagging
Bi-LSTM	Bidirectional Long-Short Term Memory
BOW	Bag of Words
CBOW	Continuous Bag of Words
CNJ	National Council of Justice
CNN	Convolutional Neural Network
DL	Deep Learning
DT	Decision Tree
EGOV	E-government, Digital Inclusion and Knowledge Society
eJRM	Justice Relationship Management
ELMo	Embeddings from Language Model
EN	Elastic Net
EV	Ensemble Voting
FN	False Negative
FP	False Positive
GB	Gradient Boosting
GPT-3	Generative Pre-trained Transformer-3
GPU	Graphics Processing Unit
GR	General Repercussion
IDF	Inverse Document Frequency
JEC	Special Civel Court
kNN	k Nearest Neighbors
LDA	Latent Dirichlet Allocation
LR	Logistic Regression
LSA	Latent Semantic Analysis
LSTM	Long-Short Term Memory
MAE	Mean Absolute Error
MI	Mutual Information
ML	Machine Learning
NB	Naïve Bayes
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
NMF	Non-negative Matrix Factorization
NN	feed-forward Neural Networks

OCR	Optical Character Recognition
PC	Principal Component
PCA	Principal Component Analysis
PDF	Portable Document Format
POS	Part-of-Speech
RBF	Radial Basis Function
RF	Random Forest
RFE	Recursive Feature Elimination
RG	Ridge
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
RTF	Rich Text Format
SG	Skipgram
SOTA	State of the Art
SRL	Systematic Review of the Literature
STF	Federal Supreme Court
STJ	Superior Court of Justice
SVM	Support Vector Machine
t-SNE	t-distributed Stochastic Neighbor Embedding
TF	Term Frequency
TF-IDF	Term Frequency-Inverse Document Frequency
TJ	State Court
TJ-SC	State Court of Santa Catarina
TM	Text Mining
TN	True Negative
TP	True Positive
UFSC	Federal University of Santa Catarina
UnB	University of Brasilia
VSM	Vector Space Model
XAI	Explainable Artificial Intelligence

CONTENTS

1 INTRODUCTION

According to the last report *Justiça em Números*, published annually by the National Council of Justice (CNJ), by the end of 2019, there were about 77.1 million ongoing processes waiting for a solution in the Brazilian Judiciary. In total, in 2019, 30.2 million lawsuits were filled in all Judiciary, an increase of 6.8% in relation to 2018. From those processes, about 5.2 million were filled in the Special Civel Court (JEC) (CNJ, 2020a).

Justice institutions around the world are being impacted by the application of automation solutions in the most varied contexts. It is a trend the use of Artificial Intelligence (AI) to define a series of strategic measures both for the execution of the core activity and for strategic decisions from the point of view of management and processing flow (HARTMANN, F.; SILVA, R. Z. M. d., 2019).

In view of the large amount of textual data generated by the the justice institutions, empirical methods to analyze that data, such as the Machine Learning (ML) and Text Mining (TM), may be of use (MEDVEDEVA et al., 2019; WEISS et al., 2010). In the literature, ML and TM based techniques have been applied in the legal domain to solve a variety of legal problems, such as the information extraction from contracts (HASSAN; LE, T., 2020), and the prediction of decisions in lower courts and superior courts (ŞULEA et al., 2017; VIRTUCIO et al., 2018).

According to Mitchell (1997), ML focuses on constructing computer systems that learn through experience. Thus, such systems can learn to classify texts, control robots, predict weather and others (SEBASTIANI, 2002; KOBER et al., 2013; SHI et al., 2015). ML based systems can learn using some approaches (SCHMIDHUBER, 2015; CARUANA, 1997), including supervised and unsupervised. In supervised ML, the system maps a relationship between inputs and outputs based on labeled data. In unsupervised ML, the system tries to find patterns in the data taking into consideration the similarities among the unlabeled data points (THEODORIDIS; KOUTROUMBAS, 2009).

TM relates, according to Aggarwal and Zhai (2012), to the idea of discovering and analyzing patterns, such as trends and outliers, from textual data. TM also focus on helping users to analyze and digest information towards a better decision making. Thus, the necessity of TM and ML based applications emerges with the large amount of textual data generated by users, companies, universities and so on, and human limitations to analyze it (LECUN et al., 2015; KHAN et al., 2014).

TM and ML have been used to address challenges regarding supervised learning, such as text classification, text regression, and unsupervised learning like text clustering (AGGARWAL; ZHAI, 2012; TRUSOV et al., 2016; MEDVEDEVA et al., 2019; ZHANG; ZHOU, L., 2019).

1.1 PROBLEM DEFINITION

Considering the high level of litigation in Brazil, where the Judiciary is increasingly assigned to solve the day-to-day conflicts, less disputes (or cases) are decided with agreements between the parties in the earliest phases. So they shall wait for a judgment, that is, the judge's final decision on the case (HILL, G.; HILL, K., 2021), that may take from days to years (CURY et al., 2019; MANCUSO, 2020).

JEC are Judiciary small agencies regulated by Law no. 9,099/1995, which seek to facilitate citizens' access to Justice through simpler and cost-free procedures. As a result, JEC tend to approach the legal problems of ordinary people who find themselves involved in daily conflicts of small economic expression, whether in the purchases they make, in the services they hire or in the accidents they suffer (WATANABE, 1985).

In the JEC, the cases are decided manually by the judge, and there is no automation in that sense. Moreover, depending on the legal contexts, the judge sets an amount of compensation for immaterial damage to be paid by a party to the other. However, there is no explicit rules to define them. Thus, the amount of compensation for immaterial damage depends on the judge's interpretation of the case (SADIKU, 2020).

If the parties had more information on the possible outcomes, based on the judge's previous decisions, they would achieve a consensus and finish the case, without the need of waiting for a judgment. In this context, the use of ML and TM may help in the conciliation between the parties by predicting the possible outcomes based on the previous decisions.

Thus, this work intends to make contributions to the state of the art of ML and TM applied to legal texts by investigating possible solutions for the prediction of legal cases outcomes. To get such achievement, three challenges are addressed. First, the legal text, its complexity, and vocabulary must be represented numerically allowing the use of ML techniques. Second, the prediction of the chances of fully or partially winning or losing the case. And third, the prediction of the amount of compensation when the party wins the case.

In the literature, the first challenge have been addressed in several languages, like Polish, English and Chinese (CHALKIDIS; KAMPAS, 2018; SMYWIŃSKI-POHL et al., 2019). However, from a Systematic Review of the Literature (SRL) (see Appendix B), no work explored the representations of legal texts in the Portuguese language using techniques like word embeddings, Bidirectional Encoder Representations from Transformers (BERT) and others.

The second challenge relates to classification, a supervised ML task which tries to learn a model to map a set of records to a set of labels (AGGARWAL; ZHAI, 2012). Based on a SRL (see Appendix A), one can see that the classification task has been applied in many legal contexts (CHALKIDIS; KAMPAS, 2018; HAMMAMI et al., 2019; HASSAN;

LE, T., 2020). Although, text classification in the legal domain is a well explored subject, the papers found in the SRL do not compare as many techniques for Deep Learning (DL) and Classical ML as this work.

The third challenge can be addressed with the regression task, an supervised ML task which aims to train a model to map the relationship between an input x to a continuous output y (DRAPER; SMITH, 1998). According to a SRL (See Appendix C), to the best of our knowledge, the application of text regression in the legal domain is not addressed in any work in the literature. Thus, there is the possibility of many contributions in this regard.

1.2 RESEARCH QUESTION

"Is it possible to predict the final judgment of a legal case based on its content and predict the amount of compensation for immaterial damage using ML and TM techniques?"

The question can be divided into three:

- Does the size and specificity of the corpus used for word embeddings training impact the performance of a text classification using such representations?
- Can DL techniques outperform Classical ML techniques on the prediction of a judgment from the JEC?
- To what extent the prediction of compensation values can be *accurate* and *helpful* in the legal environment using regression?

1.3 OBJECTIVES

In this section, we introduce to the reader the main objective and the specific objectives necessary to achieve it.

1.3.1 Main Objective

To evaluate whether we can predict the final result of a legal judgment and predict the amount of compensation for immaterial damage based on its content using ML and TM techniques.

1.3.2 Specific Objectives

In order to fulfill the main objective, the following specific objectives must be achieved:

- Evaluate whether the size and specificity of the corpus used for word embeddings impact on the performance a text classification using such representations.

- Evaluate whether DL techniques outperform Classical ML techniques in the prediction of judgments from JEC.
- Evaluate whether the prediction of compensation values can be accurate and helpful in the legal environment using regression.

1.4 JUSTIFICATION

There is an increasing interest in the literature on the applications of ML and TM techniques in the legal domain. From that, the concerns about the implications of AI uses in the legal domain and others emerge (BRAZ et al., 2018; DAVIS, 2020). Worldwide, such concerns provoked the debate of regulations on AI (CATH, 2018). In 2020, the CNJ created an ordinance to regulate the uses of AI in the Brazilian Judiciary (CNJ, 2020b). Thus, the application of ML in the legal domain has important theoretical and practical relevance.

Through the construction of three SRL, detailed in Appendix A, B and C, we observed such increasing research interest on the application of ML and TM in the legal domain in the last years. With the first SRL, we focused on the literature regarding ML and TM in the legal domain. As a result, most of the selected work related to many kinds of predictions in legal documents in many languages. However, until the SRL search date, none of the papers explored and compared as many classification ML techniques on the same dataset as proposed in this work. In general, they focus on a small set of techniques.

In terms of representation of legal texts, this first SRL showed us few relevant work. However, to address the first part of our research question, more search in the literature was needed. To serve as a complement, the second SRL focus on the representation of legal texts, and the results showed us that many representation techniques have been explored in the legal domain, including word embeddings and BERT. However, the available work involving the Portuguese language did not focus on the aspects of representations, such as the size, the specificity of the corpus used to train them. They just apply existing representations pre-trained on general texts, until the SRL search date. Thus, we further analyze how the representations pre-trained on Portuguese improve the performance in text classification when compared to those pre-trained in texts from several contexts.

The third SRL focused on the application of regression in legal texts. However, there were no published papers in the literature on that matter, despite the use of many search keywords patterns. A broader literature search, detailed in Appendix C, showed work regarding other contexts, including Financial and Health. Thus, this work's main contributions relates to the use of regression in legal texts, with the focus on predicting compensations.

In terms of theoretical relevance, this work brings contributions in terms of rep-

resentation of legal texts as it trains word embeddings representations in Brazilian legal documents to be applied in supervised and unsupervised learning tasks. In terms of predicting the judgment through classification, the amount of published work limits our contributions to the evaluation on how the Classical ML and DL techniques behave when applied to judgments from JEC. In terms of predicting the amount of compensation through regression, this work brings contributions on impact of several TM and ML techniques in the pipeline as the performance of the ML models in such task.

In terms of practical contributions, the trained models and pipelines from this work can be adapted to be applied in real legal conciliation hearings in the JEC at Federal University of Santa Catarina (UFSC). A legal expert would present and explain the predictions to the parties. Thus, we expect to help them on reaching an agreement without the need of waiting for a judgment. In this way, the litigation in the JEC would decrease, contributing to faster and more efficient access for citizens to justice.

1.5 RESEARCH METHOD AND RESOURCES

To answer the main question and its three parts, the researcher followed a set of steps. Each part shares common steps.

As the first step, there is the construction of three SRL to evaluate the State of the Art of the application of ML and TM in the Legal Domain, regarding representation, classification and regression. Based on this step, the research question and objectives were set.

The second step relates to the collection of the textual datasets required to answer each question. As detailed in Chapter 4, such datasets include an unlabeled dataset of collegiate judgments from Federal Supreme Court (STF), Superior Court of Justice (STJ) and State Court of Santa Catarina (TJ-SC) and legal judgments from JEC at UFSC. They also include a set of attributes extracted by a legal expert.

The third step relates to the setup and execution of ML experiments according to each part of the research question. That is, experiments involving the training of word embeddings and their evaluation in text classification; experiments to train and compare DL and Classical ML techniques in the prediction of legal judgments from JEC; and experiments to predict the compensation for immaterial damage.

The fourth step relates to the evaluation and discussion of the achieved results in each experiment.

In terms of resources, this work required the use of a high performance computer from the E-government, Digital Inclusion and Knowledge Society (EGOV) research group at UFSC and a set of open source libraries for TM and ML.

1.6 DOCUMENT ORGANIZATION

The work is structured in five chapters, beginning with this introduction.

In **Chapter 2**, we introduce the concepts and terminology required to understand the related work and the experiments we carried out. We, thus, introduce the concepts of ML applied to text, the TM, the theory behind the three parts of the research question and the techniques involved.

In **Chapter 3**, we highlight the relevant work to this research, based on the construction of the three SRL. From this chapter and the SRL, the reader can better understand how the work is positioned in relation to the literature.

In **Chapter 4**, we describe the steps to answer each part of the research question, regarding representation, classification and regression. That is, the dataset used, the pipelines, the results and the discussion on whether the experiments answered the parts of the research question.

In **Chapter 5**, we present the conclusions from this research based on the answers to the parts of research questions. We also present the contributions in terms of publications and gains to the state of the art. Furthermore, there is a discussion on the limitations imposed during this research in terms of the data collection, the results' application in a conciliation hearing and others. Finally, we discuss the possible improvements on this research and the future work.

2 MACHINE LEARNING FOR TEXT

In this chapter, we present the essential concepts to understanding the results and discussion from this work. It starts from the basic definitions related to ML and TM, and the challenges related to them. There is an explanation regarding definitions, techniques, and evaluation of the three tasks applied during this research: text representation, text classification, and text regression. Finally, there is an explanation of pipelines for ML.

2.1 BASIC DEFINITIONS

According to Mitchell (1997), ML is a sub-field from AI and focuses on constructing computer systems that learn to perform tasks through experience. For Samuel (1959), ML is the field that focus on building systems which can learn to solve problems without being explicitly programmed. Such systems can learn to classify texts, control robots, predict weather, and others (SEBASTIANI, 2002; KOBER et al., 2013; SHI et al., 2015).

A sub-field of ML is DL, which brought improvements, achieving State of the Art (SOTA) results on the solution of problems in several areas, such as Natural Language Processing (NLP), Image Classification, and others (BROWN et al., 2020; TAN, M.; LE, Q. V., 2021; SENGUPTA et al., 2020). In addition to those relevant results, humans have been outperformed by DL in complex tasks, like the Chess and Go games (SILVER et al., 2017).

According to LeCun et al. (2015), DL techniques are representation-learning methods with multiple levels of representation. Starting from the raw input, the levels transform the representations until a more abstract level. Thus, a relevant difference among the Classical ML and DL techniques is the ability to deal with the data in its natural form, as in the former, the researcher needs to implement hard-coded feature engineering while in the latter, there is no such need. DL techniques can learn more complex functions and tend to produce larger models, with more hyper-parameters.

ML based systems, both Classical and DL can learn using some approaches (SCHMIDHUBER, 2015; CARUANA, 1997), including supervised and unsupervised. In supervised ML, the system maps a relationship between inputs and outputs based on labeled data. In unsupervised ML, the system tries to find patterns in the data taking into account the similarities among the many date points (THEODORIDIS; KOUTROUMBAS, 2009). Examples of supervised learning tasks include classification, and regression, and examples of unsupervised learning tasks include clustering and topic modeling (KOWSARI et al., 2019; AGGARWAL, 2018; RUSSELL; NORVIG, 2020).

In this work, terms related to ML are frequently used, such as task, technique and model. It is important to clarify each of them. A task in ML relates to *what* the

learning machine intends to do, i.e., the type of inference that is done based on the problem and the data available (QUINTANILLA et al., 2015).

Another terminology regards ML techniques, which relate to mechanisms and algorithms that allow the machine to learn. For example, there are classification techniques like Decision Trees and Naïve Bayes, and clustering techniques such as K-means and Lingo (OSIŃSKI et al., 2004; AGGARWAL, 2018). Based on a ML technique, and the data, the machine learns to perform a task during the process of *training*. The result of such process is a ML *model*, i.e., the representation of the learned function (QUINTANILLA et al., 2015).

Beyond ML, an important field of study applied in this work is TM. TM relates, according to Aggarwal and Zhai (2012), to the idea of discovering and analyzing patterns, such as trends and outliers, from textual data. TM also focus on helping users to analyze and digest information towards a better decision making. Thus, the necessity of TM and ML based applications emerges with the large amount of textual data generated by users, companies, universities and so on, and human limitations to analyze it (LECUN et al., 2015; KHAN et al., 2014). Examples of TM tasks are text classification, text clustering, text regression and others (AGGARWAL, 2018).

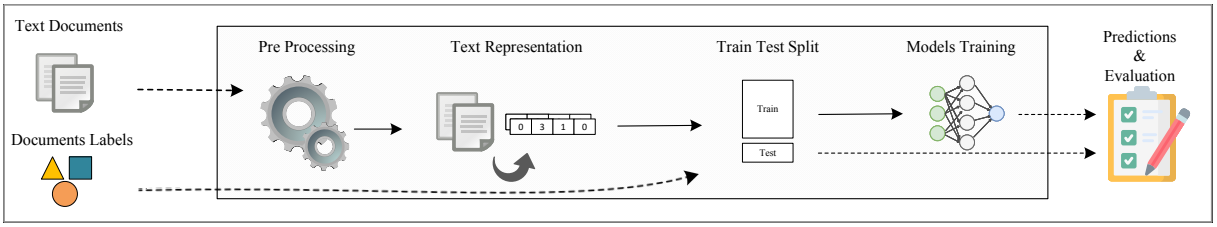
An important field related to TM is NLP, which employs computation techniques to process, learn, understand, and produce human language content. That is, they generally receive textual data as input and produce text as output (HIRSCHBERG; MANNING, 2015). Examples of NLP tasks include text summarizing, Part-of-Speech (POS) Tagging, text preprocessing, text translation, and text generation (GAMBHIR; GUPTA, 2017; BROWN et al., 2020).

When applying ML to texts, generally both TM and NLP may be used. To do so, one can build pipelines to aggregate these three and other types of techniques.

A *pipeline* is an computational architecture composed of several processing units connected in sequence and each of them is responsible for a distinct processing computation (RAMAMOORTHY; LI, 1977). In ML, pipelines help on building complete systems which can receive raw data as input, preprocess it, train the models and evaluate their performances (OBRIEN et al., 2021).

In Figure 1, there is an example of ML pipeline to text classification (KOWSARI et al., 2019).

Figure 1 – Example of pipeline for text classification



It receives two types of inputs: the text from the documents and the labels. The data passes by some steps, starting by the pre-processing and the textual representation.

To train the classification models, it is required the data and the technique. To evaluate the models, unseen data is required. Thus, the dataset may be split in train and test sets. Among the methods to do that there is the random sampling, which randomly selects instances in the dataset for the two sets, in proportions of 80% and 20%, respectively (70% and 30% is also used) (KOTU; DESHPANDE, 2019).

Finally, the pipeline outputs the trained models and the test set for performance evaluation.

When dealing with supervised learning applications, one can face some difficulties to get good results (ALZUBAIDI et al., 2021; KORNILOVA; BERNARDI, 2021). One is the *overfitting*. It occurs when the models are too specialized in the train data and they achieve a poor prediction quality when evaluated in the test set (KARYSTINOS; PADOS, 2000). According to Hawkins (2004), a model is overfitted when it achieves the same prediction quality when compared to a simpler one. Thus, the model is more complex than it should be. A possible adjustment to reduce overfitting is to reduce the complexity of the models, that is, check whether simpler models perform as well as the complex ones (LIU, R.; GILLIES, 2016).

A further common challenge that can affect the learning task is the presence of outliers in the dataset. There may be instances very distinct or inconsistent from the others. These instances are called *outliers*. Depending on the ML technique, their existence in the dataset may degrade the prediction quality of the models (FREEMAN et al., 1995). Among the existing algorithms for discovering outliers (HODGE; AUSTIN, 2004), there is the Isolation Forest. It is a simple and efficient technique that isolates the anomalies at the upper levels of random trees (LIU, F. T. et al., 2008).

The method used to split the dataset into training and test subsets can introduce some bias in the pipeline. The distribution of the examples may not be similar in those two subsets, specially in small datasets (HAWKINS, 2004). By evaluating the models several times using different and random train and test sets the prediction quality measurements could be more precise. In this case, k-fold cross-validation can be used, which splits the dataset in k subsets. One fold is used to test the models and the remaining for training. In k steps, folds are alternated. The average of the metrics in the test set is our final performance measure for this model (KUHN; JOHNSON, 2013).

2.2 TEXT REPRESENTATION

To apply certain ML techniques to textual data, some steps are required, as discussed in Section 2.1, and one of them is the text representation. The goal is to transform the text, i.e. characters, words, phrases, etc., into a numerical representation which is compatible with ML techniques (KOWSARI et al., 2019). In this work, the

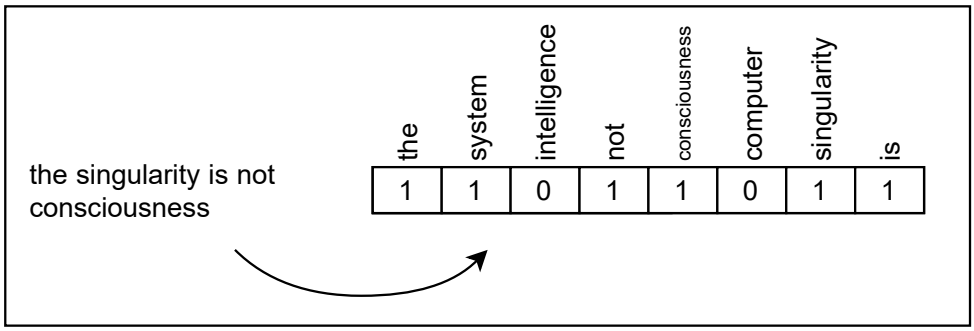
focus is on the representation of texts applied to supervised learning tasks, such as classification and regression.

2.2.1 Bag of Words

One of the many techniques to represent text in a structured format is the Vector Space Model (VSM), where each document is represented through a numerical vector. This representation can be created using the Bag of Words (BOW) where the vector values indicate some information on the words, such as frequency or its existence or not in the text, generating sparse and high-dimensional representation (AGGARWAL; ZHAI, 2012).

As an simplified example, Figure 2 show a sample text and its representation using the BOW with Term Frequency (TF) values, that is the frequency of each word in the text. Beyond TF values, BOW values can be binary representing whether such word exist in the text or not. There is also the Term Frequency-Inverse Document Frequency (TF-IDF), which reduces the weight of words appearing in many documents (AGGARWAL, 2018).

Figure 2 – Bag of Words example



From Figure 2, one may notice, beyond the count of the words from the sample text, there are other words such as *computer*, which did not appear in that text. This happens because the BOW creates a position for each word written in the documents. And, as words may appear in some documents and not in others, the representation will not be completely filled, generating sparse and high-dimensional representations. The vocabulary of a set of documents may have thousands of words, while a single document may have only hundreds of words (WITTEN et al., 2011). Furthermore, sparse and high dimensional representations may degrade the performance of ML models and some steps may be taken to improve the representation, such as text preprocessing, feature selection and dimensionality reduction.

By applying text preprocessing, one can clean the text and reduce the vocabulary size based on the following NLP techniques:

- *Normalization:* converting all characters inside the document to lowercase (JURAFSKY; MARTIN, 2019).
- *Tokenization:* grouping characters in a string into meaningful pieces. This technique identifies important components or tokens using a set of delimiters, such as space and punctuation (LEE, M. L. et al., 1999).
- *Stemming:* reducing a word variant to its stem by removing any attached suffixes and prefixes (affixes). The stem does not need to be an existing word in the dictionary, but all its variants should map to this form after the stemming has been completed.
- *Filtering:* removing such terms that do not convey specific meaning (stop words), such as articles, conjunctions, pronouns and prepositions (KOTU; DESHPANDE, 2019).

The problem of dimensionality in BOW can be reduced by using dimensionality reduction techniques, such as feature selection and feature projection. The use of feature selection techniques can improve the text representation and the prediction quality, as discussed in Chandrashekar and Sahin (2014) and Guyon and Elisseeff (2003). Given a set of documents, each of which having n features and the labels for each document, the techniques will define the k most important features to predict the labels. Examples of feature selection techniques include Mutual Information (MI) and Recursive Feature Elimination (RFE) (GUYON; ELISSEEFF, 2003).

Feature projection techniques can also be used in BOW representations. While feature selection techniques select a set of features to represent the data, feature projection techniques transform the data in a high dimensional space to a lower dimensional space, while keeping its characteristics. Examples of techniques for feature projection include Principal Component Analysis (PCA), Non-negative Matrix Factorization (NMF), and auto-encoders (JOLLIFFE; CADIMA, 2016; GUYON; ELISSEEFF, 2003; MENG et al., 2018).

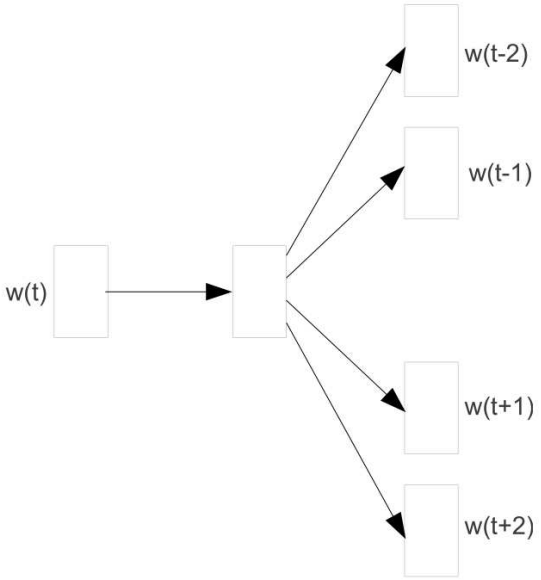
Beyond sparsity and high dimensionality, BOW models may not have the sense of order in which the words are written, as it only considers quantitative information from them, and can not detect syntactic and semantic information (KOWSARI et al., 2019). Thus, an additional preprocessing step to improve the representation is the extraction of N-grams. Such process consists on grouping words that generally appear together (KOTU; DESHPANDE, 2019). Examples of N-grams include *text mining*, *supreme court*, *special civel court*. One can define the maximum length of N-Grams, however, the higher the length, the bigger the dimensionality of the representation.

2.2.2 Word embeddings

Word embeddings, also known as distributed word representations, can capture both the semantic and syntactic information of words while representing them as n-dimensional dense vectors. Thus, unlike BOW, word embeddings representations are less likely to have the problems of high-dimensionality and sparsity (LAI et al., 2016). These representations are generated based on a training process applied to a large unlabeled corpus.

A relevant algorithm is Word2Vec, which uses a neural network to learn to associate words in phrases, and can be divided into two approaches: Skipgram (SG) and Continuous Bag of Words (CBOW). In Word2Vec SG for instance, the representations are trained as it tries to predict surrounding words in a phrase $\mathrm{w(t-2)}, ..., \mathrm{w(t+2)}$, based on the current word $\mathrm{w(t)}$ (MIKOLOV et al., 2013), as shown in Figure 3.

Figure 3 – Word2Vec SG architecture



Source – Mikolov et al. (2013).

Although Word2Vec can satisfactorily capture the semantics of words, it will produce distinct representations for the possible variations of the same word. For instance, for the verb *do*, it produces *did*, and *done*. And rare words may not be represented when not contained in the corpus used for training. Thus, FastText, an algorithm based on Word2Vec, produces representations of words based on the character N-grams, that is, it trains representations for combinations of characters, which then produces the word embeddings (BOJANOWSKI et al., 2017).

Another technique for word embeddings generation is GloVe, which creates a co-occurrence matrix containing the frequencies of words in different contexts. Then it

applies a dimensionality reduction technique to produce the final representations (PENNINGTON et al., 2014).

In terms of applications, word embeddings have been used in many tasks (See Appendix B). These tasks include clustering (MOHAMMED et al., 2020), text classification (AGGARWAL, 2018) , text summarizing (ALAMI et al., 2019), and others.

Finally, there is an distinction regarding terms for word embeddings. When the embeddings are used as input for text representation in a pipeline, they are called *pre-trained* representations. The process of creating the representations has been done before in a unlabeled corpus. However, when such process of training the representation is finished, we call them *trained* representations, as they are the outputs of the pipeline.

2.2.3 Other representation techniques

Beyond BOW, and word embeddings, there are techniques for text representation proposed recently which can represent semantics, context, order of the text. One example is the BERT, a language model that learns to represent the text in the two directions, that is, from left to right, and vice-versa. The process of training is composed of two steps: pre-training and fine-tuning. The former, the language model is trained over an unlabeled corpus, and in the latter, the representation is fine-tuned for the ML task (DEVLIN et al., 2018).

There is also Embeddings from Language Model (ELMo), a deep contextualized word representation that models both the syntax and semantics, and how they vary across linguistic contexts. These word vectors are learned functions of the internal states of a deep bidirectional language model, which is pre-trained on a large unlabeled corpus. They can be easily added to existing models and be applied to several tasks, including question answering, textual entailment and sentiment analysis (PETERS et al., 2018).

Finally, the current SOTA in text representation is the Generative Pre-trained Transformer-3 (GPT-3), a language model composed of over 175 billion parameters and trained using the openly data available in the internet. GPT-3 can generate many kinds of texts, such as novels and programming code, and it can be used in tasks like translation and text summarizing (BROWN et al., 2020).

2.3 TEXT CLASSIFICATION

Text classification is an important part of TM and it has been applied in many contexts (CARDOSO et al., 2018; KUMAR et al., 2015; SHEIKHALISHAHI et al., 2019). In the classification task, we use data to construct a model that learns to relate its features (input) to one of the available labels (output). For a given test instance for which the class is unknown, the trained model is used to predict a class label for this

instance (AGGARWAL; ZHAI, 2012).

According to Kowsari et al. (2019), the text classification can follow a pipeline composed of text pre-processing, feature extraction (or text representation), dimensionality reduction (optional), classification models training, and evaluation, as shown in Figure 2. However, in the Figure there is a step not explicitly mentioned by the author, the train test split.

Techniques for classification can be divided in classical ML and DL, following the description from Section 2.1. Among classical techniques for classification available, there is the Decision Tree (DT), which are built based on the induction task in train data, beginning with the root of the tree and proceeding down to its leaves. With the constructed tree, one can make predictions by crossing the tree according to the features from the test data (QUINLAN, 1986). Other models have been proposed based on DT, such as ensemble methods of Random Forest (RF), Gradient Boosting (GB), Bagging (BG), and AdaBoost (AB). An ensemble method tries to improve the performance by training multiple models using distinct approaches according to the techniques mentioned (BREIMAN, 2001; FRIEDMAN, 2001; BREIMAN, 1996; SCHAPIRE, 1999).

There is also the Naïve Bayes (NB) classifier, a computationally inexpensive technique that needs a very low amount of memory. It is based on the Bayes theorem and on the assumption that the attributes are conditionally independent, given the label (KOWSARI et al., 2019; PEARSON, 1925).

Logistic Regression (LR) is another technique for classification that tries to predict probabilities of the labels (MORGAN; TEACHMAN, 1988). There is also Support Vector Machine (SVM), which tries to find a hyperplane in an n-dimensional space, where n is the number of features, that distinctly classifies the data points. (CORTES; VAPNIK, 1995).

As the last classical ML technique cited in this work, there is feed-forward Neural Networks (NN) that consists of simple, connected units called neurons, each producing activations. Input neurons are activated base on the input data, other neurons get activated through weighted connections from previously active neurons. The output neurons may influence the environment (SCHMIDHUBER, 2015).

DL techniques, deep neural architectures based on NN, have proven to be effective in text classification. Among the techniques available there is Convolutional Neural Network (CNN), which uses convolutional masks to sequentially convolve over the data. For texts, a simple mechanism is to recursively convolve the nearby lower-level vectors in the sequence to compose higher-level vectors. Similar to images, such convolution can naturally represent different levels of semantics shown by the text data (PENG et al., 2018).

Beyond CNN there are sequence based DL techniques known as Recurrent Neural Network (RNN), which are networks that make predictions based on the previous

states and current input. One relevant kind of RNN is the Long-Short Term Memory (LSTM). Furthermore, the sequence may be one-directional or bi-directional, forming Bi-RNN and Bi-LSTM (KOWSARI et al., 2019; HOCHREITER; SCHMIDHUBER, 1997).

A recent improvement for sequence based DL techniques is the attention mechanism. The RNN has the problem of vanish gradient, where it may *forget* what it has learned or seen in the sequence. Thus, attention mechanisms allow the network to focus on distinct parts of the sequence (SCHMIDHUBER, 2015). Based on attention mechanisms and in the encoder-decoder architecture, a new type of neural network emerged, i.e., the transformer network. It achieved SOTA results in several NLP tasks (VASWANI et al., 2017).

To evaluate the performance of classification models, the test set is introduced to the models, in order to make predictions on unseen data. Then, the predicted outputs are compared to the actual outputs using the evaluation metrics, which may vary for binary and multi-label classifications. To calculate the metrics for classification one can use the confusion matrix, describing the predicted labels in the lines and actual labels in the columns (KOWSARI et al., 2019). Table 1 shows an example of confusion matrix for a classification task with three labels.

Table 1 – Confusion matrix example

	Label 1	**Label 2**	**Label 3**
Label 1	**51**	5	1
Label 2	0	**10**	3
Label 3	5	5	**60**

From Table 1, the example shows the classifier predicted 51 documents as Label 1 where the actual label was Label 1. It also predicted three examples as Label 2, where the actual label was Label 3, and so on.

Based on the confusion matrix, one can detect the True Positive (TP), False Positive (FP), TP, True Negative (TN) and False Negative (FN), as detailed in Kowsari et al. (2019) and Lever et al. (2016), and calculate the evaluation metrics, as follows.

A simple metric is *accuracy* that indicates the fraction of test instances in which the predicted label matches the actual label, i.e., the sum of the main diagonal (TP + TN) in the confusion matrix divided by the number of predictions, following Equation 1 (LEVER et al., 2016).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Another metric is *precision*, that indicates the percentage of instances predicted to belong to the positive label (TP and FP) that was correct (TP), following Equation 2.

Precision is calculated for each label, having the label to evaluate as the positive label and the others, the negative (LEVER et al., 2016).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall is the percentage of ground-truth positives (TP and FN) that have been recommended as positives (TP), following Equation 3. Recall is also calculated for each label, having the label to evaluate as the positive label and the others, the negative (LEVER et al., 2016).

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

Finally, there is *F1-Score* is the harmonic mean between the precision and the recall, following Equation 4.

$$\text{F1} - \text{Score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{4}$$

In a multi-label classification, i.e., when there are more than two labels, the calculus of precision, recall and F1-Score is slightly different. There are three types of calculation: micro, macro and weighted. The first sums the TP, FP, FN for all the labels, and them calculates the metrics. The second calculates the metrics individually for each label and takes the average of those values. The third uses the proportion of examples from each label to calculate an weighted average (UYSAL, 2016).

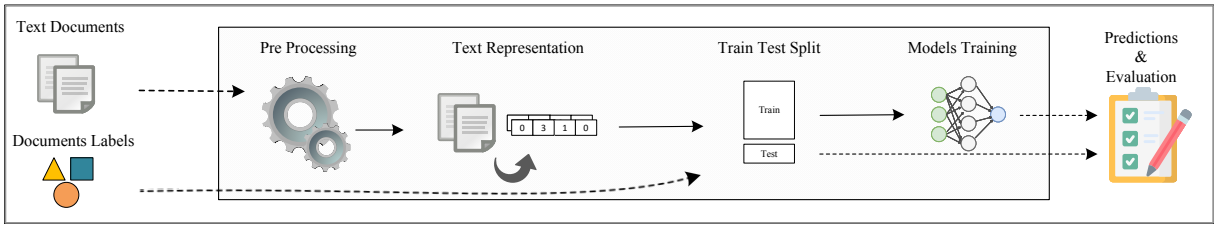
2.4 TEXT REGRESSION

The regression task is a supervised learning approach that, based on samples of pairs (x, y), aims to find a function f that predicts a continuous dependent variable y from x ($\mathrm{y} = \mathrm{f}(\mathrm{x})$). Since $\mathrm{f}(\mathrm{x})$ may not achieve a perfect mapping from x to y, there will be some amount of error, which we want to keep as small as possible (DRAPER; SMITH, 1998). The representations of x can vary in each context. When applying regression on textual data, x can be books, legal documents, etc. (AGGARWAL, 2018).

Based on Kowsari et al. (2019) and the related work for regression, one can use texts in a regression task to predict one or more dependent variables using the pipeline from Figure 4. Due to the unstructured nature of textual data, some specific steps to apply machine learning in texts are necessary, as we discuss in the following paragraphs (AGGARWAL, 2018).

The pipeline from Figure 4 receives as inputs the textual documents and their labels. To prepare this data to further use, some preprocessing operations were applied (GARCIA et al., 2015), which includes tokenization, normalization, filtering, and others (LEE, M. L. et al., 1999; JURAFSKY; MARTIN, 2019; KOTU; DESHPANDE, 2019).

Figure 4 – Simple regression pipeline



The next step is to transform the text into a numerical representation which will serve as inputs to regression models. With the numerical representations of the text and their labels, the data can be split into two new datasets: train and test. Models are trained using the train set and the regression techniques. Using these models, one can make predictions on some continuous output (KOTU; DESHPANDE, 2019).

Among regression techniques available, there are linear based techniques such as Linear Regression (HASTIE, 2009) and its derivatives, Ridge (HOERL; KENNARD, 1970), Elastic Net (ZOU, H.; HASTIE, 2005) and Lasso (TIBSHIRANI, 1996). And techniques based on DT (BREIMAN et al., 2017) can be used, such as RF (BREIMAN, 2001), Gradient Boosting (FRIEDMAN, 2001), Bagging (BREIMAN, 1996), AdaBoost (SCHAPIRE, 1999), and XGBoosting (CHEN, T.; GUESTRIN, 2016). Beyond linear and tree-based models, Support Vector Machine (DRUCKER et al., 1997) and feed-forward Neural Networks (KINGMA; BA, 2015) can be adapted for the regression task.

Due to the inner differences among the regression techniques, they can achieve better or worse performances in different situations. Thus, it may be useful to apply some of those models together, so they complement one another. The final prediction of this combination is the average output among the models. This approach is called Ensemble Voting (MENDES-MOREIRA et al., 2012).

Focusing again Figure 4, at the final step, the estimation of the prediction quality of the models on the test set is carried out using metrics for the regression. A common metric is Root Mean Square Error (RMSE), which represents the average error of the square differences between the predicted (y_i) and the actual ($\hat{y}_i$) values (AGGARWAL, 2018), as shown in Equation 5. This metric is sensitive to outliers and tend to penalize bigger errors, i.e., as the RMSE applies a quadratic function on the error, bigger errors have more impact in the final metric (CHAI; DRAXLER, 2014).

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \tag{5}$$

Another metric is the Mean Absolute Error (MAE), which represents the average of the errors when predicting the dependent variable. MAE is also simple to interpret and it is less sensible to outliers than RMSE (CHAI; DRAXLER, 2014).

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{6}$$

An additional metric is the coefficient of determination, or R^2, interpreted, as shown in Equation 7, as the proportion of observed variation in y that can be explained by the regression model. So, the higher R^2, the better the model can explain the variation in y (DEVORE, 2011). $\bar{y}$ represent the mean of the y values.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{7}$$

Acceptable values of error and coefficient of determination may depend on the domain's requirements and the range of possible values for the dependent variable (ZHOU, Q.; CHEN, Y., 2011; TORRES et al., 2005).

3 RELATED WORK

This chapter presents the related work relevant to this research. Part of them came from the three SRL, but we included others to complement the chapter's purpose. Each section in this chapter focus on describing the relevant work regarding each part of the research question.

3.1 TEXT REPRESENTATION

Nathan Hartmann et al. (2017) evaluated different word embeddings models trained on a sizable Portuguese corpus (1,395,926,282 tokens in total). They trained thirty-one word embeddings models using FastText, GloVe, Wang2Vec, and Word2Vec, and evaluated them intrinsically on syntactic and semantic analogies and extrinsically on POS tagging and sentence semantic similarity tasks. The results obtained from intrinsic and extrinsic evaluations were not aligned, as opposed to what they expected. GloVe produced the best results for syntactic and semantic analogies, and the worst, together with FastText, for both POS tagging and sentence similarity.

Rodrigues et al. (2020) evaluated different word representation models on semantic similarity tasks, trained on a Portuguese corpus provided by a workshop (10,000 sentences). They used word embeddings (Word2Vec and FastText) and deep neural language models (ELMo and BERT). The results indicated that the ELMo language model was able to achieve better accuracy than any other pre-trained model which has been made publicly available for the Portuguese language. They also demonstrate that FastText Skipgram embeddings can have a significantly better performance on semantic similarity tasks.

Chalkidis and Kampas (2018) trained a word embeddings model on a large legal corpus from various public legal sources in English (UK legislation, European legislation, Canadian legislation, Australian legislation, English-translated legislation from EU countries, English-translated legislation from Japanese, US Supreme Court decisions, and US Code). The corpus sums up to a total of approximately 492,000,000 tokens. They trained word embeddings based on the Word2Vec Skipgram model, rather than the most recent FastText. They justified that Word2Vec provided better semantic representation than FastText, which tends to be highly biased towards syntactic information.

Finally, Smywiński-Pohl et al. (2019) trained word embeddings models (Word2Vec and GloVe) to find out which is best suited for establishing the correspondence between Polish legal and extra-legal terminology. The corpora are composed of text data collected from two databases: a) National Corpus of Polish, which includes texts of different genres, such as novels, transcripts of parliamentary speeches, and newspaper articles, which sums up to a total of 2,591,817,208 tokens; b) judgments from Polish

Supreme Court, Polish Constitutional Tribunal, Polish common courts, Polish National Chamber of Appeal and Polish administrative courts, which sums up to a total of 4,076,628,858 tokens. The results showed the superiority of the Word2Vec CBOW negative sampling variant in their problem.

3.2 TEXT CLASSIFICATION

In the international context, El Jelali et al. (2015) conducted the experiment at the Italian Ministry of Justice - the Justice Relationship Management (eJRM), stands out. It provides support to lay citizens, students, and mediators to obtain legal solutions on a given case, stands out. The system was powered by court decisions retrieved at the moment the natural language case is presented. Its architecture consists of four steps:

1. Indexing: intended to store court decisions in a database;
2. Core mining: trains a classification model to predict the legal field to which a given case description belongs;
3. Query processing: extracts the relevant terms from the query that will be used by core mining to predict the legal field of the disputed text;
4. Ranking: retrieves and classifies relevant court decisions (belonging to the foreseen legal field) to be presented to the disputers and mediators.

The following classification algorithms were used: Naïve Bayes, Decision Tree, Linear SVM and with Gaussian Kernel, all with and without PCA to reduce the dimensionality of the text. The maximum accuracy obtained was 91.3%, using the Linear SVM with PCA.

Aletras et al. (2016) conducted an experiment at the European Court of Human Rights, aiming at predicting decisions. The requests processed in it deal with violations, by a Member State, of the civil and political rights established in the European Convention on Human Rights. Based on the textual evidence extracted from the case, the objective was to predict whether a specific article of the Convention was violated. Such textual evidence comprises specific parts referring to *fact*, *applicable law* and the *arguments* presented by the parties involved. It is a binary classification having as input the textual data extracted from the cases and, as output, their actual judgment on the violation or not of a certain article of the Convention. A database was formed from decisions related to three articles of the Convention. The results indicated that the *facts* section of a case is the most important predictive factor, that is, judicial decision-making is significantly affected by the stimulus of the narrated facts. The Linear SVM classification algorithm was used, obtaining the maximum accuracy of 79%.

Şulea et al. (2017) carried out the experiment at the French Supreme Court to predicting cases ruling and the law area. They also investigate the influence of time period of case's ruling over the textual form. As dataset, a collection of rulings from

the French Supreme Court judged between the 1880s and 2000s, containing more the 126 hundred thousand documents and their metadata. To represent the text, they used BOW with N-Grams of length two and three, followed by the selection of the k most relevant features according to the labels of each problem. As classifier, SVM with linear kernel is applied to the three problems. In terms of results, the prediction of law area achieved an accuracy of 90.2% and F1-Score of 0.90 using eight labels. The prediction of case ruling achieved 96.9% and 0.97 of accuracy and F1-Score, respectively, for the classification using six labels. Finally, the temporal text classification with seven labels and Bi-Grams achieved 74.3% of accuracy and F1-Score of 0.732.

Maat et al. (2010) applied ML and knowledge-based techniques to classify judgments Dutch legislation. Initially, they applied a set of preprocessing techniques such as, stemming, conversion to lower case, and stopwords removal. To represent the legal cases numerically, BOW is used with both binary, TF, and TF-IDF values. As classification techniques, there were SVM and a Knowledge Based classifier based on patterns from each of the thirteen labels. To split the dataset in train and test sets, the Leave-One-Out approach. In terms of results, the best setup with SVM and binary BOW achieved an accuracy of 94.69%, and F1-Score of 0.947. The best knowledge-based system achieved accuracy of 94.37%.

Finally, in the national context, the experiment carried out by Nilton Silva et al. (2018) and Nilton Silva (2018) at the STF stands out, conducted by the VICTOR project, from the University of Brasilia (UnB), currently in progress, which aims to solve a problem of pattern recognition in texts. of the judicial processes that enter there. Specifically, the problem to be solved is the classification (binding) of processes into topics of General Repercussion (GR). The GR is a procedural instrument that acts as an *appeal filter*, allowing the STF to select the resources that it will analyze according to criteria of legal, political, social or economic relevance. The project consisted of two stages:

1. Classification of five types of parts within a process (secondary goal): judgment, extraordinary appeal (RE), extraordinary appeal (ARE), order and sentence;
2. Classification of themes with general repercussion (main goal): after the classification of pieces, the phase of classification of themes begins, counting on the automation of the segmentation of the five types of important pieces for the identification of GR themes.

The main goal is to relate an entire process to one or more GR themes. The maximum accuracy of 90.35% was obtained in the secondary goal, using CNN as classification algorithm.

3.3 TEXT REGRESSION

Joshi et al. (2010) used text regression to predict a movie's opening weekend revenue. They collected data for movies released in 2005–2009. For these movies, they obtained metadata and a list of hyperlinks to movie reviews by MetaCritic, and each movie's production budget, opening weekend gross revenue, and the number of screens on which it played during its opening weekend from The Numbers. They applied linear regression combined with N-Grams. The results revealed that review text can replace metadata and even improve the prediction quality.

Lampos et al. (2014) used text regression to predict a user impact score, estimated by combining the numbers of the user's followers, followees and listings. They formed a Twitter dataset of more than forty eight million tweets produced by 38,020 users located in UK in the period between April 14, 2011 and April 12, 2012. They applied linear as well as nonlinear learning method (Gaussian Process). The results generated strong predictions, especially with models based on the Gaussian Process, and showed that activity, oriented interactivity and engagement on a diverse set of topics are among the most decisive impact factor.

Trusov et al. (2016) used text regression to predict next year change in stock price volatility in the context of financial risk problem. They collected data from traded companies reports provided by the EDGAR system, maintained by the U.S. Securities and Exchange Commission (SEC), and stock prices via Yahoo Finance. They applied Support Vector Regression and Random Forest models associated to BOW representation with Latent Dirichlet Allocation (LDA) and TF-IDF. The results showed that models with multiple representations outperform single representation models.

Bin Zou et al. (2016) used text regression to detect and quantify infectious intestinal diseases (IIDs) from social media content. They collected Twitter data and social health surveillance records obtained from Public Health England (PHE), and applied a regularized linear (Elastic Net) as well as a nonlinear (Gaussian Process) regression function for inference. The results indicated that both in terms of prediction quality and semantic interpretation, Twitter data contain a signal that could be strong enough to complement conventional methods for IID surveillance. Regarding text regression, the nonlinear approach performs better.

Kusmierczyk and Nørvåg (2016) used text regression to predict nutritional fact values of an unknown recipe within the context of dietary pattern analysis in food-focused social networks. They collected data from the largest English online food recipe platform, namely *allrecipes.com*. Each recipe has title and information about nutritional facts (per 100 g). They applied LDA with linear regression and with Gradient Boosted regression trees. The experiments showed the extent to which it is possible to predict nutrient facts from meal name.

Finally, Xu and Chieh Lee (2020) used text regression to analyze online consumer

reviews and managerial responses from the hotel industry. They collected online consumer reviews about the well-known Marriot hotel chain from three platforms, namely Expedia, representing third-party booking platforms; TripAdvisor, representing social-media platforms; and Marriot's official booking platform, representing direct platforms (channels). They applied multinomial logistic regression combined with Latent Semantic Analysis (LSA) and TF-IDF. The results suggested that although consumers have different linguistic styles and focus on different attributes in their reviews on the three platforms, the antecedents of their overall satisfaction are the same: room, employees and services, location and access, and operations and facilities. Moreover, managers differentiate between consumers' perceptions in their review process and their perceptions about the consumption experience. Based on these results, they made recommendations for managers to provide suitable responses to the different platforms online and to improve consumer overall satisfaction.

3.4 CONCLUSIONS FOR THE CHAPTER

In the following paragraphs, we discuss how this research differs from the presented related work.

Regarding the representation task, we focus on the representation of legal texts using a neural-based representation called word embeddings. However, the related work found in the SRL from Appendix B does not deal with the training and the use of word embeddings on legal texts in Portuguese, either Brazilian as European. The researcher divided the SRL in two, one related to the application of word embeddings in Portuguese general texts and the other related to word embeddings on legal texts from any language. Thus, this research explores in depth the training and application of word embeddings in legal texts in Portuguese.

Concerning the classification task, we focus on predicting the results of the legal cases from JEC. That is, to assign one of four possible labels to the texts. From the SRL presented in Appendix A, the classification task has been explored in the literature in legal texts from several languages (including Portuguese), which narrows the possible contributions from this research. Thus, it proposes the application and comparison of DL and ML techniques on the classification of legal cases in Portuguese. Compared to the related work, this research explores a broader range of techniques to the same dataset, from SVM to Bidirectional Long-Short Term Memory (Bi-LSTM) with Self Attention.

Lastly, about the text regression task, we focus on the prediction of the compensation value for immaterial damage. Based on the SRL from Appendix C, we learned that such task has never been employed to legal documents in any language. However, one need to note the scope of this research, where the main input of the text regression pipeline is the legal cases' text, not tabular features. Taking into account the absence of

papers on text regression in the legal domain, we listed several works regarding the tasks in many contexts. This research considers a broader variety of TM and regression techniques on the prediction of the compensation and it investigates the impact on the performance of having some of these techniques in the text regression task.

4 EXPERIMENTS, RESULTS AND DISCUSSION

In this chapter, we describe the steps followed to answer the research question and its three parts.[1] The questions are answered based on three distinct experiments, regarding text representation, text classification and text regression, detailed in Section 4.1, 4.2, and 4.3, respectively.

The datasets used in each section consider the same type of data, that is, legal judgments from the JEC at UFSC. Those judgments involve failures on air transport services. However, the amount of data in each section changes as we run the experiments and performed the data collections at JEC, during the last two years.

4.1 TEXT REPRESENTATION IN LEGAL JUDGMENTS

The experiments and results presented in this section are related to text representation. First, there is the clarification of the experiments' purpose, and how they answer the research question. Then, it presents the details on the datasets, the pipelines, the results, and the discussion.[2]

4.1.1 Experiment's Purpose

The experiments' purpose is to answer the first part of the research question, i.e., whether the size and specificity of the corpus used for word embedding training impact the performance in a classification task. Thus, the experiments from this section focus on the aspects of training and testing word embeddings representations. The first step is the datasets construction, where we collected two types of data: unlabeled corpora for training embeddings and labeled dataset for classification. Significant part of those datasets were not available for use in any dataset platform, such as Kaggle, and had to be collected and prepared, producing a considerable amount work. Also, a legal expert manually labeled each of the legal judgments from the second dataset.

To answer the research question the unlabeled corpora is used to train several word embeddings representations, while varying corpora sizes and degrees of specificity (also called *context*). To do so, the corpora is divided in three, according to specificity: corpus related to all subjects, corpus related to all legal subjects and corpus related to air transport services and the legal domain. After that, each of those three corpora is divided is several subsets to form other corpora with distinct corpus size.

[1] This chapter is adapted from the papers Sabo et al. (2019), Dal Pont et al. (2020), and Sabo et al. (2021), with substantial degree of overlap. Some sections from those papers are identical to parts of this dissertation, and others are substantially the same as parts, but in different words. To avoid over-quoting, we offer this note as a general citation and do not provide further citations to these articles in the body of the chapter.

[2] The code for training and evaluating embeddings is available at `https://github.com/thiagordp/embeddings_in_law_paper`.

For each of the smaller corpora, we trained word embeddings models using GloVe technique, as we detailed later. Finally, we apply each resulting embeddings model as representation in the classification task for evaluation.

The tools for the experiments in this section embraced the collection and the preprocessing of the data, as well as the embeddings training and the text classification. To collect most of the unlabeled corpora, we used Selenium (SALUNKE, 2014), while the text prepreprocessing involved the Natural Language Toolkit (NLTK) (LOPER; BIRD, 2002). In the classification task, we applied the Keras Framework optimized to run on Graphics Processing Unit (GPU). As programming language, we adopted Python to build the experimental setup.

Finally, another aim of these experiments is to build and make available pre-trained representations which can be used by other researchers in their own TM experiments with legal documents in Portuguese.

4.1.2 Dataset

The first dataset is a collection of legal documents from the courts web portals, which enable us to evaluate the specificity influence of these legal corpora. We divided the dataset into two contexts: related to general legal texts and related to air transport services text.

Another dataset is a collection of texts from other general topics (not related to legal domains) that are already compiled and freely available. Having the corpora for legal and miscellaneous contexts, we applied some processing steps to remove noise from texts. To evaluate the influence of corpus size in embeddings training, we divided these three corpora into smaller pieces based on word count.

To train the embeddings it is required large text corpora to be able to get good embeddings. However, in the Brazilian Portuguese language, we could not find any dataset available on the Internet containing enough legal text corpora for our purposes. Thus, we had to build our legal corpora.

Our main sources of legal text are Brazilian courts platforms. We collected judgments from the webpages of STF, STJ and TJ-SC (STF, 2020; STJ, 2020; TJSC, 2020). We also collected judgments from the JusBrasil portal containing processes related only to failures on air transport service from all State Court (TJ) from Brazil (JUSBRASIL, 2020). We choose those courts based on the hierarchy of the Brazilian Judiciary presented in Figure 5.

Considering that the documents used in the task classification came from the JEC at UFSC, we follow the structure starting from JEC at the lower level until the STF at the top. As the higher courts judge the appeals of lower courts, the legal judgments' subjects from the higher courts may have a degree of similarity to the judgments from JEC.

Figure 5 – Brazilian Judiciary hierarchical structure

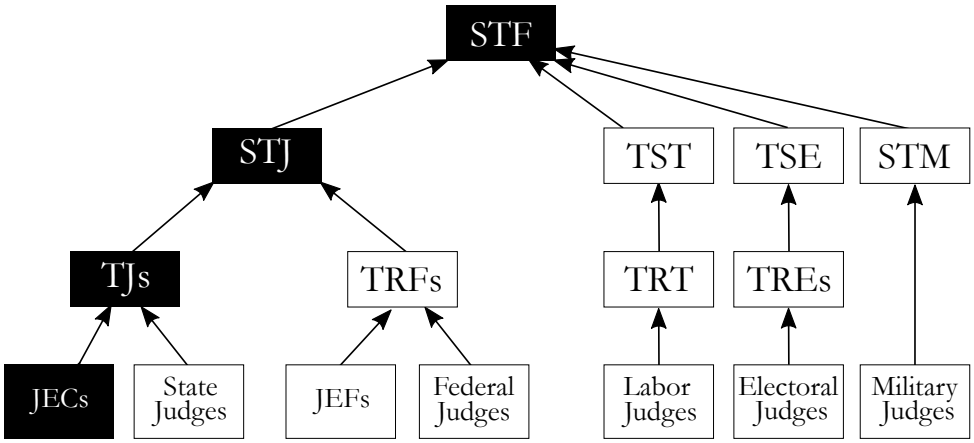


Table 2 shows the number of processes acquired and word count for each court.

Table 2 – Acquired legal judgments from courts for embeddings training

Source	Collegial Judgments	Individual Judgments	Subtotal	Word Count
STF	64,779	118,910	183,689	294,937,185
STJ	101,141	0	101,141	312,687,450
TJ-SC	989,964	662,535	1,652,499	3,060,212,814
TJs (JusBrasil)	34,239	0	34,239	78,138,337
		TOTAL	1,971,568	3,745,975,786

After downloading all the legal judgments, most of them in Portable Document Format (PDF) and Rich Text Format (RTF), we extracted raw texts from these files. We did not apply Optical Character Recognition (OCR) in scanned PDF documents, due to time limits to finish the experiments, so only digital PDF and RTF files were accounted in Table 2.

With the extracted texts, we applied some preprocessing steps, as discussed further in this section.

Then we built the legal text corpora containing all the judgments related to all law subjects, which we call *general* legal context corpora in this work. Using this base, we created another text corpora whose legal judgments are related only to air transport and consumer law, and we call it *air transport* context corpora.

To be able to compare how good embeddings trained with legal texts perform against those created with all kinds of texts, we also created other corpora from a variety of sources. Thus, we searched for free available textual datasets. In this work, we call these texts as *global* context corpora. Table 3 shows all the global text datasets used. Then we apply some preprocessing steps, as will be described further in this section.

The dataset for evaluation in the classification task is composed of almost one

Table 3 – Global context corpora for embeddings training

Dataset	Documents	Word Count	Source
Wikipedia in Portuguese	1,014,713	303,622,360	Wikipedia (2019)
Brazilian Literature Books	169	37,848,783	Tatman (2017)
Old Newspapers	617,627	26,441,581	Liling Tan (2020)
Folha de São Paulo News	165,641	74,594,367	Marlessonn (2019)
HC News Corpus	494,128	27,170,063	Christensen (2016)
Blogspot Posts	2,181,073	696,657,915	Santos et al. (2018)
Wikihow Instructions	786,283	22,471,312	Chocron and Pareti (2018)
TOTAL	5,259,634	1,188,806,381	

thousand legal judgments from the JEC at UFSC, issued between January 2014 to November 2019, and collected manually by a legal expert on site. The specific subject debated is failures in air transport services, such as flight delay, flight cancellation, baggage loss. Then the consumer injured claims compensation for immaterial damage, which is a monetary value fixed by the judge.

Compensation for immaterial damage is usually monetary. It is not possible to evaluate the painful sensation experienced by the injured person. As a mean of mitigating the consequences, money can play a satisfactory role (DINIZ, 2020). There are some circumstances considered by the judge when fixing the value, such as the person's age, health status, person's gender, place and time of injury. Anyway, these variables are weighted by the judge in a free assessment, according to his/her interpretation of each situation (SADIKU, 2020).

A legal judgment is an unstructured textual document and refers to the final decision of a lawsuit in first degree. Generally, it consists of three elements (BRAZIL, 2015):

- *Report* (summary of what happened according to the parties allegations and evidences);
- *Reasoning* (reasons that formed the judge's conviction);
- *Result* (value fixed by the judge for immaterial damage compensation).

There are four possible labels for the legal judgments:

- *Well-founded:* The consumer wins the lawsuit, 26% of the dataset.
- *Not founded:* The consumer loses the lawsuit, 10% of the dataset.
- *Partly founded:* The consumer wins part of the lawsuit (for example, when he/she plead a greater compensation than the assigned value by the judge), 62% of the dataset.

- *Dismissed without prejudice:* The consumer makes a procedural error (for example, when he/she indicates as a defendant the wrong airline company). So the consumer can file a new lawsuit, 2% of the dataset.

In the experiments from this section, to evaluate the representations using text classification, we removed the judgments' result part. Thus, to make predictions the ML techniques will have the summary of the facts and the legal reasoning, that is the same data available at the early stages of a real case. Applying this approach, we could execute experiments while overcoming the difficulties of acquiring data from the early stages of lawsuits.

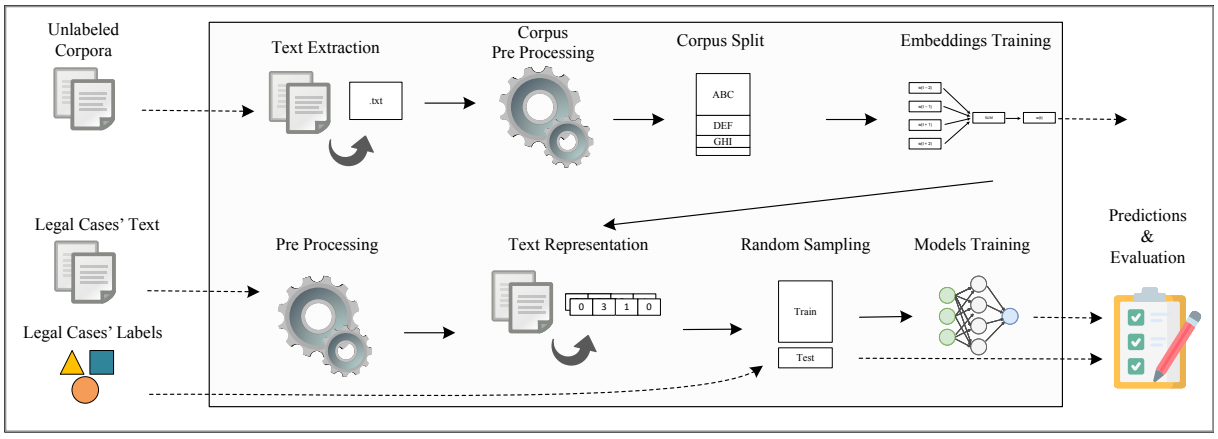
4.1.3 Pipeline

This section describes the pipelines and experimental setup for the experiments of embeddings training and evaluation.

The first part of the experiments consisted on training the word embeddings models based on unlabeled corpora. The second part of the experiments involved the evaluation of the embeddings in a text classification task of the legal judgments from JEC, using the trained embeddings to represent the text.

Figure 6 shows the pipeline used in the experiments from this section.

Figure 6 – Pipeline for embeddings training and evaluation



The pipeline from Figure 6 receives three types of input: the unlabeled corpora, the legal judgments' text and the labels. In terms of output, there are three types: the trained embeddings, the classification models and the test set.

The first step in the pipeline is the text extraction, where the original files in PDF and RTF are converted to raw text format. After text extraction, we applied some pre-processing steps, required before training the embeddings, as follows: the conversion to lower case and the removal of punctuation marks, special characters and symbol characters. We did not removed stopwords or apply stemmization or lemmatization, following the literature (MIKOLOV et al., 2013; PENNINGTON et al., 2014).

In the next step, there is the corpora split where the texts are divided in subsets. The three complete corpora comprised 3.7 billion, 100 million and 1.19 billion words for *general*, *air transport* and *global* corpora, respectively. The possible sizes of the subsets, considering word count are: 1,000; 10,000; 50,000; 100,000; 200,000; 500,000; 1,000,000; 5,000,000; 10,000,000; 25,000,000; 100,000,000; 500,000,000; 750,000,000 and 1,000,000,000.

We choose these corpora sizes to be able to compare the variation on performance metrics while increasing corpora size. For the air transport context, we could not embrace all these sizes due to limited corpora available. The largest subset for this context had 100 million words.

Finally, each of the subsets (smaller corpora) was used to train distinct word embeddings representation.

As word embeddings technique, we had to choose one among the available one, such as Word2Vec, GloVe, and FastText, due to time limits to finish the experiments for publication. We chose GloVe due to its good results in many NLP tasks, including text classification, and also for its training time which is significantly lower than other techniques like Word2Vec and FastText (PENNINGTON et al., 2014). In terms of GloVe parameters, we kept most of the default values, except for windows size, training iterations (steps in one epoch), and output vector length, which were set to 5, 100, and 100, respectively. With these values, we achieved better results in text classification.

Considering the corpus sizes and the parameters described above, we trained fifteen representations for *general* and fifteen for *global* context corpora. For *air transport* context corpora, we trained eleven word embeddings.

To evaluate the GloVe embeddings representations, we applied each of them to the task of text classification of judgments from JEC at UFSC. As shown in Figure 6, the legal judgments' text and labels serve as input for the text classification pipeline.

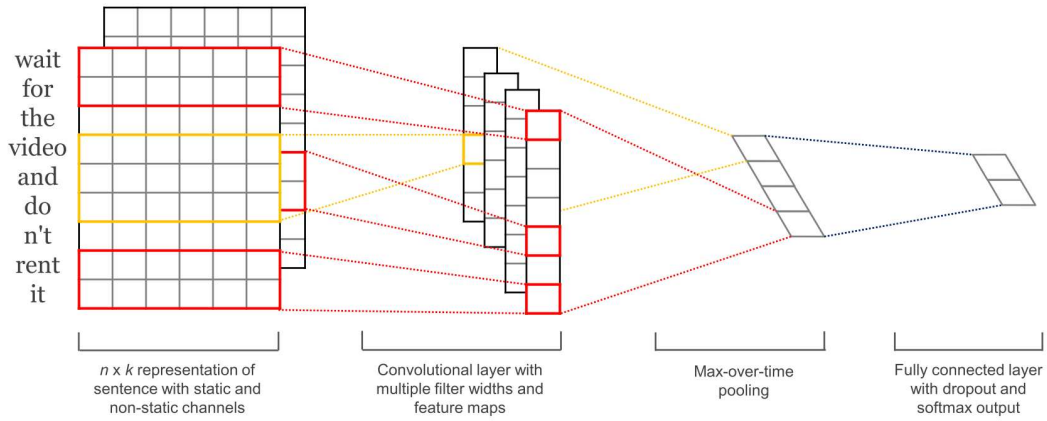
The next step is the text preprocessing, which is identical to the previous one, to reproduce the same textual structure used in the embeddings training. Using the Glove word embeddings, the text is numerically represented as detailed later.

To split the judgments in subsets for training and evaluation, we used the a random sampling method based on the proportions of 70%, 15% and 15% for train, validation and test sets, respectively. The models use the train set during the train step, the validation set used to tune the models, and the test set at the end, as final evaluation.

Then, next step is the training of the classifiers. We used CNN as a DL classification technique based on the literature (KIM, 2014). Figure 7 illustrates this model.

This CNN takes into account the order of the words by stacking the corresponding embeddings for each word as they occur in the text. Then it applies multiple convolutional masks with different dimensions that correspond to the red and yellow contours in Figure 7. Mask widths are equal to word embedding size while the heights can vary. In this context, mask height can be related to the idea of N-Grams, since

Figure 7 – CNN architecture for text classification



Source – Kim (2014).

they embrace multiple embeddings at the same time. In the original architecture, these heights were set to three, four, and five. We added one more mask of height two, which increased classification metrics. Also, based on empirical experimentation, we set the number of masks to ten for each of these sizes, without affecting our results, but decreasing the required training time.

In this work, we applied each of the embeddings trained in conjunction with the CNN described to the classification of JEC at UFSC judgments, where Out of Vocabulary (OOV) words are replaced by an vector of random values. Thus, we trained and tested 41 models. Furthermore, due to the stochastic nature of neural networks training methods (COHEN, 1995), each of these models was trained and tested 200 times and the resulting evaluation metrics were averaged.

Finally, we compare the performance in classification using accuracy and macro F1-Score.

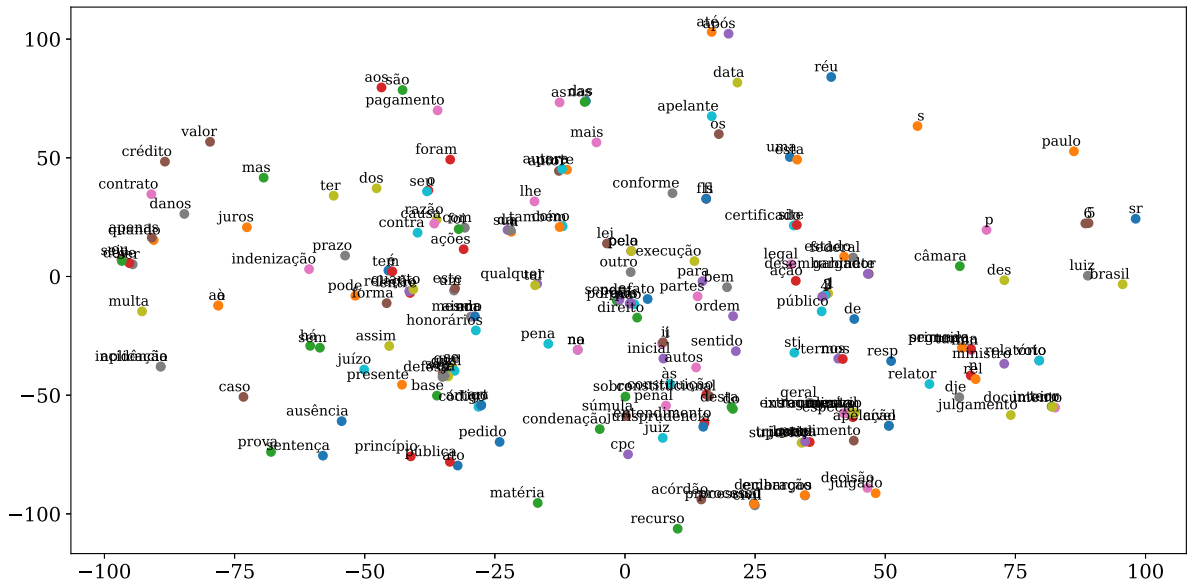
4.1.4 Results and Discussion

Following the steps presented in Section 4.1.3, we trained all 41 word embeddings representations for GloVe.

To illustrate how these embeddings behave, Figure 8 shows the projection in two dimensions, using t-distributed Stochastic Neighbor Embedding (t-SNE) (MAATEN; HINTON, 2008) of perplexity of 0.1, of a sample of words from *general* context embedding trained with 1 billion words. Each axis corresponds to a Principal Component (PC).

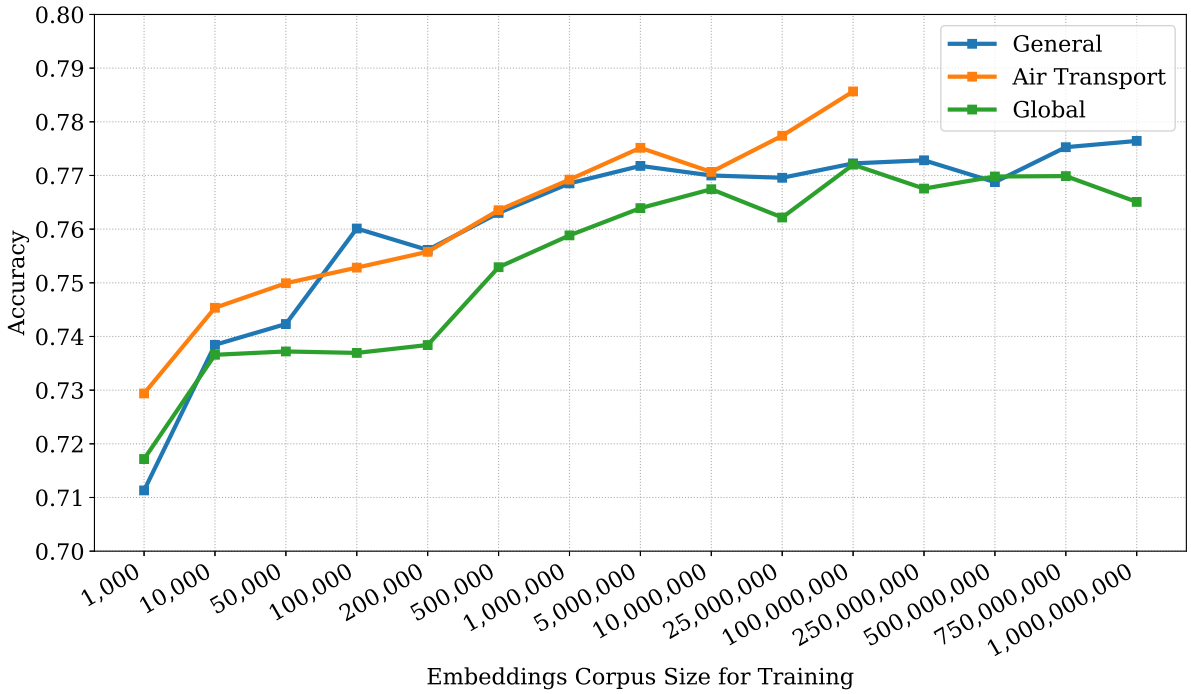
In Figure 9 and 10, we present the results, for accuracy and F1-Score, respectively, from test data applied to each CNN. These results are related to pre-trained embeddings with *general*, *air transport*, *global* texts. The x-axis denotes the corpus sizes used to train

Figure 8 – Word Embeddings projection using t-SNE



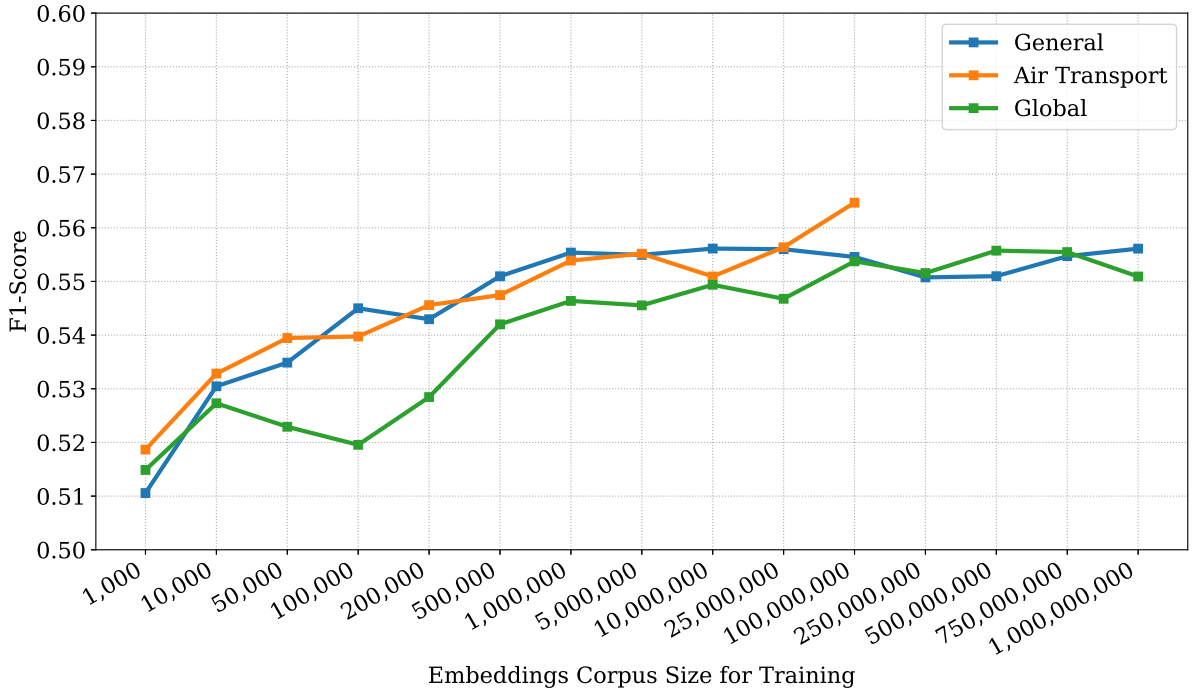
the embeddings, while the y-axis represents accuracy or F1-Score. Each data point represents the average of the evaluation metric, after 200 train and test repetitions using each specific embedding.

Figure 9 – Accuracy for test set from CNN in embeddings evaluation



We focus now on the part of research question regarding specificity: Does the *specificity* of the corpus used for word embeddings training impact the performance of a text classification using such representations?

Figure 10 – Macro F1-Score for test set from CNN in embeddings evaluation



In terms of accuracy, when we compare *global* against others (Figure 9), we have that higher text specificity leads to better results, for most of the corpus sizes used for embeddings training. Furthermore, when comparing *general* and *air transport* curves, there is a significant difference in accuracy only for the lowest and highest x-values. However, in terms of F1-Score, as shown in Figure 10, our observations change, once *general* and *air transport* curves have a similar shape. Also, for the highest corpus sizes, *general* and *global* curves converge to similar values of F1-Score. The differences in accuracy and F1-Score may emerge from the fact that our dataset to text classification is imbalanced, once the former does not take this fact into account, while the latter does. However, this result still requires further investigation.

Regarding the part of research question in terms of size: Does the *size* of the corpus used for word embeddings training impact the performance of a text classification using such representations?

When we observe both accuracy and F1-Score measures from Figure 9 and 10, it is clear the tendency for improvement while increasing corpus size. However, the metrics converge with the largest corpus sizes. There are two exceptions. The first one occurs with smaller values of corpus sizes for *global* curve, as it decreases in F1-Score measures. The second corresponds to the last data point in *air transport* curves. The former can happen when the classifier performs poorly for some classes while gets better in others. The latter may indicate that those curves could improve if we had more significant corpus sizes related to that context.

In general, we can note that the greater the corpus size in embeddings training,

the better are the results. However, this impact decreases as the corpus size increases until a point where more words in the corpus have little impact on the results.

4.2 TEXT CLASSIFICATION IN LEGAL JUDGMENTS

This section presents the results for the classification experiments involving JEC's legal judgments. The section starts from the clarification of the experiments' purpose, that is, how they help on answering the research question. Then, it presents the details on the datasets used for the experiments, the pipelines of the experiments for Classical ML and DL, and the results and discussion.

4.2.1 Experiment's Purpose

The experiments' purpose is to answer the second part of the research question, that is, whether DL techniques can achieve better performance on JEC's cases classification when compared to Classical ML techniques. Through the application of Classical and DL techniques to the prediction of the JEC legal judgments, we try to estimate the techniques' performance. Based on the results, we compare the techniques on how well they perform.

Besides the techniques comparison, in this section we tested the performance of the models using two datasets: legal judgments with full text and the legal judgments without the results section. Using such strategy, one can notice whether including or not the results part, containing textual description of the label, impact in the models performance.

In the experiments with Classical ML techniques, we applied the open-source software Orange Data Mining (Version 3.22) (DEMŠAR et al., 2013). Such tool aims at offering a variety of ML and TM techniques to the user in a simple way, without the need of any programming language. On the other hand, the experiments with DL techniques (not available in Orange) required the use of several tools: the Python programming language, version 3.8; the Keras framework,Version 2.4.3 (CHOLLET et al., 2015); the Natural Language Toolkit (NLTK), version 3.5 (LOPER; BIRD, 2002) and the Scikit-Learn framework, version 0.24.1 (PEDREGOSA et al., 2011).[3]

Finally, the results presented in this section relate to the first contact of the researcher with the areas of study. Thus, considering the extensive amount of publications on legal text classification (detailed in Appendix A) and the researcher's expertise, the classification experiments produced small contributions, that is, the applications of Classical ML and DL techniques to the prediction of the results of legal judgments from JEC.

[3] Pipeline for Orange3 and code for Keras are available at `https://github.com/thiagordp/text_classification_in_legal_docs`.

4.2.2 Datasets

For the classification experiments, two datasets were used. Both are similar as before (JEC/UFSC legal judgments), but smaller because it was the first experiment performed. The judgments were issued between January 2014 to May 2019.

The difference in the legal judgments from the two datasets used resides in their distinct textual structure. In one of them, we removed the result's part, while in the other dataset, we kept such part. Thus, considering the structure of a legal judgment described in Section 4.1.2, the first dataset contains legal judgments composed of two parts and less amount of text. The second dataset has three parts and more text.

Table 4 describes the quantity of examples for each label.

Table 4 – Label's distributions for text classification

Label	Examples
Well Founded	214
Partly Founded	379
Dismissed without prejudice	10
Not founded	70
TOTAL	673

In terms of quantitative information from the datasets, Table 5 presents the average count of tokens per document and the vocabulary size after each of the preprocessing steps for each dataset. As we will describe later, distinct preprocessing steps carried out for the experiments with Classical ML and DL. The information of tokens per document in the experiment with Classical ML was not available due to the restrictions from Orange Data Mining.

Table 5 – Information on the datasets for classification

Experiment's Preprocessing	Tokens per document (w/ result)	Vocabulary Size (w/ result)	Tokens per document (w/o result)	Vocabulary Size (w/o result)
Classical ML	-	12,994	-	12,898
DL	673.0	15,377	644.2	15,224

Similar to Section 4.1, some of the legal judgments from the datasets presented more than one result, that is, they had more than one distinct labels. An example of this type of judgments happens when two people file a joint lawsuit against an airline, and the judge set distinct judgments for each of them. In those cases, the documents were replicated for each of the labels indicated on its results section, culminating in a total of

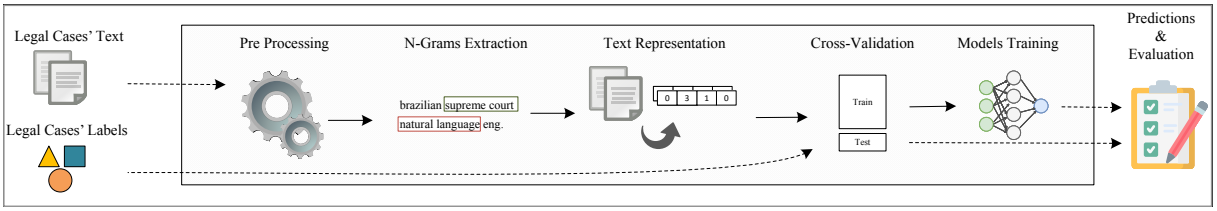
673 legal judgments, in each dataset, to serve as input for the experiments. Considering distinct legal judgments only, the dataset contains 665 documents.

4.2.3 Pipelines

This section describes the pipelines and experimental setup for the experiments with Classical ML and DL techniques.

The first set of classification experiments focused on the application of Classical ML techniques. In such experiments, Orange 3 served as execution environment and followed the pipeline from Figure 11. It receives two types of input: the texts from legal judgments, in a plain text format, and their labels, that is, the judgments' results. As outputs, the pipeline makes available the trained models and the test set which are passed to the prediction and evaluation step.

Figure 11 – Pipeline for legal text classification using Classical ML techniques



The first step in the pipeline is the data preprocessing. Considering the available techniques for preprocessing textual data, described in Chapter 2 and in the literature, in Appendix A, we applied transformation, tokenization, stemming and filtering, previously discussed in Section 2.2.1:

- **Tranformation:** the conversion to lower case to standardize the spelling of words.
- **Tokenization:** the application of regular expression (\w+) to detect the pieces of texts, while removing spaces, symbols, and punctuation.
- **Stemming:** To reduce the variability of similar words, we applied Porter Stemmer (PORTER, 1980), a simple and efficient stemming algorithm to the Portuguese language. However, it may make errors, such as contracting the word *morais* to *morai*, instead of *moral* (Portuguese words for singular and plural of *moral*, respectively).
- **Filtering:** Removing stopwords, such as prepositions and articles to keep only the meaningful words.

The next step in the pipeline relates to the extraction of N-Grams, which detects sequences of two or more words that appear together consistently in the text. In this research, the limit of the length of N-grams was two. Bigger numbers of N-grams would lead to large textual representations.

After N-Grams Extraction, the numerical representations of the documents are created using the algorithm BOW. In the experiments with Classical ML, we used the TF to calculate the values of the BOW model.

The next step consists on dividing the dataset in two subsets: train and test sets. As described in Chapter 2, there is the cross validation. In this research, k was set to 10, that is, 90% of the dataset used for training and 10% for testing the models. Such proportion is a common choice to avoid bias while keeping some level of variance in the division of the folds (AIROLA et al., 2011)

The next step consists on training the models to predict the result of the judgments from JEC according to the possible outcomes, as described in Section 4.1. Experiments involved the following techniques: k Nearest Neighbors (kNN), SVM, RF, NN, NB and LR. Table 6 presents the hyper-parameters applied to each technique, based on the values suggested by Orange 3.

Table 6 – Hyperparameters for classification techniques

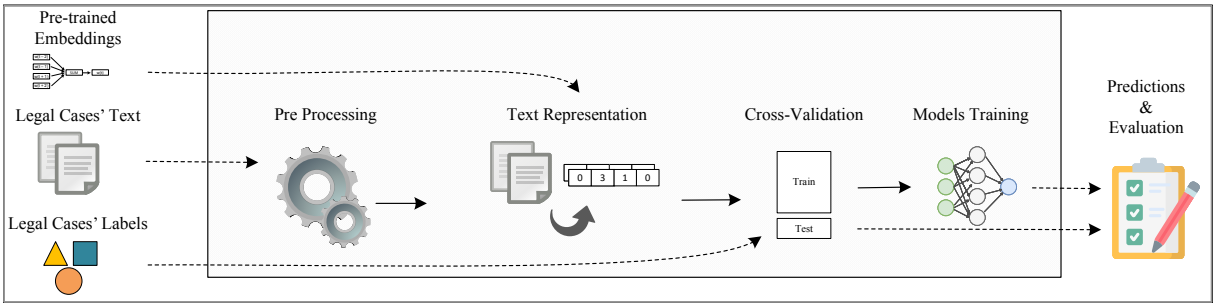
Technique	Hyper-parameters
kNN	Number of Neighbors: 4; Distance Metric: Euclidean; Weight: Uniform
LR	Regularization type: Ridge (L2); C (strength): 1
NB	–
NN	Hidden Layers: 2 Neurons in each layer: 100, 50; Activation Function: tanh; Solver: Stochastic Gradient descent (SGD)
RF	Number of Trees: 10; Minimum subset size: 5
SVM	C (cost): 1.0; ϵ (Regression loss): 0.1; Kernel: Radial Basis Function (RBF); Iteration Limit: 100

Finally, there is the evaluation step, which estimates how well the models performed on predicting unseen legal judgments. To measure the performance, we used the Accuracy, detailed in Chapter 2. At the time these experiments were published, only the accuracy was used.

The second set of experiments related to the application of DL techniques to the prediction of the legal judgments' results. The pipeline used for these experiments is presented in Figure 12.

Such steps in the pipeline are similar to those in Figure 11, however there are

Figure 12 – Pipeline for legal text classification using DL techniques



distinct settings. Thus, the pipeline receives three types of input: the legal judgments' text, their labels and the pre-trained word embeddings models.

Following, the preprocessing step prepares the text to the next steps in the pipeline, however with different settings from Figure 11. There is the transformation, tokenization using regular expression (\w+), however we did not apply stemming or filtering, following the literature (MIKOLOV et al., 2013; PENNINGTON et al., 2014). Thus, to reproduce the aspects of the corpus applied to the pre-training of the embeddings, their application in ML tasks may not include those preprocessing techniques. After preprocessing there is the text representation, where pre-trained embeddings techniques were applied. We selected the pre-trained embeddings based on best results in Section 4.1. That is, the word embeddings pre-trained using the corpus related to air transport only. In the referred section, only the GloVe technique was applied and tested due to time limits. However, we later trained other word embeddings in the same corpora to apply in the classification experiments, using the default parameters from Gensim (ŘEHŮŘEK; SOJKA, 2010).

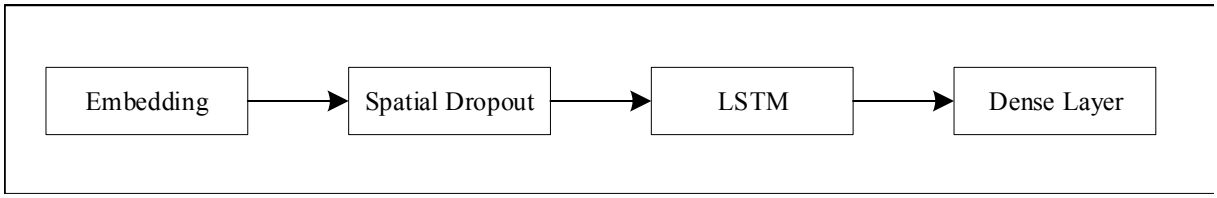
The next step, cross-validation, follows the setup from the Classical ML experiments, i.e., the number of folds set to ten.

Later, there is the models training step, involving three DL techniques: CNN, LSTM and Bi-LSTM with Self Attention. The CNN used has the same hyperparemeters from Section 4.1.

The LSTM architecture, as shown in Figure 13, receives as input the embeddings, which can be fine-tuned during the training process. Such a setting is enabled in this architecture. The embeddings pass to a Spatial Dropout layer, a regularization method to avoid overfitting on recurrent networks, especially when embeddings can be fine-tuned (GAL; GHAHRAMANI, 2016). Then, there is the LSTM layer with 100 units with a dropout and a recurrent dropout of 0.2. Finally, the output layer corresponds to a Dense Layer with four neurons having sigmoid as activation function.

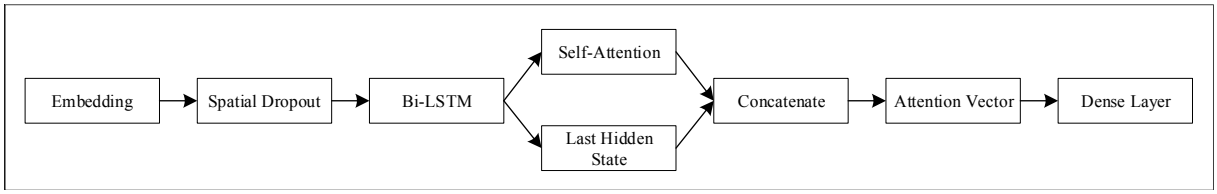
The architecture for Bi-LSTM with Self Attention is shown in Figure 14. It starts with the Embedding layer followed by Spatial Dropout, both with the same settings from previous LSTM model. The Bi-LSTM layer follows, containing 100 units. Then,

Figure 13 – LSTM architecture for text classification



there is the Many to One Attention step composed of the Self-Attention, the Concatenation with last Bi-LSTM Hidden State and the Attention Vector. Finally, there is the Dense Layer as the output layer with four neurons with sigmoid activation function.

Figure 14 – Bi-LSTM with Self-Attention architecture for text classification



To run the DL experiments based on the architecture from Figure 12, distinct setups have been made considering the three DL techniques, the two datasets, and the five embeddings models.

4.2.4 Results and Discussion

The results obtained after applying the sets of input to the pipelines from Figure 11 and 12 are presented in the following paragraphs. Table 7 details the accuracy obtained by the techniques and representations on each label using the full judgments' text.

From Table 7, one can notice that most techniques achieved accuracies superior to 70%, indicating good results on the classification of legal judgments from JEC. In terms of techniques' performances, the CNN with GloVe embeddings achieved the best accuracies for the *Well Founded*, *Partially Founded*, *Not founded* labels and for the total accuracy. The CNN also achieve good results when combined with the Word2Vec SG and FastText SG. Besides CNN, the next techniques with best performances are Classical ML techniques, that is, LR, NN and RF.

In terms of worst results, the CNN with FastText CBOW embeddings achieved the smallest total accuracy, followed by SVM, Naïve Bayes and CNN with Word2Vec CBOW. Regarding labels, on the other hand, SVM achieved the worst results for the *Well Founded*, *Partially Founded*, *Not founded*, and the Naïve Bayes for the *Dismissed without Prejudice* label.

The second part of experiments on Classical ML and DL related to the classification of legal judgments from JEC without the judgments' result. Table 8 contains the

Table 7 – Classification accuracy for the dataset with judgments' result

Technique	Type of ML technique	Representation	Well founded	Partly founded	Not founded	Dismissed without prejudice	Total Acc
kNN	Classical	BOW TF	75,9%	71,9%	93,5%	95,7%	75,8%
LR	Classical	BOW TF	87,1%	86,2%	97,6%	97,9%	87,8%
NB	Classical	BOW TF	75,2%	47,3%	91,4%	19,6%	60,3%
NN	Classical	BOW TF	85,9%	84,9%	97,5%	97,9%	86,7%
RF	Classical	BOW TF	82,6%	82,6%	96,1%	97,9%	84,2%
SVM	Classical	BOW TF	34,3%	45,5%	89,6%	98,5%	47,3%
Bi-LSTM-SA	DL	FT CBOW	81,1%	80,4%	96,6%	98,2%	78,1%
Bi-LSTM-SA	DL	FT SG	81,1%	80,2%	97,3%	98,5%	78,6%
Bi-LSTM-SA	DL	GloVe	83,1%	82,6%	96,9%	98,2%	80,4%
Bi-LSTM-SA	DL	W2V CBOW	81,3%	80,1%	97,5%	98,4%	78,6%
Bi-LSTM-SA	DL	W2V SG	80,8%	79,6%	97,5%	98,4%	78,2%
CNN	DL	FT CBOW	48,3%	48,9%	90,2%	98,5%	42,9%
CNN	DL	FT SG	94,0%	94,8%	96,9%	98,5%	92,1%
CNN	DL	GloVe	**97,2%**	**97,5%**	**98,1%**	98,4%	**95,5%**
CNN	DL	W2V CBOW	73,0%	68,7%	90,6%	98,5%	65,4%
CNN	DL	W2V SG	96,0%	96,9%	97,6%	98,5%	94,5%
LSTM	DL	FT CBOW	78,2%	74,6%	90,5%	98,5%	70,9%
LSTM	DL	FT SG	80,2%	76,4%	92,9%	98,5%	74,0%
LSTM	DL	GloVe	80,7%	78,6%	96,1%	98,5%	77,0%
LSTM	DL	W2V CBOW	77,4%	73,8%	89,6%	98,5%	69,7%
LSTM	DL	W2V SG	78,6%	73,7%	91,8%	98,5%	71,3%

accuracy achieved by each technique and representation for the four labels and total accuracy using the text of the judgments without result.

The results show the sequence of best techniques has changed, as the Classical ML techniques RF, LR and NN achieved the best total accuracies followed by the CNN with Word2Vec SG. In terms of labels, the RF performed better for the *Well founded* and *Partially founded*, while NN performed better with *Not founded*. For the label *Dismissed without prejudice*, several classes achieved good results, although it has the smallest sample. On the other hand, similar to the results from Table 7, the techniques with worse performance were the CNN with FastText CBOW followed by SVM and CNN with Word2Vec CBOW.

In general, it is inferred that the accuracy in the experiment with the removal of the judgments' result (part of the text indicating the label to which it belongs) suffered a minimal reduction for the Classical ML techniques as well as LSTM and Bi-LSTM-SA, which demonstrates that the classifiers were able to maintain their performance with the text in which the facts narrated by the parties to the process are reported and

Table 8 – Classification accuracy for the dataset without judgments' result

Technique	Type of ML technique	Representation	Well Founded	Partly founded	Not founded	Dismissed without prejudice	Total Acc
kNN	Classical	BOW TF	75,2%	72,1%	90,9%	93,2%	75,4%
LR	Classical	BOW TF	80,7%	79,2%	94,9%	97,9%	81,6%
NB	Classical	BOW TF	74,3%	46,5%	90,2%	18,1%	59,5%
NN	Classical	BOW TF	81,0%	79,0%	**95,1%**	97,9%	81,6%
RF	Classical	BOW TF	**82,2%**	**79,9%**	92,7%	97,9%	**82,2%**
SVM	Classical	BOW TF	33,4%	45,0%	89,6%	98,5%	46,7%
Bi-LSTM-SA	DL	FT CBOW	79,5%	77,1%	94,3%	98,5%	74,7%
Bi-LSTM-SA	DL	FT SG	78,5%	75,8%	94,6%	98,5%	73,7%
Bi-LSTM-SA	DL	GloVe	79,9%	77,3%	94,8%	98,4%	75,2%
Bi-LSTM-SA	DL	W2V CBOW	79,5%	76,7%	93,9%	98,5%	74,3%
Bi-LSTM-SA	DL	W2V SG	80,4%	77,4%	93,8%	98,5%	75,0%
CNN	DL	FT CBOW	50,5%	51,7%	89,9%	98,5%	45,3%
CNN	DL	FT SG	80,8%	78,0%	93,0%	98,5%	75,2%
CNN	DL	GloVe	80,4%	77,3%	93,0%	98,5%	74,6%
CNN	DL	W2V CBOW	66,6%	58,7%	90,0%	98,5%	56,9%
CNN	DL	W2V SG	81,7%	79,5%	93,3%	98,5%	76,5%
LSTM	DL	FT CBOW	79,8%	71,3%	88,6%	98,5%	69,1%
LSTM	DL	FT SG	77,5%	69,1%	88,1%	98,5%	66,6%
LSTM	DL	GloVe	81,1%	74,7%	91,2%	98,5%	72,8%
LSTM	DL	W2V CBOW	80,8%	73,7%	87,8%	98,5%	70,4%
LSTM	DL	W2V SG	79,2%	72,4%	88,4%	98,5%	69,2%

the legal grounds applicable to the case. However, for the CNN there was a significant decay in performance when removing the cases' result. And, such decay in performance may show the technique inferred that the text from the result part had significant information for the classification of the legal cases. However, the removal of such part made it harder for the CNN to infer the judgment's label based on remaining two parts of the text.

Another observation for the CNN is the impact of the representations in the performance of the technique, as changing the representation used in the pipeline was enough to reduce the accuracy by more than 30% when comparing to the best CNN results. However, for the other two DL techniques, LSTM and Bi-LSTM with Self Attention, the differences in performance while shifting representations were significantly smaller.

Finally, a good performance when classifying the judgments without results becomes a prerequisite for carrying out experiments with texts from legal proceedings in which there is still no sentence, that is, in which the judge has not yet decided the

result. In this case, the best choice may be the use of Classical ML techniques, such as LR and RF, as they perform better on the classification task than DL techniques and due to the fact those models are less complex and require less examples to train. The DL techniques, especially CNN with GloVe may be useful inside JEC when the objective is to organize existing judgments by categories such as their labels or matters, for example.

This conclusions are limited to our small dataset from JEC. Thus, if we had a larger dataset with more training examples, the DL would possibly achieve better results.

4.3 TEXT REGRESSION IN LEGAL JUDGMENTS

The experiments, results and discussions presented in this section are related to text regression. Firstly, there are the experiment's purposes, and how they help on answering the research question. Then, it presents the details on the datasets, pipeline, the results, and the discussion.[4]

4.3.1 Experiment's purpose

The experiment's purpose is to answer the third part of the research question, i.e., to what extent the prediction of compensation values can be *accurate* and *helpful* in the legal environment using regression models, we set up pipelines for text regression which include some TM and ML techniques. We start from a simple pipeline, called *baseline*. Then, it receives several improvements, or *adjustments*, forming a new pipeline, called *full pipeline*.

To setup the experiments and the pipelines, we used the Python programming language combined with the Scikit Learn (PEDREGOSA et al., 2011) version 0.24.1, the NLTK (LOPER; BIRD, 2002), the Pandas library, version 1.2.3 (MCKINNEY, 2010), and Matplotlib, version 3.3.4 (HUNTER, 2007).

4.3.2 Dataset

The dataset is similar to those presented in Section 4.1 and Section 4.2. It is composed of 940 legal judgments issued between February 2011 to September 2020 into the JEC located at the UFSC.

The dataset contains a vocabulary of 16,924 words, 712,057 total tokens and an average of 758 tokens per document (after the preprocessing step). The labels (compensation values) vary from 304 to 25,000 Brazilian *Reais* with an average of 6,344 and a standard deviation of 3,471. In other words, only judgments with *well founded* or *partly*

[4] The code is available at `https://github.com/thiagordp/text_regression_in_law_judgments`.

well founded results may appear in this dataset as they have compensation values bigger than zero.

Similar to the text representation and text classification experiments, to evaluate the model, we remove the part of the document that refers to the result of the judgment since it contains the value of compensation for immaterial damage. That way, the models predicts the compensation value based on the report and the legal reasons for the decision. The result part is only used to set the legal cases' labels.

As a complement to the textual dataset, a legal expert manually extracted some attributes and their values from each document, which was possible through a clustering step. One of the attributes identified, for example, is the flight delay period. Therefore, the expert analyzed every judgment and extracted the value of this attribute (the delay hours).

It follows the list of such attributes together with an explanation of their importance for the prediction problem.

- **Date of judgment:** The judge's perspectives may change over time. Consequently, the amount of compensation may vary by date. In the dataset, this is represented by day, month, and year.
- **Judge:** Each judge is free to set the amount of compensation according to his/her conviction on the case. In this sample period, the judgments were elaborated by different judges. In the dataset, this is represented by the name of the thirty one judges who prepared the collected judgments.
- **Type of judge:** In the JEC, there are three types of judges: chief, assistant, and voluntary. The chief judge is responsible for the court and is the one who, as a rule, judges the lawsuits. The assistant or substitute judge is the one who judges when the chief judge needs to be absent. And the voluntary judge is the one who has a law degree but is not invested in the position. He or she voluntarily prepares judgments that are submitted to the approval of the chief judge. An assistant judge can freely fix a different value of compensation than a chief judge. The voluntary judge can do this too, but the chief judge can modify the value. In the dataset, this is represented by the a categorical variable.
- **Permanent baggage loss:** It is an event that can generate compensation for immaterial damage. In the dataset, this is represented by "yes" (when there was a loss) and "no" (when there was no loss).
- **Tampered baggage:** Depending on the level of damage or in case of missing consumer's belongings (theft), it is an event that can generate compensation for immaterial damage. In the dataset, this is represented by "yes" (when there was tampering) and "no" (when there was no tampering).

- **Temporary baggage loss:** It is an event that can generate compensation for immaterial damage. In the dataset, this is represented by "yes" (when there was a loss) and "no" (when there was no loss).
  - **Loss interval:** It is a sub-attribute. The longer the delay in returning the baggage to the consumer, the greater can be the value of the compensation for immaterial damage. In the dataset, this is represented by days.
- **Flight cancellation:** It is an event that can generate compensation for immaterial damage. We consider as flight cancellation those cases with no rebooking or when the destination is changed. In the dataset, this is represented by "yes" (when there was cancellation) and "no" (when there was no cancellation).
- **Flight delay:** It is an event that can generate compensation for immaterial damage. We consider as flight delay those cases with rebooking. In the dataset, this is represented by "yes" (when there was a delay) and "no" (when there was no delay).
  - **Delay interval:** It is a sub-attribute. The longer the delay in rebooking (that is, the longer the interval between the initially contracted flight and the actual flight operated), the greater can be the value of the compensation for immaterial damage. In the dataset, this is represented by hours and minutes.
- **Adverse weather conditions:** It is an event that excludes the possibility of compensation for immaterial damage because it is an unpredictable situation. Even the airline effort is not capable of overcoming them, so there is no way to impute liability to it. In the dataset, this is represented by "yes" (when there was proven bad weather) and "no" (when there was no proven bad weather).
- **Consumer fault:** It is an event that excludes the possibility of compensation for immaterial damage because it removes the airline's liability. An example of this situation is when the consumer does not arrive at the airport in plenty of time to check his/her flight and bags. In the dataset, this is represented by "yes" (when there was the consumer fault) and "no" (when there was no consumer fault).
- **Overbooking:** Selling more tickets for a flight than are available is considered an abusive practice. Thus, it is an event that can generate compensation for immaterial damage. In the dataset, this is represented by "yes" (when there was overbooking) and "no" (when there was no overbooking).
- **No show:** Cancellation of the return ticket unilaterally when the consumer does not show up on the outward flight is considered an abusive practice.

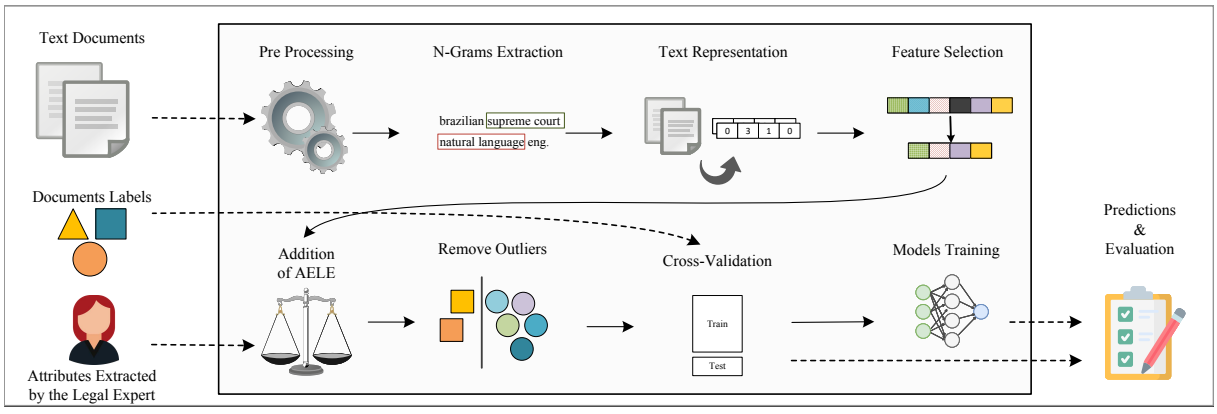
Thus, it is an event that can generate compensation for immaterial damage. In the dataset, this is represented by "yes" (when there was cancellation by no show) and "no" (when there was no cancellation by no show).

- **Right to regret and repayment claim:** Hindering the consumer's repayment when he/she decides to cancel the acquired ticket is an event that can generate compensation for immaterial damage. This situation is known by a sequence of bad experiences (called *via crucis* by judges) that the consumer must face getting the repayment. In the dataset, this is represented by "yes" (when repayment was hindered) and "no" (when the repayment was not hindered or when there was no claim).
- **Downgrade:** The airline changes a business class passenger to economy class. Besides a breach of contract, it is also a breach of the consumer's expectation, and, therefore, it is an event that can generate compensation for immaterial damage. In the dataset, this is represented by "yes" (when there was a downgrade) and "no" (when there was no downgrade).

4.3.3 Pipeline

To answer the research question, we propose the application of several TM and ML techniques on a pipeline for regression on legal texts. Thereby, we aim to create learning models capable of making accurate and helpful predictions for immaterial damage compensation. Figure 15 shows the proposed pipeline, built upon the one from Section 2.4. We incremented it with TM and ML techniques, called *adjustments*.

Figure 15 – Full pipeline for legal text regression



The pipeline receives three types of input: the text of the legal judgments, their labels, and the Attributes Extracted by the Legal Expert (AELE) for each document (cf. Section 4.3.2). The preprocessing step converts the text to lowercase and remove noise characters, punctuation, stopwords such as *de*, *para* (prepositions in Portuguese), using the NLTK (BIRD; LOPER, 2004).

The first adjustment is *N-Grams Extraction*, varying in length from one to four. However, as this range would lead to an unreasonable dimensionality, we limited the BOW representation to the 25,000 most frequent units, using Scikit-Learn (PEDREGOSA et al., 2011).

For the text representation, we use BOW using TF values. We also tested word embeddings trained with legal documents written in Portuguese and TF combined with Inverse Document Frequency (IDF), although TF achieved the best results for the experiments presented in this work.

The second adjustment is *Feature Selection*, using the Mutual Information method. It maps the relationship between each feature (unit in the BOW) and the dependent variable (COVER; THOMAS, 2005), the amount of immaterial damage compensation. As we tested a wide range of values as the number of features to select, we set it to 500 to consume less time on the experiments and still achieve good results.

The third adjustment is *Addition of AELE*, based on the attributes described in Section 4.3.2. Categorical features such as *judges* and *types of judges* were converted to one-hot encoding. Real value features, such as *delay interval*, were not modified. In the end, the final representations of the documents were composed of a concatenation of the 52 features from the legal attributes and the BOW features. In the case when feature selection is activated, there were 500 BOW features. Otherwise, there were 50,000 (the reasons to activate or not an adjustment are detailed later in this section).

The fourth adjustment is *Outliers Removal*. As previously described, outliers are very distinctive examples in the dataset, and by removing them, we make it easier for the models to learn. To detect outliers, we used the Isolation Forest with contamination set to ten per cent. Moreover, we have placed this step in two different positions: before and after cross-validation, but we did not apply both in the same pipeline. The former intends to remove outliers from the whole dataset, while the latter, from the train set. By removing outliers from all dataset, we imply that our future cases for prediction will not contain outliers.

The fifth adjustment is *Cross-Validation*, which uses multiple combinations of the train and test sets and the resulting metrics are averaged. In this work, we set the number of folds to five, so, in each step, eighty per cent and twenty per cent of the dataset is used for train and test, respectively.

The selected techniques of ML for the regression task are listed in Table 9. We used the AdaBoost (AB), Bagging (BG), Decision Trees (DT), feed-foward Neural Networks (NN), Elastic Net (EN), Ensemble Voting (EV), Gradient Boosting (GB), Random Forest (RF), Ridge (RG), Support Vector Machine (SVM), and XGBoosting (XGB).

Considering the problem of overfitting, we evaluate them for two configurations: *simple* and *complex*. In the former, we define some constraints to the models such as the number of iterations and maximum tree levels, while in the latter we let the models

free, without such constraints. The forth column of the table contains the parameter values used in both configurations and any unlisted parameters in Table 9 follow the default values from Scikit-Learn (PEDREGOSA et al., 2011).

Finally, the sixth adjustment is *Overfitting Avoidance*, that is implemented by simpler models in our pipeline. And, we note that Ensemble Voting Model is an ensemble of ensembles, so it uses models like Bagging and XGBoosting with the same parameters as described in their respective lines.

Table 9 – Regression techniques and parameters

Technique	Parameters (Complex)	Parameters (Simple)	Common Parameters
AB	Nº Estimators: 100	Nº Estimators: 50	Learning Rate: 0.1
BG	Nº Estimators: 100	Nº Estimators: 50	-
DT	Maximum Depth: Unlimited Max Leaf Nodes: Unlimited	Maximum Depth: 10 Max Leaf Nodes: 100	-
NN	Hidden Layers: 5 Neurons: 512 (Each Layer) Max Iterations: 100 Early stopping: Deactivated	Hidden Layers: 5 Neurons: 256 (Each Layer) Max Iterations: 50 Early stopping: Actived	Activation: ReLU Batch Size: 16
EN	Max Iterations: 100	Max Iterations: 50	-
EV	Bagging Neural Network Gradient Boosting XGBoosting	Bagging Neural Network Gradient Boosting XGBoosting	-
GB	Nº Estimators: 100 Max Depth: Unlimited Max Leaf Nodes: Unlimited	Nº Estimators: 50 Max Depth: 10 Max Leaf Nodes: 100	-
RF	Nº Estimators: 100 Max Depth: Unlimited Max Leaf Nodes: Unlimited	Nº Estimators: 50 Max Depth: 10 Max Leaf Nodes: 100	-
RG	Max Iterations: 100	Max Iterations: 50	Alpha: 0.1 Tolerance: 0.001
SVM	Max Iterations: 100	Max Iterations: 50	C: 1.0 Epsilon: 0.2 Kernel: RBF
XGB	Nº Estimators: 100 Max Depth: Unlimited	Nº Estimators: 50 Max Depth: 10	-

The final step is the evaluation of our models. From their predictions on the test set, we measure the prediction quality using three metrics: RMSE, MAE and the Coefficient of Determination (R^2).

In the experimental setup, we initially evaluate the two pipelines of Figure 4 and Figure 15, which we call *baseline* and *full* pipelines, respectively. Thereby, we can

have an overall estimate of how much our adjustments in the *full* pipeline improve the regression metrics.

Furthermore, to verify in what extent the prediction can be accurate, we also performed some experiments with other combinations of adjustments, for instance, bypassing *N-Grams Extraction* and *Feature Selection*, while keeping *Addition of AELE*, *Outliers Removal*, *Cross-Validation*, and *Overfitting Avoidance*. With the experiments for the different pipelines, we can also measure how much each adjustment contributes for the performance of the models.

To run the experiments, we first set which adjustments to use, that in total embraced 80 combinations. For each combination, we executed the pipeline twenty five times. If *Cross-Validation* is disabled, we only train and test the models once, and we do five times, otherwise. To get the final metrics of the set of repetitions, we took the average for MAE, RMSE and R^2 among the repetitions.

4.3.4 Results and Discussion

This section presents the results from the experiments regarding the different pipelines: *baseline*, *full pipeline*, and the eighty combinations of adjustments. We analyze the adjustments' influence in terms of prediction quality and execution time.

4.3.4.1 Results from baseline and full pipelines

Considering the steps described in Section 4.3.3, we run the experiments for the *baseline* as shown in Figure 4. This setup does not include the adjustments. The Figure 16 presents the results for each regression model in terms of R^2, where higher values indicate better prediction quality, and Figure 17 presents the results in terms of errors (RMSE and MAE), where smaller values indicate better prediction quality.

Figure 16 – Results for R^2 from baseline pipeline

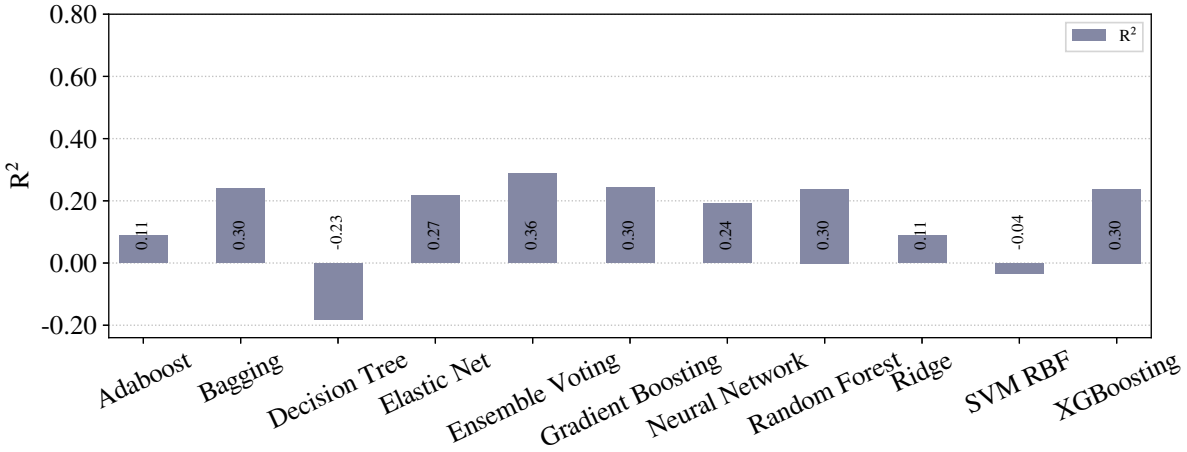
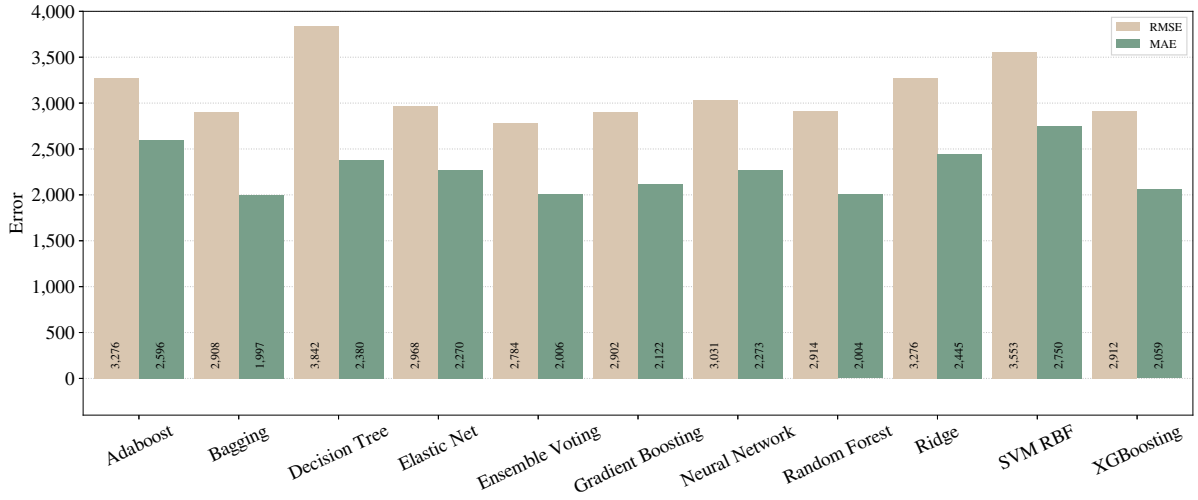
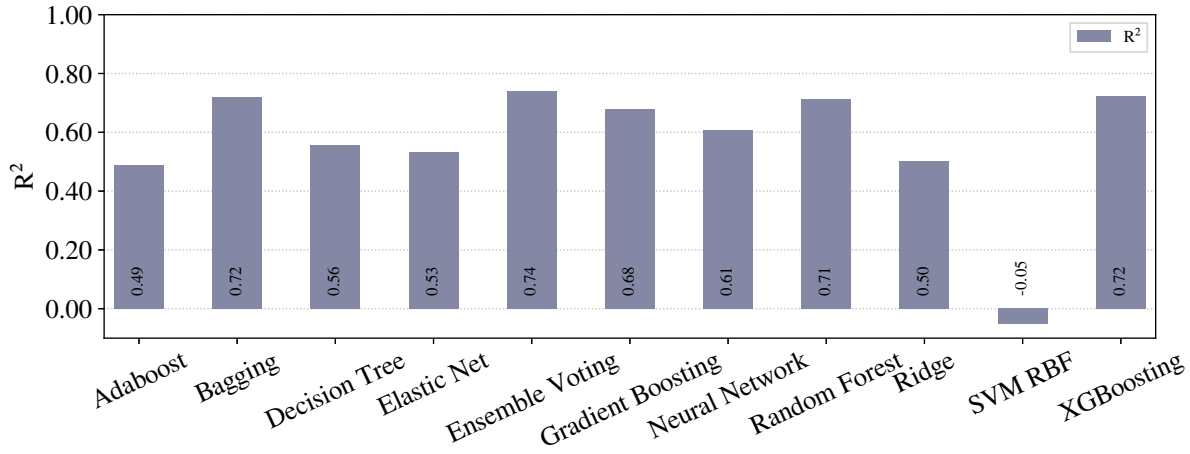


Figure 17 – Results for MAE and RMSE from baseline pipeline



We repeated the steps from Section 4.3.3 for the *full* pipeline with all adjustments activated (except outliers removal in training data) and the achieved results are shown in Figure 18 and Figure 19.

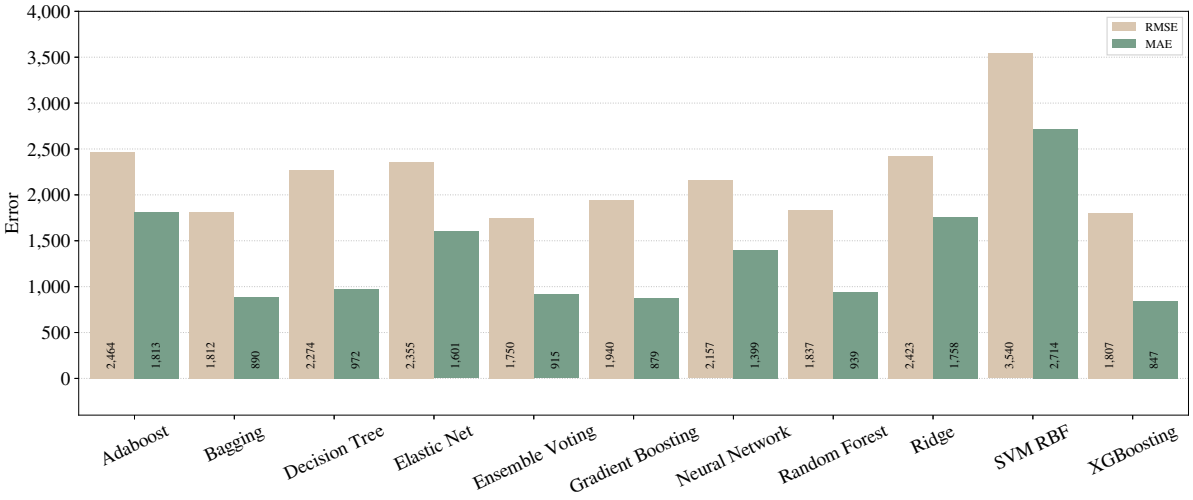
Figure 18 – Results for R^2 from full pipeline



From Figures 16, 17, 18, and 19, one can note significant improvements on the three metrics for most of the techniques, when we compare the *baseline* and *full* pipeline. The exception is SVM with RBF kernel. In that case, we can affirm that SVM is underfitted, as the poor results stood regardless the pipelines we would apply. On the other hand, in terms of the best techniques, we can realize that Ensemble Voting achieved the best results among the techniques in terms of RMSE and R^2. Thus, merging the techniques in Ensemble Voting achieved better results when compared to the models alone for R^2 and RMSE. XGBoosting achieved the best prediction quality in terms of MAE.

As expected, we can conclude that the *full* pipeline leads to better results than

Figure 19 – Results for MAE and RMSE from full pipeline



baseline. Moreover, from the legal expert experience, a MAE of less than 1,000 can be considered almost irrelevant in the context of legal compensation.

4.3.4.2 Results from combinations of adjustments

This section shows the performance of combinations of adjustments and whether they achieved any better result when compared to the *full* pipeline. Considering again Figure 15, we randomly selected a total of eighty different pipelines. When an adjustment is deactivated, its predecessor step in the pipeline is connected to its successor. For example, if we deactivate *N-Grams Extraction*, the preprocessing step will be connected to the representation step and so on. Furthermore, the pipeline stays the same despite the (de)activation of *Overfitting Avoidance* adjustment. This adjustment is more related to the configuration for the training step, that is, use complex (when deactivated) or simpler models (when activated) from Table 9.

We represent a combination of adjustments as a binary number. If the adjustment is deactivated the digit is zero, and it is one otherwise. We assigned the positions to adjustments in the binary number in this order, from left to right: *Feature Selection*, *Outliers Removal (Train Set)*, *N-Grams Extraction*, *Addition of AELE*, *Cross-Validation*, *Overfitting Avoidance* and *Outliers Removal (All Dataset)*.

Figure 20 shows the results, in which the x-axis represents the combinations, the y-axis represents the R^2 metric and each line is a different technique. To better detect the patterns, we have arranged the combinations in decreasing order of R^2 from Ensemble Voting regression. Following the same idea, Figure 21 shows the results for RMSE with the same order of combinations in the x-axis.

The first observation is that we can achieve prediction qualities better than the *full* pipeline. The best pipeline, based on RMSE and R^2, is represented as 1010011, with *Feature Selection*, *N-Grams Extraction*, *Overfitting Avoidance* and *Outliers Removal*

Figure 20 – R^2 for the pipelines based on combinations of adjustments

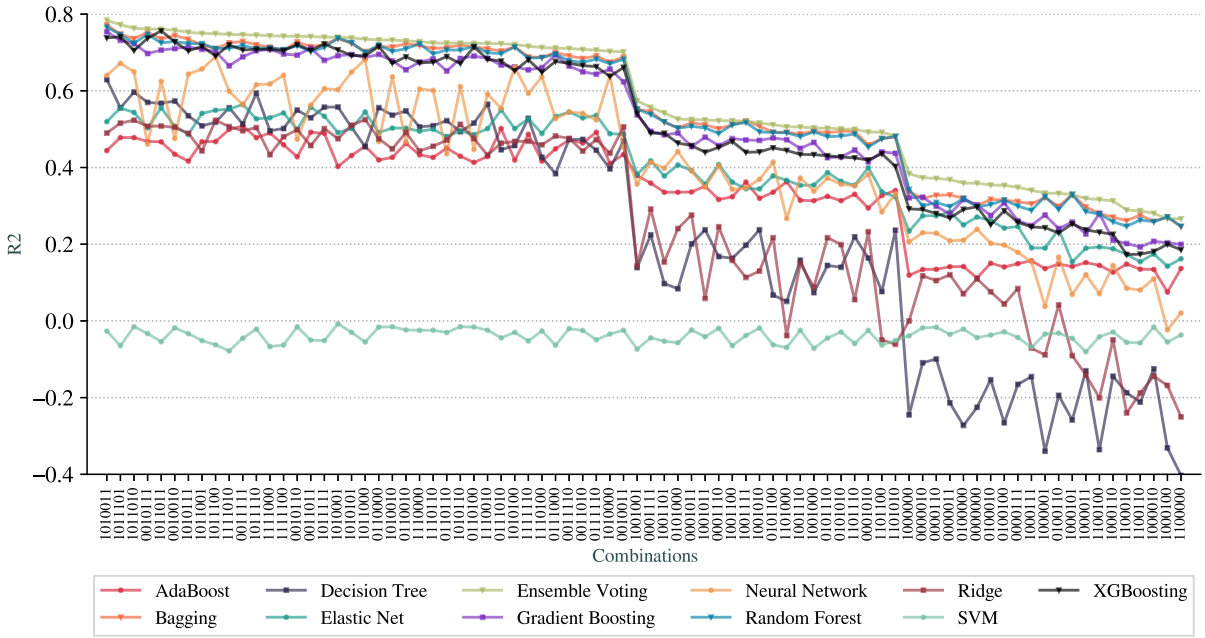
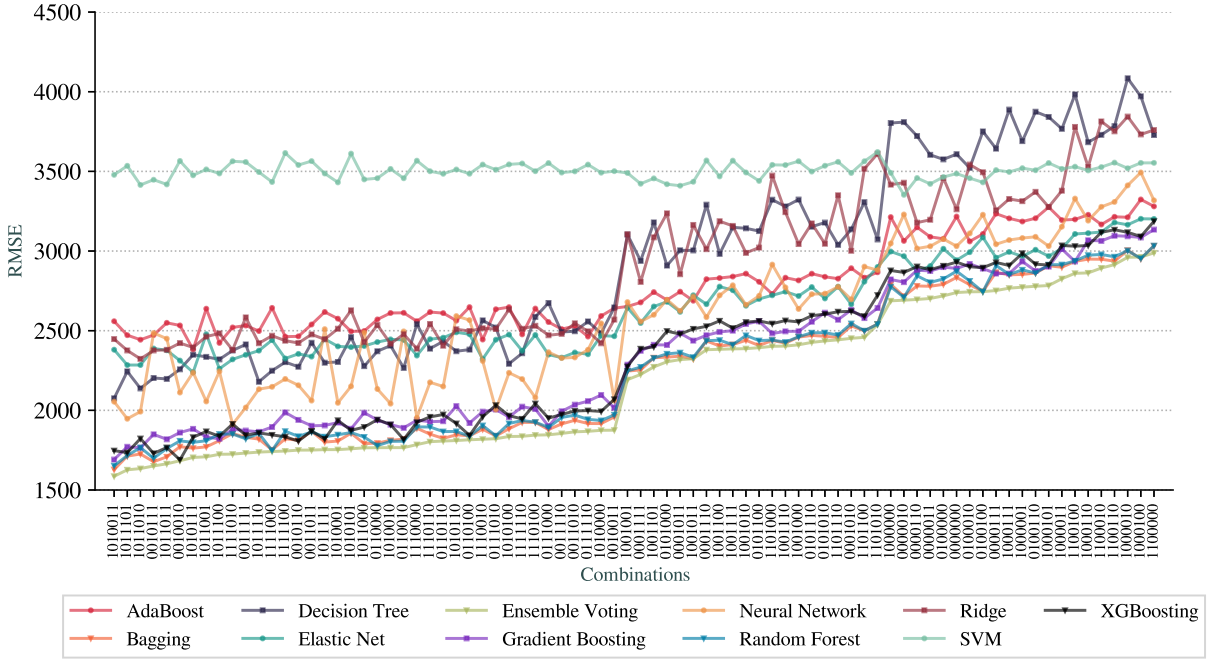


Figure 21 – RMSE for the pipelines based on combinations of adjustments



(All Dataset) activated while *Outliers Removal (Train Set)*, *Addition of AELE* and *Cross-Validation* deactivated. The best technique is the Ensemble Voting with R^2 of 0.78, RMSE of 1,586 and MAE of 803. The results for combinations for MAE are not included due to the similarity to RMSE results in terms of best and worst techniques.

We can observe that Ensemble Voting, Bagging, Random Forest, Gradient Boosting, and XGBoosting have the best result metrics as they stand at the top of the charts

for most combinations. As worse technique, we have SVM, that performed poorer than other techniques in most combinations. Another observation is that the *baseline* pipeline is better than 20 combinations in terms of prediction quality. That is, in terms of prediction quality, the pipeline without any *adjustments* has better results than other pipelines that have some of them.

From a more global analysis of Figures 20 and 21, there are two sudden changes in the prediction quality. The first happens in the middle of the graphs for RMSE and R^2 and the second appears in the third quarter. One can notice that most of the techniques exhibit this behavior of sudden change. The first change happens when the combinations no longer contain *N-Grams Extraction* (third digit in the combinations) as, before that point, all combination had this adjustment. We can also note that *Addition of AELE* (fourth digit in the combinations) starts to appear consistently in all combinations from this point.

The second sudden change happens when *Addition of AELE* stops appearing in the combinations. As we described, the best combination does not have this adjustment, but at this point, it has such an impact on the prediction quality. Thus, we can point the cause of this difference to the presence of *N-Grams Extraction*. When N-Grams is activated, the impact of *Addition of AELE* on the prediction quality is reduced and it increases in the combinations without *N-Grams Extraction*. This pattern is shared among all models except SVM, which did not perform well in any combinations.

Although different combinations of adjustments can lead to results better than the full pipeline, it is important to quantify how much each adjustment impacts the results. As demonstrated, *N-Grams Extraction* and *Addition of AELE* have a considerable impact on the prediction quality.

4.3.4.3 Impact of each adjustment on the performance

This section discusses how each adjustment impacts the prediction quality and execution time (total time to run a pipeline). We investigate how much RMSE, R^2, and execution time increase or decrease as we add or remove steps in the pipeline. To do so, we consider all the eighty combinations of pipelines we have tested, and an adjustment we want to evaluate. First, we select the combination with an adjustment bypassed, which is compared directly to the corresponding combination with the same adjustment activated.

Then, for each pair, we take the difference of the metrics. Finally, to estimate the overall influence of the adjustment on the results, the average plus/minus the standard deviation of the differences is considered. By doing this, we can determine how much a specific adjustment influences the results since it is the only difference. Table 10 contains the results for RMSE and Table 11 the results for R^2.

Each value represents the average plus/minus standard deviation of how much

the adjustment increased/decreased the metric in the technique. We also highlighted the values in which the adjustment has its biggest impact. For instance, in average, including the *Addition of AELE* decreased the RMSE in 304 ± 365 in Decision Trees. We can notice that the impact can have a significant variance according to the combinations, since standard deviation shows how much the adjustments can *influence* each other's impacts.

Table 10 – Impact of adjustments on RMSE

Technique	Feature Selection	Addition of AELE	Cross-Validation	N-Grams Extraction	Overfitting Avoidance	Outliers Removal (Train)	Outliers Removal (All Dataset)
AB	4 ± 61	-250 ± 149	6 ± 52	-447 ± 161	-21 ± 56	**-35 ± 48**	-93 ± 83
BG	12 ± 99	-219 ± 255	6 ± 80	-816 ± 258	0 ± 78	20 ± 82	-104 ± 96
DT	-40 ± 156	**-304 ± 365**	18 ± 111	**-1078 ± 359**	**-168 ± 107**	47 ± 107	-90 ± 139
EN	75 ± 78	-222 ± 124	15 ± 64	-493 ± 139	8 ± 65	25 ± 56	-54 ± 61
EV	17 ± 107	-222 ± 228	11 ± 84	-829 ± 229	7 ± 88	31 ± 84	-93 ± 95
GB	9 ± 129	-236 ± 244	17 ± 88	-811 ± 240	39 ± 107	43 ± 100	-86 ± 115
NN	**-57 ± 255**	-279 ± 210	5 ± 95	-742 ± 316	21 ± 119	87 ± 98	-31 ± 104
RF	9 ± 95	-220 ± 257	5 ± 79	-811 ± 260	28 ± 74	15 ± 81	-108 ± 91
RG	206 ± 209	-181 ± 188	**-23 ± 99**	-860 ± 249	-1 ± 105	125 ± 125	-52 ± 98
SVM RBF	21 ± 47	33 ± 54	10 ± 42	11 ± 57	-10 ± 56	30 ± 58	-32 ± 65
XGB	37 ± 119	-244 ± 234	8 ± 84	-873 ± 233	27 ± 99	29 ± 81	**-117 ± 101**

Table 11 – Impact of adjustments on R^2

Technique	Feature Selection	Addition of AELE	Cross-Validation	N-Grams Extraction	Overfitting Avoidance	Outliers Removal (Train)	Outliers Removal (All Dataset)
AB	0.00 ± 0.02	0.12 ± 0.08	0.00 ± 0.02	0.22 ± 0.08	0.01 ± 0.02	**0.02 ± 0.02**	0.02 ± 0.02
BG	-0.01 ± 0.03	0.10 ± 0.11	0.00 ± 0.02	0.32 ± 0.11	0.00 ± 0.02	-0.01 ± 0.03	0.02 ± 0.03
DT	**0.02 ± 0.07**	**0.18 ± 0.20**	-0.01 ± 0.05	**0.55 ± 0.20**	**0.09 ± 0.06**	-0.03 ± 0.05	0.01 ± 0.07
EN	-0.03 ± 0.04	0.10 ± 0.06	-0.01 ± 0.02	0.23 ± 0.07	0.00 ± 0.02	-0.01 ± 0.02	0.00 ± 0.02
EV	-0.01 ± 0.04	0.10 ± 0.10	0.00 ± 0.03	0.31 ± 0.10	0.00 ± 0.03	-0.01 ± 0.03	0.02 ± 0.03
GB	-0.01 ± 0.05	0.11 ± 0.11	-0.01 ± 0.03	0.33 ± 0.11	-0.02 ± 0.04	-0.02 ± 0.04	0.01 ± 0.04
NN	0.01 ± 0.11	0.14 ± 0.11	0.00 ± 0.04	0.34 ± 0.15	0.00 ± 0.06	-0.04 ± 0.04	-0.01 ± 0.05
RF	-0.01 ± 0.03	0.10 ± 0.11	0.00 ± 0.02	0.32 ± 0.11	-0.01 ± 0.02	-0.01 ± 0.03	0.02 ± 0.03
RG	-0.11 ± 0.12	0.10 ± 0.10	**0.01 ± 0.05**	0.44 ± 0.14	0.00 ± 0.05	-0.07 ± 0.07	0.00 ± 0.04
SVM RBF	-0.01 ± 0.01	-0.02 ± 0.02	0.00 ± 0.01	0.01 ± 0.02	0.01 ± 0.02	-0.01 ± 0.02	-0.01 ± 0.02
XGB	-0.02 ± 0.04	0.11 ± 0.11	0.00 ± 0.03	0.35 ± 0.11	-0.01 ± 0.04	-0.01 ± 0.03	**0.03 ± 0.04**

From Tables 10 and 11, we confirm the observations from the previous sections about the sudden changes in which the combinations stopped to have *N-Grams Extraction* or *Addition of AELE*. Here, we measure the impact of these two adjustments and their impact on the prediction quality.

In terms of techniques, the adjustments have more significant impact on the Decision Tree and least impact on SVM. Tree-based methods, such as XGBoosting, Random Forest and Bagging, also performed significantly better with the application of the adjustments. Tables 10 and 11 also show that *Feature Selection* has a small negative impact.

Regarding the *execution time*, Table 12 contains the results. It presents the average execution time plus/minus standard deviation of the whole pipeline. The value is based on the ratio between the execution time when the adjustment is activated and when it is bypassed. Thus, if we have a positive percentage, the activated adjustment increases the execution time, and it decreases when the percentage is negative. We do not show absolute values of time due to the differences in the available computer power in the market.

Table 12 – Percentage impact of adjustments on execution time

Feature Selection	**Addition of AELE**	**Cross-Validation**	**N-Grams Extraction**	**Overfitting Avoidance**	**Outliers Removal (Train)**	**Outliers Removal (All Dataset)**
-91.7% $\pm$ 5.0	48.1% $\pm$ 35.6	371.4% $\pm$ 49.5	29.6% $\pm$ 30.7	-75.8% $\pm$ 11.9	-10.8% $\pm$ 5.5	-11.0% $\pm$ 5.5

Even though *Feature Selection* leads to worse results, when it is activated, the execution time decreases by more than ninety per cent. Something similar happens to *Overfitting Avoidance* since it also has little impact, but the execution time reduces by more than seventy five per cent. Therefore, there is a trade-off between execution time and prediction quality we have to choose. It pays off to have *Feature Selection* and *Overfitting Avoidance* adjustments in our pipeline, since when we have both bypassed executing the whole pipeline took hours in our experiments, while it took minutes in the opposite situation. Reducing the sizes of the input and the models also reduce the execution time.

Although we can see that *Cross-Validation* also has little impact on the prediction quality in the regression task, it increases the execution time almost five times. As presented in Section 2.1, cross-validation tends to produce results with less bias since it generates five different combinations to train and test our models. But, in this case, we do not see this effect. The twenty five repetitions we run without *Cross-Validation* produced different combinations of train and test sets with a good amount of variability to capture a good estimate of the model's prediction quality. Thus, the five-fold extra combinations from cross-validation do not impact the results significantly in our experiments. Except for the time execution, which took almost five times longer.

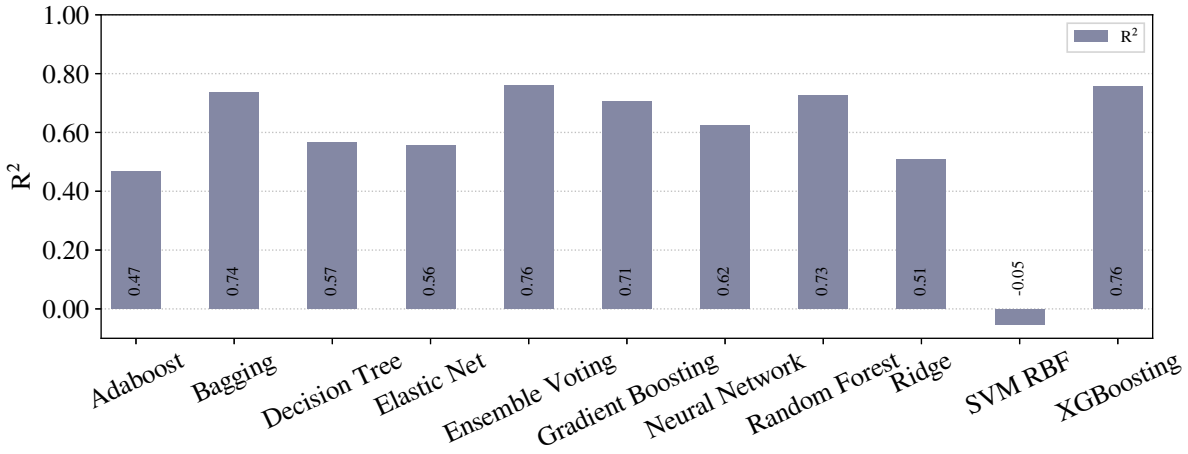
Outliers Removal also do not impact significantly the results. But we can see that the two approaches, that is, removing outliers in the train data or the whole dataset,

influence the results in different ways. While the former tends to lead to worse results, the latter leads to better results. When we remove outliers from train data and keep them in the test data, the models make poor predictions for the outliers but, if we keep our entire dataset away from anomalies, the models get better prediction quality results.

We also note that *N-Grams Extraction* and *Addition of AELE*, in terms of execution time, tend to impact negatively the pipeline, as shown in Table 12. Still, the improvement in the pipeline results overlaps this additional execution time when considering the high gains in prediction quality on adding these adjustments.

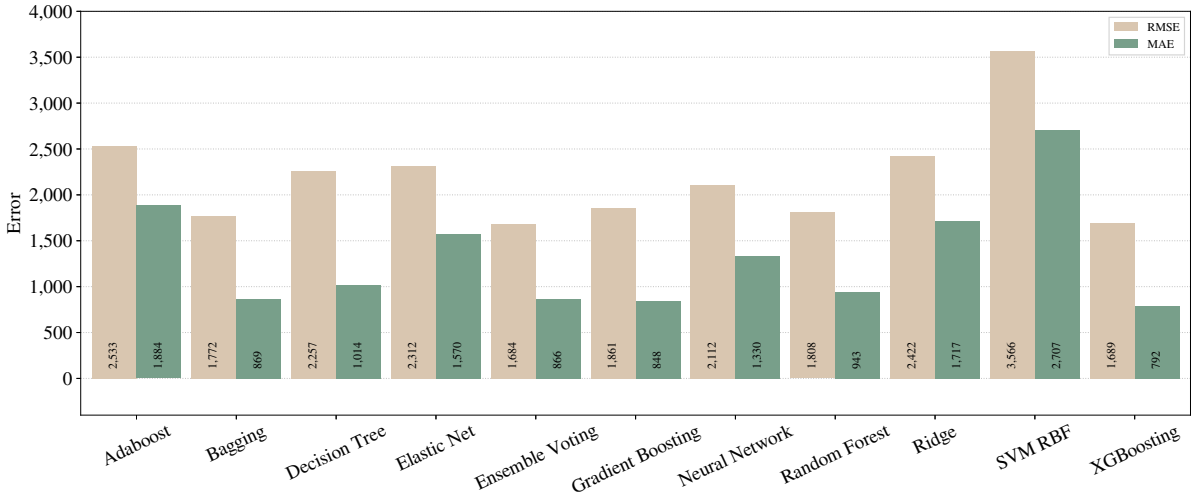
Considering prediction quality and execution time, the best combination is 1011011, with the adjustments *Feature Selection*, *N-Grams Extraction*, *Addition of AELE*, *Overfitting Avoidance* and *Outliers Removal* from all dataset activated and the remaining deactivated. This is the fifth combination from Figures 20 and 21. In terms of prediction quality, the best models are Ensemble Voting and XGBoosting, where the former has a RMSE of 1,684, a MAE of 866 and R^2 of 0.76, and the latter has a RMSE of 1,689, a MAE of 792 and R^2 of 0.76, as shown in Figure 22 and Figure 23.

Figure 22 – Results for R^2 from the best pipeline



Finally, recalling the research question and based on the discussion above, the prediction of compensation can be *accurate* by using the combination 1011011, when *Feature Selection*, *N-Grams Extraction*, *Addition of AELE*, *Overfitting Avoidance* and *Outliers Removal* from all dataset are activated and the remaining deactivated. Based on the evaluation of the results, the legal expert's experience shows that the predictions are *helpful* in the legal environment and can encourage the parties involved (consumer and airline) in an agreement. The MAE error of the best pipeline was 792.00. That way, giving up approximately 1,000 Brazilian Reais of the compensation is acceptable in conciliation hearings (an initial lawsuit stage in which the parties try to negotiate to solve the case themselves). For example, the consumer who will earn R$ 5,000 only at

Figure 23 – Results for MAE and RMSE from the best pipeline



the end of the lawsuit, will agree more easily to being compensated in R$ 4,000 in the beginning, so the case is closed immediately.

5 FINAL REMARKS

In this chapter, we conclude this work by recalling the research question, the objective, and the steps for their achievement. We also present the contributions from this research. Beyond contributions, it is highlighted the limitations faced during the executions of this research. Finally, there is the description of possibilities of future work to further explore and improve the experiments and results achieved as well as perspective of relevant research in the areas of study.

5.1 CONCLUSIONS

This work proposes the application of TM and ML techniques in legal judgments from the JEC at UFSC to predict the possible results and the amount of compensation of immaterial damages. Thus, the research aims at contributing to the increase of agreements in conciliation hearings in the JEC at UFSC.

It is noteworthy that the conclusions presented in this work are limited to judgments regarding failures in air transport services judged in the JEC of UFSC. Thus, the use of the ML models trained in this work are limited to that court and that context. However, the word embeddings models are the exception, as they are trained on texts from several courts.

The research question from this work relates to whether it is possible to predict the result of a legal judgment based on its content and predict the amount of compensation for immaterial damage using ML and TM techniques. To answer the question, it is divided into three parts, considering three distinct ML tasks: representation, classification, and regression.

Regarding the question on representation, the research evaluates the (in)existence of an impact of the specificity and the size of the corpora, used to train word embeddings, in the performance of a text classification task.

We notice an improvement in having word embeddings trained with texts similar to those used in the classification task. There is also an improvement on performance when increasing the corpus size, however until a certain point. We concluded that it is not required to have a corpus with billions of tokens for embeddings training to achieve good results in the classification of judgments from JEC. A corpus with 100 million tokens related to air transport produced the best results in our experiments.

Concerning the question about classification, we evaluate and compare the performance of Classical ML and DL techniques in the classification of judgments from JEC at UFSC. We conclude that the application of the judgments prediction in the JEC at UFSC requires the use of the text from the facts and the applicable law, without results, i.e., without the final judgment. Only the facts and applicable law are available in the early stages of the legal case. Therefore, the use of Classical ML techniques yield

the best results. However, in the case of applying text classification for to organize the judgments according to their labels, using the complete judgments, CNN with Glove embeddings would bring the best performance.

About the question on regression, we aim at evaluating whether the prediction of compensation for immaterial damage can be accurate and helpful in the legal environment using regression techniques. From the testing of several pipelines, we discovered that the adjustments *N-grams Extraction* and *Addiction of AELE* have the biggest impact on prediction quality, while *Feature selection*, *Cross-validation* and *Overfitting Avoidance* impact the execution time. Finally, based on the evaluation from the legal expert, the best results for MAE are accurate in terms of compensation for immaterial damage and can be helpful in the conciliation hearings as they may encourage agreements between the parties.

In response to the research question, we conclude that it is possible to predict the judgment's outcome based on the text without the result part, as the best classifier, Random Forest, achieved an accuracy of 82,2%. As for the prediction of the amount of compensation for immaterial damage, it is possible to achieve accurate results when using the set of the proposed adjustments and the regression techniques. The prediction quality achieved with them is acceptable, which facilitates the application in conciliation hearings.

We can also highlight some lessons learned during the execution of this research. Most of them relate to the subjects of study, that is, Text Mining and Machine Learning, as the researcher and the legal expert had to learn the theory, the techniques, and their implementations from the beginning. Furthermore, the researcher had the opportunity to adventure in a distinct area, the legal domain. This new knowledge will also be helpful in the daily life. In terms of acquired knowledge to the researchers, this work was very fruitful.

Another important lesson from this research is the team work. The experiments, papers, results and discussion presented here are the product of a joint work between a master's student at PGEAS (the researcher), a doctoral student in law (the legal expert), and their advisors. Contributions also came from the EGOV research group. The experience of working with researchers from distinct areas brought many opportunities of sharing methods, ideas, knowledge and others.

5.2 CONTRIBUTIONS

During the research, the experiments conducted to answer the research questions resulted in publications in journals and conferences, as follows:

SABO, Isabela Cristina; DAL PONT, Thiago Raulino; ROVER, Aires José; HÜBNER, Jomi Fred. Classificação de sentenças de Juizado Especial Cível utilizando

aprendizado de máquina. **Revista Democracia Digital e Governo Eletrônico**, v. 1, n. 18, p. 94–106, 2019.

DAL PONT, Thiago Raulino; SABO, Isabela Cristina; HÜBNER, Jomi Fred;ROVER, Aires José. Impact of Text Specificity and Size on Word Embeddings Performance: An Empirical Evaluation in Brazilian Legal Domain. In: CERRI, Ricardo; PRATI, Ronaldo C (Eds.) **Intelligent Systems**. Cham: Springer International Publishing, 2020. p. 521–535. DOI: 10.1007/978-3-030-61377-8_36.

SABO, Isabela Cristina; DAL PONT, Thiago Raulino; WILTON, Pablo Ernesto Vigneaux; ROVER, Aires José; HÜBNER, Jomi Fred. Clustering of Brazilian legal judgments about failures in air transport service: an evaluation of different approaches. **Artificial Intelligence and Law**, Springer Netherlands, n. 0123456789, p. 1–37, Apr. 2021. ISSN 0924-8463. DOI:10.1007/s10506-021-09287-3.

DAL PONT, Thiago Raulino; SABO, Isabela Cristina; HÜBNER, Jomi Fred; ROVER, Aires José. Regression applied to legal judgments to predict compensation for immaterial damage. **Natural Language Engineering**, 2021. **(Prelo)**

The implemented code for each the publications is freely available in repositories, as follows:

- Experiments regarding Text Classification with Classical ML and DL (`https://github.com/thiagordp/text_classification_in_legal_docs`)
- Experiments regarding Word Embeddings in Portuguese (`https://github.com/thiagordp/embeddings_in_law_paper`)
- Experiments regarding Clustering (`https://github.com/thiagordp/clustering_jec`)
- Experiments regarding Text Regression (`https://github.com/thiagordp/text_regression_in_law_judgments`)

Considering the published works and the experiments applied, the contributions to the state of the art in ML applied to legal texts are as follows:

- Pre-trained word embeddings models for Brazilian legal texts, as there were no representations available until then.
- New application of regression in legal textual data.
- Impact of adjustments in the pipeline for regression in judgments from JEC at UFSC.
- Performance of Classical ML and DL techniques in the classification of legal judgments from JEC.

In terms of practical contributions, the models and pipelines from this work may be adapted for the application in real legal conciliation hearings in the JEC at UFSC. A legal expert may present and explain the predictions to the parties. Thus, we expect to help them on reaching an agreement without the need of waiting for a judgment. In this way, the litigation in the JEC would decrease, contributing to faster and more efficient access for citizens to justice.

5.3 LIMITATIONS

The first limitation concern the difficulties to gather the dataset for the experiments. All the legal judgments had to be collected manually by the legal expert into the JEC. This is due to the fact that we wanted to avoid repeated judgments or judgments about a subject not fully related to failures in air transport service. We have the support of the current JEC/UFSC judge in this step. Although the Brazilian Judiciary indexes its processes according to the subject, there were changes in procedural electronic systems in this period, and the indexation may be incorrect due to human error. This is due to the fact that the lawsuit can be filed by different operators such as lawyers, consumers themselves, or by Judiciary employees. Efforts to unify data management in the Brazilian Judiciary are recent, but there is still difficulties to access data for experiments. If the datasets were unified and available under proper request, the amount of work for this research would be considerably reduced.

Another limitation for the dataset relates to possibility of sharing the data in public datasets platforms. The current law does not allow the sharing of personal data contained in the legal judgments. However, it would be interesting to create mechanisms to allow datasets sharing, keeping due care with third-party data, and allowing the experiments to be reproducible for other researchers who read the published articles.

In terms of predictions, although we achieved good results in classification and regression, the models do not provide the reasons that implied their predictions. When applying the models in a real conciliation hearing, the parties would ask how did the system come to this decision? Thus, there is the necessity of providing explanation to predictions from ML models.

5.4 FUTURE WORK

Considering the application of ML in the legal judgments from JEC, a possible improvement relates to the dataset size increasing. A larger dataset would allow future experiments regarding DL tasks that were not explored in this research, such as text summarizing, generation and others. Two possible paths to do that are the uses of Vari-

ational Auto-Encoders and Generative Adversarial Networks (KINGMA; WELLING, 2019; IQBAL; QURESHI, 2020).

In terms of text representation, new neural based representation techniques are often proposed in the literature, each of which having more and more trainable parameters. Some examples of representation techniques include: GPT-3 (BROWN et al., 2020), ELMo (PETERS et al., 2018), BERT (DEVLIN et al., 2018) and others. Thus, a first improvement in the text representation proposed in this research is the training of models for these newer techniques using the collected corpora. However, training these representations may take longer. Furthermore, considering the complexity of the written language used in the legal domain, it may yield relevant contributions the use of sophisticated representation techniques in applications like automatic judgments' generation or the *translation* of the legal text to a simpler writing, understandable to the common citizen.

Regarding the prediction of legal judgments, both classification and regression, an important improvement is the use of legal documents from the early stages of a lawsuit, that is, the complaint from the client, the arguments from the air company, and others. Considering the described limitations of acquiring this type of data, the presented experiments involved only judgments with the summary of the arguments from the parties, and the applicable law.

Despite time constraints to finish this work, in the future, we would like to adapt the pipelines used in the regression experiments to the classification task. The pipelines for the classification task using Classical ML techniques were similar to the *baseline* for regression. Thus, there is a possibility of improvements by applying the adjustments in the pipelines for classification.

Another improvement relates to how the users see the predictions from the ML models, that is, the interpretation of the decisions. The current implementation of the pipelines and techniques does not allow the user to understand the steps taken to produce the final prediction. A possible solution to overcome such limitation is the concept of Explainable Artificial Intelligence (XAI) (TJOA; GUAN, 2019; BIBAL et al., 2020). For example, considering the practical use case in a conciliation hearing. The legal expert would introduce the system and communicate its predictions on the case at hand. However, the parts could ask how the systems got such prediction. Using XAI enhancements, the system would highlight the relevant facts, the applicable law, the previous decisions in similar cases, and finally, its predictions.

REFERENCES

AGGARWAL, Charu C. **Machine Learning for Text**. Cham: Springer International Publishing, 2018. ISBN 9783319735306. DOI: 10.1007/978-3-319-73531-3.

AGGARWAL, Charu C.; ZHAI, Cheng Xiang. **Mining Text Data**. Ed. by Charu C. Aggarwal and ChengXiang Zhai. Boston, MA: Springer US, 2012. v. 9781461432, p. 1–522. ISBN 978-1-4614-3222-7. DOI: 10.1007/978-1-4614-3223-4.

AIROLA, Antti; PAHIKKALA, Tapio; WAEGEMAN, Willem; DE BAETS, Bernard; SALAKOSKI, Tapio. An experimental comparison of cross-validation techniques for estimating the area under the ROC curve. **Computational Statistics & Data Analysis**, Elsevier B.V., v. 55, n. 4, p. 1828–1844, Apr. 2011. ISSN 01679473. DOI: 10.1016/j.csda.2010.11.018.

ALAMI, Nabil; MEKNASSI, Mohammed; EN-NAHNAHI, Noureddine. Enhancing unsupervised neural networks based text summarization with word embedding and ensemble learning. **Expert Systems with Applications**, Elsevier BV, v. 123, p. 195–211, June 2019. DOI: 10.1016/j.eswa.2019.01.037.

ALETRAS, Nikolaos; TSARAPATSANIS, Dimitrios; PREOŢIUC-PIETRO, Daniel; LAMPOS, Vasileios. Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective. **PeerJ Computer Science**, v. 2, n. 10, e93, Oct. 2016. ISSN 2376-5992. DOI: 10.7717/peerj-cs.93.

ALZUBAIDI, Laith et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. **Journal of Big Data**, Springer International Publishing, v. 8, n. 1, p. 53, Dec. 2021. ISSN 2196-1115. DOI: 10.1186/s40537-021-00444-8.

BIBAL, Adrien; LOGNOUL, Michael; STREEL, Alexandre de; FRÉNAY, Benoıt. Legal requirements on explainability in machine learning. **Artificial Intelligence and Law**, Springer Science and Business Media LLC, v. 29, n. 2, p. 149–169, July 2020. DOI: 10.1007/s10506-020-09270-4.

BIRD, Steven; LOPER, Edward. NLTK: The Natural Language Toolkit. In: PROCEEDINGS of the ACL Interactive Poster and Demonstration Sessions. Barcelona, Spain: Association for Computational Linguistics, 2004. P. 214–217. Available from: https://www.aclweb.org/anthology/P04-3031.

BOJANOWSKI, Piotr; GRAVE, Edouard; JOULIN, Armand; MIKOLOV, Tomas. Enriching Word Vectors with Subword Information. **Transactions of the Association for Computational Linguistics**, v. 5, p. 135–146, Dec. 2017. ISSN 2307-387X. DOI: 10.1162/tacl_a_00051.

BRAZ, Fabricio Ataides et al. Document classification using a Bi-LSTM to unclog Brazil's Supreme Court, 2018. arXiv: 1811.11569.

BRAZIL. **Lei No 12.105, de 16 de março de 2015**. [S.l.: s.n.], 2015. http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13105.htm. Accessed 12 jan 2020.

BREIMAN, Leo. Bagging predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, 1996. ISSN 08856125. DOI: 10.1007/bf00058655.

BREIMAN, Leo. Random Forests. **Machine Learning**, Springer Science and Business Media LLC, v. 45, n. 1, p. 5–32, 2001. DOI: 10.1023/a:1010933404324.

BREIMAN, Leo; FRIEDMAN, Jerome H; OLSHEN, Richard A; STONE, Charles J. **Classification And Regression Trees**. [S.l.]: Routledge, 2017. DOI: 10.1201/9781315139470.

BROWN, Tom B. et al. Language Models are Few-Shot Learners. **CoRR**, abs/2005.14165, 2020. arXiv: 2005.14165.

CARDOSO, Emerson F.; SILVA, Renato M.; ALMEIDA, Tiago A. Towards automatic filtering of fake reviews. **Neurocomputing**, v. 309, p. 106–116, Oct. 2018. ISSN 09252312. DOI: 10.1016/j.neucom.2018.04.074.

CARUANA, Rich. Multitask Learning. **Machine Learning**, v. 28, n. 1, p. 41–75, 1997. ISSN 08856125. DOI: 10.1023/A:1007379606734.

CATH, Corinne. Governing artificial intelligence: ethical, legal and technical opportunities and challenges. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 376, n. 2133, p. 20180080, Nov. 2018. ISSN 1364-503X. DOI: 10.1098/rsta.2018.0080.

CHAI, T.; DRAXLER, R. R. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature. **Geoscientific Model**

Development, v. 7, n. 3, p. 1247–1250, June 2014. ISSN 1991-9603. DOI: 10.5194/gmd-7-1247-2014.

CHALKIDIS, Ilias; KAMPAS, Dimitrios. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. **Artificial Intelligence and Law**, Springer Science and Business Media LLC, v. 27, n. 2, p. 171–198, Dec. 2018. DOI: 10.1007/s10506-018-9238-9.

CHANDRASHEKAR, Girish; SAHIN, Ferat. A survey on feature selection methods. **Computers & Electrical Engineering**, Elsevier BV, v. 40, n. 1, p. 16–28, Jan. 2014. DOI: 10.1016/j.compeleceng.2013.11.024.

CHEN, Tianqi; GUESTRIN, Carlos. XGBoost: A Scalable Tree Boosting System. In: PROCEEDINGS of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA: Association for Computing Machinery, 2016. (KDD '16), p. 785–794. DOI: 10.1145/2939672.2939785.

CHOCRON, Paula; PARETI, Paolo. Vocabulary Alignment for Collaborative Agents: a Study with Real-World Multilingual How-to Instructions. In: PROCEEDINGS of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-18. [S.l.]: International Joint Conferences on Artificial Intelligence Organization, July 2018. P. 159–165. DOI: 10.24963/ijcai.2018/22.

CHOLLET, François et al. **Keras**. [S.l.: s.n.], 2015. https://keras.io.

CHRISTENSEN, Hans. **HC Corpora**. [S.l.]: Kaggle, 2016. Available from: https://web.archive.org/web/20161021044006/http://corpora.heliohost.org/.

CNJ. **Justiça em Números 2020**. Ed. by CNJ. Brasília: CNJ, 2020a. P. 236.

CNJ. **Portaria N 271**. Brasília: [s.n.], 2020b. Available from: https://atos.cnj.jus.br/atos/detalhar/3613.

COHEN, Paul R. **Empirical Methods for Artificial Intelligence**. Cambridge, MA, USA: MIT Press, 1995. ISBN 0262032252.

CORTES, Corinna; VAPNIK, Vladimir. Support-vector networks. **Machine Learning**, Springer Science and Business Media LLC, v. 20, n. 3, p. 273–297, Sept. 1995. DOI: 10.1007/bf00994018.

COVER, Thomas M.; THOMAS, Joy A. **Elements of Information Theory**. [S.l.]: Wiley, Sept. 2005. P. 1–748. ISBN 9780471241959. DOI: 10.1002/047174882X.

CURY, Augusto et al. **Soluções Pacíficas de Conflitos: Para um Brasil Moderno**. 1. ed. [S.l.]: Forense, 2019. P. 250. ISBN 978-8530982089.

DAL PONT, Thiago Raulino; SABO, Isabela Cristina; HÜBNER, Jomi Fred; ROVER, Aires José. Impact of Text Specificity and Size on Word Embeddings Performance: An Empirical Evaluation in Brazilian Legal Domain. In: CERRI, Ricardo; PRATI, Ronaldo C (Eds.). **Intelligent Systems**. Cham: Springer International Publishing, 2020. P. 521–535. DOI: 10.1007/978-3-030-61377-8_36.

DAVIS, Anthony E. The Future of Law Firms (and Lawyers) in the Age of Artificial Intelligence. **Revista Direito GV**, v. 16, n. 1, 1dummt, 2020. ISSN 2317-6172. DOI: 10.1590/2317-6172201945.

DEMŠAR, Janez et al. Orange: Data Mining Toolbox in Python. **Journal of Machine Learning Research**, v. 14, n. 35, p. 2349–2353, 2013.

DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. **CoRR**, abs/1810.04805, 2018. arXiv: 1810.04805.

DEVORE, Jay L. **Probability and Statistics for Engineering and the Sciences**. 8th. [S.l.]: Brooks/Cole, 2011.

DINIZ, Maria Helena. Proteção jurıdica da existencialidade. **Revista Eletrônica Direito e Sociedade - REDES**, Centro Universitario La Salle - UNILASALLE, v. 8, n. 2, p. 181, July 2020. DOI: 10.18316/redes.v8i2.6885.

DRAPER, Norman Richard; SMITH, Harry. **Applied regression analysis**. 3. ed. New York: Wiley, 1998. (Wiley series in probability and mathematical statistics). ISBN 0471221708.

DRUCKER, Harris; SURGES, Chris J.C.; KAUFMAN, Linda; SMOLA, Alex; VAPNIK, Vladimir. Support vector regression machines. **Advances in Neural Information Processing Systems**, v. 1, p. 155–161, 1997. ISSN 10495258.

EL JELALI, Soufiane; FERSINI, Elisabetta; MESSINA, Enza. Legal retrieval as support to eMediation: matching disputant's case and court decisions. **Artificial Intelligence and Law**, v. 23, n. 1, p. 1–22, Mar. 2015. ISSN 0924-8463. DOI: 10.1007/s10506-015-9162-1.

FREEMAN, Jim; BARNETT, Vic; LEWIS, Toby. Outliers in Statistical Data. **The Journal of the Operational Research Society**, v. 46, n. 8, p. 1034, 1995. ISSN 01605682. DOI: 10.2307/3009915.

FRIEDMAN, Jerome H. Greedy function approximation: A gradient boosting machine. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 29, n. 5, Oct. 2001. DOI: 10.1214/aos/1013203451.

GAL, Yarin; GHAHRAMANI, Zoubin. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. In: PROCEEDINGS of the 30th International Conference on Neural Information Processing Systems. Barcelona, Spain: Curran Associates Inc., 2016. (NIPS'16), p. 1027–1035.

GAMBHIR, Mahak; GUPTA, Vishal. Recent automatic text summarization techniques: a survey. **Artificial Intelligence Review**, Springer Netherlands, v. 47, n. 1, p. 1–66, Jan. 2017. ISSN 0269-2821. DOI: 10.1007/s10462-016-9475-9.

GARCIA, Salvador; LUENGO, Julián; HERRERA, Francisco. **Data Preprocessing in Data Mining**. [S.l.]: Springer International Publishing, 2015. DOI: 10.1007/978-3-319-10247-4.

GUYON, Isabelle; ELISSEEFF, André. An Introduction to Variable and Feature Selection. **J. Mach. Learn. Res.**, JMLR.org, v. 3, p. 1157–1182, 2003. ISSN 1532-4435.

HAMMAMI, Eya; AKERMI, Imen; FAIZ, Rim; BOUGHANEM, Mohand. Deep Learning for French Legal Data Categorization. In: LECTURE Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). [S.l.: s.n.], 2019. P. 96–105. DOI: 10.1007/978-3-030-32065-2_7.

HARTMANN, Fabiano; SILVA, Roberta Zumblick Martins da. **Inteligência Artificial e Direito**. first. [S.l.]: Alteridade Editora, 2019. v. 1. (Direito Racionalidade e Inteligência Artificial).

HARTMANN, Nathan; FONSECA, Erick; SHULBY, Christopher; TREVISO, Marcos; RODRIGUES, Jessica; ALUISIO, Sandra. Portuguese Word Embeddings: Evaluating on Word Analogies and Natural Language Tasks. Section 3, Aug. 2017. arXiv: 1708.06025.

HASSAN, Fahad Ul; LE, Tuyen. Automated Requirements Identification from Construction Contract Documents Using Natural Language Processing. **Journal of Legal Affairs and Dispute Resolution in Engineering and Construction**, v. 12, n. 2, p. 04520009, May 2020. ISSN 1943-4162. DOI: 10.1061/(ASCE)LA.1943-4170.0000379.

HASTIE, Trevor. The Elements of Statistical Learning. **The Mathematical Intelligencer**, v. 27, n. 2, p. 83–85, 2009. ISSN 03436993.

HAWKINS, Douglas M. The Problem of Overfitting. **Journal of Chemical Information and Computer Sciences**, v. 44, n. 1, p. 1–12, Jan. 2004. ISSN 0095-2338. DOI: 10.1021/ci0342472.

HILL, Gerald; HILL, Kathleen. **Legal Dictionary - Judgment**. [S.l.: s.n.], 2021. Available from: https://dictionary.law.com/Default.aspx?selected=1056.

HIRSCHBERG, Julia; MANNING, C. D. Advances in natural language processing. **Science**, v. 349, n. 6245, p. 261–266, July 2015. ISSN 0036-8075. DOI: 10.1126/science.aaa8685.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. **Neural Computation**, v. 9, n. 8, p. 1735–1780, Nov. 1997. ISSN 0899-7667. DOI: 10.1162/neco.1997.9.8.1735.

HODGE, Victoria; AUSTIN, Jim. A Survey of Outlier Detection Methodologies. **Artificial Intelligence Review**, v. 22, n. 2, p. 85–126, Oct. 2004. ISSN 0269-2821. DOI: 10.1023/B:AIRE.0000045502.10941.a9.

HOERL, Arthur E.; KENNARD, Robert W. Ridge Regression: Applications to Nonorthogonal Problems. **Technometrics**, v. 12, n. 1, p. 69–82, 1970. ISSN 15372723. DOI: 10.1080/00401706.1970.10488635.

HUNTER, J. D. Matplotlib: A 2D graphics environment. **Computing in Science & Engineering**, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

IQBAL, Touseef; QURESHI, Shaima. The survey: Text generation models in deep learning. **Journal of King Saud University - Computer and Information Sciences**, The Authors, n. 40, p. 1–14, Apr. 2020. ISSN 13191578. DOI: 10.1016/j.jksuci.2020.04.001.

JOLLIFFE, Ian T; CADIMA, Jorge. Principal component analysis: a review and recent developments. **Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences**, v. 374, n. 2065, p. 20150202, Apr. 2016. ISSN 1364-503X. DOI: 10.1098/rsta.2015.0202.

JOSHI, Mahesh; DAS, Dipanjan; GIMPEL, Kevin; SMITH, Noah A. Movie Reviews and Revenues: An Experiment in Text Regression. In: HUMAN Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Los Angeles, California: Association for Computational Linguistics, June 2010. P. 293–296. Available from: https://www.aclweb.org/anthology/N10-1038.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 3rd. Stanford University: Draft, 2019.

JUSBRASIL. **JusBrasil. Conectando pessoas à justiça**. [S.l.], 2020. Accessed in Feb 10, 2020. Available from: https://www.jusbrasil.com.br/home.

KARYSTINOS, G.N.; PADOS, D.A. On overfitting, generalization, and randomly expanded training sets. **IEEE Transactions on Neural Networks**, v. 11, n. 5, p. 1050–1057, 2000. ISSN 10459227. DOI: 10.1109/72.870038.

KHAN, Nawsher; YAQOOB, Ibrar; HASHEM, Ibrahim Abaker Targio; INAYAT, Zakira; MAHMOUD ALI, Waleed Kamaleldin; ALAM, Muhammad; SHIRAZ, Muhammad; GANI, Abdullah. Big data: Survey, technologies, opportunities, and challenges. **Scientific World Journal**, v. 2014, January 2018, 2014. ISSN 1537744X. DOI: 10.1155/2014/712826.

KIM, Yoon. Convolutional Neural Networks for Sentence Classification. In: PROCEEDINGS of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics, Sept. 2014. P. 1746–1751.

KINGMA, Diederik P.; BA, Jimmy Lei. Adam: A method for stochastic optimization. **3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings**, p. 1–15, 2015. arXiv: 1412.6980.

KINGMA, Diederik P.; WELLING, Max. An Introduction to Variational Autoencoders. **Foundations and Trends® in Machine Learning**, v. 12, n. 4, p. 307–392, 2019. ISSN 1935-8237. DOI: 10.1561/2200000056.

KOBER, Jens; BAGNELL, J. Andrew; PETERS, Jan. Reinforcement learning in robotics: A survey. **International Journal of Robotics Research**, v. 32, n. 11, p. 1238–1274, 2013. ISSN 02783649. DOI: 10.1177/0278364913495721.

KORNILOVA, Anastasiia; BERNARDI, Lucas. Mining the Stars: Learning Quality Ratings with User-facing Explanations for Vacation Rentals. In: PROCEEDINGS of the 14th ACM International Conference on Web Search and Data Mining. New York, NY, USA: ACM, Mar. 2021. P. 976–983. DOI: 10.1145/3437963.3441812. arXiv: 2101.10737.

KOTU, Vijay; DESHPANDE, Bala. **Data Science: Concepts and Practicen**. 2nd. Cambridge, MA: Elsevier, 2019. DOI: 10.1016/c2017-0-02113-4.

KOWSARI; JAFARI MEIMANDI; HEIDARYSAFA; MENDU; BARNES; BROWN. Text Classification Algorithms: A Survey. **Information**, v. 10, n. 4, p. 150, Apr. 2019. ISSN 2078-2489. DOI: 10.3390/info10040150.

KUHN, Max; JOHNSON, Kjell. **Applied Predictive Modeling**. [S.l.]: Springer New York, 2013. DOI: 10.1007/978-1-4614-6849-3.

KUMAR, Gunupudi Rajesh; MANGATHAYARU, N.; NARASIMHA, G. Intrusion Detection Using Text Processing Techniques. In: PROCEEDINGS of the The International Conference on Engineering & MIS 2015 - ICEMIS '15. [S.l.]: ACM Press, 2015.

KUSMIERCZYK, Tomasz; NØRVÅG, Kjetil. Online Food Recipe Title Semantics: Combining Nutrient Facts and Topics. In: PROCEEDINGS of the 25th ACM International on Conference on Information and Knowledge Management. Indianapolis, Indiana, USA: Association for Computing Machinery, 2016. (CIKM '16), p. 2013–2016. DOI: 10.1145/2983323.2983897.

LAI, Siwei; LIU, Kang; HE, Shizhu; ZHAO, Jun. How to Generate a Good Word Embedding. **IEEE Intelligent Systems**, Institute of Electrical and Electronics Engineers (IEEE), v. 31, n. 6, p. 5–14, Nov. 2016. DOI: 10.1109/mis.2016.45.

LAMPOS, Vasileios; ALETRAS, Nikolaos; PREOŢIUC-PIETRO, Daniel; COHN, Trevor. Predicting and Characterising User Impact on Twitter. In: PROCEEDINGS of the 14th Conference of the European Chapter of the Association for Computational Linguistics. Gothenburg, Sweden: Association for Computational Linguistics, Apr. 2014. P. 405–413. DOI: 10.3115/v1/E14-1043.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, May 2015. ISSN 0028-0836. DOI: 10.1038/nature14539.

LEE, Mong Li; LU, Hongjun; LING, Tok Wang; KO, Yee Teng. Cleansing Data for Mining and Warehousing. In: LECTURE Notes in Computer Science. [S.l.]: Springer Berlin Heidelberg, 1999. P. 751–760. DOI: 10.1007/3-540-48309-8_70.

LEVER, Jake; KRZYWINSKI, Martin; ALTMAN, Naomi. Classification evaluation. **Nature Methods**, v. 13, n. 8, p. 603–604, Aug. 2016. ISSN 1548-7091. DOI: 10.1038/nmeth.3945.

LIU, Fei Tony; TING, Kai Ming; ZHOU, Zhi Hua. Isolation forest. **Proceedings - IEEE International Conference on Data Mining, ICDM**, p. 413–422, 2008. ISSN 15504786. DOI: 10.1109/ICDM.2008.17.

LIU, Raymond; GILLIES, Duncan F. Overfitting in linear feature extraction for classification of high-dimensional image data. **Pattern Recognition**, v. 53, p. 73–86, May 2016. ISSN 00313203. DOI: 10.1016/j.patcog.2015.11.015.

LOPER, Edward; BIRD, Steven. NLTK: The Natural Language Toolkit. In: IN Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics. [S.l.: s.n.], 2002.

MAAT, Emile de; KRABBEN, Kai; WINKELS, Radboud. Machine Learning versus Knowledge Based Classification of Legal Texts. In: PROCEEDINGS of the 2010 Conference on Legal Knowledge and Information Systems: JURIX 2010: The Twenty-Third Annual Conference. NLD: IOS Press, 2010. P. 87–96.

MAATEN, Laurens van der; HINTON, Geoffrey. Visualizing Data using t-SNE. **Journal of Machine Learning Research**, v. 9, n. 86, p. 2579–2605, 2008.

MANCUSO, Rodolfo de Camargo. **A Resolução dos Conflitos e a Função Judicial no Contemporâneo Estado de Direito**. 3. ed. Belo Horizonte: Juspodivm, 2020. P. 912.

MARLESSONN. **News of the Brazilian Newspaper**. [S.l.]: Kaggle, 2019. Available from: https://www.kaggle.com/marlesson/news-of-the-site-folhauol.

MCKINNEY, Wes. Data Structures for Statistical Computing in Python. In: WALT, Stéfan van der; MILLMAN, Jarrod (Eds.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. P. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

MEDVEDEVA, Masha; VOLS, Michel; WIELING, Martijn. Using machine learning to predict decisions of the European Court of Human Rights. **Artificial Intelligence and Law**, 2019. ISSN 15728382. DOI: 10.1007/s10506-019-09255-y.

MENDES-MOREIRA, João; SOARES, Carlos; JORGE, Alípio Mário; SOUSA, Jorge Freire De. Ensemble approaches for regression. **ACM Computing Surveys**, v. 45, n. 1, p. 1–40, Nov. 2012. ISSN 0360-0300. DOI: 10.1145/2379776.2379786.

MENG, Qinxue; CATCHPOOLE, Daniel; SKILLICORN, David; KENNEDY, Paul J. Relational Autoencoder for Feature Extraction. **Proceedings of the International Joint Conference on Neural Networks**, 2017-May, p. 364–371, Feb. 2018. DOI: 10.1109/IJCNN.2017.7965877.

MIKOLOV, Tomas; CHEN, Kai; CORRADO, Greg; DEAN, Jeffrey. Efficient Estimation of Word Representations in Vector Space. **1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings**, p. 1–12, Jan. 2013. arXiv: 1301.3781.

MITCHELL, Thomas M. **Machine Learning**. 1. ed. [S.l.]: McGraw-Hill Education, 1997. ISBN 9780070428072.

MOHAMMED, Shapol M.; JACKSI, Karwan; ZEEBAREE, Subhi R. M. Glove Word Embedding and DBSCAN algorithms for Semantic Document Clustering. In: 2020 International Conference on Advanced Science and Engineering (ICOASE). [S.l.]: IEEE, Dec. 2020. DOI: 10.1109/icoase51841.2020.9436540.

MORGAN, S. Philip; TEACHMAN, Jay D. Logistic Regression: Description, Examples, and Comparisons. **Journal of Marriage and the Family**, JSTOR, v. 50, n. 4, p. 929, Nov. 1988. DOI: 10.2307/352104.

OBRIEN, Larry; GILLEY, Sheri; COULTER, David; LU, Peter; BROCKSCHMIDT, Kraig; MARTIN, Avatar Olivier. **What are Machine Learning Pipelines**. [S.l.: s.n.], 2021. https://docs.microsoft.com/en-us/azure/machine-learning/concept-ml-pipelines. Accessed 10 jun 2021.

OSIŃSKI, Stanisław; STEFANOWSKI, Jerzy; WEISS, Dawid. Lingo: Search Results Clustering Algorithm Based on Singular Value Decomposition. In: KŁOPOTEK, Mieczysław A.; WIERZCHOŃ, Sławomir T.; TROJANOWSKI, Krzysztof (Eds.). **Intelligent Information Processing and Web Mining**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. P. 359–368.

PEARSON, Egon S. Bayes' Theorem, Examined in the Light of Experimental Sampling. **Biometrika**, JSTOR, v. 17, n. 3/4, p. 388, Dec. 1925. DOI: 10.2307/2332088.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PENG, Hao; LI, Jianxin; HE, Yu; LIU, Yaopeng; BAO, Mengjiao; WANG, Lihong; SONG, Yangqiu; YANG, Qiang. Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN. In: PROCEEDINGS of the 2018 World Wide Web Conference. Lyon, France: International World Wide Web Conferences Steering Committee, 2018. (WWW '18), p. 1063–1072.

PENNINGTON, Jeffrey; SOCHER, Richard; MANNING, Christopher. Glove: Global Vectors for Word Representation. In: 5. PROCEEDINGS of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg, PA, USA: Association for Computational Linguistics, 2014. P. 1532–1543.

PETERS, Matthew; NEUMANN, Mark; IYYER, Mohit; GARDNER, Matt; CLARK, Christopher; LEE, Kenton; ZETTLEMOYER, Luke. Deep Contextualized Word Representations. In: PROCEEDINGS of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). Stroudsburg, PA, USA: Association for Computational Linguistics, 2018. P. 2227–2237. DOI: 10.18653/v1/N18-1202.

PORTER, Martin F. An algorithm for suffix stripping. **Program**, MCB UP Ltd, v. 14, n. 3, p. 130–137, 1980.

QUINLAN, J.R. Induction of Decision Trees. **Machine Learning**, Springer Science and Business Media LLC, v. 1, n. 1, p. 81–106, 1986. DOI: 10.1023/a:1022643204877.

QUINTANILLA, Luis et al. **Machine learning tasks in ML.NET**. [S.l.: s.n.], 2015. https://docs.microsoft.com/en-us/dotnet/machine-learning/resources/tasks. Accessed 10 jun 2021.

RAMAMOORTHY, C V; LI, H F. Pipeline Architecture. **ACM Computing Surveys**, Association for Computing Machinery, New York, NY, USA, v. 9, n. 1, p. 61–102, Mar. 1977. ISSN 0360-0300. DOI: 10.1145/356683.356687.

ŘEHŮŘEK, Radim; SOJKA, Petr. Software Framework for Topic Modelling with Large Corpora. English. In: PROCEEDINGS of the LREC 2010 Workshop on New Challenges for NLP Frameworks. Valletta, Malta: ELRA, May 2010. P. 45–50.

RODRIGUES, Ruan Chaves; RODRIGUES, Jéssica; CASTRO, Pedro Vitor Quinta de; SILVA, Nádia Felix Felipe da; SOARES, Anderson. Portuguese Language Models and Word Embeddings: Evaluating on Semantic Similarity Tasks. In: LECT Notes Comput Sc. [S.l.]: Springer International Publishing, 2020. P. 239–248.

RUSSELL, Stuart; NORVIG, Peter. **Artificial Intelligence: A Modern Approach**. [S.l.]: Pearson, 2020. ISBN 978-0-13-604259-4.

SABO, Isabela Cristina; DAL PONT, Thiago Raulino; ROVER, Aires José; HÜBNER, Jomi Fred. Classificação de sentenças de Juizado Especial Cível utilizando aprendizado de máquina. **Revista Democracia Digital e Governo Eletrônico**, v. 1, n. 18, p. 94–106, 2019.

SABO, Isabela Cristina; DAL PONT, Thiago Raulino; WILTON, Pablo Ernesto Vigneaux; ROVER, Aires José; HÜBNER, Jomi Fred. Clustering of Brazilian legal judgments about failures in air transport service: an evaluation of different approaches. **Artificial Intelligence and Law**, Springer Netherlands, n. 0123456789, p. 1–37, Apr. 2021. ISSN 0924-8463. DOI: 10.1007/s10506-021-09287-3.

SADIKU, Asmir. Immaterial Damage and Some Types of its Compensation. **Prizren Social Science Journal**, Prizren Social Science Journal, v. 4, n. 1, p. 50–56, Apr. 2020. DOI: 10.32936/pssj.v4i1.142.

SALUNKE, Sagar Shivaji. **Selenium Webdriver in Python: Learn with Examples**. 1st. North Charleston, SC, USA: CreateSpace Independent Publishing Platform, 2014. ISBN 1497337364.

SAMUEL, A L. Some Studies in Machine Learning Using the Game of Checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210–229, July 1959. ISSN 0018-8646. DOI: 10.1147/rd.33.0210.

SANTOS, Henrique; WOLOSZYN, Vinicius; VIEIRA, Renata. BlogSet-BR: A Brazilian Portuguese Blog Corpus. In: PROCEEDINGS of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). Miyazaki, Japan: European Language Resources Association (ELRA), 2018.

SCHAPIRE, Robert E. A Brief Introduction to Boosting. In: PROCEEDINGS of the 16th International Joint Conference on Artificial Intelligence - Volume 2. Stockholm, Sweden: Morgan Kaufmann Publishers Inc., 1999. (IJCAI'99), p. 1401–1406.

SCHMIDHUBER, Jürgen. Deep learning in neural networks: An overview. **Neural Networks**, Elsevier Ltd, v. 61, p. 85–117, Jan. 2015. ISSN 08936080. DOI: 10.1016/j.neunet.2014.09.003. arXiv: 1404.7828.

SEBASTIANI, Fabrizio. Machine learning in automated text categorization. **ACM Computing Surveys**, v. 34, n. 1, p. 1–47, Mar. 2002. ISSN 0360-0300. DOI: 10.1145/505282.505283. eprint: 0110053 (cs).

SENGUPTA, Saptarshi; BASAK, Sanchita; SAIKIA, Pallabi; PAUL, Sayak; TSALAVOUTIS, Vasilios; ATIAH, Frederick; RAVI, Vadlamani; PETERS, Alan. A review of deep learning with special emphasis on architectures, applications and recent trends. **Knowledge-Based Systems**, Elsevier B.V., v. 194, p. 105596, Apr. 2020. ISSN 09507051. DOI: 10.1016/j.knosys.2020.105596. arXiv: 1905.13294.

SHEIKHALISHAHI, Seyedmostafa; MIOTTO, Riccardo; DUDLEY, Joel T; LAVELLI, Alberto; RINALDI, Fabio; OSMANI, Venet. Natural Language Processing of Clinical Notes on Chronic Diseases: Systematic Review. **JMIR Medical Informatics**, JMIR Publications Inc., v. 7, n. 2, e12239, Apr. 2019.

SHI, Xingjian; CHEN, Zhourong; WANG, Hao; YEUNG, Dit-Yan; WONG, Wai-kin; WOO, Wang-chun. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. **Advances in Neural Information Processing Systems**, 2015-Janua, p. 802–810, June 2015. ISSN 10495258. arXiv: 1506.04214.

SILVA, Nilton. Notas iniciais sobre a evolução dos algoritmos do Victor: o primeiro projeto de inteligência artificial em supremas cortes do mundo. In: FERNANDES, Ricardo Vieira de Carvalho; CARVALHO, Angelo Gamba Prata de (Eds.). **II Congresso Internacional de Direito, Governo e Tecnologia**. 1. ed. Belo Horizonte: Fórum, 2018. chap. 3, p. 89–94.

SILVA, Nilton et al. Document type classification for Brazil's supreme court using a Convolutional Neural Network. In: PROCEEDINGS of The Tenth International Conference on Forensic Computer Science and Cyber Law. [S.l.]: HTCIA, Oct. 2018. P. 7–11. DOI: 10.5769/C2018001.

SILVER, David et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, p. 1–19, Dec. 2017. arXiv: 1712.01815.

SMYWIŃSKI-POHL, Aleksander; LASOCKI, Karol; WRÓBEL, Krzysztof; STRZAŁTA, Marek. Automatic Construction of a Polish Legal Dictionary with Mappings to Extra-Legal Terms Established via Word Embeddings. In: PROCEEDINGS of the Seventeenth International Conference on Artificial Intelligence and Law - ICAIL '19. [S.l.]: ACM Press, 2019.

STF. **Supremo Tribunal Federal**. [S.l.], 2020. Accessed in Feb 15, 2020. Available from: http://portal.stf.jus.br/.

STJ. **STJ - Jurisprudência do STJ**. [S.l.], 2020. Accessed in Feb 05, 2020. Available from: https://scon.stj.jus.br/SCON/.

ŞULEA, Octavia Maria; ZAMPIERI, Marcos; VELA, Mihaela; VAN GENABITH, Josef. Predicting the law area and decisions of French supreme court cases. In: INTERNATIONAL Conference Recent Advances in Natural Language Processing, RANLP. [S.l.: s.n.], 2017. P. 716–722. DOI: 10.26615/978-954-452-049-6-092. arXiv: 1708.01681.

TAN, Liling. **Old Newspapers**. [S.l.]: Kaggle, 2020. Available from: https://www.kaggle.com/alvations/old-newspapers.

TAN, Mingxing; LE, Quoc V. EfficientNetV2: Smaller Models and Faster Training. v. 1, p. 1–12, Apr. 2021. arXiv: 2104.00298.

TATMAN, Rachael. **Brazilian Literature Books**. [S.l.]: Kaggle, 2017. Available from: https://www.kaggle.com/rtatman/brazilian-portuguese-literature-corpus.

THEODORIDIS, Sergios; KOUTROUMBAS, Konstantinos. **Pattern Recognition**. 4. ed. Burlington: Academic Press, 2009.

TIBSHIRANI, Robert. Regression Shrinkage and Selection Via the Lasso. **Journal of the Royal Statistical Society: Series B (Methodological)**, v. 58, n. 1, p. 267–288, Jan. 1996. ISSN 00359246. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

TJOA, Erico; GUAN, Cuntai. A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI. **CoRR**, abs/1907.07374, 2019. arXiv: 1907.07374.

TJSC. **Jurisprudência Catarinense - TJSC**. [S.l.], 2020. Accessed in Feb 09, 2020. Available from: http://busca.tjsc.jus.br/jurisprudencia/.

TORRES, J.L.; GARCÍA, A.; DE BLAS, M.; DE FRANCISCO, A. Forecast of hourly average wind speed with ARMA models in Navarre (Spain). **Solar Energy**, v. 79, n. 1, p. 65–77, July 2005. ISSN 0038092X. DOI: 10.1016/j.solener.2004.09.013.

TRUSOV, Roman; NATEKIN, Alexey; KALAIDIN, Pavel; OVCHARENKO, Sergey; KNOLL, Alois; FAZYLOVA, Aida. Multi-representation approach to text regression of financial risks. **Proceedings of Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference, AINL-ISMW FRUCT 2015**, v. 7, p. 110–117, 2016. DOI: 10.1109/AINL-ISMW-FRUCT.2015.7382979.

UYSAL, Alper Kursat. An improved global feature selection scheme for text classification. **Expert Systems with Applications**, Elsevier BV, v. 43, p. 82–92, Jan. 2016. DOI: 10.1016/j.eswa.2015.08.050.

VASWANI, Ashish; SHAZEER, Noam; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Lukasz; POLOSUKHIN, Illia. Attention Is All You Need. **Advances in Neural Information Processing Systems**, 2017-Decem, Nips, p. 5999–6009, June 2017. ISSN 10495258. arXiv: 1706.03762.

VIRTUCIO, Michael Benedict L. et al. Predicting Decisions of the Philippine Supreme Court Using Natural Language Processing and Machine Learning. In: 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC). [S.l.]: IEEE, July 2018. P. 130–135. DOI: 10.1109/COMPSAC.2018.10348.

WATANABE, Kazuo. In: WATANABE, Kazuo (Ed.). **Juizado Especial de Pequenas Causas: lei n. 7.244/1984**. São Paulo: Revista dos Tribunais, 1985. Filosofia e características básicas do Juizado Especial de Pequenas Causas.

WEISS, Sholom M.; INDURKHYA, Nitin; ZHANG, Tong. **Fundamentals of Predictive Text Mining**. London: Springer London, 2010. v. 42, p. 823. (Texts in Computer Science). ISBN 978-1-84996-225-4. DOI: 10.1007/978-1-84996-226-1.

WIKIPEDIA. **PT Wiki dump progress on 20191120**. [S.l.]: Wikipedia, 2019. Available from: http://wikipedia.c3sl.ufpr.br/ptwiki/20191120/.

WITTEN, Ian H; FRANK, Eibe; HALL, Mark A. **Data Mining: Practical Machine Learning Tools and Techniques**. [S.l.]: Elsevier, 2011. P. 664. ISBN 9780123748560. DOI: 10.1016/C2009-0-19715-5.

XU, Xun; LEE, Chieh. Utilizing the platform economy effect through EWOM: Does the platform matter? **International Journal of Production Economics**, v. 227, p. 107663, 2020. ISSN 0925-5273. DOI: 10.1016/j.ijpe.2020.107663.

ZHANG, Haoyang; ZHOU, Liang. Similarity Judgment of Civil Aviation Regulations Based on Doc2Vec Deep Learning Algorithm. In: 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). [S.l.]: IEEE, Oct. 2019. P. 1–8. DOI: 10.1109/CISP-BMEI48845.2019.8965709.

ZHOU, Qiming; CHEN, Yumin. Generalization of DEM for terrain analysis using a compound method. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier B.V., v. 66, n. 1, p. 38–45, Jan. 2011. ISSN 09242716. DOI: 10.1016/j.isprsjprs.2010.08.005.

ZOU, Bin; LAMPOS, Vasileios; GORTON, Russell; COX, Ingemar J. On Infectious Intestinal Disease Surveillance Using Social Media Content. In: PROCEEDINGS of the 6th International Conference on Digital Health Conference. Montréal, Québec, Canada: Association for Computing Machinery, 2016. (DH '16), p. 157–161. DOI: 10.1145/2896338.2896372.

ZOU, Hui; HASTIE, Trevor. Regularization and variable selection via the elastic net. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, v. 67, n. 5, p. 768–768, Nov. 2005. ISSN 1369-7412. DOI: 10.1111/j.1467-9868.2005.00527.x.

APPENDIX A – SYSTEMATIC REVIEW OF THE LITERATURE: AI, ML AND LAW

The Systematic Review of the Literature, detailed in this section, focused on finding works related to the applications of TM and ML in the legal domain.

A.1 DEFINITION OF SEARCH QUESTIONS

The main question in this SRL is: "Which are the Text Mining Techniques applied in the legal domain?"

As secondary questions, there is:

- Which is the legal application?
- What are the pre-processing techniques used?
- What are the representation techniques used?
- What are the feature extraction techniques used?
- What are the classification techniques used?
- What are the clustering techniques used?
- What are the regression techniques used?
- What are the evaluation techniques used?

A.2 SEARCH STRATEGIES

In order to have an overview of the publications regarding TM and legal domain, we searched in February 29, 2020, for works published on conferences or journals from 2010 to 2020. The search used the following search string:

```
("text mining" OR "natural language processing" OR "nlp" OR
"language processing") AND ("deep learning" OR "machine learning")
AND ("classification" OR "cluster*" OR "regression" OR
"categorization" OR "embedding*" OR "representation" OR "predict*")
AND ("text" OR "document") AND ("law" OR "legal" OR "judicial" OR
"justice" OR "court" OR "legislation" OR "juridical" OR "lawful")
```

In this search, we also added synonymous for Machine Learning, Text Mining, and ML Tasks, and the legal domain.

The search embraced the papers' titles, abstracts and keywords.

A.3 KNOWLEDGE BASES

In this SRL, we included bases predominantly related to computing as well as interdisciplinary basis. The list of knowledge bases follows:

- Scopus
- IEEE Xplore
- Web of Science
- ACM Digital Library

A.4 INCLUSION AND EXCLUSION CRITERIA

Following the questions of this research, we defined a set of inclusion and exclusion criteria. The process of selection embraced the reading of title, abstract and keywords and the accordance with the criteria:

The following is the list of inclusion criteria:

- Published in journal or conference
- Involves legal processes, or texts with juridical language;
- Involves Machine Learning or Text Mining;
- The techniques used are named;
- Empirical Works.
- Published from 2010 and February 2020.

And the following is the list of exclusion criteria:

- Not written in Portuguese or English;
- Published over 10 years ago;
- Does not involve Machine Learning or Text Mining;
- Does not involve the legal domain;
- Theoretical works.

A.5 DATA EXTRACTION PLAN

Information to extract from the papers:

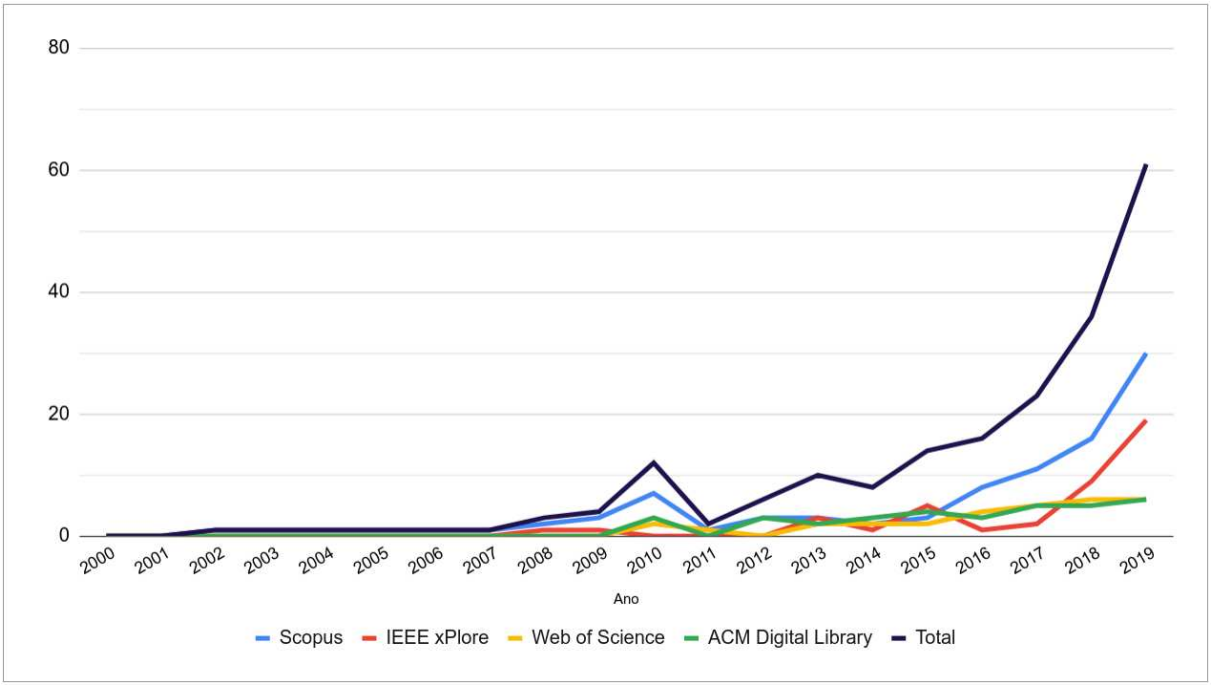
- Title
- Keywords
- Abstract
- Authors
- Year of publication
- Journal or Conference
- Authors affiliation
- Pre-processing techniques
- Representation / Feature Extraction Techniques
- Classification / Clustering / Regression Techniques

- Other techniques
- Model Evaluation Techniques

A.6 SEARCH EXECUTION AND PRELIMINARY ANALYSIS

After applying the search strings to the knowledge bases on February 29, 2020 without filters on year of publication, we obtained the number of publications shown in Figure 24.

Figure 24 – Publications on ML and TM applied to the legal domain from 2000 to 2019



After applying the search strings to the knowledge bases with time filtering on February 29, 2020, the search returned 195. After duplicates removal, the number of works reduced to 147. Finally, from the reading of title, abstract and keywords and the application of the selection criteria, the number of works selected for full reading reduced to 46.

The selected works were used as references in the Introduction chapter, in the Related Works and in the Background.

A.7 RESULTS AND ANALYSIS

In this section we show a sequence of quantitative data in terms of the tens most frequent authors, journals, text mining techniques and others.

In the following tables, there is ten most frequent authors and conferences found in the SRL and the most frequent techniques for pre-processing, representation, classification, clustering and evaluation.

Table 13 – Ten most frequent authors

Authors	Papers
Matthes, Florian	5
Glaser, Ingo	4
Chalkidis, Ilias	3
Scepankova, Elena	3
Quaresma, Paulo	2
Gonçalves, Teresa	2
Galgani, Filippo	2
Compton, Paul	2
Hoffmann, Achim	2
Androutsopoulos, Ion	2

Table 14 – Ten most frequent journals and conferences

Journal / Conference	Papers
Lecture Notes in Computer Science	8
CEUR Workshop Proceedings	4
Frontiers in Artificial Intelligence and Applications	3
Artificial Intelligence and Law	2
2010 6th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2010	1
Proceedings of the ACM Conference on Computer and Communications Security	1
Foundations and Trends in Information Retrieval	1
Conference on Legal Knowledge and Information Systems	1
Expert Systems with Applications	1
International Conference on Cloud Computing and Services Science	1

Table 15 – Ten most frequent pre-processing techniques

Preprocessing	Papers
Stop words removal	15
Lemmatization	8
Stemming	7
Tokenization	4
Lowercase	4
POS Tagging	3
Normalization	3
Remove punctuation	3
Remove noise	1
Regularization	1

Table 16 – Ten most frequent representation techniques

Representation	Papers
TF-IDF	19
Word2Vec	14
Bag of Words	10
N-Gram	7
Part-of-Speech Tag	6
Named Entity Recognition	5
Doc2Vec	4
Word Embeddings	3
FastText	3
BERT	2

Table 17 – Ten most frequent classification techniques

Classification Tech	Papers
Support Vector Machine	24
Convolution Neural Network	19
Naïve Bayes	18
Decision Tree	17
k Nearest Neighbors	14
Recurrent neural network	14
Random Forest	13
Long Short Term Memor y	13
Logistic Regression	11
Conditional Random Field	7

Table 18 – Most frequent clustering techniques

Clustering Tech	Papers
Hierarchical Clustering	2
Fuzzy C-Means	1
Hierarchical LDA	1
k-Means	1

Table 19 – Ten most frequent evaluation metrics

Evaluation Metric	**Papers**
F1-score	23
Accuracy	21
Precision	20
Recall	19
Cross-validation	6
Rouge	2
Area under curve ROC	2
ROC	1
BLEU	1

This SRL did not find works that applied regression techniques in the legal domain.

APPENDIX B – SYSTEMATIC REVIEW OF THE LITERATURE: TEXT REPRESENTATION

In this SRL, we tried to find work related to the application of text representation techniques on legal texts written in the Portuguese language. However, such task did not succeed due to the absence of work in that sense. Thus, two smaller SRLs were conducted to find papers with broader searches. The first focused on the representation of legal texts in any language and the second in the representation of general texts in Portuguese.

B.1 DEFINITION OF SEARCH QUESTIONS

The question of the first search: "What are the text representation techniques applied to texts from the legal domain?"

The question of the second search: "What are the text representation techniques applied to texts written in Portuguese?"

B.2 SEARCH STRATEGIES

Search for representation of legal texts in any languages:

```
("legal" OR "law" OR "court" OR "justice") AND ("embedding*" OR
"language model*" OR "machine learning" OR "deep learning" OR
"natural language processing" OR "text mining") AND ("doc2vec"
OR "paragraph2vec" OR "word2vec" OR "glove" OR "wang2vec" OR
"fasttext" OR "bert" OR "elmo" OR "law2vec")
```

Search for representation of general texts in Portuguese:

```
("portuguese" OR "brazil*") AND ("embedding*" OR "deep learning" OR
"machine learning" OR "natural language processing" OR "text mining")
AND ("doc2vec" OR "paragraph2vec" OR "word2vec" OR "glove" OR
"wang2vec" OR "fasttext" OR "bert" OR "elmo" OR "law2vec" )
```

B.3 KNOWLEDGE BASES

In this SRL, we searched on the following bases:

- Scopus
- ACM Digital Library
- IEEE Xplore
- Web of Science

B.4 INCLUSION AND EXCLUSION CRITERIA

Following the questions of this research, we defined a set of inclusion and exclusion criteria. The process of selection embraced the reading of title, abstract and keywords and the accordance with the criteria:

The following is the list of inclusion criteria:

- Published in journal or conference
- Involves legal texts in any languages or general texts in Portuguese;
- Involves Machine Learning or Text Mining;
- The techniques used are named;
- The work evaluate or train representations
- Empirical Work;
- Published from 2010 and May 2020.

And the following is the list of exclusion criteria:

- Work not written in Portuguese or English;
- Published over 10 years ago;
- Does not involve legal texts in any language neither general texts in Portuguese;
- Does not involve Machine Learning or Text Mining;
- Theoretic work

B.5 DATA EXTRACTION PLAN

In the SRL from this section, we focused on just retrieving the representation techniques used and the application.

B.6 SEARCH EXECUTION AND PRELIMINARY ANALYSIS

After applying the first search string to the knowledge base on May 7, 2020 for researches published from 2010 to May 2020, the search returned 52 documents. After reading title, abstract and keywords and applying the selection criteria, the number of papers reduced to 12.

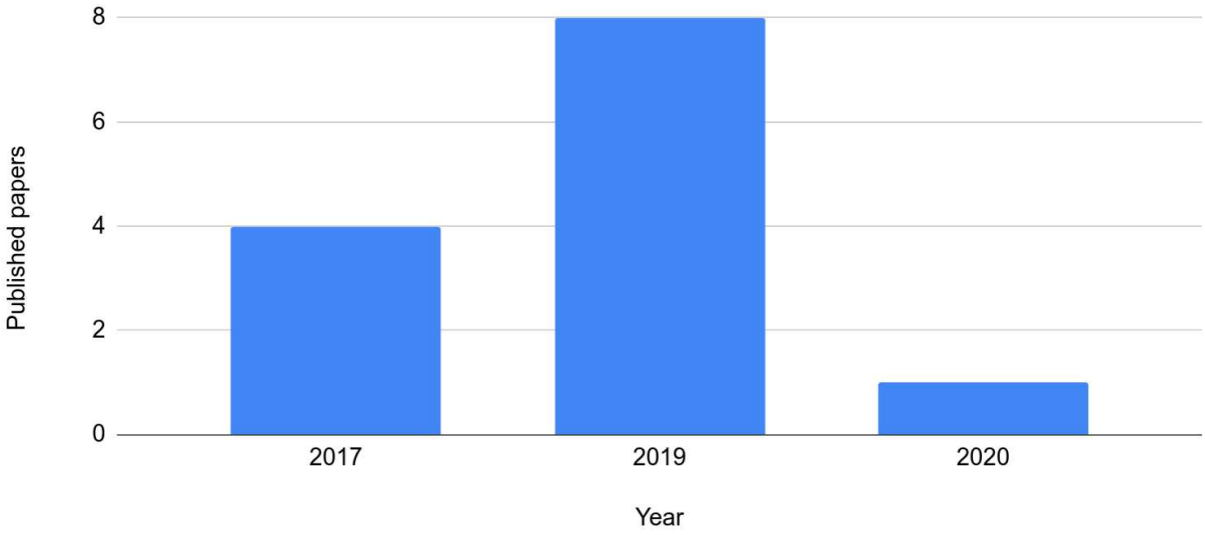
In terms of the second search string, after applying to the knowledge base on May 7, 2020 with the same time filtering, the search returned 136 documents. After reading title, abstract and keywords and applying the selection criteria, the number of papers reduced to 20.

B.7 RESULTS AND ANALYSIS

In this section, we show the results and analysis in terms of representation techniques for the first part of this SRL related to legal texts from many languages and general texts in Portuguese.

In the first part of this research, we focused on the representation techniques applied in the legal domain. In Figure 25, one can see the distribution by year of the research interest, on the area of representation of legal documents for ML tasks, considering our selection criteria. Although we set the interval to ten years, the selected work only embraced three distinct years.

Figure 25 – Researches by year for text representation in legal documents



In Table 20, there is the list of representation techniques used in the selected work. Note that, many papers reported using more than one representation techniques in their experiments.

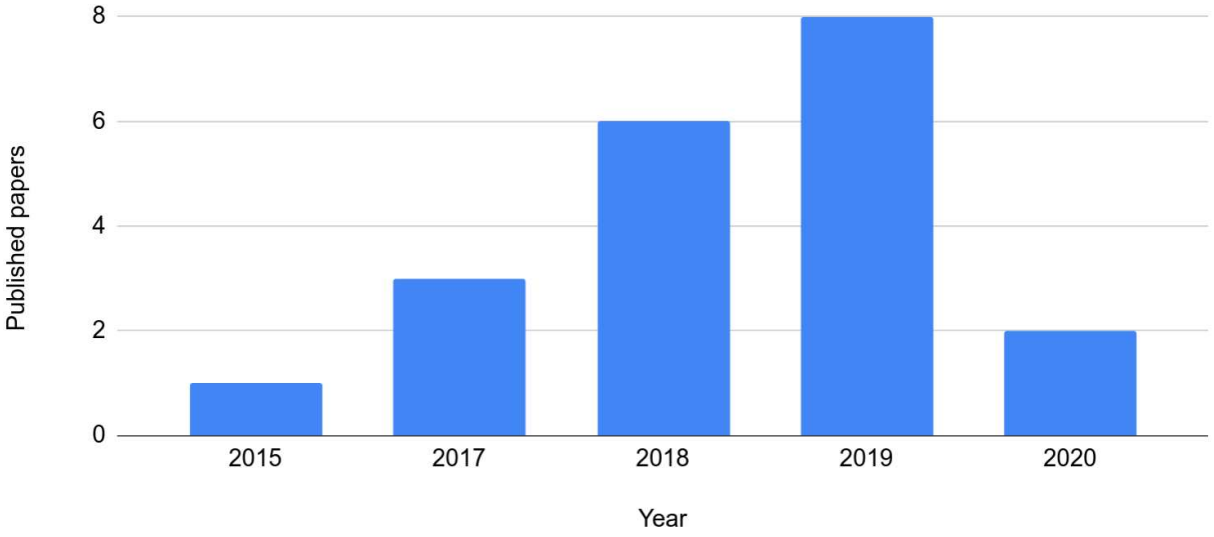
Table 20 – Representations applied to legal texts

Representation	Papers
Word2Vec	8
Doc2Vec	2
WordVec CBOW	2
Glove	2
FastText	2
ELMo	1
Bag of Words	1
Law2Vec	1

In the second part of this research, we focused on the representation techniques applied in general texts written in the Portuguese language. In Figure 26, one can see

the distribution by year of the research interest on the area, considering our selection criteria. Although, the SRL embraced the last ten years the selected work embraced the last five.

Figure 26 – Researches by year for text representation in Portuguese documents



In Table 21, there is the list of representation techniques used in the selected work. Note that, many papers reported using more than one representation techniques in their experiments.

Table 21 – Representation used in Portuguese texts

Representation	Papers
Word2Vec	7
Word2Vec Skipgram	7
Glove	6
TF-IDF	3
Wang2Vec Skipgram	3
Bag of Words	2
ELMo	2
FastText	2
FastText Skipgram	2
LDA	2

As mentioned, we do not find, until the date of the SRLs, any work related to evaluation or training of representations of legal texts in the Portuguese language.

APPENDIX C – SYSTEMATIC REVIEW OF THE LITERATURE: TEXT REGRESSION

In this SRL, we focused on finding papers related to the application of regression techniques on legal texts written in the Portuguese language. However, such a task did not succeed due to the lack of work in that matter. Thus, we applied a broader search regarding the application of regression in texts without context or language limitations.

C.1 DEFINITION OF SEARCH QUESTIONS

The question of this search is: "What are the regression techniques applied to texts considering any applications and languages?"

C.2 SEARCH STRATEGIES

The search used the following search string:

```
( "regression on text*"  OR  "text* regression"  OR  "regression text*"
OR  "regression from text*"  OR  "regression for text*" )
AND NOT  "logistic regression"
```

C.3 SEARCH RESOURCES

In this SRL, we searched on the following bases:

- Scopus
- ACM Digital Library
- IEEE Xplore
- Web of Science

C.4 SELECTION CRITERIA

Following the questions of this research, we defined a set of inclusion and exclusion criteria. The process of selection embraced the reading of title, abstract and keywords, and the accordance with the criteria.

The following is the list of inclusion criteria:

- Published in journal or conference
- Involves regression applied to textual data;
- Involves Machine Learning or Text Mining;
- The techniques used are named;
- Empirical Work;
- Published from 2010 and December 2020.

And the following is the list of exclusion criteria:

- Work not written in Portuguese or English;
- Published over 10 years ago;
- Does not involve regression applied to textual data;
- Does not involve Machine Learning or Text Mining;
- Theoretic work.

C.5 DATA EXTRACTION

In the SRL from this section, we focused on retrieving the text representation, the regression techniques used and the evaluation metrics.

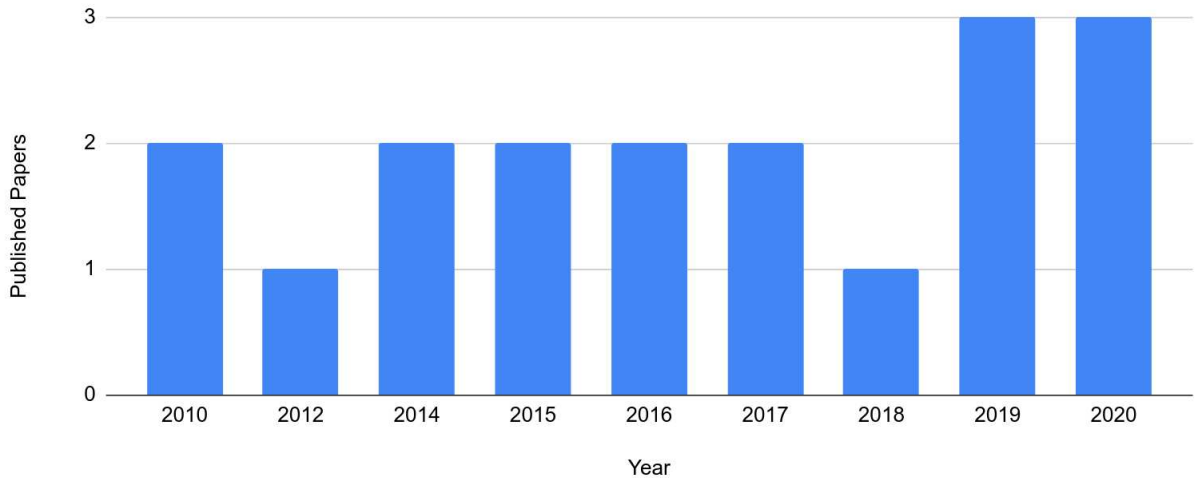
C.6 SEARCH EXECUTION AND PRELIMINARY ANALYSIS

After applying the first search string to the knowledge base on December 1, 2020 for researches published from 2010 to December 2020, the search returned 124 documents. After reading title, abstract and keywords and applying the selection criteria, the number of papers reduced to 18.

C.7 RESULTS AND DISCUSSION

In this section, we show the results and analysis in terms of regression techniques applied to text. In this part of this research, we focused on the regression techniques applied in any domains. In Figure 27, one can notice the distribution by year of the research interest on the area of the regression applied to texts.

Figure 27 – Papers published by year



In Table 22, there are the representation techniques used in the selected work from the SRL. The most common technique is the Bag of Words followed by the N-

Grams and TF-IDF, that is Vector Space Models representations. There is also neural based techniques such as word embeddings.

Table 22 – Representation techniques in papers from regression SRL

Representation	Papers
Bag of Words	8
N-gram	4
TF-IDF	4
Metadata	2
Part-of-Speech Tag	2
Word Embeddings	2
Context ependence	1
Dependency relations	1
LDA	1
LSA	1

In Table 23, there are the most frequent regression techniques applied in the selected work. The most common is the Support Vector Machine, followed by Linear Regression, Convolution Neural Network, Elastic Net, Gaussian Copula, and Gradient Boosting.

Table 23 – Regression techniques applied in the papers from SRL

Regression Tech	Papers
Support Vector Machine	7
Linear Regression	5
Convolutional Neural Network	2
Elastic Net	2
Gaussian Copula	2
Gradient Boosting	2
Conditional Generative Adversarial Network	1
Gaussian Process	1
kNN	1
Lasso	1
Multinomial logistic text regression	1
Random Forest	1
Ridge	1
XGBoosting	1

In Table 24, there are the most common evaluation metrics for regression applied in the select work. One can note the Mean Absolute Error (MAE) is the most common, followed by Mean Square Error (MSE) and Root Mean Square Error (RMSE).

Table 24 – Evaluation metrics for regression

Evaluation Metric	Papers
Mean Absolute Error	9
Mean Square Error	3
Root Mean Square Error	3
Pearson's correlation	2
Relative Absolute Error	2
Root Relative Squared Error	2
Adjusted R2	1
F-variation	1
Kendall's Tau	1
R^2	1
RAE	1
Spearman's Correlation	1
Standardization on Beta	1
Symmetric Mean Absolute Percentage Error	1
Value of t	1