UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ADMINISTRAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO

Alfredo Montelongo Flores

A DEEP LEARNING FRAMEWORK FOR CONTINGENT LIABILITIES RISK MANAGEMENT: PREDICTING BRAZILIAN LABOR COURT DECISIONS

Porto Alegre
2021

Alfredo Montelongo Flores

A DEEP LEARNING FRAMEWORK FOR CONTINGENT LIABILITIES RISK MANAGEMENT: PREDICTING BRAZILIAN LABOR COURT DECISIONS

Dissertation for Doctoral degree in
Business Administration at the School
of Administration of Federal University
of Rio Grande do Sul.
Supervisor: João Luiz Becker, PhD

Porto Alegre, 2021

Alfredo Montelongo Flores

A DEEP LEARNING FRAMEWORK FOR CONTINGENT LIABILITIES RISK MANAGEMENT: PREDICTING BRAZILIAN LABOR COURT DECISIONS

Dissertation for Doctoral degree in Business
Administration at the School of Administration
of Federal University of Rio Grande do Sul.

Porto Alegre, 2021

Alfredo Montelongo Flores

A DEEP LEARNING FRAMEWORK FOR CONTINGENT LIABILITIES RISK MANAGEMENT: PREDICTING BRAZILIAN LABOR COURT DECISIONS

Dissertation for Doctoral degree in Business Administration at the School of Administration of Federal University of Rio Grande do Sul.

Approved research. Porto Alegre, June 18, 2021:

João Luiz Becker, PhD
Dissertation Advisor

Carla Bonato Marcolin, PhD
Committee Member

Luciano Ferreira, PhD
Committee Member

Rodrigo Dalla Veccia, PhD
Committee Member

Porto Alegre, 2021

ACKNOWLEDGMENTS

This is the last page that I write of my dissertation, but I believe it is the most difficult. It has been over four years since I started my studies to get a Ph.D. degree and multiple moments pass in my head of the people who contributed to ending this stage, but I will mainly thank the following people: Professor Becker, who accepted to be my doctoral advisor and encouraged me to always perform state-of-the art research, but mainly by his great humanity and support during the difficult moments. Professor Walter Nique, who unfortunately died before I ended my Ph.D. degree but who was a crucial part during all my post-graduate studies, thanks for all, Nique. My girlfriend Anaí, whom I met at the start of my Ph.D. studies and always go along with me during the good and bad moments. My brother Yunuen, who is an example for me, always helps me in difficult moments and encourages me never to give up. My mother, Lourdes who every time has supported me on my life decisions. All my colleagues from faculty, Rafa, Afonso, Lucas that suffered along with me in the process of a Ph.D. To George and Graciela who guided me to understand how a petition process is performed. I would particularly like to thank the people from the 4TRT court, Jorge Alberto, Natacha and Andre, who kindly provided me the dataset for the project and the people from organizations that brought me an interview as part of my dissertation. To professor members of the examination board Carla, Luciano and Rodrigo for their comments and time dedicated to this dissertation work. Finally, to the Brazilian Government for bringing me the opportunity of having an education of quality.

ABSTRACT

Estimating the likely outcome of a litigation process is crucial for many organizations. A specific application is the "Contingents Liabilities," which refers to liabilities that may or may not occur depending on the result of a pending litigation process (lawsuit). The traditional methodology for estimating this likelihood is based on the opinion from the lawyer's experience which is based on a qualitative appreciation. This dissertation presents a mathematical modeling framework based on a Deep Learning architecture that estimates the probability outcome of a litigation process (accepted & not accepted) with a particular use on Contingent Liabilities. The framework offers a degree of confidence by describing how likely an event will occur in terms of probability and provides results in seconds. Besides the primary outcome, it offers a sample of the most similar cases to the estimated lawsuit that serve as support to perform litigation strategies. We tested our framework in two litigation process databases from: (1) the European Court of Human Rights (ECHR) and (2) the Brazilian 4th regional labor court. Our framework achieved to our knowledge the best-published performance (precision = 0.906) on the ECHR database, a widely used collection of litigation processes, and it is the first to be applied in a Brazilian labor court. Results show that the framework is a suitable alternative to be used against the traditional method of estimating the verdict outcome from a pending litigation performed by lawyers. Finally, we validated our results with experts who confirmed the promising possibilities of the framework. We encourage academics to continue developing research on mathematical modeling in the legal area as it is an emerging topic with a promising future and practitioners to use tools based as the proposed, as they provides substantial advantages in terms of accuracy and speed over conventional methods.

Keywords: Deep Learning, NLP, Legal Analytics

RESUMO

Estimar o resultado de um processo em litígio é crucial para muitas organizações. Uma aplicação específica são os "Passivos Contingenciais", que se referem a passivos que podem ou não ocorrer dependendo do resultado de um processo judicial em litígio. A metodologia tradicional para estimar essa probabilidade baseia-se na opinião de um advogado quem determina a possibilidade de um processo judicial ser perdido a partir de uma avaliação quantitativa. Esta tese apresenta a um modelo matemático baseado numa arquitetura de *Deep Learning* cujo objetivo é estimar a probabilidade de ganho ou perda de um processo de litígio, principalmente para ser utilizada na estimação de Passivos Contingenciais. A arquitetura, diferentemente do método tradicional, oferece um maior grau de confiança ao prever o resultado de um processo legal em termos de probabilidade e com um tempo de processamento de segundos. Além do resultado primário, a arquitetura estima uma amostra dos casos mais semelhantes ao processo estimado, que servem de apoio para a realização de estratégias de litígio. Nossa arquitetura foi testada em duas bases de dados de processos legais: (1) o Tribunal Europeu de Direitos Humanos (ECHR) e (2) o 4º Tribunal Regional do Trabalho brasileiro (4TRT). Ela estimou de acordo com nosso conhecimento, o melhor desempenho já publicado (precisão = 0,906) na base de dados da ECHR, uma coleção amplamente utilizada de processos legais, e é o primeiro trabalho a aplicar essa metodologia em um tribunal de trabalho brasileiro. Os resultados mostram que a arquitetura é uma alternativa adequada a ser utilizada contra o método tradicional de estimação do desfecho de um processo em litígio realizado por advogados. Finalmente, validamos nossos resultados com especialistas que confirmaram as possibilidades promissoras da arquitetura. Assim, nos incentivamos os académicos a continuar desenvolvendo pesquisas sobre modelagem matemática na área jurídica, pois é um tema emergente com um futuro promissor e aos usuários a utilizar ferramentas baseadas como a desenvolvida em nosso trabalho, pois fornecem vantagens substanciais em termos de precisão e velocidade sobre os métodos convencionais.

Palavras-chave: Deep Learning, NLP, Direito, Analytics

Table of Contents

List of Figures

List of tables

1. INTRODUCTION

1.1 Motivation

Since the seminal work of image classification (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), the world experimented with a new wave of believing that machines could replicate complex tasks performed by humans – Artificial Intelligence (AI). One innate behavior is the Natural Language that humans learn unconsciously. But for a machine, it is a complex task. This feature encourages us to explore a problem that involves Natural Language in organizations: litigation processes.

We explored a fundamental problem that challenges organization: how to estimate the probability of winning or losing a litigation process with a quantitative methodology. Estimating this probability is a fundamental task as it involves forecasting resources that can be earned or loose (*e.g.*, Tax law & labor demands). It is a critical part of risk management. In financial terms, a resource that can be loose in a legal dispute is defined as a "Contingent Liability," which refers to an uncertain obligation. The traditional methodology to deal with this problem of Contingent Liabilities depends on Accounting Standard rules which stays that the possibilities of loose of resources that depend on litigation outcomes must be quantified according to the opinion from lawyers into the categories: high, low, or remote chance of losing (FASB, 2010). This classification is a qualitative validation as it depends on a particular appreciation from lawyers according to his experience. Many financial statement users have complained that the estimation of the likelihood by this methodology provides qualitative clues about the probability of loss, but are limited in quantitative detail (HENNES, 2014; HOFFMAN; PATTON, 1997).

As an attempt to solve this problem, we propose an AI framework, where its input is a legal claim (petition) and its output a probability of loss. As an example, an organization has a legal dispute that involves an amount of resources to be paid to the government. Translated into our framework the input will be the legal claim in its original form and the output the probability of the organization loose the dispute.

A primary aspect of a litigation process is that it is stored as a text document (lawsuit). In its basic form, a lawsuit contains a petition and a verdict. Thus, the legal outcome estimation is a Natural Language Problem (NLP) that aims to classify a text document into two categories (win or loose) according to a probability. Our proposed framework is composed of three main blocks: the first pre-process and transforms the petition text into a structure array. The second transforms the text into a tensor representation and estimates the probability of loose. The third provides a ranking of the more similar litigation cases to the one estimated. The framework is based on a Deep Learning (DL) architecture that has provided promissory results for Natural Language problems (COLLOBERT *et al.*, 2011; LECUN; BENGIO; HINTON, 2015; MIYATO; DAI; GOODFELLOW, 2016). The results from this study will contribute both to academics and practitioners. For academics, it will provide new insights of DL architectures applied to modeling legal texts (CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019). For practitioners, it will provide a tool to manage risk in the context of "Contingent Liabilities" (e.g., lawyers, accountants, clients) (FISHER; GARNSEY; HUGHES, 2016).

Text modeling has been successfully applied to problems such as mail spam detection (WU et al., 2017), sentiment analyses (YENTER; VERMA, 2017), and social media hate speech detection (MALMASI; ZAMPIERI, 2017). However, this technique has been little explored in the problem of estimating litigation process resolutions. Some pioneering works have used the technique to trial documents of countries from the US (KATZ et al., 2014) and the EU (CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019). However, there is a lack of research for the Brazilian context. Therefore, this work aims to answer the following research question: what is the probability of winning or losing a labor court litigation process using a AI framework for CL's management?

1.2 Objectives

To answer the research question, we divide our study into four specific objectives:

- To perform a review of modeling techniques used in litigation documents from literature and users.

- To develop an AI framework that can predict the probability of winning or losing a litigation process.

- To test the proposed framework with international and local (Brazilian) litigation databases.

- To validate our results with experts.

1.2 Structure of the Dissertation

This document is divided into 8 chapters described as following:

In chapter 2,

- We explore the problem of Contingent Liabilities estimation, providing a description of the actual process of estimation and its limitations.
- We performed a set of 3 in-depth interviews from members of representative organizations involved in the estimation of Contingent Liabilities, who described how the process is estimated in their organizations, limitations, and the importance of having a framework like the one we constructed.

In chapter 3,

- We reported a literature review of DL uses in the legal area, the methodology basis of our framework. The review includes categories of use, journals of publication, collaboration networks, and future trends.

In chapter 4,

- We describe the problem of Legal Judgment Prediction, which is the reference to model a litigation process, by providing its mathematical formulation, and previously reported works.

In chapter 5,

- We outline the structure of our framework, which includes three main blocks: (1) Transformation and structuring of PDF files into a suitable input form for the model. (2) Representation of texts into a tensor structure and estimation of a probability. (3) Calculation of similar litigation cases to the one provided as input from all database.

In Chapter 6,

- We report the results of the experiments from our framework performed into two databases, the ECHR collection and a labor litigation process from a Brazilian regional labor court (Tribunal Regional do Trabalho 4 região – TRT4).

In Chapter 7 and 8,

- We end the discussion of the results, conclusions, and suggestions for future research.

Finally, as a result of this work: (1) two congress papers were published, the first in the 2020 IEEE International Conference of Big Data (MONTELONGO; BECKER, 2020b) and the second in the 2020 International Conference on Data Mining (ICDM) (MONTELONGO; BECKER, 2020a), (2) the project was awarded by the Nvidia company with a Graphic Process Unit (GPU) Titan XP to perform the experiments of our proposed framework, and (3) we managed a contract for future opportunities of research between the labor court (TRT4), and the Federal University of Rio Grande do Sul (UFRGS), where TRT4 will provide lawsuits processes in a complete form from its data centers.

2. CONTINGENT LIABILITIES

Contingent Liabilities (CLs) refer to obligations whose timing and magnitude depend on some uncertain event outside the control of an organization, such as a pending lawsuit. Previous research on CLs is divided between works performed on the public and the private sector. Some examples in the public sector include discussions on CLs' approaches to deal with government fiscal risks (BRIXI; SCHICK, 2002), and policy implications of CLs not being reported on the balance sheet (off-balance sheet) (BLEJER; SCHUMACHER, 2000). In the private sector, an example is the relationship between the companies turnover and the number of CLs (lawsuit demands) (AHARONY; LIU; YAWSON, 2015).

The most conventional form of accounting, a liability, is in its discrete form (a cost that has been or not used with 100% o certainty - discrete). However, there exist situations in which potentially costs depend on uncertain events, not over an organization's control, such as a pending lawsuit. Hence it represents a risk for organizations. To manage this uncertainty the actual mechanisms of corporate governance relies on providing transparency about the possibility of this event occurring. The main mechanism of control from the US is the Financial Accounting Standard Board (FASB) that delineates the regulations that US companies must adhere to when reporting their financial position and preparing financial statements. We cited the example from the US as most of the international literature is based on these standards and accounting rules from other countries are transiting into an international convergence (CONSONI; COLAUTO, 2016). Thus, the corporate governance mechanisms of CLs between different countries behave similarly. In Brazil, CLs' mechanisms of control (disclosures) are regulated by the CPC25 (Comitê de Pronunciamentos Contábeis) (CPC, 2005), and on Europe by the IAS37 (International Accounting Standard) (COMMITTEE, 1998). FASB and similarly the CPC25 dictates that CLs must be categorized according to its likelihood of a loss into: probable, reasonably possible, and remote. The FASB defines probable as when the future event is likely to occur, Reasonably possibly as when the chance of the future event is less than probable and remote as when there is a slight chance of the future event occurring. Accounting standards also define the conditions under which accountants must accrue CLs (report on a balance sheet). Contingencies that are probable must be compulsory accrued on the balance sheet, probable only need to provide a disclosure note and remote does not need to be reported (KUNZ, 2015). In practical terms, when a litigation process that involves resources against an organization is in dispute, the possibilities of the process to be loose are estimated by a lawyer who provides the information to the accountant who registers in the company's balance sheet (accrued) the liability into one of the three categories (probable, reasonably possible and remote). **Table 1** summarizes information dictated by the FASB.

Table 1. Contingent Liabilities decision matrix. On the upper side, the range of possibilities in which a litigation class can be classified according to its possibility of loss (Probable, Reasonably Possible and Remote). On the left side, the confidence of estimation (known, yes, no). Depending on the combination from this matrix, the FASB dictates if the CL must be accrued.

		Likelihood of occurrence		
		Probable	**Reasonably Possible**	**Remote**
Is Contingent Liability reasonably **estimated?**	**Known**	Liability accrued and disclosure note	Disclosure note only	No disclosure required*
	Yes	Liability accrued and disclosure note	Disclosure note only	No disclosure required*
	No	Disclosure note only	Disclosure note only	No disclosure required*

*Except for certain guarantees and other specified off-balance sheet risk situations. Adapted from Kunz (2015).

Although accounting standards have been applied as the primary form of regulation, the literature identifies two significant problems in their use. The first is the variation in interpreting probability meaning, between lawyers, as they have to assign a degree of it into a category (probable, reasonably possible and remote) according to their experience (AMER; HACKENBRACK; NELSON, 1994). Second, the existing disputes between lawyers and accountants (auditor’s) positions on liabilities as they work with different foundations. Lawyers' work is based on the American Bar Association (ABA) statement of Policy no 12 and auditors on the FASB. Particularly, ABA suggests that lawyers abstain from expressing judgment on the outcome of a claim when the prospect of failure is doubtful or highly doubtful. ABA Statement of Policy provides the following basis:

In view of the inherent uncertainties, the lawyer should normally refrain from expressing judgments as to the outcome except in those relatively few clear cases where it appears to the lawyer that an unfavorable outcome is either” probable” or” remote” (Association, 1976) ;

With this prerogative, when auditors need information from lawyers, they confront an obstacle. Usually, auditors receive just a note from lawyers stating their inability to express an opinion. **Figure 1** exhibits a chart representing the information flow process to estimate the CL likelihood of loss. As it shows, a CL in a lawsuit must be first disclosed by a lawyer on an ABA basis. Later on, auditors estimate the likelihood of loss using lawyer's disclosures using FSBA basis. A conflict might exist when auditors need information from lawyers.

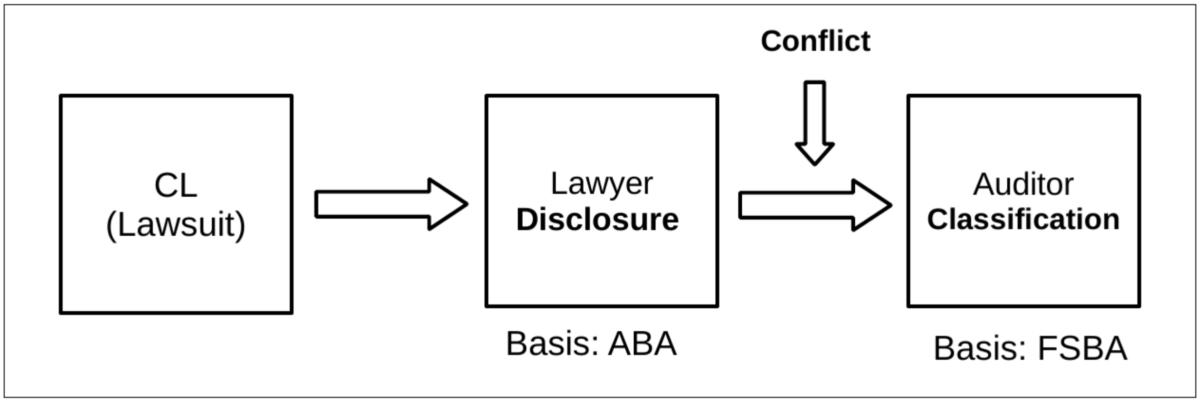


Figure 1. Information flow process for Contingent Liabilities estimation. On the left, a resource of an organization that is on a legal dispute (CL) is disclosed (estimated with a probability of loss) by a lawyer that uses ABA as a basis and transmits the information to an auditor (accountant) to accrue the information into a balance sheet. Both the lawyer and the auditor can have a conflict of interpretation because the first use ABA as a basis and the second the FSBA basis.

2.2 Contingent liabilities users

After performing our literature research, we also feel the need of understand from primary sources the process of CLs management and its importance for organizations. We interviewed the directors from a set of companies who directly manage the pending lawsuit processes. To perform the interviews we selected a sample of companies listed on B3 from different sectors and get in contact to ask the possibility to contribute to our study by providing an interview. We selected the enterprises from the Brazilian Stock Exchange (B3) as their financial information is publicly available and they manage a substantial amount of litigation processes. Three organizations from the construction, financial and media sector accepted to contribute with the interview. Our interviews were not structured (without a protocol) as our intention was to have a first understanding of the CLs management process.

2.2.1 Construction company

We began by visiting a construction company that is one of the largest companies in southern Brazil. The group is 50 years old and employs over 2,000 workers. We had an interview with its legal director, who has been working in the company for over 26 years. The interviewer has a law degree and specialization in corporate and environmental affairs. We talked about many topics during the interview that could be summarized into four broad categories: the Brazilian law system, the law department structure of the company, the process of lawsuit handling, and the risk management of pending lawsuits.

The law director described the Brazilian law system. He said that during the last years, it had been through a process of transformation, going from using traditional paper sheets on processes to an electronic system. He visualized that in a short time, all court systems would be digital, that this transformation would enable users to use technologies like the one we were proposing. He mentioned that the Federal Court (“Tribunal Regional Federal - TRF”) has the best electronic system structure. He cited as an example of a type of case disputed in the Federal Court, federal taxes. He suggested us to review the platform of the Federal Court because it offers the best electronic facilities to provide information. To conclude this topic, he added that the second-best electronic court, in his opinion, is the STJ (“Superior Tribunal de Justiça”). However, that in the future, all of them would be standardized, he said.

At this moment, the construction company has a volume of about 2,400 ongoing lawsuit processes. The oldest was from 1980, and the newest was from 2018. The director said that the company used to have a larger volume, about 6,000 processes, because there existed more legal instability in previous years. The company has an internal legal team but also worked with external law offices. At the time of the interview, the company works with six external law offices. The in-house lawyer’s department disputes lawsuits that the company considers easy resolution. External offices take care of more complicated cases. As an example of an easy resolution case, he mentioned a tax dispute, particularly the Property and Urban Land Tax (IPTU - “Imposto Predial e Territorial Urbano”). As an example of a regular complexity case, he mentioned returning a house to a company. And a complex case related to environmental affairs.

He continued explaining that sometimes it is better to lose a legal case in the first instance than spend money and going into further instances. The decision of losing or going into further instances is mainly based on two aspects: the money disputed and the possibility that the case turns into a precedent. Losing a case could be risky because it can be used in subsequent cases with similar issues as a precedent.

He explained about pending lawsuits management. He said that public listed companies from B3 have to report pending lawsuits and that The General Accounting Principles establish the criteria to register expenditures that depend on a pending litigation process. He described the management of ongoing lawsuits in accounting terms and how the company deals with them. He said that the accounting principles establish that pending lawsuits must be classified into remote, possible, and likely, according to the possibility of loss. The management of pending lawsuits inside the construction company is performed by reports that should be carried out every three months by each of the lawyers that is in charge of the litigation process.

He explained how the process to estimate the possibility of losing the pending lawsuits inside the company is organized. He said that every lawyer is responsible for a specific lawsuit case. They estimated in which of the three status a process belongs to: remote, possible, and likely. He added that cases could change status as long as they are not static. That is, a case that company estimated that is won for sure in the first instance but advanced to a second instance with solid pieces of evidence from the author will change its status from remote to possible. He reinforced that the law has many particularities, which could be a difference between losing and winning a process. "When a lawsuit goes to a second instance, the possibility of reverting a case in court is minimum", he added.

He described some particularities of estimating a lawsuit status in more detail. That when the possibility of losing a case is remote, it was rarely reported. When it is possible, it must be reported, and when it is probable, it must be compulsory reported. In the three cases, the money disputed must also be updated. He cited that the public document where lawsuits can be consulted is the reference form from B3.

He said that law processes are not determinant for a company to go bankrupt. However, they are an excellent indicator of a company's health performance. They are good future predictors, a signal of how well organized a company is. For example, he said that cases involving extra hours of work as a type of case that signal that a business is not well organized. He added there are classes of cases that a business will always have according to its core, eg. a financial institution will have cases related to financial debts. We also discussed if there existed a particular type of cases critical for a company. He said that all depends on the company. For the ones with many employees, like a factory, cases related to work could be critical. While for a construction company, environmental processes would be critical.

We discussed if some outcome cases were easier to predict than others. He cited consumption relations as easier because there was a slight bias in the Brazilian law to protect the consumer. Cases difficult to predict are the ones related to the labor law. To conclude, he said that estimating a pending lawsuit always involves the subjective appreciation of a lawyer.

2.2.2 Financial company

Our second interview confirmed the importance of methodological analyses in the estimation of contingent liabilities. We interviewed the lawyer director of one of the biggest financial companies in the South of Brazil. The company's market share is about 5% and 6%, but it reaches between 10% and 15% in the south region.

After we described the objective of our research, the introductory phrase of the director was, "the study that you are performing makes perfect sense. The lawyer needs to face the impacts that information technology is having on the law. We are in the process of renewing our management system of lawsuits. We would like to use the software e-law, which provides more accurate results in searching on legal databases. The tool will help us to be more assertive in strategies to construct legal petitions. It works with Watson's IBM. Our institution is a financial cooperative audited by Brazilian's Central bank. We need to perform disclosures about the situation of juridical cases as the law establishes", he said

During the last years, the enterprise had been developing analyzing methodological procedures to analyses contingents liabilities. It implemented a process based on technical analysis from an internal law group supported by an external law firm that reports the information for the accounting area. The director recognized that the strategy is simple but efficient. It was implemented two years ago and he estimated that the strategy has saved about 5 million *Reais* in two years. The lawyer director added that the technical challenge is how to be the most assertive possible in predicting the result of a lawsuit. He said, "A system that compares lawsuit to lawsuit and with jurisprudence will help a lot."

The methodology of estimation through technical analyses at the moment is only used for labor cases. But the financial company aims to implement it in other types of cases. He said, "all labor complaints are saved and make available for been consulted at any time by our lawyer team. When a new case is disputed, a technical analyses is done based on similar cases from historical legal petitions. For example, our enterprise is classified as a cooperative. A recurrent claim is that workers demand to be recognized as bank employees, but the precedent 76 of TRT4 (Regional Labor Court 4th region) does not recognize workers as bank employees". Therefore, when a new petition of this kind is performed against the company, it is identified and the strategy is clearly defined with high chances of success. Using value estimation and probability of loss has been the basic strategy of the enterprise during the last years. The interviewer said, “the strategy is based on estimating a cost for a case. How much does it cost?” If the probability of loss is high, the company tries to negotiate extra-judicially or with the author.

The director added that the enterprise has a severe concern for employers that work for the company, that when a case involves an employee that promotes a legal petition against the company and loses the case, the company tries to end the case by hurting the other party as little as possible. He cited as an example the moral damage cause that has a high probability of being lost by the petitioner “our company is successful in winning this type of cases, however, when we win a lawsuit, we tried to agree with the other party by having the best possible between the parties”. During the last five years, the director added that the enterprise had been qualified among the best companies to work, that the number of labor lawsuits at that moment was about 1,000, approximately 50% less than previous years.

We talked about other types of cases, the tax cases. He added, “there is a small amount of this type of cases. They are not relevant for our business.” The director also said that a successful procedure for the company for the legal petitions management was to detect the root causes of a demand and to decrease the probability of happening again, in other words identifying the fault and acting in a predictive way. For example, the incorrect inclusion of a customer into the list of debt people the methodology of the company is:

- Root: the case was not deleted from the list of debt people at the correct time.
- Action: look up in jurisprudence.
- Police: delete in time this type of cases of this type

We asked if there exists a particular kind of recurrent demand. He responded that services from outsourcing call centers, that often this kind of enterprises disappear, and the company was affected by labor demands that come from workers that use to belong to the outsourcing company.

The interviewer described the law department of the enterprise. According to him, the number of lawsuits at the moment is around: 4,000 civil, 1,000 labor, 20 tax, and no environmental. For the company, the environmental cases were relevant but small in quantity. The internal lawyer team of the company is integrated by 23 people, 13 of them lawyers. Cases are managed using two models:

- Local: each financial agency hires its own legal office considering the central office models. The company has around 300 legal offices of this type.
- Systematically: cases are managed by legal offices that are controlled by the headquarters offices. This type of management is in charge of around 50% of the legal cases.

One important point that the company is concerned about is the management of the external offices. The company intends to use technological resources to have more information available to make more accurate decisions, however, they are concerned about how to provide this information to the external offices. Regarding the possibilities of patterns among the legal system, the interviewer said that in his opinion it could exist partiality in some legal verdicts depending on the judge that performs a decision. For example, some judges make decisions in favor of employees. He concluded the conversation by highlighting that the great advantage of the company against the competitors is that the clients are also the owners of the company.

2.2.3 Media company

Our third interview was from one of the most prominent media communication groups from Brazil. The company employs approximately 6,000 people and participates in television, radio, and newspaper segments. Its revenues over last year were over a million of Reais. Unlike the other interviews, this one was with a multidisciplinary team of directories from TI, human resources, and law, as they were interested in our research.

We made an introduction to our research purposes. We explained the possibilities of estimating pending lawsuits with mathematical approaches instead of classifying them with the traditional methods performed by lawyers. The IT director responded, “We agree that people who have more historical information about lawsuits will have more chances to win a lawsuit process.” We added our vision that technology will help to make fairer judgments, using less human appreciation from the judges. We said that in our perspective, an ideal system of justice would consist of an algorithm that automatically received facts from petitioners and accusers and that it estimated a verdict, that it would provide fairer decisions.

The interviewed lawyer exposed that he worked in jurimetrics, and explained that it consists of a science that aims to map behaviors from the judiciary, that it exists patterns of behavior among historically judged lawsuits. That he performed research on data from labor courts and found that some judges have specific behaviors, for example, with a tendency to make decisions in favor of workers or in organizations. He added that the selection of variables to perform his research was made manually, that some difficulties arose in structuring the text, and that working with labor court cases is a challenge as they involve lots of variables.

The company has about 1300 ongoing lawsuit processes. From this quantity, 950 were labor lawsuits. We discussed that this phenomenon is due to the company’s nature which depends on service from people. He added that the company has a turnover of between 80 and 90 people per month. In the same line as refereed in the interview from the financial sector, the strategy that the company follows consists in identifying the root causes of lawsuit processes in order to avoid a systematic repetition.

The legal department of the company has 12 lawyers. Five worked with civil law, five with labor law, and 2 with Contingent Liabilities. The function of the law department is to distribute the lawsuits to external legal firms as they do not perform in-house process strategies. The lawyer ended, “ we are here to clean the content of a lawsuit data so we can decide which external attorney is the most appropriate depending on the type of case”.

3. DEEP LEARNING IN THE LAW CONTEXT

Lawsuits are formulated by humans to humans, thus in Natural Language. This assumption implies that interpreting the concept of a lawsuit by a machine involves a deep understanding of Natural Language structures. But this is not an easy task. Language is a complex cognitive, adaptive communication system with complex particularities (LARSEN-FREEMAN; CAMERON, 2008) that include: its construction consists of multiple agents, is adaptive; it suffers from past, and present actions that interact to form future constructions, its structures of language emerge from interrelated patterns of experience, social interaction, and cognitive mechanisms; the meaning of a text relies on a text as an overall, not from single words, it evolves, is dynamic (HAUSER; CHOMSKY; FITCH, 2002).

In this context, the field of Natural Language Processing (NLP) aims to convert these complex behaviors into formal representations accessible for computers to manipulate (NIRENBURG; MCSHANE, 2016). Neural Networks (NN's) and the subfield of Deep Learning (DL) have become the state-of-the-art methodology for NLP (SUTSKEVER; VINYALS; LE, 2014). Although NN's have recently gained attention, this technique has been utilized in texts from different fields including the legal domain since the late '80s (BELEW, 1987). However, the scope of these first approaches was limited due to the lack of large data sets and computational resources. Much of the work was just demonstrative (BENCH-CAPON, 1993) with small data sets (MERKL; SCHWEIGHOFFER; WINIWARTER, 1999). New improvements in hardware capacity and data availability have enabled the design of complex structures of NN's with multiple hidden layers. This is the so-called DL that has enabled the advancement in language modeling (LECUN; BENGIO; HINTON, 2015; SCHMIDHUBER, 2015). Due that our framework is based on a DL architecture we believed it was important to identified studies in legal texts (legal domain) that used DL as primary methodology. Therefore we performed a systematic bibliographic review that focused on three key topics:

- The problems (tasks) that have been solved using DL.
- The corpus (texts) that have been used to train the models.
- The future directions of DL in the legal domain.

3.1 Artificial Neural Networks and Deep Learning

In its basic form, a NN is a collection of connected units (nodes) that can transmit a signal from one node to another and that allows to solve AI problems such as classification and regression. Nodes are disposed of as layers. The first one is the input of raw data, and the last one produces the result (classification or regression). Layers between the input and output are known as hidden layers, connections between neurons are known as edges and have a weight that adjusts while the learning process takes place. Commonly, the signal from a node is restricted and transmitted if it crosses a threshold composed of the sum of a non-linear function (BASHEER AND HAJMEER 2000).

DL is a type of NN structure composed of multiple hidden layers named Deep Architecture (BENGIO, 2009; LECUN; BENGIO; HINTON, 2015) that can be complemented with other techniques, such as Convolutional Neural Networks (CNN) (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) and Long Short-Term Memory (LSTM) (HOCHREITER; SCHMIDHUBER, 1997). This methodology enables the transformation of raw data into higher abstract features by learning complex non-linear functions. Over the last years, DL has become the state-of-the-art methodology of NLP (ABOOD; FELTENBERGER, 2018; CHALKIDIS; KAMPAS, 2018; KOWSRIHAWAT; VATEEKUL; BOONKWAN, 2018; SADEGHIAN et al., 2018).

Figure 2 illustrates the basic structure of a DL architecture composed of 4 layers. Where *x* represents the *i* input, *w* the weight value from the *i* input in the *j* layer and *y* the activation *k.* The first layer (bottom) represents the input, and the last layer (top) represents the output. It can be observed that between the input and the output, there are two hidden layers represented as H1 and H2. The input exemplifies raw data that is transmitted into the hidden layers, and finally a classification output is estimated. In this example, the input signal is represented as "*i*". The second signal on layer H1 is represented by the value j, the third signal on layer H2 is represented by the value k, and the output signal is represented by the value *I*. Each input of the node is computed by functions illustrated on the right side of **figure 2**.

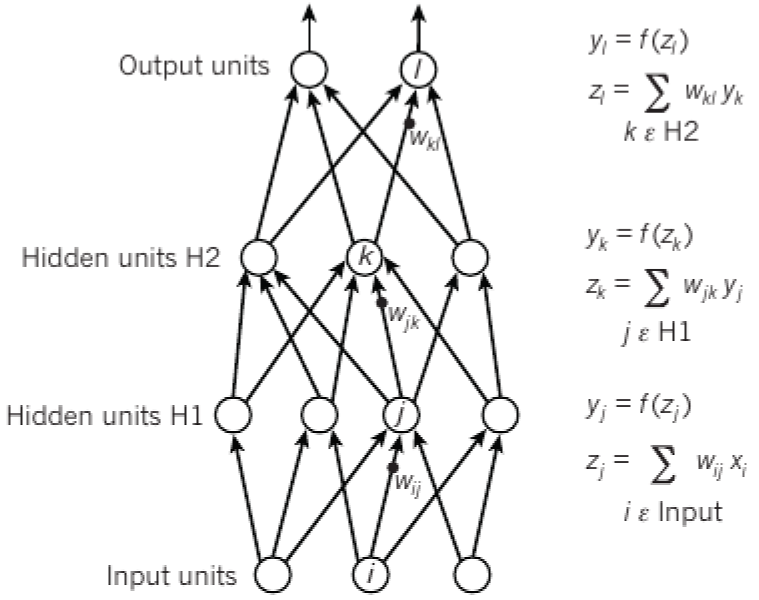

Output units
l
w_{kl}
Hidden units H2
k
w_{jk}
Hidden units H1
j
w_{ij}
Input units
i
$y_l = f(z_l)$
$z_l = \sum_{k \varepsilon H2} w_{kl} y_k$
$y_k = f(z_k)$
$z_k = \sum_{j \varepsilon H1} w_{jk} y_j$
$y_j = f(z_j)$
$z_j = \sum_{i \varepsilon \text{Input}} w_{ij} x_i$

Figure 2. Representation of a DL architecture. A DL architecture composed of 4 layers that perform a classification task. At the bottom, the signal is introduced and then processed until it reaches the top section, where a classification is performed. On the right, the activation functions of each layer are formulated (LECUN; BENGIO; HINTON, 2015).

3.2 Legal Documents

A particular feature of most current litigation systems is that records are stored as electronic text documents. Over the last years, the quantity of legal information in digital formats has been exponentially increased. Thanks to the availability of this source of information, the quality of DL models has become more reliable (it should be highlighted that the quality of the output of the DL model dramatically depends on the quantity of information provided as input) (NAJAFABADI et al., 2015). Legal documents are provided from two sources, either processed for research use (VOGEL; HAMANN; GAUER, 2018) or provided for public access. For example, courts in the United States provide public information of legal petitions on its website https://www.pacer.gov/, whereas other public legal courts as the Brazilian provide public information as a summary, not the complete legal petition documents. There also exist other independent organizations, such as the Free Law Project, that offers a wide range of resources on their website free law.

3.2.1 Systematic review of the literature

To perform our research of legal documents that use NN or DL as primary methodology, we retrieved a set of articles from the most extensive databases, including IEEE Xplore, Science Direct, Emerald, Springer, Web of Science, and Google Scholar. We utilized the terms: “Neural Networks” and” Deep Learning” in combination with “legal” or “law.” Nevertheless, the word “law” appeared in multiple ambiguous contexts, such as law of motion. Therefore, only the word” legal” was utilized. Later, we examined specialized journals of law: Law and AI, Stanford law review, Yale law review, Columbia law review, Computer law and security review, Law probability and risk, and Harvard law review. Regardless of the many databases analyzed, we noticed that some relevant articles were missing during this process. Therefore, we included an additional search within the top journals of law (herein, a top journal should be in the top 10 of the Scimago Journal & Country Rank and Journal Citation Report) and added referenced works we do not find by search mechanisms during the analysis and belonged to the category. From the results of the databases, we identified a final sample of 137 works that satisfied our criteria. We classified each article according to the objective (we defined nine categories through criteria expanded in the following subsection). Finally, we retrieved the datasets utilized to train the models and organized the information in a comprehensible structure.

3.2.2 Categories of the selected works

Our primary interest was the understanding of the research objectives of the selected articles. Therefore, we created a taxonomy to classify each article into 1 of 9 categories based on the objectives of each work. The categories and criteria utilized to create the taxonomy were:

- Classification: Works that aimed to discriminate an object into one of several known categories (e.g., patent classification) (LI et al., 2018);
- Feature extraction: works that tackled the problem of reducing the number of resources required to describe a large data set (e.g., derive the profile of the attackers) (ADDERLEY; MUSGROVE, 2001);
- Information extraction: works that identified named entities such as places, persons,

organizations, and works that extract other complex information such as events and narratives (e.g., recognize parts in legal texts) (NGUYEN et al., 2018);

- Information retrieval: works that retrieved articles of interest out of a collection of legal documents that entail a query (e.g., automated identification of directives) (NANDA et al., 2018);
- Pre-processing: works that prepared data before processing, including outliers detection and network pre-structuring (e.g., pre-processing texts) (VIJAYARANI; ILAMATHI; NITHYA, 2015);
- Summarization: works that condensed new versions of the original documents (e.g., automatic summarization) (YOUSEFI-AZAR; HAMEY, 2017);
- Text generation: models that aimed to produce human languages from some underlying non-linguistic representation (e.g., automatic production of legal texts) (JOHN et al., 2017); and
- Theoretical: works that lacked of a direct implementation such as discussions, exemplifications, and reviews (Discussion of logic-based and data-centric approaches) (BRANTING, 2017).

From these created categories, we estimated the frequency of works. As **fig3a** shows, we found two broad groups defined as “high” and “low” in the number of works. The high number is represented by classification (0.39), theory (0.28), information extraction (0.15), and information retrieval (0.12). The low number is represented by text generation (0.02), preprocessing (0.01), feature extraction (0.01), translation (0.01) and summarization (0.01). The results show that classification and theoretical categories dominate the work with 39% of the total sample. We interpreted that classification has the highest value because DL is mainly used for solving classification problems. The theoretical category that we defined as works lacking an explicit implementation, such as discussions, exemplifications, and reviews, represents a significant proportion of 28%. The majority of theoretical works were published between the years 1987 to 2002, when the NNs were first proposed as a methodology to solve problems in the legal area. Information extraction and information retrieval stay in third and fourth positions, both with 12%. In this group, we included the articles from the Competition on Legal Information Extraction/Entailment COLIEE, which is the only competition of AI devoted to the legal domain that we identified. Among the low-numbered areas, we distinguished text generation with 2% and preprocessing, feature extraction, translation, and summarization with 1%. We identified text generation and summarization as future research opportunities. For text generation because new architectures (BROWN et al., 2020) have improved the accuracy in natural language generation, with applications such as interactive conversations (chatbots). For summarization as it represents an essential tool for legal professional users as long as they need to consult large quantities of information. Notice that pre-processing in our interpretation is not an area of opportunity because new architectures such as BERT (DEVLIN et al., 201can) can handle inputs from raw text.

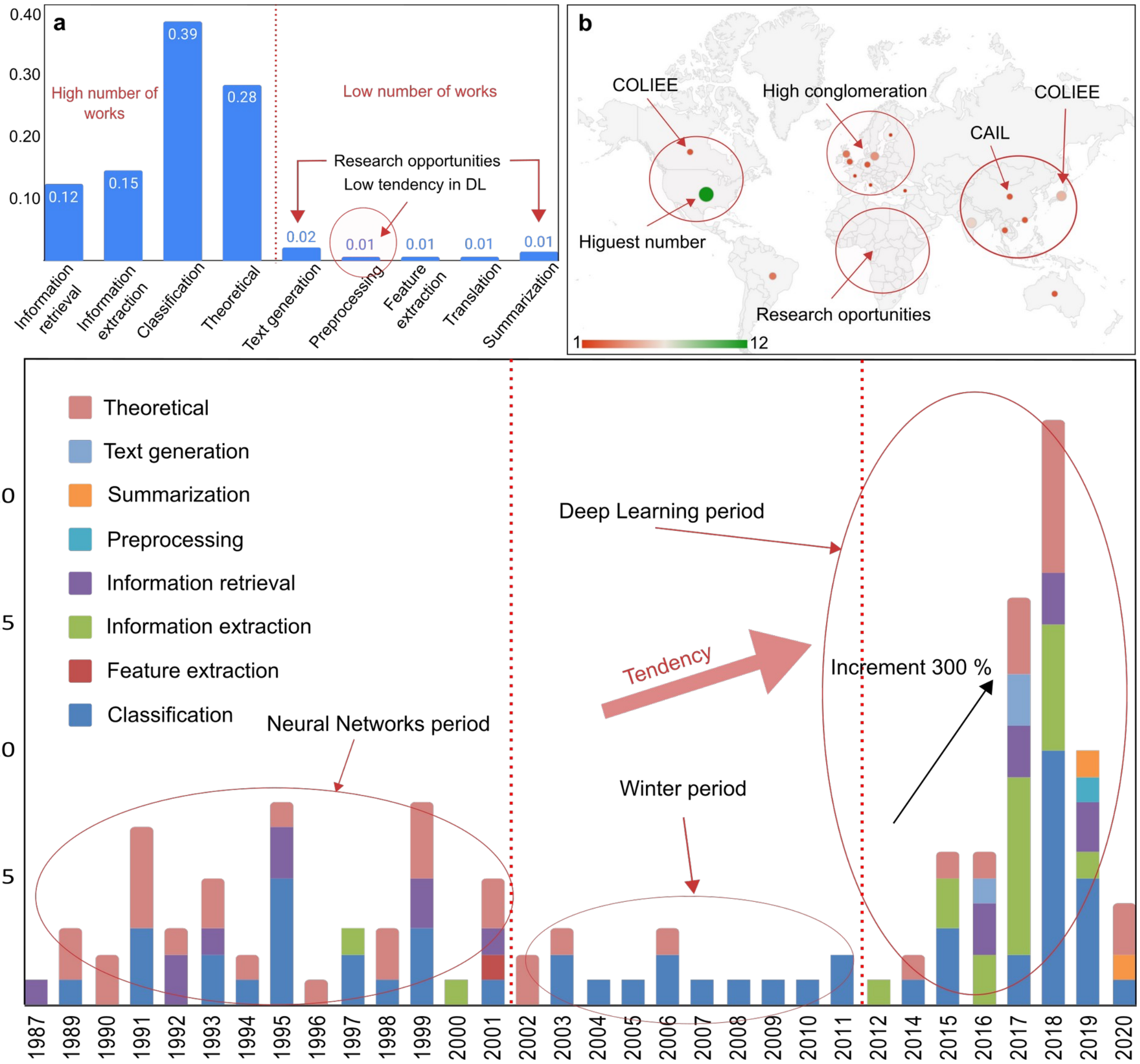


Figure 3. Tasks, location and a longitudinal representation of DL works from our sample. (a) Frequency of works according to the task performed. We divided the groups into a high and low number of published studies. Classification is the group with the highest number of reported works, while text generation and summarization are with low number and research opportunities. **(b)** Corpus used to train the models according to its country. Europe exhibits a high conglomeration, COLIEE and CAIL are the only competitions focused on problems using legal documents. **(c)** We divided the published works into three broad periods according to a visual inspection. The NN's period belongs to the first era of DL when theoretical studies that gained attention. After that, a second era started with a winter period, with only some classification works. Finally, the age of DL in which a resurgence of the method began and the number of works increased exponentially by 300% - between 2015 and 2018. In addition, diversity has increased, in particular with information extraction work. Interestingly, the number of articles decreased over 2019. Further research is needed to understand the nature underlying this phenomenon.

Fig 3b illustrates the geolocation areas of the data sets utilized to train the models. The most significant number of databases within a country is the USA (23.53%). Europe (21.57%) represents an area with a substantial number of works by country, and it is the only one that provides datasets that belongs to an entire region (Europe Union) (CHALKIDIS; KAMPAS, 2018). The corpus of the COLLIE (Japan & Canada) (TRAN et al., 2020) and CAIL (China) (CHEN et al., 2019) are the only datasets that we found devoted for competition purposes. The last region was Africa, with no dataset found in our search. **Fig 3c** Depicts a longitudinal representation of the DL works and the performed tasks. The first article we found was in 1987 and the last one in 2020. As it can be seen, the works performed with a DL have exponentially increased over the last years, which evidenced the increased interest in the methodology.

Table 2. Categories and corpus of selected articles.

Category	Objectives of selected works	Corpus
Classification	Case classification (DA SILVA et al., 2018; NGUYEN et al., 2018)	45532 Brazilian appeals;
	Legal court classification (UNDAVIA; MEYERS; ORTEGA, 2018)	8419 USA Supreme Court opinions;
	Contract resolution (CHAPHALKAR; SANDBHOR, 2015)	419 Indian contracts;
	Court decision predictions (BOCHEREAU; BOURCIER; BOURGINE, 1991)	1000 judgments of Thailand Supreme Court;
	Geospatial criminal activity prediction (CORCORAN et al., 2001)	Collection
	Patent classification (ABOOD; Case FELTENBERGER, 2018)	2679443 utility patents.
Feature extraction	Profile of sexual attackers (ADDERLEY; MUSGROVE, 2001)	2370 recorded sexual offenses from the UK;
	Recognizing parts of legal texts (NGUYEN et al., 2018)	130000 citations from the US code;
Information extraction	Identify information	Collection;

	(THAMMABOOSADEE; WATANAPA; CHAROENKITKARN, 2012)	
	Contract elements extraction (CHALKIDIS; ANDROUTSOPOULOS, 2017)	3500 English contracts;
	Exploratory analysis of concepts (MERKL; SCHWEIGHOFFER; WINIWARTER, 1999)	75 court decisions from the European Community;
Exploratory analysis of concepts	Identify national implementations (NANDA et al., 2018)	43 European directives;
Pre-processing	Detect outliers (SANDBHOR; CHAPHALKAR, 2019)	3094 cases of property Indian sale instances;
Summarization	Summarization of legal texts (TRAN et al., 2020)	COLIEE 2019 dataset;
	Creation of a bilateral investment text (ALSCHNER; SKOUGAREVSKIY, 2017)	Collection;
Text generation	Dialogue system (JOHN et al., 2017)	Collection.
Theoretical	Analyze the representation of neurons (BORGES; BORGES; BOURCIER, 2003)	Collection.

Table 2 presents a sample of each category, objective, and corpus utilized to train the models (the complete list is presented as supplementary material). We founded 47 data sets. An increment in the size of the corpus has been observed in recent years. For example, in the work of LI et al. (2018), a model with 2,679,443 patents was trained, while an older work as the one from Bourcier et al. (1999) used 378 judgments of public order. It can be highlighted that the quality of results in DL models dramatically depends on the size of the corpus (SHAHINFAR; MEEK; FALZON, 2020). It was identified that 31 of the 47 corpus were published between 2017 to 2019, reflecting the electronic availability of data has increased in recent years.

3.2.3 Works published by journal

Fig 4b reports the percentage of articles published by journal from a total sample of 138 works. We identified AI and Law (21.7%), ICAIL (8.76%), and JURIX (2.92%) as the specialized journals/congress proceedings that concentrate most of the sample, 32.85%. By specialized, we mean journals/congress whose scope mainly publishes studies of AI systems used in the legal domain. The COLIEE contest on legal information and extraction was the only competition conference we identified in the specific scope of legal documents. Cardozo Law Review was the only journal in the law area that appeared in our research. Interestingly, the construction area appeared within two journals JCE Management and KSCEJ of Management. For space reasons, the plot shows only the journals with two or more publications. The ones with only one article were condensed into the category “other.”

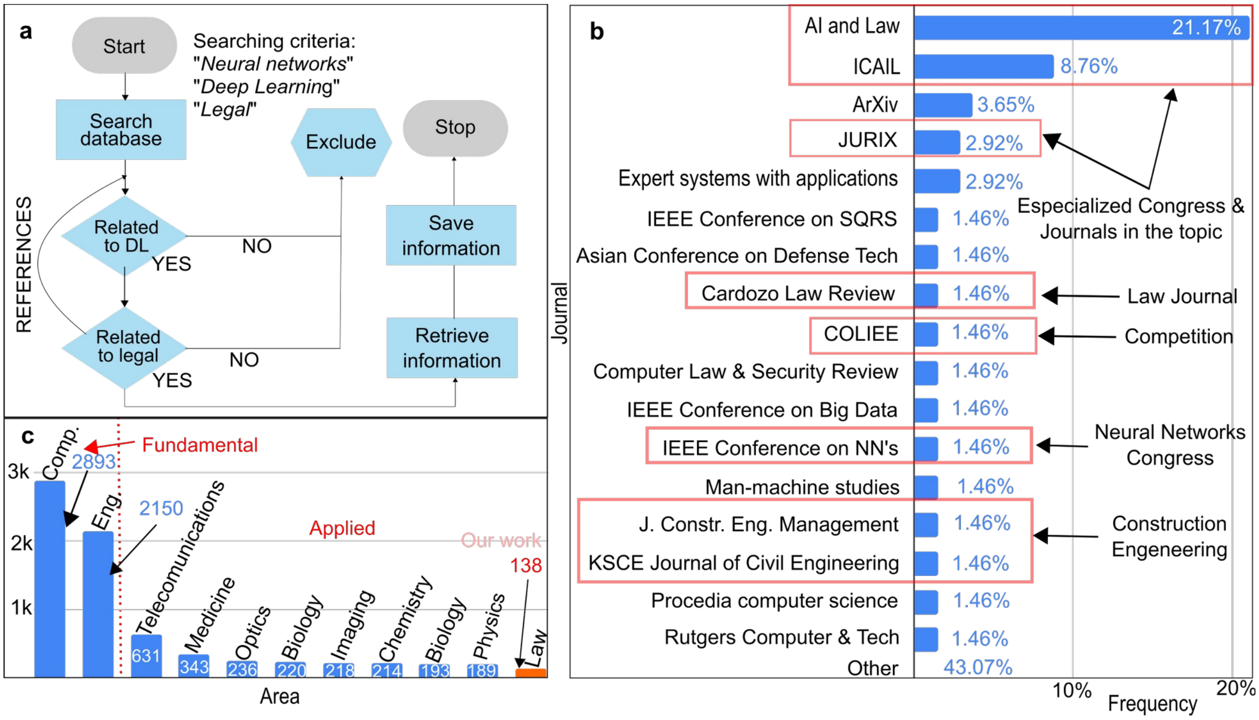


Figure 4. Research methodology and frequency of articles in the legal domain with DL as a primary methodology. (a) Diagram of the three main stages for retrieving our sample: identifying the survey target, selecting the relevant articles according to criteria and retrieving the relevant information of the article. **(b)** Frequency of works by Journal & Congress. The research sample contains 138 works mainly concentrated on the specialized journals AI and Law, JURIX, and ICAIL. Some also appear in journals dedicated to NNs, such as the IEEE conference on NNs. **(c)** The number of publications by area of knowledge using DL as the central methodology. Most publications are concentrated among Computer Science and Engineering areas (fundamental research). Among the rest of the areas (applied research), "Law" appeared in the last position with 138 works.

These results revealed that DL works in the legal domain are concentrated within three journals & specialized congress proceedings. External publications show a low rate of reported works. In **fig 4c,** we compare the number of publications using DL in the legal domain compared to other areas. We used as reference the bibliometric review from Li et al. (2020) that cited the number of works using DL among different research areas. We divided the results into two groups: fundamental and applied. The first group refers to areas that traditionally performed fundamental AI research (Computer Science and Engineering). The second group to areas where AI is used as support (applied). As observed, works of DL in the legal domain belong to the applied research group. Our sample has 138 publications (our complete sample), 51 less than the closer reported area (physics) with 189. However, this result reflects a lack of interest in applying this methodology in the legal domain.

3.2.4 Collaboration network

Figure 5 shows a representation of the collaboration network from the selected sample. Using a visual inspection, we identified the four most prominent groups according to the time and works published: Connectionism (1), NN's (1), and (2). The circle size is proportional to the number of publications of each author ranges from 1 to 6, as shown in the **fig 5a**. The authors that centralize the groups are Dieter Merkl (CHALKIDIS; ANDROUTSOPOULOS, 2017; MERKL, 1995a, 1995b; MERKL; SCHWEIGHOFER, 1997; MERKL; SCHWEIGHOFER; WINIWATER, 1995; MERKL; SCHWEIGHOFFER; WINIWARTER, 1999), who wrote most of his articles during the 90s in “Connectionism” (a term utilized to describe NNs). Karl Branting (BRANTING, 2017; BRANTING et al., 2018; SADEGHIAN et al., 2016, 2018; SARTOR; BRANTING, 1998) centralized the “NN's.” He has been an influential author from 2000 to recent years. Finally, Chalkidis (CHALKIDIS; ANDROUTSOPOULOS, 2017; CHALKIDIS; ANDROUTSOPOULOS; MICHOS, 2017, 2018; CHALKIDIS; KAMPAS, 2018) and Adebayo (JOHN et al., 2017; NANDA et al., 2017) [19], [29] centralize the DL groups. The plot also highlights two other essential authors Zeleznikow that published four articles (OATLEY; EWART; ZELEZNIKOW, 2006; STRANIERI et al., 1999; STRANIERI; ZELEZNIKOW, 2006; ZELEZNIKOW; VOSSOS; HUNTER, 1993). With his four publications, Philipps pioneered the topic (BROWN et al., 2020; PHILIPPS, 1989a, 1989b, 1991).

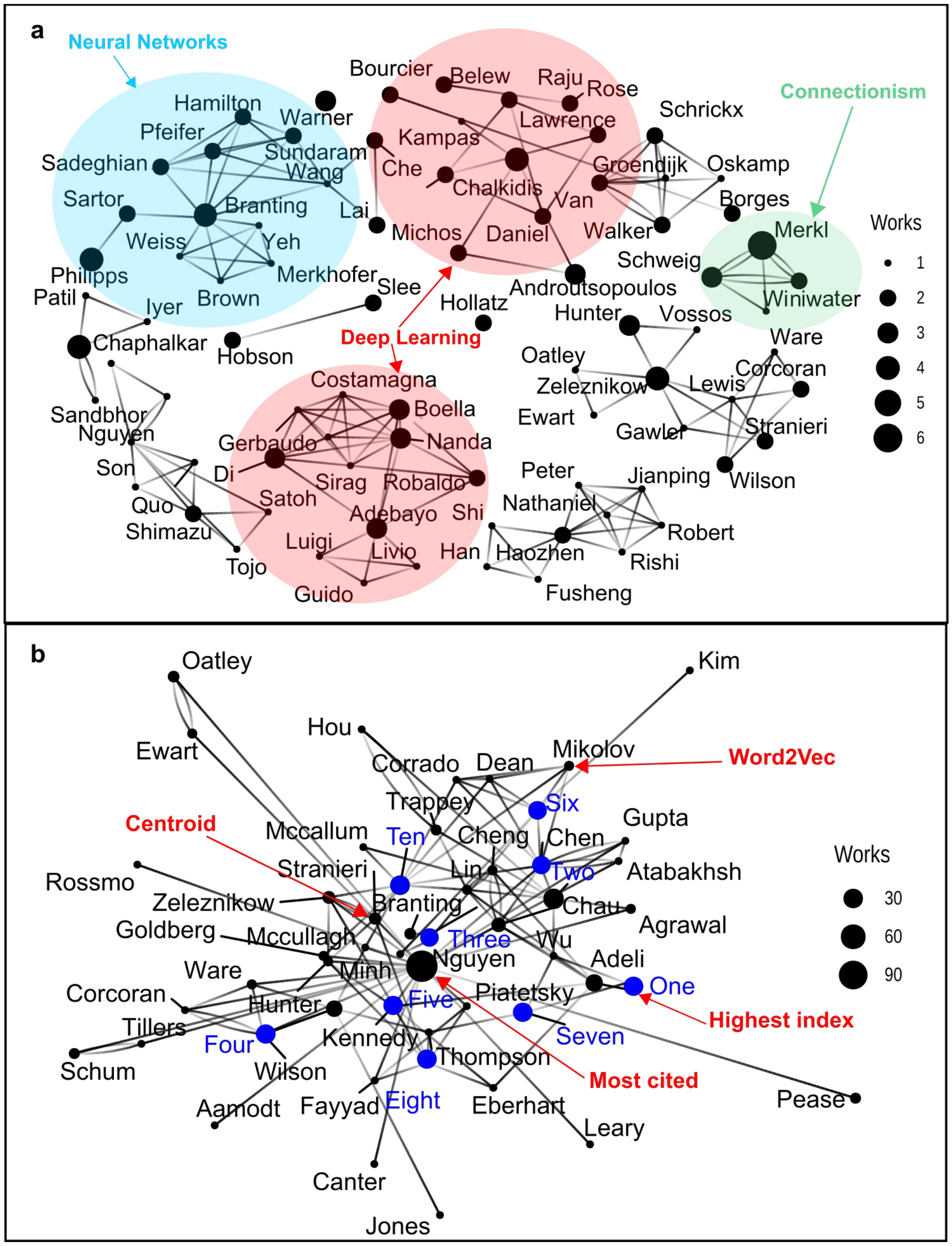


Figure 5. Network graphs of co-authorship and studies with high impact on DL works applied to the legal domain. (a) Co-authorship network of the selected works from the authors with more publications in the topic. The size of the circle is proportional to the number of articles published by their corresponding authors. We identified three groups that mainly depend on the publishing time of their works. They are Connectionism (the 80s), NN's (90s, 2000s), and DL (since 2012). The authors that concentrate on these groups are Merkl, Branting, and Chalkidis **(b)** Network of works that have been highly-cited connected with their respective citations. The works are identified by the numbers: “one” till “ten.” The authors five (OATLEY; EWART; ZELEZNIKOW, 2006) and three centralize the network. While the well-known work of Mikolov et al. (MIKOLOV; YIH; ZWEIG, 2013) appears as a reference from the network. A description of the works is depicted in Table 2.

In the last section, we analyzed the most cited articles from our sample. However, we observed that the age of the publication had a considerable impact on our measure. This means that older articles have a more significant number of citations. To mitigate this impact of time, we created a rate index consisting of the number of citations divided by the number of years from the publication date.

Table 3 shows the top 5 articles of the sample according to an index composed of citations divided by years of publication. The ranking is led by Chau, 2007 (CHAU, 2007) that applied NNs to predict outcomes from litigation construction disputes. This work introduced one of the first NNs models that demonstrated accurate results in a sector with one of the most complex litigation processes. The second one is Trappey et al., 2006 (TRAPPEY et al., 2006) that developed a classification model for patent documents. An area that has recently acquired an interest (ABOOD; FELTENBERGER, 2018). The third one is Branting, 2017 that questioned the capabilities of logic and AI-based methodologies. He also proposes that an intelligent system should be composed of both methodologies (logic & AI). The fourth from Corcoran, 2003 (CORCORAN; WILSON; WARE, 2003) proposed a crime incident forecast method by focusing on geographical areas of concern. The fifth (OATLEY; EWART; ZELEZNIKOW, 2006) from Oatley, 2006 presenting a system to support police against Burglary Dwelling houses.

Complementing our analysis, we selected the ten most extensive indexes. We also plotted them on **fig. 5b** on a network with their corresponding references used as support. Their ranking tags the selected works to avoid overcrowding in the plot (e.g., “First” node refers to the highest index (CHAU, 2007) and “Four” to the fourth-largest). Through a visual inspection on **fig. 5b**, we found that the author with the highest centroid in the network is “Nguyen” (MORIMOTO et al., 2017; NGUYEN et al., 2018; SON et al., 2016; TRAN et al., 2020), who has developed his work in applications such as recognizing legal parts, summarization and legal questioning answering using LSTM and CNN neural architectures. Those are prominent topics in the area. The second centroid is “Branting” (BRANTING, 2017; BRANTING et al., 2018; BROWN et al., 2020; SADEGHIAN et al., 2016), who appeared both as a high-cited author and with the third highest-index work “Three.” In this work, “Three” (OATLEY; EWART; ZELEZNIKOW, 2006), Branting et al. 2017 describe approaches for intelligent legal machines and has been a high cited reference by the AI & Law community research in recent years.

The highest impact “First” (CHAU, 2007) does not centralize the network. This result is due to the low quantity of references utilized. Finally, another well-known work that appears in the network as a reference is the one from “Micholov et al.” (MIKOLOV; YIH; ZWEIG, 2013) who developed the seminal technique of Word2vec, which objective is to represent a text into a vector space, a fundamental processing tool in DL methodologies.

Table 3. Top 5 cited works according to the proposed index.

Number	Authors	Index	Objective
One	(CHAU, 2007)	19.92	Predicts the outcome of construction claims.
Two	(TRAPPEY et al., 2006)	10.69	Propose a method for document patent classification.
Three	(BRANTING, 2017)	9	Discuss capabilities and challenges of logic and data-centric models.
Four	(CORCORAN; WILSON; WARE, 2003)	6.69	Proposes a method for crime incident forecast by focusing on geographical areas of concern.
Five	(OATLEY; EWART; ZELEZNIKOW, 2006)	5.92	Propose a system to support police against Burglary Dwelling houses.

From this systematic research, we found clear evidence of the rising interest in applying DL as a method in the legal domain and that is suitable to be used in our problem Legal Judgment Prediction, described in the following section 4. As shown in **fig 4**, 16% of all sample articles were published in 2018, while in 2014, only 2% of articles were published. The legal domain has been lagging in applying state-of-art computational methodologies. For example, Word2Vec (STRANIERI et al., 1999), a seminal development for NLP with DL, was first proposed in 2013, while the first work that used this method in the legal documents, appeared in 2018 (BANSAL; SHARMA; SINGH, 2019). This reflects the lack of transdisciplinary effort between computational and legal areas. DL in the legal domain is in the early adoption stage and will seemingly increase in the coming years.

Publications of DL in law are concentrated in a few specialized publications. We believe that these phenomena occur because the law area depends on particular knowledge that limits researchers from quantitative areas such as Computer Science to perform studies on the topic. On the other side, researchers from the legal area have not historically focused on using quantitative methodologies. It can be observed in the generated sample that only two works (PHILIPPS, 1991; THAGARD, 1991) were published in law journals. Hence, an approach involving both groups will improve the quality and understanding of the models. The availability of resources such as the increase in public legal datasets will escalate the collaboration from interdisciplinary areas such as Computer Science and Law. Our co-authorship analysis has shown that networks of researchers were deployed according to the time research period. We identified those in three groups: Connectionism, NNs, and DL. It is visually evident that the number of researchers during the DL period has increased consistently. This phenomenon is due to the availability of better hardware and larger data sets (for example (LAI; CHE, 2009) uses 65 patent infringement lawsuits (LI et al., 2018) 2,679,443 patents to train their models). Finally, our author network plot showed the two main groups of DL research. Those are centered by the authors Chalkidis and Adebayo, and Nguyen and Branting, who are the most highly-cited authors used.

4. LEGAL JUDGMENT PREDICTION

This chapter intends to explain the Legal Judgment Prediction (LJP) problem, which is the basis of our proposed framework. The chapter is divided into four topics:

- A general description of how a litigation process takes place.
- The mathematical formulation to represent the LJP.
- Types of text representations as vector spaces.
- Review of published works that have solved the LJP problem.

4.1 The process of a litigation

A legal proceeding or lawsuit is a systematic procedure where a dispute between two parties is decided in court. Three participants generally characterize a lawsuit:

- A *petitioner* or complainant who is the party that promotes a legal action.
- A defendant party who is indicted for committing an offense.
- And the institution named the *court* with the authority to judge or adjudicate.

The procedure to conduct a lawsuit is called litigation (HERR, HAYDOCK & STEMPEL, 2018). The overall process of litigation involves three stages: initial petition, analysis, and resolution. The *initial petition* (statement of claim) is the starting point of the process. It is the document where the petitioner describes its claims. Brazilian legislation states that the initial petition must contain (art 329):

I.- The type of judgment.
II.- Identity of people and organizations involved in the legal case.
III.- The claim.
IV.- Value of the claim.
V.- Proofs that identify the veracity of the facts based on a legal basis.

VI.- Option of mediation schedule.

The *analysis* involves examining the facts, both from the petitioner and defendant, by the court. The *resolution* is the decision taken by a court, win or lose. If one party disagrees, the decision can be appealed and goes to further instances.

An illustrative example of the overall process is depicted in **fig. 6**: A person named *A* buys a TV set broken from store *B*. Person *A* wants a complete refund of his money, but store *B* does not want to make the refund. Store *B* argues that the TV was in good condition at the moment when it was sold. Person *A* promotes a legal claim in court trying to enforce store *B* to give his money back. The starting point is the initial petition promoted by person *A* in court, including facts and evidence about the purchase. For example, how did he notice the problem? It will also include petitions that person *A* claims from store *B* and basement on a law that support claims for buyer *A* (ex. the law of consumer). The court authority will analyze the petition and give a resolution. If some of the parties do not agree, the resolution can be appealed and go through the next instance.

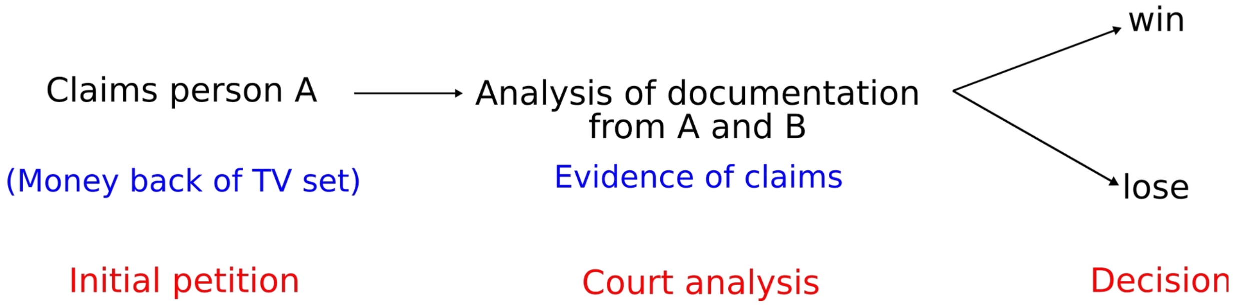
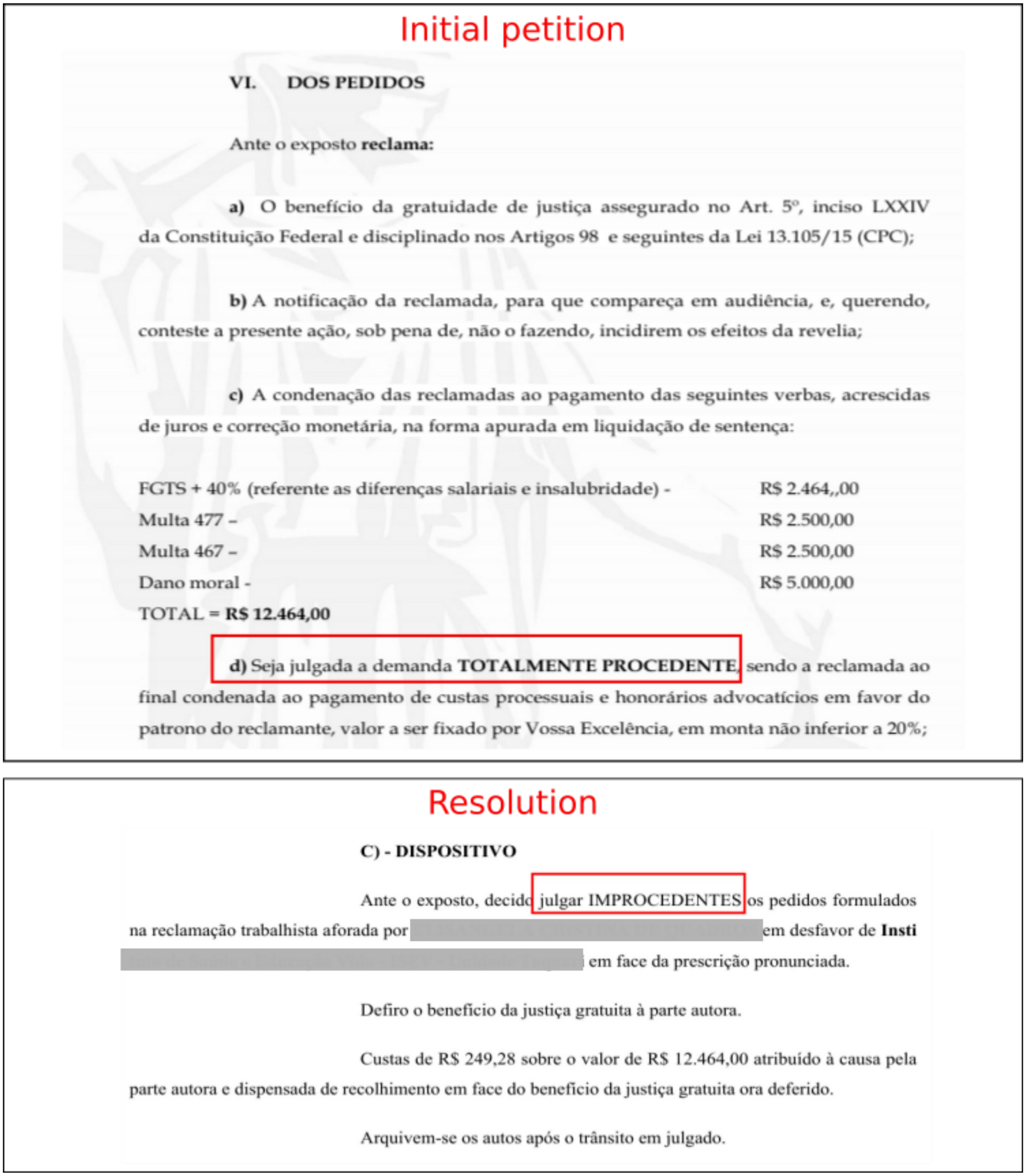


Figure 6. Stages of a litigation process. The overall process includes three stages: an initial petition, a court's analysis, and a final decision (resolution). The starting point is the initial petition where person *A* promotes a legal petition against store *B* (money back of TV set), then the court analyses the evidence from A and B, finally, a verdict (decision) is performed by the judge (win or lose). In legal terms, win & lose is represented as accepted or not accepted a petition.

A sample from a real lawsuit process is illustrated in **fig 7,** which includes the essential parts from the petition and resolution. The upper part of the document describes the initial petition composed be four claims from the petitioner, and the lower part shows the decision. All lawsuits have a similar structure that starts with the petition and follows a chronological order. A limitation of working with this type of documents is that the language used to describe the process usually contains words that are specific to lawyers. For people who are not involved in the legal area, these characteristic limits to have a clear understanding of the process. In our opinion, the language of these documents could be simplified to more conventional words that will enable people from outside of the legal area to understand in a clearer way the core of the process.



Initial petition

VI. DOS PEDIDOS

Ante o exposto **reclama:**

a) O benefício da gratuidade de justiça assegurado no Art. 5º, inciso LXXIV da Constituição Federal e disciplinado nos Artigos 98 e seguintes da Lei 13.105/15 (CPC);

b) A notificação da reclamada, para que compareça em audiência, e, querendo, conteste a presente ação, sob pena de, não o fazendo, incidirem os efeitos da revelia;

c) A condenação das reclamadas ao pagamento das seguintes verbas, acrescidas de juros e correção monetária, na forma apurada em liquidação de sentença:

FGTS + 40% (referente as diferenças salariais e insalubridade) -	R$ 2.464,,00
Multa 477 –	R$ 2.500,00
Multa 467 –	R$ 2.500,00
Dano moral -	R$ 5.000,00
TOTAL = **R$ 12.464,00**	

d) Seja julgada a demanda **TOTALMENTE PROCEDENTE**, sendo a reclamada ao final condenada ao pagamento de custas processuais e honorários advocatícios em favor do patrono do reclamante, valor a ser fixado por Vossa Excelência, em monta não inferior a 20%;

Resolution

C) - DISPOSITIVO

Ante o exposto, decido julgar IMPROCEDENTES os pedidos formulados na reclamação trabalhista aforada por [redacted] em desfavor de **Insti** [redacted] em face da prescrição pronunciada.

Defiro o benefício da justiça gratuita à parte autora.

Custas de R$ 249,28 sobre o valor de R$ 12.464,00 atribuído à causa pela parte autora e dispensada de recolhimento em face do benefício da justiça gratuita ora deferido.

Arquivem-se os autos após o trânsito em julgado.

Figure 7. Sample of a real lawsuit from a labor court. In the upper is depicted the petition composed by a set of 4 petition : a) "benefício da gratuidade" free legal assistance, b) "A notificação da reclamada" notification to the accused party, c) a total payment of R$ 12,464 d)"TOTALMENTE PROCEDENTE" all decisions to be accepted. In the lower part it is illustrated the resolution defined by the word "IMPROCEDENTES" so the petitions were not accepted and the claimer lose the case. The involved people were unidentified.

4.2 Problem formulation

In mathematical terms, the problem of a legal litigation decision is defined as the Legal Judgment Prediction problem (LJP) (YANG, JIA, ZHOU, & LUO, 2019). The LJP aims to predict the judgment results of legal cases according to the factual descriptions. Formally, the LJP is described as a supervised binary text classification problem, where the input is a starting petition X, and the output is a binary label $y \epsilon \{0,1\}$ with a corresponding probability. This indicates the loss or wins a legal dispute (KATZ et al., 2014; KOWSRIHAWAT; VATEEKUL; BOONKWAN, 2018; SHARMA et al., 2015). The problem will be solved using a DL architecture (LECUN; BENGIO; HINTON, 2015), thus the objective will be to optimize the cost function (MIYATO; DAI; GOODFELLOW, 2016):

$$J(w) = \frac{1}{M} \sum_{m=1}^{M} L(\hat{y}^{(m)}, y^{(m)}) \tag{1}$$

where:

M is the sample size.

$\hat{y}^{(m)}$ is the predicted probability denoted in the logistic function $\frac{1}{(1+e^{-w^T x})}$ where w is a vector of the model parameters and x are the independent variables.

$\hat{y}^{(m)}$ is the assigned label 1 to win 0 for loss to the petition.

Text classification is a NLP problem that has the objective of discriminating a source of text into predefined classes (MIRONCZUK; PROTASIEWICZ, 2018). Formally, given a description $d \in X$ of a document, where X is the *document space* and $C = \{c_1, c_2, \ldots, c_j\}$ a set of classes, the objective is to learn a classifier or a classification function that maps documents to classes (MANNING; RAGHAVAN; SCHÜTZE, 2010):

$$\gamma : X \rightarrow C \tag{2}$$

Classes are also called categories of labels and are human-defined according to the needs of an application. Typically, the document X is high-dimensional. This learning is called supervised learning because it contains examples to teach the model how the function must be learned. As an example of a lawsuit outcome, a training set D of labeled documents $\langle d, c \rangle$ where $\langle d, c \rangle \in \langle X, C \rangle$ will be:

<*Person A wants its money back from store B because store B sold a broken TV to person A*, *Win*>[1]

The methodology for classifying text is broadly divided into six steps (MIRONCZUK; PROTASIEWICZ, 2018):

1) Data acquisition: The process of obtaining the documents either from public repositories or particular domains. It also includes pre-processing such as lemmatization and steam.

2) Data analysis and labeling: The process of allocating labels, single or multiple, for each instance.

3) Element construction and weighting: The process of transforms the text into a digital form.

4) Selecting and projecting elements: The process of constructing the elements and projecting the data into a lower dimension.

5) Functional learning: The methodology used to construct the model that learns to discriminate against a class, typically a Machine Learning technique.

6) Assessment: The metrics used to measure the performance of the algorithm. **Table 4** describes the phases and examples of work that describe or use the techniques.

Table 4. Stages to perform a classification problem.

Stage	**Methodologies**
Data acquisition	- Pre-processing techniques such as lemmatization and stemming (KORENIUS et al., 2004). - Some public data sets are Reuters (LEWIS et al., 2004), TDT2 (WAYNE, 2000), and WebKB (CRAVEN et al., 1998).
Data analysis and labeling	- Multi-instance learning (YANG et al., 2016).

[1] This particular example is a simplified version of a real lawsuit. In a real context, initial petitions have a minimum length of 3 pages.

Feature construction and weighting	Feature construction: - Keywords or phrases, including uni-grams, bi-grams, and n-grams (ABOU-ASSALEH et al., 2004; WANG; MANNING, 2012) - Taxonomies or ontologies (DE ARAUJO; RIGO; BARBOSA, 2017; LI; YANG; PARK, 2012); - Embedded features (BENGIO et al., 2003; COLLOBERT et al., 2011; DEVLIN et al., 2018; MIKOLOV; YIH; ZWEIG, 2013; PETERS et al., 2018); Weighting: - Term frequency (*tf*), Inverse term frequency document (*idf* frequency) and term-frequency inverse document frequency (*tf.idf*), uni-grams, bi-grams, and n-grams (CHEN et al., 2016; FATTAH, 2015; HADDOUD et al., 2016).
Feature selection and projection.	Feature selection: - Multivariate relative discrimination criterion (MRDC) (LABANI et al., 2018) - Feature unionization (JALILVAND; SALIM, 2017) Feature projection: - Principal component analysis (PCA) (AITCHISON, 1983) - Latent semantic index (DUMAIS, 1995); - Convex sparse PCA (CSPCA) (CHANG et al., 2016)
Model trains	- Naive Bayes (NB) (KIM et al., 2006; NG; JORDAN, 2002; RISH, 2001) - Hidden Markov Models (KANG; AHN; LEE, 2018; KUSHMERICK; JOHNSTON; MCGUINNESS, 2001; YI; BEHESHTI, 2009) - K-nearest neighborhood (BAOLI; QIN; SHIWEN, 2004; ZHANG; ZHOU, 2005) - Maximum entropy (ME) (NIGAM; LAFFERTY; MCCALLUM,

	1999; ZHU et al., 2005) Regression Classifiers (WOOFF, 2004; ZHANG; OLES, 2001) - SVM (JOACHIMS, 1998; SCHÖLKOPF; SMOLA; BACH, 2002; ZHANG; OLES, 2001) - DL (BORGES; BORGES; BOURCIER, 2003; KIM, 2014; SHARMA et al., 2015)
Evaluation methods	- Accuracy, precision, recall, and F-measure (FORMAN, 2003; SOKOLOVA; JAPKOWICZ; SZPAKOWICZ, 2006).

4.3 Text representation

NLP tasks require the text to be represented in a numerical vector space. Approaches to perform this procedure are divided into three categories (SOCHER; MUNDRA, 2016): Word Vectors, Singular Value Decomposition (SVD), and iteration methods. Word Vectors are the most basic methodology. The corpus is represented as R^{Vx1} one-hot vector encoding, with all 0's and 1's at the index of each word. This technique represents syntactic knowledge but lacks frequency and relationship information (semantic knowledge). The second category performs some word co-occurrence counts in a matrix and then a SVD over X is performed to estimate a USV^T where "U" is a mxm orthogonal matrix, S is a m by n diagonal matrix, and V is a nxn orthogonal matrix. Some of the most used SVD methodologies are the Word-Document Matrix (SCHUETZE, 1997), the Latent Semantic Analysis (DUMAIS, 1995), and Global Vectors for Word Representation (GLOVE) (PENNINGTON; SOCHER; MANNING, 2014) models. These methodologies provide information on frequencies but have some limitations: (1) the size of the matrix changes often is sparse since most words do not co-occur, (2) it is high dimensional and (3) has a quadratic cost to train (SOCHER; MUNDRA, 2016).

The third category, named iteration methods, optimize word representations by making use of local contexts. The first iteration method is the language models that assign a probability to a sequence of tokens, an *n*-gram. The most basic form of a language model is a bi-gram, where the probability of a word depends on the previous word. Formally, a bigram is represented as:

$$P(word_n|word_{n-1})=\frac{C(word_{n-1}}{C(word} \tag{3}$$

Where $word$ represents the n^{th} word of a sentence. Despite bigrams optimize numerical representation by using local contexts, they also have the limitation. They learn only pairwise connections.

A more refined language model approach is the pre-trained vector space representations (DEVLIN et al., 2018; PENNINGTON; SOCHER; MANNING, 2014; PETERS et al., 2018). With this method, a NN is trained over a massive corpus of data, usually of millions of words which enables to learn intrinsic properties of words, such as relationships and frequencies. These approaches have proved to be efficient in learning both syntactic and semantic attributes from words. Among the vector space representation models, the works from Bangui Bengio et al., (2003) and Collobert et al., (2011) were precursors of the technique. However, the Continuous Bag of Words (CBOW) (MIKOLOV; YIH; ZWEIG, 2013) was the first work that brought attention to the academic community. In the same direction, FastText an extended version of the CBOW improved the model by representing words as characters. For example, the word apple is the app, ppl, and pale (ignoring the starting and ending boundaries of words) (JOULIN et al., 2016). The main advantage of this process is that words that are out of the corpus can be take into account. Recently, the model Embeddings from Language Models (ELMO) (PETERS et al., 2018), Bidirectional Encoder Representations from Transformers (BERT) (DEVLIN et al., 2018), and GPT-3 (BROWN et al., 2020) are works based on DL architectures that have proved to be the state-of-art in the language modeling representation. The following **table 5** summarizes categories and proposed models for transforming a text into a numerical representation.

Table 5. Classification of methodologies for transforming a text into a numerical representation. On the left-side, each of the three categories: Word Vectors, SVD, and Iteration methods. On the right side, methodologies for each category. As it has been described, Word Vectors are the fundamental techniques. SVD provides discrete results while the state-of-art methods are the Iteration.

Category	Methodology
Word Vectors	One-hot vector encoding. Bag of words.

SVD	Word-Document Matrix. Latent Semantic Analysis. GLOVE.
Iteration	CBOW, TagLM, context2vec, FastAI, ELMO, CoVe, BERT, GPT-3.

4.4 Modeling legal court process

Advances in information retrieval have allowed academics to propose quantitative methods for estimating outcomes of court decisions, as information is stored in electronic form and can be processed by algorithms. One of the first works was from Rugers et al., (2004), who compared prediction outcomes of the United States Supreme Court (USSC) between a statistical model and legal specialists. The model was trained using a Random Forest model constructed with six features. The work cited that the model predicted 75% of cases correctly, while the experts got 59.1% right. The statistical model considered the outcome of 628 cases, and the legal experts did not have limitations on information to consult. Katz et al., (2014) published a highly cited work, as it was first one that used a high volume of legal petitions, sixty years of decisions by the Supreme Court of the United States (1953 -2013). The authors stated that the model correctly forecasted 70.9% of 7700 tested cases, used 100 variables, and applied an Extremely Randomized Trees (ERT) model. The study from Aletras et al., (2016) predicted outcomes of cases tried by the European Court of Human Rights (ECHR) using a Support Vector Machine (SVM) classifier. The authors argued to be the first systematic study of predicting cases based solely on textual content without feature engineering. The model was referred to have a 79% of accuracy on average, and results from work suggested that the “formal facts” of a case are the most important predictive factor.

Initial works of LJP were in the English language, particularly from the United States Supreme Court. However, recent studies that use databases in other languages were published, such as the CAIL2018, which contains 2.6 million criminal cases published by the Supreme Court of China and it is the basis for the only LJP competition found (Zhong et al., 2018), which consists of attending the maximum accuracy prediction according to a chronological list of litigation processes. **Table 6** lists the sample of works that proposes models for the LJP problem. As it can be seen, the accuracy of the models goes from 70.9% to 88.3%. Older models use conventional Machine Learning techniques such as SVM, while recent approaches are based on DL.

Table 6. List of works that proposes models to solve the LJP problem. The column from the left specifies its authors, the rest to the litigation collection (database), methodology and reported accuracy

Authors	Database	Methodology	Reported accuracy
(RUGER et al., 2004)	268 cases of USSC.	Classification tree with 6 features	75%
(MONTGOMERY; HOLLENBACH; WARD, 2012)	214 cases of USSC.	Ensemble Bayesian Model Averaging	77.10%
(KATZ et al., 2014)	Sixty years of decisions from the USSC. Tested over 7700 cases	Extremely randomized trees ERT with the manual feature of 100 variables.	69.7%
(ALETRAS et al., 2016)	584 cases of ECHR	Contiguous Word sequences with an SVM classifier.	79%
(SULEA et al., 2017)	126425 cases from French Supreme Court	SVM classifier trained on lexical features	75.9%
(LIU; CHEN, 2017)	584 cases of ECHR	Compared performance of SVM, logistic regression, Random Forest, bagging, and K-means.	73.4%
(YANG et al., 2019)	1,588,894 cases from the Chinese AI law challenge.	Multi-Perspective based BiFeedback Network (MPBFN) and a Word Collocation Attention (WCA) mechanism	88.3%
(KOWSRIHAWAT; VATEEKUL; BOONKWAN, 2018)	1,207 cases of Thai Supreme Court Cases	Bidirectional GRU Neural Network.	79.87%

	(TSCC)		
(CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019)	584 cases of ECHR	Hierarchical BERT-MODEL	82.00%

5. PROPOSED FRAMEWORK

To model our problem of LJP which is the basis for Contingent Liabilities estimation we propose a framework composed of three primary blocks:

(1) A pre-process section that transforms the raw files into a structured array form.

(2) A DL architecture section that convert the documents into a numerical tensor representation and estimates a probability.

(3) A similarity estimator section that provides the most similar documents to the one provided as input.

Fig. 8 depicts an illustration of the complete framework with its corresponding blocks, where its input (left) is a litigation petition document and its output (right) is the probability outcome and set of similar petitions to the estimated document.

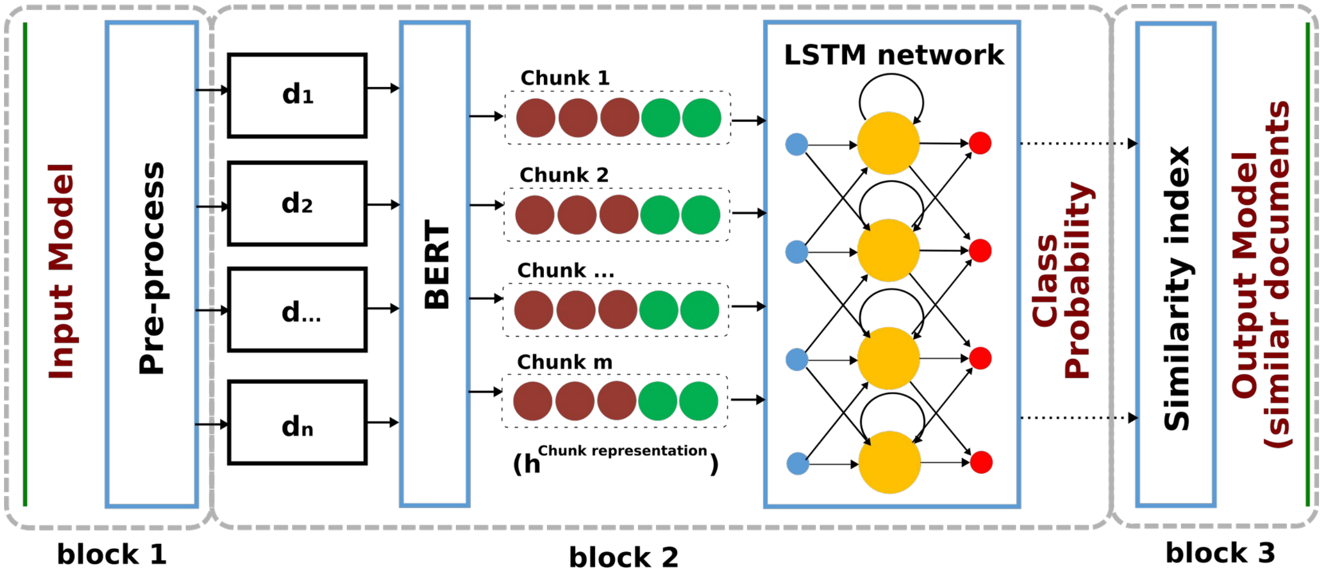


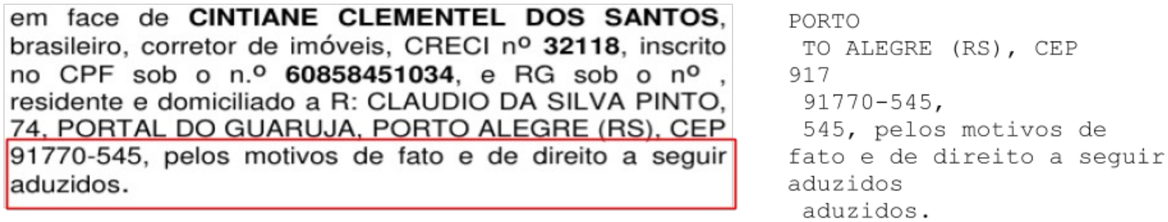
Figure 8. Proposed framework of the study. The framework comprises three main blocks. *Block 1* transforms raw lawsuits into a structured array suitable to train/predict the model by converting image files into text, detecting the type of outcome, and structuring the information. *Block 2* transforms the information into a vector representation by dividing long texts into chunks, representing a high dimensional tensor using a BERT architecture, unifying and assigning a class probability. *Block 3* provides a ranking of similar documents to identify intrinsic properties of lawsuits accepted/rejected (winners vs. losers) by estimating a similarity index.

5.1 Pre-processing input (block 1)

This section discusses the methodology used to perform block one that transforms the PDF documents to text and structures the information according to the required input to create a numerical tensor representation. The code used to perform this operation is provided as an attachment (a, b, c).

5.1.1 Documents to text

A basic assumption of a NLP process is to have the corpus in machine-encoded text. The sample we will use to train the framework was a set of PDF documents provided by the Brazilian labor court state of Rio Grande do Sul (*4 Tribunal Regional do Trabalho 4- TRT4*) processed by an Optical Character Recognition (OCR) engine, explained in more detail on section 6.2. The OCR processed the documents because, in their original form, they were printed and submitted by users. This process is used because it provides flexibility to scan different documents, *ex*. Photos, that frequently are used as proofs of the court From the sample of the PDFs provided by the court, we extracted their text using an open pdf to text extractor. However, the documents exhibited inconsistencies. The extracted text does not match with the ground truth. The following **fig 9** illustrates the problem:



em face de **CINTIANE CLEMENTEL DOS SANTOS**, brasileiro, corretor de imóveis, CRECI nº **32118**, inscrito no CPF sob o n.º **60858451034**, e RG sob o nº , residente e domiciliado a R: CLAUDIO DA SILVA PINTO, 74, PORTAL DO GUARUJA, PORTO ALEGRE (RS), CEP 91770-545, pelos motivos de fato e de direito a seguir aduzidos.

```
PORTO
 TO ALEGRE (RS), CEP
917
 91770-545,
 545, pelos motivos de
fato e de direito a seguir
aduzidos
 aduzidos.
```

Figure 9. Inconsistencies of a PDF document when the text is extracted. The left image shows a set of rows of a PDF of the sample. The right-hand one is the extraction of the last two lines of that text. As we can see, there are inconsistencies in this provision. The word “PORTO” is not extracted in the same line. The numbers 917 and 545 and the word “aduzidos” are repeated.

To correct the problem, we look upon literature for open-source OCRs engines that could transform images, with more accuracy, into text. The state-of-art OCR engines are based on NN's, particularly Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997), a kind of Recurrent NN's

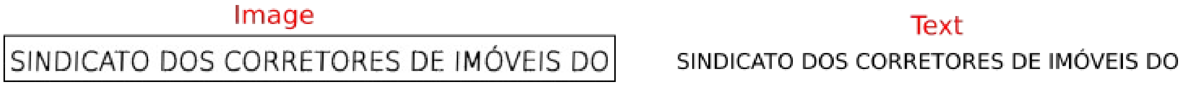
(RNN) (BREUEL et. al, 2013; Wick, Reul, & Puppe, 2018b). LSTMs methods reported accuracies beyond 98% (Wick, Reul, & Puppe, 2018a) on a variety of typographies, ranging from early printed books to modern prints.

Based on benchmark results from (WICKET AL., 2018B), we tested four LSTMs OCR based algorithms: OCRopus (BREUEL ET AL., 2013), Tesseract (SMITH, 2007), Calamari (WICKET AL., 2018A), and Kraken (ROMANOV, et al., 2017), using the default models. The first engine we tested was OCRopus, which was the pioneering algorithm to implement bidirectional LSTM networks. However, results on the sample were not satisfactory, as reported in the literature. Tesseract was the second engine we tested. It is the oldest of the engines we analyzed, developed since 1984. The results were superior to the ones from OCRopus. However, during this step, we realized that to have successful results, fine-tuning or training from scratch must be performed. However, Tesseract lacks the flexibility to perform these two procedures. Next, we tested with Calamari, which implements a combined deep CNN-LSTM network structure instead of the shallow LSTM used by OCRopus (WICKET AL., 2018B). However, it lacked elements to be a complete OCR framework, including the flexibility to train. Kraken was the last engine we tested. It showed the best results, and it also provides flexibility to train a model either from scratch or fine-tuning.

The first tests we performed were using the default pre-trained models provided by the OCR engines. However, for the documents we needed to perform, they showed limitations. The provided models were trained in English, while the lawsuits are in Portuguese. That contains accents and characters different from English. They are trained in conventional layouts, while the lawsuit is not always in conventional layouts. They are trained with the most used typographies, while some lawsuits do not use conventional typographies. In summary, the task to process the lawsuit documents by an OCR will need to be performed by training a custom model for this purpose.

LSTMs engines work as a conventional supervised Machine Learning Algorithm trained using image/text pairs. Two files are provided: one from the image and the other as a text file. This capability enables to train of specific documents where conventional OCRs do not have the capability. For example, perform complex tasks such as number recognition using street-level photos

(GOODFELLOW ET AL. 2013). **Fig 10** shows a simple example of how the image/text pair data are provided.



SINDICATO DOS CORRETORES DE IMÓVEIS DO

SINDICATO DOS CORRETORES DE IMÓVEIS DO

Figure 10. Sample of image-pair data. The image from the left is a sample of a scanned image. The one from the right is its transcription (text file). They are provided as two files: one for the image and the other for the text file.

The straightforward method to create the data is by human labeling. A person transcribes the text from the image. One requirement is to provide images/pairs as line texts, not the whole document (ROMANOV ET AL., 2017). Kraken enables us to perform this stage flexibly. It subsets the transcription of the whole document into its corresponding lines of text. To perform the first test, we manually transcribe a document lawsuit, trained the model, and tested in an out-of-sample line from the same document. To check the results, we count the number of errors from the text.

The results were with 0 errors. However, this process was biased and had its limitations. The document we transcribed contained one type of typography, while the universe of documents to be processed contains different types. Tests were made in the same document. So, the model overfitted. It was a biased test. We realized that making the transcription by human will be high time-consuming so we look for alternatives. Literature suggests a second approach to train the model by creating synthetic data (ROMANOV ET AL., 2017; SIMISTIRA ET AL., 2015). Kraken also offers a module to create synthetic data. From a text, an image is created that can be tuned in distortion, type of typography, size, and width. Thus, tuning these parameters, a universe of substantial typographies can be reached. **Fig 11** illustrates an example of synthetic data.

entidade representativa de classe social sem fins

Figure 11. Sample of synthetic data. The phrase "entidade representativa de classe social sem fins" is transformed into an image with distortion. This example is tuned with extreme parameters to illustrate the concept. Mixing. The parameters allow the creation of the different typographies (*ex*. Bold).

The first test we performed was using a corpus of 7776 words representing the Portuguese language. We created artificial data using default parameters. But the model did not learn. Results were most flawed than with default pre-trained models. We realized that the text-to-data creation has to be provided in lines (sentences), not only words. We used documents similar to our domain. So, we look up templates of labor cases from Brazil. We used a set of 12 templates and merged them into a single document. That gave us an approximate total length of 1500 lines, which is the suggestion of the algorithm.

We created artificial data using the set of petitions. We first used the default parameters of distortion and noticed that the model start learning. We continued using this strategy and tuning the parameters. We adjust parameters based on a visual inspection between the data to be tested and the created artificial data. **Table 7** lists results with different parameters. As it can be seen, the worst result is the model with an error rate of 38% and the best with 2%, which coincides with reported results (Wicket al., 2018a) of state-of-art OCRs.

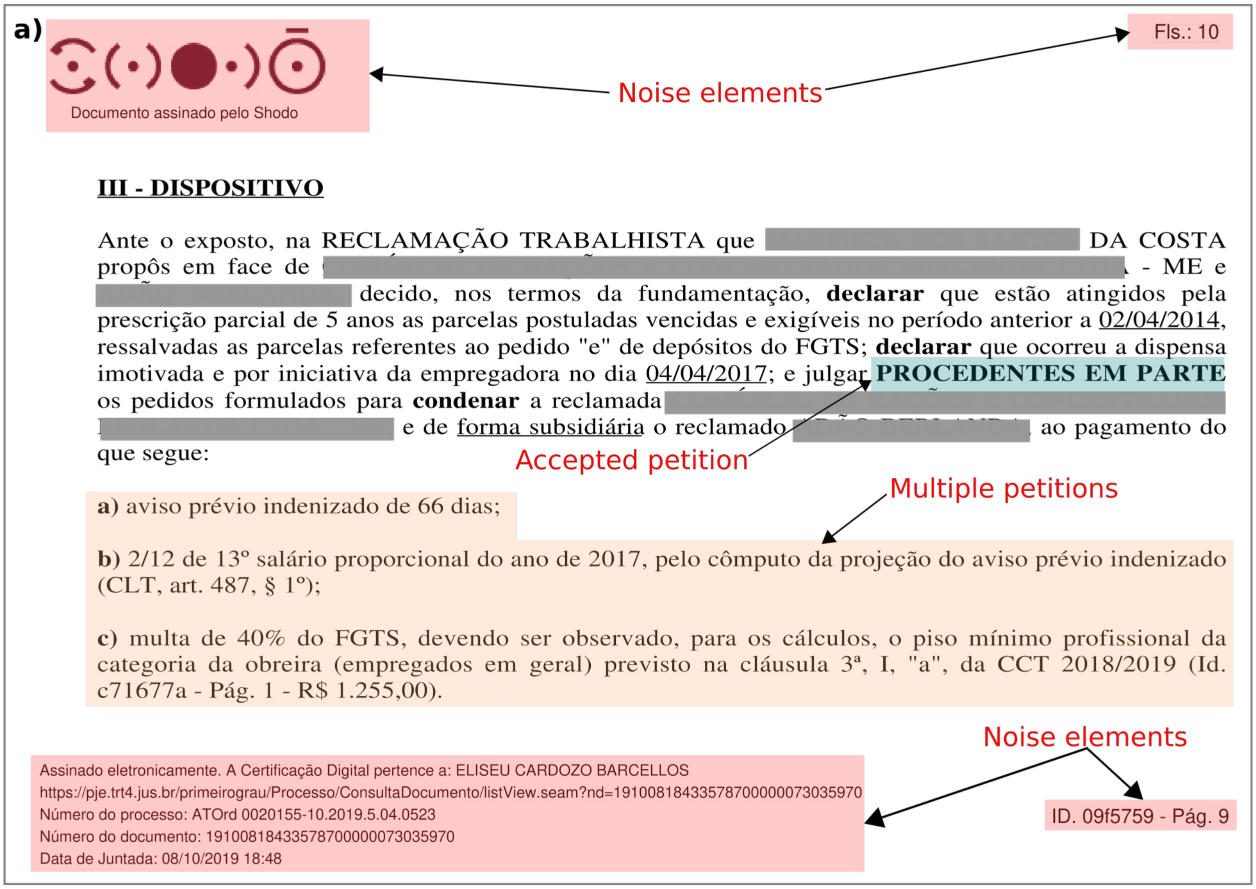
Table 7. Results of synthetic data with different parameters. The left column depicts the model used, and the right its error rate. Results are ordered by error rate in increasing order. The worst result was with the corpus of individual words. The model does not learn there, and the best with a 2% of error. Four parameters were tuned: typography size (s), distortion (d), font size (fs) and font-weight (fw). The first line corresponds to the synthetic data created from the corpus of individual words. The subsequent is from the corpus of petitions templates. Parameter (s) is described as size. However, evidence suggested that accuracy is better controlled by the parameter (fs). Few is referring to font-weight, which in practice is tuned to have bold typographies.

Model	**Error rate**
Individual words	Not learning
Petitions default	38%

Petitions s_12 d_0	34%
Petitions s_12 d_0 fw_400	31%
Petitions fw_350	27%
Petitions d_0 fs_64 fw_350	14%
Petitions d_0 fs_40 fw_350	9%
Petitions d_0 fs_47 fw_350	2%

5.1.2 Structuring text

After transforming PDF files into text, block 1 performs additional processes: (a) Remove noise elements that are not part of the process. (b) Detect verdicts within the text decisions as long as they are written as a free text, not as a single variable word (c) the decisions are long documents with a range between 3-120 pages, (d) each lawsuit involve multiple petitions, as these cases (labor court) usually involve multiple demands. Therefore, multiple decisions are decided in a single outcome document. **Fig. 12** illustrates an example of the exposed issues.


a)
Noise elements
Accepted petition
Multiple petitions
Noise elements

Documento assinado pelo Shodo

Fls.: 10

III - DISPOSITIVO

Ante o exposto, na RECLAMAÇÃO TRABALHISTA que [redacted] DA COSTA propôs em face de [redacted] - ME e [redacted] decido, nos termos da fundamentação, **declarar** que estão atingidos pela prescrição parcial de 5 anos as parcelas postuladas vencidas e exigíveis no período anterior a 02/04/2014, ressalvadas as parcelas referentes ao pedido "e" de depósitos do FGTS; **declarar** que ocorreu a dispensa imotivada e por iniciativa da empregadora no dia 04/04/2017; e julgar **PROCEDENTES EM PARTE** os pedidos formulados para **condenar** a reclamada [redacted] e de forma subsidiária o reclamado [redacted] ao pagamento do que segue:

a) aviso prévio indenizado de 66 dias;

b) 2/12 de 13º salário proporcional do ano de 2017, pelo cômputo da projeção do aviso prévio indenizado (CLT, art. 487, § 1º);

c) multa de 40% do FGTS, devendo ser observado, para os cálculos, o piso mínimo profissional da categoria da obreira (empregados em geral) previsto na cláusula 3ª, I, "a", da CCT 2018/2019 (Id. c71677a - Pág. 1 - R$ 1.255,00).

Assinado eletronicamente. A Certificação Digital pertence a: ELISEU CARDOZO BARCELLOS
https://pje.trt4.jus.br/primeirograu/Processo/ConsultaDocumento/listView.seam?nd=19100818433578700000073035970
Número do processo: ATOrd 0020155-10.2019.5.04.0523
Número do documento: 19100818433578700000073035970
Data de Juntada: 08/10/2019 18:48

ID. 09f5759 - Pág. 9

Figure 12. Sample of an accepted lawsuit decision (win). The lawsuits are provided with noise elements uniformly distributed on all documents. The outcome is defined in the phrase “PROCEDENTES EM PARTE” (accepted petition) that is write inside a text, not as a single word and usually appears at the end of the decision document. Multiple petitions are decided into a single document (*aviso prévio*, *salário proporcional*, *multa*.)

In addition to the described limitations the verdict of a petition is not write uniform (eg. accepted or not accepted). Each judge has its style to write, *e.g.,* to define that a petition was lost. They can write “improvement,” “reject,” or some custom words/phrases, and some cases do not have a decision. **Fig. 13** shows a sample that illustrates this issue.

To overcome these limitations, we developed an algorithm that performs the following steps

1) Store all text petitions and their corresponding text outcome into an array matrix.

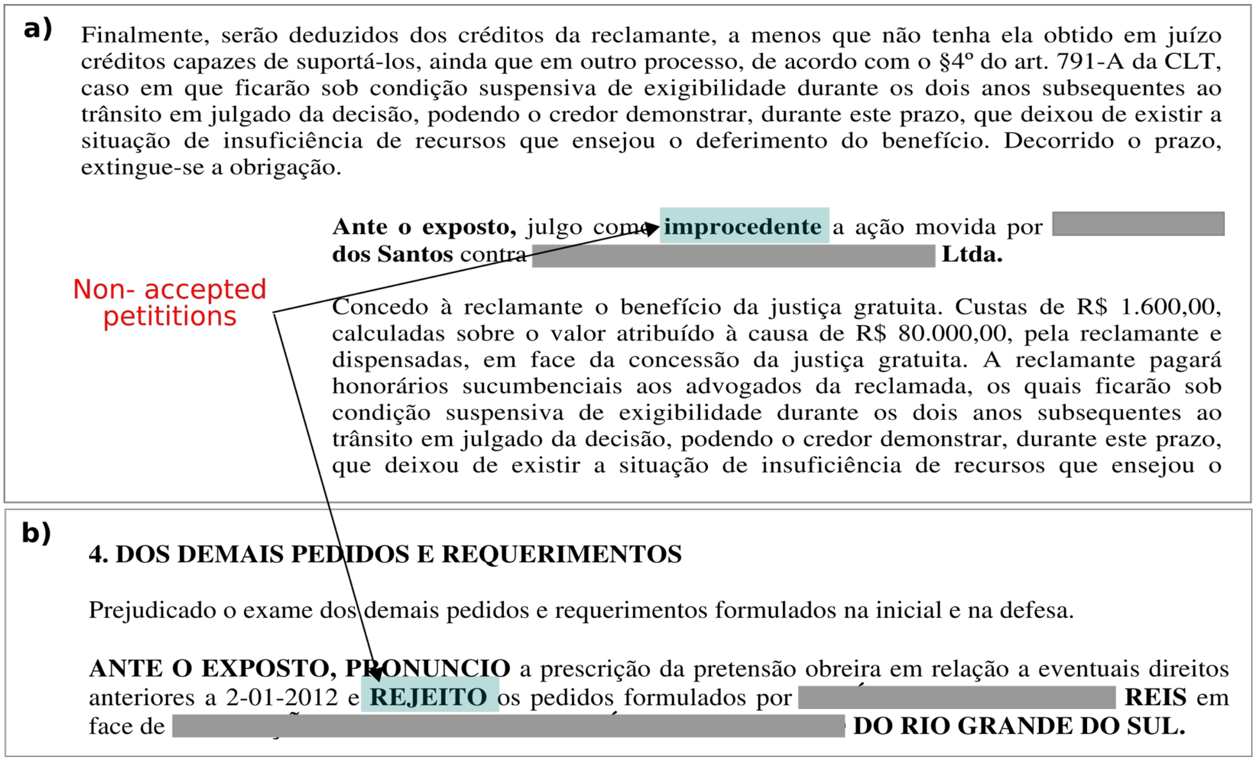
2) Sample documents, visually inspect to identify systematic noise elements such as logos, and remove them from the texts.

3) Sample some of the decision documents and identify the most frequent terms that judges use to define an outcome (e.g., "*procedentes em parte*" = win, "*improcedente*" = loose). Search these terms over all the decision documents and classify each as a *win* or *lose*. Some terms appear multiple times among a decision, *e.g.*, starting the paragraph, the term "lose" (*improcedente)* appeared, referring to a historical decision and at the end appeared "win" (*procedente).* We identified that final decisions are written in the last paragraphs. We, therefore, decided that if we could find more than one term, the one used would be the one that eventually appeared. Some cases matched none of our criteria either. The judge did not have the elements to make a decision or because the terms did not match our criteria.

4) We found that most win decisions were "partly win" because it is common that multiple petitions are performed within one lawsuit in labor cases, therefore multiple petitions are decided.

5) Finally, we make a qualitative inspection to validate our process.

6) Our output is a structured array of the form *mxn* where *m* corresponds to the i*th* lawsuit and *n* the petition text and its corresponding outcome.


a)
Non- accepted petititions
b)

Finalmente, serão deduzidos dos créditos da reclamante, a menos que não tenha ela obtido em juízo créditos capazes de suportá-los, ainda que em outro processo, de acordo com o §4º do art. 791-A da CLT, caso em que ficarão sob condição suspensiva de exigibilidade durante os dois anos subsequentes ao trânsito em julgado da decisão, podendo o credor demonstrar, durante este prazo, que deixou de existir a situação de insuficiência de recursos que ensejou o deferimento do benefício. Decorrido o prazo, extingue-se a obrigação.

Ante o exposto, julgo como **improcedente** a ação movida por [redacted] **dos Santos** contra [redacted] **Ltda.**

Concedo à reclamante o benefício da justiça gratuita. Custas de R$ 1.600,00, calculadas sobre o valor atribuído à causa de R$ 80.000,00, pela reclamante e dispensadas, em face da concessão da justiça gratuita. A reclamante pagará honorários sucumbenciais aos advogados da reclamada, os quais ficarão sob condição suspensiva de exigibilidade durante os dois anos subsequentes ao trânsito em julgado da decisão, podendo o credor demonstrar, durante este prazo, que deixou de existir a situação de insuficiência de recursos que ensejou o

4. DOS DEMAIS PEDIDOS E REQUERIMENTOS

Prejudicado o exame dos demais pedidos e requerimentos formulados na inicial e na defesa.

ANTE O EXPOSTO, PRONUNCIO a prescrição da pretensão obreira em relação a eventuais direitos anteriores a 2-01-2012 e **REJEITO** os pedidos formulados por [redacted] **REIS** em face de [redacted] **DO RIO GRANDE DO SUL.**

Figure 13. Samples of two non-accepted lawsuit petitions. (a) The term to identify that a petition was not accepted is defined in the word "improcedente" **(b)** A second sample of a non-accepted petition, but with a different writing style. Here, it is write with the word "rejeito." In both cases, the personal information of the involved people was removed.

5.2 Tensor representation (block 2)

Block 2 from the framework represents the text into a tensor form and estimates a class with its corresponding probability. This section comprises a Bidirectional Encoder Representations from Transformer (BERT) (DEVLIN et al., 2018) and a Long Short-Term Memory (LSTM) DL architectures. We used BERT because is the DL technique that has shown the most accurate results in NLP during the last years (LI et al., 2019). In addition with the LSTM, to overcome the maximum number of words (n<512) that BERT restricts - detailed in the following section. The code to perform the experiments is provided as an attachment 1.d.

5.2.1 BERT

BERT is a DL architecture that uses pre-trained models to perform specific problem solutions on custom datasets (*e.g.,* classification of a litigation process). The pre-trained models are trained on large corpuses that usually are texts from Wikipedia or book collections. This methodology has shown to be beneficial for NLP tasks as pre-trained models stores information from the large collection and complements by fine-tuning on a custom dataset (ALSENTZER et al., 2019). Its principle is based in the same way as humans process language by storing information and retrieving to perform a specific language requirement. Pre-trained models used to be only available in English. However,versions in other languages such as Brazilian Portuguese have been recently trained and provided to perform research (RODRIGUES et al., 2020).

BERT is pre-trained using a Masked Language Model (MLM) objective, where some tokens from the input are randomly masked, and the objective is to predict the original vocabulary. The architecture uses a bidirectional network that enables to consider words before and after the tokens. BERT is a model that contains between 110 and 345 million parameters in its base and large versions. So, training from scratch demands substantial hardware resources. For our particular problem of litigation predictions, we used pre-trained BERT base uncased model in English (DEVLIN et al., 2018) and

Brazilian Portuguese (RODRIGUES et al., 2020) as support, and then fine-tuned (trained) in our lawcase databases.

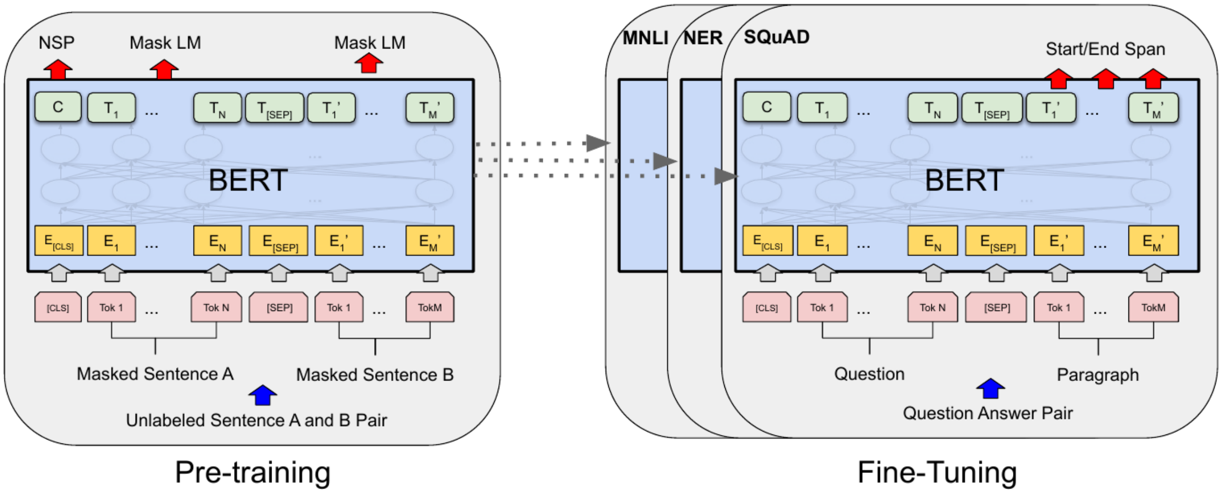


Figure 14. The process to train a BERT model from scratch. On the left side, the pre-training stage trains the model from scratch in extensive collections such as Wikipedia or book documents which usually take several days of training and demand high computational resources. On the right fine-tuning (training) for specific datasets on problems such as Named Entity Recognition (NER) and Text Classification. For the objective of our work, pre-trained English and Portuguese models were used a support and fine-tuned in our custom collection.

BERT model exhibits one important limitation. The maximum number of words (tokens) that can be processed for each text is less than 512. This limitation is due to the fact that most of the problems developed to train the models involve text datasets that satisfied this restriction - *e.g.*, the Google Play app reviews dataset (MCILROY et al., 2017), a widely cited problem that consist to classify according to reviews from users, has a maximum length of 250 words. But litigation processes collections have higher lengths of up to 20000 words, a difference of 100x. When texts excess the limitation, a commonly proposed approach is to truncate the number of words up to 512 as performed in the IMBD review dataset - a database that involves the classification of reviews from text films (ADHIKARI et al., 2019). This approach has succeeded for datasets such as the IMBD for the reason that the number of documents that surpass the restriction represents a small proportion of the entire sample. Therefore, truncating the texts do not take out important information. However, in our custom lawsuit dataset, almost all the samples exceed the limitation of 512 words and the maximum length of the texts is ~20000. As **Fig. 15** shows the number of words from our custom legal database dataset (TRT4) is almost 17x bigger than the IMBD (3071 vs 174) which evidence the limitations of working with conventional methodologies on large documents such as legal petitions.

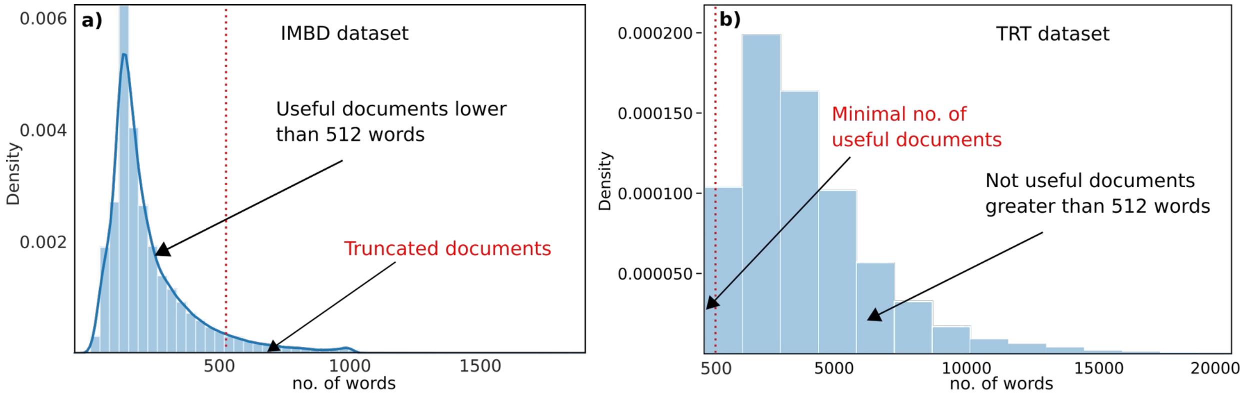


Figure 15: Frequency of words from conventional and lawsuits datasets.(a) The frequency of words from the IMBD reviews dataset. Most of the text has less than 512 words which is the maximum acceptable length of BERT. The ones that surpass the restriction are truncated with minimal information loss **(b)** The frequency of words from our custom litigation process database (TRT4). A minimal number of documents is suitable to be processed by BERT restriction less than 512 words. The maximum number of words from the lawsuits dataset is 20000, almost 10x more than from the IMBD reviews dataset.

To alleviate this limitation of size, we divided the text into parts (chunks < 512) trained (fine-tuned) separately with its corresponding class and then unified using a second DL structure - LSTM. The final block comprised a BERT-LSTM architecture.

5.1.2 LSTM

LSTM (HOCHREITER; SCHMIDHUBER, 1997) is a Recurrent Neural Network (RNN) that can process sequence elements. Therefore, it is suitable for time dependency situations (*e.g.*, speech recognition, time series forecasting). We chose LSTM as a second DL architecture to unify the chunks created by BERT as they followed an ordered sequence of elements. Formally we divided each document d into a sequence of $x_1, x_2, \dots x_m$ chunks. Where x_1 corresponded to the first document section of 512 < words, x_2 to the second document section of 512 < words, and x_m to the last document section of < 512 of words. We processed each x_i by BERT that provides a R^{768}vector representation for each $x_{i.}$ Thus, the final representation of the first architecture (BERT) for each document will be a R^{Mx768} , where M is the number of chunks. In the second stage, the LSTM architecture unified the vectors (chunks) into a single vector and estimated $P(c|d)$ that document d belongs to class c (accepted or non-accepted).

Sequence modeling problems depend on timely information that can have close or long dependencies. This requirement is observed in text structure as some words have a close dependency. e.*g.,* in the phrase “The president of France is Macron,” the word “Macron” depends on the previous side-by-side “The president of France.” But other phrases have a dependency on information from more prolonged periods, *e.g.,* information detailed at the beginning of the document, as the name of a person, is required to model a part of the text at the end of the document. This is why LSTM architectures have this name, as they can store information from Short (close) and Long (extended) periods. To unify the chunks created by BERT into the complete document, we identified the LSTM as a suitable architecture for the reason that the chunks follow an ordered side-by-side sequence (Short) and depend on information not necessarily together (Long).

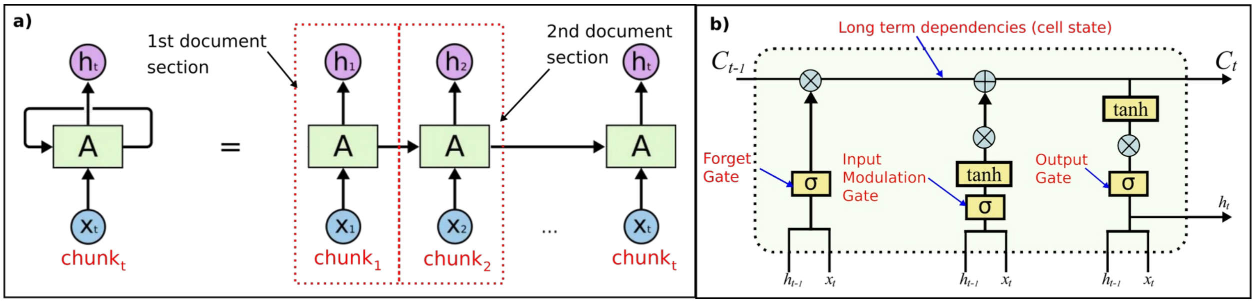


Figure 16. Representation of a RNN and LSTM cell. (a) Illustration of the time dependency of an RNN. On the left side, an input X (blue) is processed at each time *t* by a unit A (green) that stores information used to provide a feedback, and the rest of the information is sent to the hidden cell h_1. On the right, the same process is represented as a set of multiple NNs. The first one refers to input x_1 (chunk $_1$) that stores helpful information for the second input x_2 (chunk $_2$) in memory A up to time x_t (chunk$_t$). **(b)** An internal LSTM cell comprises 4 main sections. The LSTM cell uses input information from the current x_t, previous state $x_{t\ -1}$ (Short), and Long states $C_{t,}$ which is $_t$he Cell State (upper) - a memory that interacts over all the process and stores Long dependencies. The Forget Gate (left) defines which information to dismiss previous states. The Input Modulation Gate σ adds helpful information to the Cell State memory, and the Output Gate provides the output used in the next state $h_{t.}$

The behavior of a LSTM can be regarded as a set of networks that can store and reject information depending on the time and importance of the data **fig. 16**. The main difference between a conventional RNN and an LSTM is its capability to store information from long dependency periods (Cell state) (HOCHREITER; SCHMIDHUBER, 1997). To represent a sequence, the LSTM depends on a current state x_t that interacts with the previous state $h_{(t-1)}$ (Short memory) and historical states C_t (Long memory). This process is performed in three broadly steps:

1) Forget Gate: The section discards the information that is not useful in the Cell State (Memory). The process is performed using a NN with a sigmoid of the following form:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{1}$$

2) Input modulation Gate: The section that selects information to be added. The process is performed using two NN'*s*, the first one (2) decides which information will be added using a sigmoid form and (3) this information that must have to be added to the Cell State (Long term) using a hyperbolic tangent function.

$$i_t = \sigma(W_i . [h_{t-1, x_t}] + b_i) \tag{2}$$

$$C_t = \tanh(W_C . [h_{t-1}, x_t] + b_C) \tag{3}$$

3) Output Gate: Selects information used as support to the next state (Short Term). Using two NN's, the

first one is a sigmoid function (4) that decides which information will be used and a hyperbolic tangent function (5) that decides the intensity of this information used.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_0) \tag{4}$$

$$h_t = o_t * \tanh(C_t) \tag{5}$$

Our architecture of block 2 (BERT-LSTM) is the probability that a litigation process (document) belongs to a binary class *c (*lose or win). Each document is represented as a set of chunks (words < 512) trained on a BERT architecture and unified using an LSTM. BERT represents each chunk as an R^n vector. Therefore, the final result will be a matrix R^{mxn} for each document, where *m* is the number of chunks and *n* the vector size representation. By convention, BERT represents the *n* vector by a size of 768. The vectors of each chunk are merged using an LSTM that estimates a probability *p*(*d*|*c*), that document *d* belongs to class *c*.

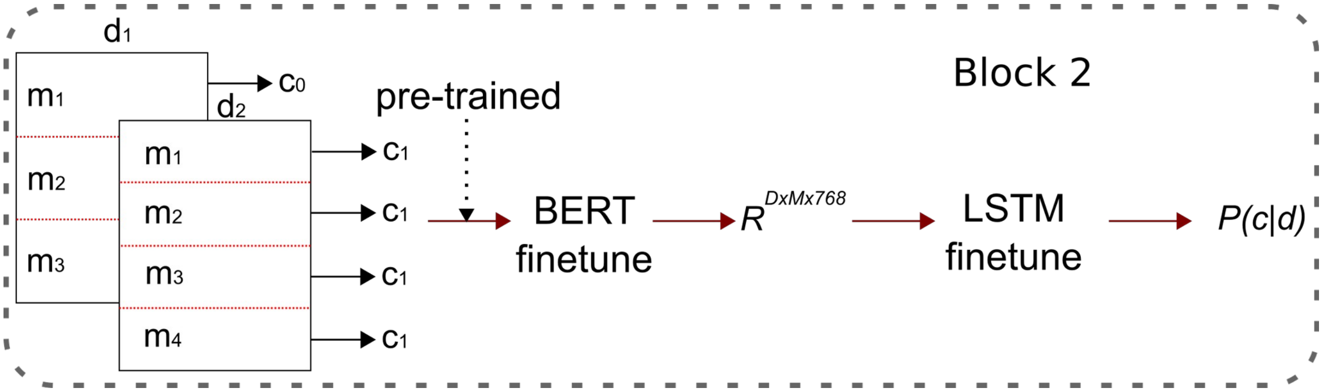


Figure 17. Framework block 2 representation. The process of block 2 that represents and estimates a probability from a lawsuit. In the example documents, d_1 and d_2 are cut into *m* chunks and fine-tuned by the BERT architecture using a pre-trained model as support. The result is an $R^{DxMx768}$ tensor representation that is merged using an LSTM, which estimates the probability and its corresponding class according to the maximum probability value.

5.3 Document similarity (block 3)

Argumentation is an essential tool used by lawyers to develop a court petition. Identifying components used in previously judged similar cases will provide support elements that can be used in favor of a new petition. We identified this particular feature from the interviews that we performed with experts of the area (directors of the organizations), as they agree that similar cases tend to have similar results and that a feature to identified similar lawsuits will be suitable for his work. In addition to the probability outcome, our framework provides in block 3 the most similar cases to the case provided as input, using as reference decisions from the training database. For example, regarding our previous example of a person asking for money back after buying a TV set, identifying similar judged cases of legal petitions from the consumer protection law area will provide the argumentation elements used as support, such as a particular law that led to favorable (win) or not favorable results (not accepted). With this information lawyers and users could create strategies such as reformulating a case before submitting to court. The code used to perform the operations from this block is provided in attachment 1e. **Fig 18** illustrates the process and possibilities of this block.

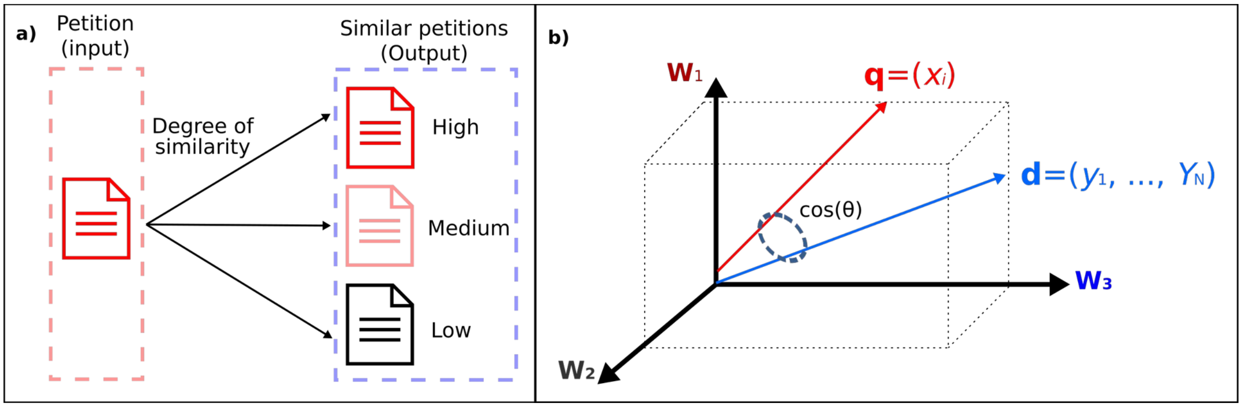


Figure 18. Framework block 3 illustration. (a) A petition is used as input and compared against the set of all petitions from the database. The result is a measure of similarity. A high similarity means that documents are almost identical, a medium means that the document has elements in common and a low that are different. **(b)** The degree of similarity is estimated using a normalized dot product of a vector space representation between a query (q) and a set of documents (d). A document x_i is query against a set of documents y_N from a database using a normalized dot product. The result is the angle $cos\theta$ for each of the y_N documents. For illustration purposes the example is in a 3-dimensional space, however, the technique is generalized for a R^n dimensional space.

To estimate the similarity from the set of documents, the block of the framework calculates a similarity index using a dot product of each of the documents represented as a tensor array. The results are ordered from the lowest to the highest value, where the lowest will represent the closest distance between two documents (the most similar). We chose this method (dot product) as it can be analytically solved, which provides optimization of computer performance. Formally, a lawsuit $\vec{q}$ to be query is compared to a set of documents $\vec{d}_j$ (previous judged cases) represented in a R^tvector space. Where t is the tensor dimension, represented by block 2 of our framework (BERT-LSTM), the result will be a set of j number of pairwise comparisons. In practice, the first value will be 0 because the document is compared against itself. For practical purposes, we defined that the framework to provide the 50 most similar documents, but this number can be adjusted.

$$\cos\left(d_j,q\right)=\frac{\vec{d}_j.\vec{q}}{\left|\vec{d}_j\right|.\left|\vec{q}\right|}=\frac{\sum_{i=i}^{t} W_{ij}.W_{iq}}{\sqrt{\sum_{i=i}^{t} W_{ij}^2.\sum_{i=i}^{t} W_{iq}^2}} \tag{4}$$

5.4 Baseline (Fast Text)

Besides our proposed architecture, we used as baseline the FasText model (JOULIN et al. 2016) to perform faster experiments as long as it has provided closer results to the state-of-art models but with better computer performance. It is important to highlight that litigation documents have distinctive features of being long texts, making the performance a critical feature, therefore we considered a baseline model to accelerate the experiments as a desirable element. FastText is a model based on the CBOW structure that works by estimating the probability of the presence of a word due to its context, according to a defined asymmetric window (MIKOLOV et al. 2017). Formally, given a sequence of T words, $w_1,\ldots,w_T$ the objective of the CBOW model is to maximize the log-likelihood of the probability of the words given their surroundings:

$$\sum_{t=1}^{T} \log p\left(w_t \vee C_t\right) \tag{5}$$

Where C_T is the context of the t-*th* word, e.g., the words $w_{t-c}, \ldots w_{t-1,} w_{t+1,}, \ldots, w_{t+c,}$for a context window of size $2c$. A natural candidate for the conditional probability in Eq. 5 is a softmax function. However, it is cited that it is impractical for large vocabulary (MIKOLOV et. Al, 2017). An alternative is to replace this probability with independent binary classifiers over words. More precisely, the conditional probability of a word *w* given its context c in Eq. 5 is replaced by the following quantity:

$$\log\left(1+e^{-s(w,C)}\right)+\sum_{n\epsilon N_c}\log\left(1+e^{s(w,C)}\right) \tag{6}$$

Where $s(w,C)$ is a scoring function between a word *w* and it is context *C*
N_c is a set of negative examples sampled from the vocabulary. The maximized CBOW objective function is obtained by replacing the log probability in Eq (5) by the quantity defined in Eq (6):

$$\sum_{t=1}^{T}\left[\log\left(1+e^{-s(w_{t,}C_t)}\right)+\sum_{n\epsilon N_{Ct}}\log\left(1+e^{s(n,C_t)}\right)\right] \tag{7}$$

A parameterization for this model is to represent each word w by a vector v_w. The context is represented by the average of the word vectors $\upsilon_{w'}$ of each word w' in its window. The scoring function is simply the dot product between these two quantities:

$$s(w,C)=\frac{1}{C}\sum_{w'\in C}u_{w'}^{T}V_{w.} \tag{8}$$

6. EXPERIMENTS

6.1 ECHR dataset

The first step to test our framework was to estimate its performance in a database with reported benchmarks. We used the ECHR database that analyses human rights violations (ALETRAS et al., 2016). We chose this database as it has the highest reported accurate results on the LJP problem in English, and it is available as open-source (CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019). It is essential to highlight that the language is a fundamental factor to consider in the NLP area as most of the state-of-art literature is based on English corpus (documents), and the pre-trained models are primarily published to be used in English texts. The ECHR describes judicial proceedings related to violations of political or civil rights. The text below illustrates a sample of the ECHR dataset. An applicant (Mr. Murat Arslan) demanded that his rights were violated as long as he was taken to the headquarters of the anti-terrorism security police. Then the case was judged as non-violated.

> *"The applicant, Mr Murat Arslan, is a Turkish national who was born in 1979 and is currently detained in Nazilli Prison (Turkey). He was represented before the Court by Mr E. Yildiz, a lawyer practicing in Izmir., On 9 October 2001, the applicant was arrested and taken into police custody at the headquarters of the anti-terrorism branch of the Izmir security police., On 12 October 2001, after being interviewed by the public prosecutor at the Izmir National Security Court, he was taken before a judge of that court who on 13 October 2001 ordered his detention pending trial., On 19 October 2001 the public prosecutor committed the applicant for trial in the National Security Court., The criminal proceedings against the applicant are still pending., The applicant's lawyer dated his application 12 April 2002 and took it on 19 April to the post office in Konak (central Izmir), where post is collected regularly several times a day."* Judged = 0 (non-violated).

The ECHR dataset contains 11748 cases distributed in 5263 (non-violated) and 6485 (violated). We used the division for training (90%) and validation (10%) provided by the authors. We also tagged the decisions as 0 when the cases were judged as no human rights violation and 1 when cases were judged as human rights violations. The data were provided as a set of JSON format files that we transformed into a matrix array form. The first algorithm we tested was our baseline (Fast Text).

6.1.1 Baseline (Fast Text)

We performed a pre-process of the ECHR dataset by transforming it into lower case and removing non-alphanumeric characters. **Tables 8** and **9** show our experiments' results in decreasing order according to their macro-F1 value. We executed tests with different learning rates and epoch values. We divided the results for each class into (label = 0) for non-violated and (label =1) for violated. We make this distinction as we wanted to validate in which class the framework performed the best. **Table 8** refers to the performance of the algorithm for the class non-violated (label 0), with the highest values (precision = 0.614), (recall = 0.706), (f-score = 0.657) and a (macro-F1=0.729).

Table 8. Results of the ECHR database for cases judged as not human rights violated (Label 0 - Baseline).

Precision	Recall	F1	Macro-F1	Epochs	LR
0.614	**0.706**	**0.657**	0.729	100	0.1
0.614	0.702	0.655	0.728	50	0.1
0.614	0.698	0.654	0.727	200	0.1
0.607	0.697	0.649	0.723	1000	0.1
0.608	0.693	0.648	0.722	500	0.1
0.612	0.677	0.643	0.721	40	0.1
0.613	0.657	0.634	0.717	30	0.1
0.608	0.657	0.632	0.714	5	1.9

In the same line, **Table 9** refers to the performance for the class accepted petitions (label 1). The highest values were (precision = 0.835), (recall = 0.780), (f-score = 0.801) and a (macro-F1 = 0.729). Using our baseline algorithm (FastText), we identified that the class violated human rights (label 1) has a better performance than the class non-violated (label 0) and the overall performance of the algorithm provides a macro-F1 value of 0.729. It was also important to note that the best accuracy performance was attended with 100 epochs (macro-F1 = 0.729), but approximated results were reached using 50 epochs (macro-F1 = 0.728) which let us conclude that the performance does not have a linear dependency.

Table 9. Results of the ECHR database for cases judged as human rights violated (Label 1 - Baseline).

Precision	Recall	F1	Macro-F1	Epochs	LR
0.835	0.770	**0.801**	0.729	100	0.1
0.833	0.771	0.801	0.728	50	0.1
0.832	0.773	0.801	0.727	200	0.1
0.830	0.766	0.797	0.723	1000	0.1
0.828	0.768	0.797	0.722	500	0.1
0.823	0.778	0.799	0.721	40	0.1
0.815	**0.785**	0.800	0.717	30	0.1
0.814	0.780	0.797	0.714	5	1.9

6.1.2 Proposed Framework

Our second algorithm to test was our proposed framework (BERT-LSTM). A limitation of BERT architectures is the requirement for high computational resources as it contains about 110 million parameters so it is suggested to train the models using a Graphic Process Unit (GPU). To train our framework, we use a Nvida Titan XP GPU (we can use this resource due to Nvidia's grant contribution to our project) using the library Pytorch during 123 epochs. In contrast to conventional text ML algorithms such as FastText BERT- based algorithms do not need pre-processed text (transform to lowercase, remove accents, etc.) as input. Therefore, we did not perform this pre-process. As we already cited in our work, the BERT model requires a pre-trained model. Therefore we used the BERT-base uncased pre-trained model in English, which is a widely used model with the suggested parameters (batch = 10, learning rate=6e-5) and tested with different epoch values (DEVLIN et al., 2018). The time to process was ~30 mins/epoch. We also analyzed the results separately as our baseline model for each non-accepted class (label 0), accepted class (label 1) and arranging with an increased order according to its macro-F1 value. **Table 10** shows the results of non-accepted petitions, with the highest values (precision = 0.979), (recall = 0.807) and (macro-F1 = 0.891). The number of epochs that provided the best results was 10. Therefore the total processing time was of 4 hrs (30min/epoch).

Table 10. Results of the ECHR database for cases judged as not human rights violated (Label 0 - Proposed Framework).

Precision	Recall	F1	Macro-F1	Epochs
0.979	0.797	**0.879**	**0.891**	10
0.996	0.780	0.875	0.888	11
0.979	0.796	0.878	0.887	8
0.977	0.791	0.874	0.886	13
0.977	0.791	0.874	0.886	14
0.980	0.780	0.868	0.883	7
0.939	**0.807**	0.868	0.879	9
0.956	0.794	0.868	0.879	6

In the same line, **Table-11** shows the results for the accepted petitions category (label 1), with the highest values of (precision = 0.832), (recall=0.997), and a (macro-F1=0.891).

Table 11. Results of the ECHR database for cases judged as human rights violated (Label 1 - Proposed Framework).

Precision	Recall	F1	Macro-F1	Epochs
0.829	0.986	**0.901**	0.891	10
0.819	**0.997**	0.899	0.888	11
0.823	0.983	0.896	0.887	8
0.825	0.981	0.896	0.886	13
0.825	0.981	0.896	0.886	14
0.821	0.983	0.894	0.883	7
0.832	0.955	0.889	0.879	9
0.824	0.964	0.888	0.879	6

We compared our results to the reported on literature for the ECHR dataset. **Table 12** details the macro-F1 values for each of the cited methodologies. These values were reported as a macro average for both labels that we estimated to create a direct comparative measurement. These results exceeded our expectations, as shown in **Table 12,** the most accurate values in all the measurements to the best of our knowledge are from our proposed framework (BERT-LSTM) As it can be seen the highest reported values are (HIER-BERT) (CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019) with a precision (0.906 vs. 0.904), recall (0.876 vs. 0.793), and macro-F1 (0.884 vs. 0.884). Our baseline algorithm (FastText) showed the lowest accurate results (precision = 0.75), (recall = 0.738) and (f1 = 0.729), just above the BOW-SVM (precision = 0.715), (recall = 0.720) and (f1 = 0.718) that is the only algorithm that does not belong to the category of DL methodologies. The BIGRU-ATT and HAN are DL models that depend on attention mechanisms and provide similar results to the HIER-BERT (macro-F1 ~0.80). Finally, the results were also compared to randomly COIN-TOSS p (0.5) values, which provided precision and recall (~0.50) as the dataset comprises equal sample sizes of binary categories. Finally, using a BERT single model (precision=0.240) demonstrates that using a strategy without chunks provide weak results.

Table 12. Macro result values for the ECHR dataset. Our proposed framework BERT-LSTM shows the best performance in precision, recall, and F1 metrics against the highest reported results in the literature. The best results overall are based on BERT architectures that utilize chunk strategies. BERT single model estimated the weakest result from the sample. HAN and BIGRU are based on attention mechanisms that provide close results to BERT. Our baseline model FastText performed better than conventional ML techniques (BOW-SVM).

Author(s)	**Precision**	**Recall**	**F1**
* BERT-LSTM (our work)	**0.906**	**0.876**	**0.884**
HIER-BERT (CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019)	0.904	0.793	0.820
BERT	0.240	0.500	0.170
HAN (CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019)	0.882	0.780	0.805
BIGRU-ATT (CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019)	0.871	0.772	0.795
FAST-TEXT* (baseline)	**0.725**	0.738	0.729
BOW-SVM (ALETRAS et al., 2016; CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019)	0.715	0.720	0.718
COIN-TOSS	0.504	0.505	0.397

6.2 TRT4 dataset

After testing the performance of our framework in a public dataset with reported benchmarks, we evaluated it in a Brazilian custom dataset. The Brazilian court system is divided into first, second and third instance. We chose litigation processes from the first instance, because we defined as criteria to use judicial sentences without any previous appeal (second and third instance). We used data provided directly from court data-centers because the public available information on sites from Brazilian courts does not contain the complete text petitions, only the processes summary. To have access for the full litigation documents we searched for possibilities on some of the Brazilian courts. We look up on the federal (TRF4) and state (TJRS) courts with unsuccessful results. After searching this two possibilities we performed an agreement with the Brazilian labor court, “Tribunal Regional do Trabalho 4 região” (TRT4) that demanded multiple meetings and agreements but finally collaborated with the information for our study. The TRT4 dataset exhibits differences against the public ECRH dataset. The size of the TRT4 database was composed 100,000 litigation processes provided as a set of raw PDF files structured into two files (petition and sentence) in Brazilian Portuguese language. We pre-processed the dataset using block one from our framework and discarded the documents that did not satisfied the established criteria   (e.g., did not have a verdict defined by any of the sample of words that we established as basis). The final size of our sample comprised 58169 lawsuits, divided into 34265 as “not accepted” and 23904 as “accepted” as plotted in **fig. 19**.

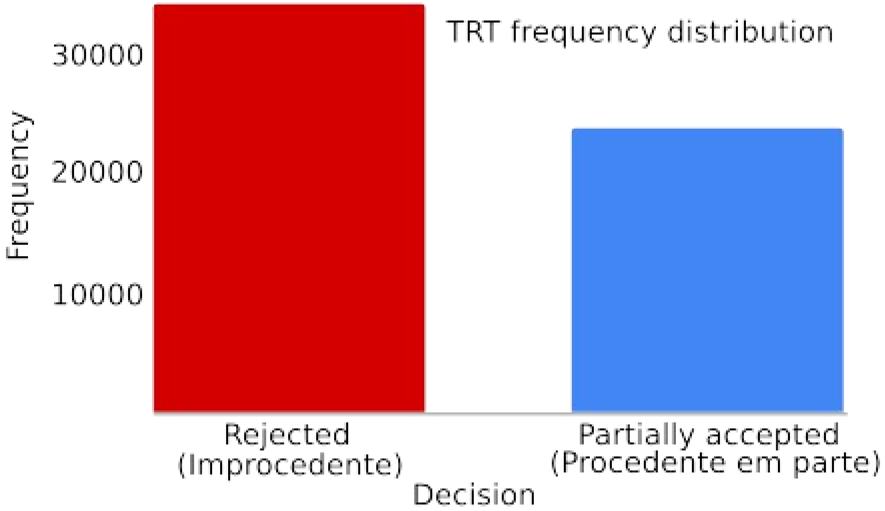


Figure 19. Distribution of lawsuits according to its final decision from TRT4. “Improcedente” refers to petitions that lose and “Procedente em parte” to petitions that win.

6.2.1 Baseline (FastText)

We first performed the experiments using the baseline algorithm in a computer with a conventional CPU processor (4 cores). Each epoch delayed ~0.72 mins. Therefore, the total time for 25 epochs took 18 min. We used the suggested parameters of learning rate ranges (0.1 – 1) and trained till we perceive that the model attend a maximum accuracy (no. epochs = 25). **Table 13** shows the results of the class 0 (non-accepted) petition. The highest value for the precision was 0.675 (epochs = 20 & lr = 1), recall of 0.854 (epochs = 5 & lr = 0.1), F1 of 0.726 (epochs = 5 & lr =0.8) and macro-F1 of 0.613 (epochs = 15 & lr =1).

Table 13. Results of the TRT4 database for cases judged as non-accepted (Label 0 - Baseline)

Precision	Recall	F1	Macro-F1	Epochs	LR
0.673	0.744	0.707	**0.613**	15	1.0
0.668	0.782	0.721	0.612	10	1.0
0.675	0.718	0.696	0.611	20	1.0
0.674	0.716	0.694	0.610	25	1.0
0.656	0.806	0.723	0.597	5	0.9
0.654	0.809	0.723	0.594	5	0.5
0.653	0.816	**0.726**	0.593	5	0.8
0.617	**0.854**	0.716	0.522	5	0.1

Similarly, **Table 14** shows statistic values for the accepted petitions (class 1). The highest precision value was 0.589 (epochs = 5 & lr = 0.8), recall 0.503 (epochs = 20 & lr = 1.0), F1 0.527 (epochs = 20 & lr = 1.0) and macro-F1 0.522 (epochs = 15 & lr = 1.0) . Overall, the accuracy was lower than the class 0.

Table 14. Results of the TRT4 database for cases judged as accepted (Label 1 - Baseline).

Precision	Recall	F1	Macro-F1	Epochs	LR
0.566	0.479	0.519	**0.613**	15	1.0
0.554	**0.503**	**0.527**	0.611	20	1.0
0.551	0.502	0.525	0.610	25	1.0

0.573	0.427	0.489	0.602	4	1.0
0.585	0.394	0.470	0.597	5	0.9
0.584	0.386	0.464	0.594	5	0.5
0.589	0.377	0.460	0.593	5	0.8
0.531	0.237	0.328	0.522	5	0.1

In addition to the previous estimates, we trained the model using first a pre-processing text by converting to lowercase and removing non-alphanumeric characters and accents. We used the same parameter suggestions of learning rate ranges (0.1-1) and epochs (1-25). **Table 15** shows the results for class 0 (non-accepted). Overall, the accuracy increased against the non-processed text values, precision from 0.675 to 0.735, recall from 0.502 to 0.852, f1 from 0.527 to 0.752, and the macro-F1 from 0.613 to 0.684.

Table 15 Results of the TRT4 database for cases judged as accepted with pre-processing (Label 0 - Baseline).

Precision	Recall	F1	Macro-F1	Epochs	LR
0.734	0.762	0.748	0.648	10	1.0
0.729	0.771	0.749	**0.682**	25	1.0
0.729	0.767	0.747	0.680	15	1.0
0.735	0.742	0.738	0.679	20	1.0
0.719	0.782	0.749	0.674	7	1.0
0.696	**0.817**	**0.752**	0.654	5	0.8
0.696	0.809	0.748	0.653	4	1.2
0.631	0.859	0.725	0.552	5	0.1

Table 16 also shows the results for the pre-processed text for class 1 (accepted). As well as class 0, results were superior to the non-processed texts. The results goes on precision from 0.589 to 0.650, recall from 0.502 to 0.615, f1 from 0.613 to 0.620, and a macro-F1 from 0.613 to 0.684.

Table 16. Results of the TRT4 database for cases judged as non-accepted with pre-processing (Label 1 - Baseline).

Precision	Recall	F1	Macro-F1	Epochs	LR
0.638	0.603	**0.620**	**0.684**	10	1.0
0.641	0.589	0.614	0.682	25	1.0

0.638	0.590	0.613	0.680	15	1.0
0.624	**0.615**	0.620	0.679	20	1.0
0.642	0.560	0.598	0.674	7	1.0
0.650	0.487	0.557	0.654	5	0.8
0.643	0.494	0.558	0.653	4	1.0
0.570	0.289	0.383	0.553	5	0.1

We concluded that pre-processing a text using a conventional ML text algorithm as the FastText increased accuracy. That class 0 (non-accepted) performed better than class 1 (accepted), which coincides with the results ECHR dataset.

6.2.2 Proposed framework

After performing the first tests using our baseline algorithm, we run the experiments on our proposed framework (BERT-LSTM). We processed the framework using a GPU. But in contrast to the experiments performed on the ECHR dataset, our first tests failed due to the high demand for RAM resources as long as the length of the texts was substantially longer than those from the ECHR. To overcome this limitation, we create a swap space of 40 GB additional to the 10 GB of memory of the machine. Each epoch demanded ~2.7 hours and trained until we observe that accuracy does not have a better performance (123 epochs) . Therefore, the total processing time was 14 days. This high demand of processing time also demanded to create mechanisms to store partial results. **Table 17** shows the accuracy measures of the framework for the class non-accepted (label 0). The values are ordered in decreasing order according to the macro-F1. The highest precision value (0.741) was obtained on the 13 epochs. While the highest recall value (0.723) was obtained in the last epoch (123), which implies the framework tried to obtain better results for label 0 in the first training stages, but then the model compensated the results for label 1 for increasing the overall accuracy.

Table 17. Results of the TRT4 database for cases judged as accepted (Label 0 – Proposed Framework).

Precision	Recall	F1	Macro-F1	Epochs
0.724	**0.718**	**0.721**	0.677	47
0.731	0.690	0.710	0.672	68
0.716	0.719	0.717	0.670	79

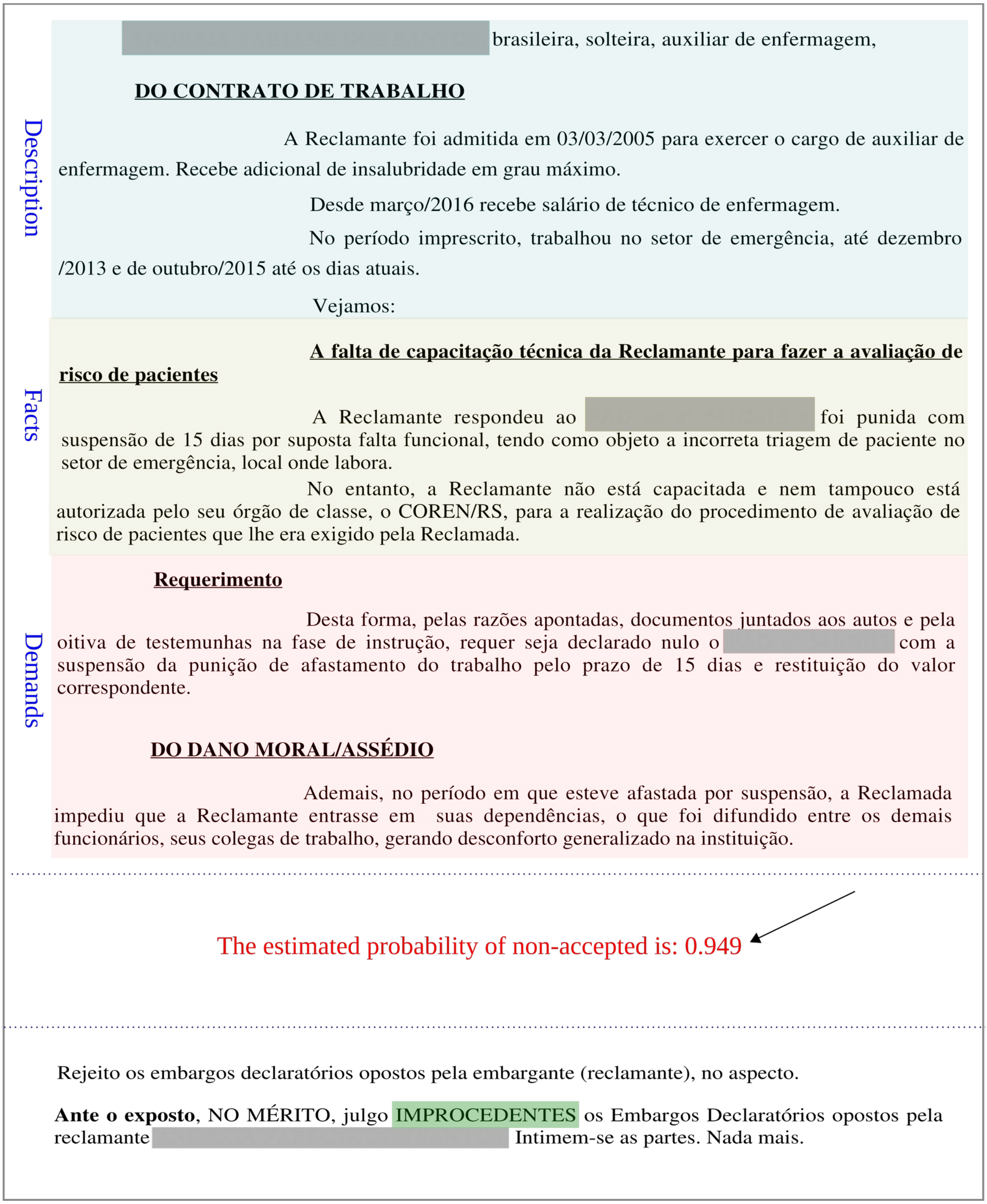
0.738	0.668	0.701	0.669	92
0.723	0.694	0.708	0.668	113
0.706	0.723	0.715	0.665	123
0.741	0.604	0.665	0.646	13
0.728	0.504	0.596	0.602	9

Table 18 complements the results for label 1 (accepted petitions). Overall, results were less accurate than with label 0. We obtained the highest precision value (0.614) on epoch 47, which coincides with the highest macro-F1 value (0.677). The highest recall value (0.742) in the initial steps (epoch 13), similar to the F1 value (0.627) obtained at epoch (13).

Table 18. Results of the TRT4 database for cases judged as non-accepted (Label 1 – Proposed Framework).

Precision	Recall	F1	Macro-F1	Epochs
0.614	0.618	0.616	**0.677**	47
0.600	**0.647**	**0.623**	0.672	68
0.607	0.604	0.605	0.670	79
0.592	0.671	0.629	0.669	92
0.598	0.632	0.615	0.668	113
0.604	0.585	0.594	0.665	123
0.563	0.707	0.627	0.646	13
0.517	0.742	0.609	0.602	9

Finally, we illustrate the examples of 2 lawsuits processed by our proposed framework. **Figure 20** shows a lawsuit that the framework classified as non-accepted and **figure 21** as accepted. For space purposes, we only show some parts of both lawsuits to exemplify the estimation of the model. **Figure 20** refers to the case of a technician nurse (description) who was punished (facts) for performing an incorrect triage of a patient. The petitioner (nurse) is demanding the suspension of the punishment. The framework estimated the class 0 (non-accepted) with a probability of 0.949 that agrees with the historical decision of non-accepted.


Model Input
Description
Facts
Demands
Model Output
True Decision

[redacted] brasileira, solteira, auxiliar de enfermagem,

DO CONTRATO DE TRABALHO

A Reclamante foi admitida em 03/03/2005 para exercer o cargo de auxiliar de enfermagem. Recebe adicional de insalubridade em grau máximo.

Desde março/2016 recebe salário de técnico de enfermagem.

No período imprescrito, trabalhou no setor de emergência, até dezembro /2013 e de outubro/2015 até os dias atuais.

Vejamos:

A falta de capacitação técnica da Reclamante para fazer a avaliação de risco de pacientes

A Reclamante respondeu ao [redacted] foi punida com suspensão de 15 dias por suposta falta funcional, tendo como objeto a incorreta triagem de paciente no setor de emergência, local onde labora.

No entanto, a Reclamante não está capacitada e nem tampouco está autorizada pelo seu órgão de classe, o COREN/RS, para a realização do procedimento de avaliação de risco de pacientes que lhe era exigido pela Reclamada.

Requerimento

Desta forma, pelas razões apontadas, documentos juntados aos autos e pela oitiva de testemunhas na fase de instrução, requer seja declarado nulo o [redacted] com a suspensão da punição de afastamento do trabalho pelo prazo de 15 dias e restituição do valor correspondente.

DO DANO MORAL/ASSÉDIO

Ademais, no período em que esteve afastada por suspensão, a Reclamada impediu que a Reclamante entrasse em suas dependências, o que foi difundido entre os demais funcionários, seus colegas de trabalho, gerando desconforto generalizado na instituição.

The estimated probability of non-accepted is: 0.949

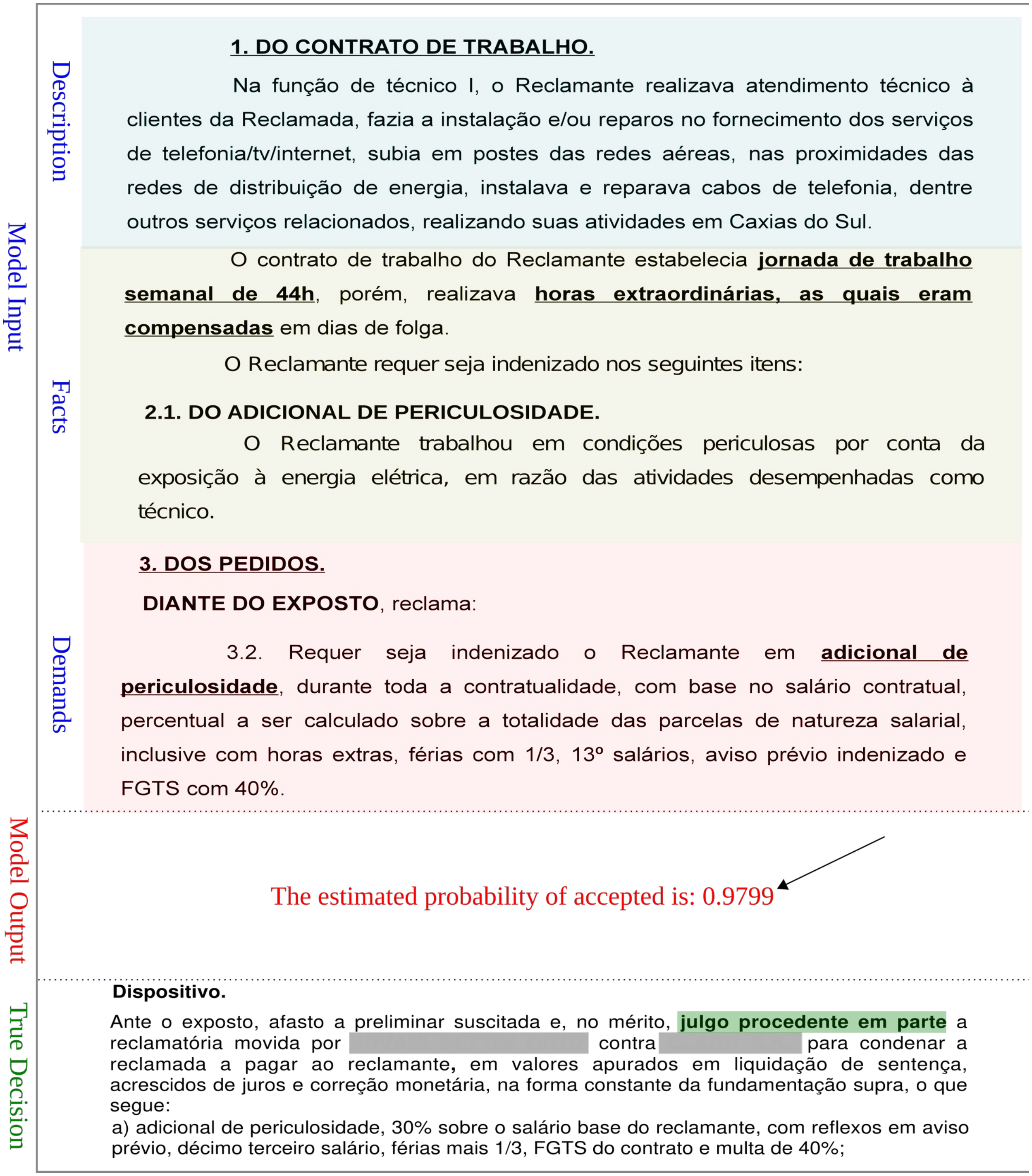
Rejeito os embargos declaratórios opostos pela embargante (reclamante), no aspecto.

Ante o exposto, NO MÉRITO, julgo IMPROCEDENTES os Embargos Declaratórios opostos pela reclamante [redacted] Intimem-se as partes. Nada mais.

Figure 20. Example of a lawsuit estimated as non-accepted (lose). The document illustrates the input and output result from our proposed framework of a sample from the TRT4 database that refers to a nurse labor case. The upper sections refer to the input (initial petition) written as free text and include three sections: description, facts, and demands. The middle section shows the estimated (not accepted) class and its corresponding probability (0.9799) from the framework. Then, the lower

section shows the real decision from the judge who decided as non-accepted the claims from the petitioner defined in the word "IMPROCEDENTES," and agrees with the estimated result from our framework.

Figure 21 describes a case that the framework estimated as accepted. The petition refers to a telephone technician (description) who claims that some of his labor rights were not respected (facts). He is asking for a compensation (demands). The output of the framework estimated the petition as accepted with a corresponding probability of 0.9799. The right decision agrees with the framework result as accepted described in the phrase "procedure em part."


Model Input
Description
Facts
Demands
Model Output
True Decision

1. DO CONTRATO DE TRABALHO.

Na função de técnico I, o Reclamante realizava atendimento técnico à clientes da Reclamada, fazia a instalação e/ou reparos no fornecimento dos serviços de telefonia/tv/internet, subia em postes das redes aéreas, nas proximidades das redes de distribuição de energia, instalava e reparava cabos de telefonia, dentre outros serviços relacionados, realizando suas atividades em Caxias do Sul.

O contrato de trabalho do Reclamante estabelecia **jornada de trabalho semanal de 44h**, porém, realizava **horas extraordinárias, as quais eram compensadas** em dias de folga.

O Reclamante requer seja indenizado nos seguintes itens:

2.1. DO ADICIONAL DE PERICULOSIDADE.

O Reclamante trabalhou em condições periculosas por conta da exposição à energia elétrica, em razão das atividades desempenhadas como técnico.

3. DOS PEDIDOS.

DIANTE DO EXPOSTO, reclama:

3.2. Requer seja indenizado o Reclamante em **adicional de periculosidade**, durante toda a contratualidade, com base no salário contratual, percentual a ser calculado sobre a totalidade das parcelas de natureza salarial, inclusive com horas extras, férias com 1/3, 13º salários, aviso prévio indenizado e FGTS com 40%.

The estimated probability of accepted is: 0.9799

Dispositivo.

Ante o exposto, afasto a preliminar suscitada e, no mérito, **julgo procedente em parte** a reclamatória movida por [redacted] contra [redacted] para condenar a reclamada a pagar ao reclamante**,** em valores apurados em liquidação de sentença, acrescidos de juros e correção monetária, na forma constante da fundamentação supra, o que segue:

a) adicional de periculosidade, 30% sobre o salário base do reclamante, com reflexos em aviso prévio, décimo terceiro salário, férias mais 1/3, FGTS do contrato e multa de 40%;

Figure 21. Example of a lawsuit estimated as accepted (win). The document illustrates the input and estimated result from our framework that corresponds to a labor case from a telephone technician (description), that performed dangerous activities (facts), and claimed compensation due to this fact (demand). The upper sections show the input (initial petition) written in free text, which includes the description, facts and demands. The following section shows the estimated class (accepted) with its corresponding probability. The last section shows the real decision (accepted), defined in the phrase "procedente em parte".

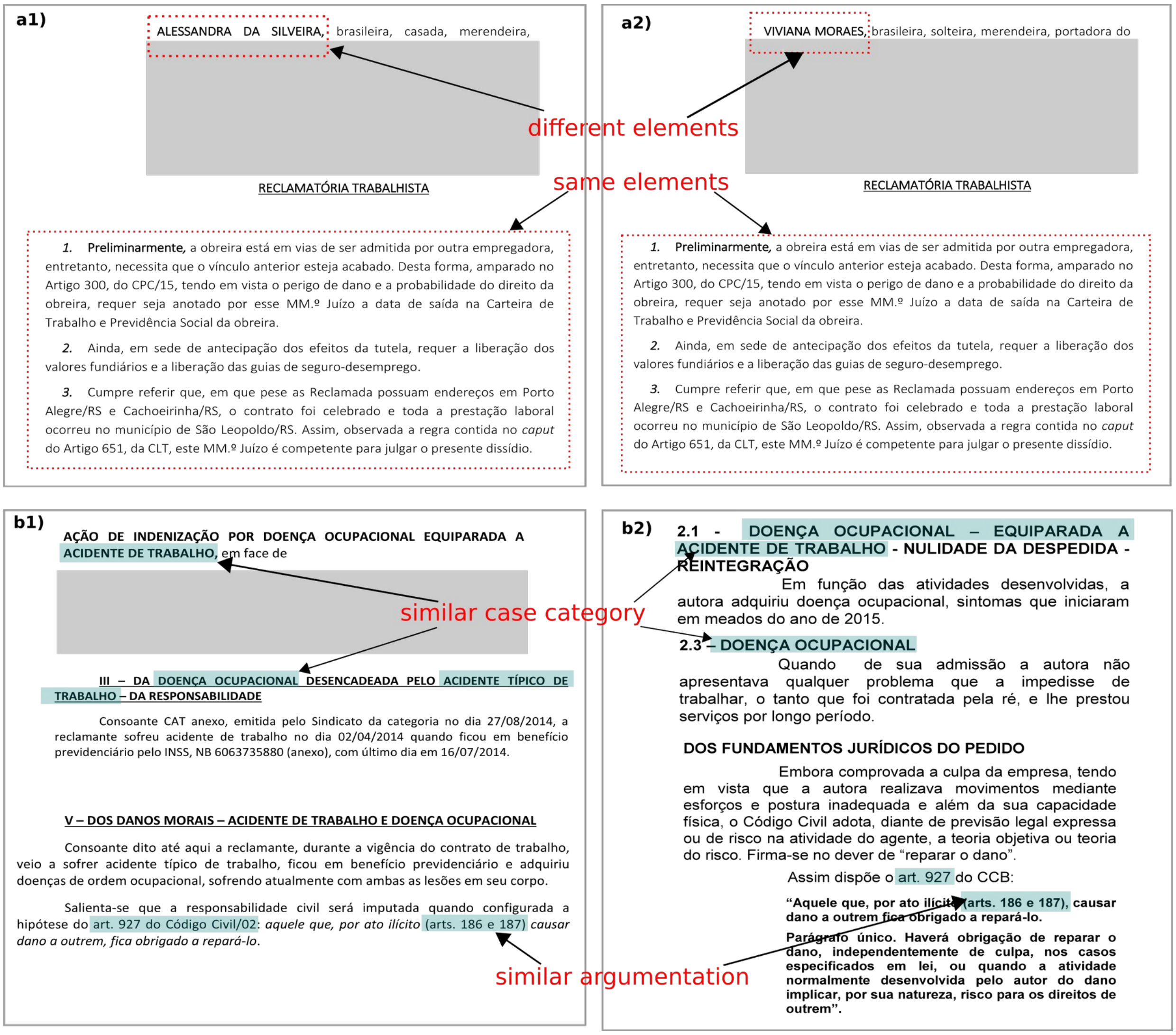
6.2.4 Document similarity

Using the last block of our framework, we estimated similar cases to the document provided as input. This process is crucial, as it was referred to in our text. It provides clue elements that were used by similar previous litigation processes. **Table 19** exposes the example of a particular litigation case of the TRT4 dataset that was estimated from our experiments. The column “query id” (left) represents the index of the lawsuit. The column “distance” (center) represents the degree of similarity from the query id index against all the sets of documents from the database. The column “retrieved id” (right) represents the id of the similar retrieved documents. The rows are in increased order according to the “distance” (similarity column). For practical purposes our framework is limited to provide the 50 more similar documents. The first line represents the most similar document, the distance is 0 because the retrieved lawsuit is precisely the same. The following line is the second more similar lawsuit and subsequently to the 50th most similar.

Table 19. Example of similarity estimation. The left column identifies a document of a particular lawsuit compared against the rest of the documents from a database. The center column estimates how similar are the query id document against the rest of the documents. A distance 0 means exactly the same document. For practical purposes, it is limited to provide the 50 most similar documents. In this example the retrieved id 20518 is the 1stsimilar and the 26512 id the 50th more similar to the query id document (7553).

Query id	**Distance (similarity)**	**Retrieved id**
7553	0.00000000000	7553 (same)
	0.00044500828	20518 (1st)
	0.0004966259	14711 (2nd)
	...	...
	0.0009752512	26512 (50th)

To provide a better understanding of the process, in **fig 22,** we illustrate two examples of similar cases estimated by our framework from the TRT4 database. In both examples, we use anonymous information of the involved people. The first pair, **fig 22 a1** and **fig 22 a2,** shows a section of two lawsuits with a lower index value (very similar). It can be seen that both lawsuits have exactly the same elements in the argumentation section (the three paragraphs: *Pleliminarmente*, *ainda, cumpre*). They only differ in the petitioner's name. In **fig 22 a1** *Alessandra da Silveira* in **fig 22 a2** *Viviana Moraes***.** **Fig 22 b1** and **fig 22 b2** compare two cases with a medium similarity. In contrast to the previous example, both lawsuits have elements in common but are not entirely the same. The categories of the case (“*Accidente de trabalho*” & “*Doença ocupacional*”) and the argumentation codes (*997, 186* and *187)* are the same but not the rest of the lawsuit.


a1)
a2)
b1)
b2)
different elements
same elements
similar case category
similar argumentation

ALESSANDRA DA SILVEIRA, brasileira, casada, merendeira,

[redacted]

RECLAMATÓRIA TRABALHISTA

1. **Preliminarmente,** a obreira está em vias de ser admitida por outra empregadora, entretanto, necessita que o vínculo anterior esteja acabado. Desta forma, amparado no Artigo 300, do CPC/15, tendo em vista o perigo de dano e a probabilidade do direito da obreira, requer seja anotado por esse MM.º Juízo a data de saída na Carteira de Trabalho e Previdência Social da obreira.

2. Ainda, em sede de antecipação dos efeitos da tutela, requer a liberação dos valores fundiários e a liberação das guias de seguro-desemprego.

3. Cumpre referir que, em que pese as Reclamada possuam endereços em Porto Alegre/RS e Cachoeirinha/RS, o contrato foi celebrado e toda a prestação laboral ocorreu no município de São Leopoldo/RS. Assim, observada a regra contida no *caput* do Artigo 651, da CLT, este MM.º Juízo é competente para julgar o presente dissídio.

VIVIANA MORAES, brasileira, solteira, merendeira, portadora do

[redacted]

RECLAMATÓRIA TRABALHISTA

1. **Preliminarmente,** a obreira está em vias de ser admitida por outra empregadora, entretanto, necessita que o vínculo anterior esteja acabado. Desta forma, amparado no Artigo 300, do CPC/15, tendo em vista o perigo de dano e a probabilidade do direito da obreira, requer seja anotado por esse MM.º Juízo a data de saída na Carteira de Trabalho e Previdência Social da obreira.

2. Ainda, em sede de antecipação dos efeitos da tutela, requer a liberação dos valores fundiários e a liberação das guias de seguro-desemprego.

3. Cumpre referir que, em que pese as Reclamada possuam endereços em Porto Alegre/RS e Cachoeirinha/RS, o contrato foi celebrado e toda a prestação laboral ocorreu no município de São Leopoldo/RS. Assim, observada a regra contida no *caput* do Artigo 651, da CLT, este MM.º Juízo é competente para julgar o presente dissídio.

AÇÃO DE INDENIZAÇÃO POR DOENÇA OCUPACIONAL EQUIPARADA A ACIDENTE DE TRABALHO, em face de

[redacted]

III – DA DOENÇA OCUPACIONAL DESENCADEADA PELO ACIDENTE TÍPICO DE TRABALHO – DA RESPONSABILIDADE

Consoante CAT anexo, emitida pelo Sindicato da categoria no dia 27/08/2014, a reclamante sofreu acidente de trabalho no dia 02/04/2014 quando ficou em benefício previdenciário pelo INSS, NB 6063735880 (anexo), com último dia em 16/07/2014.

V – DOS DANOS MORAIS – ACIDENTE DE TRABALHO E DOENÇA OCUPACIONAL

Consoante dito até aqui a reclamante, durante a vigência do contrato de trabalho, veio a sofrer acidente típico de trabalho, ficou em benefício previdenciário e adquiriu doenças de ordem ocupacional, sofrendo atualmente com ambas as lesões em seu corpo.

Salienta-se que a responsabilidade civil será imputada quando configurada a hipótese do art. 927 do Código Civil/02: *aquele que, por ato ilícito* (arts. 186 e 187) *causar dano a outrem, fica obrigado a repará-lo*.

2.1 - DOENÇA OCUPACIONAL – EQUIPARADA A ACIDENTE DE TRABALHO - NULIDADE DA DESPEDIDA - REINTEGRAÇÃO

Em função das atividades desenvolvidas, a autora adquiriu doença ocupacional, sintomas que iniciaram em meados do ano de 2015.

2.3 – DOENÇA OCUPACIONAL

Quando de sua admissão a autora não apresentava qualquer problema que a impedisse de trabalhar, o tanto que foi contratada pela ré, e lhe prestou serviços por longo período.

DOS FUNDAMENTOS JURÍDICOS DO PEDIDO

Embora comprovada a culpa da empresa, tendo em vista que a autora realizava movimentos mediante esforços e postura inadequada e além da sua capacidade física, o Código Civil adota, diante de previsão legal expressa ou de risco na atividade do agente, a teoria objetiva ou teoria do risco. Firma-se no dever de “reparar o dano”.

Assim dispõe o art. 927 do CCB:

“Aquele que, por ato ilícito (arts. 186 e 187), causar dano a outrem fica obrigado a repará-lo.

Parágrafo único. Haverá obrigação de reparar o dano, independentemente de culpa, nos casos especificados em lei, ou quando a atividade normalmente desenvolvida pelo autor do dano implicar, por sua natureza, risco para os direitos de outrem”.

Figure 22. Comparison of similar lawsuits a1.2) a pair of lawsuits with a low index value (high similarity) that are identical. They only differ in the name of the petitioners. **b1.2)** a pair of lawsuits with a medium index value. They both corresponded to the same case category (*DOENÇA OCUPACIONAL*) and used the same argumentation elements *(art 927 CCB, art 186-187).* In both

cases, information about the people involved in the cases was unidentified (gray region).

7. CONCLUSION AND FUTURE RESEARCH

We proposed a framework to estimate the probability of loss of liabilities subject to a litigation process (Contingent Liabilities) that we represented as solving the LJP problem. We identified from literature review and primary sources (interviews) the lack of existence of a framework like the one we proposed. Using a literature review, we identified that DL is the methodology that has shown better performance on the NLP area, and its use is exponentially increasing among the academic community. We developed a framework based on a DL architecture and tested it in two lawsuit databases: ECHR an international database with reported benchmarks, and TRT4 a Brazilian litigation database composed of ~ 100,000 lawsuits from a regional state labor court. Our tests provided to our knowledge the highest estimated reported accuracy on the ECHR collection compared to published results with a precision of 0.906 (CHALKIDIS; ANDROUTSOPOULOS; ALETRAS, 2019). The TRT4 as far as we know is the first work to estimate the probability outcome from a Brazilian labor court litigation database, using a mathematical model (LJP problem).

Despite using the same framework in both databases (ECHR & TRT4), the estimated outcomes provided different accuracies allowing us to identify important points to be discussed. The language is a fundamental aspect to be considered when using a DL framework that depends on pre-trained models such as the one that we used (BERT) because they are mainly published to be used in problems that involve English language texts. Regardless of using a pre-trained model in Brazilian Portuguese language (RODRIGUES et al., 2020), English pre-trained models are provided with high quality since they are trained on more extensive databases (DEVLIN et al., 2018) and offer broader possibilities, for example, BioBERT is a model pre-trained in medical and biological specialized literature texts (LEE et al., 2020). Nonetheless, the most important fact of using pre-trained models in English is that the state-of-art NLP literature is published and validated in the English language, which motivates the use and increases the quality of the models.

The structure of both databases (ECHR & TRT4) also exhibited substantial differences. The ECHR was provided in a structure form ready to be trained. It was already used in previous works (ALETRAS et al., 2016; CHALKIDIS; KAMPAS, 2018), which offers historical validations. On the other side, the TRT4 database was not structured. It was provided as a set of PDF files that we need to transformed into text and organized into an array form. We believe that during this process, some potential noise could be added to the texts. The no. of lawsuits (n=7100) and their average length (median = 1573 words) were considerable smaller on the ECHR than on the TRT4 (n=58169) (median = 3071 words), which let us conclude that the size of a document is a fundamental piece that affects to the model, bigger texts have less power to be modeled. We believe that that the ECHR dataset is more homogeneous and less stochastic as long as the no. of possibilities from the ECHR is lower than the TRT4, that is cases from the ECHR are more objective, less redundant than the TRT4, but we suggest a qualitative analysis for future research to corroborate this possibility. On the other side, a similar aspect between both databases was that the framework has a better performance in the class 0 (non-accepted) than accepted (1). On the ECHR, the precision were (label 0 = 0.97, label 1 = 0.832) and on the TRT4 (label 0 = 0.741, label 1 = 0.614), which let us conclude that the models have a better performance in identifying cases that will be not accepted than accepted. Other possibilities for this performance will be that accepted cases exhibit a more stochastic form.

We also validated our framework with experts in that area of Contingent Liabilities (two lawyers and two accountants) by presenting the objectives and results of our work. The first observation that lawyers brought was about the difference between both databases of the studies (ECHR and TRT4). That there is an impossibility of performing an analogy between them because law systems among countries provide substantial differences and depend on local cultural perceptions. They explained that law systems are divided into two broad groups: Common Law and Civil Law. Common-law is mainly used in English-speaking countries. Its primary characteristic is that decisions are based on prior cases, and they depend on the similarities and differences of the cases. They added that the Civil Law (Brazilian system) is based on codes that judges interpret, that precedents are less important critical, and that every case is intended to be framed into a legal concept. However, they emphasized that both systems converged in many aspects. Such as implementing the jury, the appeal of a court ruling, and the construction of legal precedents. However, the lawyers concluded that the structure of a triple argumentation (first, second, third instance) is the same. Synthetically the only difference is the way of how a litigation process is structured. Based on the comments we hypothesize that these structural differences between both databases affected the framework performance. In particular, on how the algorithm identified similar terms to estimate the output. In the ECHR dataset previous cases have more impact. Therefore, a case to be estimated will be more predictable than the TRT4. According to our results, we interpret that the Civil law system (Brazilian) relies more on interpreting a case by the judge against a code - more subjective - than the Common law, which promotes more homogeneity between decisions.

The second aspect relates to the areas of law. The interviewers highlighted that each area has its peculiarities. For example, tax law has substantial differences from civil law. The labor area, as the one we worked on (TRT4), use to have multiple petitions and multiple decisions. Legal actions from other legal areas usually have one petition and one decision, such as moral damage from the civil courts. This clarification went in line with our findings when we performed the data pre-processing from the TRT4, as most accepted petitions were marked as "partially accepted", which means that some petitions were not accepted. For future research, it will be desirable to experiment with other types of Brazilian law areas.

A point remarked from the interviewers is the block of our framework to analyze the similarity between cases. They said that performing previous analyses of a case is important to understand if a litigation case is worthwhile and affirmed that similar cases usually tend to have similar decisions. "With this tool, a petition can be verified to look at the chances of success. If it does not have success, a reconstruction can be performed before submitted to the court, a system for an initial review," they added. From these comments, we interpreted that a litigation process could depend on how a lawsuit is structured and not exactly on concrete facts. Therefore providing suggestions of changes to be performed on a process text to increase the probability of favorable decisions will be a research opportunity. This interpretation was also reinforced with the comment, "what is written on the process are abstractions, words, concepts. None of this is an actual reality." It was also emphasized that the law is not an exact science, and the organic process is not sealed from society's mistakes, that there are innumerable external factors involved, so it will not reach the same limits as the exact sciences. We concluded from these comments that a legal process will always have a limitation of being exactly translated into a mathematical representation.

Our interviewers also pointed out that this tool can bring better ethical issues for users & organizations involved in a petition. For example, lawyers sometimes know in advance that a process will not have chances of success. But they make the petition in a non-ethical course of action. In this line, we identified that our framework tool, besides organizations support, can potentially help users who are part of a litigation process to provide an accessible way to understand the possibilities of win & lose a litigation process, mainly because lawsuit documents look like a black-box due to the specific language used by lawyers. That is a tool like the one we constructed will provide more transparency to users outside of the law area.

Interviewers also noted that the tool could also be helpful for law firms whose strategy is to search for more straightforward cases with high odds of winning as their core business is based on submitting a high quantity of cases instead of analyzing in-depth a particular case. For example a law firm that will prefer to submit multiple cases related to consumer protecting rights than a case that demands more time to by analyzed because the protecting consumer rights type will have higher chances of victory. Finally, it was also emphasized the complexity of working with such an amount of data as lawyers are usually limited to review a minimal set of documents, that it is impossible to review all the related information that a process demand and that a tool like this one will make more efficient their work.

It is plausible that a number of limitations may have influenced the results obtained. The first is that we only use a type of method (DL) and pre-trained model (RODRIGUES et al., 2020) to perform our estimations. The second is that the process for detecting decisions in the TRT4 database depended on a Regex search with defined criteria (a set of pre-defined terms that appear in the last paragraphs), however some of the resolutions may appeared with other type of terms and in other parts of the text. The third is that we only use a specific type of cases from Brazil (labor – rito ordinario), and each type of cases have their particularities such as multiple decisions for one type of case.

Future studies on the current topic are therefore recommended, we propose to perform a qualitative analysis of the estimated results from the framework to understand how the algorithm is internally allocating its weights due to the fact that DL-based algorithms are black-box limited in explaining (cause-reasoning) (CASTELVECCHI, 2016). It will also be helpful to test with a different class of algorithms, such as Random Forests, that provide a logical understanding. Other possibilities can be a topic modeling technique to create clusters of winning and losers. We believe that testing models in other classes of Brazilian litigation databases, such as tax law, will provide helpful insights into the differences between law litigation classes. In addition, there is also a recent interest in developing algorithms not constrained to a fixed number of characters, such as the BERT base. One of these is the “Longformer” (BELTAGY; PETERS; COHAN, 2020) that has gained substantial attention. However, there is no version available in Portuguese. So, there is an opportunity to pre-train this model in Portuguese texts.

References

ABOOD, A.; FELTENBERGER, D. Automated patent landscaping. **Artificial Intelligence and Law**, p. 1–23, 2018.

ABOU-ASSALEH, T. et al. **N-gram-based detection of new malicious code**. Proceedings of the 28th Annual International Computer Software and Applications Conference, 2004. COMPSAC 2004. **Anais**...IEEE, 2004

ADDERLEY, R.; MUSGROVE, P. B. **Data mining case study: Modeling the behavior of offenders who commit serious sexual assaults**. Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. **Anais**...ACM, 2001

ADHIKARI, A. et al. Docbert: Bert for document classification. **arXiv preprint arXiv:1904.08398**, 2019.

AHARONY, J.; LIU, C.; YAWSON, A. Corporate litigation and executive turnover. **Journal of Corporate Finance**, v. 34, p. 268–292, 2015.

AITCHISON, J. Principal component analysis of compositional data. **Biometrika**, v. 70, n. 1, p. 57–65, 1983.

ALETRAS, N. et al. Predicting judicial decisions of the European Court of Human Rights: A natural language processing perspective. **PeerJ Computer Science**, v. 2, p. e93, 2016.

ALSCHNER, W.; SKOUGAREVSKIY, D. **Towards an automated production of legal texts using recurrent neural networks**. Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law. **Anais**...ACM, 2017

ALSENTZER, E. et al. Publicly available clinical BERT embeddings. **arXiv preprint arXiv:1904.03323**, 2019.

AMER, T.; HACKENBRACK, K.; NELSON, M. Between-auditor differences in the interpretation of probability phrases. **Auditing**, v. 13, n. 1, p. 126, 1994.

BANSAL, N.; SHARMA, A.; SINGH, R. K. **A review on the application of deep learning in legal domain**. IFIP International Conference on Artificial Intelligence Applications and Innovations. **Anais**...Springer, 2019

BAOLI, L.; QIN, L.; SHIWEN, Y. An adaptive k-nearest neighbor text categorization strategy. **ACM Transactions on Asian Language Information Processing (TALIP)**, v. 3, n. 4, p. 215–226, 2004.

BARTH, M. E.; MCNICHOLS, M. F.; WILSON, G. P. Factors influencing firms' disclosures about environmental liabilities. **Review of Accounting Studies**, v. 2, n. 1, p. 35–64, 1997.

BELEW, R. K. A connectionist approach to conceptual information retrieval. **Proceedings of the 1st international conference on Artificial intelligence and law**, p. 116–126, 1987.

BELTAGY, I.; PETERS, M. E.; COHAN, A. Longformer: The long-document transformer. **arXiv preprint arXiv:2004.05150**, 2020.

BENCH-CAPON, T. **Neural networks and open texture**. Proceedings of the 4th international conference on Artificial intelligence and law. **Anais**...ACM, 1993

BENGIO, Y. et al. A neural probabilistic language model. **Journal of machine learning research**, v. 3, n. Feb, p. 1137–1155, 2003.

BENGIO, Y. Learning deep architectures for AI. **Foundations and trends® in Machine Learning**, v. 2, n. 1, p. 1–127, 2009.

BLEJER, M. I.; SCHUMACHER, L. Central Bank Use of Derivatives and Other Contingent Liabilities: Analytical Issues and Policy Implications. **Cosponsored by the European Commission and the World Bank (A European Borrowers Network Initiative)**, p. 126, 2000.

BOCHEREAU, L.; BOURCIER, D.; BOURGINE, P. **Extracting legal knowledge by means of a multilayer neural network application to municipal jurisprudence**. Proceedings of the 3rd international conference on Artificial intelligence and law. **Anais**...ACM, 1991

BORGES, F.; BORGES, R.; BOURCIER, D. **Artificial neural networks and legal categorization**. The 16th Annual Conference on Legal Knowledge and Information Systems (JURIX'03). **Anais**...2003

BRANTING, L. K. Data-centric and logic-based models for automated legal problem solving. **Artificial Intelligence and Law**, v. 25, n. 1, p. 5–27, 2017.

BRANTING, L. K. et al. Inducing Predictive Models for Decision Support in Administrative Adjudication. In: **Lecture Notes in Computer Science**. [s.l.] Springer International Publishing, 2018. p. 465–477.

BRIXI, H. P.; SCHICK, A. **Government at risk: contingent liabilities and fiscal risk**. [s.l.] World Bank Publications, 2002.

BROWN, T. B. et al. Language models are few-shot learners. **arXiv preprint arXiv:2005.14165**, 2020.

BUCHMAN, T. A.; COLLINS, D. Uncertainty about litigation losses and auditors' modified audit reports. **Journal of Business Research**, v. 43, n. 2, p. 57–63, 1998.

BUI, D. D. A.; DEL FIOL, G.; JONNALAGADDA, S. PDF text classification to leverage information extraction from publication reports. **Journal of biomedical informatics**, v. 61, p. 141–148, 2016.

CAMBRIA, E.; WHITE, B. Jumping NLP curves: A review of natural language processing research. **IEEE Computational intelligence magazine**, v. 9, n. 2, p. 48–57, 2014.

CASTELVECCHI, D. Can we open the black box of AI? **Nature**, v. 538, n. 7623, p. 20–23, out. 2016.

CER, D. et al. SemEval-2017 Task 1: Semantic Textual Similarity-Multilingual and Cross-lingual Focused Evaluation. **arXiv preprint arXiv:1708.00055**, 2017.

CERKA, P.; GRIGIENE, J.; SIRBIKYTE, G. Liability for damages caused by artificial intelligence. **Computer Law & Security Review**, v. 31, n. 3, p. 376–389, 2015.

CHALKIDIS, I.; ANDROUTSOPOULOS, I. **A Deep Learning Approach to Contract Element Extraction**. . In: JURIX. 2017

CHALKIDIS, I.; ANDROUTSOPOULOS, I.; ALETRAS, N. Neural legal judgment prediction in English. **arXiv preprint arXiv:1906.02059**, 2019.

CHALKIDIS, I.; ANDROUTSOPOULOS, I.; MICHOS, A. Extracting contract elements. **Proceedings of the 16th edition of the International Conference on Articial Intelligence and Law - ICAIL 17**, 2017.

CHALKIDIS, I.; ANDROUTSOPOULOS, I.; MICHOS, A. Obligation and prohibition extraction using hierarchical rnns. **arXiv preprint arXiv:1805.03871**, 2018.

CHALKIDIS, I.; KAMPAS, D. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. **Artificial Intelligence and Law**, p. 1–28, 2018.

CHANG, X. et al. Convex sparse PCA for unsupervised feature learning. **ACM Transactions on Knowledge Discovery from Data (TKDD)**, v. 11, n. 1, p. 3, 2016.

CHAPHALKAR, N.; SANDBHOR, S. S. Application of neural networks in resolution of disputes for escalation clause using neuro-solutions. **KSCE Journal of Civil Engineering**, v. 19, n. 1, p. 10–16, 2015.

CHAU, K. Application of a PSO-based neural network in analysis of outcomes of construction claims. **Automation in construction**, v. 16, n. 5, p. 642–646, 2007.

CHEN, K. et al. Turning from TF-IDF to TF-IGM for term weighting in text classification. **Expert Systems with Applications**, v. 66, p. 245–260, 2016.

CHOI, D. et al. Text analysis for detecting terrorism-related articles on the web. **Journal of Network and Computer Applications**, v. 38, p. 16–21, 2014.

COLLOBERT, R. et al. Natural language processing (almost) from scratch. **Journal of Machine Learning Research**, v. 12, n. Aug, p. 2493–2537, 2011.

COMMITTEE, I. A. S. **Provisions, contingent liabilities and contingent assets**. [s.l.] The Committee, 1998. v. 37

CONNEAU, A. et al. **Very deep convolutional networks for text classification**. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. **Anais**...2017

CONSONI, S.; COLAUTO, R. D. Voluntary disclosure in the context of convergence with International Accounting Standards in Brazil. **Revista brasileira de gestão de negócios**, v. 18, n. 62, p. 658–677, 2016.

CORCORAN, J. et al. Data Clustering and Rule Abduction to Facilitate Crime Hot Spot Prediction. In: **Computational Intelligence. Theory and Applications**. [s.l.] Springer Berlin Heidelberg, 2001. p. 807–821.

CORCORAN, J. J.; WILSON, I. D.; WARE, J. A. Predicting the geo-temporal variations of crime and disorder. **International Journal of Forecasting**, v. 19, n. 4, p. 623–634, out. 2003.

CPC. Pronunciamento Técnico CPC 25: Provisões. **COMITÊ, DE PRONUNCIAMENTOS CONTÁBEIS–Passivos Contingentes e Ativos Contingentes. Brasília, DF**, 2005.

CRAVEN, M. et al. **Learning to extract symbolic knowledge from the World Wide Web**. [s.l.] Carnegie-mellon univ pittsburgh pa school of computer Science, 1998.

DA SILVA, N. C. et al. **Document type classification for Brazil's supreme court using a convolutional neural network**. 10th International Conference on Forensic Computer Science and Cyber Law (ICoFCS), Sao Paulo, Brazil. **Anais**...2018

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

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DIKMEN, I.; BIRGONUL, M. T. Neural network model to support international market entry decisions. **Journal of Construction Engineering and Management**, v. 130, n. 1, p. 59–66, 2004.

DO, P.-K. et al. Legal question answering using ranking SVM and deep convolutional neural network. **arXiv preprint arXiv:1703.05320**, 2017.

DUMAIS, S. T. Latent semantic indexing (LSI): TREC-3 report. **Nist Special Publication SP**, p. 219–219, 1995.

FASB. **Financial Accounting Standards Board. Proposed Accounting Standards Update: Contingencies (topic 450): Disclosure of Certain Loss Contingencies. Exposure Draft.** Norwalk: CT, 2010.

FATTAH, M. A. New term weighting schemes with combination of multiple classifiers for sentiment analysis. **Neurocomputing**, v. 167, p. 434–442, 2015.

FISHER, I. E.; GARNSEY, M. R.; HUGHES, M. E. Natural language processing in accounting, auditing and finance: a synthesis of the literature with a roadmap for future research. **Intelligent Systems in Accounting, Finance and Management**, v. 23, n. 3, p. 157–214, 2016.

FORMAN, G. An extensive empirical study of feature selection metrics for text classification. **Journal of machine learning research**, v. 3, n. Mar, p. 1289–1305, 2003.

GARCÍA-PABLOS, A.; CUADROS, M.; RIGAU, G. W2VLDA: almost unsupervised system for aspect based sentiment analysis. **Expert Systems with Applications**, v. 91, p. 127–137, 2018.

HADDOUD, M. et al. Combining supervised term-weighting metrics for SVM text classification with extended term representation. **Knowledge and Information Systems**, v. 49, n. 3, p. 909–931, 2016.

HAUSER, M. D.; CHOMSKY, N.; FITCH, W. T. The faculty of language: what is it, who has it, and how did it evolve? **science**, v. 298, n. 5598, p. 1569–1579, 2002.

HENNES, K. M. Disclosure of contingent legal liabilities. **Journal of Accounting and Public Policy**, v. 33, n. 1, p. 32–50, 2014.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, v. 9, n. 8, p. 1735–1780, 1997.

HOFFMAN, V. B.; PATTON, J. M. Accountability, the dilution effect, and conservatism in auditors' fraud judgments. **Journal of Accounting Research**, v. 35, n. 2, p. 227–237, 1997.

JALILVAND, A.; SALIM, N. Feature unionization: a novel approach for dimension reduction. **Applied Soft Computing**, v. 52, p. 1253–1261, 2017.

JINDAL, N.; LIU, B. **Review spam detection**. Proceedings of the 16th international conference on World Wide Web. **Anais**...ACM, 2007

JOACHIMS, T. **Text categorization with support vector machines: Learning with many relevant features**. European conference on machine learning. **Anais**...Springer, 1998

JOHN, A. K. et al. Legalbot: A Deep Learning-Based Conversational Agent in the Legal Domain. In: **Natural Language Processing and Information Systems**. [s.l.] Springer International Publishing, 2017. p. 267–273.

JOULIN, A. et al. Bag of tricks for efficient text classification. **arXiv preprint arXiv:1607.01759**, 2016.

KANG, M.; AHN, J.; LEE, K. Opinion mining using ensemble text hidden Markov models for text classification. **Expert Systems with Applications**, v. 94, p. 218–227, 2018.

KATZ, D. M. et al. Predicting the behavior of the supreme court of the united states: A general approach. **arXiv preprint arXiv:1407.6333**, 2014.

KIM, S.-B. et al. Some effective techniques for naive bayes text classification. **IEEE transactions on knowledge and data engineering**, v. 18, n. 11, p. 1457–1466, 2006.

KIM, Y. Convolutional neural networks for sentence classification. **arXiv preprint arXiv:1408.5882**, 2014.

KOPROWSKI, W.; ARSENAULT, S. J.; CIPRIANA, M. Financial Statement Reporting of Pending Litigation: Attorneys, Auditors, and Differences of Opinions. **Fordham J. Corp. & Fin. L.**, v. 15, p. 439, 2009.

KORENIUS, T. et al. **Stemming and lemmatization in the clustering of finnish text documents**. Proceedings of the thirteenth ACM international conference on Information and knowledge management. **Anais**...ACM, 2004

KOWSRIHAWAT, K.; VATEEKUL, P.; BOONKWAN, P. **Predicting Judicial Decisions of Criminal Cases from Thai Supreme Court Using Bi-directional GRU with Attention Mechanism**. 2018 5th Asian Conference on Defense Technology (ACDT). **Anais**...IEEE, 2018

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. **Imagenet classification with deep convolutional neural networks**. Advances in neural information processing systems. **Anais**...2012

KUNZ, S. N. From Legally Confidential to Financially Confident: Resolving the Tension between Lawyers and Auditors over Contingent Liability Disclosure. **CMC Senior Theses**, p. Paper 1073, 2015.

KUSHMERICK, N.; JOHNSTON, E.; MCGUINNESS, S. **Information extraction by text classification**. In The IJCAI-2001 Workshop on Adaptive Text Extraction and Mining. **Anais**...Citeseer, 2001

LABANI, M. et al. A novel multivariate filter method for feature selection in text classification problems. **Engineering Applications of Artificial Intelligence**, v. 70, p. 25–37, 2018.

LAI, Y.-H.; CHE, H.-C. Modeling patent legal value by Extension Neural Network. **Expert Systems with Applications**, v. 36, n. 7, p. 10520–10528, 2009.

LARSEN-FREEMAN, D.; CAMERON, L. **Complex systems and applied linguistics**. [s.l.] Oxford University Press Oxford, 2008.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, v. 521, n. 7553, p. 436–444, 2015.

LEE, J. et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. **Bioinformatics**, v. 36, n. 4, p. 1234–1240, 2020.

LEWIS, D. D. et al. Rcv1: A new benchmark collection for text categorization research. **Journal of machine learning research**, v. 5, n. Apr, p. 361–397, 2004.

LI, C. H.; YANG, J. C.; PARK, S. C. Text categorization algorithms using semantic approaches, corpus-based thesaurus and WordNet. **Expert Systems with Applications**, v. 39, n. 1, p. 765–772, 2012.

LI, S. et al. DeepPatent: patent classification with convolutional neural networks and word embedding. **Scientometrics**, v. 117, n. 2, p. 721–744, 2018.

LI, X. et al. Exploiting BERT for end-to-end aspect-based sentiment analysis. **arXiv preprint arXiv:1910.00883**, 2019.

LIU, Y. et al. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint arXiv:1907.11692**, 2019.

LIU, Z.; CHEN, H. **A predictive performance comparison of machine learning models for judicial cases**. 2017 IEEE Symposium Series on Computational Intelligence (SSCI). **Anais**...IEEE, 2017

MALMASI, S.; ZAMPIERI, M. Detecting hate speech in social media. **arXiv preprint arXiv:1712.06427**, 2017.

MANNING, C.; RAGHAVAN, P.; SCHÜTZE, H. Introduction to information retrieval. **Natural Language Engineering**, v. 16, n. 1, p. 100–103, 2010.

MCILROY, S. et al. User reviews of top mobile apps in Apple and Google app stores. **Communications of the ACM**, v. 60, n. 11, p. 62–67, 2017.

MERKL, D. **A connectionist view on document classification**. Australasian Database Conference. **Anais**...1995a

MERKL, D. **Content-based document classification with highly compressed input data**. Int. Conference on Artificial Neural Networks. **Anais**...Citeseer, 1995b

MERKL, D.; SCHWEIGHOFER, E. **The exploration of legal text corpora with hierarchical neural networks: A guided tour in public international law**. ICAIL. **Anais**...Citeseer, 1997

MERKL, D.; SCHWEIGHOFER, E.; WINIWATER, W. **Analysis of legal thesauri based on self-organising feature maps**. 1995 Fourth International Conference on Artificial Neural Networks. **Anais**...IET, 1995

MERKL, D.; SCHWEIGHOFFER, E.; WINIWARTER, W. Exploratory analysis of concept and document spaces with connectionist networks. **Artificial Intelligence and Law**, v. 7, n. 2–3, p. 185–209, 1999.

MIKOLOV, T.; YIH, W.; ZWEIG, G. **Linguistic regularities in continuous space word representations**. Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. **Anais**...2013

MIRONCZUK, M. M.; PROTASIEWICZ, J. A recent overview of the state-of-the-art elements of text classification. **Expert Systems with Applications**, 2018.

MIYATO, T.; DAI, A. M.; GOODFELLOW, I. Adversarial Training Methods for Semi-Supervised Text Classification. **arXiv preprint arXiv:1605.07725**, 2016.

MONTELONGO, A.; BECKER, J. L. **Tasks performed in the legal domain through Deep Learning: A bibliometric review (1987–2020)**. 2020 International Conference on Data Mining Workshops (ICDMW). **Anais**... In: 2020 INTERNATIONAL CONFERENCE ON DATA MINING WORKSHOPS (ICDMW). Sorrento, Italy: IEEE, nov. 2020aDisponível em: <https://ieeexplore.ieee.org/document/9346339/>.

MONTELONGO, A.; BECKER, J. L. **A bibliometric network analysis of Deep Learning publications applied into legal documents**. 2020 IEEE International Conference on Big Data (Big Data). **Anais**... In: 2020 IEEE INTERNATIONAL CONFERENCE ON BIG DATA (BIG DATA).

Atlanta, GA, USA: IEEE, 10 dez. 2020bDisponível em: <https://ieeexplore.ieee.org/document/9377970/>.

MONTGOMERY, J. M.; HOLLENBACH, F. M.; WARD, M. D. Improving predictions using ensemble Bayesian model averaging. **Political Analysis**, v. 20, n. 3, p. 271–291, 2012.

MORIMOTO, A. et al. **Legal Question Answering System using Neural Attention**. COLIEE@ICAIL. **Anais**...2017

NAJAFABADI, M. M. et al. Deep learning applications and challenges in big data analytics. **Journal of Big Data**, v. 2, n. 1, p. 1, dez. 2015.

NANDA, R. et al. **Legal Information Retrieval Using Topic Clustering and Neural Networks.** COLIEE@ ICAIL. **Anais**...2017

NANDA, R. et al. Unsupervised and supervised text similarity systems for automated identification of national implementing measures of European directives. **Artificial Intelligence and Law**, p. 1–27, 2018.

NG, A. Y.; JORDAN, M. I. **On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes**. Advances in neural information processing systems. **Anais**...2002

NGUYEN, T.-S. et al. Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts. **Artificial Intelligence and Law**, p. 1–31, 2018.

NIGAM, K.; LAFFERTY, J.; MCCALLUM, A. **Using maximum entropy for text classification**. IJCAI-99 workshop on machine learning for information filtering. **Anais**...1999

NIRENBURG, S.; MCSHANE, M. Natural language processing. **The Oxford Handbook of Cognitive Science**, p. 337, 2016.

OATLEY, G.; EWART, B.; ZELEZNIKOW, J. Decision support systems for police: Lessons from the application of data mining techniques to soft forensic evidence. **Artificial Intelligence and Law**, v. 14, n. 1–2, p. 35–100, 2006.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. **Glove: Global vectors for word representation**. Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). **Anais**...2014

PETERS, M. E. et al. Deep contextualized word representations. **arXiv preprint arXiv:1802.05365**, 2018.

PHILIPPS, L. A Neural Network to Identify Legal Precedents. 1989a.

PHILIPPS, L. Are Legal Decisions based on the Application of Rules or Prototype Recognition? 1989b.

PHILIPPS, L. Distribution of damages in car accidents through the use of neural networks. **Cardozo L. Rev.**, v. 13, p. 987, 1991.

RISH, I. **An empirical study of the naive Bayes classifier**. IJCAI 2001 workshop on empirical methods in artificial intelligence. **Anais**...2001

RODRIGUES, R. et al. **Multilingual Transformer Ensembles for Portuguese Natural Language Tasks**. 2020

RUGER, T. W. et al. The Supreme Court forecasting project: Legal and political science approaches to predicting Supreme Court decisionmaking. **Columbia Law Review**, p. 1150–1210, 2004.

SADEGHIAN, A. et al. **Semantic edge labeling over legal citation graphs**. Proceedings of the workshop on legal text, document, and corpus analytics (LTDCA-2016). **Anais**...2016

SADEGHIAN, A. et al. Automatic semantic edge labeling over legal citation graphs. **Artificial Intelligence and Law**, v. 26, n. 2, p. 127–144, 2018.

SANDBHOR, S.; CHAPHALKAR, N. Impact of Outlier Detection on Neural Networks Based Property Value Prediction. In: **Information Systems Design and Intelligent Applications**. [s.l.] Springer, 2019. p. 481–495.

SARTOR, G.; BRANTING, L. K. Introduction: judicial applications of artificial intelligence. In: **Judicial Applications of Artificial Intelligence**. [s.l.] Springer, 1998. p. 1–6.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, v. 61, p. 85–117, 2015.

SCHÖLKOPF, B.; SMOLA, A. J.; BACH, F. **Learning with kernels: support vector machines, regularization, optimization, and beyond**. [s.l.] MIT press, 2002.

SCHUETZE, H. **Document information retrieval using global word co-occurrence patterns**, out. 1997.

SHAHINFAR, S.; MEEK, P.; FALZON, G. "How many images do I need?" Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. **Ecological Informatics**, p. 101085, 2020.

SHARMA, R. D. et al. **Using Modern Neural Networks to Predict the Decisions of Supreme Court of the United States with State-of-the-Art Accuracy**. International Conference on Neural Information Processing. **Anais**...Springer, 2015

SOCHER, R.; MUNDRA, R. S. CS 224D: Deep Learning for NLP1. 2016.

SOKOLOVA, M.; JAPKOWICZ, N.; SZPAKOWICZ, S. **Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation**. Australasian joint conference on artificial intelligence. **Anais**...Springer, 2006

SON, N. T. et al. **Recognizing logical parts in legal texts using neural architectures**. Knowledge and Systems Engineering (KSE), 2016 Eighth International Conference on. **Anais**...IEEE, 2016

STRANIERI, A. et al. A hybrid rule–neural approach for the automation of legal reasoning in the discretionary domain of family law in Australia. **Artificial Intelligence and Law**, v. 7, n. 2–3, p. 153–183, 1999.

STRANIERI, A.; ZELEZNIKOW, J. Knowledge discovery from legal databases using neural networks and data mining to build legal decision support systems. In: **Information Technology and Lawyers**. [s.l.] Springer, 2006. p. 81–117.

SUGATHADASA, K. et al. Synergistic Union of Word2Vec and Lexicon for Domain Specific Semantic Similarity. **arXiv preprint arXiv:1706.01967**, 2017.

SULEA, O.-M. et al. Predicting the law area and decisions of french supreme court cases. **arXiv preprint arXiv:1708.01681**, 2017.

SUTSKEVER, I.; VINYALS, O.; LE, Q. V. **Sequence to sequence learning with neural networks**. Advances in neural information processing systems. **Anais**...2014

THAGARD, P. Connectionism and legal inference. **Cardozo law review**, v. 13, p. 1001, 1991.

THAMMABOOSADEE, S.; WATANAPA, B.; CHAROENKITKARN, N. A framework of multi-stage classifier for identifying criminal law sentences. **Procedia Computer Science**, v. 13, p. 53–59, 2012.

TIAN, J. et al. **ECNU at SemEval-2017 Task 1: Leverage Kernel-based Traditional NLP features and Neural Networks to Build a Universal Model for Multilingual and Cross-lingual Semantic Textual Similarity**. Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017). **Anais**...2017

TRAN, V. et al. Encoded summarization: summarizing documents into continuous vector space for legal case retrieval. **Artificial Intelligence and Law**, p. 1–27, 2020.

TRAPPEY, A. J. C. et al. Development of a patent document classification and search platform using a back-propagation network. **Expert Systems with Applications**, v. 31, n. 4, p. 755–765, 2006.

UNDAVIA, S.; MEYERS, A.; ORTEGA, J. E. **A Comparative Study of Classifying Legal Documents with Neural Networks**. 2018 Federated Conference on Computer Science and Information Systems (FedCSIS). **Anais**...IEEE, 2018

VALDIVIA, A.; LUZON, M. V.; HERRERA, F. Sentiment analysis in tripadvisor. **IEEE Intelligent Systems**, v. 32, n. 4, p. 72–77, 2017.

VIJAYARANI, S.; ILAMATHI, M. J.; NITHYA, M. Preprocessing techniques for text mining-an overview. **International Journal of Computer Science & Communication Networks**, v. 5, n. 1, p. 7–16, 2015.

VOGEL, F.; HAMANN, H.; GAUER, I. Computer-assisted legal linguistics: corpus analysis as a new tool for legal studies. **Law & Social Inquiry**, v. 43, n. 4, p. 1340–1363, 2018.

WANG, S.; MANNING, C. D. **Baselines and bigrams: Simple, good sentiment and topic classification**. Proceedings of the 50th annual meeting of the association for computational linguistics: Short papers-volume 2. **Anais**...Association for Computational Linguistics, 2012

WAYNE, C. L. **Multilingual Topic Detection and Tracking: Successful Research Enabled by Corpora and Evaluation.** LREC. **Anais**...2000

WIETING, J.; GIMPEL, K. Revisiting Recurrent Networks for Paraphrastic Sentence Embeddings. **arXiv preprint arXiv:1705.00364**, 2017.

WOOFF, D. **Logistic regression: a self-learning text**. [s.l.] JSTOR, 2004.

WU, T. et al. **Twitter spam detection based on deep learning**. Proceedings of the australasian computer science week multiconference. **Anais**...2017

YANG, W. et al. Legal Judgment Prediction via Multi-Perspective Bi-Feedback Network. **arXiv preprint arXiv:1905.03969**, 2019.

YANG, Z. et al. **Hierarchical attention networks for document classification**. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. **Anais**...2016.

YENTER, A.; VERMA, A. **Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis**. 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON). **Anais**...IEEE, 2017

YI, K.; BEHESHTI, J. A hidden Markov model-based text classification of medical documents. **Journal of Information Science**, v. 35, n. 1, p. 67–81, 2009.

YOUSEFI-AZAR, M.; HAMEY, L. Text summarization using unsupervised deep learning. **Expert Systems with Applications**, v. 68, p. 93–105, 2017.

ZELEZNIKOW, J.; VOSSOS, G.; HUNTER, D. The IKBALS project: Multi-modal reasoning in legal knowledge based systems. **Artificial Intelligence and Law**, v. 2, n. 3, p. 169–203, 1993.

ZHANG, M.-L.; ZHOU, Z.-H. A k-nearest neighbor based algorithm for multi-label classification. **GrC**, v. 5, p. 718–721, 2005.

ZHANG, T.; OLES, F. J. Text categorization based on regularized linear classification methods. **Information retrieval**, v. 4, n. 1, p. 5–31, 2001.

ZHU, S. et al. **Multi-labelled classification using maximum entropy method**. Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. **Anais**...ACM, 2005

Attachment 1

a) Unify petitions and resolutions

```
####################################################################################################
###
# Programming language: R
# Description: Unify petitions and resolutions that are stored in different folders. We created this script as information
were provided in two separated folders one corresponding to the petitions and the other to the resolutions. Moreover, the
script check that each of the petitions have its resolution, as some of the files were incomplete.
# Input: two folders: one containing a set of petitions and the other the resolutions in the form of PDF.
# Output: a matrix array form saved as csv.
####################################################################################################
###

library(data.table)

# Diretory to read
setwd("/media/alfredo/F/iniciais_com_sentenca/RtOrd")

# Read file names. Distinguish between incial and sentenca.
inicial <- list.files(pattern="inicial.pdf", full.names = FALSE, ignore.case = TRUE)
sentenca <- list.files(pattern="sentenca.pdf", full.names = FALSE, ignore.case = TRUE)

# Match inicial with sentencas. There are some sentences that do not have their pair. I make a match.
inicial_clean <- gsub("_inicial.pdf", "", inicial)
sentenca_clean <- gsub("_sentenca.pdf", "", sentenca)
inicial_clean <- data.table(no_processo=inicial_clean)
sentenca_clean <- data.table(no_processo=sentenca_clean)
complete_process <-merge(inicial_clean, sentenca_clean, by = "no_processo")
complete_process[, inicial:=paste0(no_processo, "_inicial.pdf")]
complete_process[, sentenca:=paste0(no_processo, "_sentenca.pdf")]
#setwd("~/MEGA/2020/Doutorado/Defensa/pdf_to_text/lists")
#write.csv2(complete_process, "complete_process.csv", row.names = FALSE)

# Read data from file
complete_process <- read.csv("/home/alfredo/MEGA/2020/Doutorado/Defensa/pdf_to_text/list_of_process_numbers/
RTord_all.csv", sep=";", stringsAsFactors = FALSE)
complete_process <- as.data.table(complete_process)

# Select samples
samples_to_select <- sample.int(dim(complete_process)[1], 300)
selected_processes <- complete_process[samples_to_select]

# All samples
selected_processes <- complete_process
initial_sample <- selected_processes$inicial
sentenca_sample <-selected_processes$sentenca

#setwd("/home/alfredo/MEGA/2020/Doutorado/Defensa/pdf_to_text/samples")
#write.csv2(selected_processes, "samples_300.csv", row.names = FALSE)

# Folder of origin
setwd("/media/alfredo/F/iniciais_com_sentenca/RtOrd")

# Divide iniciais and sentencas
new_folder <- "/media/alfredo/F/working/RTOrd/iniciais"
file.copy(initial_sample, new_folder)
new_folder <- "/media/alfredo/F/working/RTOrd/sentencas"
file.copy(sentenca_sample, new_folder)
```

b) Transform PDF into text

```
##############################################################################################
######
# Programming language: R
# Description: Transform a set of petitions in the form of PDF into text and save as an array form.
# Input: a folder containing a set of petitions in the form of PDF.
# Output: a matrix array form saved as csv.
##############################################################################################
######

# Extract text and find resolution sentences.
library(pdftools)
library(stringr)
library(data.table)
library(hunspell) # Check spelling

# Path for a folder to read
process.path <- "/media/alfredo/F/RTSum/iniciais"
setwd(process.path)

# Create a vector of file names to extract.
file.list <- list.files(".", full.names = TRUE, pattern = '.pdf$')

# Empty list to store sentence
# resolution.list = list()
files.processed = list()

# Read all sequence of files
for (i in 1:length(file.list)){
  no.process <- file.list[i]
  print(i)
  setwd("/media/alfredo/F/RTSum/iniciais")
  process <-pdf_text(no.process) # Read data

  # Extract type of resolution
  no.petition <-substr(no.process, 3, nchar(no.process)) # Create id to store table
  process <- tolower(process)

  # Save sentence resolution into a DT
  files.processed[[i]] <- no.petition

  # Save into a data table. Each row represents a page
  mylist <- do.call(rbind, as.list(process))
  process <- data.table(mylist)

  ### Section to process an save resolutions as text.
  # Remove last page of sentences. It seems extra information
  last.row <- dim(process)[1]
  process <- process[1:last.row-1]

  # Create output file names
  petition.no <- substr(no.process, 1, nchar(no.process)-3) # Substract "pdf" strings
  petition.no <- substr(petition.no, 3, nchar(petition.no)) # Substract ./ to avoid possible errors in future reading.
  f.name.output <- paste0("petition_", petition.no, "txt")
  f.name.output.erro <- paste0("erro_", petition.no, "txt")
  ### Remove unecessary lines
  petition <- process
  text.lines <- lapply(petition$V1, function(x)readLines(textConnection(x))) # Convert each line to row
  text.lines <-lapply(text.lines, str_squish) # Remove white spaces from start

  # Specify text pattern to remove
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Fls")]) # Start of the page
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Documento assinado pelo Shodo")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Assinado eletronicamente.")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "https://pje.trt4.jus.br/")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Número do processo:")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Número do documento:")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Data de Juntada:")])

  # Colapse text
  text.lines.collapsed <- lapply(text.lines, paste, collapse = " ") # Collapse vector of each page.
  vec.text.lines <- unlist(text.lines.collapsed) # Unlist to create a unique vector document
  petition.text <-paste(vec.text.lines, collapse = " ") # Transform the vector into a piece of text

  # Write file
  setwd("/media/alfredo/F/RTSum/iniciais_text")
  fileConn <- file(f.name.output)
  writeLines(petition.text, fileConn)
  close(fileConn)
}
```

c) Detect decisions

```
##############################################################################################
######
# Programming language: R
# Description: Detect the decision according to a  predefined set of words.
# Input: a folder containing a set of resolutions in the form of PDF.
# Output: a matrix array form saved as csv.
##############################################################################################
######

# Libraries
library(pdftools)
library(stringr)
library(data.table)
library(hunspell)

# Seth process path to read the samples.
process.path <- " " # Set the path of the folder
setwd(process.path)

# Create a list to extract
file.list <- list.files(".", full.names = TRUE, pattern = '.pdf$')

# Empty list to store resolutions
resolution.list = list()

# Read all sequence of files
for (i in 1:length(file.list)){
  setwd(process.path)
  no.process <- file.list[i]
  no.process
  print(i)
  process <-pdf_text(no.process) # Read data

  # Extract type of resolution
  no.petition.resolution <-substr(no.process, 3, nchar(no.process)) # Create id to store table
  process <- tolower(process)

  improcedente <- str_detect(process, c("improcedente", "improcedentes")) # Detect words for improcedente.
  improcedente.pos <- max(which(improcedente, TRUE)) # Select the maximum position. Locating the page
number.
  procedente_em_parte <- str_detect(process, c("procedente em parte|procedentes em parte"))
  procedente_em_parte.pos <- max(which(procedente_em_parte, TRUE))
  sem_resolucao.pos <- str_detect(process, c("sem resolução de mérito"))
  sem_resolucao.pos <- max(which(sem_resolucao.pos, TRUE))
  resolucoes <- data.table(no_petition=no.petition.resolution, improcedente=improcedente.pos,
procedente_em_parte=procedente_em_parte.pos, sem_resolucao=sem_resolucao.pos)

  # Save into a DT
  resolution.list[[i]] <- resolucoes

  # Save into a data table. Each row represents a page
  mylist <- do.call(rbind, as.list(process))
  process <- data.table(mylist)
  # Remove last page of sentences. It seems extra information
  last.row <- dim(process)[1]
  process <- process[1:last.row-1]

  # Create output file names
  petition.no <- substr(no.process, 1, nchar(no.process)-3) # Substract "pdf" strings
  petition.no <- substr(petition.no, 3, nchar(petition.no)) # Substract ./ to avoid possible errors in future reading.
  f.name.output <- paste0("petition_", petition.no, "txt")
  f.name.output.erro <- paste0("erro_", petition.no, "txt")

  ### Remove unecessary lines
  petition <- process
  text.lines <- lapply(petition$V1, function(x)readLines(textConnection(x))) # Convert each line to row
  text.lines <-lapply(text.lines, str_squish) # Remove white spaces from start

  # Specify text pattern to remove
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Fls")]) # Start of the page
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Documento assinado pelo Shodo")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Assinado eletronicamente.")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "https://pje.trt4.jus.br/")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Número do processo:")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Número do documento:")])
  text.lines <- lapply(text.lines, function(x) x[!startsWith(x, "Data de Juntada:")])

  # Colapse text
  text.lines.collapsed <- lapply(text.lines, paste, collapse = " ") # Collapse vector of each page.
  vec.text.lines <- unlist(text.lines.collapsed) # Unlist to create a unique vector document
  petition.text <-paste(vec.text.lines, collapse = " ") # Transform the vector into a piece of text

  # Write file
  # setwd("/home/alfredo/MEGA/2020/Doutorado/Defensa/pdf_to_text/samples/RTOrd_results/sentencas")
  #fileConn <- file(f.name.output)
  #writeLines(petition.text, fileConn)
  #close(fileConn)
}

resolutions.table <- rbindlist(resolution.list)
resolutions.table.melt <- melt(resolutions.table, id.vars = c("no_petition"))
resolutions.table.melt[, max.page:=max(value), by=c("no_petition")]
resolutions.table.melt[, max.value:=value-max.page, by=c("no_petition")]
resolutions.table.melt[max.value==0, no.sentencas:=.N, by=c("no_petition")]
resolutions.table.melt[max.value==0, sentenca:=variable]

# Organize data
resolutions_clean <- resolutions.table.melt[max.value==0]
resolutions_clean_one <- resolutions_clean[no.resolucoes==1]

write.csv2(resolutions.table.melt, "resolucoes_all.csv", row.names = FALSE)
write.csv2(resolutions_clean_one, "resolucoes_one.csv", row.names = FALSE)
```

d) Numerical representation

```
#############################################################################################
######
# Programming language: Python
# Description: Train and predict a set of document texts and predict is classification, using a BERT-LSTM
architecture.
# Input: a matrix array of texts with their corresponding class.
# Output: a trained model used to predict a class according to a text.
#############################################################################################
######

from transformers import CONFIG_NAME, WEIGHTS_NAME
from transformers.modeling_bert import BertConfig
from transformers.tokenization_bert import BertTokenizer
from torch import nn
import torch,math,logging,os
from sklearn.metrics import f1_score, precision_score, recall_score

from .document_bert_architectures import DocumentBertLSTM

def encode_documents(documents: list, tokenizer: BertTokenizer, max_input_length=512):
    tokenized_documents = [tokenizer.tokenize(document)[:10200] for document in documents]  #added by AD (only
take first 10200 tokens of each documents as input)
    max_sequences_per_document = math.ceil(max(len(x)/(max_input_length-2) for x in tokenized_documents))
    assert max_sequences_per_document <= 20, "Your document is to large"

    output = torch.zeros(size=(len(documents), max_sequences_per_document, 3, 512), dtype=torch.long)

    for doc_id in range( len(documents) ):
        for seq_id in range( max_sequences_per_document ):
            output[doc_id,seq_id,0]=torch.LongTensor( tokenizer.convert_tokens_to_ids( [ '[CLS]' , '[SEP]' ] )
+[0]*(512-2)  ) #input_ids
            output[doc_id,seq_id,2]=torch.LongTensor( [1]*2+[0]*(512-2)  ) #attention_mask


    document_seq_lengths = [] #number of sequence generated per document
    #Need to use 510 to account for 2 padding tokens
    for doc_index, tokenized_document in enumerate(tokenized_documents):
        max_seq_index = 0
        for seq_index, i in enumerate(range(0, len(tokenized_document), (max_input_length-2))):
            raw_tokens = tokenized_document[i:i+(max_input_length-2)]
            tokens = []
            input_type_ids = []

            tokens.append("[CLS]")
            input_type_ids.append(0)
            for token in raw_tokens:
                tokens.append(token)
                input_type_ids.append(0)
            tokens.append("[SEP]")
            input_type_ids.append(0)

            input_ids = tokenizer.convert_tokens_to_ids(tokens)
            attention_masks = [1] * len(input_ids)
            while len(input_ids) < max_input_length:
                input_ids.append(0)
                input_type_ids.append(0)
                attention_masks.append(0)

            assert len(input_ids) == 512 and len(attention_masks) == 512 and len(input_type_ids) == 512

            #we are ready to rumble
            output[doc_index][seq_index] = torch.cat((torch.LongTensor(input_ids).unsqueeze(0),
                                                           torch.LongTensor(input_type_ids).unsqueeze(0),
                                                           torch.LongTensor(attention_masks).unsqueeze(0)),
                                                          dim=0)
            max_seq_index = seq_index
        document_seq_lengths.append(max_seq_index+1)
    return output, torch.LongTensor(document_seq_lengths)


document_bert_architectures = {
    'DocumentBertLSTM': DocumentBertLSTM,
}

class BertForDocumentClassification():
    def __init__(self,args=None,
                 labels=None,
                 device='cuda',
                 bert_model_path='bert-base-uncased',
                 architecture="DocumentBertLSTM",
                 batch_size=10,
                 bert_batch_size=7,
                 learning_rate = 5e-5,
                 weight_decay=0,
                 use_tensorboard=False):
        if args is not None:
            self.args = vars(args)
        if not args:
            self.args = {}
            self.args['bert_model_path'] = bert_model_path
            self.args['device'] = device
            self.args['learning_rate'] = learning_rate
            self.args['weight_decay'] = weight_decay
            self.args['batch_size'] = batch_size
            self.args['labels'] = labels
            self.args['bert_batch_size'] = bert_batch_size
            self.args['architecture'] = architecture
            self.args['use_tensorboard'] = use_tensorboard
        if 'fold' not in self.args:
            self.args['fold'] = 0

        assert self.args['labels'] is not None, "Must specify all labels in prediction"

        self.log = logging.getLogger()
        if 'Distil' in self.args['architecture']:
            ArchitectureConfig=DistilBertConfig
            self.bert_tokenizer = DistilBertTokenizer.from_pretrained( self.args['bert_model_path']  )

        else:
            ArchitectureConfig=BertConfig
            self.bert_tokenizer = BertTokenizer.from_pretrained( self.args['bert_model_path']  )


        if os.path.exists(self.args['bert_model_path']):
            if os.path.exists(os.path.join(self.args['bert_model_path'], CONFIG_NAME)):
                config = ArchitectureConfig.from_json_file(os.path.join(self.args['bert_model_path'], CONFIG_NAME))
            elif os.path.exists(os.path.join(self.args['bert_model_path'], 'bert_config.json')):

                config = ArchitectureConfig.from_json_file(os.path.join(self.args['bert_model_path'], 'bert_config.json'))
            else:
                raise ValueError("Cannot find a configuration for the BERT based model you are attempting to load.")
        else:
            config = ArchitectureConfig.from_pretrained(self.args['bert_model_path'])
        config.__setattr__('num_labels',len(self.args['labels']))
        config.__setattr__('bert_batch_size',self.args['bert_batch_size'])

        if 'use_tensorboard' in self.args and self.args['use_tensorboard']:
            assert 'model_directory' in self.args is not None, "Must have a logging and checkpoint directory set."
            from torch.utils.tensorboard import SummaryWriter
            self.tensorboard_writer = SummaryWriter(os.path.join(self.args['model_directory'],
                                                                  "..",
                                                                  "runs",
                                                                  self.args['model_directory'].split(os.path.sep)[-
1]+'_'+self.args['architecture']+'_'+str(self.args['fold'])))


        self.bert_doc_classification =
document_bert_architectures[self.args['architecture']].from_pretrained(self.args['bert_model_path'], config=config)


        #Change these lines if you want to freeze bert, unfreeze bert, or only freeze last layers of BERT
        self.bert_doc_classification.freeze_bert_encoder()
        self.bert_doc_classification.unfreeze_bert_encoder_last_layers()

        self.optimizer = torch.optim.Adam(
            self.bert_doc_classification.parameters(),
            weight_decay=self.args['weight_decay'],
            lr=self.args['learning_rate']
        )


    def fit(self, train, dev):
        """
        A list of
        :param documents: a list of documents
        :param labels: a list of label vectors
        :return:
        """

        train_documents, train_labels = train
        dev_documents, dev_labels = dev

        self.bert_doc_classification.train()

        document_representations, document_sequence_lengths  = encode_documents(train_documents,
self.bert_tokenizer)
```

```
        correct_output = torch.FloatTensor(train_labels)

        loss_weight = ((correct_output.shape[0] / torch.sum(correct_output, dim=0))-1).to(device=self.args['device'])
        self.loss_function = torch.nn.BCEWithLogitsLoss(pos_weight=loss_weight)

        assert document_representations.shape[0] == correct_output.shape[0]

        if torch.cuda.device_count() > 1:
            pass
            #self.bert_doc_classification = torch.nn.DataParallel(self.bert_doc_classification)
        self.bert_doc_classification.to(device=self.args['device'])

        for epoch in range(1,self.args['epochs']+1):
            # shuffle
            permutation = torch.randperm(document_representations.shape[0])
            document_representations = document_representations[permutation]
            document_sequence_lengths = document_sequence_lengths[permutation]
            correct_output = correct_output[permutation]

            self.epoch = epoch
            epoch_loss = 0.0
            for i in range(0, document_representations.shape[0], self.args['batch_size']):

                batch_document_tensors = document_representations[i:i +
self.args['batch_size']].to(device=self.args['device'])
                batch_document_sequence_lengths= document_sequence_lengths[i:i+self.args['batch_size']]
                #self.log.info(batch_document_tensors.shape)
                batch_predictions = self.bert_doc_classification(batch_document_tensors,
                                                                batch_document_sequence_lengths,
                                                                device=self.args['device'])

                batch_correct_output = correct_output[i:i + self.args['batch_size']].to(device=self.args['device'])
                loss = self.loss_function(batch_predictions, batch_correct_output)
                epoch_loss += float(loss.item())
                loss.backward()
                self.optimizer.step()
                self.optimizer.zero_grad()

            epoch_loss /= int(document_representations.shape[0] / self.args['batch_size'])  # divide by number of batches
per epoch

            if 'use_tensorboard' in self.args and self.args['use_tensorboard']:
                self.tensorboard_writer.add_scalar('Loss/Train', epoch_loss, self.epoch)

            self.log.info('Epoch %i Completed: %f' % (epoch, epoch_loss))

            if epoch % self.args['checkpoint_interval'] == 0:
                self.save_checkpoint(os.path.join(self.args['model_directory'], "checkpoint_%s" % epoch))

            # evaluate on development data
            if epoch % self.args['evaluation_interval'] == 0:
                self.predict((dev_documents, dev_labels))

    def predict(self, data, threshold=0):

        document_representations = None
        document_sequence_lengths = None
        correct_output = None
        if isinstance(data, list):
            document_representations, document_sequence_lengths = encode_documents(data, self.bert_tokenizer)
        if isinstance(data, tuple) and len(data) == 2:
            self.log.info('Evaluating on Epoch %i' % (self.epoch))
            document_representations, document_sequence_lengths = encode_documents(data[0], self.bert_tokenizer)
            correct_output = torch.FloatTensor(data[1]).transpose(0,1)
            assert self.args['labels'] is not None

        self.bert_doc_classification.to(device=self.args['device'])
        self.bert_doc_classification.eval()
        with torch.no_grad():
            predictions = torch.empty((document_representations.shape[0], len(self.args['labels'])))
            for i in range(0, document_representations.shape[0], self.args['batch_size']):
                batch_document_tensors = document_representations[i:i +
self.args['batch_size']].to(device=self.args['device'])
                batch_document_sequence_lengths= document_sequence_lengths[i:i+self.args['batch_size']]

                prediction = self.bert_doc_classification(batch_document_tensors,
                                                          batch_document_sequence_lengths,device=self.args['device'])
                predictions[i:i + self.args['batch_size']] = prediction

        for r in range(0, predictions.shape[0]):
            for c in range(0, predictions.shape[1]):
                if predictions[r][c] > threshold:
                    predictions[r][c] = 1
                else:
                    predictions[r][c] = 0
        predictions = predictions.transpose(0, 1)


        if correct_output is None:
            return predictions.cpu()
        else:
            assert correct_output.shape == predictions.shape
            precisions = []
            recalls = []
            fmeasures = []

            for label_idx in range(predictions.shape[0]):
                correct = correct_output[label_idx].cpu().view(-1).numpy()
                predicted = predictions[label_idx].cpu().view(-1).numpy()
                present_f1_score = f1_score(correct, predicted, average='binary', pos_label=1)
                present_precision_score = precision_score(correct, predicted, average='binary', pos_label=1)
                present_recall_score = recall_score(correct, predicted, average='binary', pos_label=1)

                precisions.append(present_precision_score)
                recalls.append(present_recall_score)
                fmeasures.append(present_f1_score)
                logging.info('F1\t%s\t%f' % (self.args['labels'][label_idx], present_f1_score))

            micro_f1 = f1_score(correct_output.reshape(-1).numpy(), predictions.reshape(-1).numpy(), average='micro')
            macro_f1 = f1_score(correct_output.reshape(-1).numpy(), predictions.reshape(-1).numpy(),
average='macro')

            if 'use_tensorboard' in self.args and self.args['use_tensorboard']:
                for label_idx in range(predictions.shape[0]):
                    self.tensorboard_writer.add_scalar('Precision/%s/Test' % self.args['labels'][label_idx].replace(" ", "_"),
precisions[label_idx], self.epoch)
                    self.tensorboard_writer.add_scalar('Recall/%s/Test' % self.args['labels'][label_idx].replace(" ", "_"),
recalls[label_idx], self.epoch)
                    self.tensorboard_writer.add_scalar('F1/%s/Test' % self.args['labels'][label_idx].replace(" ", "_"),
fmeasures[label_idx], self.epoch)
                self.tensorboard_writer.add_scalar('Micro-F1/Test', micro_f1, self.epoch)
                self.tensorboard_writer.add_scalar('Macro-F1/Test', macro_f1, self.epoch)

            with open(os.path.join(self.args['model_directory'], "eval_%s.csv" % self.epoch), 'w') as eval_results:
                eval_results.write('Metric\t' + '\t'.join([self.args['labels'][label_idx] for label_idx in
range(predictions.shape[0])]) +'\n' )
                eval_results.write('Precision\t' + '\t'.join([str(precisions[label_idx]) for label_idx in
range(predictions.shape[0])]) + '\n' )
                eval_results.write('Recall\t' + '\t'.join([str(recalls[label_idx]) for label_idx in range(predictions.shape[0])])
+ '\n' )
                eval_results.write('F1\t' + '\t'.join([ str(fmeasures[label_idx]) for label_idx in range(predictions.shape[0])])
+ '\n' )
                eval_results.write('Micro-F1\t' + str(micro_f1) + '\n' )
                eval_results.write('Macro-F1\t' + str(macro_f1) + '\n' )

        self.bert_doc_classification.train()

    def save_checkpoint(self, checkpoint_path: str):
        """
        Saves an instance of the current model to the specified path.
        :return:
        """
        if not os.path.exists(checkpoint_path):
            os.mkdir(checkpoint_path)
        else:
            raise ValueError("Attempting to save checkpoint to an existing directory")
        self.log.info("Saving checkpoint: %s" % checkpoint_path )

        #save finetune parameters
        net = self.bert_doc_classification
        if isinstance(self.bert_doc_classification, nn.DataParallel):
            net = self.bert_doc_classification.module
        torch.save(net.state_dict(), os.path.join(checkpoint_path, WEIGHTS_NAME))
        #save configurations
        net.config.to_json_file(os.path.join(checkpoint_path, CONFIG_NAME))
        #save exact vocabulary utilized
        self.bert_tokenizer.save_vocabulary(checkpoint_path)
```

e) Similarity estimation

```
#############################################################################################
######
# Programming language: Python
# Description: Provides a ranking of similar documents according to a vector representations.
# Input: a list of vectors.
# Output: list of similar vectors.
#############################################################################################
######

# Import libraries
import faiss
import math
import numpy as np
import pandas as pd
from sklearn.preprocessing import normalize


# Read VECTORS
train_vec = pd.read_table('data/rtord_vec.txt', delim_whitespace=True, header=None)
xb = train_vec.to_numpy().astype('float32') # Convert to array
xb = np.ascontiguousarray(xb) # Transform with this operation because it was giving an error.
d = 100            # dimension
xb = normalize(xb, axis=1, norm='l2')

# Example of a vector form.
print(xb[0:1])
[[ 0.14301404  0.17411718  0.00039257 -0.11284501  0.2616674   0.28336972
  -0.2148459  -0.13802391  0.20567684  0.27512777 -0.13181874  0.10296308
  -0.17787118  0.1345684  -0.17087588  0.04051801 -0.01847647 -0.00425969
  -0.03471071 -0.02843214  0.03463942  0.05297402  0.0276715  -0.00775349
  -0.1279664  -0.03411056  0.07524522 -0.0520996  -0.05577986  0.02421388
  -0.03763841  0.00938046 -0.02362673 -0.01913421 -0.055822    0.00487459
  -0.05375713 -0.03195367 -0.01429262  0.01252589 -0.07041313  0.13401005
   0.02516134  0.12824382  0.2058138  -0.08227916  0.1867839  -0.05888772
  -0.00113354  0.15429011 -0.18790762  0.06527199 -0.06849924  0.04962737
  -0.07879204  0.0244892  -0.06730878  0.04656867  0.07403369  0.01811968
  -0.02515713  0.04348891  0.15082055  0.11404952 -0.03084362  0.05174843
  -0.16937989 -0.02137608 -0.01360714  0.00182015 -0.0289919   0.08673903
   0.0152162  -0.02668788 -0.03831969  0.00378737  0.03792286  0.01373075
   0.01265161 -0.01787807  0.01951171  0.01286231  0.0562118   0.00988052
   0.04438088 -0.02346238  0.00570827  0.07870074 -0.16239864  0.04618239
   0.1350214  -0.0010809   0.1891929  -0.10092981 -0.05334626 -0.14973193
  -0.03131454  0.02555465 -0.05517234 -0.11232177]]


 # Build the index
index = faiss.IndexFlatL2(d)

 # Add vectors to the index
index.add(xb)
print(index.ntotal)

# Sanity check
k = 50                    # we want to see 50 nearest neighbors
```

```
D, I = index.search(xb[0:1], k) # sanity check
print(I)                        # Index
print(D)                        # distance of each index
last = xb.shape[0]

# Read all queries
D, I = index.search(xb[0:last], k) #
#print(I)                       # Index
#print(D)                       # distance of each index

# Change into one list
index_query = np.sort(np.array(list(np.arange(last))*k))
distances = np.concatenate(D, axis=0)
index_retrieved = np.concatenate(I, axis=0)

# Create a df with all information
pd.set_option("display.precision", 15)
distances_all = pd.DataFrame({"index_query": index_query, "distances": distances,
"index_retrieved":index_retrieved})
distances_all = distances_all.sort_values(by="distances")
distances_all_1 = distances_all.loc[distances_all.index_query!=distances_all.index_retrieved]
```

f) Baseline

```
#############################################################################################
######
# Programming language: Python
# Description: Provides a ranking of similar documents according to a vector representations.
# Input: a list of vectors.
# Output: list of similar vectors.
#############################################################################################
######

# Import libraries
import csv
import datetime
import nltk
import re
import pandas as pd
import numpy as np
from io import StringIO
from datetime import datetime
from sklearn.model_selection import train_test_split
from nltk.tokenize import RegexpTokenizer
from nltk.stem import WordNetLemmatizer,PorterStemmer
from nltk.corpus import stopwords
from unidecode import unidecode
np.random.seed(1337)

# Read data
data_input = pd.read_csv("data/rtord/process_all_rtord.csv", sep=";") # Processes
process["id"] = process.index
train = process
train = train[["petition_clean", "text", "sentenca"]]
train.columns = ["petition_clean", "text", "label"]
train = train[["text", "label"]]

# Preprocess text
train['cleanText']=train['text'].str.lower()
train['cleanText'] = train['cleanText'].apply(unidecode)
train['cleanText']=train['cleanText'].replace('{html}',"")
train['cleanText']=train['cleanText'].replace(r'[^A-Za-z0-9 ]+', ' ', regex=True)
train['cleanText']=train['cleanText'].replace(r'\d+',' ')
train['cleanText']=train['cleanText'].replace(r"http\S+", " ")
train['cleanText']=train['cleanText'].replace(r"\S*@\S*\s?", " ")
train['cleanText']=train['cleanText'].str.replace('\W', ' ')
train['cleanText']=train['cleanText'].replace('\s+', ' ', regex=True)
train_clean = train

# Remove sem resolucao
train_clean = train_clean[["text", "sentenca"]]
train_clean.columns = ["text", "label"]
train_clean = train_clean.loc[train_clean.label!="sem_resolucao"]

# Split samples into train and validation
train, val = train_test_split(train_clean, test_size=0.2, random_state=35)

# Reset indexs
```

```
train.reset_index(drop=True, inplace=True)
val.reset_index(drop=True, inplace=True)
train.shape, val.shape

# Using the split data with FastText
train_df = train
val_df = val

train_df["label"] = train_df.label.astype(str)
val_df["label"] = val_df.label.astype(str)

# Transform into the suitable form FastText
col = ['label', 'text']
train_df = train_df[col]
train_df['label']=['__label__'+ s for s in train_df['label']]
train_df['text']= train_df['text'].replace('\n',' ', regex=True).replace('\t',' ', regex=True)

col = ['label', 'text']
val_df = val_df[col]
val_df['label']=['__label__'+ s for s in val_df['label']]
val_df['text']= val_df['text'].replace('\n',' ', regex=True).replace('\t',' ', regex=True)

# Save output as desired to process on C++.
train_df.to_csv(r'data/rtord/not_clean/no_chunks/rtord_train.txt', index=False, sep=' ', header=False,
quoting=csv.QUOTE_NONE, quotechar="", escapechar=" ")
val_df.to_csv(r'data/rtord/not_clean/no_chunks/rtord_val.txt', index=False, sep=' ', header=False,
quoting=csv.QUOTE_NONE, quotechar="", escapechar=" ")

# Create a model
!fastText-0.9.2/fasttext supervised -input "data/rtord/not_clean/no_chunks/rtord_train.txt" -output
"data/rtord/not_clean/no_chunks/model_rtord_notclean_lr1_e4_n2" -lr 1 -epoch 4 -wordNgrams 2

# Metrics
!fastText-0.9.2/fasttext test "data/rtord/clean/no_chunks/model_rtord_lr1_e25_n2.bin"
"data/rtord/clean/no_chunks/rtord_val.txt"
!fastText-0.9.2/fasttext test "data/rtord/not_clean/no_chunks/model_rtord_notclean_lr1_e4_n2.bin"
"data/rtord/not_clean/no_chunks/rtord_val.txt"

# Get probabilities
!fastText-0.9.2/fasttext predict-prob "data/rtord/clean/no_chunks/model_rtord_lr1_e25_n2.bin"
"data/rtord/clean/no_chunks/rtord_val.txt" > "data/rtord/clean/no_chunks/probs_rtord_rtord_lr1_e25_n2.txt"
!fastText-0.9.2/fasttext predict-prob "data/rtord/not_clean/no_chunks/model_rtord_notclean_lr0.9_e5_n2.bin"
"data/rtord/not_clean/no_chunks/rtord_val.txt" >
"data/rtord/clean/no_chunks/probs_rtord_notclean_lr0.9_e5_n2.txt"

# Test with different parameters
parameters = [25, 50, 100, 200]

for i in parameters:
  a = str(i)
  dir_prob = "data/rtord/clean/no_chunks/probabilities/probs_lr_1_e" + a + "n2.txt"
  !fastText-0.9.2/fasttext supervised -input "data/rtord/clean/no_chunks/rtord_train.txt" -output
"data/rtord/clean/no_chunks/model" -lr 0.1 -epoch $i -wordNgrams 2
# Create a model
!fastText-0.9.2/fasttext predict-prob "data/rtord/clean/no_chunks/model.bin"
"data/rtord/clean/no_chunks/rtord_val.txt" > $dir_prob
```

```
names = ["model_rtord_notclean_lr0.1_e5_n2", "model_rtord_notclean_lr0.5_e5_n2",
"model_rtord_notclean_lr0.8_e5_n2", "model_rtord_notclean_lr0.9_e5_n2", "model_rtord_notclean_lr1_e4_n2",
"model_rtord_notclean_lr1_e5_n5", "model_rtord_notclean_lr1_e7_n2", "model_rtord_notclean_lr1_e10_n2",
"model_rtord_notclean_lr1_e15_n2", "model_rtord_notclean_lr1_e20_n2", "model_rtord_notclean_lr1_e25_n2",
"model_rtord_notclean_lr1.2_e5_n2", "model_rtord_notclean_lr1.5_e5_n2", "model_rtord_notclean_lr2_e5_n2"]

for i in names:
  dir_model = "data/rtord/not_clean/no_chunks/" + i +".bin"
  dir_prob = "data/rtord/not_clean/no_chunks/probabilities/"+ i + ".txt"

!fastText-0.9.2/fasttext predict-prob $dir_model "data/rtord/not_clean/no_chunks/rtord_val.txt" > $dir_prob

# Extract vectors
!fastText-0.9.2/fasttext print-sentence-vectors "data/rtord/clean/no_chunks/model_rtord_lr1_e25_n2.bin" <
"data/rtord/clean/no_chunks/rtord_train.txt" > "data/rtord/clean/no_chunks/rtord_vec.txt" # Train
!fastText-0.9.2/fasttext print-sentence-vectors "data/rtord/clean/no_chunks/model_rtord_lr1_e25_n2.bin" <
"data/rtord/clean/no_chunks/rtord_val.txt" > "data/rtord/clean/no_chunks/rtord_val_vec.txt" # Val

# Read vectors
train_vec = pd.read_table('data/rtord/clean/chunks/rtord_vec.txt', delim_whitespace=True, header=None)
tr_emb = train_vec.to_numpy() # Convert to array
test_vec = pd.read_table('data/rtord/clean/chunks/rtord_val_vec.txt', delim_whitespace=True, header=None)
val_emb = test_vec.to_numpy()
tr_emb.shape, val_emb.shape

# Label to numeric
train.loc[train.label=="improcedente", "label"] = 0
train.loc[train.label=="procedente_em_parte", "label"] = 1
val.loc[val.label=="improcedente", "label"] = 0
val.loc[val.label=="procedente_em_parte", "label"] = 1
```

g) Statistics measurements

```
##############################################################################################
######
# Programming language: Python
# Description: Estimate statistics from FastText output
# Input: an array of results estimated by FastVector with its corespondent true result
# Output: an array of the statistics: Acuracy, Precision, Recall, F1, TP, TN, FP, FN.
##############################################################################################
######

# Import libraries
import pandas as pd
import numpy as np
from os import listdir
from os.path import isfile, join
from sklearn.metrics import precision_recall_fscore_support as score

# Read all probabilities from fast text
mypath = '’ # Set path
onlyfiles = [f for f in listdir(mypath) if isfile(join(mypath, f))]
#true_val = pd.read_table('data/rtord/not_clean/no_chunks/df_val.csv', sep=";")

# One sample, to test
true_val = pd.read_table('data/rtord/clean/no_chunks/df_val.csv', sep=";")
val_prob = pd.read_table('data/rtord/clean/no_chunks/probabilities/probs_lr_1_e20n2.txt', delim_whitespace=True,
header=None)

data = pd.concat([true_val.reset_index(drop=True), val_prob], axis=1)
data.loc[data[0]== "__label__improcedente", 0 ] = 0
data.loc[data[0]== "__label__procedente_em_parte", 0 ] = 1
data.columns = ["text", "padrao_ouro", "algo_result", "probability"]
data.loc[data["padrao_ouro"]== "improcedente","padrao_ouro"] = 0
data.loc[data["padrao_ouro"]== "procedente_em_parte","padrao_ouro"] = 1
data.loc[data["algo_result"]== 0,"probability"] = 1 - data["probability"]
data_1 = data

pd.options.display.max_colwidth = 100
procedente = data_1.sort_values('probability', ascending=False)
procedente.loc[procedente.probability < 0.98][0:10]

# Style sklearn
data_results=pd.DataFrame()
for i in range(len(onlyfiles)):
  print(onlyfiles[i])
  file_to_read = "./data/rtord/not_clean/no_chunks/probabilities/" + onlyfiles[i]

  # Estimate from FastText
  val_prob = pd.read_table(file_to_read, delim_whitespace=True, header=None)

  # Arrange data
  data = pd.concat([true_val.reset_index(drop=True), val_prob], axis=1)
  data.loc[data[0]== "__label__improcedente", 0 ] = 0
  data.loc[data[0]== "__label__procedente_em_parte", 0 ] = 1
  data.columns = ["text", "padrao_ouro", "algo_result", "probability"]
  data.loc[data["padrao_ouro"]== "improcedente","padrao_ouro"] = 0
  data.loc[data["padrao_ouro"]== "procedente_em_parte","padrao_ouro"] = 1
  data.loc[data["algo_result"]== 0,"probability"] = 1 - data["probability"]
  data_1 = data

  # Sklearn style
  predicted = data_1['algo_result'].tolist()
  y_test = data_1['padrao_ouro'].tolist()

  precision, recall, fscore, support = score(y_test, predicted)
  results_list = [precision, recall, fscore, support]
  results_1 = pd.DataFrame(results_list).T
  results_1.columns = ["precision", "recall", "fscore", "support"]
  results_1["Macrof"] = results_1["fscore"].mean()
  results_1["label"] = results_1.index
  results_1["id"] = onlyfiles[i]
  data_results=data_results.append(results_1, ignore_index=True)

data_results = data_results.sort_values('fscore', ascending=False)
data_results

data_results.to_csv("./data/rtord/not_clean/no_chunks/results/results_not_clean_no_chunks.csv", sep=";",
index=False)

#data_results

# List to store results
Treshold_results = []
Total_results = []
TP_results = []
TN_results = []
FP_results = []
FN_results = []

# Iterate over all posible values
prob_1 = data_1['probability'].tolist()

for i in prob_1:
    data_1.loc[(data_1["probability"] >= i), "algo_result"] = 1
    data_1.loc[(data_1["probability"] < i), "algo_result"] = 0

    # Estimate statistics
    data_1['TP'] = np.where((data_1["algo_result"]==1) & (data_1["padrao_ouro"]==1), 1,0)
    data_1['TN'] = np.where((data_1["algo_result"]==0) & (data_1["padrao_ouro"]==0), 1,0)
    data_1['FP'] = np.where((data_1["algo_result"]==1) & (data_1["padrao_ouro"]==0), 1,0)
    data_1['FN'] = np.where((data_1["algo_result"]==0) & (data_1["padrao_ouro"]==1), 1,0)

    # Estimate measures
    data_1 = data_1.fillna(0)
    TP =  data_1["TP"].sum()
    TN =  data_1["TN"].sum()
    FP =  data_1["FP"].sum()
    FN =  data_1["FN"].sum()
    Total = TP + TN + FP + FN

    # Apend results
    Treshold_results.append(i)
    Total_results.append(Total)
    TP_results.append(TP)
    TN_results.append(TN)
    FP_results.append(FP)
    FN_results.append(FN)

# Create df to estimate metrics
metrics = pd.DataFrame(list(zip( Treshold_results, Total_results, TP_results, TN_results, FP_results, FN_results)),
                  columns =['Treshold', "Total", 'TP', 'TN', 'FP', 'FN'])

# Measures
metrics["Total"] = metrics.TP + metrics.TN + metrics.FP + metrics.FN
metrics["Acuracy"] = (metrics.TP + metrics.TN) / metrics.Total
metrics["Precision"] = metrics.TP / (metrics.TP + metrics.FP)
metrics["Recall"] = metrics.TP / (metrics.TP + metrics.FN)
metrics["Total_N"] = metrics["TN"] + metrics["FN"]
metrics["Total_P"] = metrics["TP"] + metrics["FP"]
metrics["F1"] = metrics.TP / (metrics.TP + (0.5*(metrics.FP + metrics.FN)))
metrics = metrics.sort_values('Acuracy', ascending=False)metrics.to_csv('data/rtord/clean/no_chunks/metrics.csv',
sep =";", index=False, float_format='%.3f', decimal= ",")
```

Attachment 2

Year	Authors	Tittle
2020	Felipe Maia Polo, Itamar Ciochetti, Emerson Bertolo	Predicting Legal Proceedings Status: an Approach Based on Sequential Text Data
2020	Adrien Bibal, Michael Lognoul, Alexandre de Streel, Benoît Frénay	Legal requirements on explainability in machine learning
2020	Philipp Hacker, Ralf Krestel, Stefan Grundmann, Felix Naumann	Explainable AI under contract and tort law: legal incentives and technical challenges
2020	Vu Tran, Minh Le Nguyen, Satoshi Tojo, Ken Satoh	Encoded summarization: summarizing documents into continuous vector space for legal case retrieval
2019	Sandbhor, Sayali and Chaphalkar, NB	Impact of Outlier Detection on Neural Networks Based Property Value Prediction
2019	Rupali Sunil Wagh and Deepa Anand	A Novel Approach of Augmenting Training Data for Legal Text Segmentation by Leveraging Domain Knowledge
2019	Keet et al.	Legal Document Retrieval Using Document Vector Embeddings and Deep Learning
2019	Deepa Ananda, Rupali Waghb	Effective deep learning approaches for summarization of legal texts
2019	Arunprasath Shankar, Venkata Nagaraju Buddarapu	Legal Query Reformulation using Deep Learning
2019	Ge Yan, Yu Li, Siyuan Shen, Shu Zhang, Jia Liu	Law Article Prediction Based on Deep Learning
2019	Baogui Chen, Yu Li, Shu Zhang, Hao Lian, Tieke He	A Deep Learning Method for Judicial Decision Support
2019	Eya Hammami, Imen Akermi, Rim Faiz, Mohand Boughanem	Deep Learning for French Legal Data Categorization
2019	William Paulo Ducca Fernandes et al.	Appellate Court Modifcations Extraction for Portuguese
2019	Marco Lippi et. al.	CLAUDETTE: an automated detector of potentially unfair clauses in online terms of service
2018	Chalkidis, Ilias and Androutsopoulos, Ion and Michos, Achilleas	Obligation and Prohibition Extraction Using Hierarchical RNNs
2018	Abood, Aaron and Feltenberger, Dave	Automated patent landscaping
2018	Nguyen, Truong-Son and Nguyen, Le-Minh and Tojo, Satoshi and Satoh, Ken and Shimazu, Akira	Recurrent neural network-based models for recognizing requisite and effectuation parts in legal texts
2018	Sadeghian et. al.	Automatic semantic edge labeling over legal citation graphs
2018	Nanda, Rohan et. al.	Unsupervised and supervised text similarity systems for automated identification of national implementing measures of European directives
2018	Chalkidis, Ilias and Kampas, Dimitrios	Deep learning in law: early adaptation and legal word embeddings trained on large corpora
2018	Undavia, Samir and Meyers, Adam and Ortega, John E	A Comparative Study of Classifying Legal Documents with Neural Networks
2018	Kowsrihawat, Kankawin and Vateekul, Peerapon and Boonkwan, Prachya	Predicting Judicial Decisions of Criminal Cases from Thai Supreme Court Using Bi-directional GRU with Attention Mechanism
2018	Ma, Yong and Zhang, Rongxia and Zong, Tongqiang	Law, discipline and governance based on cognitive process simulation–analysis of legal forms in the modern rights relationship
2018	Mironczuk, Marcin Michal and Protasiewicz, Jaroslaw	A recent overview of the state-of-the-art elements of text classification
2018	Branting, L Karl and Yeh, Alexander and Weiss, Brandy and Merkhofer, Elizabeth and Brown, Bradford	Inducing Predictive Models for Decision Support in Administrative Adjudication
2018	Rishi Chhatwal and Peter Gronvall and Nathaniel Huber-Fliflet and Robert Keeling and Jianping Zhang and Haozhen Zhao	Explainable Text Classification in Legal Document Review. A Case Study of Explainable Predictive Coding
2018	Fusheng Wei and Han Qin and Shi Ye and Haozhen Zhao	Empirical Study of Deep Learning for Text Classification in Legal Document Review
2018	Lopes, Susana Almeida and Duarte, Maria Eduarda and Almeida Lopes, João	Can artificial neural networks predict lawyers' performance rankings?
2018	Mi-Young Kim and Yao Lu and Randy Goebel	Textual Entailment in Legal Bar Exam Question Answering Using Deep Siamese Networks
2018	Singh, Jaspreet and Sharma, Yashvardhan	Encoder-Decoder Architectures for Generating Questions
2018	Keet Sugathadasa et. Al	Legal Document Retrieval Using Document Vector Embeddings and Deep Learning
2018	Li, Shaobo and Hu, Jie and Cui, Yuxin and Hu, Jianjun	DeepPatent: patent classification with convolutional neural networks and word embedding
2018	Dirk Helbing	Societal, Economic, Ethical and Legal Challenges of the Digital Revolution: From Big Data to Deep Learning, Artificial Intelligence, and Manipulative Technologies
2018	Fusheng Wei, Han Qin, Shi Ye, Haozhen Zhao	Empirical Study of Deep Learning for Text Classification in Legal Document Review
2019	Neha Bansal, Arun Sharma, and R. K. Singh	A Review on the Application of Deep Learning in Legal Domain
2018	N. Correia da Silva	Document type classification for Brazil's supreme court using a convolutional neural network
2018	Shang Li, Hongli Zhang, Lin Ye, Xiaoding Guo, Binxing Fang	Evaluating the rationality of judicial decision with LSTM-based case modeling
2018	Ahmed Elnaggar, Christoph Gebendorfer, Ingo Glaser and Florian Matthes	Multi-Task Deep Learning for Legal Document Translation, Summarization and Multi-Label Classification
2017	Mandal, Arpan and Chaki, Raktim and Saha, Sarbajit and Ghosh, Kripabandhu and Pal, Arindam and Ghosh, Saptarshi	Measuring similarity among legal court case documents
2017	Brozek, Bartosz and Jakubiec, Marek	On the legal responsibility of autonomous machines
2017	Branting, L Karl	Data-centric and logic-based models for automated legal problem solving
2017	Do, Phong-Khac and Nguyen, Huy-Tien and Tran, Chien-Xuan and Nguyen, Minh-Tien and Nguyen, Le-Minh	Legal question answering using ranking SVM and deep convolutional neural network
2017	Ayaka Morimoto and Daiki Kubo and Motoki Sato and Hiroyuki Shindo and Yuji Matsumoto	Legal Question Answering System using Neural Attention
2017	Nanda, Rohan and Adebayo, Kolawole John and Di Caro, Luigi and Boella, Guido and Robaldo, Livio	Legal information retrieval using topic clustering and neural networks
2017	Mortazavi, Melissa	Rulemaking Ex Machina
2017	Isar, Nejadgholi and Renaud, Bougueng and Samuel, Witherspoon	A Semi-Supervised Training Method for Semantic Search of Legal Facts in Canadian Immigration Cases
2017	O'Neill, James and Buitelaar, Paul and Robin, Cecile and O'Brien, Leona	Classifying sentential modality in legal language: a use case in financial regulations, acts and directives
2017	Chalkidis, Ilias and Androutsopoulos, Ion and Michos, Achilleas	Extracting contract elements
2017	Alschner, Wolfgang and Skougarevskiy, Dmitriy	Towards an automated production of legal texts using recurrent neural networks
2017	Adebayo, Kolawole John and Luigi Di Caro and Livio Robaldo and Guido Boella	Legalbot: A Deep Learning-Based Conversational Agent in the Legal Domain
2017	Nanda, Rohan and Adebayo, Kolawole John and Di Caro, Luigi and Boella, Guido and Robaldo, Livio	Legal information retrieval using topic clustering and neural networks
2017	Chalkidis, Ilias and Androutsopoulos, Ion	A Deep Learning Approach to Contract Element Extraction
2017	ASRA, FATIMA and SESHADRI, SEKHAR T. and KRISHNA, PRASAD BELLAM SIVARAMA	A Comprehensive Study of Dispute Resolution using Artificial Neural Network in Build Operate and Transfer (BOT) Project
2017	Phong-Khac Do, Huy-Tien Nguyen, Chien-Xuan Tran, Minh-Tien Nguyen, Minh-Le Nguyen	Legal question answering using ranking SVM and deep convolutional neural network
2016	John J. Nay	Gov2Vec: Learning Distributed Representations of Institutions and Their Legal Text
2016	Gurkaynak, Gonenc and Yilmaz, Ilay and Haksever, Gunes	Stifling artificial intelligence: Human perils
2016	Tang, G and Guo, H and Guo, Z and Xu, S	Matching law cases and reference law provision with a neural attention model
2016	Son, Nguyen Truong and Nguyen, Le Minh and Quoc, Ho Bao and Shimazu, Akira	Recognizing logical parts in legal texts using neural architectures
2016	Wolfgang Alschner and Dmitriy Skougarevskiy	Can Robots Write Treaties? Using Recurrent Neural Networks to Draft International Investment Agreements.
2016	Sadeghian, Ali and Sundaram, Laksshman and Wang, D and Hamilton, William F and Branting, L Karl and Pfeifer, Craig	Semantic edge labeling over legal citation graphs
2015	El Jelali, Soufiane and Fersini, Elisabetta and Messina, Enza	Legal retrieval as support to eMediation: matching disputant's case and court decisions
2015	Cerka, Paulius and Grigiene, Jurgita and Sirbikyte, Gintare	Liability for damages caused by artificial intelligence
2015	Chaphalkar, NB and Iyer, KC and Patil, Smita K	Prediction of outcome of construction dispute claims using multilayer perceptron neural network model
2015	Chaphalkar, NB and Sandbhor, Sayali S	Application of neural networks in resolution of disputes for escalation clause using neuro-solutions
2015	Sharma, Ranti Dev and Mittal, Sudhanshu and Tripathi, Samarth and Acharya, Shrinivas	Using Modern Neural Networks to Predict the Decisions of Supreme Court of the United States with State-of-the-Art Accuracy
2015	Kim, Mi-Young and Xu, Ying and Goebel, Randy	A Convolutional Neural Network in Legal Question Answering
2014	Garcez, Artur S d'Avila and Gabbay, Dov M and Lamb, Luis C	A neural cognitive model of argumentation with application to legal inference and decision making
2014	Chaphalkar, NB and Sandbhor, Sayali S	Application of neural networks in resolution of disputes for escalation clause using neuro-solutions
2012	Thammaboosadee, Sotarat and Watanapa, Bunthit and Charoenkitkarn, Nipon	A framework of multi-stage classifier for identifying criminal law sentences
2011	Yang, Rui and Olafsson, Sigurdur	Classification for predicting offender affiliation with murder victims
2011	Theresa, M.M. Janeela and Raj, V. Joseph	Analogy making in criminal law with neural network
2010	Ibrahin Yitmen and Ebrahim Soujeri	An Artificial Neural Network model for estimating the influence of change orders on project performance and dispute resolution
2009	Lai, Yi-Hsuan and Che, Hui-Chung	Modeling patent legal value by Extension Neural Network
2008	Lai, Yi-Hsuan and Che, Hui-Chung and Wang, Szu-Yi	Managing Patent Legal Value via Fuzzy Neural Network incorporated with Factor Analysis
2007	Chau, K. W.	Application of PSO-based neural network in analysis of outcomes of construction claims
2006	Oatley, Giles and Ewart, Brian and Zeleznikow, John	Decision support systems for police: Lessons from the application of data mining techniques to soft forensic evidence
2006	Trappey, Amy J.c. and Hsu, Fu-Chiang and Trappey, Charles V. and Lin, Chia-I.	Development of a patent document classification and search platform using a back-propagation network
2006	Stranieri, Andrew and Zeleznikow, John	Knowledge discovery from legal databases using neural networks and data mining to build legal decision support systems
2005	A.J.C. Trappey and S.C.I. Lin and A.C.L. Wang	Using neural network categorization to develop an innovative knowledge management technology for patent document classification
2004	Dikmen, I., Birgonul, M.T	Neural network model to support international market entry decisions
2003	Dahbur, Kamal and Muscarello, Thomas	Classification system for serial criminal patterns
2003	Corcoran, Jonathan J. and Wilson, Ian D. and Ware, J.Andrew	Predicting the geo-temporal variations of crime and disorder
2003	Borges, Filipe and Borges, Raoul and Bourcier, Daniele	Artificial neural networks and legal categorization
2002	Borges, Filipe and Borges, Raoul and Bourcier, Daniele	A connectionist model to justify the legal reasoning of the judge
2002	Artur D 'Avila et al.	Neural-Symbolic Learning and Reasoning: Contributions and Challenges
2001	Moens, Marie-Francine	Innovative techniques for legal text retrieval
2001	Corcoran, Jonathan J. and Wilson, Ian D. and Lewis, Owen M. and Ware, J. Andrew	Data Clustering and Rule Abduction to Facilitate Crime Hot Spot Prediction
2001	Adderley, Richard and Musgrove, Peter B.	Data mining case study: Modeling the behavior of offenders who commit serious sexual assaults
2001	Chen, Daqing and Burrell, Phillip	Case-Based Reasoning System and Artificial Neural Networks: A Review
2001	Zhi-Wei, Ni and Qing-Sheng, Cai and Long-Shu, Li	Data mining and neural network techniques in case based system
2000	Cheung, Sai On and Tam, CM and Harris, FC	Project dispute resolution satisfaction classification through neural network
1999	Borgulya, István	Two examples of decision support in the law
1999	Hollatz, Jurgen	Analogy making in legal reasoning with neural networks and fuzzy logic
1999	Stranieri, Andrew and Zeleznikow, John and Gawler, Mark and Lewis, Bryn	A hybrid rule–neural approach for the automation of legal reasoning in the discretionary domain of family law in Australia
1999	Merkl, Dieter and Schweighofer, Erich and Winiwarter, Werner	Exploratory analysis of concept and document spaces with connectionist networks
1999	Bourcier, Danile and Clergue, Gerard	From a rule-based conception to dynamic patterns. Analyzing the self-organization of legal systems
1999	Philipps, Lothar and Sartor, Giovanni	Introduction: from legal theories to neural networks and fuzzy reasoning
1999	Hunter, Dan	Out of their minds: Legal theory in neural networks
1999	C. K. Shiu Simon, C. C Eric, Tsang, Daniel. S. Yeung	Maintaining Case – Based Expert Systems using Fuzzy Neural Network
1998	Sartor, Giovanni and Branting, L Karl	Introduction: judicial applications of artificial intelligence
1998	Tata, Cyrus	The application of judicial intelligence and rules' to systems supporting discretionary judicial decision-making
1998	Arditi, David and Oksay, Fatih E and Tokdemir, Onur B	Predicting outcome of construction litigation using neural network
1997	Olligschlaeger, Andreas M	Artificial Neural Networks and Crime Mapping
1997	Merkl, Dieter and Schweighofer, Erich	The Exploration of Legal Text Corpora with Hierarchical Neural Networks: A Guided Tour in Public International Law
1997	D.K.H. Chua and P.K. Loh and Y.C. Kog and E.J. Jaselskis	Neural networks as tools in construction.
1996	Aikenhead, Michael	The uses and abuses of neural networks in law
1995	Robert E. Macneel	Technology Report: Intelligent Summoner
1995	Merkl, Dieter	A connectionist view on document classification
1995	Merkl, Dieter	Content-based document classification with highly compressed input data
1995	Merkl, Dieter and Schweighofer, E and Winiwater, W	Analysis of Legal Thesauri Based on Self-organising Feature Maps
1995	Egri, PETER A and Underwood, PETER F	HILDA: Knowledge extraction from neural networks in legal rule based and case based reasoning
1995	Hollatz, Jurgen	Neuro-fuzzy in legal reasoning
1995	By Wullianallur Raghupathi and Lawrence L. Schkade and Raju S. Bapi and Daniel S. Levine	Neural networks — Towards predictive law machines
1995	Winiwarter, Werner and Schweighofer, Erich and Merkl, Dieter	Knowledge Acquisition in Concept and Document Spaces by Using Self-organizing Neural Networks
1994	Hunter, Dan	Looking for law in all the wrong places: Legal theory and legal neural networks
1994	Hobson, John B. and Slee, David	Indexing the Theft Act 1968 for case based reasoning and artificial neural networks
1993	Zeleznikow, John and Vossos, George and Hunter, Dan	The IKBALS project: Multi-modal reasoning in legal knowledge based systems
1993	Crombag, Hans FM	On the artificiality of artificial intelligence
1993	Bench-Capon, Trevor	Neural networks and open texture
1993	Warner Jr, David R	A neural network-based law machine: The problem of legitimacy
1993	Hobson, John B. and Slee, David	Rules, cases and networks in a legal domain
1992	Widdison, Robin and Pritchard, Francis and Robinson, William	The European conflicts guide
1992	Mital, V and Gedeon, TD	A neural network integrated with hypertext for legal document assembly
1992	Warner Jr, David R	A Neural Network-Based Law Machine: Initial Steps
1991	By Wullianallur Raghupathi and Lawrence L. Schkade and Raju S. Bapi and Daniel S. Levine	Exploring connectionist approaches to legal decision making
1991	Philipps, Lothar	Distribution of damages in car accidents through the use of neural networks
1991	Thagard, Paul	Connectionism and legal inference
1991	Van Opdorp, GJ and Walker, RF and Schrickx, JA and Groendijk, Cees and Van den Berg, PH	Networks at work: a connectionist approach to non-deductive legal reasoning
1991	Bochereau, Laurent and Bourcier, Danièle and Bourgine, Paul	Extracting legal knowledge by means of a multilayer neural network application to municipal jurisprudence
1991	Walker, RF and Oskamp, Anja and Schrickx, JA and Van Opdorp, GJ and Van Den Berg, PH	PROLEXS: Creating law and order in a heterogeneous domain
1991	Rose, Daniel E and Belew, Richard K	A connectionist and symbolic hybrid for improving legal research
1990	Donald L. Wenskay	Intellectual property protection for neural networks
1990	Warner Jr, David R	The Role of Neutral Networks in the Law Machine Development
1989	Philipps, Lothar	Are Legal Decisions based on the Application of Rules or Prototype Recognition?
1989	Rose, Daniel E and Belew, Richard K	Legal information retrieval a hybrid approach
1989	Philipps, Lothar	A Neural Network to Identify Legal Precedents
1987	R. K. Belew	A connectionist approach to conceptual information retrieval