Assessing rule-based document segmentation and word normalization for legal ruling classification

Giliard Almeida de Godoi[a*], *Adriano* Rivolli[b] *Daniela* Lopes Freire[a], *Fabíola* Souza Fernandes Pereira[c], *Nubia* Regina Ventura[a], *Alex* Marino Gonçalves de Almeida[d], *Luís* Paulo Faina Garcia[e], *Márcio* de Souza Dias[f], *André* Carlos Ponce de Leon Ferreira de Carvalho[a]

[a]Institute of Mathematics and Computer Sciences (ICMC), University of Sao Paulo (USP) São Carlos - SP, Brazil.
E-mail: *giliard@usp.br**.

[b]Federal University of Technology - Paraná (UTFPR), Cornélio Procópio - PR, Brazil.

[c]Federal University of Uberlândia (UFU), Uberlândia - MG, Brazil.

[d]Ourinhos College of Technology, Ourinhos - SP, Brazil.

[e]University of Brasilia (UnB), Brasília - DF, Brazil.

[f]Federal University of Catalão (UFCAT), Catalão - GO, Brazil.

Submitted: 31 January 2025, Revised: 26 March 2025, Accepted: 21 April 2025, Published: 19 May 2025

Abstract. The Brazilian judiciary has shown promising interest in applying Natural Language Processing (NLP) techniques to various legal tasks. One such application is classifying legal rulings by the topics of recurring appeals. This study investigates two key strategies for preprocessing legal documents, drawing on insights from legal domain experts: whether using specific sections of the document is more effective for legal ruling classification than analyzing the entire document, and which expressions can be normalized to standardize the document vocabulary. The experimental results indicate that combining normalization preprocessing with the extraction of the judge's manifestation section yields better performance, as measured by the F1 score. Additionally, we demonstrate how the Jaccard similarity index provides valuable insight into the impact of the preprocessing pipeline on the TF-IDF feature extraction method and, by extension, on document representation. This paper underscores the importance of leveraging domain expertise to guide an optimal set of preprocessing operations.

Keywords. Legal document classification, text preprocessing operations, document segmentation and legal terms normalization.
Research paper, DOI: https://doi.org/10.59490/dgo.2025.948

1. Introduction

The Brazilian judiciary has shown a promising interest in developing system and applications that involves Natural Language Processing (NLP) tasks (CNJ, 2024). These efforts aim to process the huge amount of textual data produced by the judiciary and to improve its efficiency and overcome the heavy workload of prosecutions faced by courts across the country.

One relevant problem is the automatic classification of legal rulings by recurring appeal themes - also known as themes of general repercussion (Castro Júnior et al., 2022). These themes are complex legal questions shared by a profuse number of lawsuits and their resolution has the potential to impact thousands or even millions of similar cases. The main goal of this mechanism is to standardize legal agreement on fundamental issues and expedite the analysis of similar legal proceedings.

While the theme judgment is carried on (e.i. the underlying legal case), the higher courts may order special measures for all related cases in lower courts (e.g. the suspension or pause of all related cases). Hence, the importance of correctly identifying such lawsuits. Traditionally, the court staff must read lengthy documents to determine whether a lawsuit fits a specific theme, a time consuming task. If there is a suspension order for that theme, all the time spent on this task is misused.

This study examines how different preprocessing techniques affect the F1 score metric for the theme classification. The dataset comprises almost 30.000 legal rulings (e.i. Acórdãos) issued by the São Paulo State Court of Justice. We analyze groups of preprocessing operations, one that segment the original document in its constituents sections, and other that standardize common legal expressions to a unique pattern. Then, we compare the effectiveness of such operations by classifying the documents using common Machine Learning algorithms.

Despite the recognized importance of the preprocessing stage, the techniques and operations involved are not standardized or well established (AlMasaud et al., 2024). And the best choice of steps and operations rely heavily on empirical experimentation (HaCohen-Kerner et al., 2020). In others words, there are no common guidelines to aid in selecting the most suitable preprocessing steps.

The language commonly used in the legal domain also has its own uniqueness: a specific vocabulary with precise meanings that may differ from common usage; written documents adhere to a more standardized organizational structure; long and redundant citations of previous similar cases; and the tendency of repeating the same reasoning throughout different documents. Such characteristics should be taken into account during the preprocessing stage.

This paper investigates preprocessing strategies based on expert insights and intuitions: (1) whether using segments of the document can be more effective for its classification rather than the entire content, and (2) whether normalizing specific terms and expressions also could benefit this task. Additionally, this study highlights the importance of incorporating domain knowledge from experts to guide the choices in the preprocessing stage.

Experimental results indicate that expression normalization was the most influential factor in improving classification performance. The document segmentation did not bring the expected performance increasing that we would expected. Still, when combined expression normalization and the extraction of the judge's manifestation segment, these two preprocessing pipelines together tend to outperform the others.

Each preprocessing pipeline produces a modified version of the dataset by altering the content of the original documents. As a result, the Term Frequency–Inverse Document Frequency (TF-IDF) algorithm, which extracts a vocabulary of unique and frequent words from the data, may yield different feature sets depending on the preprocessing applied. To analyze the impact of these pipelines on the feature extraction step, we compute the Jaccard index between the vocabularies sets generated by each pair of TF-IDF models. These pairwise comparisons are then visualized in a heatmap, allowing us to assess whether different preprocessing strategies could lead to distinct textual representations of the same documents.

Therefore, the main contributions of this study are twofold: first, to highlight the importance of incorporating domain expertise in data preparation—specifically, in designing the document preprocessing pipeline; and second, to propose a deeper investigation into feature extraction models, such as TF-IDF, by analyzing their vocabulary sets and visualizing Jaccard index variations through a heatmap. Although still in its early stages, this approach suggests that examining the behavior of feature extraction algorithms may offer valuable insights into selecting the most effective preprocessing operations—potentially reducing the need to exhaustively test all combinations of individual operations directly within classification models.

2. Related works

The preprocessing stage includes operations to clean irrelevant and noisy information for a subsequent task such as text classification. Although it is recognized as a critical step, assessing the impact of different operations is often underestimated (HaCohen-Kerner et al., 2020). The lack of clear guidelines for selecting an optimal set of preprocessing operations, and the need for extensive empirical experimentation may contribute to the limited exploration of different preprocessing pipelines (AlMasaud et al., 2024).

Uysal and Gunal, 2014, HaCohen-Kerner et al., 2020 and Siino et al., 2024 agree that no single combination consistently outperforms the others, and the impact on classification effectiveness highly depends on the problem domain and datasets characteristics. They emphasize the importance of thoroughly testing all possible combinations of preprocessing operations to determine an optimal set for each specific scenario. However, the number of required experiments can increase dramatically as the number of available operations grows.

In the literature, there are no common guidelines to chose the best set of preprocessing operations. For instance, Gôlo et al., 2019 test all combinations of feature extraction techniques and hyperparameter configurations to classify documents from several domains. While Brandão et al., 2023 test a small set of preprocessing pipelines, and Silva et al., 2023 come up with an intricate rule system to find out an optimal set of preprocessing operations. None of them considers to incorporate domain expertise knowledge in the process to choose the best set of preprocessing operations.

Despite the improvement of modern Neural Network Architectures (e.g. Transformers based models) extracting numeric features using Term Frequency - Inverse Document Frequency (TF-IDF) is still a competitive option for lengthy documents, such as in the legal domain. Costa et al., 2023 employed six BERT-derived models for feature extraction in a legal document classification task, and found that the traditional TF-IDF approach produced superior results compared to the BERT-based encoding.

Araújo et al., 2021 compared several supervised machine learning techniques for classifying legal documents. The study used a simple TF-IDF approach for feature extraction. Among the models tested, SVM delivered the best performance with an F1-score of 96.4%. Other methods, including Random Forest, Adaboost, Multilayer Perceptron, and K-Nearest Neighbors, also performed well, each achieving an F1-score above 90%.

Luz de Araujo et al., 2020 proposed the VICTOR dataset for two related tasks: document type classification and theme assignment. Once the dataset originated from scanned documents, they applied regular expressions to infer the text quality and remove OCR tags and special characters. In addition, they applied preprocessing operations such as stemming, stop-word removal, lowercasing, and tokenization of emails, URLs, and legislation citations.

Noguti et al., 2020 compares traditional machine learning approaches with Neural Network Architectures. They conclude that preprocessing operations are more impactful for feature extraction when using the TF-IDF approach rather than the word embedding used by Deep Learning models.

Although Aumiller et al., 2021 explore a more advanced techniques for document segmentation, framing the problem as topical change detection, simpler approaches can be used to test an hypothesis, such as in Feijó and Moreira, 2018 where legal ruling were manually segmented for testing summarization techniques with different parts of the document.

3. Preprocessing pipelines

This section provides a concise overview of the preprocessing pipelines and their respective operations. Firstly, we describe the word-level operations that modify the text itself. Then, we present the structural organization of a legal ruling and briefly discuss how each section was extracted using simple regular expressions patterns. These pipelines are then combined to explore more possibilities of preprocessing pipelines.

3.1. Textual operation

Initially, two sets of operations are conceived: firstly, the cleaned preprocessing pipeline (C) simply removes the document header and footers, line breaks and replaces other white-spaces for a single space character. The normalization pipeline (N) converts text to lowercasing, removes word accentuation, sentence punctuation, and stopwords. The stopword list comprises 416 general Portuguese stopwords, sourced from the spaCy library, and 31 legal terms identified by domain experts. Besides, it removes all words with less than three characters.

And more importantly, it standardizes special words and expressions to a uniform representation employing 156 regular expressions, based on a vocabulary of expressions, abbreviations and acronyms from the Standard Writing Guide of the Superior Court of Justice (Brasil. Superior Tribunal de Justiça, 2016).

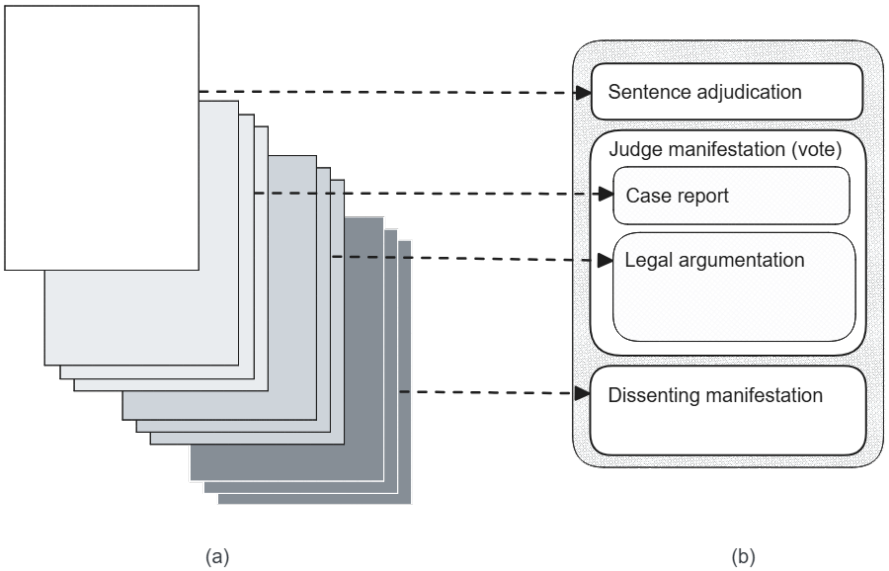

Fig. 1 – While a legal ruling can span several pages, such as illustrated in the left (a), the diagram in the right (b) shows how its internal sections are commonly organized.

For instance, both the name of the Brazilian Superior Court of Justice (e.i. Superior Tribunal de Justiça) and its abbreviation "STJ" are standardized to "`superior_tribunal_justica`." Citations of legislation and other regulatory orders (e.g. Laws, Decrees, Resolutions) are also standardized. In addition, prosecution codes, identification numbers, URLs, email addresses, monetary and percentage values, and other numeric patterns are also regularized.

Initial experiments showed that neither stemming nor lemmatization improved classification performance, so these operations were not included in the study. Furthermore, adapting these techniques to align with the regular expressions used in the normalization pipeline proved to be both complex and potentially counter-productive.

3.2. Document segmentation

In the Brazilian judiciary system, one of the primary roles of higher courts is to serve as appellate bodies, reviewing cases and rulings from lower court jurisdictions. Certain decisions are made by a panel of judges, and the ruling is referred to as an "Acórdão." Typically, the panel of judges is organized into chambers or sections, each composed of at least three members. Moreover, an authoring judge, known as the rapporteur, is appointed to analyze the case, decide on its admission, draft a formal opinion, and recommend the appropriate legal action.

For collegiate decisions, the panel of judges reviews the authoring judge's draft and decides whether to agree with it, aiming to form a majority. In cases of disagreement, the dissenting judge writes a separate opinion, presenting their reasons for disagreeing with the authoring judge's initial position . This dissenting opinion can significantly extend the document's length, as it provides counterarguments for each specific point of disagreement.

Although there is no entirely uniform structure, the legal ruling generally consists of three main parts, as illustrated in Figure 1: the adjudication of the sentence itself, the authoring judge's opinion and, eventually, a dissenting manifestation. The adjudication is found on the first page and contains the decision statement, the judges who participated in the judgment, and the authoring judge responsible for drafting the opinion. This is a more uniform section, which uses standard terms, making it ideal to extract information by applying regular expression.

The authoring judge's formal opinion, also known as vote, includes a summary of the case, references to relevant legislation, citations of previous ruling on similar issues, and suggests a legal action for the case. This

opinion is typically divided into at least two sections: the case report and the legal argumentation (see Figure 1). The case report summarizes information related to the specific case, while the argumentation section cites the legislation and precedent cases that support their decision. For similar cases, the judges tend to repeat the citations of legislation and the previous rulings, leading to documents that are alike to each other.

Legal domain experts suggest common terms and expressions that indicate the transition from one section to another. Hence, regular expressions are able to identify the beginning and the end of each section using this knowledge. Additionally, two strategies help in this process: a sequential section identification, using earlier information to aid the next section detection; and testing a wide list of regular expressions to spot each section.

For example, the authoring judge's name appears as a signature, in the last line of the decision adjudication part (first section) and also in the last line of his draft manifestation marking the beginning and the end of the judge manifestation. The first part of his manifestation is the case report, and common expressions mark the end of this section and the beginning of the legal argumentation. Notwithstanding, the authoring judge's name may differ along the signature lines, sometimes including an additional middle name or abbreviating it. In these circumstances it is necessary to programmatically adapt the regular expressions to accommodate possible variances.

It is worth mentioning the iterative refinement of each regular expression pattern. Whenever possible, two or more patterns are merged into a more powerful and efficient one. Although it shrinks the quantity of rules applied, it also increases its complexity for debugging purposes. Moreover, application of the rules goes from the more generically, that handle more expression variations, to the more restricting ones, aiming to speed up the verification procedure.

One may recognize that detecting one section depends on the success of the previous one. For instance, locating the argumentation section depends on successfully identifying the case report summary. If some segmentation is unsuccessful, we use the last identified section until we default to the entire document. For example, if we try to identify the argumentation section and it fails, we return the content of the judge's manifestation (summary case report plus the argumentation).

The segmentation strategy originates the datasets named Legal Argumentation (LA) corresponding only to the juridical argumentation, and Judge Manifestation (JM) that also includes the case report section. Both versions exclude the dissenting vote section. Furthermore, these two dataset went through the normalization preprocessing operations, obtaining its versions with expressions regularized, and indicated by signs $LA+N$ and $JM+N$, respectively.

The downside of this based-rule approach is that the regular expressions are specific to our collection of documents. Although some writing patterns are similar across judges and courts, adapting to other contexts might require additional effort. However, we use this procedure only to test the hypothesis whether using segments of the documents might be more efficient on legal ruling classification.

4. Experimental setup

In this study we evaluated at least seven machine learning algorithms across six versions of the dataset, each corresponding to a different preprocessing pipeline. The experiments were conducted in a Google Colab Pro environment. In the following we detail the dataset characteristics, the learning algorithms, and the experiment setup in detail.

4.1. The dataset

The dataset, developed under a broader research project funded by the São Paulo State Court of Justice, consists of 33,016 legal rulings considering Special Appeals in the Private and Consumer Law domains. Due to the project guidelines on sensitive information disclosure, we are not able to make the dataset publicly available.

A legal ruling might entail multiple themes, configuring a multiclass classification task. Table 1 shows the six themes of interest in this study, its definition and the number of positive instances for each class. Since we use algorithms that induce binary classification, we divided the original task into six binary classification tasks. This decision corresponds to the one-versus-all strategy, which is a feasible practice in such circumstances.

Tab. 1 – Positive class instances for each label in absolute values. Each theme is represented by a alphanumeric code. The letter S in some labels indicate whether the theme originates at STJ Brazilian Court, rather than the STF Court.

Label Code	Theme Definition	Positive Instances
S0929	Discussion on the conditions under which the double refund stipulated in Article 42 of the Consumer Defense Code (CDC) applies.	2479
S1101	Final date for the application of compensatory interest in collective and individual lawsuits seeking the restitution of inflationary purges in savings accounts.	2004
1011	Controversy regarding the legal interest of Caixa Econômica Federal in joining as a party or third-party intervenor in lawsuits involving housing loan insurance under the Housing Finance System, and, consequently, the jurisdiction of the Federal Court to process and adjudicate such cases.	1038
S1039	Establishment of the starting date for the prescription period of indemnity claims against insurers in contracts, whether active or terminated, within the Housing Finance System.	830
S1016	Validity of a contractual clause in a collective health plan that provides for adjustments based on age brackets; and burden of proof regarding the actuarial basis for the adjustment.	669
S0958	Validity of the charges in banking contracts for services provided by third parties, contract registration, and/or asset appraisal.	509

4.2. Feature extraction

Despite the improvement of modern Neural Network architectures and word embedding representation, Term Frequency - Inverse Document Frequency (TF-IDF) is still a competitive feature extraction option for lengthy document classification, as in the case of the legal domain. The scikit-learn implementation of the TF-IDF algorithms was employed. Previous experiments have shown a vocabulary size of 3,000 tokens and the use of unigrams and bigrams yields better results.

The TF-IDF algorithm selects the 3,000 tokens most frequent in corpus and computes the document frequency for each token, e.i. how many documents that token appear in the corpus. Then it calculates the term frequency of the token in each document and weighted this value by the inverse of the document frequency. Hence a document is encoded in a vector with dimension equal to the number of tokens in the vocabulary. These vector values can also be normalized by the sum of absolute values (l_1 normalization) or by the sum of squares of its elements (l_2 normalization) which was the selected option.

4.3. Classification algorithms

To evaluate the efficiency of the preprocessing pipelines, we use standard classification algorithms provided by the scikit-learn library Pedregosa et al., 2011: Decision Tree classifier (DT), Logistic Regression (LR), Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Random Forest (RF), and Extra Trees Classifier (ET). We also employed the Extreme Gradient Boosting (XGB) algorithm from the XGBoost library. We use the default parameters for all classifiers as recommended by the respective libraries.

We also employ cross-validation with five folds, using the stratified variation, and collect the precision, recall and F1-score for the test set. However, Tables 2 and 3 reports only the F1-score for conciseness. Therefore, seven machine learning algorithms were tested across six preprocessing versions of the same dataset of legal documents, for each of the six themes (or labels) considered in this study.

5. Results and discussion

Tables 2 and 3 show the average F1 scores obtained from the experiments. Table 2 contains the results solely from the document segmentation pipeline experiments, while Table 3 displays the results from the normalized

Tab. 2 – Average F1-score considering the document segmentation experiments. The preprocessing pipelines are: the cleaned text (C), legal argumentation only (LA), and all the judge manifestation (JM). The classification algorithms refer to Decision Trees (DT), Logistic Regression (LR), Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Random Forest Classifier (RF), Extra Trees Classifier (ET), and Extreme Gradient Boosting (XGB).

Label	Pipeline	DT	LR	SVM	SGD	RF	ET	XGB
S0929	C	0.8979	0.9452	0.9593	0.9506	0.9485	0.9463	**0.9686**
	LA	0.9079	0.9608	0.9748	0.9669	0.9580	0.9576	**0.9788**
	JM	0.9126	0.9613	0.9744	0.9671	0.9606	0.9607	**0.9828**
S1101	C	0.8232	0.7418	0.8304	0.8050	0.8432	0.8462	**0.9179**
	LA	0.8976	0.7909	0.9103	0.8019	0.9037	0.8842	**0.9352**
	JM	0.8843	0.7894	0.8888	0.8015	0.9051	0.8767	**0.9402**
1011	C	0.8081	0.8359	0.8611	0.8479	0.8449	0.8400	**0.8716**
	LA	0.7802	0.8456	**0.8658**	0.8462	0.8303	0.8284	**0.8634**
	JM	0.7753	0.8491	**0.8750**	0.8493	0.8492	0.8454	**0.8764**
S1039	C	0.8452	0.8944	0.9266	0.9193	0.8694	0.8612	**0.9300**
	LA	0.8530	0.8873	0.9189	0.9075	0.8369	0.8387	**0.9433**
	JM	0.8722	0.8976	0.9237	0.9241	0.8497	0.8588	**0.9493**
S1016	C	0.8878	0.9267	**0.9535**	0.9416	0.8929	0.9142	0.9395
	LA	0.8614	0.9152	**0.9348**	**0.9308**	0.8778	0.8818	**0.9326**
	JM	0.8983	0.9329	**0.9514**	0.9411	0.9061	0.9056	0.9436
S0958	C	0.6241	0.6815	0.7539	0.7252	0.6317	0.6332	**0.7816**
	LA	0.6662	0.7219	0.7828	0.7436	0.6176	0.6403	**0.7900**
	JM	0.6828	0.7293	0.7825	0.7415	0.6252	0.6386	**0.7946**

version of the datasets. The bold values indicate the highest value for each row, i.e. the highest value among the classifiers for a single pipeline. The shaded values indicate the highest value for a classifier across different pipelines (i.e. the highest value in the column) specifically for each label. Values are considered a tie if the first two decimal digits are equal–hence, all similar values are highlighted.

Table 2 suggests that there is no clear pattern indicating the best document segmentation pipeline for all classes. Nevertheless, the LA (Legal Argumentation) and JM (Judge Manifestation) preprocessing pipelines tend to perform better than the C (Cleaned) preprocessing, which considers the entire document.

The results in Table 3 indicate that the normalized version of the documents outperforms the non-normalized ones. Incorporating domain knowledge through standardized expressions is a promising approach for preprocessing legal documents. In this context, the JM (Judge Manifestation) preprocessing method performs better than the others, particularly for underrepresented classes such as 1011, S1039, S1016, and S0958.

For both cases (Table 2 and 3) the Extreme Gradient Boosting (XGB) achieved the best results in most scenarios (bolded values) followed by the SVM classifier. However, the variation between the best and worst results for the XGB algorithm is small, regardless of the preprocessing pipeline used. This suggests that XGB is not significantly affected by changes introduced through preprocessing.

The extraction of the legal argumentation segment (LA pipeline), although its high complexity, did not produce superior results than the JM pipeline. In other words, the judge's manifestation segment (i.e. removing the adjudication page and the dissenting vote declaration, when present) is preferable than the LA pipeline due to its lower complexity and cost. Additionally, associating the JM preprocessing with the expression normalization led to even better results, as shown in Table 3.

Essentially the results suggest that not all preprocessing pipeline produces significant differences on document representation. One way to visualize this situation is to plot a 2-dimension or 3-dimension scatter plot representing the embedding space, using a dimensionality reduction method such as Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE)(Cao & Cui, 2016). However, this representation produces one plot for each dataset version that is difficult to compare them directly.

Tab. 3 – Average F1-score considering the same preprocessing before, plus the normalization preprocessing. Hence, the preprocessing pipelines are: cleaned content normalized (C+N), legal argumentation normalized (LA + N), and the judge manifestation normalzied (JM + N). The classification algorithms refer to Decision Trees (DT), Logistic Regression (LR), Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), Random Forest (RF), Extra Trees Classifier (ET), and Extreme Gradient Boosting (XGB).

Label	Pipeline	DT	LR	SVM	SGD	RF	ET	XGB
S0929	C + N	0.9434	0.9559	0.9680	0.9613	0.9602	0.9587	**0.9721**
	LA + N	0.9331	0.9680	**0.9812**	0.9749	0.9634	0.9650	0.9794
	JM + N	0.9466	0.9702	**0.9834**	0.9768	0.9710	0.9698	**0.9867**
S1101	C + N	0.9041	0.7761	0.9177	0.8925	0.9241	0.9277	**0.9390**
	LA + N	0.9037	0.8081	0.9260	0.8099	0.9197	0.9215	**0.9338**
	JM + N	0.8985	0.8084	0.9277	0.8238	0.9046	0.8913	**0.9400**
1011	C + N	0.8204	0.8367	**0.8638**	0.8480	0.8475	0.8418	**0.8630**
	LA + N	0.8177	0.8457	0.8655	0.8492	0.8311	0.8380	**0.8777**
	JM + N	0.8343	0.8523	**0.8768**	0.8527	0.8542	0.8513	**0.8774**
S1039	C + N	0.9032	0.9126	0.9308	0.9285	0.9001	0.8958	**0.9495**
	LA + N	0.8850	0.9170	0.9399	0.9266	0.8729	0.8708	**0.9549**
	JM + N	0.9132	0.9209	0.9469	0.9365	0.8870	0.8914	**0.9716**
S1016	C + N	0.8887	0.9357	**0.9553**	0.9459	0.9212	0.9168	0.9438
	LA + N	0.8772	0.9202	**0.9375**	0.9331	0.9019	0.9083	**0.9338**
	JM + N	0.8893	0.9369	**0.9568**	0.9476	0.9185	0.9197	0.9422
S0958	C + N	0.7001	0.7672	0.7896	0.7791	0.6630	0.6927	**0.7926**
	LA + N	0.6904	0.7611	**0.8049**	0.7761	0.6451	0.6781	0.7854
	JM + N	0.7461	0.7829	**0.8322**	0.8005	0.6881	0.6984	0.8082

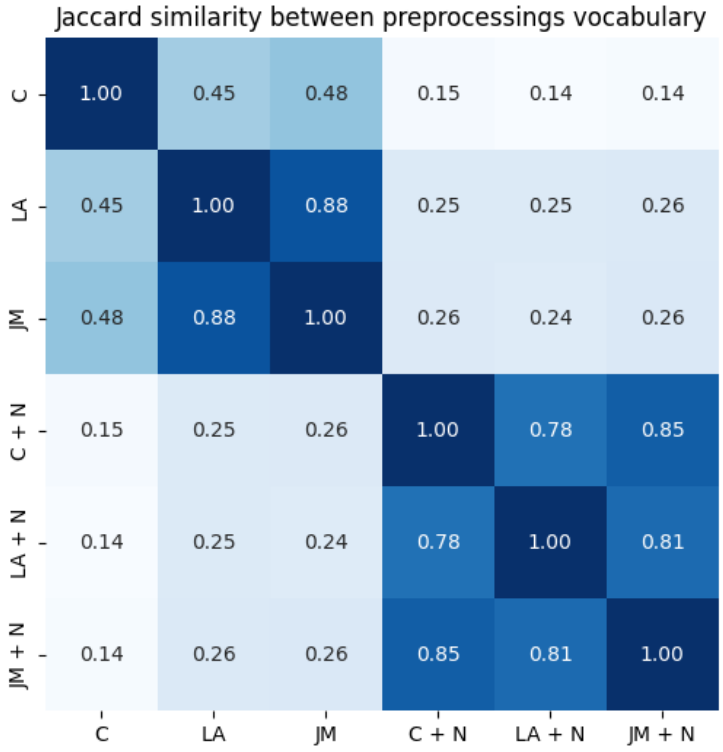

Fig. 2 – Jaccard index heatmap comparing vocabulary overlap across all pairs of preprocessing pipelines based on the TF-IDF algorithm.

Therefore, we propose analyzing the Jaccard similarity index between the vocabulary sets generated by the TF-IDF algorithm for each preprocessing pipeline. This index is calculated as the ratio of the number of elements in the intersection to the number of elements in the union of two distinct sets. Although the Jaccard index does not account for word frequency, it still offers valuable insights into how preprocessing operations influence the vocabulary computed by the TF-IDF algorithm and, consequently, the document representation.

The results are presented in Figure 2. The non-normalized preprocessing group (C, LA, and JM) and their

normalized counterparts (C+N, LA+N, and JM+N) exhibit high similarity within their respective groups but differ significantly from each other. This pattern aligns with the findings in Tables 2 and 3, where the F1-score does not vary as much within each table.

6. Conclusion

In this study, we examined the impact of various preprocessing pipelines on the classification of legal documents, focusing on judicial decisions in the Brazilian legal system. The normalization of legal expressions consistently enhanced classification performance, demonstrating the importance of tailored preprocessing in legal domain. Segmenting documents considering its constituent sections, particularly the judge's manifestation pipeline, enhance the classification performance, yielding better results than using the full text with only basic cleansing operations.

Moreover, we emphasize the importance of examining how preprocessing operations influence feature extraction models and the resulting feature sets. Although the literature provides no clear guidelines for selecting the most appropriate preprocessing strategies, analyzing their impact on feature extraction algorithms (e.g. TF-IDF) can offer valuable insights into this decision-making process. To illustrate this point, we propose investigating the vocabulary sets generated by each preprocessing pipeline by calculating the pairwise Jaccard index over the outputs of the feature extraction model. This type of analysis can be further extended through alternative visualizations and techniques, which we consider promising directions for future work.

Acknowledgement

- **Funding or Grant**: The refered project was funded by the São Paulo State Court of Justice, whom the authors are grateful for the institutional and financial support. The first author is currently supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - CAPES (process number 88887.084310/2024-00) whom would like to thank for the financial support.
- **Data/Software Access Statement**: Due to the project agreement, we are not able to release the compiled version of the dataset. However, the preprocessing pipeline and scripts for the experiments are available at https://github.com/giliardgodoi/tj-datasets.
- **Contributor Statement**: Giliard Godoi is responsible for conceptualization, methodology, investigation, software, formal analysis, visualization, and writing the original draft. Adriano Rivolli, Daniele Lopes Freire and Núbia Regina Ventura contributed to methodology discussion, software, investigation and writing (review and editing). Fabíola Souza Fernandes Pereira, Alex Marino Gonçalves de Almeida, Luís Paulo Faina Garcia, and Márcio de Souza Dias contributed to methodology discussion and writing (review and editing). André C. P. L. F. de Carvalho contributed to supervision, funding acquisition and resource, and writing (review and editing).
- **Use of AI**: During the preparation of this work, the author(s) used *ChatGPT* AI in order to improve the text fluency for the first draft. After using this tool/service, the author(s) reviewed, edited, made the content their own and validated the outcome as needed, and take(s) full responsibility for the content of the publication.
- **Conflict Of Interest (COI)**: There is no conflict of interest.

References

AlMasaud, A., Sampaio, S., & Sampaio, P. (2024). Mining Data Wrangling Workflows for Design Patterns Discovery and Specification. *Information Systems Frontiers*. DOI: https://doi.org/10.1007/s10796-023-10458-7.

Araújo, D. C., Lima, A., Lima, J. P., & Costa, J. A. (2021). A Comparison of Classification Methods Applied to Legal Text Data. In G. Marreiros, F. S. Melo, N. Lau, H. Lopes Cardoso, & L. P. Reis (Eds.), *Progress in artificial intelligence* (pp. 68–80). Springer International Publishing.

Aumiller, D., Almasian, S., Lackner, S., & Gertz, M. (2021). Structural text segmentation of legal documents. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Law*, 2–11. DOI: https://doi.org/10.1145/3462757.3466085.

Brandão, M., Silva, M., Oliveira, G., Hott, H., Lacerda, A., & Pappa, G. (2023). Impacto do Pré-processamento e Representação Textual na Classificação de Documentos de Licitações. *Anais do XXXVIII Simpósio Brasileiro de Bancos de Dados*, 102–114. DOI: https://doi.org/10.5753/sbbd.2023.231658.

Brasil. Superior Tribunal de Justiça. (2016). *Manual de padronização de textos do STJ* (2nd ed.). STJ. Brasília.

Cao, N., & Cui, W. (2016). Overview of Text Visualization Techniques. In *Introduction to text visualization* (pp. 11–40). Atlantis Press. DOI: https://doi.org/10.2991/978-94-6239-186-4_2.

Castro Júnior, A. P., Wainer, G. A., & Calixto, W. P. (2022). Application of Artificial Intelligence in the automatic identification and classification repetitive demand resolution incident in the Brazilian Court of Justice. *Revista da Faculdade de Direito da UFG*, *45*(2). DOI: https://doi.org/10.5216/rfd.v45i2.70086.

CNJ. (2024). *Justice 4.0 program* (tech. rep.). Conselho Nacional de Justiça. Brasília.

Costa, J. A. F., Dantas, N. C. D., & Silva, E. D. S. A. (2023). Evaluating Text Classification in the Legal Domain Using BERT Embeddings. In *Lecture notes in computer science* (pp. 51–63). Springer Nature Switzerland. DOI: https://doi.org/10.1007/978-3-031-48232-8_6.

Feijó, D. d. V., & Moreira, V. P. (2018). RulingBR: A Summarization Dataset for Legal Texts. In A. Villavicencio, V. Moreira, A. Abad, H. Caseli, P. Gamallo, C. Ramisch, H. Gonçalo Oliveira, & G. H. Paetzold (Eds.), *Computational processing of the portuguese language* (pp. 255–264). Springer International Publishing.

Gôlo, M., Marcacini, R., & Rossi, R. (2019). Uma extensa avaliação empírica de técnicas de pré-processamento e algoritmos de aprendizado supervisionado de uma classe para classificação de texto. *Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2019)*, 262–273. DOI: https://doi.org/10.5753/eniac.2019.9289.

HaCohen-Kerner, Y., Miller, D., & Yigal, Y. (2020). The influence of preprocessing on text classification using a bag-of-words representation (W. Zhang, Ed.). *PLOS ONE*, *15*(5), e0232525. DOI: https://doi.org/10.1371/journal.pone.0232525.

Luz de Araujo, P. H., de Campos, T. E., Ataides Braz, F., & Correia da Silva, N. (2020, May). VICTOR: a Dataset for Brazilian Legal Documents Classification. In N. Calzolari, F. Béchet, P. Blache, K. Choukri, C. Cieri, T. Declerck, S. Goggi, H. Isahara, B. Maegaard, J. Mariani, H. Mazo, A. Moreno, J. Odijk, & S. Piperidis (Eds.), *Proceedings of the twelfth language resources and evaluation conference* (pp. 1449–1458). European Language Resources Association. https://aclanthology.org/2020.lrec-1.181

Noguti, M. Y., Vellasques, E., & Oliveira, L. S. (2020). Legal Document Classification: An Application to Law Area Prediction of Petitions to Public Prosecution Service. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–8. DOI: https://doi.org/10.1109/IJCNN48605.2020.9207211.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Siino, M., Tinnirello, I., & La Cascia, M. (2024). Is text preprocessing still worth the time? A comparative survey on the influence of popular preprocessing methods on Transformers and traditional classifiers. *Information Systems*, *121*, 102342. DOI: https://doi.org/10.1016/j.is.2023.102342.

Silva, M. D., Santana, E., Lobato, F., & Jr., A. J. (2023). Preprocessing Applied to Legal Text Mining: analysis and evaluation of the main techniques used. *Anais do XX Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2023)*, 1010–1021. DOI: https://doi.org/10.5753/eniac.2023.234555.

Uysal, A. K., & Gunal, S. (2014). The impact of preprocessing on text classification. *Information Processing & Management*, *50*(1), 104–112. DOI: https://doi.org/10.1016/j.ipm.2013.08.006.