Comparing main textual feature extraction algorithms in the task of Brazilian legal document clustering

João Pedro Lima[1], José Alfredo Costa[2*†] and Diógenes Carlos Araújo[2†]

[1*]Instituto Metrópole Digital, Universidade Federal do Rio Grande do Norte.

[2]Department of Electrical Enginering, Universidade Federal do Rio Grande do Norte, BR 101 - Campus Central, Natal, 59078-970, RN, Brazil.

*Corresponding author(s). E-mail(s): alfredo.costa@ufrn.br;
Contributing authors: joaopedrodasilvalima@gmail.com;
diogenes.carlos@hotmail.com;
†These authors contributed equally to this work.

Abstract

The digital revolution has accelerated the growth of data volume in all sectors of society, including the legal environment. In Brazil, most of the trials are processed virtually, but the judicial system still suffers from jammed cases, showing that some degree of automatization is needed. Clustering is a Machine Learning process that can help speed up the system, but some problems need to be addressed. This article aims to evaluate the impact of different textual feature extraction methods in the task of clustering Brazilian legal texts. We compared Binary Bag of Words, Bag of Words, Term Frequency-Inverse Document Frequency, Word2vec, and Doc2vec models in different dimensions. The comparison is done using external clustering evaluation metrics and qualitative aspects related to the legal scope. We use a database of 30,000 documents in Brazilian Portuguese of judicial moves of the Tribunal de Justiça do Rio Grande do Norte. The research results suggest that the TF-IDF method seems to be the most suitable for the task, outperforming the other models in considered metrics.

Keywords: feature extraction, clustering, comparison, natural language processing, law

1 Introduction

The digital revolution has accelerated the growth of data volume in all sectors of society, such as agriculture, finance, industry and legal systems. The judicial system in several countries, such as Brazil, is facing challenges in storing and processing this large volume of information. According to the Brazilian National Justice Council's (CNJ) Justice in Numbers report[1], which annually analyzes the state of the Brazilian legal system, the number of cases in the country reached 77 million in 2019.

Due to the strong policy of digitization of justice, most of these cases are processed completely digitally, on platforms such as the PJe, used by most courts across the country. Even in this scenario the percentage of jammed cases is still a problem, as the migration to digital platforms is not enough to give the needed system's flow rate, showing that some degree of automation is crucial. These factors create the ideal scenario for applying Machine Learning techniques, such as clustering, which can increase the speed and efficiency of the system.

There are already a few projects applying machine learning to help Brazilian judicial system. One of the examples is the Victor [2] project that was built in a partnership between the Supremo Tribunal Federal (STF), the brazilian supreme court, and the university of Brasília. The project is responsible for automatizing several steps in the lifetime of a process in STF, with the abilities to convert images into texts, separate the pieces that compose a legal document and classify the main themes within a document.

Clustering can be understood as the process of discovering groups in data without any prior knowledge besides the data itself. There is no general agreement on what exactly is a group or how it should be defined, and each clustering algorithm will have its own premises about the data [3]. For our research, we considered clustering as a partitioning process of a dataset X in a convenient way. We used K-Means as our clustering algorithm, and more details about this decision are presented in the following sections.

Clustering texts in legal domain can be useful in many different tasks, [4] explores this technique to help lawyers find thematically similar amending acts in Polish Civil Code, [5] uses a fuzzy c-means approach to cluster similar argumentative sentences and determine what arguments are used in a case and [6] uses clustering to merge controversial issues in Chinese legal texts.

However, working with legal texts is a complex task, both from the computational and ethical point of view. It is increasingly important to discuss ethical issues in the use of AI systems and their implications, including personal data usage in social networks, autonomous cars, industrial automation, among others. Great challenges for AI systems include lack of transparency and accountability in decision making by machine learning algorithms, e.g., in deep learning models. [7] addressed ethical factors that are essential for future artificial intelligence for social good (AI4SG) initiatives, including: (1) falsifiability and incremental deployment; (2) safeguards against the manipulation of predictors; (3) receiver-contextualised intervention.

Ethical impact and unintended consequences of new technologies must be considered in developing AI systems, especially when dealing with personal data. Research dealing with clustering with multiple sensitive attributes has been published, e.g., [8], that described a FairKM (Fair K-Means) for scenarios involving multiple multi-valued or numeric sensitive attributes.

Creating AI applications is also difficult from the computational perspective. Texts are unstructured, non-numeric and complex data, transforming them into information for machine learning algorithms is not a trivial task. The legal environment still imposes additional difficulties, since all texts have a specific form of writing with a specific vocabulary that differs from common texts, requiring special attention. Many efforts have already been made to optimize this representation and nowadays we have a huge range of algorithms of the most different types.

Many Deep Learning techniques have proven very effective for numerically capturing the 'syntax' and 'semantics' of long, complex texts, such as BERT[9] and ULMi FIT[10] and also for single words, such as Word2Vec[11]. However, much of this efficiency comes at the cost of a complex model to train and almost impossible to audit.

Classical techniques such as TF-IDF vectors, in turn, do not have complex text representation mechanisms, can be trained in a much shorter time and have more human-friendly calculations. However, they fail to capture complex relationships between words and are often surpassed in more complex tasks.

Choosing an algorithm for text representation is a problem-dependent decision, which can have a strong impact on the performance of the models[12]. In the context of clustering, the proper choice becomes even more critical, as the numerical representation of the data is all the information the algorithm has.

In this paper, an experiment were conducted to empirically assess the impact of the main textual feature extraction methods on the quality of textual clustering of legal documents in Brazilian Portuguese. Binary Bag of Words, Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF), Word2Vec and Doc2vec algorithms with different dimensions were tested. K-means[13] were used for clustering.

The proposed experiment is divided into two stages. In the first, the algorithms are evaluated in a set of different numbers of clusters and, in the second, when the number of clusters is equal to the original number of classes in the database used.

The quantitative evaluation was performed by calculating the Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) scores between the true classes and the final result of the clustering. We also made qualitative ethical assessments about each algorithm. The experiments conducted on a database of 30,000 legal documents, divided between 10 classes, made available by the Tribunal de Justiça do Rio Grande do Norte.

2 Related Works

In [12], four feature extraction techniques were evaluated: Best Matching 25 (BM25), Latent Semantic Analysis (LSA), Doc2Vec, and the Latent Dirichlet allocation (LDA) for semantic representation of digital educational resources. The authors assess the results with unsupervised metrics and demonstrate that the performance of these techniques varies according to the content of the texts.

[14] also compares four different feature extraction techniques: GloVe[15] Embeddings, Siamese-BERT, BERT and TF-IDF to measure the similarity between different data protection legislations of different countries, including Brazilian legislation. The results show that, for this case, BERT outperforms all other models.

[16] evaluates word embeddings trained from different datasets in NLP tasks aimed to medical field. The results demonstrate that vector training in databases of the same subject generates more accurate semantic representations, but does not necessarily increase the final performance of NLP tasks.

In legal field, [17] compares four feature extraction techniques in the task of similarity measurement of Indian Supreme Court cases. The results shows that recent embedding methods outperforms more traditional methods, such as TF-IDF and LDA. However, the authors also report that these traditional algorithms performed well, and can be further improved.

Still in the legal field, efforts have been made to improve feature extraction of legal documents. Notoriously, [18] developed Law2vec, a proprietary algorithm for creating embeddings for legal texts.

Efforts have also been made to improve the performance of legal document clustering, as in [19], where a technique for large-scale soft clustering with topic segmentation is demonstrated, creating an efficient algorithm for information retrieval and processing, and in [20], where a variant of K-Means is implemented for a legal document search system.

Recently, [21] presented unsupervised approaches for measuring textual similarity between legal court case reports. They investigated the performance of 56 different methodologies for computing textual similarity across court case statements when applied to the dataset of Indian Supreme Court Cases, including BERT and Law2Vec. The authors reported that traditional methods (such as the TF-IDF) performed better than the more advanced context-aware methods.

One of the most similar works to this one, [22] compares the coupling TF-IDF+LSA against Doc2Vec in the task of clustering small corpora with K-Means. The authors share the perspective that, if a text representation is good, then the clustering will also be good even with very simple models such as the K-Means. The results show that the TF-IDF pipeline performs better than Doc2vec accordingly to the ARI, but the authors report that the Doc2vec was less time and memory-consuming.

All these works shows that has no optimal choice for textual clustering, and the most important thing is to be aware of the application scope and needs. Some research also suggests that this reality can be extended even to some supervised tasks. The work in [23] compares several algorithms in medical documents classification. The performance of the Word2Vec, BOW, and Doc2Vec algorithms is compared, and the authors conclude that has no clear winner between BOW and Word2vec.

The comparison made in [24] also shows interesting results. The work compares TF-IDF, BOW, Paragraph2Vec, Kate Autoencoder and Glove in paraphrase detection. According to the authors, none of the models tested has satisfactory performance, but the frequency-based models, TF-IDF and BOW, surpass other approaches.

3 Theoretical Basis

This section is a briefly discussion of the tested algorithms and metrics. For the following topics, consider $D = \{d_0, d_1, ..., d_n\}$ the set of documents in a corpus, $V = \{w_0, w_1, ..., w_m\}$ the vocabulary of this corpus, and $P = \{p_0, p_1, ..., p_n\}$, $p_i \in \mathbb{R}^k$ the respective vector representations of documents D. A dataset partition is defined as $C = \{c_1, c_2, \ldots, c_t\}$, where c_i is the set of points that compose the i^{th} partition.

3.1 Binary Bag of Words

The Binary Bag of Words is one of the simplest models, it considers each document as a binary vector of terms occurrence, in the One-Hot-Encoding style. The algorithm assigns the value 1 to the weight p_{ij} if word j is in document i, and 0 otherwise:

$$p_{ij} = \begin{cases} 1 & \text{if } w_j \in d_i \\ 0 & \text{otherwise} \end{cases}$$

3.2 Bag of Words

In the Bag of Words algorithm, each document is represented by a vector containing the count of its terms. Thus, the algorithm assigns to each word j in document i: $p_{ij} = \text{count}(d_i, w_j)$, where $\text{count}(d_i, w_j)$ is exactly how many times the word j occurs in document i.

3.3 TF-IDF

TF-IDF is a classic textual vectorization algorithm, which is based on two coefficients: text frequency (TF), and documental frequency (IDF), to build numerical vectors. The algorithm assigns a weight to each word j of the text i according to: $p_{ij} = \text{TF}(w_j, d_i) \times \log(\text{IDF}(w_j))$, where $\text{TF}(w_i, d_j)$ is the frequency of the word j in the text i, and $\text{IDF}(w_i)$ the inverse of the documentary frequency of the word throughout the corpus. The idea is that the weights can translate the importance of each word to the given document, based on its rarity.

3.4 Word2vec

Word2Vec is an algorithm proposed by [11] that tries to assign a constant size dense vector for each word using a artificial neural network (ANN).

The ANN is used in a standard prediction task, that can be Skip-gram, where the network tries to predict the context around a word, or Continuous Bag-of-words, where the network tries to predict the central word given a context. The weights learned by the ANN's layers are the actually values of the embeddings. Word2vec is the most famous word embedding algorithm, and its has not only been widely used in research, but was very important, as it introduced of a new era in NLP models, with context-aware methods designed using deep learning.

In the original paper, the authors show that the model was able to incorporate meaningful relationships between words in the vectors created, allowing 'semantic' relations to be expressed mathematically, such as: $\text{vector}(king) - \text{vector}(queen) = \text{vector}(man) - \text{vector}(woman)$.

3.5 Doc2vec

Doc2vec[25] is a paragraph vectorization algorithm heavily based on Word2vec, it is able to map documents of generic sizes to fixed size dense vectors. Its training is very similar to Word2vec, and it also uses the weights of a neural network to learn the representations. However, the algorithm expands the scope of representation from the domain of words to the domain of documents, becoming capable of creating relationships between independent documents from the similarity of their terms.

3.6 K-Means

K-Means is the most widely used clustering algorithms and also one of the simplest in implementation. Its premise is to find k points $C = \{C_1, C_1, \ldots, C_k\}$ in the data space, each one representing a cluster, that minimizes the data's inertia[1].

These points are randomly initialized and iteratively moved to optimal values, in a two step process called Expectation-Maximization (EM). Leaving the implementation details aside [26][27][28], the important thing to note is that, when fully converged, each data point x_i is assigned to a cluster y_i based on its Euclidean distance to the centroids:

[1] The inertia objective function is the within cluster variance sum

$$y_i = \underset{1<j<k}{\operatorname{argmin}} \left\{ \|x_i - C_j\|^2 \right\}$$

This assign method actually divides the data space in Voronoi cells which, in practical terms, means that this algorithm is only able to find globular-like clusters, what is often pointed as a weakness. As the main goal of this research is to evaluate the performance of different feature extraction algorithms, and not necessarily to achieve the best scores possible, the choose of k-means seems to be adequate, as its a common used algorithm, with a simple implementation.

3.7 Adjusted Rand Index

The Rand Index[29] is a statistical method to evaluate agreement between two partitions C and S of a dataset $X = \{x_1, x_2, \ldots, x_n\}$. Its implementation is based on the notion of pairwise agreement, i.e, if two points x_i and x_j are or not grouped together in both partitions. It is defined as follows:

$$RI(C,S) = \frac{N_{00} + N_{11}}{\binom{N}{2}}$$

Where N_{11} is the total number of pairs grouped together in both partitions, N_{00} is the total number of pairs that aren't grouped in any partition, and the bottom term is the total number of pairs. The Adjusted Rand Index (ARI) is a correction made in RI to account for chance, and can be any value in the [-1,1] interval, where low values indicate low agreement.

3.8 Normalized Mutual Information

The Normalized Mutual Information is a measure of agreement based on Information Theory concepts. Considering two partitions C and S of a dataset $X = \{x_1, x_2, \ldots, x_n\}$, the NMI score is defined as follows:

$$NMI(C,S) = \frac{I(C,S)}{\left(\mathrm{H}(C) + \mathrm{H}(S)\right)/2}$$

Where I is the information shared between the two partitions, and $H(\cdot)$ is the entropy of each individual partition:

$$\mathrm{I}(C,S) = \sum_i \sum_j P(c_i \cap s_j) \log \frac{P(c_i \cap s_j)}{P(c_i)P(s_j)}$$

$$\mathrm{H}(C) = -\sum_k P(c_k) \log P(c_k)$$

$P(g)$ is the probability of a point belong to a group g. NMI values are bounded in the interval $[0,1]$, and values closest to 1 indicate better agreement.

4 Experiments

This research aims to evaluate the impact of the main feature extraction algorithms in the results of Brazilian legal documents clustering. Figure 1 contains a summary of the metodology applied. This section details the models tested, the evaluation methods and the database used.

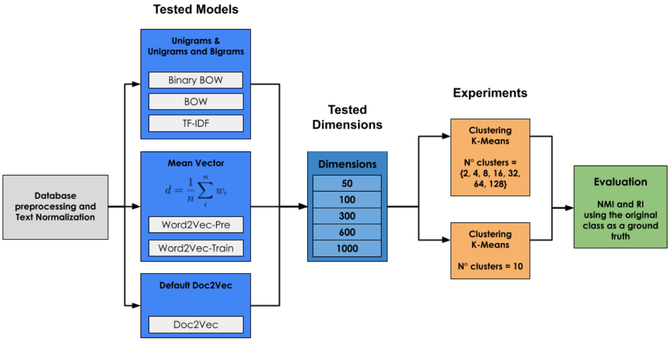

Fig. 1 Metodology

4.1 Models description

The tested models were Binary BoW, BoW, TF-IDF, Word2Vec and Doc2Vec. We tested them in five different vector dimensions, $d \in \{50, 100, 300, 600, 1000\}$. For the traditional models, i.e. Binary BoW, BoW and TF-IDF, versions that also account for bigram have also been included to increase the experiments' variability. Word2vec was tested both with vectors trained from scratch in the database, as well as with pre-trained vectors obtained from the NILC-Embeddings repository[30]. In both cases, the Skip-gram strategy was choose. As Word2vec vectors only incorporate words, the document vector is computed as the average of the vectors of the words that make it up. Besides the dimension, Doc2vec did not have any other parameters tested. In the end, there is a total of 45 models. The model chosen for clustering was K-Means.

The tests were developed in Python with the help mainly of the Scikit-learn[31] and Gensim[32] libraries, and Table 1 shows the hyperparameters used.

4.2 Database and Preprocessing

All experiments were conducted in the database of judicial transactions of the Tribunal de Justiça do Rio Grande do Norte (TJRN). This database contains a total of 30,000 documents of judicial movements distributed equally among 10 classes, each class representing a specific type of movement. Table 2 provides a description of the database's classes.

Table 1 Models' hyperparameters

Model	Hyperparameters changed	Source
BagOfWords	max_features=dimension ngram_range $\in \{(1,1),(1,2)\}$	Scikit-Learn CountVectorizer

Doc2Vec	n_components=dimension
	epochs=10
		Gensim
		doc2vec

BinaryBagOfWords	max_features=dimension
	ngram_range $\in \{(1,1),(1,2)\}$
	binary=True
		Scikit-Learn
		CountVectorizer

TF-IDF	max_features=dimension ngram_range $\in \{(1,1),(1,2)\}$	Scikit-Learn TfidfVectorizer

Word2Vec	n_components=dimension
	sg=1, iter=10
		Gensim
		word2vec

Word2VecPre	Skip-Gram	NILC-Embeddings

Table 2 Dataset classes description

Label	Name(pt)	Name(en)
196	Extinção da execução ou Cumprimento da sentença	Discharge of the execution or Judgement enforcement
198	Acolhimento de embargos de declaração	Acceptance of clarification motions
200	Não acolhimento de embargos de declaração	Not acceptance of clarification motions
219	Procedência	Validity
220	Improcedência	Denial
339	Liminar	Preliminary
458	Abandono de causa	Abandonment of the claim
461	Ausência das condições da ação	Absence of the action's conditions
463	Desistência	Discontinuance
785	Antecipação de tutela	Interlocutory remedy

As shown in figure 1, before being fed to any language model, the database is preprocessed as follows: (1) The text has been converted to lowercase; (2) Were removed special symbols, punctuation and numbers; (3) The Portuguese stopwords have also been removed.

4.3 Evaluation

The experiment is divided in two sections: The performance of each of the 45 models is computed as a function of the number of clusters c, with $c \in \{2, 4, 8, 16, 32, 64, 128, 256\}$. The idea behind this analysis is to visualize the general algorithms' behavior in different situations since, in real applications, the 'real' number of clusters in a dataset is not known. In the second, the algorithms' performance is computed when the number of clusters is known, that is, when number of clusters c is equal to the number of original classes in the dataset, which is 10. The metrics used were Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) between the predicted clusters and the true classes. In practical terms, what is really being assessed is the capacity of the feature extraction methods to create vectors that allows clustering to 'reproduce human behavior'.

Qualitative assessments regarding the needs of legal AI have also being made. They can be divided into ethical and computational. The first includes manly characteristics such as model's transparency and interpretability, needed to ensure fair and unbiased trials. The second refers to computational cost, since not all courts, especially in Brazil, have the infrastructure needed to train complex models. This cost is approximated by the time spend in model training.

5 Results and discussion

5.1 Experiments with many clusters

Figure 2 shows the models' best performance in each metric versus the number of clusters. TF-IDF outperforms all other models in pratically all cases, especially when the number of clusters is low. In particular, when the number of clusters is 8, the difference between the first and second places in both scores is greater than 0.1.

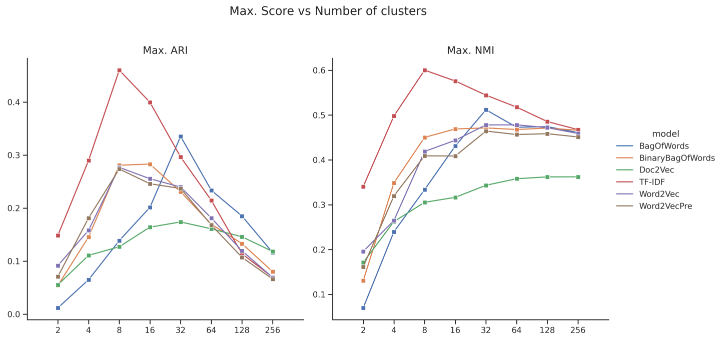

Fig. 2 Best scores of each model in each number of clusters

This fact is not surprising, the TF-IDF vectors have an intelligent way of determining the importance of terms based on their document frequency, in a strategy that favors clustering, as it reduces the importance of very frequent tokens, emphasizing the less frequent ones, which are exactly the ones that have the potential to split the corpus.

Despite its simplicity, Binary BoW was able to achieve good scores, repeatedly coming in second place. This is a interesting fact because this binary approach was manly introduced in this research as a 'baseline model'. However, looking at the results, especially the best ARI values, its performance behavior similarly as the two Word2Vec models.

The BoW model, although it is very similar to its binary version, performs significantly differently. Considering the left graph on fig 2, its best ARI values are initially low, but slowly grow up and suddenly became the only model in all the experiments to consistently surpass the TF-IDF.

Word2vec vectors, in their two versions, perform very similarly, and it is difficult to determine which is better. These vectors perform very similarly to the binary approach, demonstrating that the complexity is not necessarily correlated with efficiency. Reiterating this last point, Doc2vec was the worst model, getting very low scores compared to the others.

Another important point drawn from fig. 2 is that the TF-IDF is the only model with well defined peak at c=8 in both scores, which is consistent with the previous knowledge about the number of original classes in the database (c=10).

Table 3 summarizes the scores statistics from all runs for each model. TF-IDF has the highest mean and the highest maximum score in both scores, being the best model. Next, Word2vec trained from scratch in our base has the second best average in both scores, slightly beating its pre-trained version. Binary BoW and Bow also have similar scores, but the second is slightly better. Doc2vec was the worst model in all aspects.

Table 3 Scores summary

	NMI			*ARI*		
	min	mean	max	min	mean	max
BagOfWords	0.02	0.32	0.51	0.00	0.11	0.33
Doc2Vec	0.04	0.27	0.36	0.01	0.11	0.17
BinaryBagOfWords	0.01	0.30	0.47	0.00	0.11	0.28
TF-IDF	**0.23**	**0.45**	**0.60**	0.05	**0.20**	**0.46**
Word2Vec	0.18	0.38	0.47	**0.06**	0.16	0.27
Word2VecPre	0.14	0.34	0.46	0.05	0.14	0.27

Figure 3 contains plots of the scores' distributions. In these TF-IDF model visually looks better, with distributions centered on higher scores. Although the BoW and Binary BoW methods have higher maximum scores than the two versions of Word2Vec, they are much more inconsistent and they distributions are centered bellow.

Table 5 contains the results of the experiment were the number of clusters in K-Means was equal to the number of original classes in the database $c = 10$.

Once again, TF-IDF surpass all other models in all metrics. All models had scores very similar to those in table 3, and none benefited greatly from the equality between number of clusters and number of classes. In fact, the table shows that the performance, in general, has dropped.

Figure 4 shows the best score for each model in each dimension. As a general rule, their score tends to rise little with dimension, and the performance seems to be unchanged for any dimension greater than $d = 300$.

Table 4 Execution time

	Execution Time	
	mean	std
BagOfWords	18 sec	10 sec
Doc2Vec	17 min 20 sec	06 min 10 sec
OneHotEncoding	19 sec	10 sec
TF-IDF	18 sec	09 sec
Word2Vec	26 min 33 sec	10 min 23 sec
Word2VecPre	44 sec	08 sec

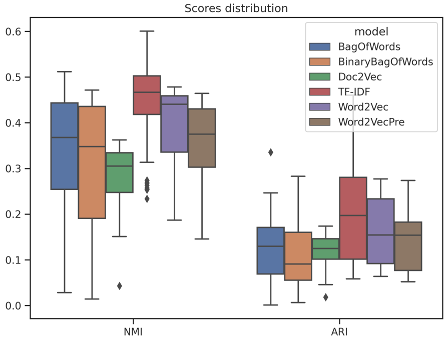

Fig. 3 Boxplot of each model's scores' distribution

Table 5 Scores' statistics when c=10

	NMI		*ARI*	
	mean	max	mean	max
BagOfWords	0.31	0.34	0.13	0.15
Doc2Vec	0.25	0.27	0.12	0.13
OneHotEncoding	0.32	0.43	0.18	0.26
TF-IDF	**0.51**	**0.60**	**0.36**	**0.45**
Word2Vec	0.42	0.43	0.27	0.28
Word2VecPre	0.35	0.40	0.23	0.26

When dealing with Machine Learning on legal environment is important considering factors such as interpretability, that represents our ability to understand and explain models' decisions. The ideal is to try to avoid 'Black Box algorithms' and opt for algorithms that can be audited. In this aspect, it is preferable to use traditional methods (BoW, TF-IDF), which are based on simple rules and produce vectors with easily interpretable weights, in detriment of methods based on neural networks, which produce dense vectors that are difficult to explain, such as Word2Vec.

Another aspect taken in consideration is computational cost. This cost is approximated as the time spend on model training, Table 4 shows the results.

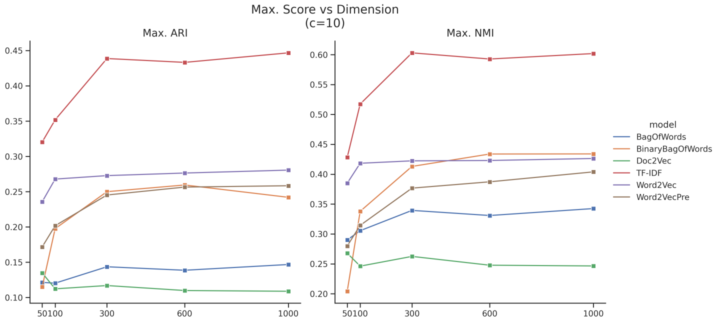

Fig. 4 Best scores of each model in each dimension when c=10

Table 6 Final scores

Model	Performance	Interpretability	Cost	**Total**
BagOfWords	3	5	5	13
Doc2Vec	2	1	2	5
BinaryBagOfWords	3	5	5	13
TF-IDF	5	4	5	**14**
Word2Vec	4	1	1	6
Word2VecPre	4	1	5	10

In this experiment, the traditional methods' execution time is somewhat constant, as none of the hyperparameters considered can change it drastically. For Word2Vec and Doc2Vec, the execution time is proportional to the number of epochs used in training.

Table 6 summarizes all the complex aspects considered trough the paper. Each model receives a score from 1 to 5 regarding its performance, its agreement with the ethical needs and its computational cost.

6 Conclusion and Future Work

Working with clustering in NLP for legal texts is a hard task. It mixes the uncertainties of clustering, with the problems of textual vectorization and still imposes limitations due the legal application scope.

This paper empirically assess the impact of textual feature extraction algorithms on the task of clustering Brazilian legal documents. The work considers both quantitative aspects, calculating the NMI and ARI scores, as well as qualitative aspects, such as interpretability and computational cost.

The experiments demonstrate that the TF-IDF method is superior in all aspects to other models, in agreement with the results of [33] and [21]. It is capable of obtaining the highest scores in all of the metrics, is one of the models that best suits the legal environment in terms of ethical needs and still has one of the lowest training costs.

Although the Word2vec method has better results than BoW and Binary BoW, its interpretability is lower, and the decision between these two models is up to the application. The scores' performance difference between the pretrained and trained versions of Word2Vec is small, and the cost of training new vectors may not justify the gain. Doc2vec had the worst performance. In general, was demonstrated that simple and traditional methods, based on word counting, seem to be better suited to this task.

As future extensions, a greater number of vectorization and clustering methods can be considered, in addition to evaluating the impact of textual normalization techniques, such as lemmatization and stemming. It would also be interesting to make the experiment in other Brazilian legal bases.

Acknowledgments. This research is in part supported by CAPES - Brazilian Agency of Superior Education and CNPq - Brazilian Agency for Resarch and Development

References

[1] CNJ - Conselho Nacional de Justiça: Relatório Justiça em Números. https://www.cnj.jus.br/pesquisas-judiciarias/justica-em-numeros/ (2020)

[2] Maia Filho, M.S., Junquilho, T.A.: Projeto victor: perspectivas de aplicação da inteligência artificial ao direito. Revista de Direitos e Garantias Fundamentais **19**(3), 218–237 (2018)

[3] Hennig, C.: What are the true clusters? Pattern Recognition Letters **64**, 53–62 (2015)

[4] Górski, L.: Towards legal change analysis: Clustering of polish civil code amendments. In: ASAIL@ ICAIL (2019)

[5] Poudyal, P., Gonçalves, T., Quaresma, P.: Using clustering techniques to identify arguments in legal documents. In: ASAIL@ ICAIL (2019)

[6] Tian, X., Fang, Y., Weng, Y., Luo, Y., Cheng, H., Wang, Z.: K-means clustering for controversial issues merging in chinese legal texts. In: JURIX, pp. 215–219 (2018)

[7] Floridi, L., Cowls, J., King, T.C., Taddeo, M.: How to design ai for social good: Seven essential factors. In: Ethics, Governance, and Policies in Artificial Intelligence, pp. 125–151. Springer, ??? (2021)

[8] Abraham, S.S., Sundaram, S.S., et al.: Fairness in clustering with multiple sensitive attributes. arXiv preprint arXiv:1910.05113 (2019)

[9] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv :1810.04805 (2018)

[10] Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv: 1801.06146 (2018)

[11] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. arXiv preprint arXiv:1310.4546 (2013)

[12] Aguilar, J., Salazar, C., Velasco, H., Monsalve-Pulido, J., Montoya, E.: Comparison and evaluation of different methods for the feature extraction from educational contents. Computation **8**(2), 30 (2020)

[13] Rokach, L., Maimon, O.: Clustering methods. In: Data Mining and Knowledge Discovery Handbook, pp. 321–352. Springer, ??? (2005)

[14] Kawintiranon, K., Liu, Y.: Towards automatic comparison of data privacy documents: A preliminary experiment on gdpr- like laws. arXiv preprint arXiv:2105.10117 (2021)

[15] Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)

[16] Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., Liu, H.: A comparison of word embeddings for biomedical natural language processing. Journal of biomedical informatics **87**, 12–20 (2018)

[17] Mandal, A., Chaki, R., Saha, S., Ghosh, K., Pal, A., Ghosh, S.: Measuring similarity among legal court case documents. In: Proc. of the 10th Annual ACM India Compute Conference, pp. 1–9 (2017)

[18] Chalkidis, I.: Law2vec: Legal word embeddings (2018)

[19] Lu, Q.g., Conrad, J.G., Al-Kofahi, K., Keenan, W.: Legal document clustering with built-in topic segmentation. In: Proc. of the 20th ACM Inter. Conf. on Information and Knowledge Management, pp. 383–392 (2011)

[20] Wagh, R.S.: Exploratory analysis of legal documents using unsupervised text mining techniques. International Journal of Engineering Research & Technology (IJERT) **3**(2) (2014)

[21] Mandal, A., Ghosh, K., Ghosh, S., Mandal, S.: Unsupervised approaches for measuring textual similarity between legal courtcase reports. Artificial Intelligence and Law, 1–35 (2021)

[22] Amalia, A., Sitompul, O.S., Nababan, E.B., Mantoro, T.: A comparison study of document clustering using doc2vec versus tfidf combined with lsa for small corpora. Journal of Theoretical and Applied Information Technology **98**(17), 3644–3657 (2020)

[23] Shao, Y., Taylor, S., Marshall, N., Morioka, C., Zeng-Treitler, Q.: Clinical text classification with word embedding features vs. bag-of-words features. In: 2018 IEEE Inter. Conf. on Big Data (Big Data), pp. 2874–2878 (2018). https://doi.org/10.1109/BigData.2018.8622345

[24] Shahmohammadi, H., Dezfoulian, M., Mansoorizadeh, M.: An extensive comparison of feature extraction methods for paraphrase detection. In: 2018 8th Inter. Conf. on Computer and Knowledge Engineering (ICCKE), pp. 47–51 (2018). https://doi.org/10.1109/ICCKE.2018.8566303

[25] Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Inter. Conf. on Machine Learning, pp. 1188–1196 (2014). PMLR

[26] Elkan, C.: Using the triangle inequality to accelerate k-means. In: Proc. of the 20th Inter. Conf. on Machine Learning (ICML-03), pp. 147–153 (2003)

[27] Lloyd, S.: Least squares quantization in pcm. IEEE transactions on information theory **28**(2), 129–137 (1982)

[28] Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. Technical report, Stanford (2006)

[29] Steinley, D.: Properties of the hubert-arable adjusted rand index. Psychological methods **9**(3), 386 (2004)

[30] Hartmann, N., Fonseca, E., Shulby, C., Treviso, M., Rodrigues, J., Aluisio, S.: Portuguese word embeddings: Evaluating word analogies and natural language tasks. arXiv preprint arXiv:1708,06025 (2017)

[31] 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.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[32] Rehurek, R., Sojka, P.: Gensim–python framework for vector space modelling. NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic **3**(2) (2011)

[33] Yeung, C.M.: Effects of inserting domain vocabulary and fine-tuning bert for german legal language. Master's thesis, University of Twente (2019)