Applied Soft Computing 124 (2022) 108987

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Weighting construction by bag-of-words with similarity-learning and supervised training for classification models in court text documents

Antonio P. Castro Junior [a,d,*], Gabriel A. Wainer [c], Wesley P. Calixto [a,b]

[a] Electrical, Mechanical & Computer Engineering School, Federal University of Goias, Goiania, Goias, Brazil
[b] Experimental & Technological Research and Study Group, Federal Institute of Goias, Goiania, Goias, Brazil
[c] Visualization, Simulation and Modeling, Carleton University, Ottawa, Canada
[d] Court of Justice, Goias, Brazil

ARTICLE INFO

Article history:
Received 11 July 2021
Received in revised form 18 April 2022
Accepted 5 May 2022
Available online 14 May 2022

Keywords:
Artificial intelligence
Similarity-learning
Text classification
Machine learning
Knowledge management

ABSTRACT

Traditional models of bag-of-words for text classification are unable to identify weights for the co-occurrence of terms, and, mainly, for this reason, they are being replaced by models of word embedding. This article proposes a method to enhance traditional bag-of-words models in two aspects: (a) build features on the co-occurrence of terms and (b) smooth the non-linearity or make the terms linear for different *corpus* categories. The datasets used are characterized by the non-linearity of the terms, having four different categories of documents. Two computational representations of the datasets are generated: binary and frequency, being used for supervised training of nine classification technologies: random forest, multilayer perceptron neural networks, adaptive boosting, gradient boosting, Gaussian process, support vector machine, Naive Bayes, *k*-nn and decision trees, its results are compared with nine other algorithms used in other research work. The combinations of each obtained result are compared and assessed using the accuracy, f-measure, precision, and recall metrics. The research and studies generated resulted in the construction of an API that will integrate the Department of Justice software that controls the judicial proceedings. The results of the evaluation metrics and the comparisons with other studies demonstrate that the proposed methodology is feasible to be applied in meeting the needs of the court, allowing to speed up the judgment of lawsuits.

Code metadata

Permanent link to reproducible Capsule: https://doi.org/10.24433/CO.5422851.v1

1. Introduction

There are several areas of expertise working on document classification using machine learning, such as (i) medicine [1], (ii) biology [2], (iii) engineering [3], (iv) law [4], (v) education [5], among others. Information retrieval innovations use techniques to define document type in a *corpus*, with automatic knowledge generation and information handling [3]. Many studies are moving in this direction, like (i) Ceci and Gangemi [6], (ii) Fawei et al. [7], (iii) Calambás et al. [8], (iv) Zhang et al. [9], (v) Ni et al. [4] continue the work of Zhang et al. [9], (vi) Rani et al. [3], (vii) Huang et al. [10], (viii) Wu and Zhang [11], (ix) Agarwal et al. [12], (x) Seo et al. [13], (xi) Li et al. [14], and (xii) Abualigah et al. [15].

In the literature, several works on the classification of text documents and vectorization of terms apply the concept of bag-of-words [16–20], others bag-of-concepts [14,21–24], and others apply both solutions (bag-of-words and bag-of-concepts) [10,14,25]. In the bag-of-words model, the document is transformed into a vector of size n, in which n is the number of words used to represent the document, each vector field is associated with the word value, calculated by traditional methods such as *tf*, *tf-idf*, *Okapi BM25*, and others. In the bag-of-concepts model, also known as word embeddings, the document is transformed into a vector space of size $n \times d$, in which n is the number of words and d the number of dimensions of the word embeddings, considering that the dimension d of a word is generally defined by the own word and by the words that accompany the word in

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: Electrical, Mechanical & Computer Engineering School, Federal University of Goias, Goiania, Goias, Brazil.
E-mail address: apcastro@tjgo.jus.br (A.P. Castro Junior).

https://doi.org/10.1016/j.asoc.2022.108987
1568-4946/

the text, creating a co-occurrence among the words. Also, if the representation of the document is separated by its sentences, the vector space can be expanded to $s \times n \times d$, where s is the number of sentences in the document. Methods known and applied in the construction of words embeddings are word2Vec [21], GloVe [26], ELMo [27], BERT [28], and others. After obtaining the document attributes, through the one-dimensional model, bag-of-words, or the multi-dimensional model, bag-of-concepts, it is possible to train different machine learning algorithms to classify documents.

Recent works as Agarwal et al. [12], Seo et al. [13], and Li et al. [14] inform the disadvantages of using the bag-of-words model, reaffirm the limitations in applying the model *tf-idf*, demonstrating that model cannot relate to the co-occurrence among the terms in the documents and value the techniques of the bag-of-concepts. Agarwal et al. [12] compare two solutions of vectorization, *tf-idf* and *lfw*, in web services documents, in the WSDL format, then apply *k*-means to group them. Seo et al. [13] use the *tf-idf* model to identify unusual terms in the customer's answers database. Li et al. [14] propose a new way to apply the bag-of-concepts, using an external knowledge base and multidimensional analysis of the text, generating an additional dimension based on external knowledge. However, the best results in Li et al. [14] were obtained when applying the bag-of-concepts model proposed with the bag-of-words. It is observed that each scenario or area of expertise has characteristics in its documents (*corpus*) whether or not they require high processing costs or the use of complex techniques. Thus, it is noticed that simple solutions with less processing time can achieve the objective of the application [29,30]. Gomaa and Fahmy [31] inform that some weight construction models are traditionally researched and applied, as term frequency (*tf*), term frequency–inverse document frequency (*tf-idf*), and Okapi best matching 25 (*BM25*).

This article seeks to address two main gaps: (a) limitations in the application of the model *tf-idf*, due to not being able to build features for the co-occurrence between similar terms and (b) divergent judgments occur for lawsuits with similar histories and facts, which can lead to inequalities and even possible discrimination. Thus, this paper proposes a methodology to improve the application of the bag-of-words model in the classification of documents and vectorization of terms, for it can generate co-occurrence among the terms of a document in a single vector of size n, it is obtained, currently, by the techniques of word embeddings, in which a multi-dimensional vector space represents the document. The proposed method enhances the application of the *tf-idf* model when in conjunction with the *tf-idf* model, adding similarity learning between the terms of the documents. This joint application **innovates** the bag-of-words model in three aspects: (a) after establishing the values of the terms through the model *tf-idf*, applies the Jaccard similarity to relate the co-occurrence of the terms in the documents, solving the limitation informed in the works of Agarwal et al. [12], Seo et al. [13], and Li et al. [14], (b) the co-occurrence weighting values of similar terms are inserted into a single vector of size n, and (c) it softens the nonlinearity of the terms in the classification of documents in text. The **innovations** are noticed and presented in the obtained results, as in the application in the corpus of documents of the Court of Justice, as much as in the traditional datasets used in other studies, comparing the values in the accuracy, f-measure, precision, and recall metrics.

This work presents an artificial intelligence method that performs the recognition of document patterns using similarity-learning in the construction of features, in conjunction with *tf-idf* model to identify documents by their co-occurrence of the terms. The knowledge obtained by the similarity-learning are inserted in nine classification technologies: random forest, multilayer perceptron with backpropagation neural network (MLPNN), adaptive boosting, gradient boosting, Gaussian process, support vector machine, Naive Bayes, *k*-nearest neighbors and decision trees, enabling the making of taxonomic forecasts for new documents. These nine classification algorithms are tested and evaluated. The *Okapi BM25* model is also applied with the similarity-learning in order to compare its results with *tf-idf*.

The studies and researches carried out in this article are in the application in the *corpus* of lawsuits documents in the governmental institution of justice in Brazil, the state of Goias. The data used in this research are documents of a judgment of judges, known as decisions. These documents describe the event that occurred, covering all aspects of the law inherent in the case and solicitations by lawyers. They are documents rich in information, integrating generated facts and applied laws. These decisions shape the way the law is interpreted and applied by the many attorneys and access professionals across the government [29,30, 32,33].

The **relevance** of this research is to provide celerity in the judgments of the judicial proceedings, as well as to reduce inequalities and discrimination, since it identifies and classifies the received lawsuits, allowing connection with cases already judged. To **apply the proposed method** in this study, an API was created, allowing to insert the functionality of text classification on the Court of Justice software. In addition to the **contribution to society** applied to the Department of Justice, this article brings five **relevant contributions** to the academic environment: (a) it evaluates different training techniques, binary and frequency, as well as nine classification algorithms that present better accuracy, f-measure, precision, and recall metrics for the court's unstructured text documents, (b) it compares and evaluates the proposed method with other research works, applying other datasets, (c) it compares and evaluates the application of the *tf-idf* and *BM25* model, together with the Jaccard similarity, (d) it builds and makes the source code available on Github to perform preprocessing on justice text documents (in Portuguese), and (e) datasets generated and the implementation of the nine classification algorithms can be found on Github, for the evolution of current research. The results found have a low processing time in learning the proposed method and predicting new documents. Its application is not onerous in a production environment.

By way of comparing and verifying the achieved results in this paper, similar studies as in Abualigah et al. [34], are faced and presented in the results section. Abualigah et al. [34] present and assess algorithms used to solve the problem of text classification and grouping. Experiments in Abualigah et al. [34] were recorded, and their accuracy, f-measure, precision, and recall results were assessed in the optimization algorithms to solve problems of clustering, including Harmony Search (HS) Algorithm, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Krill Herd Algorithm (KHA), Cuckoo Search (CS) Algorithm, Gray Wolf Optimizer (GWO), Bat-inspired Algorithm (BA), and *k*-means. Abualigah et al. [34] use datasets freely available on the Internet to validate the performance of the tested algorithms. The Gray Wolf Optimizer (GWO) algorithm presented the best results in the assessment metrics in Abualigah et al. [34]. Two datasets used in the study of Abualigah et al. [34] were operated in this article, allowing the comparison of the results of the assessment metrics between the two types of research.

The **novelties** (**originality**) inherent to this applied research and not noticed by authors in other works are: (a) in the academic context, the proposed model improves the traditional BOW with the joint application of the similarity-learning technique to automatically find the co-occurrence of terms, allowing the vectorization of text documents in only one dimension, (b) this work develops and makes available, in an open Github repository, a preprocessing solution for law texts in Portuguese, and (c) in the social context in Brazil, a developing country with serious

problems such as poverty and discrimination, this work applies and delivers a tool that helps standardize judgments, as the proposed model manages to relate new cases to cases already judged.

The results obtained in this work meet two sustainable goals in the world, according to the 17 goals established by the United Nations, with indicators 10.3 (Ensure equal opportunities and reduce inequalities in results, including the elimination of discriminatory laws, policies, and practices and the promotion of legislation, appropriate policies and actions in this regard) and 16.6 (Develop effective, accountable and transparent institutions at all levels).

This paper contains the following structure: Section 2 describes related works and theoretical background. Sections Section 3 details the proposed methodology, Section 4 and Section 5 present the results obtained and the discussion of the work, respectively, while the conclusions are provided in Section 6.

2. Related works

One of the fields with several recovering information applications is Law [35]. The volume of lawsuits filed in the judiciary, as opposed to the task force of the judges, as well as the possibilities of flows imposed in the legislations of the countries, has hindered the celerity in the attendance to the rights of the society [29,30,32]. Thus, since human resources to judge the number of lawsuits brought to justice is scarce, it is believed that optimization techniques with information technology can be applied to improve this scenario [29,30,32]. In the last years, one of the main innovations in the computational area has been artificial intelligence in text classification in law, with the construction of a network in a semantic context that allows meaning (semantics) and automatic treatment of information [3].

Sulis et al. [36] work on the identification and classification of judicial documents in text in the European Union, using simultaneously two methods to generate the features, being (a) manual annotation of eight categories established by the authors and (b) network of co-occurrence between terms. Various metrics are applied in the co-occurrence network to establish values in the combination of terms. After generating the features by this manual and automatic method, the classification algorithms are applied: logistic regression, decision trees, support vector machine, and *k*-nn. The dataset used is European Union standards. The best classifier was SVM in the results, reaching the value of 83% in the f-measure and 76% in the accuracy.

Mandal et al. [37] work with judicial texts known as precedents, dealing with statements from prior cases. The dataset used is Indian Supreme Court cases. Mandal et al. [37] investigate the performance of 56 different methodologies to calculate textual similarity in precedents. Among the solutions analyzed are models such as *tf-idf*, LDA, PScoreVect, Bert, Word2Vec, and Law2Vec. The authors noted that more traditional methods, such as *tf-idf* and LDA, which rely on bag-of-words representation, perform better than more advanced context-sensitive methods, such as BERT and Law2Vec [38], for document-level similarity computation. The average accuracy achieved in each model used was 80.8% for *tf-idf*, 77.1% for LDA, 67.1% for PScoreVect, 68.4% for Bert, 73.4% for Word2Vec, and 69.1% for Law2Vec. The article by Saura et al. [39] applies LDA to model sentimental analysis sample topics from tweets datasets, finding that the main issues are related to security in Internet of Things environments.

Skrlj et al. [40] propose a vectorization method where the construction of features is mixed using the *tf-idf* model simultaneously together with tax2vec. This proposed model applies prior knowledge of terms based on a WordNet taxonomic network. Tax2vec serves as a preprocessing method for enriching data with semantic features. Classifiers can use the resulting semantic features for learning. For the tests, six different types of datasets are used: three sets of tweets, one set of BBC news, and two related to biomedical drugs. SVM was the classification algorithm used. The authors used only the f-measure metric to evaluate, and the only satisfactory result in applying the proposed method was in the BBC news dataset, reaching 98%. The values are not significant for all other datasets, with 62% being the second-highest value found in the tweets' dataset, named PAN(Gender) in the article. The results in the two biomedical drug datasets, related to legal norms, reached 47% for the drug effect and 52.3% for the drug side.

Radygin et al. [41] developed software to search and analyze documents from arbitral tribunals under Federal Law. The proposal is to use AI to detect violations of Federal Law in Russian. In the document preparation flow, before the construction of features, the texts are preprocessed. For vectorization, *tf-idf* is used, and the applied classification algorithm is SVM. Training documents were obtained from an open Internet repository in Russia called Electronic Justice. The article does not detail the documents or how they were selected to perform the classification tests, but informs that they reached the value of 87% in the f-measure in the simulations.

Hausladen et al. [42] apply machine learning to U.S. Circuit Court documents, intending to explore and evaluate classifiers to predict conservative or liberal decisions, terms typical of the research by Hausladen et al. [42]. In the United States, judges wield significant power due to the common law system. The extent of the influence of U.S. judges is a motivation for extensive research into the determinants of judicial decision-making. In particular, there is a vast literature on how opinions are affected by the judge's ideology. In the vectorization of documents, the model *tf-idf* uni-grams and big-grams are used. The classifiers used are passive-aggressive, SVM, logistic regression, ridge, and Naive Bayes. The best classifier, passive-aggressive, reached the value of 74.5% in the f-measure and 77.1% in the accuracy on the dataset named Economic Activity.

Waltl et al. [43] work in the manual and automatic classification of norms in the German civil law domain, consisting of nine categories/labels of text documents. Bag-of-words is used in vectorization, and the *tf-idf* model is applied by the authors. The proposed methodology for manual and automatic classification is compared using f-measure, precision, and recall metrics. Five classification algorithms are used in simulations: support vector machine, random forest, multilayer perceptron neural network, Naive Bayes, and logistic regression. The best values were found for the automatic classification method, using the SVM classifier, 85% for precision, 84% for recall, and 83% for f-measure.

Katz et al. [44] work in the context of the Supreme Court of Justice of the United States to predict their behavior. A temporal evolution was developed using the random forest classifier and a solution based on metadata already recorded in the supreme court's database. A dataset of 240,000 court votes and 28,000 case results over nearly two centuries are used. The results in this volume of data reached the accuracy values of 70.2% in the attempt to predict the outcome of cases and 71.9% in the prediction of votes.

Medvedeva et al. [45] present a method to predict future case decisions, based on past cases, works with judgments, text documents of the European Court of Human Rights. The authors inform that with the availability of judgments, big data on justice, jurisprudence, it is possible to carry out studies in justice applying machine learning solutions. For document vectorization, the *tf-idf* model with n-gram method is applied. The support vector machine classifier is used to predict only two categories of outcomes: violation and non-violation. The accuracy achieved was 75%.

Ceci and Gangemi [6] develop work using the semantic web, a library of lawsuit knowledge based on the metadata contained in judicial documents, where semantic relationships are constructed by extracting fragments in lawsuit texts. As a result, the library called JudO, where it presents the interpretations of the judges in the conduct of their lawsuit reasoning. From the computer science area, the authors contribute to the modeling of judicial knowledge, resulting from studies based on cases and design patterns using ontology.

In the judiciary system, the finalization of the lawsuit occurs after the decision of the judge. The judge must analyze case by case with the view of value, fact, and norm [46]. If the law established in the codes were sufficient, immutable, perfect, and timeless, it would suffice to locate which norm would fit each case and apply it. Thus, the server of justice would be a simple automaton programmed to that task. However, the judges are more than automatons; they count on a cultural and life background, morality, and decisions they have taken in previous. In some countries, the decisions' database forms the basis of pacified understanding on related issues in judicial proceedings, attaching greater importance to the study and application by analogy of past cases. In other countries, with Roman–Germanic tradition, it is taught that the law in the codes is the primary source of the law, leaving the other sources as secondary in the case of the absence of norm resulting from the law [47].

However, in practice, it is noticed that the decisions already uttered are increasingly used as references in the decision in the trial. The search for the decisions already taken is a crucial stage of the judicial process in any country. The legal advisors are responsible for the research on the internet/intranet, consuming time to study the lawsuit and verifying of the judicial decisions similar to the current one through the text search. This search depends on employee interpretation, and available database structuring, which can result in information accuracy and retrieval quality problems. This search directly influences the draft decision forwarded to the judge to study and deliver the verdict [29].

In the face of the growth of legal demands, it is important to produce fast database mechanisms, automatic and intelligent to search, filter, classify, and choose the information, aggregating to improve search mechanisms. These automated solutions demand less human actions in the research process. Fawei et al. [7] describe an initial attempt to model and implement the automatic application of legal knowledge using a rule-based approach. It presents ontology as a promising method in the judiciary, applied in knowledge management where ontological elements are associated with legal rules.

Calambás et al. [8] use the ontology in the judicial area, being constructed a semantic relation of the base of lawsuit decisions uttered. Authors present progress in system development using natural language processing technology and clustering to optimize the recovery process and analysis of the bases of judicial decisions. Zhang et al. [9] discuss the difficulties encountered in the construction of Chinese lawsuit text classification. The authors argue that the challenges lie in the classification and the organization of data in the Chinese judicial system. There are three obstacles to the Chinese judicial system: (i) ambiguity of lawsuit language, (ii) deficiency in the inference of judicial decisions, and (iii) limited role of the lawsuit in China. The authors state that, based on these difficulties, the proposed judicial classification is the mixture between normative documents and lawsuits. Because it is a conceptual work, the results focus on presenting the challenges for the construction of judicial taxonomy.

Ni et al. [4] continue the work in Zhang et al. [9], creating the information retrieval system based on judicial precedents and lawsuits. The system allows recovering judgments and norms by relevance. The methodology used supports logical reasoning and incorporates a hierarchical structure. In addition to the ontological contribution generated, the authors say that they can improve the accuracy metrics in information retrieval using genetic algorithms integrated with the K-Nearest Neighbor (K-nn).

Wu and Zhang [11] propose an electronic evidence analysis model based on Chinese criminal prosecution data mining. This work is important in China and other countries because electronic data is a kind of independent evidence, and it has received attention in criminal trials. In this paper, the authors apply the machine learning method based on rules to find relations among variables on databases with a large volume of data. An improved algorithm (ISPO-tree algorithm) is put forward based on the FP-growth algorithm, and the theoretical proof is given. The results present an improvement in the time efficiency of data preprocessing.

Castro et al. [29,30] apply similarity learning among the terms in text documents in the Judiciary; their results are presented by precision and recall metrics in data mining. However, this work does not apply any classification model to demonstrate the ability to predict new text documents. The authors apply the term frequency model (tf) with the Jaccard similarity algorithm, combining and using terms 2×2 to mine data using the SQL language, obtaining the best result: 64% in precision and 63% in the recall, in the model named hybrid. Castro et al. [32] apply similarity-learning to lawsuits to group them. With the unification of the processes in clusters and the integration of this solution with the government system, it was possible to inform the lawsuits that have similarities in fact and legal thesis, alerting and facilitating the analysis by the judge [32].

2.1. Environment for case study: unstructured data from the department of justice

In the world, there is no standard of the functioning of the judiciary, and each country has established its way of judging and structuring its judiciary. Some countries prioritize laws in codes, while other recurring decisions override laws in codes. In the first case, countries are known as Civil Law, while in the latter, they are known as Common Law [48,49]. Mandal et al. [37] claim that one of the most followed legal systems globally is the Common Law System. Mandal et al. [37] keep claiming that it is followed by one-third of the world's population in several countries, including India, Australia, the USA, and the United Kingdom.

In Brazil, the judiciary is one of the three pillars that controls and organizes its democratic society, being responsible for enforcing the laws in the country, becoming the guardian of the rules, which establish the relationship between people [50]. According to the National Council of Justice in Kim and Toffoli [51], the judiciary in Brazil ends the year 2018 with more than 80 million lawsuits in progress. A Fig. 1, adapted from Kim and Toffoli [51], shows the lawsuit stock increased to the previous year, and it is increasing year by year [52,53].

Fig. 1 shows that the judiciary is unable to reduce pending court cases. The judiciary needs to increase the number of judges and their advisers or construct software tools capable of speeding up the procedures for judging and filing lawsuits. This work is trying to use artificial intelligence software to help in this scenario. Judging the lawsuits, the Departments of Justice manages to reduce this large volume shown in Fig. 1.

The documents used in this research are texts of a judgment of judges, known as decisions. They are unstructured documents rich in information, integrating generated facts and applied laws [29,30,32,33]. Judicial decisions have no standards or structures; they are linked only to the type of lawsuits that

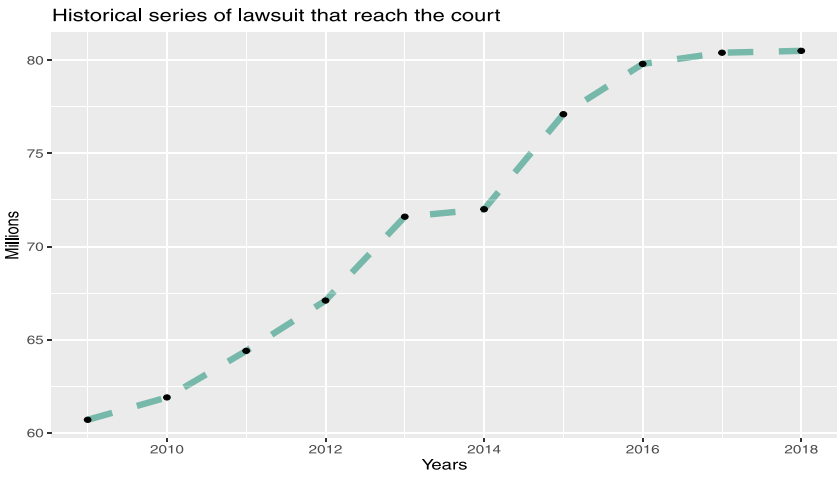


Fig. 1. Department of Justice branch's historical lawsuit series.

is in progress, like (i) contract, (ii) family, (iii) car accident, (iv) drugs, (v) medical malpractice, (vi) crime, (vii) product liability, (viii) workers' compensation and others. Using data mining techniques that allow us to classify situations and facts, creating predictive connections in new cases when recognized, allowing batch judgment to help accelerate the Department of Justice's progress is an option for computing application [29,30].

2.2. Quantitative model in information retrieve

Information retrieval is responsible for handling and retrieving data objects such as text, images, sounds, and so on. Mooers [54], Almeida [55], Delicato et al. [56], and others describe and evolve information retrieval (IR), making it more sophisticated and interactive.

The objective of IR is to find and present the correct information, from the contents of the document to the user, satisfying their need in the search expression. The search engine is the most relevant point in retrieving the information, and it compares the *query* of the users/systems with the *corpus*. Most of the search engines are quantitative, based on logic, statistics, and set theory disciplines. Boughanem et al. [57] state that quantitative models have boosted the development of information retrieval systems, including (i) booleans, (ii) vector, (iii) probabilistic, and (iv) clustering. The efficiency of the IR system is directly linked to the applied model. These IR models are also known as search engines.

The quantitative models used in information retrieval can associate weights in terms of indexing and search expression. These weights calculate the degree of similarity between search expressions established by the user for each document or between documents. Thus, it is possible to obtain documents ordered by degree of similarity based on search expression [58,59].

The term is the word that represents the concept or meaning present in the document. Identifying the relevance of the term in the description of the document's content is an onerous task [58,59]. The weight calculation is an important aspect, and it can be applied in several ways, as described by Salton and McGill [60], Salton [61], Ponte and Croft [58,59]. Well-known feature weighting schemes are extensively researched and applied, as term frequency (*tf*), term frequency-inverse document frequency (*tf-idf*), and Okapi best matching 25 (*BM25*) [31].

The *corpus* is commonly represented by a matrix with various documents and indexing terms. In the matrix, the information can be retrieved through the similarity calculation, where it is intended to quantify the similarity of content between two documents or between the search expression and each of the documents in the *corpus*. Some traditional models for calculating similarity are: (i) Jaccard model; (ii) the cosine model; (iii) coefficient of Dice, and (iv) other [62].

The Jaccard S_ϑ similarity is a metric used in statistics. It returns values in the range [0 1], being indicated to compare significant volumes of documents and terms, as in the case of *Big Data*. Jaccard's expression measures the similarity relation between documents D_1 and D_2 and is given by:

$$S_\vartheta(D_1, D_2) = \frac{\sum_{j=1}^{n}(w_{1,j} \cdot w_{2,j})}{\sum_{j=1}^{n}(w_{1,j})^2 + \sum_{j=1}^{n}(w_{2,j})^2 - \sum_{j=1}^{n}(w_{1,j} \cdot w_{2,j})} \quad (1)$$

where $w_{1,j}$ is the weight of the jth term of document D_1 and $w_{2,j}$ is the weight of the jth term of document D_2. The result of the expression (1) is presented in percentage of similarity of document D_1 with document D_2.

3. Methodology

The methodology developed in this paper has the main object of providing celerity in the judgments of the judicial proceedings because it identifies and classifies the received lawsuits, allowing connection with cases already judged. The method proposed collaborates with researches in the field of machine learning in text classification because it innovates in two aspects: (a) solve the limitation of the model *tf-idf* presented by Agarwal et al. [12] and Seo et al. [13], given that the application of the Jaccard similarity, jointly with the *tf-idf*, can relate the co-occurrence of the terms in the documents and (b) it softens the non-linearity of the similar terms in different categories of documents. The stages of being implemented are: (i) feature extraction from a signed judicial document; (ii) preprocessing to remove HTML *tags* and unnecessary features; (iii) separation of terms by a term frequency–inverse document frequency model (*tf-idf*); (iv) application of the similarity metric in (3) to find combined terms, two by two, and choose the weight; (v) creation of binary and frequency vectors and (vi) supervised training of classification models. The flow constructed to apply the proposed methodology is illustrated in Fig. 2.

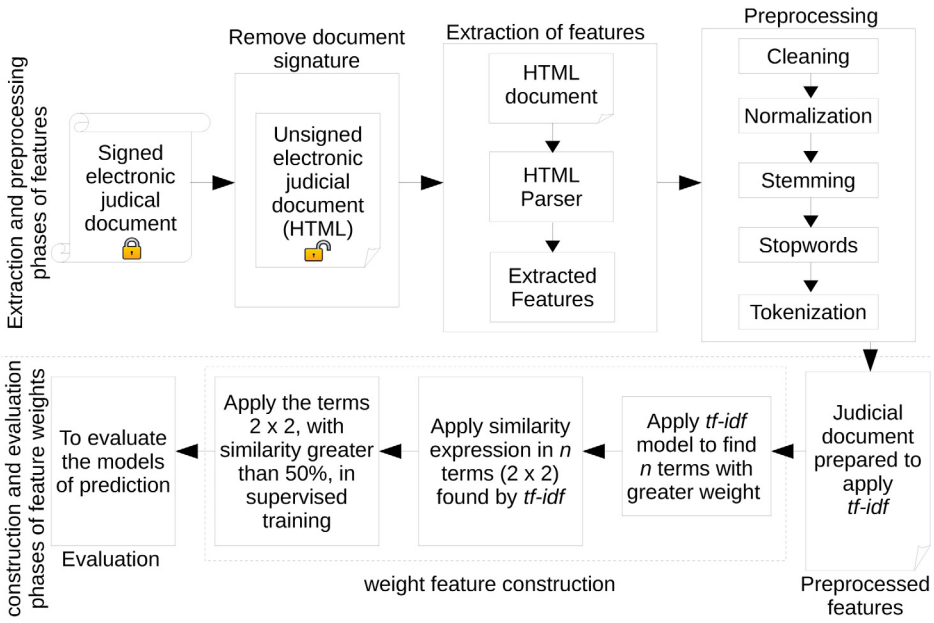


Fig. 2. Overview of applied methodology.

3.1. Feature extraction from signed judicial document

The first phase of the methodology consists of preparing the documents for the preprocessing phase. The electronic lawsuit documents are signed using a digital certificate and are stored in the database in a *blob* format, in binary type. At this stage, it is necessary to remove the digital signature from the documents and transform them into the ascii text format, in utf-8. The resulting after conversion is HTML ascii format.

3.2. Preprocessing of extracted features from judicial document

In the context in which the universe of unstructured documents D of the court of justice is inserted, it is necessary to apply preprocessing phase since there are countless terms, HTML *tags* that are not necessary and may interfere with the methodology. The text documents used are judgments delivered by judges of the Court of Justice in 2016. After removing the signatures and transforming them from binary to HTML format, 8,734 judgments in text documents are preprocessed.

Preprocessing is performed using routines built in the Python language with NLTK library and the Ruby language. However, no specific libraries were found in Python or Ruby that implemented cleaning, normalization, stemming, and stopwords for texts judged in Portuguese. These libraries were made available on Github[1], under GNU General Public License v 3.0, and on Code Ocean[2] aiming to share them in the new research.

- HTML parser/remove: after feature extraction from the signed judicial document, the document is converted to HTML format. In this step, it is necessary to eliminate language-specific *tags* from HTML format.
- Cleaning: it is usual in all judgments documents to include special characters, like ampersand; double quotes; left and right single quotation; ordinal indicator; section sign; vertical bar; semicolon; brace; bracket; hyphen, and others.
- Normalization: in normalization, all words are placed in lowercase, the accents are removed, characters like ç are placed as the letter c. Words with the same meaning are also treated in this normalization phase, such as pharmacy and drugstore. This solution was implemented by the authors and is available on Github.
- Stemming: stemming reduce inflected or derived words to their base. Several Latin words are found in the judgment texts. At first, a module was developed to convert Latin into Portuguese, but it became clear after the first simulations that it was unnecessary. The similarity learning technique aims to find situations of combined words that allow classifying texts, so words in Latin can be helpful in this process. So, the conversion of Latin to Portuguese module was no longer used.
- Stopwords: stopwords remove all articles, prepositions, conjunctions without meanings, words with no semantic meaning to the text, include in this preprocessing common words in the area of Law that are not relevant for the context.
- Vectorization: vectorization transforms the text into a vector of words. Each document was transformed into a vector of combined words. Since this step occurred after several others during preprocessing, the vector is now ready to be used by the algorithm in Fig. 3. The repeated words were kept inside the vector.

Fig. 3 illustrates the algorithm of the proposed model.

3.3. Separation of terms by frequency–inverse document frequency

In the *corpus* of unstructured text documents of the same category (D), a matrix of document $\times$ terms is structured, and *tf-idf* model calculates the weights of the terms in the documents. The weights found for each of the terms in each document is added to arrive at the total weight of each term in all documents, given by:

$$W_{t_i}(t_i) = \sum_{i=1}^{m} (p_{t_i}) \tag{2}$$

where m is the total document in D, p_{t_i} is the weight of the term t_i in document D_i and $i = 1, 2, \ldots, m$. The results of calculating

[1] https://github.com/apcastrojr/court_of_law_pre_processing.
[2] https://doi.org/10.24433/CO.3115967.v1.

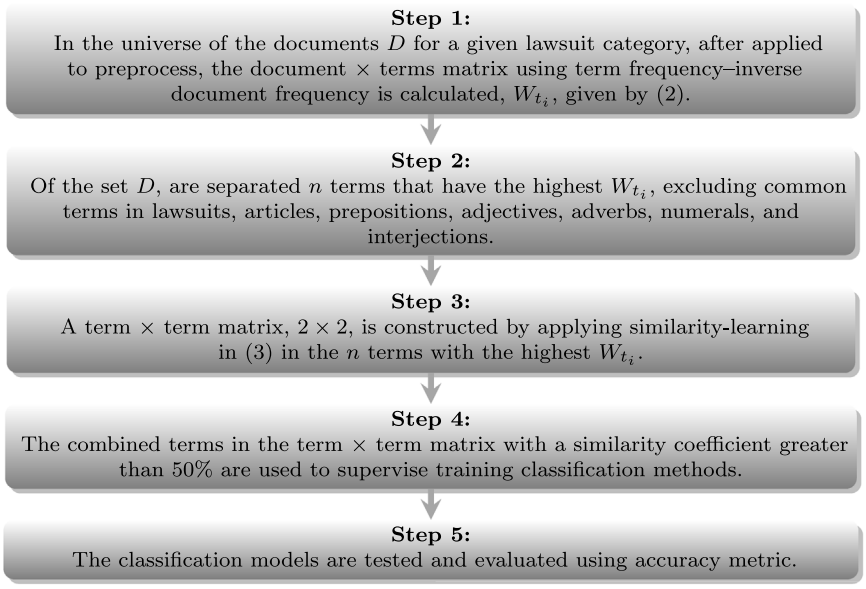


Fig. 3. Summary of the joint application of the *tf-idf* model with Jaccard's altered similarity in (3).

the weights in (2) for terms are used to identify n terms with the highest W_{t_i}. These n terms are used in the similarity-learning for combined terms phase. To calculate the weight t_i, it is used term frequency–inverse document frequency (*tf-idf*). For the comparative purpose of the *tf-idf* model, it is also applied to the *Okapi BM25*, which is a model traditionally known and applied to the constructions of weights in the terms.

3.4. Modified similarity-learning technique for weight construction

A different way of applying the similarity in (1) is presented in this article to generate knowledge and identity to the *corpus*. The similarity technique will not be used to compare documents, D_1 and D_2, or compare search expressions and documents. It will be applied to find similarities between n terms, t_1 and t_2, in all text documents D in a database for a certain category. The n terms used are those that had the highest W_{t_i} found by *tf-idf* model. The Jaccard expression in (1) is altered to construct the relationship between the terms or combined terms. The altered Jaccard expression S_α, is given by:

$$S_\alpha(t_1, t_2) = \frac{\sum_{i=1}^{m}(w_{i,1} \cdot w_{i,2})}{\sum_{i=1}^{m}[(w_{i,1})^2 + (w_{i,2})^2] - \sum_{i=1}^{m}(w_{i,1} \cdot w_{i,2})} \tag{3}$$

In (3), the similarity between the terms t_1 and t_2 is calculated on all documents in the *corpus D*, for the same category, where t_1 is the first term and t_2 is the second term, $w_{i,1}$ is the frequency–inverse document frequency of the term t_1 in the ith document, and $w_{i,2}$ is the frequency–inverse document frequency of the term t_2 in the ith document. This process is repeated by calculating the similarity between all n terms $t_j = t_1, t_2, \ldots, t_n$.

After calculating combinations in all n terms and all m documents in D, it is possible to infer the similarity relation of the terms in the *corpus* of a specific category, as set out in Table 1, which considers the relationship is automatically constructed between the terms t_1 and t_2. In (3) is constructed to calculate the similarity between two terms (2×2), but this expression can be generalized to perform the calculation by combining several terms ($n \times n$).

Table 1
Relationship between terms built automatically, co-occurrence between the terms given by S_α(3).

	t_1	t_2	t_3	$\cdots$	t_n
t_1	–	$S_\alpha(t_1, t_2)$	$S_\alpha(t_1, t_3)$	$\cdots$	$S_\alpha(t_1, t_n)$
t_2	$S_\alpha(t_2, t_1)$	–	$S_\alpha(t_2, t_3)$	$\cdots$	$S_\alpha(t_2, t_n)$
t_3	$S_\alpha(t_3, t_1)$	$S_\alpha(t_3, t_2)$	–	$\cdots$	$S_\alpha(t_3, t_n)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
t_n	$S_\alpha(t_n, t_1)$	$S_\alpha(t_n, t_2)$	$S_\alpha(t_n, t_3)$	$\cdots$	–

Table 1 is the matrix that represents the knowledge extracted from the *corpus D* of the same category. It is observed that this matrix has the main diagonal null and that the values above the main diagonal are identical to the values below the main diagonal. The result of the application of this methodology creates the relationship between terms, generating the digital fingerprint of the *corpus* for a specific lawsuit category. This same computational procedure can be implemented for any *corpus*, constructing the understanding in the relations between combined terms and *corpus D*. This built-in fingerprint is used for supervised training of classification models.

3.5. Supervised training and classification models

After establishing the knowledge of the *corpus*, two types of vectors for combined terms found by the percentage of similarity given by (3) are used in two supervised training approaches for classification models: binary and frequency. The objective is to analyze which of the two vectors is better for supervised training. Thus, each court document is represented by a vector. Each field in a vector is represented by a combined term (2×2), generating two vector approaches: (i) binary vector and (ii) frequency vector. In the binary vector approach, if the court document has the combined terms in its content, the field in a vector receives one, otherwise zero. If the court document has combined terms in your content in the frequency vector approach, the value of the lowest frequency of one of the double terms is inserted in the field in a vector, otherwise zero. Then, a vector is the fingerprint of one court text document.

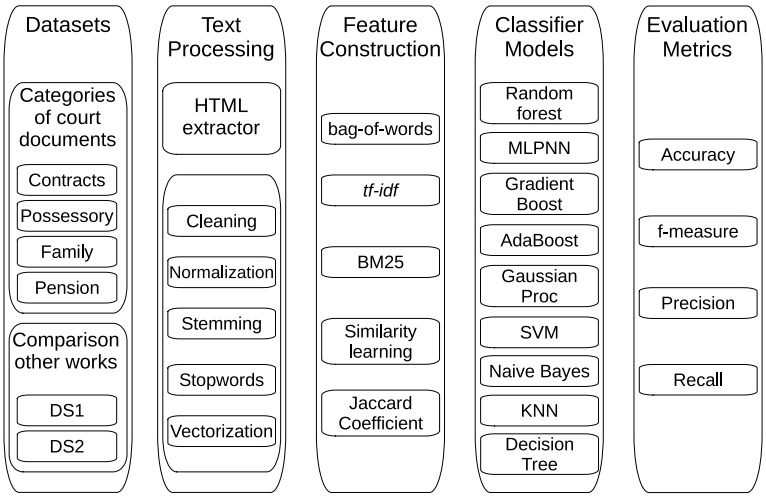


Fig. 4. Summary of the methodological approach with datasets and applied technologies.

This vector is the input to the classifications models. The number of fields in a vector is the same number of combined terms with the coefficient of $\geq$ 50%, found out by similarity-learning expression in (3). In this work, nine classification technologies are used and compared. They are random forest, multilayer perceptron with backpropagation neural network (MLPNN), adaptive boosting, gradient boosting, Gaussian process, support vector machine, Naive Bayes, *k*-nearest neighbors and decision trees. The objective is to verify which approach is best suited. The algorithm is shown in Fig. 3 is used in the Big Data of court judgments of a given category, generating the vectors of the most frequent combination of terms. The objective of applying the similarity coefficient of $\geq$ 50% is to reduce or avoid the repetition of possible terms combined in different categories of documents to smooth out the non-linearity of the problem.

3.6. Summary of the methodological approach

Fig. 4 provides a graphical summary of the methodological approach for a better understanding of the technologies used in the simulations.

4. Results

In this paper, the computational routine was constructed by using Ruby-on-rails and Python languages. The weighting construction using *tf-idf*, similarity-learning expression, and datasets construction were developed in Ruby-on-Rail. The dataset imports and classification models were developed in Python. The `tf-idf-similarity` ruby gem is used to implement term frequency-inverse document frequency (*tf-idf*). The similarity-learning using Jaccard in (3) was built by the authors using Ruby-on-Rails. The Scikit-learn package is used to implement text classifications: random forest, multilayer perceptron with backpropagation neural network (MLPNN), adaptive boosting (AdaBoost), gradient boosting, Gaussian process, support vector machine (SVM), Naive Bayes, *k*-nearest neighbors and decision trees.

4.1. Import and store unstructured data

It was necessary to study and understand the modeling of the Department of Justice system in the central region of Brazil, the state of Goias, and plan the integration to obtain judges' judgments. Judgments are linked in the system to the categories of lawsuits registered. Four categories of lawsuits were chosen to carry out the experiments: (i) contract; (ii) possessory; (iii) family, and (iv) pension. The unstructured data was stored as signed documents, in binary format, in a proprietary database. It was necessary to remove the signatures and transform them from binary to HTML ascii format, in utf-8. After the transformation, the HTML documents were imported into the free database, PostgreSQL. It was imported for simulations 8,734 HTML documents, distributed as shown in Table 2.

Table 2
Categories of lawsuits and amount of documents imported.

Categories	Amount
Contracts	1,948
Possessory	774
Family	838
Pension	5,174

4.2. Applying tf-idf and similarity-learning for combined terms in contract category

A *corpus* of 1,948 court decision documents of contract category are separated and applied Step 1 and Step 2, as shown in an algorithm in Fig. 3. Ten terms, $n = 10$, with higher *tf-idf* in contract category text documents are separated, disregarding common terms in preprocessing. With the terms of highest *tf-idf* incidence, a term $\times$ term matrix is structured (Step 3 in Fig. 3) using (3). With the automatic co-occurrence between terms created, the combined terms with a similarity coefficient of $\geq$ 50% are used to identify contract category documents (Step 4 in Fig. 3), as shown in Tables 3 and 4. The same process is applied to other categories: possessory, family and pension.

4.3. Applying tf-idf and similarity-learning for combined terms in all category

The same processing performed in Section 4.2 for a category of contracts is applied to all other categories shown in Table 2. Step 1 and Step 2 of the algorithm illustrated in Fig. 3 are applied, resulting in the terms with the highest weight, found by the *tf-idf* method. The number of terms was fixed at ten, with the highest weight for each category, i.e., $n = 10$, to standardize the calculations in this article. Table 5 shows 40 terms found by the

Table 3
Court decisions indexations terms - similarity matrix 1/2.

	interest	*contract*	*review*	*billing*	*payment*
interest	-	0.85	0.37	0.68	0.39
contract	0.85	-	0.48	0.67	0.49
review	0.37	0.48	-	0.51	0.65
billing	0.68	0.67	0.51	-	0.52
payment	0.39	0.49	0.65	0.52	-
capitalization	0.79	0.71	0.48	0.78	0.5
value	0.47	0.61	0.66	0.54	0.69
rate	0.83	0.74	0.41	0.76	0.41
clauses	0.53	0.63	0.68	0.67	0.62
credit	0.49	0.55	0.63	0.65	0.62

Table 4
Court decisions indexations terms - similarity matrix 2/2.

	capitalization	*value*	*rate*	*clauses*	*credit*
interest	0.79	0.47	0.83	0.53	0.49
contract	0.71	0.61	0.74	0.63	0.55
review	0.48	0.66	0.41	0.68	0.63
billing	0.78	0.54	0.76	0.67	0.65
payment	0.5	0.69	0.41	0.62	0.62
capitalization	-	0.52	0.76	0.69	0.6
value	0.52	-	0.47	0.63	0.63
rate	0.76	0.47	-	0.58	0.53
clauses	0.69	0.63	0.58	-	0.65
credit	0.6	0.63	0.53	0.65	-

tf-idf weight models, with $n = 10$ in each of the four document categories.

After applying the *tf-idf* method, equal terms in different categories were found, making the problem nonlinear. In Table 5, we have the following terms: (i) **value** found in the categories **contract**, **possessory**, and **family**; (ii) **contract** found in the **contract** and **possessory** categories; (iii) **deadline** found in the categories **possessory**, **family**, and **pension**; (iv) **own** found in the categories **possessory** and **family**; (v) **proof** found in the **possessory** and **pension** categories and (vi) **benefit** found in the **family** and **pension** categories. In percentage terms, we have the **possessory** category, 50% of the terms are also in the other categories. In the same analysis, we have that 40% of the terms in the **family** category are also in the other categories, 30% of the **pension** category terms are also used in the other categories and 20% of the **contract** category terms are used in the other categories. Table 6 shows the repeated terms found and the number of repetitions.

After finding the terms with the highest weights, the Step 3 and Step 4 of the algorithm presented in Fig. 3 are applied. Using the similarity-learning in (3) method for combined $n = 10$ terms, 2×2, the number of terms combined with coefficient of $\geq 50\%$ is disposed in Table 7. The purpose of using the similarity coefficient of $\geq$ 50% is to identify and use a high degree of relationship between the terms of the *corpus* and to smooth the non-linearity, i.e., to make the problem linear for the terms in the document categories. Table 5 shows common terms between different categories of documents, such as own, value, deadline, benefit, contract, and others. These common terms found in different categories make the problem non-linear. However, with the application of (3), the problem becomes linear, as it can be seen in Table 7, there are no combined terms repeated in different categories of documents. This methodology reduces the processing time in training the classification methods and the forecasting time for new documents. This knowledge is used to train the classification models of this work.

4.4. Training approaches and classification models

Following the methodology stamped in Fig. 3, the combined terms found by (3), in term $\times$ term matrix with similarity coefficient greater than 50% are used in the supervision of training classification methods. These combined terms are shown in Table 7. In this scenario, the court of a law text document is represented by a vector. Each field in a vector is represented by a combined term. This vector is the input to the text classification models. The number of fields in a vector is the same number of combined terms discovered by similarity-learning applied. The combined terms shown in Table 7 are transformed into a vector, depending on whether or not the court document has the combined terms in its content.

Two dataset approaches are generated to verify the best training suite: binary vector or frequency vector. For generating the dataset, the documents of each category indicated in Table 2 are randomly separated. The complete training dataset consists of 1,619 documents, divided between the categories of contract, possessory, family, and pension, as shown in Table 8. To separate the dataset with a binary vector, if the term combination exists in the text document, the vector field is set to **one**, otherwise **zero**. To separate the dataset with frequency vector, if there is a combination of terms within the document, the frequency of the two terms within the document is counted, the **lowest frequency** between the two combined terms is inserted in the vector field. If the two terms of the combination do not exist in the document, the vector field is defined as **zero**.

First, supervised training is carried out with the binary dataset, and the results are evaluated on a test basis containing 200 documents, of which 50 are from each category. After obtaining the results, new supervised training is carried out with the frequency dataset of the combined terms. The results are evaluated on a test basis containing the same 200 documents. These 200 test documents are different from the 1,619 training documents. For supervised training, nine text classification models are used: (i) MLP neural networks; (ii) Random Forest; (iii) Gradient Boosting; (iv) Adaptive Boosting; (v) Gaussian Process, (vi) Support Vector Machine, (vii) Naive Bayes; (viii) *k*-nearest neighbors and (ix) Decision Trees. The Python *scikit-learn* framework is used to implement these methods.

The `sklearn.neural_network` library was used in Python language to implement MLP neural network classifier. For the `MLPClassifier()` method of the `MLPClassifier` class, the following parameters are used: `activation='logistic', solver='lbfgs', alpha=1e-5, hidden_layer_sizes=(100,50), random_state=1, max_iter=300`. The `sklearn.ensemble` library was used in Python language to implement Random Forest, AdaBoost and Gradient Tree Boosting classifiers. For the `RandomForestClassifier()`, `AdaBoostClassifier()` and `GradientBoostingClassifier()` methods, respectively of the `RandomForestClassifier`, `AdaBoostClassifier` and `GradientBoostingClassifier` classes, was used `n_estimator=300` as a parameter. The `sklearn.gaussian_process` library was used in Python language to implement Gaussian Process Classification (GPC). For the `GaussianProcessClassifier()` method of the `GaussianProcessClassifier` class, was used `max_iter_predict=300` as a parameter. The `sklearn.svm` library was used in Python language to implement Support Vector Machines classification (SVM). For the `SVC()` method of the SVC class, was used `max_iter=300` as a parameter. For the `DecisionTreeClassifier()` and `GaussianNB()` methods, respectively of the `tree`, `GaussianNB` classes, their default parameters are used. For the `KNeighborsClassifier()` method of the `KNeighborsClassifier` class, the `n_neighbors=3` parameter is used.

Table 5
Terms found by *tf-idf* models in *corpus* reported in the Table 2 (Step 1 and Step 2 in Fig. 3).

Categories	Terms found by *tf-idf* weight constructions
Contract	contract, interest, rate, value, capitalization, billing, review, payment, clauses, credit
Possessory	possess, reintegration, deadline, own, ownership, form, mortgage, proof, contract, value
Family	divorce, deadline, own, foods, agreement, assistance, benefit, value, guardianship, litigious
Pension	benefit, deadline, inss*, federal, countryside, concession, proof, national, retirement, age

* National Institute of Social Security.

Table 6
Non-linearity of terms found in different categories in Table 5.

Repeated terms	Number of repetitions
contract, own, proof, benefit	2
value, deadline	3

All classification methods used in the evaluation were configured to use the same value, *iterations* = 300. The number of iterations equal to 300 standardizes supervised training across all classification models. The classifier software, training, and tests datasets are available on GitHub,[3] under GNU General Public License v 3.0. The objective is to analyze which of the classification models is the most suitable for the similarity method of the combined terms and the weighting models applied in the *corpus* of the Court of Justice. Accuracy, f-measure, precision, and recall are used to measure the best methods used.

4.5. Accuracy, f-measure, precision, and recall results

This evaluation process needs to combine two types of training datasets, binary, and frequency, with nine classification models, generating eighteen different results for each evaluation metric. As four evaluation metrics are used, this section presents seventy-two results. The tests' results are shown in Fig. 5, Fig. 6, Fig. 7, and Fig. 8 for the evaluation metrics: accuracy, f-measure, precision, and recall. In each figure, the results are presented by comparing the binary and frequency datasets classification results. For the classification tests, 200 judicial documents were used.

The datasets represent the knowledge extracted from the *corpus* of the four categories of unstructured text documents of the court of justice. For knowledge generation, the terms with the highest *tf-idf* weights were separated first. With the $n = 10$ terms with the highest *tf-idf*, they were combined 2×2 using Jaccard's similarity in (3). Terms combined with a coefficient greater than 50% are used for training.

The results presented in Fig. 5, Fig. 6, Fig. 7, and Fig. 8 show the capacity of the weight building methodology applied in this article to represent the documents, reaching the result of 85% as the best result in accuracy and recall, 85.6% as the best result in f-measure and 88.5% as best result in precision, with the MLP neural network classification model using frequency dataset for training. The most significant values in the accuracy, f-measure, and recall metrics in the binary dataset are from the SVM model. However, in the precision metric, the highest value was the Naive Bayes. The results presented in Fig. 6 show the harmonic mean between precision and recall, reinforces the accuracy values obtained in Fig. 5, demonstrating the relevance of the proposed method. The average processing time was 23 s for supervised learning and prediction of the nine classification models and datasets used. The longest processing time was for the AdaBoost algorithm, with 31 s.

[3] https://github.com/apcastrojr/court_of_law_datasets_and_text_classifiers2.

4.6. Comparison of results with other work

Abualigah et al. [34] present and assess algorithms used to solve the problem of text classification and grouping, use corpus from text documents freely available at the University of São Paulo (USP), institute of mathematics and computer sciences.[4] The same *corpus* of text documents used and assessed in Abualigah et al. [34], named DS1 and DS2, is used in this paper. The goal is to compare the results of the metrics of accuracy, f-measure, precision, and recall obtained by the algorithms Harmony Search (HS) Algorithm, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Krill Herd Algorithm (KHA), Cuckoo Search (CS) Algorithm, Gray Wolf Optimizer (GWO), Bat-inspired Algorithm (BA), and *k*-means technique in the study of Abualigah et al. [34], with the algorithms Random Forest (RF), MultiLayer Perceptron Neural Network (MLPNN), Gradient Boosting (GB), Adaptive Boosting (AB), Gaussian Process (GP), Support Vector Machine (SVM), Naive Bayes (NB), *k*-nearest neighbors (KNN), Decision Trees (DT), utilized in this article. Table 9 shows the used datasets in the experiments.

Fig. 9 shows the comparative results between the metrics of accuracy, f-measure, precision, and recall obtained in this paper and Abualigah et al. [34] for the DS1 dataset. Fig. 10 shows the comparative results between the metrics of accuracy, f-measure, precision, and recall obtained in this paper and Abualigah et al. [34] for the DS2 dataset. In Fig. 9, for the DS1 dataset, it is noticed that the proposed method in this paper obtains better results than the one from Abualigah et al. [34]. In Fig. 10, for the DS2 dataset, excluding the GWO algorithm, which presents results similar to the ones from this paper, all the other algorithms in Abualigah et al. [34] had values in the assessment metrics lower than the ones presented in this study. In the general analysis of the assessment metrics from Fig. 9, DS1 dataset, the algorithms MLPNN and NB in this study present better results. In the overview of the assessment metrics from Fig. 10, the DS2 dataset, the GWO in the study of Abualigah et al. [34], and MLPNN in the model proposed in this study present better results. Analyzing only the metric of precision, in Fig. 10, DS2 dataset, the algorithm GP in this study presents the best result among all the other algorithms.

Table 10, Table 11, Table 12, and Table 13 present the best results applying the method proposed in this study. Table 10 shows the results of the accuracy metric, Table 11 shows the results of the f-measure metric, Table 12 shows the results of the precision metric, and Table 13 the ones of recall metric. The source code of the programs, the datasets, and the text documents used are available on Github,[5] under GNU General Public License v 3.0, and on Code Ocean[6] aiming to share them in the new research.

[4] http://sites.labic.icmc.usp.br/text_collections/.
[5] https://github.com/apcastrojr/text_documents_ds1_ds2_to_compare.
[6] https://doi.org/10.24433/CO.5422851.v1.

Table 7
Combined terms found by *tf-idf* and similarity (3) with a coefficient of $\geq$ 50% (Step 3 and Step 4 in Fig. 3), makes the problem linear.

Categories	Combined terms found by *tf-idf* and altered Jaccard in (3)
Contract	interest-contract, interest-billing, interest-capitalization, interest-rate, interest-clauses, contract-billing, contract-capitalization, contract-value, contract-rate, contract-clauses, contract-credit, review-billing, review-payment, review-value, review-clauses, review-credit, billing-payment, billing-capitalization, billing-value, billing-rate, billing-clauses, billing-credit, payment-capitalization, payment-value, payment-clauses, payment-credit, capitalization-value, capitalization-rate, capitalization-clauses, capitalization-credit, value-clauses, value-credit, rate-clauses, rate-credit, clauses-credit
Possessory	reintegration-mortgage, reintegration-own, possess-mortgage, possess-ownership, possess-own, possess-reintegration
Family	own-foods, divorce-agreement, divorce-own, assistance-benefit
Pension	inss*-federal, benefit-national, benefit-countryside, benefit-age, deadline-inss, concession-national, proof-retirement, benefit-proof, concession-proof, inss-retirement, countryside-proof, concession-retirement, inss-concession, countryside-retirement, benefit-retirement, retirement-age, benefit-inss, inss-national, benefit-concession, countryside-age

* National Institute of Social Security.

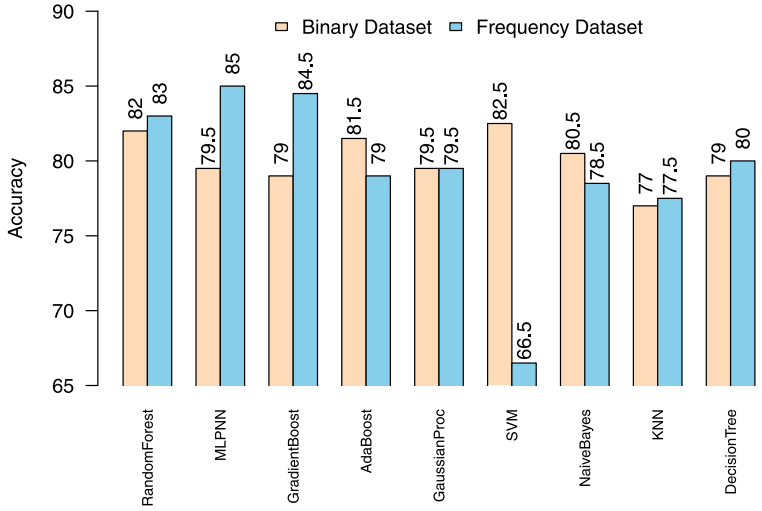


Fig. 5. Comparison of accuracy results using binary and frequency dataset.

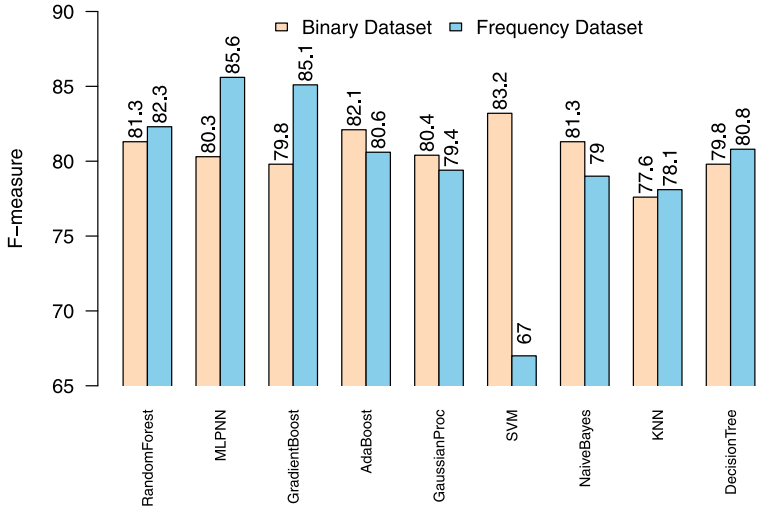


Fig. 6. Comparison of f-measure results using binary and frequency dataset.

Analyzing the origin of the corpus DS1 and DS2 informed in Table 9, it is noticed that to the technical documents, like the ones from the DS1 dataset (Technical Reports), the proposed method is better than the one applied in Abualigah et al. [34], in comparison to the generic documents, like the ones from the DS2 dataset (Web Pages). Since the documents' nature in which the research in this paper is inserted has a technical character, for it is related to judgments from the Court of Justice, respecting governmental

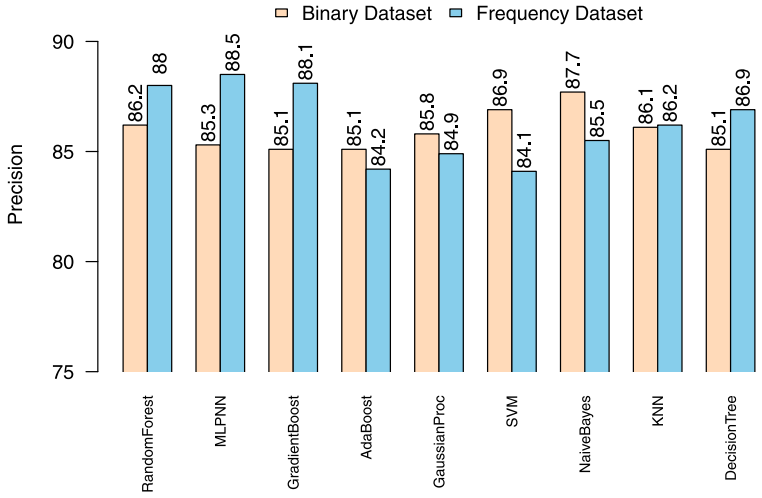


Fig. 7. Comparison of precision results using binary and frequency dataset.

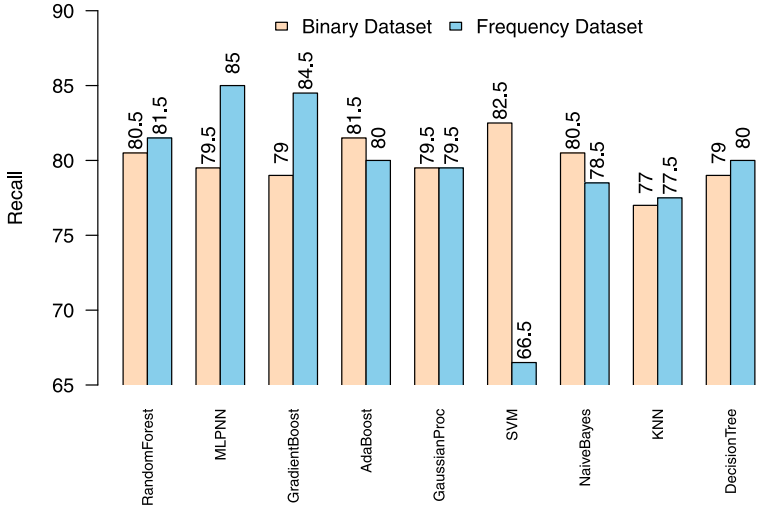


Fig. 8. Comparison of recall results using binary and frequency dataset.

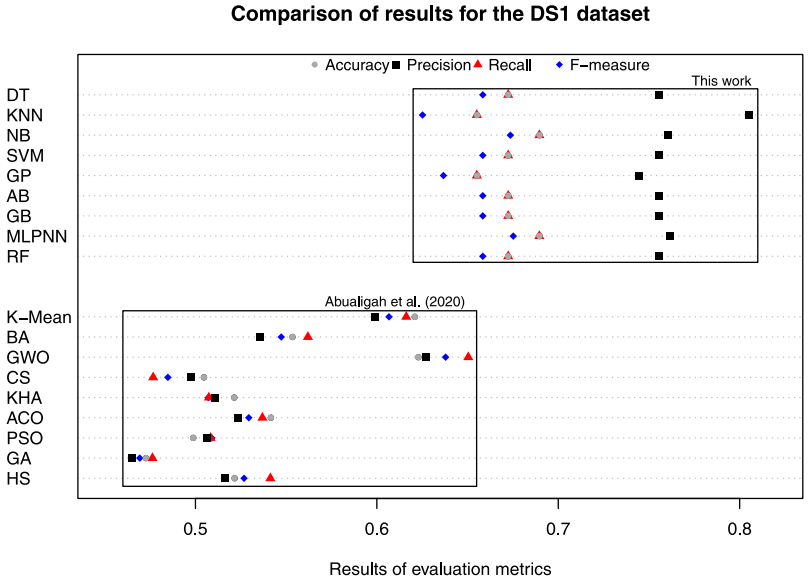


Fig. 9. Comparison of accuracy, f-measure, precision and recall metrics between Abualigah et al. [34] article and this work for DS1 dataset.

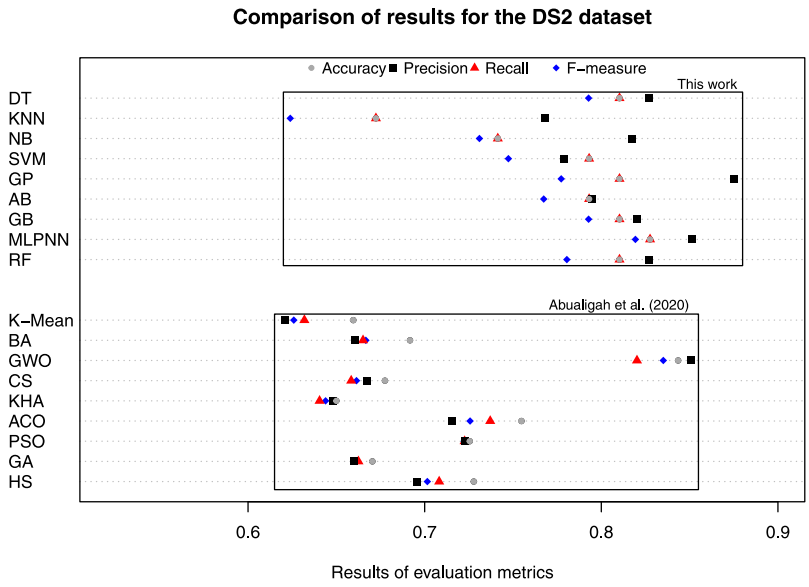


Fig. 10. Comparison of accuracy, f-measure, precision and recall metrics between Abualigah et al. [34] article and this work for DS2 dataset.

Table 8
Categories of lawsuits and amount of documents randomly separated for training.

Categories	Amount
Contracts	440
Possessory	371
Family	368
Pension	440

Table 9
Corpus used in the experiments.

Datasets	Number of Documents	Categories	Sources
DS1	299	4	Technical Reports*
DS2	333	4	Web Page*

*Abualigah et al. [34].

laws and standards to solve the population problems, the comparison with Abualigah et al. [34] reinforces the proposed method in this study and its application in the production environment, through API.

4.7. Apply Okapi BM25 instead of tf-idf and compare results

In the method proposed in this study, the *tf-idf* model is replaced by the *Okapi BM25* (*BM25*) in the identification of the terms. Table 14 presents the found terms applying the model *BM25*, step 1, and step 2 of the algorithm illustrated in Fig. 3 to the same *corpus*, presented in Table 2. Table 15 shows the repeated terms found through the *BM25* model, non-linear problems. Applying steps 3 and 4 of the illustrated algorithm in Fig. 3, it is given in Table 16, the co-occurrence of the terms found through the application of the similarity in (3), transforming the linear problem.

The co-occurrence of the found terms, presented in Table 16, is used in the supervised training for the models of classification Random Forest (RF), MultiLayer Perceptron Neural Network (MLPNN), Gradient Boosting (GB), Adaptive Boosting (AB), Gaussian Process (GP), Support Vector Machine (SVM), Naive Bayes (NB), *k*-nearest neighbors (KNN), and Decision Trees (DT). The same datasets utilized in the tests for the *tf-idf* model are applied for the *BM25* model, being the results of the metric presented in Table 17. The training and tests datasets using *Okapi BM25* model are available on GitHub,[7] under GNU General Public License v 3.0.

It is noticed in the results that the *tf-idf* model presents better values in the assessment metrics in comparison to the model *Okapi BM25*, Fig. 11. Therefore, it reinforces the use of the *tf-idf* model with the Jaccard similarity in (3), in the implementation of the API of integration of the proposed method with the platform of process management in the Court of Justice.

4.8. Results summary

Fig. 12 shows the results in a summarized form in applying the method proposed in the text documents of the Court of Justice of the State of Goias, Brazil. Through the radar plot, it is possible to check comparatively all 72 results of applying the model and training to the Court's document, using a vector with binary values (radar-plot on the left) and a vector with frequency values (right) in nine different classification algorithms.

The radar graph in Fig. 12 summarizes the results as (a) the average of the evaluation metrics was 80%, some cases reaching almost 90%, which demonstrates the efficiency of the model proposed in this work, (b) the best results are when supervised training is applied with frequency vectors, (c) however, using the frequency vectors in training, there are more significant variations between the results of the metric in classification algorithms, unlike when binary vectors are used, (d) the achieved values in the accuracy and recall metrics are close, (e) evaluating all classification metrics together, for each classification algorithm, the best classifier found is MLPNN, with results close to 90%.

4.9. Implementation of the integration API

In order for the solution to be effective in practice, in the day-to-day activities of the Department of Justice, it is necessary to implement a solution that allows notifying the Judges when the method predicts new lawsuits in categories: (i) contract; (ii) possessory; (iii) family and (iv) pension. The electronic lawsuits software system used in the Department of Justice is known by the name of Projudi. The Projudi tool controls and

[7] https://github.com/apcastrojr/court_of_law_datasets_and_text_classifiers2.

Table 10
The results of the comparative methods in terms of accuracy measure.

Dataset	Type	RF	MLP	GB	AB	GP	SVM	NB	KNN	DT
DS1	Binary	0.6724	0.6897	0.6724	0.6724	0.6552	0.6724	0.6897	0.6552	0.6724
	Frequency	0.6552	0.6724	0.6552	0.6724	0.6552	0.6724	0.6897	0.5517	0.6379
DS2	Binary	0.8103	0.7931	0.7931	0.7759	0.8103	0.7931	0.7414	0.6379	0.8103
	Frequency	0.8103	0.8276	0.8103	0.7931	0.8103	0.5345	0.7241	0.6724	0.8103

Table 11
The results of the comparative methods in terms of f-measure measure.

Dataset	Type	RF	MLP	GB	AB	GP	SVM	NB	KNN	DT
DS1	Binary	0.6584	0.6752	0.6584	0.6584	0.6367	0.6568	0.6736	0.6252	0.6584
	Frequency	0.6360	0.6526	0.6360	0.6584	0.6360	0.6584	0.6691	0.4965	0.6144
DS2	Binary	0.7805	0.7674	0.7674	0.7393	0.7590	0.7474	0.7310	0.5677	0.7805
	Frequency	0.7773	0.8193	0.7928	0.7674	0.7773	0.4040	0.7163	0.6239	0.7928

Table 12
The results of the comparative methods in terms of precision measure.

Dataset	Type	RF	MLP	GB	AB	GP	SVM	NB	KNN	DT
DS1	Binary	0.7555	0.7618	0.7555	0.7555	0.7248	0.7312	0.7411	0.7430	0.7555
	Frequency	0.7446	0.7507	0.7446	0.7555	0.7446	0.7555	0.7603	0.8051	0.7188
DS2	Binary	0.8271	0.7947	0.7947	0.7574	0.8752	0.7790	0.8172	0.6587	0.8271
	Frequency	0.8169	0.8514	0.8200	0.7947	0.8169	0.3624	0.7826	0.7679	0.8200

Table 13
The results of the comparative methods in terms of recall measure.

Dataset	Type	RF	MLP	GB	AB	GP	SVM	NB	KNN	DT
DS1	Binary	0.6724	0.6896	0.6724	0.6724	0.6551	0.6724	0.6896	0.6551	0.6724
	Frequency	0.6551	0.6724	0.6551	0.6724	0.6551	0.6724	0.6896	0.5517	0.6379
DS2	Binary	0.8103	0.7931	0.7931	0.7759	0.8103	0.7931	0.7414	0.6379	0.8103
	Frequency	0.8103	0.8276	0.8103	0.7931	0.8103	0.5345	0.7241	0.6724	0.8103

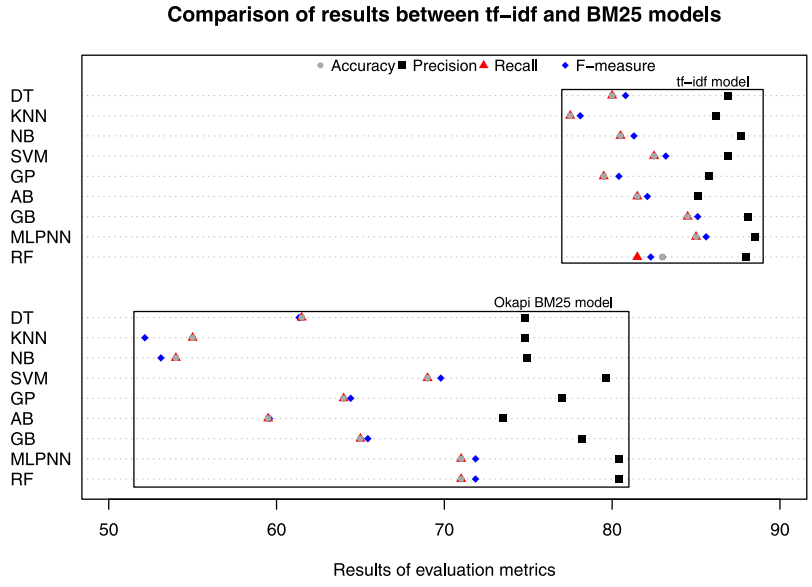
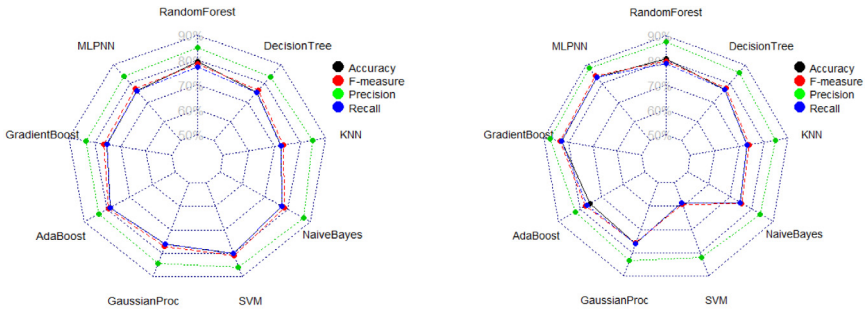


Fig. 11. Comparison of accuracy, f-measure, precision, and recall metrics between *tf-idf* and *Okapi BM25* models.

processes the lawsuits. Every day, the API implemented takes the Projudi database, the first document filed in a lawsuit, known as a complaint, and predicts its type. After classifying the lawsuit, the API informs the result to the Projudi. Thus, the Projudi user can verify this notice itself. With the notice, the Projudi user will become aware of the cases and take the necessary procedures. The information appears both on the main screen of the Projudi and on the screen of the lawsuit itself.

5. Discussion

The studies developed in this article are relevant because they are applied research in a government agency about the Court of Justice. When there is no peaceful solution to social conflicts, the Court of Justice is activated to resolve the divergence, applying the rules and regulations established in the country. However, the volume of approximately two million lawsuits under study,



(a) data vector with binary values

(b) data vector with frequency values

Fig. 12. Summary of model application and training of nine different classification algorithms: (a) data vector with binary values and (b) data vector with frequency values.

Table 14
Terms found by *BM25* models in *corpus* reported in the Table 2 (Step 1 and Step 2 in Fig. 3).

Categories	Terms found by *BM25* weight constructions
Contract	Contract, value, review, deadline, payment, own, form, proof, credit, clauses
Possessory	Deadline, possess, mortgage, ownership, contract, area, property, value, concession, requirements
Family	Foods, value, guardianship, proof, deadline, couple, children, agreement, constitution, separation
Pension	Deadline, countryside, insured, age, proof, retirement, value, concession, activity, payment

Table 15
Non-linearity of terms found in different categories in Table 14.

Repeated terms	Number of repetitions
contract, payment, concession	2
proof	3
value, deadline	4

Table 16
Combined terms found by *BM25* and similarity (3) with a coefficient of $\geq 50\%$ (Step 3 and Step 4 in Fig. 3), makes the problem linear.

Categories	Combined terms found by *BM25* and altered Jaccard in (3)
Contract	Contract-review, contract-clauses, contract-form, credit-clauses, review-payment, contract-own, contract-credit, review-form, Review-own, own-form, contract-proof, payment-form
Possessory	Possess-mortgage, possess-ownership, concession-requirements ownership-property
Family	Foods-children, guardianship-children, foods-guardianship
Pension	Countryside-age, retirement-activity, value-payment, Insured-retirement, age-activity, age-retirement, Countryside-retirement, insured-activity, proof-activity

with approximately three hundred and eighty judges to judge them [52] at the Department of Justice in the central region of Brazil, state of Goias, with an estimated population in 2020 of 7,113,540 people [63], make it difficult to resolve conflicts quickly. In addition, about four hundred thousand new lawsuits are received each year [52]. In this context, the application of artificial intelligence to relate the new lawsuits with the judgments already handed down would become an essential tool for judges, as they can speed up the judgment, linking related lawsuits already judged and avoiding divergent judgments for related lawsuits. The API is installed to integrate the proposed method and is currently working in the Court of Justice.

Simulations have shown that the application of altered Jaccard in (3), given by S_α, to find similarity in the co-occurrence of terms can recognition court documents, and it can be applied together with different text classification models in to predict new types of lawsuits. From the results presented in Fig. 5, Fig. 6, Fig. 7, and Fig. 8, summarized in Fig. 12, it is observed that the combination of traditional weight construction models *tf-idf* with similarity Jaccard measure to generate knowledge of the *corpus* reached 85% in the accuracy and recall, 85.6% in f-measure and 88.5% in precision in the classification of documents, using MLP neural network with supervised training with dataset structured by the frequency vector. These values are significant coefficients because they refer to unstructured texts. Therefore, demonstrating the potential of the proposed methodology.

The preprocessing phase is important in this work, mainly due to the unstructured characteristics of the documents and the common terms routinely applied in the *corpus* of the Court of Justice. The preprocessing for specific terms in this branch of knowledge was built. Therefore it was possible to develop this work because, as an inherent difficulty in this research, no specific libraries were found in Python or Ruby that implemented cleaning, normalization, stemming, and stopwords for texts judged in Portuguese. These libraries were made available on Github[8], under GNU General Public License v 3.0, and on Code Ocean[9] to share them in the new research.

In the results presented using binary vectors for training, a slight variation of 5.5% in accuracy and recall, 5.6% in f-measure, and 2.6% in precision are observed between the classification models used. The lowest value found was 77% in the accuracy and recall, 77.6% in f-measure in the *k*-nn model, while the highest value was 82.5% in the accuracy and recall, 83.2% in the Support Vector Machine (SVM) model. For the precision evaluated, the highest value was 87.7% in Naive Bayes and the lowest value was 85.1% in Gradient Boosting, Adaptive Boosting, and Decision Trees. In the results presented using frequency vectors for training, there was a more significant variation among the classification models used, 18.5% in the accuracy and recall, 18.6% in f-measure. The lowest value found was 66.5% in the accuracy and recall, 67% in f-measure in the Support Vector Machine (SVM) model. In contrast, the highest value found was 85% in the accuracy and recall, 85.6% in f-measure and 88.5% in precision in the Multilayer Perceptron neural network (MLPNN) model. Through the results presented, if the vectors are binary, it is recommended to apply the SVM model for prediction, and if the

[8] https://github.com/apcastrojr/court_of_law_pre_processing.
[9] https://codeocean.com/capsule/4399430/tree/v1.

Table 17
The results metrics of classification methods and *Okapi BM25* in conjunction of similarity by (3).

ClassificationMethods	Type of Vector	Accuracy	F-measure	Precision	Recall
RF	Binary	71%	71.86%	80.43%	71%
	Frequency	67.5%	68.28%	76.22%	67.5%
MLPNN	Binary	71%	71.86%	80.43%	71%
	Frequency	68.5%	69.39%	77.28%	68.5%
GB	Binary	65%	65.44%	78.18%	65%
	Frequency	63%	62.75%	73.62%	63%
AB	Binary	59.5%	59.58%	72.06%	59.5%
	Frequency	56.5%	56%	73.47%	56.5%
GP	Binary	64%	64.24%	77.01%	64%
	Frequency	64%	64.42%	75.59%	64%
SVM	Binary	69%	69.79%	79.66%	69%
	Frequency	58%	59%	73.19%	58%
NB	Binary	54%	53.11%	70.01%	54%
	Frequency	53.5%	52.38%	74.91%	53.5%
KNN	Binary	51.5%	47.96%	70.91%	51.5%
	Frequency	55%	52.15%	74.82%	55%
DT	Binary	61.5%	61.34%	74.79%	61.5%
	Frequency	59%	58.01%	72.98%	59%

vectors are frequency, it is recommended to apply the MLPNN model for prediction. We do not recommend using the frequency vector dataset for the SVM model since it did not present good accuracy, f-measure, and recall results, well below the average of the other models applied in the studies. However, excluding the SVM model, it is clear that the frequency vector dataset is the most advisable to be applied in supervised training, and we recommend its use in place of the binary vector dataset, mainly with the MLPNN classification model.

Analyzing the general results of the metrics of accuracy, f-measure, precision, and recall, it is noticed that the Random Forest, MLPNN, Gradient Boosting, and Decision Trees classification models presented better results with frequency vectors. In contrast, the Adaptive Boosting, SVM, Naive Bayes classification models presented better results with binary vectors. The Gaussian Process and *k*-nn classification models had, practically, the same results in both types of vectors. The processing time for supervised learning and prediction averaged 23 s for all classification models and types of datasets.

To bring reliability to the proposed method, the results of this work are compared with recent work by Abualigah et al. [34]. In the results section, two datasets used in Abualigah et al. [34], related to Technical Reports and Web Pages, freely available on Internet, are also used and evaluated in this article as shown in Table 9. In the comparisons, it is observed in Figs. 9 and 10 that the proposed method in this study presents better results in the assessment metrics used, with relevance to datasets related to technical documents created by professionals. In these comparative studies, it can be seen in Figs. 9 and 10 that the model proposed in this article presents better results in classifying technical text documents than non-technical ones. Considering that the Court of Justice documents are technical, it reinforces the social and academic benefits of the proposed methods and this research work.

Other recent research works, such as Sulis et al. [36], Mandal et al. [37], Skrlj et al. [40], Radygin et al. [41], Hausladen et al. [42], Waltl et al. [43], and Medvedeva et al. [45], are applying BOW with *tf-idf* in vectorization and classifiers in legal documents. Although applied to relatively different datasets in law, these studies present results similar to those obtained by the method proposed in this article. To summarize the discussion, Table 18 presents the works, datasets used and their countries of application, the values achieved by the evaluation metrics used, the classifiers that presented the best results, the BOW model applied, and the year of publication of the works. Except for the f-measure metric of 87% in the work of Radygin et al. [41], the results achieved in this article were higher than the articles listed in Table 18. However, the result of 85.6% achieved in the f-measure in this article was very close to the 87% value achieved by Radygin et al. [41], for the same metric.

Another comparison carried in the result's section is the replacement of the model *tf-idf* in the construction of the features vector through the model *Okapi BM25*. Since they are two models traditionally applied in the vectorization of terms, according to Mironczuk and Protasiewicz [64], their comparison becomes relevant in this study. In the results obtained in Fig. 11, it is observed that the combination of traditional models of representation of text documents in vectors with the technique of Jaccard similarity learning, the application of the *tf-idf* model is better than the application of the *Okapi BM25* model.

Finally, the method proposed in this paper enhanced the *tf-idf* model when applied in conjunction with Jaccard's similarity in (3), as it solves the limitation reported in the articles of Agarwal et al. [12], Seo et al. [13], and Li et al. [14] because it can bring the co-occurrence among the terms of the *corpus*. The co-occurrence of the terms found by calculation of similarity, given by S_α in (3), was performed two by two (2×2), this combination can be modified to perform the calculation with larger combinations of terms and producing more accurate results.

The research is being developed, not limited to the text classification models applied in this work. Researches with other recent models such as convolutional neural networks (CNN) are already underway. Studies with other weight construction models, text embedding encodings, such as doc2vec and BERT, are also being carried out.

6. Conclusion

In this work, a different way of using the Jaccard similarity technique is applied in conjunction with the definition of weight by the *tf-idf* model, generating knowledge of the *corpus* and allowing supervised training of different classification models to predict new lawsuits in the Court of Justice. The results show that it is a viable methodology to be applied and used in the context of document categories, which has as a characteristic the non-linearity of the terms in the categories, and in practice, to help streamline the work of judges in their judgments. The API is built to allow the use of the proposed methodology in the government

Table 18
Comparison of recent research work applying BOW and classifiers to relatively different datasets in law.

Research papers	Year	BOW used	Dataset countries	Metric evaluation used	Best classifier
This work	–	*tf-idf* and altered Jaccard in (3)	Brazil, court of the state of Goias	85.6% f-measure 88.5% precision 85% accuracy 85% recall	MLPNN
Sulis et al. [36]	2021	*tf-idf* and manual method	European Union	83% f-measure 76% accuracy	SVM
Mandal et al. [37]	2021	*tf-idf* LDA and PScoreVect	Indian Supreme Court cases	Mean accuracy *tf-idf:* 80.8% LDA: 77.1% PScore: 67.1%	Similarity coefficient in two classes
Skrlj et al. [40]	2021	*tf-idf* and tax2vec	Biomedical* Drugs effect and Drugs side	f-measure 47% drugs effect 52.3% drugs side	SVM
Radygin et al. [41]	2021	*tf-idf*	Russian Federal Law	87% f-measure	SVM
Hausladen et al. [42]	2020	*tf-idf*	U.S. Courts	74.5% f-measure 77.1% accuracy	Passive aggressive
Medvedeva et al. [45]	2020	*tf-idf*	European Human Rights	75% accuracy 83% f-measure	SVM
Waltl et al. [43]	2018	*tf-idf* and manual method	German Civil Law	85% precision 84% recall	SVM
Katz et al. [44]	2017	manual based on metadata	U.S. Supreme Courts	70.2% accuracy 71.9% precision	Random Forest

* No country, Drugs.com and Druglib.com.

agency. Through the API, it is possible to inform the result in the correlation of judgments already handed down with new lawsuits that go to the Department of Justice.

The proposed weight construction methodology is used to generate two types of vectors: binary and frequency. These two features vector models are compared and used for supervised training of nine text classification technologies: Random Forest, MLP neural networks, Adaptive Boosting, Gradient Boosting, Gaussian Process, SVM, Naive Bayes, *k*-nn and Decision Trees. In general, as covered in the discussion section, the best results were obtained with supervised training using the frequency vector. However, three classification models had their results worsened when using the frequency vectors compared to the binary vectors.

The results generated in this work are relevant to know the text classification model most adherent to the area of knowledge in which this article is applied. The studies help choose the best predictive solutions to be implemented in production environments, especially in places with intense movement and flow of unstructured documents, such as Courts of Justice in Brazil, state of Goias. With the accuracy of 85%, the f-measure of 85.6%, the precision of 88.5%, and the recall of 85%, by supervised training using frequency vector dataset, the results have shown that the best classifier model, among those applied in this work, is the MLP neural network.

A relevant analysis in this work was carried out replacing the *tf-idf* model with the BM25 model in the vectorization process. The BM25 model was applied together with the altered Jaccard in (3) to vectorize the documents with the co-occurrence of the terms. However, from the results of Fig. 11, it can be seen that the values of the classification metrics were better using the *tf-idf* model. We do not recommend replacing *tf-idf* with BM25 for the proposed model.

Comparisons with other studies are carried out to bring credibility to the results of the proposed method, as follows: (a) the model proposed in this work was applied to the same datasets as the Abualigah et al. [34], with the results of the evaluation metrics being superior, as shown in Figs. 9 and 10, and (b) with recent articles, shown in Table 18, that despite the studies using relatively different datasets in the area of law, there is a certain balance between the values of the classification metrics, but with superior results by the method applied in this article. Comparisons with other studies in the results sections reinforce the quality of the proposed method and address a gap reported in the introduction, and may also pave the way for the academic community to continue studies on improving BOW models, successfully applying joint similarity-learning techniques for vectoring text documents.

The research carried out, and the proposed AI method consolidate the solution implemented in a production environment in the Court of Justice of the State of Goias, Brazil. In the social sphere, based on the results obtained, the proposed solution fills the second gap reported in the introduction, as it allows for standardizing judgments, reducing divergences, possible inequalities, and, in certain cases, even discrimination. Thus, this article complies with indicator 10.3 and indicator 16.6, within two sustainable goals in the world, goal 10 and goal 16, following the 17 goals established by the United Nations.

As additional contributions, the Python programs are built to perform the accuracy, f-measure, precision, and recall calculations for the different classification models used in this work. The training and test datasets are available on the GitHub platform. The aim of this availability is that other researchers can continue the work, comparing it with different classification models. Availability was made using the GNU General Public License v 3.0. The method is still being refined and improved, especially when using other similarity-learning techniques, such as Cosine, combined with other weight generation models. New studies are being carried out, and the results are compared. The authors of this paper are testing other classification models. Additionally, it is important to inform that the positive results of this work led the Goias Judiciary in Brazil to allocate a team of professionals to study and include new subcategories of lawsuits in the applied classification method. The objective is to identify repetitive demands on a large scale, with possibilities to establish current standards in judgments.

Finally, it is important to mention that the method proposed in this work can be applied in several branches of knowledge that have large volumes of text documents and need to automate knowing, establishing intelligent and automatic relationships with new text documents inputs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank National Council for Scientific and Technological Development (CNPq), Brazil, Foundation for Research Support of the State of Goias (FAPEG), Brazil, Brazilian Federal Agency for Support and Evaluation of Graduate Education (CAPES), Brazil and Court of Justice of the State of Goias, Brazil.

References

[1] O. Arsene, I. Dumitrache, I. Mihu, Medicine expert system dynamic Bayesian network and ontology based, Expert Syst. Appl. 38 (12) (2011) 15253–15261.
[2] J.-B. Lamy, Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies, Artif. Intell. Med. 80 (2017) 11–28.
[3] M. Rani, A.K. Dhar, O. Vyas, Semi-automatic terminology ontology learning based on topic modeling, Eng. Appl. Artif. Intell. 63 (2017) 108–125.
[4] Z. Ni, Y. fei Pu, S.-Q. Yang, J.-L. Zhou, J. kang Gao, An ontological Chinese legal consultation system, IEEE Access 5 (2017) 18250–18261.
[5] A. Grubišić, S. Stankov, I. Peraić, Ontology based approach to Bayesian student model design, Expert Syst. Appl. 40 (13) (2013) 5363–5371.
[6] M. Ceci, A. Gangemi, An OWL ontology library representing judicial interpretations, Semant. Web 7 (3) (2016) 229–253.
[7] B. Fawei, A. Wyner, J.Z. Pan, M. Kollingbaum, Using legal ontologies with rules for legal textual entailment, in: AI Approaches to the Complexity of Legal Systems, Springer, 2015, pp. 317–324.
[8] M.A. Calambás, A. Ordóñez, A. Chacón, H. Ordoñez, Judicial precedents search supported by natural language processing and clustering, in: 2015 10th Computing Colombian Conference, 10CCC, IEEE, 2015, pp. 372–377.
[9] N. Zhang, W. Ping, Y. fei Pu, Challenges and related issues for building Chinese legal ontology, in: 2015 International Conference on Mechatronics, Electronic, Industrial and Control Engineering, MEIC-15, Atlantis Press, 2015, pp. 1260–1265.
[10] L. Huang, D. Milne, E. Frank, I.H. Witten, Learning a concept-based document similarity measure, J. Am. Soc. Inf. Sci. Technol. (2012).
[11] Y. Wu, J. Zhang, Building the electronic evidence analysis model based on association rule mining and FP-growth algorithm, Soft Comput. J. (2019).
[12] N. Agarwal, G. Sikka, L.K. Awasthi, Enhancing web service clustering using length feature weight method for service description document vector space representation, Expert Syst. Appl. J. (2020).
[13] S. Seo, D. Seo, M. Jang, J. Jeong, P. Kang, Unusual customer response identification and visualization based on text mining and anomaly detection, Expert Syst. Appl. J. (2020).
[14] P. Li, K. Maoa, Y. Xu, Q. Li, J. Zhang, Bag-of-concepts representation for document classification based on automatic knowledge acquisition from probabilistic knowledge base, Knowl.-Based Syst. (2020).
[15] L. Abualigah, A. Diabat, S. Mirjalili, M.A. Elaziz, A.H. Gandomi, The arithmetic optimization algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021).
[16] K.P. Murphy, Machine Learning: A Probabilistic Perspective, The MIT Press (Cambridge Massachusetts), 2013.
[17] V. Garg, S. Vempati, C.V. Jawahar, Bag of visual words: A soft clustering based exposition, in: Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, 2011.
[18] A. Bosch, A. Zisserman, X. Munoz, Scene classification using a hybrid generative/discriminative approach, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008).
[19] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, CVPR, 2006.
[20] R. Fergus, P. Perona, A. Zisserma, Object class recognition by unsuper-vised scale-invariant learning, CVPR, 2003.
[21] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, Comput. Res. Reposit. (2013) (CoRR).
[22] D.N. Milne, I.H. Witten, D.M. Nichols, A knowledge-based search engine powered by wikipedia, in: Proceedings of the 16th Association for Computing Machinery (ACM) Conference on Information and Knowledge Management, ACM Press, 2007.
[23] R. Mihalcea, C. Corley, C. Strapparava, Corpus-based and knowledge-based measures of text semantic similarity, in: Proceedings of the 21st National Conference on Artificial Intelligence, AAAI Press, 2006.
[24] E. Gabrilovich, S. Markovitch, Feature generation for text categorization using world knowledge, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence, Kaufmann, 2005.
[25] Y. Kim, Convolutional neural networks for sentence classification, Comput. Res. Reposit. (2014) (CoRR).
[26] J. Pennington, R. Socher, C. Manning, GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2014, pp. 1532–1543.
[27] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: North American Chapter of the Association for Computational Linguistics - NAACL, ACM Digital Library, 2018.
[28] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, Comput. Res. Reposit. (2018) (CoRR).
[29] A.P. Castro, W.P. Calixto, V.M. Gomes, E.F. Veiga, L.F. Silva, L.L.O.P. Castro, J.L.F. Barbosa, P.H. Campos, Ontology applied in the judicial sentences, in: 2017 CHILECON Conference on Electrical, Electronics Engineering, Information and Communication Technologies, CHILECON, IEEE, 2017a, pp. 1–6.
[30] A.P. Castro, W.P. Calixto, V.M. Gomes, E.F. Veiga, L.F. Silva, P.H. Campos, Ontology to mining judicial sentences big data, in: Alive Engineering Education, UFG, 2017b, pp. 187–196.
[31] W.H. Gomaa, A.A. Fahmy, A survey of text similarity approaches, Int. J. Comput. Appl. (2013).
[32] A.P. Castro, W.P. Calixto, C.H. Araujo, Application of artificial intelligence in the identification of connections by fact and thesis in the judicial complaint and integration with the electronic system of lawsuits (in Portuguese), CNJ Magazine 4 (1) (2020) 10.
[33] S. Jasanoff, Science, common sense & judicial power in U.S. courts, Daedalus – J. Am. Acad. Arts Sci. (2018).
[34] L. Abualigah, A.H. Gandomi, M.A. Elaziz, A.G. Hussien, A. Khasawneh, M. Alshinwan, E.H. Houssein, Nature-inspired optimization algorithms for text document clustering—A comprehensive analysis, Algorithms (2020).
[35] K.D. Ashley, Artificial Intelligence and Legal Analytics: New Tools for Law Practice in the Digital Age, Cambridge University Press, 2017.
[36] E. Sulis, L. Humphreys, F. Vernero, I.A. Amantea, D. Audrito, L.D. Caro, Exploiting co-occurrence networks for classification of implicit inter-relationships in legal texts, Inf. Syst. (2021).
[37] A. Mandal, K. Ghosh, S. Ghosh, S. Mandal, Unsupervised approaches for measuring textual similarity between legal court case reports, Artif Intell. Law (2021).
[38] I. Chalkidis, Law2Vec - Legal Word Embeddings, Archive.org, 2021.
[39] J.R. Saura, D. Palacios-Marqués, D. Ribeiro-Soriano, Using data mining techniques to explore security issues in smart living environments in Twitter, Comput. Commun. (2021).
[40] B. Skrlj, M. Martinc, J. Kralj, N. Lavrac, S. Pollak, Tax2vec: Constructing interpretable features from taxonomies for short text classification, Comput. Speech Lang. (2021).
[41] V. Radygin, D. Kupriyanov, R. Bessonov, M. Ivanov, I. Osliakova, Application of text mining technologies in Russian language for solving the problems of primary financial monitoring, Procedia Comput. Sci. (2021).
[42] C.I. Hausladen, M.H. Schubert, E. Ash, Text classification of ideological direction in judicial opinions, Int. Rev. Law Econ. (2020).
[43] B. Waltl, G. Bonczek, E. Scepankova, F. Matthes, Semantic types of legal norms in German laws: classification and analysis using local linear explanations, Artif. Intell. Law (2018).
[44] D.M. Katz, M.J. Bommarito, J. Blackman, A general approach for predicting the behavior of the supreme court of the United States, PLOS ONE (2017).
[45] M. Medvedeva, M. Vols, M. Wieling, Using machine learning to predict decisions of the European court of human rights, Artif. Intell. Law (2020).
[46] F. Glitz, Incoterms and Brazilian legislation on contracts, Educ. Sci. Borders J. 2 (3) (2011) 40–44.
[47] S.H. Bailey, M.J. Gunn, The Modern English Legal System, fifth ed., Sweet & Maxwell, 2007.
[48] R. David, Major Legal Systems in the World Today (in Portuguese), Martins Fontes, 2014.
[49] J.C.C. Moura, M.T.C. Sousa, Towards judiciary: brief psychoanalyst and historical considerations about voluntary subjection to the law and Judiciary (in Portuguese), in: Cad. Pesq., Vol. 20, São Luís, 2013, 3.
[50] V.C.C. Donato, The Judiciary in Brazil: structure, criticism and control (in Portuguese), (Ph.D. thesis), University of Fortaleza (UNIFOR) - Brazil, 2006.
[51] R.P. Kim, J.A.D. Toffoli, Justice in Numbers: document produced by the Brazilian judiciary, Digital Magazine of the National Council of Justice – CNJ (in Portuguese), 2019.
[52] C.L.A. Rocha, Justice in Numbers: document produced by the Brazilian judiciary, Digital Magazine of the National Council of Justice – CNJ (in Portuguese), 2018.

[53] R.P. Kim, J.A.D. Toffoli, Justice in Numbers: document produced by the Brazilian judiciary, Digital Magazine of the National Council of Justice – CNJ (in Portuguese), 2020.
[54] C.N. Mooers, Zatocoding applied to mechanical organization of knowledge, Am Document. 2 (1) (1951) 20–32.
[55] M.B. Almeida, Revisiting ontologies: A necessary clarification, J. Am. Soc. Inf. Sci. Technol. 64 (8) (2013) 1682–1693.
[56] F.C. Delicato, L. Pirmez, L.F.R.C. Carmo, Fenix–personalized information filtering system for WWW pages, Internet Res. 11 (1) (2001) 42–48.
[57] M. Boughanem, A. Brini, D. Dubois, Possibilistic networks for information retrieval, Internat. J. Approx. Reason. 50 (7) (2009) 957–968.
[58] J.M. Ponte, W.B. Croft, A language modeling approach to information retrieval, (Ph.D. thesis), University of Massachusetts at Amherst, 1998.
[59] J.M. Ponte, W.B. Croft, A language modeling approach to information retrieval, in: ACM SIGIR Forum, Vol. 51, ACM, 2017, pp. 202–208.
[60] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, McGraw-Hill, Inc., 1986.
[61] G. Salton, Automatic text processing: The transformation, analysis, and retrieval of Reading, Addison-Wesley, 1989.
[62] V. Thada, V. Jaglan, Comparison of jaccard, dice, cosine similarity coefficient to find best fitness value for web retrieved documents using genetic algorithm, Int. J. Innov. Eng. Technol. 2 (4) (2013) 202–205.
[63] J.M. Bolsonaro, P.R.N. Guedes, W.R. Junior, S.C. Guerra, F.J. de Araújo Abrantes, Estimates of the resident population (in Portuguese), Technical Report, IBGE, 2020.
[64] M.M. Mironczuk, J. Protasiewicz, A recent overview of the state-of-the-art elements of text classification, Expert Syst. Appl. (2018).