Using Complex Networks to Improve Legal Text Hierarchical Classification

Rilder S. Pires[1,2], Raquel Silveira[3], Carlos G. O. Fernandes[2,4], João A. Monteiro Neto[5], and Vasco Furtado[1,2,6]

[1] Laboratório de Ciência de Dados e Inteligência Artificial, Universidade de Fortaleza, Fortaleza, Ceará, Brasil
[2] Programa de Pós Graduação em Informática Aplicada, Universidade de Fortaleza, Fortaleza, Ceará, Brasil
[3] Instituto Federal de Educação, Ciência e Tecnologia do Ceará, Tianguá, Ceará, Brasil
[4] BNB - Banco do Nordeste do Brasil S.A., Fortaleza, Ceará, Brasil
[5] Centro de Ciências Jurídicas, Universidade de Fortaleza, Fortaleza, Ceará, Brasil
[6] ETICE - Empresa de Tecnologia da Informação do Ceará, Fortaleza, Ceará, Brasil

Abstract. Legal topics are typically organized into trees of labels, where each branch, from the root (generic topics) to the leaf (specific topics), categorizes the vocabulary that describe lawsuits. In this article, we describe an approach, innovative in the Brazilian Justice System, of automatic hierarchical classification of a petition topic. The approach integrates methods based on transformers and complex networks to capture, in addition to the characteristics of the text, the relationship between the legal citations present in the petitions and the topic to which the petition refers. The validation of this approach is done through a benchmark that shows accuracy gains, as well as, a practical scenario with the implementation of a microservice on a National Justice Platform whose front-end implementation is already being used by a State Court to automatically suggest the classification of the petitions topic in the National Procedural System.

Keywords: Text Classification · Brazilian Legal System · Complex Networks.

1 Introduction

Classification of the topic addressed by a legal document is a task where Artificial Intelligence, specifically through Natural Language Processing (NLP), can provide automatic inference capabilities [21]. Legal topics are typically organized into hierarchical trees, where each branch, from the root (e.g., consumer law) to the leaf (e.g., moral/material damage), categorizes the vocabulary that describes lawsuits. Topic classification generally occurs in the early stages of the judicial process when a petition is submitted to the court. At this point, the petitioner is required to indicate the relevant subject matter of the demand. In the Brazilian context, a petitioner must choose the topic from a hierarchy encompassing over

4,000 subjects, which are part of the Unified Procedural Tables (TPU system) maintained by the National Council of Justice (CNJ) [9]. Making the correct association within this hierarchy is not a trivial task and is often done incorrectly by the petitioner. This misclassification generates delays in the judicial process, leading to negative impacts both financially, due to rework, and socially, by fostering a sense of impunity.

The scientific literature presents various methodologies for the automatic classification of judicial documents [21, 29, 4, 14, 2, 1]. The advent of textual classification via deep learning, particularly with transformer models, has significantly advanced the field of text classification [10, 17, 28, 31]. Nonetheless, the intricate and extensive hierarchical structure of topics, along with the specialized nature of legal texts, poses significant challenges to traditional methods. Taking a closer look at our problem, we see that it is a hierarchical text classification problem, where we typically aim to categorize a text into a set of labels organized in a structured class hierarchy. The big challenge with this type of task is to adequately model the label hierarchy, which is usually large-scale, unbalanced, and highly structured [34]. These challenges have motivated us to develop an innovative approach.

Our approach assumes that the numerous legal citations present within petitions are crucial for understanding the document's content and defining its relevant subject matter. These legal citations not only help formalize the arguments presented by both petitioners and decision-makers but also serve as a bridge connecting the petition to the applicable legal provisions, such as laws, decrees, precedents, and articles. By examining the citations within each petition, we can construct a network linking legal provisions to the petition topics. This network offers supplementary information to enhance the automatic classification process, thereby significantly improving its accuracy.

Using this network-based approach, we developed a classifier based on BERT that leverages not only the petition text but also vector information generated from the network connections between topics and legal provisions. We validated our methodology with a dataset comprising 300,000 petitions from various Brazilian courts, maintained by the CNJ's Codex system. The results demonstrate the validity of our approach, showcasing significant gains in accuracy, in some situations, when compared to a hierarchical classification method that do not use the network information. Additionally, we show a example of application of the proposed approach in a version of a hierarchical classification model that is available on the SINAPSE platform of the Council and is currently being used by a State Court to automatically suggest topic classifications for petitions entered in the National Procedural System.

2 Related Work

The modeling of complex networks based on the characteristics of a lawsuit has its potential evidenced in several works as described in [27, 11, 5, 13, 25, 24]. Classification of legal documents has been used in the forensic and Law Enforcement

context (e.g. identifying potential criminals [32], predicting crimes [22, 23] or discovering antisocial behavior [7]).

Mariana et al. [21] investigated, in the Brazilian context, different methods of classification and data representation to conclude that deep learning (more specifically) with Word2Vec trained on a domain-specific corpus, are promising, but still limited, methods for classifying many classes.

Luz de Araujo et al. [4] presented a new dataset constructed from legal documents from the Federal Supreme Court of Brazil, containing labeled text data for two types of tasks: document type classification and assignment of themes. The authors explored and compared different methods for classification, such as bag-of-words models, CNN, RNN, boosting algorithms, and linear-chain Conditional Random Fields. CNN with word embedding also proved to be more performing. Silva et al. [30] proposed a structured model in a CNN to classify types of documents received by the Federal Supreme Court of Brazil. Aguiar et al. [1] investigated the application of topic modeling to identify the subject of legal documents and evaluated the applicability in classifying Brazilian legal processes. The classification model was trained on documents from the Court of Justice of Ceará, in Brazil, and the cases were classified into the five most representative classes of the National Council of Justice (CNJ) of Brazil. Recently, pre-trained models based on BERT has shown to be more performing to classify topics. Importantly, the effectiveness of BERT is mainly due to the transfer learning ability that leverages semantic and syntactic knowledge from pre-training on a large non-labeled corpus [10]

Aguiar et al. [2] investigate different text classification methods and different combinations of embeddings, extracted from Portuguese language models, and information about legislation cited in the initial documents to classification task of lawsuits. The models were trained with a Golden Collection from the Court of Justice of the State of Ceará, in Brazil, whose lawsuits were classified in the five more representative CNJ's classes. The best result was obtained by the BERT model. Shaheen et al. [29] studied the performance of several transformer-based models (namely BERT [10], RoBERTa [17], DistilBERT [28] and XLNet [36]) in combination with strategies such as pre-generative training, gradual unfreezing and discriminating learning rates to achieve competitive ranking performance. Chalkidis et al. [6] developed an English legal judgment prediction dataset, containing cases from the European Court of Human Rights. A wide variety of neural models are evaluated on this dataset, establishing strong baselines that outperform previous feature-based models.

Wang et al. [34] presented the Hierarchy-aware Prompt Tuning (HPT) method, aimed at addressing hierarchical text classification from the perspective of a masked language model with multiple labels. The method involves creating a dynamic virtual template and using label words that act as soft prompts, integrating knowledge of the label hierarchy. Additionally, a zero-bounded multi-label cross-entropy loss is introduced to align the processes of hierarchical text classification with masked language models. The authors conducted experiments

that demonstrated HPT achieves a good performance on several popular hierarchical text classification datasets.

3 The Proposed Approach

In the proposed approach, we assume that the primary source of data is a set of textual petitions P that make up the initial piece of the judicial case. Each initial petition is associated with a *legal topic* that the case addresses. In the text of these petitions, there are references to *legal provisions* that support the arguments put forward by the petitioner. Our hypothesis is that the relationship of a *legal topic* with one or more *legal provisions* is relevant for automatically classifying the matter of a petition.

Figure 1 shows the pipeline of the proposed approach, which depends on the texts of the petitions and the citations to legislation made within them. Additional background knowledge about the hierarchical structure of legal topics and provisions is used. This additional information and the citations of legal provisions within the text form a complex network of topics and legal provisions that we will model here through a weighted graph. Then, the embedding of vertices in this graph allows us to generate a vector representation of the petitions. Next sections will provide formalization of these concepts.

3.1 Bipartite Graph of Legal Provisions and Topics

Given a set of petitions P, we define G_B as the graph that represents the *co-occurrence* relationship between the *legal topics* of the *petitions* of P and the

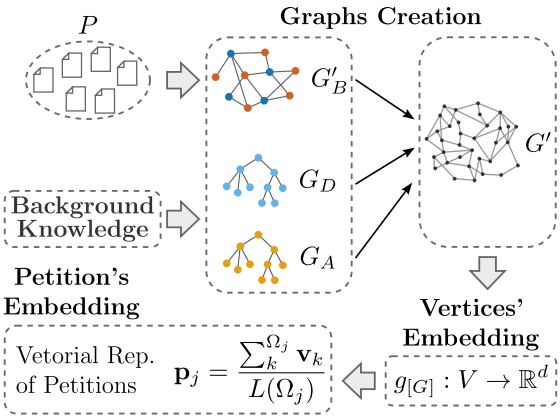

Fig. 1. Pipeline of the proposed approach. In the upper left corner are the primary source of data namely the petitions and the legal citations made within them as well as background knowledge about the hierarchical structure of legal topics and provisions. This data is transformed in a weighted graph of legal topics and provisions that allows to generate a vector representation of the petitions.

respective *legal provisions* cited therein. Here, we define $G_B = (A, D, E_B, w_B)$, where

- $A = \{a_1, a_2, \ldots, a_n\}$, $n \in \mathbb{N}$, is a finite set of *vertices* related to legal topics.
- $D = \{d_1, d_2, \ldots, d_m\}$, $m \in \mathbb{N}$, is a finite set of *vertices* related to the legal provisions
- $E_B \subset A \times D$ is a finite set of *edges* connecting subsets of vertices of A and D.
- $w_B : E_B \to \mathbb{R}$ is a *edge weight function* defined on the edges of G_B.

In this graph, there is an edge between the vertices a_i and d_j if there is at least one petition in P whose *legal topic* corresponds to a_i and which makes at least one reference to *legal provision* d_j. Generally speaking, we assume that the weight $w_{Bi,j}$ of this edge will be a function of the number of times $q_{i,j}$ that the legal provisions d_j is referenced by the petitions of the topic a_i to P.

The G_B graph bears similarities with other graphs already studied in the literature, such as the bibliographic citations graph [19]. There, the vertices represent documents that make references to other documents. An important characteristic of this type of graph is the existence of documents with a high amount of citations [19]. These vertices with many links end up making it difficult to identify the clustering patterns existing in the graph [3].

Similarly to what happens in *bipartite graphs of users and objects*, we define a weight function that uses a strategy based on a popular technique in *information retrieval* [3] called *tfidf*,

$$w_{Bi,j} \equiv w_B(a_i, d_j) = f(a_i, d_j) \times \log\left(\frac{n}{\deg(d_j)}\right) \tag{1}$$

where $f(a_i, d_j)$ is similar to "tf" and represents the frequency that the legal provision d_j is cited in petitions of a certain topic a_i,

$$f(a_i, d_j) = \frac{q_{i,j}}{\sum_k^{\langle a_i \rangle} q_{i,k}},$$

and $\langle \cdot \rangle$ means that the sum in k is done at the *neighborhood* of a_i. The term $\log(n/\deg(d_j))$ of Eq. (1) is similar to "idf", measuring how much the legal provision is rare or common among all legal topics.

3.2 Forests of Legal Topics and Legal Provisions

Legal topics have taxonomic hierarchical relationships and are predefined in the *Unified Procedural Tables* [9] system. The graph $G_A = (A, E_A)$ represents the *taxonomy* of *legal topics*, where:

- $A = \{a_1, a_2, \ldots, a_n\}$, $n \in \mathbb{N}$, is a finite set of *vertices* related to legal topics.
- $E_A \subset A \times A$ is a finite set of *directed edges* that connect the vertices of G_A.

The intrinsic asymmetry of the relationships between the vertices of G_A makes it fall into the class of *directed rooted forests*. Such graphs have important properties from a topological point of view. In these graphs, we can define two types of special vertices, the *root* vertices and the *leaf* vertices. Considering that the connections between two vertices always occur from the most general legal subject to the most specific subject, a root vertex is defined as a vertex that only has connections "going" from it. Complementary, a leaf vertex can be defined as a vertex that only has links "entering" it.

From the point of view of legal topics, we can interpret leaf vertices as the most specific topics existing in G_A and root vertices as the most generic ones in G_A. Furthermore, the topological structure of *directed rooted trees* allows us to define for each leaf vertex a_i a unique corresponding root vertex $\text{root}(a_i)$, that is, for each specific subject a unique corresponding generic subject.

Similarly to what happens with topics, legal provisions have hierarchical relationships that allow us to define a graph $G_D = (D, E_D)$, where

- $D = \{d_1, d_2, \ldots, d_n\}$, $m \in \mathbb{N}$, is a finite set of *vertices* describing legal provisions.
- $E_D \subset D \times D$ is a finite set of *directed edges* that connect the vertices of G_D representing the *part-of* relationship between the pieces that make up a law.

Although semantically different from the previously described taxonomy, the asymmetry of the relationships between the vertices of G_D also classifies it as a *directed rooted forest*. Legal provisions are hierarchically organized as a *partonomy*, where the vertices of the root type represent complete legal provisions, such as Laws, Decrees and Precedents that are composed of parts such as sections, articles, paragraphs and items. For each leaf vertex d_i there is, also, a single corresponding root vertex $\text{root}(d_i)$.

3.3 Weighted Graph of Legal Topics and Provisions

Given a set of initial petitions P, we can define a *bipartite graph of topics and legal provisions* $G_B = (A, D, E_B, w)$ as shown in Sec. 3.1. Furthermore, the background knowledge of legal topics allows us to build a *forest of legal topics* $G_A = (A, E_A)$ as shown in Sec. 3.2. Thus, we can define a new bipartite graph $G'_B = (A_{\text{leaf}}, D, E'_B, w'_B)$ as the subgraph of G_B, where:

- A_{leaf} is the subset of *vertices* that are of *leaf type* in G_A representing specific topics.
- D is the set of *vertices* of G_B representing the legal provisions.
- E'_B is the subset of *edges* of G_B where $w_{i,j} \geq \tau$.
- $w'_B : E' \to \mathbb{R}$ is an *edge weighting function* defined in E' as $w'_{Bi,j} = w_{Bi,j}$.

Here, τ corresponds to a threshold of *tfidf* below which we consider that an edge does not justify being created in G'_B. It is important to note that, unlike to what happens in G_B,, in G'_B there are only vertices related to legal topics of leaf type. Now, we can define the *weighted graph of legal topics and provisions* $G = (V, E, w)$, through the junction of G'_B and the undirected versions of the forests $G'_A = (A, E'_A)$ and $G'_D = (D, E'_D)$, where:

- $V = A \cup D = \{v_1, v_2, \ldots, v_{n+m}\}$ is the set of *vertices* of G', defined as the union of the vertices related to topics and legal provisions.
- $E = E'_B \cup E'_A \cup E'_D$ is the set of *edges* connecting the vertices in V.
- $w : E \to \mathbb{R}$ is an *edge weighting function* defined in E as:

$$w_{i,j} = \begin{cases} w'_B(v_i, v_j) \text{ case edge } (v_i, v_j) \in E'_B \\ \tau \qquad\qquad\qquad \text{otherwise} \end{cases}$$

In order to the condition $(v_i, v_j) \in E'_B$ to be satisfied, necessarily, the conditions $v_i \in A$ and $v_j \in D$ must also be satisfied, since G'_B is a bipartite graph. As defined before, the graph G is a simple *weighted graph*. This allows us to generate vector representations for the vertices of this graph, as we will see in the next subsection.

3.4 Generating Embeddings

In this section, we describe the process of generating the vector representation of legal citations from *weighted graph of topics and legal provisions* and the embeddings of petitions that, in turn, use these representations.

Generating vertices' embeddings of *G*. In this step, we use the graph presented in the previous section to generate the embeddings of its vertices. For this, we define the mapping function

$$g_{[G]} : V \to \mathbb{R}^d,$$

which associates to each vertex $v_i \in V$ a vector $\mathbf{v}_i$ of d dimensions. The intuition of this approach is that, in this space, the vector $\mathbf{v}_i$ will be close to vertices belonging to the *neighborhood* of v_i. Then, legal topics will be "placed" close to legal provisions that occur concurrently in petitions and/or next to more general topics in the topic hierarchy. The citation embeddings aforementioned can be seen as an alternative way, other than the natural language text, of representing the arguments used by the petitioner. In this sense, legal provisions are cited to represent and support the matter of a petition and are relevant to determine the legal topic the petition refers to. In order to generate the embedding for G, we used the *node2vec+* algorithm implemented in *pecanpy* [16] library, which was chosen for its ability to capture conformal communities [15].

Generation of a petition's embeddings. From the vector representation of each legal citation made in a given petition, the petition's embedding can be calculated. The representation $\mathbf{p}_j$ for a petition from the legal provisions it cites is defined as

$$\mathbf{p}_j = \frac{\sum_k^{\Omega_j} \mathbf{v}_k}{L(\Omega_j)},$$

where the sum is performed on the set Ω_j of all representations corresponding to the legal provisions mentioned in the petition of the process j and $L(\Omega_j)$ is the total number of elements in Ω_j.

3.5 The Classifier

The petition set P is the main input of the classifier, as illustrated in the architecture of our model in Fig. 2. The text of the initial petition goes through a textual standardization phase with the objective of standardizing legal terms (for example, art. 1 of Law No. 11,419 of DECEMBER 19, 2006 to art. 1 of Law 11,419/2006), removing excess spaces, repeated lines and special characters and extending acronyms of agencies and states of the federation. Next, the legal provisions cited in the petition are identified using regular expressions and then we proceed as described in the previous section in order to generate the petition embedding.

The automatic classification is done according to a Hierarchy-aware Prompt Tuning architecture [34]. At a high level, the standardized text of the application is sent to the BERT module, whose architecture is structured in a multi-layered bidirectional Transformer encoder [10]. The input text is encoded in a sequence of tokens generated using WordPiece embeddings [35]. The textual features are then passed to a stack of encoder layers, after augmentation with positional encoding. Positional embeddings are added to each layer of the transformer's encoder to help the transformer learn dependencies between tokens. After applying the transformer coding layers, the BERT module results in a linear pooling layer.

The combination module receives as input the text resources emitted from the Transformer BERT model and the numerical resources coming from the "vector representation of legal citations". From there the module generates a combined multimodal representation. We use the combination module proposed by [34], which receives as input the textual features generated from a Transformers model and numerical features and generates a combined representation, through different methods that combine the different representations in their respective spaces of features.

To incorporate the label hierarchy, we propose a layer-wise prompt (according to [34]). Since the label hierarchy is a tree, we construct templates based on the depth of the label hierarchy. The hierarchy constraint only introduces depth of labels but lacks their connectivity. To make full use of the label hierarchy in an MLM-manner, we further inject the per-layer label hierarchy knowledge into template embedding. A K-layer stacked Graph Attention Network (GAT) [12] is adopted to model the label hierarchy.

Since hierarchical text classification is a multi-label classification problem, in this paper, instead of calculating the score of each label separately, we expect the scores of all target labels to be greater than those of all non-target labels. We use a multilabel cross entropy (MLCE) loss [33], introducing an anchor label with a constant score of 0 in MLCE and hope that the scores of the target labels and non-target labels are all greater and less than 0 respectively. Thus, we form a zero-bounded multi-label cross entropy (ZMLCE) loss. To be consistent with the hierarchy constraint, we adopt ZMLCE at each label hierarchy layer for the layer-wise prediction.

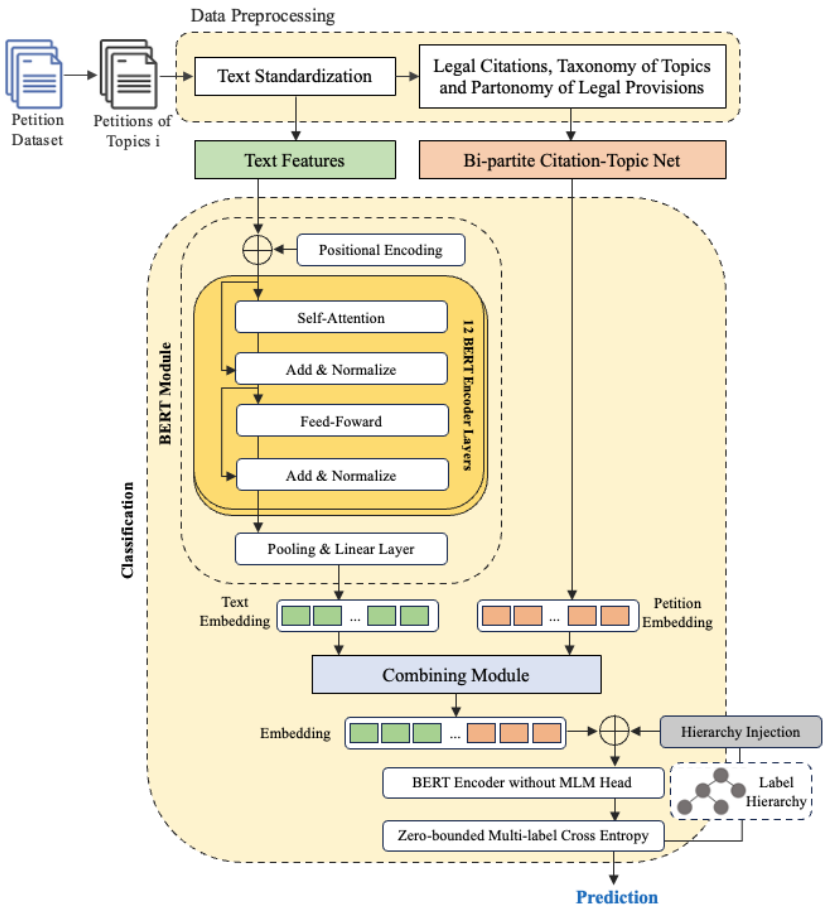

Fig. 2. Architecture of our model. The architecture is divided into two stages: (i) data preprocessing: the text of the initial petition is pre-processed to standardize legal terms and to identify the articles and laws cited in the text, and (ii) classification: combines the input data (text and bi-partite citation topic-net) and uses the architecture of Hierarchy-aware Prompt Tuning (HPT) [32] during training. HPT transforms hierarchical text classification (HTC) into a hierarchy-aware multi-label masked language model (MLM) problem, incorporing the label hierarchy knowledge. HPT transforms HTC into a multi-label MLM task with a zero-bounded multi-label cross entropy loss.

4 The Classifier in Use

The performance evaluation of the model was made from its application in real data of 300,000 petitions of legal processes coming from Labor Courts, Federal Regional Courts and Courts of Justice of the Brazilian states.

4.1 The Petition Dataset

As already stated in section 3.2, legal topics are categorized according to a specific table of topics called *Unified Procedural Table* TPU [9] developed by the *Brazilian National Council of Justice* CNJ. The hierarchy present in these tables allows us to aggregate specific topics into generic ones. We used this strategy to separate our dataset into very distinct topic groups, which allowed us to test our approach at different levels of complexity.

In table 1, we show the percentage distribution of initial petitions for each generic subject in the dataset of 300,000 petitions.

Generic Topic	Lawsuits (%)
14 - Tax Law	4.4
195 - Pension Law	4.7
287 - Criminal Law	5.7
864 - Labor Law	11.1
899 - Civil Right	29.0
1156 - Consumer Rights	25.2
1209 - Criminal Procedural Law	0.4
8826 - Civil Procedural and Labor Law	13.0
9985 - Administrative Law and other matters of Public Law	6.3

Table 1. Percentage distribution of lawsuits by legal topics. The codes used in this table are defined by the TPU and are the same ones that will be used in the results section.

The set of 300,000 petitions was then divided following a ratio of 80% for training and 20% for testing. We took care that the dataset only contained petitions with at least 50 words and that more than 80% of valid words were valid[7].

4.2 Training Process

The graph G was created from the legal provisions identified in the set of petitions P of the training data. Then, the vector representation of the petition citations was generated as described in Sec. 3.4. The value $\tau \approx 8.48 \cdot 10^{-3}$ was empirically adjusted by identifying the value that best separated the topics from G_B. This was done by analyzing the *assortativity coefficient* [20] of the one-mode projection [19] relative to the *legal topic* partition.

A fine-tuning of the pre-trained BERTimbau [31] model was done, where all parameters were tuned using the specific topics of the petitions used in the training. In the mesun_etal_2020rge module, we use the gated summation merge method in the transformer output for the petition text and the vector

[7] The verification of valid words was performed using the lexicons proposed by [18]

representation of the citation, before the final classifier layer[8]. The model was trained with batch size of 16 examples, 30 epochs max and a early stopping of 5 epochs. Adam Optimizer was chosen with learning rate of 10^{-5} and 5-fold cross-validation. Random oversampling was applied in order to equalize number of examples of each class.

4.3 Comparative Evaluation and Deployment

For a detailed evaluation of our approach, we compare our model (descibed in Sec. 3.5) with the original hierarchical classification method [34] (that uses only text as input). The results for each specific topic were calculated in terms of Precision, Recall and F1-score, obtained from the actual and predicted labels on the test dataset. In Tab. 2, we show the results of the classifiers for the different generic topics shown in the previous section.

Generic		Only Text			Text + G		
Topic	N	Precision	Recall	F1-score	Precision	Recall	F1-score
14	9	**0.899**	**0.835**	**0.859**	0.898	0.833	0.857
195	13	**0.743**	**0.719**	**0.725**	0.709	0.675	0.690
287	12	**0.829**	**0.795**	**0.811**	0.823	0.791	0.806
864	22	0.528	0.487	0.488	**0.556**	**0.538**	**0.541**
899	38	0.749	0.635	0.680	**0.779**	**0.637**	**0.692**
1156	24	**0.664**	**0.599**	0.622	**0.664**	0.597	**0.625**
1209	3	0.694	**0.395**	**0.484**	**0.735**	0.364	0.462
8826	22	**0.823**	0.491	0.603	0.807	**0.521**	**0.624**
9985	10	**0.766**	**0.586**	**0.649**	0.758	0.548	0.622

Table 2. Precision, Recall and F1-score results for different generic topics. Here, N is the number of specific topics present in each generic topic. The metrics show the comparison between the classifier that uses only the text of the petitions and the one that uses the text and the vector representation defined by our approach.

The results in this table show that the classifier that uses information from the citations performs better in some situations. Precisely, it shows better results in "864 - Labor Law", "899 - Civil Right" and "1209 - Criminal Procedural Law" generic topics. We believe that, in these cases, our approach is able to better capture the relationships that exist between legal provisions and specific topics due to the existence of legal provisions that detail several specific situations in these branches of Law.

An example of an application where this approach is very useful occurs when a petitioner only informs a generic topic when submitting the petition. In this case, a meta-model would be capable of identifying which approach is most

[8] This merge method is inspired by [26].

suitable for classifying the specific subject of that petition based on the generic topic informed.

Indeed, we incorporated a version of a hierarchical classification model into the CNJ's SINAPSES platform [8] in the form of a micro-service. This platform aims to share, among Brazilian judicial institutions, models of artificial intelligence developed by the public and private sector. In Figure 3, we shows the interface of the judicial process registration system, PJe, after the implementation of the leaf topic prediction functionality with the call of the classification micro-service implemented in the SINAPSES platform.

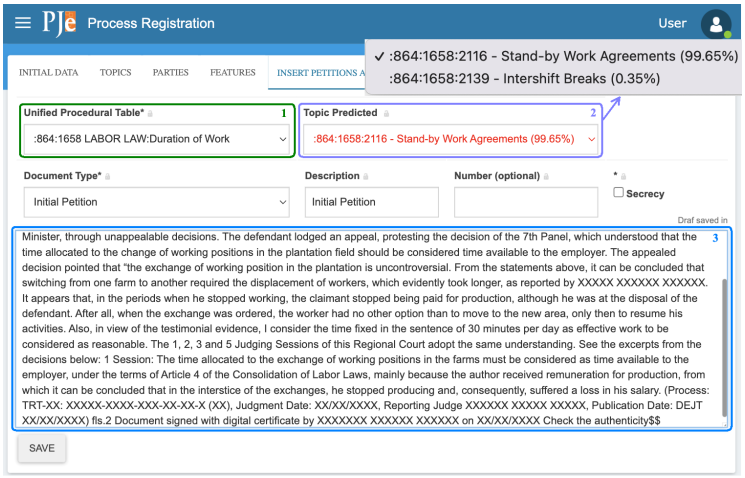

Fig. 3. Screenshot of PJe system with the feature of predicting the topic of a petition. The screenshot is numbered to show: 1) is the main topic informed by the user, 2) is the predicted specific topic and 3) is the text of the petition.

5 Conclusion

Citations to existing legal provisions in lawsuits is an important particularity of this type of document. They add additional information to the text and can be used by automatic natural language processing methods for a variety of tasks. In this article we describe how the automatic topic classification of a legal petition can be enhanced based on these citations. The approach integrates transformer-based methods with a complex network in order to capture, in addition to text features, the relationship between legal citations that are present in the petitions and the topic to which the petition refers to.

This research work characterizes itself by the proposition of a novel method based on the combination of Deep Learning and Complex Networks, and in the

application of these ideas in the real world. The implementation of a micro-service into the Justice National Platform as well as the implementation of a front-end that uses this service constitutes a novel tool to be used country wide by different institutions of the Brazilian Justice. It is expected bringing productivity augmentation, reduction of costs and more agility to the judicial sentences.

Future work involves improving accuracy of the method via fine-tuning of parameters and techniques used to manipulate the network. For instance, investigations of alternative methods for graph embedding generation for capturing temporal relations may improve the final results. Also, monitoring the use of the classifier by the different institutions may shed lights on the user experience and the real benefits the approach brings to the citizen.

Acknowledgments. We gratefully acknowledge CNPq, CAPES, FUNCAP, BNB, and the Edson Queiroz Foundation for financial support.

References

1. Aguiar, A., Silveira, R., Furtado, V., Pinheiro, V., Neto, J.A.M.: Using topic modeling in classification of brazilian lawsuits. In: Pinheiro, V., Gamallo, P., Amaro, R., Scarton, C., Batista, F., Silva, D., Magro, C., Pinto, H. (eds.) Computational Processing of the Portuguese Language. pp. 233–242. Springer International Publishing, Cham (2022)
2. Aguiar, A., Silveira, R., Pinheiro, V., Furtado, V., Neto, J.A.: Text classification in legal documents extracted from lawsuits in brazilian courts. In: Anais da X Brazilian Conference on Intelligent Systems. SBC, Porto Alegre, RS, Brasil (2021), https://sol.sbc.org.br/index.php/bracis/article/view/19093
3. Alupoaie, S., Cunningham, P.: Using tf-idf as an edge weighting scheme in user-object bipartite networks. arXiv preprint arXiv:1308.6118 (2013)
4. Luz de Araujo, P.H., de Campos, T.E., Ataides Braz, F., Correia da Silva, N.: VICTOR: a dataset for Brazilian legal documents classification. In: Proceedings of the 12th Language Resources and Evaluation Conference. pp. 1449–1458. European Language Resources Association, Marseille, France (May 2020), https://aclanthology.org/2020.lrec-1.181
5. Carmichael, I., Wudel, J., Kim, M., Jushchuk, J.: Examining the evolution of legal precedent through citation network analysis. NCL Rev. **96**, 227 (2017)
6. Chalkidis, I., Androutsopoulos, I., Aletras, N.: Neural legal judgment prediction in english. CoRR **abs/1906.02059** (2019), http://arxiv.org/abs/1906.02059
7. Cheng, J., Danescu-Niculescu-Mizil, C., Leskovec, J.: Antisocial behavior in online discussion communities. CoRR **abs/1504.00680** (2015), http://arxiv.org/abs/1504.00680
8. CNJ: Sinapses. https://sinapses.ia.pje.jus.br/ (2022), [Online; accessed 9-August-2022]
9. CNJ: Sistema de gestão de tabelas processuais unificadas. https://www.cnj.jus.br/sgt/consulta_publica_assuntos.php (2022), [Online; accessed 9-August-2022]
10. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of

the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-1423, https://doi.org/10.18653/v1/n19-1423
11. Fowler, J.H., Johnson, T.R., Spriggs, J.F., Jeon, S., Wahlbeck, P.J.: Network analysis and the law: Measuring the legal importance of precedents at the us supreme court. Political Analysis **15**(3), 324–346 (2007)
12. Kipf, T.N., Welling, M.: Semisupervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations (ICLR) (2017)
13. Koniaris, M., Anagnostopoulos, I., Vassiliou, Y.: Network analysis in the legal domain: A complex model for european union legal sources. Journal of Complex Networks **6**(2), 243–268 (2018)
14. Limsopatham, N.: Effectively leveraging BERT for legal document classification. In: Proceedings of the Natural Legal Language Processing Workshop 2021. pp. 210–216. Association for Computational Linguistics, Punta Cana, Dominican Republic (Nov 2021). https://doi.org/10.18653/v1/2021.nllp-1.22, https://aclanthology.org/2021.nllp-1.22
15. Liu, R., Hirn, M., Krishnan, A.: Accurately modeling biased random walks on weighted graphs using node2vec+. arXiv preprint arXiv:2109.08031 (2021)
16. Liu, R., Krishnan, A.: Pecanpy: a fast, efficient and parallelized python implementation of node2vec. Bioinformatics **37**(19), 3377–3379 (2021)
17. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019), http://arxiv.org/abs/1907.11692
18. Muniz, M.: A construcão de recursos linguistico-computacionais para o português do Brasil: o projeto Unitex-PB. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo (01 2004). https://doi.org/doi:10.11606/D.55.2020.tde-19022020-151305
19. Newman, M.: Networks. Oxford university press (2018)
20. Newman, M.E.: Mixing patterns in networks. Physical review E **67**(2), 026126 (2003)
21. Noguti, M.Y., Vellasques, E., Oliveira, L.S.: Legal document classification: An application to law area prediction of petitions to public prosecution service. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (2020). https://doi.org/10.1109/IJCNN48605.2020.9207211
22. Pérez-Rosas, V., Mihalcea, R.: Experiments in open domain deception detection. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 1120–1125. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015). https://doi.org/10.18653/v1/D15-1133, https://aclanthology.org/D15-1133
23. Pinheiro, V., Pequeno, T., Furtado, V., Nogueira, D.: Information extraction from text based on semantic inferentialism. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) Flexible Query Answering Systems. pp. 333–344. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
24. Pires, R.S., Oliveira, E.A., Almeida, V.F., Neto, J.A.M., Furtado, V.: Análise da ontologia dos assuntos jurídicos e suas respectivas legislações através de redes complexas. In: Anais do XI Brazilian Workshop on Social Network Analysis and Mining. pp. 181–191. SBC (2022)

25. Pires, R.S., Oliveira, E.A., Fernandes, C.G., Neto, J.A.M., Furtado, V.: Mapping landmark cases in the us legal system. CEUR Workshop Proceedings (2021)
26. Rahman, W., Hasan, M.K., Lee, S., Bagher Zadeh, A., Mao, C., Morency, L.P., Hoque, E.: Integrating multimodal information in large pretrained transformers. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 2359–2369. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.214, https://aclanthology.org/2020.acl-main.214
27. Ruhl, J., Katz, D.M., Bommarito, M.J.: Harnessing legal complexity. Science **355**(6332), 1377–1378 (2017)
28. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR **abs/1910.01108** (2019), http://arxiv.org/abs/1910.01108
29. Shaheen, Z., Wohlgenannt, G., Filtz, E.: Large scale legal text classification using transformer models. CoRR **abs/2010.12871** (2020), https://arxiv.org/abs/2010.12871
30. Silva, N., Braz, F., de Campos, T.: Document type classification for brazil's supreme court using a convolutional neural network. In: 10th International Conference on Forensic Computer Science and Cyber Law (ICoFCS), Sao Paulo, Brazil. pp. 7–11 (10 2018). https://doi.org/10.5769/C2018001
31. Souza, F., Nogueira, R., Lotufo, R.: Bertimbau: Pretrained bert models for brazilian portuguese. In: Cerri, R., Prati, R.C. (eds.) Intelligent Systems. pp. 403–417. Springer International Publishing, Cham (2020)
32. Sumner, C., Byers, A., Boochever, R., Park, G.J.: Predicting dark triad personality traits from twitter usage and a linguistic analysis of tweets. In: 2012 11th International Conference on Machine Learning and Applications. vol. 2, pp. 386–393 (2012). https://doi.org/10.1109/ICMLA.2012.218
33. Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., Wei, Y.: Circle loss: A unified perspective of pair similarity optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
34. Wang, Z., Wang, P., Liu, T., Lin, B., Cao, Y., Sui, Z., Wang, H.: Hpt: Hierarchy-aware prompt tuning for hierarchical text classification. arXiv preprint arXiv:2204.13413 (2022)
35. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J.: Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR **abs/1609.08144** (2016), http://arxiv.org/abs/1609.08144
36. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. CoRR **abs/1906.08237** (2019), http://arxiv.org/abs/1906.08237