

A Classifier-Based Approach to Predict the Approval of Legislative Propositions

Ilo César Duarte Cabral and Glauco Vitor Pedrosa^a

Graduate Program in Applied Computing (PPCA), University of Brasilia (UnB), Brasilia, Brazil

Keywords: Decision Support System, Natural Language Processing, Data Mining, Supervised Classification, Imbalanced Dataset.

Abstract: This paper presents a data mining-based approach to predict the approval of Legislative Propositions (LPs) based on textual documents. We developed a framework using machine learning and natural language processing algorithms for automatic text classification to predict whether or not a proposition would be approved in the legislative houses based on previous legislative proposals. The major contribution of this work is a novel kNN-based classifier less sensitive to imbalanced data and a time-wise factor to weight similar documents that are distant in time. This temporal factor aims to penalize the approval of LPs with subjects that are far from current political, social and cultural trends. The results obtained show that the proposed classifier increased the F1-score by 30% when compared to other traditional classifiers, demonstrating the potential of the proposed framework to assist political agents in the legislative process.

1 INTRODUCTION


Legislative Propositions (LPs) consists of documents addressing a subject that will be deliberate in the legislative houses. It is from the formulation of LPs that political agents establish the laws, which will govern society, regularizing social, mercantile, labor practices, among others. In most of the modern government systems, such as the Westminster system and the federal system of the United States, the core function of the legislative process is to vote on LPs (Cheng et al., 2017). In this sense, predicting which LPs are more likely to be approved can assist public and private entities to direct their strategic planning. Examples of these entities are: Companies, Service Providers, Financial Market, Civil Society Organizations, Federal Government, State Governments and Municipal Governments.

The big challenge for machine learning algorithms to predict the approval of LPs is dealing with imbalanced data. A dataset is imbalanced when there is a clear disproportion between the number of examples of one or more classes in relation to the other classes. For example, from the beginning of 2001 to the end of 2021, only 8.2% of proposals were approved in the Brazilian legislative houses. Most

classifiers in the area of machine learning face serious problems in a context where there is an imbalance, which can lead to, among other consequences, classification biases, affecting the efficiency and reliability of the models. Therefore, learning from imbalanced datasets is one of the top 10 challenging problems in data mining research (Yang and Wu, 2006).

In this work, we developed a data-mining based approach comprising machine learning and natural language processing algorithms in order to predict the chance of approval of LPs using its textual content. The developed framework consists of: (i) representing LPs documents in sparse vectors using Inverse Document Frequency (TF-IDF); (ii) applying a dimensionality reduction technique using Singular Value Decomposition (SVD); (iii) introducing a cost sensitive factor of misclassification in the classification algorithm to mitigate the imbalance of the database; (iv) introducing a time factor in the classification to penalize proposals with subjects outside current trends.

The main contribution of our framework is a classifier-based approach to dealing with imbalanced data and a time-wise analysis. The proposed classifier is a kNN-based approach comprising two weighted-based factors: time and dataset imbalance. The time factor aims to generate predictions based on the

^a  <https://orcid.org/0000-0001-5573-6830>

composition of the current chamber in order to reflect the trends (political, economic, financial and cultural) of the current time. For example, it is expected that LPs addressing similar issues with other proposals already approved in a short period of time will also be approved. In this sense, when comparing the similarity of propositions, the time frame between the LPs is an important issue that must be considered by the classifier. Besides that, kNN faces difficulty in imbalanced datasets as it treats all neighbors of the query instance equally and most of the neighbors will be of the majority class. To deal with this issue, we use a distance-based approach to provide more importance to neighbors of the minority class with a higher proximity weighted confidence.

The proposed framework was validated using the public database of LPs available by the Brazilian Chamber of Deputies. In our experiments, we used LPs related to laws and amendments and with the status completed between 2001 and 2021. Experimental tests were performed to compare the proposed classifier with different modifications in the kNN algorithm and with other machine learning classifiers.

This paper is structured as follows: Section 2 presents the context of the work and related works; Section 3 formalizes the classification problem; Section 4 presents the proposed approach; Section 5 shows the results of the experimental tests carried out to evaluate the technique proposed and, finally, Section 6 presents the conclusions of this work.

2 CONTEXT AND RELATED WORKS

The legislative process comprises the elaboration, analysis and voting of various types of proposals, such as: Ordinary Laws, Provisional Measures, Amendments to the Constitution, Legislative Decrees, Resolutions, among others. It is from the formulation of these propositions that political agents establish the laws that will govern society and regulate social, mercantile, labor practices, among others.

The work of (Nay, 2017) presented a model to predict whether or not a proposition would be approved in the US Congress. They used a database with the proposed laws from 1993 to 2015. From this database, they extracted 12 characteristics and performed data analysis. A model was trained using word2vec (Le and Mikolov, 2014) and tree-based models as well as ensemble stacking techniques. The technique developed had a 96% success rate in

predicting the approval of a legislative project.

The work of (Cheng et al., 2017) also focused on the US Congress. They presented a technique to analyze legislators' profile data and also the textual data of the propositions. To carry out the text analysis of the projects, the authors used Bag-of-Words model, and also the way in which the ideological profile data of legislators were used in a Euclidean spatial model called policy location.

Historically, the rate of approval of laws in Brazil is less than 0.9% of the total of propositions presented in the legislative houses. In the US Congress, this rate is approximately 4%. In this scenario of so many disapprovals of propositions, it is necessary to know in advance which projects are worth paying attention to, that is, what is the probability of each project being approved. Therefore, proposing mechanisms based on natural language processing and machine learning is one of the ways to develop computational mechanisms capable of helping the entities that follow the legislative work to direct their strategic planning.

In the last decades, the automatic categorization of textual documents has become an area with wide application in the treatment of a large amount of text data and in the literature there is a considerable number of works related to this topic (Sebastiani, 2002). Many statistical classification methods and machine learning techniques were used, such as the kNN (k-Nearest Neighbor) (Tan, 2006) classifier, Naive Bayes algorithms, decision trees, generative probabilistic classifiers, multivariate regression models, among others.

However, most classifiers face serious problems in a context where there is an imbalance in the distribution of classes. Currently, there are many approaches in the literature to mitigate classification with imbalanced datasets (Wang et al., 2021), ranging from more basic techniques of subsampling and oversampling the dataset to computationally more sophisticated methods that combine neural networks with ensemble models and achieve good results (Li and Zhang, 2021). Algorithmic level-based classification methods and sensitive cost functions are also widely used to deal with problems related to imbalanced datasets (Barot and Jethva, 2021).

3 PROBLEM FORMULATION

There are different machine learning based algorithms for classification tasks. According to (Wu et al., 2007), one of the top-10 data mining classification algorithms is the kNN (*k-Nearest-Neighbors*), which

is also considered by some works as the most popular algorithm for classifying textual data and kNN has shown superior performance for textual classification when compared to other classifiers (Imandoust et al., 2013) (Trstenjak et al., 2014) (Jiang et al., 2012). However, kNN is a sample-based learning method, which uses all documents in the database to predict the labels (classes) of new documents. Classification using kNN is done assuming that similar documents will belong to the same category.

In the case of this work, there is a binary classification (two classes): a document that represents the text of a LP can be classified as “approved” or “disapproved”. Formally, considering a set of documents $D = \{d_1, d_2, d_3, \dots, d_m\}$, each $d_j \in D$ is associated with a label/class using the following function:

$$\delta(d_j) = \begin{cases} 1, & \text{if } d_j \text{ is approved} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

To classify a new document d_q , the kNN algorithm searches for the k -documents in D most similar to d_q using a similarity distance. For example, to measure the similarity between two documents d_j and d_q , we can use the similarity by cosines, which is defined by:

$$\text{sim}(d_j, d_q) = \frac{d_i \cdot d_q}{\|d_i\| \cdot \|d_q\|} \quad (2)$$

The advantage of using similarity by cosines over other distance functions is based on the fact that when working with Natural Language Processing - and even more when making a dimensionality reduction - the angles between the vectors are better preserved than their distances (Rahutomo et al., 2012).

Considering $D^k \subseteq D$ the set of k -documents most similar to the document d_q , the classic version of kNN algorithm attributes the class of the new document as the majority class of its k - neighbors, that is:

$$\text{kNN}_{\text{traditional}}(D^k, d_q) = \arg \max_{c \in \{0,1\}} \sum_{j=1}^k I(c, \delta(d_j)) \quad (3)$$

where:

$$I(a, b) = \begin{cases} 1, & \text{if } a = b \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

In the traditional version of the kNN algorithm (Eq. 3), the class of each neighbor has the same weight for the classifier. However, we can give more weight to the nearest neighbors and less weight to the distant ones. In other words, the distance-based weighted version of kNN can be defined as:

$$\text{kNN}_{\text{dist}}(D^k, d_q) = \arg \max_{c \in \{0,1\}} \sum_{j=1}^k \text{sim}(d_q, d_j) \cdot I(c, \delta(d_j)) \quad (5)$$

However, both kNN-traditional (Eq. 3) and kNN-dist (Eq. 5) face the problem of imbalance database. This means that, in both approaches, the majority class will contribute to more neighbors which will tend to define the class of the new document. Figure 1 shows this problem: we can note that among the k -neighbors closest to the yellow circle there are more blue squares than red triangles, so the yellow circle will be classified as belonging to the class of blue squares, even when its k -neighbors are all the red triangles in the database.

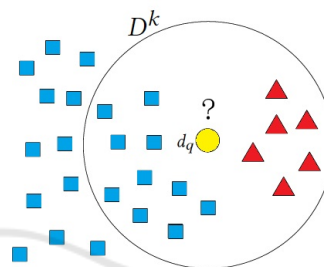


Figure 1: Example of classification in imbalanced datasets: the majority class (blue squares) contributes with more neighbors than the minority class (red triangles).

4 PROPOSED APPROACH

The proposed approach aims to represent and classify LPs based on its textual documents. For this purpose, we developed a framework defined in three steps carried out as follows:

1. representing textual documents in sparse vectors based on the TF-IDF technique;
2. reducing the vectors dimensionality using the SVD (*Singular Value Decomposition*) approach;
3. incorporating a distance-weighted factor to the kNN algorithm in order to mitigate the database imbalance and a temporal-based factor to maximize the similarity of documents close in time.

Each of these steps will be detailed as follows.

4.1 Textual Document Representation

To represent the textual documents of LPs, the proposed approach uses the TF-IDF model (*Term Frequency-Inverse Document Frequency*), which measures the importance of a word for a document

based on a collection (or corpus). The TF-IDF model is composed of two calculations: the first computes the Term Normalized Frequency (TF) and the second computes the Inverse Document Frequency (IDF).

Consider D the document set (corpus) with a vocabulary of words of size n . Let $d_j \in D$ such that $d_j = \{x_1, x_2, x_3, \dots, x_n\}$, where x_i denotes the number of occurrences of the i -th term (word) in d_j . Formally, the TF calculation of the i th word is defined as:

$$TF(i) = \frac{x_i}{\sum_{k=0}^n x_k} \quad (6)$$

The IDF is incorporated to decrease the weight of words that occur more frequently in D and increase the weight of those that occur rarely. Formally, the IDF calculation of the i th word is defined as:

$$IDF(i) = \log \left(\frac{|D|}{t_i + 1} \right) \quad (7)$$

where $|D|$ is the number of documents in the corpus and t_i denotes the number of documents in D that contain the i th word.

The final representation of each document $d \in D$ is given by the product TF and IDF of each of the n words in the vocabulary, that is:

$$TF_IDF(i) = TF(i) \times IDF(i) \quad (8)$$

for $i = 1, 2, \dots, n$.

4.2 Dimensionality Reduction

The document representation using the TF-IDF model will generate sparse n -dimensional vectors, where n is the number of words (vocabulary) in the corpus. This means that, for each document, the generated vector is of high dimensionality and, because of this, the efficiency of machine learning algorithms may be affected.

To reduce dimensionality, we use the Singular Value Decomposition (SVD) technique (Karl et al., 2015) to decompose the document matrix $D_{(m \times n)}$ in the following matrix:

$$D = USV^T \quad (9)$$

where $U_{(m \times m)}$ and $V_{(n \times n)}$ are two orthogonal matrices and $S_{(m \times n)}$ is a diagonal matrix, and m is the number of documents in the corpus and n is the number of words.

Using the decomposition of the matrix D by Eq. 9, it is possible to reconstruct the matrix D in a p -dimensional space (where $p \ll n$) considering the sub-matrix ($m \times p$) formed by the first p columns and the m rows of the respective original matrices U , S and V .

However, it is not trivial to choose the size of the dimension p that best represents the original data without losing the discriminative power of the vector. Therefore, in this work, this choice was made using brute force, where we analyzed different dimensionality values and we choose the dimensionality value with the best F1 score. This procedure will be discuss in Section 5.

4.3 Time and Imbalance Factors

After representing the LP documents in compact vectors, the next step is to use a classifier to predict the approval (or disapproval) of these documents. For this purpose, we developed two modifications in the kNN -dist algorithm (Eq. 5) to:

1. weight those documents that are closer in time;
2. handle database imbalance.

In order to weight the documents closer in time, a time-based factor was defined to measure how far two documents are in time. That is, how far two LPs were presented in the legislative houses for voting. In other words, the time factor between two documents d_i and d_q is given by the difference of the years of its proposals, which is formally defined as:

$$factor_{time}(d_j, d_q) = \log \left(\frac{TimeFrame}{|Year(d_j) - Year(d_q)| + 1} \right) \quad (10)$$

where, $Year(d_i)$ refers to the year that the document d_i was proposed and $TimeFrame$ refers to the time lapse calculated by the difference between the year of the oldest document and the newest document in the database. The smaller the value of the time-factor, the further apart (in time) two documents will be.

The proposal to deal with the imbalance consists of increasing the distance of k -neighbors belonging to the majority class, which in the case of this work refers to the class of disapproved propositions. This imbalance factor is a constant " α " that is multiplied by the similarity distance of documents $d_j \in D$ that belong to the majority class:

$$factor_{imbalance}(d_j) = \begin{cases} 1, & \text{if } \delta(d_j) = 1 \\ \alpha & \delta(d_j) = 0 \end{cases} \quad (11)$$

Formally, by integrating the time factor (Eq. 10) and the imbalance factor (Eq. 11) in the kNN_{dist} , we have the following proposal:

$$kNN_{proposed}(D^k, d_q) = \arg \max_{c \in \{0,1\}} \sum_{j=1}^k sim(d_q, d_j) \cdot I(c, \delta(d_j)) \cdot factor_{imbalance}(d_j) \cdot factor_{time}(d_j, d_q) \quad (12)$$

5 EXPERIMENTAL RESULTS

In this section, we present the experimental tests conducted to evaluate the proposed technique in the classification of LPs. First, we discuss the dataset we used to perform the experiments, next we present the performance metrics used to compare the proposed technique with other classifiers and, the obtained results.

5.1 Database

To perform the experiments, we used the open data website of the Chamber of Deputies ¹ which provides annual data files with information on each LP presented in the respective year. In this work, we downloaded the files referring to the years between 2001 and 2021 and we considered only the propositions of laws amendments types and with completed process, that is, with situation approved or disapproved. The database resulted in 28,049 documents, where 25,753 are LPs disapproved and 2,296 are LPs approved. It means that, from 2001 to 2021 there was an approval rate of only 8.2%.

After defining the database, we pre-processed the text files in order to clean and standardize them for the modeling stage. This “cleaning” and standardization in the texts is an approach carried out in natural language processing to guarantee the quality in the processing of textual documents and it contributes, among other things, to reduce the generated dictionary, as some words will be suppressed and/or encoded in the same pattern. Figure 2 shows the sequence of the four steps used in this work for the pre-processing of the LPs document texts.

5.2 Performance Evaluation Metrics

In the context of imbalanced data, it is important to consider performance measurements that provide insight about the imbalance of the database. According to (Brzezinski et al., 2020), the measure used to evaluate classifiers in an imbalanced environment must be selected individually, as each of these problems comes with its own set of challenges, that is, classical metrics are not a reliable means of evaluating a model trained on imbalanced data.

Most of the classic evaluation measures derive from a table called the Confusion Matrix, which contains the amount of correct classifications versus the predicted classifications for each class over a

set of examples, that is, it indicates the errors and successes of the model comparing with the expected results. For each class, four values can be extracted:

- TP: true positive is an outcome where the model correctly predicts the positive class;
- TN: true negative is an outcome where the model correctly predicts the negative class;
- FP: false positive is an outcome where the model incorrectly predicts the positive class;
- FN: false negative is an outcome where the model incorrectly predicts the negative class.

Based on these four variables, four evaluation metrics can be defined:

- Precision = $\frac{TP}{TP+FP}$
- Recall = $\frac{TP}{TP+FN}$
- F1-Score = $\frac{2 \times Accuracy \times Recall}{Accuracy + Recall}$
- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

Among these four metrics, the most important for this work is the F1-Score, which is a harmonic mean between Precision and Recall, which is much closer to the smallest values than a simple arithmetic mean. That is, when we have a low F1-Score, it is an indication that either the accuracy or the recall is low.

5.3 Techniques Evaluated and Parameter Settings

To perform comparative results, four classification techniques were evaluated:

- $kNN_{proposed}$ (Eq. 12)
- $kNN_{traditional}$ (Eq. 3)
- kNN_{dist} (Eq. 5)
- XGBoost

The implementation of the $kNN_{traditional}$ and kNN_{dist} classifiers need the definition of parameter K. In our experiments, this value was defined using brute force: we performed exhaustive test using different values of K within the interval 1 and 150 and, for each value, the F1-Score was calculated. Then, the value of K choose for each technique was that one with best value of F1-Score. Figure 3 shows these results.

The $kNN_{proposed}$ needs two setup parameters: the value of K and the value of the imbalance factor (Eq. 11). To choose the best values for these two parameters, we also performed tests varying the value of the imbalance factor and the value of K. Figure 4 shows a heat map of the results obtained by varying

¹<https://dadosabertos.camara.leg.br/swagger/api.html>

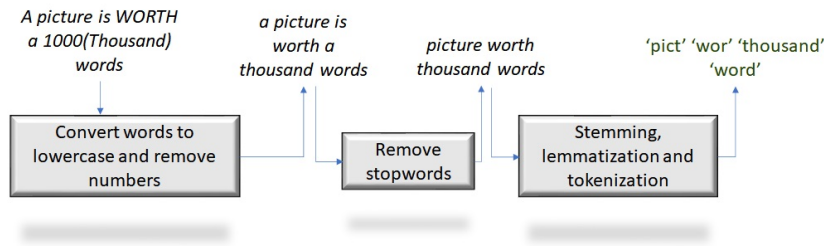


Figure 2: Pre-processing steps performed on the texts of the LPs.

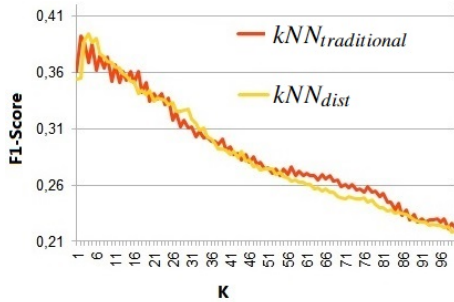


Figure 3: F1-score values obtained for each K in the best result scenarios for the $kNN_{traditional}$ and kNN_{dist} techniques.

these two parameters, where warm colors indicate high values for the F1-Score. It can be seen from Figure 4 that the best values are obtained for low values of the imbalance factor and for high values of K.

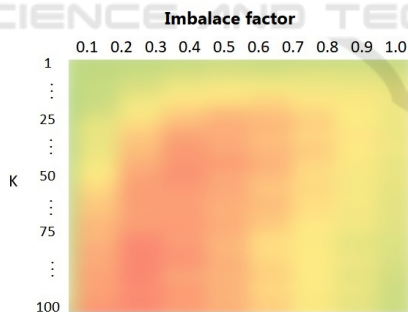


Figure 4: Heat map of the values obtained by varying the two parameters necessary to execute the $kNN_{proposed}$: warmer colors indicate higher F1-Score values.

The best value for the dimensionality reduction of the TF-IDF vector was also chosen by brute force: we started with a dimensionality equal to 100 (with increments of 100) and for each dimensionality value we computed the F1-Score. The results obtained can be seen in Figure 5. For performance reasons, in cases where there was a tie in the highest F1-Score, the dimensionality chosen was the lowest.

Table 1 summarizes the parameters used in each classifier that generated the best F1-Score values.

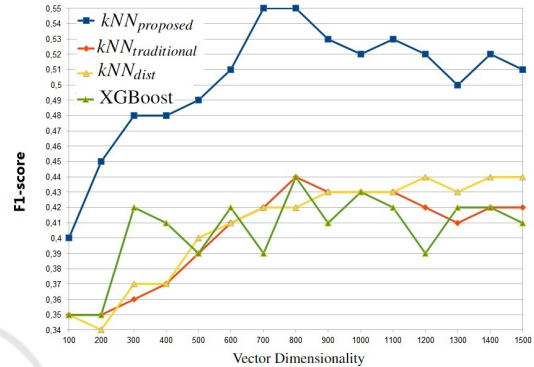


Figure 5: F1-Score values obtained by reducing the dimensionality of the original TF-IDF vector using the SVD technique.

5.4 Obtained Results

Table 2 shows the values obtained for each of the classifiers evaluated. All techniques had high accuracy values. However, this performance is not enough, as all the classifiers had low Recall values because of the imbalance in the database

The best Revocation result was obtained by $kNN_{proposed}$, which correctly classified most documents of the minority class. Due to this, $kNN_{proposed}$ also had the best F1-Score among the evaluated classifier, being 30% higher than the others. This shows that, in fact, the proposals presented in this work allowed increasing the discriminative power of the classifier when predicting the approval of PLs.

6 CONCLUSIONS

This work presented an approach that uses Natural Language Processing and Machine Learning algorithms to predict the approval of Legislative Propositions based on its textual documents. The main challenge of the proposed approach was to deal with imbalanced datasets, as there are more proposals that are disapproved than approved. Most machine learning techniques will ignore the minority class and consequently will perform poorly in that class.

Table 1: Parameters that generated the best F1-Score for each evaluated classifier.

	$kNN_{proposed}$	$kNN_{traditional}$	kNN_{dist}	XGBoost
Vector Dimensionality	700	800	1200	800
K	37	2	4	-
Imbalance Factor	0.29	-	-	-

Table 2: Results obtained for each of the techniques evaluated.

	$kNN_{proposed}$	$kNN_{traditional}$	kNN_{dist}	XGBoost
F1-Score	0.55 ★	0.44	0.44	0.44
Accuracy	0.93	0.95	0.93	0.93
Precision	0.57	0.75	0.66	0.75
Recall	0.52	0.31	0.33	0.31

However, in general, the performance in the minority class is the most important to consider, especially for the application scenario of this work.

The approach proposed in this work considered two important aspects when predicting the approval of LPs: (i) the database imbalance factor and (ii) the time in which proposals were submitted. These two aspects were mitigated by modifying the traditional kNN algorithm: we increased the distance between disapproved documents and those that were further away in time. Experimental results showed that the two modifications proposed in the kNN algorithm allowed to increase the classifier's performance. The proposed technique obtained a high recall rate among the evaluated techniques, showing its potential in predicting propositions with high chances of approval in legislative houses, contributing with a valuable tool to be used in the legislative process.

REFERENCES

- Barot, P. and Jethva, H. (2021). Imbtree: Minority class sensitive weighted decision tree for classification of unbalanced data. *International Journal of Intelligent Systems and Applications in Engineering*, 9(4):152–158.
- Brzezinski, D., Stefanowski, J., Susmaga, R., and Szczech, I. (2020). On the dynamics of classification measures for imbalanced and streaming data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(8):2868–2878.
- Cheng, Y., Agrawal, A., Liu, H., and Choudhary, A. (2017). Legislative prediction with dual uncertainty minimization from heterogeneous information. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 10(2):107–120.
- Imandoust, S. B., Bolandraftar, M., et al. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International journal of engineering research and applications*, 3(5):605–610.
- Jiang, S., Pang, G., Wu, M., and Kuang, L. (2012). An improved k-nearest-neighbor algorithm for text categorization. *Expert Systems with Applications*, 39(1):1503–1509.
- Karl, A., Wisnowski, J., and Rushing, W. H. (2015). A practical guide to text mining with topic extraction. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(5):326–340.
- Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Beijing, China. PMLR.
- Li, X. and Zhang, L. (2021). Unbalanced data processing using deep sparse learning technique. *Future Generation Computer Systems*, 125:480–484.
- Nay, J. J. (2017). Predicting and understanding law-making with word vectors and an ensemble model. *PLOS ONE*, 12(5):1–14.
- Rahutomo, F., Kitasuka, T., and Aritsugi, M. (2012). Semantic cosine similarity. In *The 7th international student conference on advanced science and technology ICAST*, volume 4, page 1.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47.
- Tan, S. (2006). An effective refinement strategy for knn text classifier. *Expert Systems with Applications*, 30(2):290–298.
- Trstenjak, B., Mikac, S., and Donko, D. (2014). Knn with tf-idf based framework for text categorization. *Procedia Engineering*, 69:1356–1364.
- Wang, L., Han, M., Li, X., Zhang, N., and Cheng, H. (2021). Review of classification methods on unbalanced data sets. *IEEE Access*, 9:64606–64628.
- Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., and Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37.
- Yang, Q. and Wu, X. (2006). 10 challenging problems in data mining research. *Int. J. Inf. Technol. Decis. Mak.*, 5:597–604.