










Text Classification in the Brazilian Legal Domain

Gustavo M. C. Coelho¹^a, Alimed Celecia¹^b,
Jefferson de Sousa¹^c, Melissa Cavaliere¹^d, Maria Julia Lima¹^e, Ana Mangeth²^f, Isabella
Frajhof²^g, Cesar Cury³^h, Marco Casanova¹ⁱ

¹*Tecgraf - PUC-Rio - Rio de Janeiro - Brazil*
{gustavocoelho, alimed, alvesjefferson, melissa, mjulia}@tecgraf.puc-rio.br, casanova@inf.puc-rio.br

²*LES - PUC-Rio - Rio de Janeiro - Brazil*
{ana.mangeth, isabella.zfrajho}@les.inf.puc-rio.br

³*Escola da Magistratura do Estado do Rio de Janeiro - Rio de Janeiro - Brazil*
cesarcury.com@gmail.com

Keywords: Document Embedding, Text Classification, Natural Language Processing.


Abstract: Text classification is a popular Natural Language Processing task that aims at predicting the categorical values associated with textual instances. One of the relevant application fields for this task is the legal domain, which involves a high volume of unstructured textual documents. This paper proposes a new model for the task of classifying legal opinions related to consumer complaints according to the moral damage value. The proposed model, named MuDEC (Multi-step Document Embedding-Based Classifier), combines Doc2vec and SVM for feature extraction and classification, respectively. To optimize the classification performance, the model uses a combination of methods, such as oversampling for imbalanced datasets, clustering for the identification of textual patterns, and dimensionality reduction for complexity control. For performance evaluation, a 6-class dataset of 193 legal opinions related to consumer complaints was created in which each instance was manually labeled according to its moral damage value. A 10-fold stratified cross-validation resampling procedure was used to evaluate different models. The results demonstrated that, under this experimental setup, MuDEC outperforms baseline models by a significant margin, achieving 78.7% of accuracy, compared to 61.1% for a SIF classifier and 65.2% for a C-LSTM classifier.


1 Introduction


Text classification is a popular Natural Language Processing (NLP) task that aims at predicting the categorical values associated with textual instances. Such instances might be composed of phrases, paragraphs, or even entire documents, naturally increasing the task's complexity.


Kowsari et al. (2019) summarizes most text classification systems as a four-step procedure. The first step is feature extraction, where textual units are converted into numerical features. Typical methods for this purpose are Word2vec (Mikolov et al., 2013) for word-level representation and Doc2vec (Le and Mikolov, 2014) for document-level representation. The second step is an optional dimensionality reduction over the results of the first step. The third step follows with a classification method, such as Naïve Bayes, support vector machines (SVM), gradient boosting trees and random forests. Finally, the fourth step is evaluation, where the applicable metrics are analysed.


One of the relevant applications of text classification is the legal domain, which involves a high volume of unstructured textual documents (Fernandes et al., 2022). The high cost of manually reviewing those documents is one of the main reasons for the growing


^a <https://orcid.org/0000-0003-2951-4972>


^b <https://orcid.org/0000-0001-9889-795X>


^c <https://orcid.org/0000-0002-5928-9959>


^d <https://orcid.org/0000-0003-1723-9897>

^e <https://orcid.org/0000-0003-3843-021X>

^f <https://orcid.org/0000-0003-1624-1645>

^g <https://orcid.org/0000-0002-3901-4907>

^h <https://orcid.org/0000-0003-1400-1330>

ⁱ <https://orcid.org/0000-0003-0765-9636>

research in this domain (Wei et al., 2018). If properly classified by relevant labels, such documents can provide valuable structured information with regard to past lawsuits, allowing better assessment by legal professionals, supporting automated recommendation systems, or other data-driven applications.

This paper is positioned in the context of the automated analysis of legal opinions related to consumer complaints. A *legal opinion* is “a written explanation by a judge or group of judges that accompanies an order or ruling in a case, laying out the rationale and legal principles for the ruling”¹. A *consumer complaint* is “an expression of dissatisfaction on a consumer’s behalf to a responsible party”². In such cases, the legal opinion contains specific provisions referring to the plaintiff’s claim, such as moral damage, material damage, and legal fees due by the defeated party. These provisions are always related to a monetary value. The use of the term *legal opinion* is restricted to this particular context in what follows.

As a proof of concept, this paper focuses on the following specific problem:

- *Identify the moral damage value in legal opinions in the context of consumer complaints.*

This problem is challenging since legal opinions are typically long pieces of text, running through several pages, and are written in a variety of styles. A traditional approach, based on Natural Language tools, would resort to a manually constructed set of rules that locate the section of the text that expresses the desired clause and extract the associated value (Minaee et al., 2021). However, such rules are difficult to define and maintain.

This paper then introduces MuDEC (Multi-step Document Embedding-Based Classifier), a new model that addresses the above question as a classification problem, justified by the small number of unique values for moral damage compensation observed in practice. MuDEC combines Doc2vec and SVM for feature extraction and classification, respectively. To optimize the classification performance, we propose a combination of methods, such as oversampling for imbalanced datasets, clustering for identification of textual patterns, and dimension reduction for complexity control. We demonstrate that these methods combined with our main model can decrease the need for larger annotated datasets when compared to other models.

To validate the model, the paper describes experiments that use a dataset containing 193 legal opinions (in Brazilian Portuguese) enacted by lower court judges in the State Court of Rio de Janeiro in the con-

text of consumer complaints involving electric power companies. It must be noted that such documents are public. Each legal opinion in the dataset was manually classified as explained in Section 4.1.

A 10-fold stratified cross-validation resampling method was used to evaluate different models. We demonstrate that, under this experimental setup, our model outperforms by a significant margin the baseline models, which are based on Smooth Inverse Frequency (SIF) and a combination of Convolutional Neural Network and Long Short-Term Memory (C-LSTM).

The rest of this paper is organized as follows. Section 2 covers related work. Section 3 contains a detailed description of the proposed model. Section 4 presents the results achieved. Section 5 summarizes the conclusions and suggests future work.

2 Related Work

Feature extraction from textual instances evolved from Term Frequency (TF) and Term Frequency-Inverse Document Frequency (TF-IDF) (Salton and Buckley, 1988) to word embeddings such as Word2vec (Mikolov et al., 2013) and Global Vectors for Word Representation (GloVe) (Pennington et al., 2014). The main contribution from word embeddings is the representation of a word as an n dimensional vector in an unsupervised approach, allowing the use of pre-trained vectors in different applications.

Despite the success of word embeddings for semantic representation, feature extraction for text classification through the use of this method is still an open research topic. This is due to the fact that documents are composed of a sequence of words, which, after conversion, are mapped into a sequence of n dimensional vectors of various lengths. Since most machine learning methods require fixed-length feature vectors, the use of word embeddings for text classification requires a specific strategy to map the sequence of vectors into a valid format while preserving enough information from the original feature sequence.

To overcome this limitation, several strategies have been proposed by different authors. Zhou et al. (2015) applied a padding method, using the maximum document length in the dataset as a reference and filling the remaining documents with special symbols at the end. By doing so, each document can be converted to a vector of fixed dimension composed of its word embeddings, and the output can be further used as input for a classifier. Arora et al. (2017) proposed a simple average of the word vectors in the sentence, weighted by their inverse frequency and later removal of the projections of the average vectors on their first

¹https://en.wikipedia.org/wiki/Legal_opinion

²https://en.wikipedia.org/wiki/Consumer_complaint

singular vector (“common component removal”). The method is named Smooth Inverse Frequency (SIF), and interestingly, besides its simplicity, it outperforms several complex models, including Recurrent Neural Networks (RNNs) and LSTMs.

In a different approach, Le and Mikolov (2014) proposed an unsupervised algorithm for a fixed-length feature representation from sequence of words. Initially named as Paragraph Vector, the model became known as Doc2vec, inheriting important aspects from Word2vec, such as the semantic representation of words. As an important advantage when compared to SIF, the model takes into account the word order, at least for a small context.

Depending on the method chosen for feature extraction and its hyperparameters, the resulting features vector might have high dimension, leading to problems with time complexity and memory consumption. To mitigate this problem, different methods have been used for dimensional reduction. Kowsari et al. (2019) lists some of the applicable methods for this task, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Non-Negative Matrix Factorization (NMF), and Random Projection.

One of the natural challenges of a regular classification task is the presence of imbalanced datasets. This is the case of our domain since some compensation values for moral damage were significantly less frequent than others, as observed in figure 2. The simple approach of treating these values as outliers and removing them from the training dataset has serious limitations since although rare, they are still significant instances that should be addressed by the model. The option of balancing the dataset by under-sampling the majority classes can also lead to poor classification performance, especially in contexts where the volume of labeled data is scarce. Menardi and Torelli (2014) treated the dataset imbalance problem by oversampling the minority class based on a smoothed bootstrap resampling technique. Chawla et al. (2002) follows a similar oversampling strategy, but with the introduction of synthetic minority class examples. The method, called Synthetic Minority oversampling Technique (SMOTE), has shown significant improvements in imbalanced datasets for a variety of applications from several different domains (Fernández et al., 2018).

There is a small number of recent papers that address the task of document classification in the Brazilian legal domain. For example, de Araujo et al. (2020) introduced a dataset constructed from digitalized legal documents of the Brazilian Supreme Court, with 692 thousand instances. In the paper, several models, based on bag-of-words, CNNs, RNN’s and boosting algorithms, were implemented as baselines. Although

similar, our work differs from this paper by the task definition since their focus was on document classification and theme assignment, while our task aims at classifying lower-level classes, such as the moral damage values.

In a different approach, Luz de Araujo et al. (2018) addressed the problem of information extraction of documents from Brazilian legal text. More specifically, the authors focused on named entity recognition (NER) for the extraction of entities related to the legal context. Similarly, Fernandes et al. (2022) proposes a model to extract value from Brazilian Court decisions by using models based on Bidirectional LSTMs and Conditional Random Fields (CRFs). These methods could be seen as another valid approach to our proposed task, where the moral damage value would be interpreted as an entity class. The main reason for not following this approach was the need for textual annotations of word positions in the text and their related entity classes, which we believe adds complexity to the annotation process when compared to text classification, where the annotation is limited to simply assigning one class for an entire document.

Lastly, a notable trend in recent research is the use of Deep Learning methods for the text classification task (Minaee et al., 2021). As an example, Zhou et al. (2015) proposed a combination of pre-trained word embeddings, a CNN, and a RNN connected to a softmax layer for final classification. The model, called C-LSTM, extracts higher-level phrase representations from the sequence of word embeddings by implementing a one-dimensional convolution. The resulting output is fed to an LSTM to obtain the sentence representation. The performance evaluation described in this work showed that this model provides excellent results for the sentiment classification and question classification tasks, which are closely related to our goal.

3 Description of the proposed Method

Our proposed method is divided into six sequential steps. Figure 1 illustrates these steps by representing the training and evaluation procedure.

The first step adjusts the textual input according to regular text pre-processing procedures. This is a specially important step considering that legal opinions are structured differently from other domains. The punctuation, line breakers, and excessive spaces are removed, and all upper case letters are converted to lower case.

To minimize the task’s complexity, the document

is filtered to contain only the operative part of the judgement, where the lower or Appellate Court judge presents the judicial solution to the lawsuit (Fernandes et al., 2022). To identify this section, a list of regular expressions was used. The list is composed of expressions such as “given these considerations” and “in the face of the above” as a reference for splitting the document and removing the first section. If the filter fails to identify these expressions, the end section is obtained by considering the document’s last m characters. For our experimental setup, $m = 2,500$ was empirically found to be the most suitable value to comprise most of the end sections. Lastly, stopwords are removed, and the words are tokenized.

The first step is therefore highly dependent on the type of legal document in question and must be adjusted accordingly for other contexts.

After the division between train and test sets, the second step extracts the features by using Doc2vec. The method is trained exclusively by the training text instances resulting from the previous filtering step, with no use of external knowledge from pre-trained embeddings. After training, the model is applied to both the train and test sets, converting each instance composed of a list of textual tokens with multiple sizes to a numerical feature vector of pre-defined dimension n . In this step, the moral damage values expressed in different formats within the filtered text are expected to be mapped into the n -dimensional vectors.

Following feature extraction, the third step applies an oversampling method to the train set for treating its imbalance. For this task, we chose SMOTE as our oversampling algorithm. By doing so, synthetic instances are created using a k nearest neighbors approach, where k instances features are randomly chosen from the minority classes, and a synthetic instance is created along the line segments joining them (Chawla et al., 2002). This procedure is repeated until the dataset is evenly distributed.

The fourth step aims at identifying clusters of similar instances by applying the k-means clustering algorithm to the feature vectors. The idea behind this step is based on the existence of sets of instances with a similar format (e.g., legal opinions using a similar writing style). Intuitively, this task provides a variable that indicates a relation between instances of the same cluster, which may improve the classifier’s performance. The cluster to which each instance belongs is turned into a new one-hot-encoded categorical feature and added to the related feature vector.

The fifth step applies a dimensional reduction by a pre-defined continuous factor within the range of 0 and 1. The output dimension is equal to this factor multiplied by the input dimension. For this step, we

chose PCA as the reduction method. PCA aims at extracting the important information from the features by creating a set of orthogonal variables called principal components. Essentially, the principal components are obtained as linear combinations of the original features, where the first component is required to have the largest possible variance and the second is orthogonal to the first. This process is repeated for the remaining components until the number of components is equal to the output dimension (Abdi and Williams, 2010). Ideally, the dimension reduction factor should be chosen to minimize the model complexity while preserving enough information from the original features.

The last step implements a machine learning classifier that assigns each instance to one of the possible classes. The classifier is fed with the real instance classes as target variables and the results of the last step as input, i.e., the PCA components from the concatenation of (i) the feature vectors of filtered end sentences and (ii) the one-hot-encoded representation of the cluster to which each end sentence belongs. This step adopts an SVM classifier, which is one of the most popular machine learning algorithms, based on the concept of a hyperplane (also called kernel) construction that maximizes the margins that separate the classes in the feature space (Vapnik, 2006). Although the kernel can be defined by different functions, such as polynomials and radius basis functions, a linear kernel is used since in practice, it consistently outperforms the other kernel types for this task.

4 Results

4.1 Experimental setup

The experiments used a dataset containing 193 manually annotated legal opinions (in Brazilian Portuguese) enacted by lower court judges in the State Court of Rio de Janeiro in the context of consumer complaints involving electric power companies. Each legal opinion in the dataset was manually analysed to locate the moral damage value and labeled with the value v found. The labels were then mapped to one of six possible classes – 0; 1,000; 2,000; 3,000; 4,000; and 5,000 – where Class 0 indicates no moral damage compensation ($v = 0$), Class 1,000 that $v = 1,000$, and so on. Values outside these classes are rare and not present in the dataset. The distribution of v in the dataset is expressed by Figure 2.

The experiments for hyperparameter tuning were performed in a 10-fold stratified cross-validation setup, implemented using the 193 annotated instances, resulting in the definition of 10 different combinations of

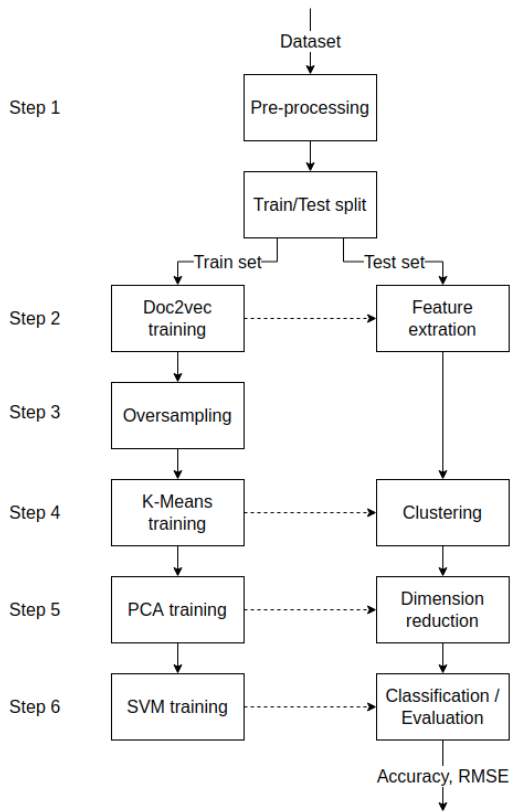


Figure 1: Sequential steps of the model’s training and evaluation process.

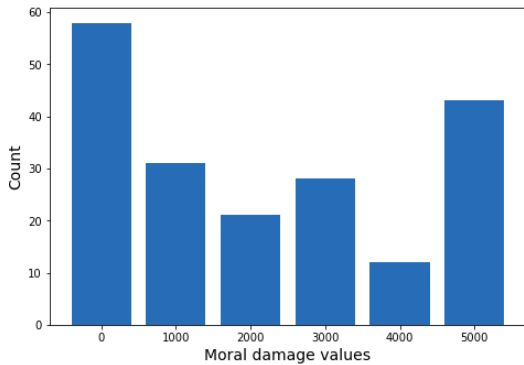


Figure 2: Distribution of classes in the dataset.

train and test datasets, where the test set contained 19 instances on average. In every combination, the test sampling was adjusted to result in a similar distribution of classes when compared to the train set. In section 4.3 we discuss the effect of different dataset sizes in this setup.

A grid search method was used to explore 2,200 combinations of the following parameters:

- Doc2vec vector size: 100, 200, 300, 400 and 500.
- Number of Doc2vec training epochs: 100, 200,

300, and 500.

- Use of oversampling: true or false.
- Number of k-means clusters: range of 0 to 100 with a step of 10, where 0 means k-means is not used.
- PCA factor for dimension reduction: range of 0.2 to 1.0, with a step of 0.2, where 1.0 means PCA is not used.

The linear SVM kernel was empirically found to be the best option for this setup and was fixed during the grid search. The Doc2vec version used was the distributed bag of words (PV-DBOW), and the Euclidean distance was used for the K-Means clustering.

For each grid search iteration, two main evaluation metrics were stored: the overall accuracy and the root mean squared error (RMSE). Even though the problem is generally treated as a classification task, the RMSE was used to analyse how far off are the wrong predictions compared to the real moral damage values. This is an important aspect of the domain, where the quality of a classifier is impacted not only by the proportion of the corrected predictions but also by how distant are the wrong predictions from the real values. In other words, a 5,000 Reais moral damage value, predicted as 4,000 Reais, is less harmful than if it is predicted as zero, for instance.

For a baseline comparison, two other models were implemented over the same dataset and cross-validation setup. The first model follows the same steps described in Figure 1, replacing Doc2vec for the smooth inverse frequency (SIF) (Arora et al., 2017) as the feature extraction method. For the extraction of word embeddings to be summarized by SIF, we used the skip-gram version of Word2vec. For this model, the following combination of parameters was used during the grid search process:

- Word2vec vector size: 100, 300, and 600.
- SVM kernel used: linear, polynomial, or radial basis function.
- Use of oversampling: true or false.
- Number of K-Means clusters: range of 0 to 100 with a step of 10, where 0 means k-means is not used.
- PCA factor for dimension reduction: range of 0.2 to 1.0, with a step of 0.2, where 1.0 means PCA is not used.

The second model is based on C-LSTM as a deep learning version of text classification. The model was structured based on the original paper, as the example in Figure 3 shows, where: the Word2vec dimension is set to 100; 300 filters of size 3 are used during the

convolution, and the instances are padded to the length of 253.

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 253, 300)	90300
lstm (LSTM)	(None, 100)	160400
dense (Dense)	(None, 6)	606

Figure 3: Summary of implemented C-LSTM model

Once again, skip-gram Word2vec was used as word embeddings and, for parameter tuning, the following combination of parameters was tested during grid search:

- Word2vec vector size: 100, 300, and 600.
- Number of filters for convolution: 100, 200, and 300.
- Size of filter used for convolution: 3, 4, and 5.
- Dimension of the LSTM model: 100, 200, and 300.
- Use of oversampling: true or false.

Following the original C-LSTM implementation, an L2 regularization factor of 0.001 is applied to the softmax layer and a dropout probability of 0.5 to the LSTM layer.

All models were implemented in Python, using standard libraries such as Numpy, Scikit-learn, Tensorflow, Keras, and Gensim. Pre-trained skip-gram Word2vec embeddings in Portuguese, provided by Hartmann et al. (2017), were used. The experiments were performed under an Intel i7 CPU with 32 GB of RAM memory and an 8 GB NVIDIA graphic card (used for C-LSTM training).

4.2 Main results

Table 1 shows the main performance metrics of the different models when the optimal parameters found are applied. The results for the optimal parameters obtained through the grid search show that MuDEC outperforms the baseline models SIF and C-LSTM by 17.6% and 13.5% in accuracy and 589.7 and 307.7 in RMSE, respectively.

Model	Mean accuracy	Accuracy standard deviation	Mean RMSE
MuDEC	78.7%	3.7%	1304.3
SIF	61.1%	10.3%	1894.0
C-LSTM	65.2%	11.1%	1611.0

Table 1: Model’s performance under 10-fold stratified cross-validation.

The best combination of hyperparameters of our model, which resulted in the lowest mean accuracy and mean RMSE, were:

- 400-dimensional Doc2vec embeddings
- 200 Doc2vec training epochs
- Oversampling used
- 50 K-Means clusters
- PCA not used

SIF had the best combination of parameters with 600-dimensional Word2vec embeddings, a radial basis function as SVM kernel, oversampling used, and no PCA reduction. C-LSTM achieved the best performance when using 600-dimensional Word2vec embeddings, 300 convolutional filters of size 5, 100-dimensional LSTM layer, 300 training epochs, and oversampling enabled.

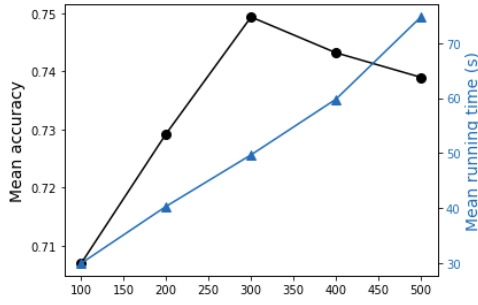
A more detailed analysis of the sensitivity of the parameters in Figure 4, shows their impact on the proposed model. The graphs are plotted by computing the mean accuracy and mean running time for each parameter value throughout the entire grid search examples, providing a general overview of its results.

Figure 4a shows a peak in the mean accuracy for 300-dimensional Doc2vec feature vectors, implying that lower dimension vectors are not able to properly map the information granularity needed for the task, while higher dimension vectors may add unneeded complexity to the model. The running time increases with the vector dimension, as an expected result of complexity added by higher dimensions.

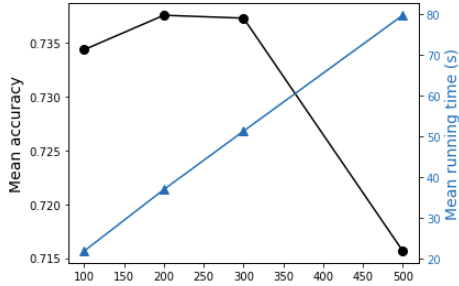
The different Doc2vec training epochs shown in Figure 4b indicate that the model is subject to overfitting when Doc2vec is trained for more than 200 iterations, while the time complexity increases almost linearly with the number of epochs.

The number of clusters illustrated in Figure 4c interestingly shows a constant increase of accuracy with the number of clusters, with a sudden drop when reaching 100. This may indicate a high variety of different patterns seen in the documents. However, further conclusions regarding this method are complex due to the low number of training instances (174 on average). The increase in the number of clusters naturally adds complexity to the K-Means algorithm and input features to the classifier since the cluster features are one-hot-encoded. This effect is expressed by the running time curve.

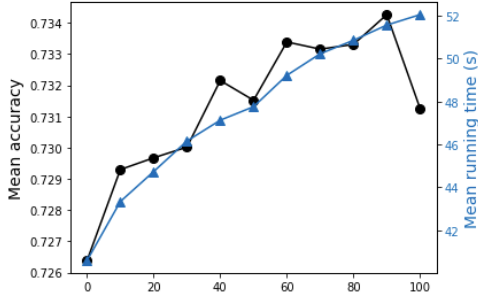
The PCA dimension reduction in Figure 4d indicates that its use slightly degrades the model’s performance in terms of accuracy. At the same time, PCA significantly reduces the model’s time complexity. The reduction of feature dimensions to 80% of its original



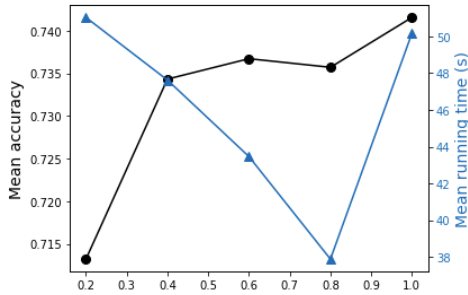
(a) Vector dimensions.



(b) Doc2vec training epochs.



(c) Number of clusters.



(d) PCA output dimension factor (1.0 means PCA is not applied).

Figure 4: Grid search results for different combinations of parameters.

size degrades the accuracy by around 0.5% on average, while the running time is reduced from 44 to 38 seconds.

Lastly, Table 2 shows a small increase in accuracy

by the use of SMOTE, on average by 0.4%.

Metric	Oversampling used	Oversampling not used
Mean accuracy	73.3%	72.9%
Mean running time	49.7 s	44.5 s

Table 2: Grid search results for oversampling.

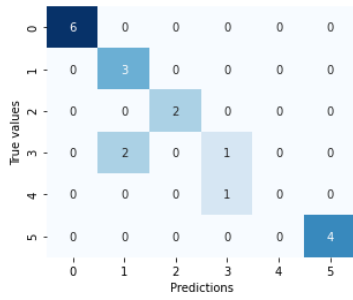
Figure 5 illustrates another comparison between the models by presenting the confusion matrices resulting from the predictions of each model when the optimal parameters are applied. In this comparison, the models are trained and evaluated under the same combination of train and test datasets by using one of the 10-fold cross validation distributions. Note that, since Figure 5 refers to a particular dataset combination, the accuracy values differ from the cross-validation mean accuracy shown in Table 1.

4.3 Model analysis

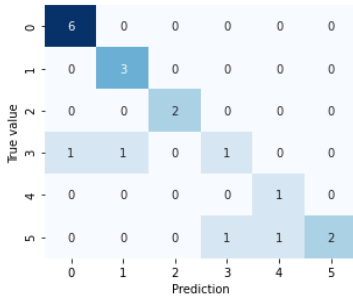
The superior performance of the proposed model, when compared to the baselines, is especially relevant when considering the scarcity of training instances. The majority of the related research on text classification relies on substantially larger datasets. C-LSTM, for instance, was originally introduced by training datasets of at least 5,000 instances. In the context of expensive data annotation, the ability to achieve good results with fewer instances can be a significant advantage.

It is also relevant to note that Doc2vec is, in fact able to map a detailed information, such as the moral damage value, from relatively large pieces of texts, of 1,700 characters on average. This shows the potential of expanding this classification task to additional provisions, other than the moral damage value, possibly allowing the model to become a useful tool for information extraction from legal documents, even pertaining to other domains.

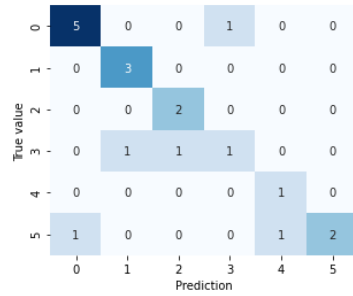
In spite of outperforming the baselines, we acknowledge that the model’s end performance should be improved for its effective utilization due to the sensitivity of the legal domain. The example illustrated by Figure 5a shows the possibility of errors such as predicting a value of 1,000 Reais for an instance where the true value is 3,000 Reais. This can lead to unfair assessments caused by misled information. As a possible solution, the increase of the training dataset can lead to more acceptable results. Figure 6 shows the relation between the cross-validation accuracy and the dataset size, suggesting a growing trend in accuracy when adding new instances. This relation suggests



(a) MuDEC. Accuracy: 84%.



(b) SIF. Accuracy: 79%.



(c) C-LSTM. Accuracy: 74%.

Figure 5: Confusion matrices of each model under the same test data.

that the increase of the dataset leads to more accurate Doc2vec embeddings, better oversampling or cluster definitions, leading to higher overall accuracy. The optimal dataset size is to be evaluated both by the real model’s performance, when applied to larger datasets, and by the domain requirements related to accuracy and other evaluation metrics. In any case, we have strong evidence that relates our model to the most cost-effective method when related to annotation efforts.

5 Conclusions

The results of the paper show that the proposed model, mainly based on Doc2vec and SVM, outperforms the baselines by a substantial margin in the task of classi-

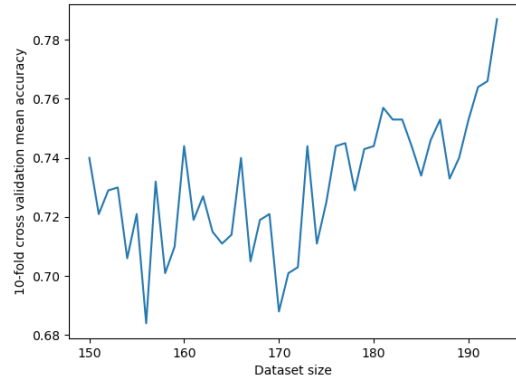


Figure 6: MuDEC’s mean accuracy by dataset size

ifying the moral damage value in the legal opinion of the lower court judges related to consumer complaints involving electric power companies. The results are especially relevant when considering the reduced size of the training dataset, as compared to similar text classification approaches.

The additional methods included in the main model slightly improved the model’s performance. The grid search results show that while oversampling and clustering increase accuracy, PCA reduces the average running time. In the future, more robust hyperparameter optimization methods, such as Bayesian Optimization (Snoek et al., 2012) or Evolutionary Optimization (Kim and Cho, 2019) should be considered, possibly resulting in a more detailed analysis of the model’s sensibility to these methods. The addition of new annotated instances, especially the minority classes, might lead to better cluster definitions and synthetic oversampling, increasing the impact of these methods on the overall results.

Although PCA decreases the model’s running time, the negative impact on the accuracy suggests the need for testing different dimensionality reduction methods. In addition to other similar techniques, this step can be replaced by a feature selection method that, instead of transforming the features into principal components, simply removes the non-correlated features, leaving only a portion of the original variables.

To expand the comparison, further adjustments can be applied to the baseline models, such as fine-tuning word embedding, where the train set can be used to continue the training of word vectors, providing more domain-specific information to the model. In addition, other models can be added to the baseline by using different Deep Learning architectures or different word embedding models.

Finally, we plan to apply MuDEC for classification of other provisions of legal opinions related to

consumer complaints, such as the value of the compensation for material damage or the legal fees due by the defeated party. This would require a more robust annotation process, which considers all relevant features, and expands the number of instances in the current annotated dataset. Going one step further, we also plan to test the model in legal domains other than consumer complaints.

Acknowledgements

This work was partly funded by FAPERJ under grant E-26/200.832/2021, by CAPES under grants 88881.310592-2018/01, 88887.626833/2021-00, and by CNPq under grant 302303/2017-0. The authors wish to thank the Tecgraf Institute, PUC-Rio and the Court of Justice of the State of Rio de Janeiro (TJERJ) for supporting this research, including the following: LABLEXRIO (Núcleo de Inovação do Poder Judiciário), NUPEMASC (Núcleo de Pesquisa em Métodos Alternativos de Solução de Conflitos), CI/TJRJ (Centro de Inteligência do TJERJ) and DGTEC (Diretoria-Geral de Tecnologia da Informação e Comunicação de Dados do TJERJ).

REFERENCES

- Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.
- Arora, S., Liang, Y., and Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. In *International conference on learning representations*.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.
- de Araujo, P. H. L., de Campos, T. E., Braz, F. A., and da Silva, N. C. (2020). Victor: a dataset for brazilian legal documents classification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1449–1458.
- Fernandes, W. P. D., Frajhof, I. Z., Rodrigues, A. M. B., Barbosa, S. D. J., Konder, C. N., Nasser, R. B., de Carvalho, G. R., Lopes, H. C. V., et al. (2022). Extracting value from brazilian court decisions. *Information Systems*, 106:101965.
- Fernández, A., Garcia, S., Herrera, F., and Chawla, N. V. (2018). Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. *Journal of artificial intelligence research*, 61:863–905.
- Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., and Aluisio, S. (2017). Portuguese word embeddings: Evaluating on word analogies and natural language tasks. *arXiv preprint arXiv:1708.06025*.
- Kim, J.-Y. and Cho, S.-B. (2019). Evolutionary optimization of hyperparameters in deep learning models. In *2019 IEEE Congress on Evolutionary Computation (CEC)*, pages 831–837. IEEE.
- Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., and Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4):150.
- Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.
- Luz de Araujo, P. H., de Campos, T. E., de Oliveira, R. R., Stauffer, M., Couto, S., and Bermejo, P. (2018). LeNER-Br: a dataset for named entity recognition in Brazilian legal text. In *International Conference on the Computational Processing of Portuguese (PROPOR)*, Lecture Notes on Computer Science (LNCS), pages 313–323, Canela, RS, Brazil. Springer.
- Menardi, G. and Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data mining and knowledge discovery*, 28(1):92–122.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2021). Deep learning-based text classification: A comprehensive review. *ACM Computing Surveys (CSUR)*, 54(3):1–40.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.
- Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.
- Vapnik, V. (2006). *Estimation of dependences based on empirical data*. Springer Science & Business Media.
- Wei, F., Qin, H., Ye, S., and Zhao, H. (2018). Empirical study of deep learning for text classification in legal document review. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3317–3320. IEEE.
- Zhou, C., Sun, C., Liu, Z., and Lau, F. (2015). A c-lstm neural network for text classification. *arXiv preprint arXiv:1511.08630*.