ISSN 2278-3091
Volume 9, No.1.5, 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1691.52020.pdf
https://doi.org/10.30534/ijatcse/2020/1691.52020

The Impact of Neural Embedding Characteristics on Text Mining Tasks: Document Classification Use Case

Mariem Bounabi[1], Karim El Moutaouakil[2], Khalid Satori [3]
[1] Computer sciences, Imaging and Numerical Analysis Laboratory (LIIAN), Fes, Morocco, mariem.bounabi@usmba.ac.ma
[2] Engineering Sciences Laboratory (FPT), Taza, Morocco, karimmoutaouakil@yahoo.fr
[3] Computer sciences, Imaging and Numerical Analysis Laboratory (LIIAN), Fes, Morocco, khalidsatori@gmail.com

ABSTRACT

One of the relevant text mining tasks is the document classification, where a useful content categorization control in many domains like content analyses, retrieval information, and the recommendation systems. In general, a set of process influence the classification system effectiveness, and the data representation has an essential impact on the text categorization as we will discover in this article. Hence, the paper's goal is to adjust the Paragraph Vector-Distributed Memory (PV-DM) as a variant of the current methods for neural text representation by comparing diverse neural parameters choices control the system complexity, e.g., epoch number, and vector size. Also, we employ a collection of classifiers subsequently combined using majority voting to show the impact of the neural PV-DM embedding on the binary business sentiment analysis, and multi labeled News data classification. The experiments prove that a suitable selection of the neural embedding characteristics enhances the hybrid machine learning model to 99% accuracy for a data type.

Key words: Doc2vec, Neural parameters, Sentiment analysis, Text categorization, Hybrid ML model.

1. INTRODUCTION

The significant growth of electronic data, on the web, make access to information and find the relevant knowledge, contained in a document database, increasingly difficult. Consequently, the robust systems, like Mining sentiments systems, recommender systems, and retrieval information meaning, have achieved to address this problem. The classification task was officially identified as one of the best solutions for analyzing and extracting useful content from documents and developing the cited yield system [1][2].

A set of process governs the classification systems efficiently, such as the preprocessing of the used corpus [3]and the matrix embedding or the term weighting [4]. The first process permits a generation of the based vocabulary to produce one of the existing representation of descriptors, i.e., Boolean type [5], vector type [6], or probabilistic type [7]. Lately, the word embedded, e.g., word2vec [8][9][10] and Glove [11], patterns well suggested for word vector representations, applied to distribute the document representation. word2vec or any other similar model, presents each term of the document by a vector, which produces a large descriptor size. To solve the massive size descriptor problem, paragraph2vec or doc2vec was implemented to generate a representative vector for a complete text [12].

Moreover, Doc2vec has two architectures, i.e., Paragraph Vector-Distributed Bag of Words (PV-DBOW), and Paragraph Vector-Distributed Memory (PV-DM) version [12]. Due to its effectiveness in the related context [13], our comparative systems employ the PV-DM neural representation. Generally, the work of PV-DM requires adding an ID document vector combined with a word vector for each word in the paragraph [12]. Thus, to combine Vectors, we applied one of these methods, i.e., the add, concatenation, or standard method [12]. The choice of the merged process affects the representation and classification quality, as shown in the experimentation part of this paper.

Furthermore, our studies indicate that many neural characteristics influence the classification system productiveness. Our paper proves that the neuron network (doc2vec) parameters, e.g., epoch number and size of the word vector, must be adjusted to enhance the PV-DM performance.

After generation of the matrix embedding, practicing Doc2vec variation, we call specific Machine Learning (ML) classifiers such as SVM, Logistic Function, and the supervised Feedforward neural network merged in the following by the Majority voting technique [14]. Consequently, the given results prove that the performance of PV-DM representation depends on three primary parameters:

• The used combination method, in the projection layer of the model,

• The epoch number,

• The vector word size.

The choice of three parameters is also useful in decreasing the algorithms' complexity, the system response time, and deploying the memory loss.

Hence, we aim to find an adequate Neural Embedding (PV-DM architecture) with Hybrid ML Models (using Majority voting) to binary data analyses, use amazon customer reviews, and classify the BBC News as multi labeled documents.

We propose to organize our paper as follow:
Firstly, we present the related works in these fields. Next, a description of the several used methods and techniques which articulated in the methodology part. In the fourth section, we show the discussion and results for the proposed contribution before the conclusion.

2. RELATED WORKS

In the past decade, Le & Mikolov proposed in [12] doc2vec as an extension to word2vec [15] to learn document-level embeddings. Tow architectures, based on two-layer neural networks, characterize word2vec as a tool for vectorizing the terms of a corpus. The first type is CBOW [15], which predicts a target word from a given context, as shown in Figure 1(a). The second type is present in Figure 1(b), called Skip-gram [15], and predicts the context of a given the word.

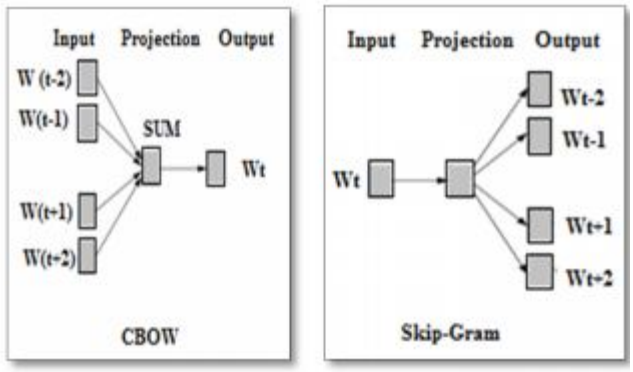

Figure 1: (a) CBOW, (b) Skip-gram word2vec architectures.

As another popular type of vector representation, where a vector represents each document's terms, is the term frequency-inverse term frequency (TF-IDF) approaches [17] [16]. Moreover, TF-IDF ignores the order of words and gives a descriptor matrix. The component of a vector term is its weight in every document in the corpus [16]. Differently to TF-IDF, word2vec gives a unique vector for each word based on the words appearing around the particular word. TF-IDF can be used either for assigning vectors to words or documents. Word2vec can be directly used to assign a vector to a word. However, to get the vector representation of a document, further processing is needed. To solve the dimensionality problem, given by word2vec representation for a large data set, doc2vec was implemented by [12] to represent sentences, paragraphs, and documents in a numeric vector. Basing on word2vec principal doc2vec produce two kinds of results context of a document or missing document [18].

Additionally, Doc2vec has been tested in different fields, and it has been used for sentiment analysis and text categorization task [19]. On the sentiment analysis task, the Doc2vec or paragraph2vec representation provides satisfying results, where the improvement is more than 16% in terms of error rate. Also, on a text classification task, the doc2vec method gives a relative improvement (= 30%) comparing to word bag patterns [12]. A further, we consider the role of the Doc2vec model in the retrieval of information field [20] [21], to realize the request matching process [20], and to measure the similarities between documents [21]. Generally, the neural network embedding performs better when using models trained on large external corpora. It can also be improved by using pre-trained word embeddings. We also provide recommendations on hyper-parameter settings for general purpose applications and release source code to induce document embeddings using the existing trained doc2vec models [22].

3. METHODOLOGY

Generally, every classification system follows the three-necessary process, as described in this section. Also, in Figure 2, we mention the adopted architecture for the text classification system. Regularly, the given tasks of the adopted classification system conform to the general case [17].

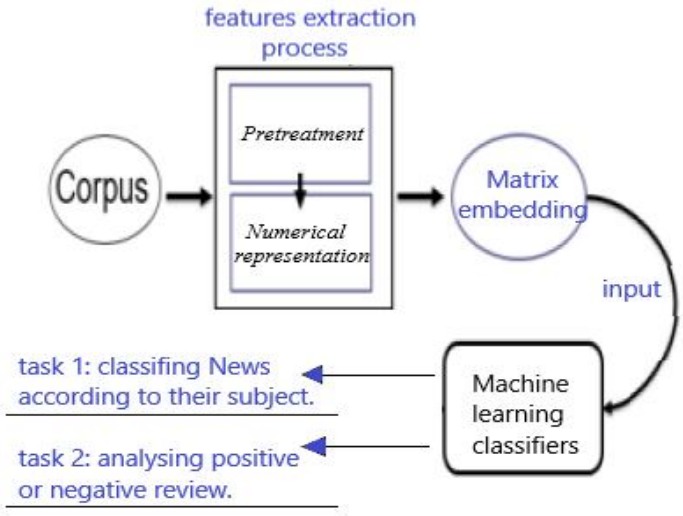

Figure 2: The adopted classification system architecture.

Based on a set of recent and useful techniques, we propose to employ the given architecture, i.e., Figure 2, which illustrates the basics of our classification system. The inputs are two different corpora, in natural language. The last outputs are the appropriate category for each document in the input database. Therefore, all processes were described in the rest of this section.

3.1 Pretreatment

In several fields, and notably on the text mining context, a set of steps is proposed to clean textual data and use numerical representation [23]. Firstly, in our work, the text is presented as a sequence of characters. As a next step, we define a function that converts text to lower-case and strips punctuation/symbols from words. The stop words were eliminated using a stop words list. Finally, the Stemming algorithms are required to find the radicals of terms [3], where we use the Lovin stemming algorithm [3]. Globally, the text pretreatment helps the next used models to predict the most results using relevant data.

3.2 Numeric text representation

For Natural Language Processing (NLP) tasks, several language modeling and training techniques, are known as word embedding, became extremely popular. word2vec became one of the wills used word embedding algorithms, giving a numerical representation of any word, and then doc2vec, working out the same function for a paragraph or a document. In this section, we describe the numeric representation of documents in a corpus, after pretreating data. Doc2vec aims to build a numeric representation of a text, regardless of its length. The word2vec model used by inserting another paragraph ID vector. Globally, it is like the CBOW principal, where the tow vectors, word vector, and ID document vector, are used to produce the final document vector. Besides, the un-supervised Doc2vec model has two forms:

- Distributed Bag of Words version of Paragraph Vector (PV-DBOW) like the skip-gram principle. The algorithm is faster to execute (as opposed to word2vec), and it is unnecessary to save the word vectors, the proposed model does not require a large memory [12].
- Paragraph Vector-Distributed Memory (PV-DM) version works like a memory that remembers the missing in the current context or the document's subject. Moreover, the document vector intends to represent the document concept [12].

Regularly, the doc2vec models provide a collection of documents for preparation. In the first step, a word vector for each word and a document vector for every document, are generated. Secondly, the algorithm also trains the weights of a SoftMax hidden layer. Furthermore, a new document can be introduced at the inference point. All weights shall be set for the computation of the document variable. Especially For the PV-DM model, several fusion methods were proposed in the projection layer, as average or concatenation functions, explained in [12]. In the experimentation part, we describe the fusion function impact on the text classification performance, the importance of neural network parameters, and the output vector size. In our work, we use the PV-DM architecture to the embedding matrix because it is useful for our context, as mentioned in [13].

3.3 Machine Learning classifiers

Once the embedding matrix generated, we use some Machine learning models to analyze amazon customers reviews and to categorize electronic News according to their subject, like:

Support Vector Machine [24] supervised learning systems with related learning algorithms that analyze the data used for classification and regression analysis. The main goals of this algorithm are to locate a hyperplane in the N-dimensional space of the features number that specifically identifies the data points. Logistic Function or the multinomial logit model [25] is the most often used multi-categorical regression model. Specifies the conditional probability of answering groups through the linear Function of the covariate vector x. When the logistic model suffers from problems such as complete separation, the parameters' estimates are not uniquely defined. The regularization methods, such as ridge regression, was added to overcome such problems.

Feedforward neural network (FNN) is an acyclic neuron network with no cycles or loops in the network. Such models employed: feedforward, because the data feeds through the Function being evaluated from the input vectors, through the intermediate calculations used to describe the hidden layer procedure, and eventually to the defined classes. There are no input relations in which the model's inputs are fed back into themselves [26]. If feedforward neural networks are generalized to include feedback connections, they are called recurrent neural networks.

The Hybrid ML model used to merge the classifiers. The voting method decides what the class value of each classifier is the output, by assigning the input sequence to the class with a majority vote. Several combination functions, including sum, product, max, min, average, and median functions, were proposed [27].

4. EXPERIMENTATIONS AND RESULTS

The whole of the algorithm’s studies has been implemented with java language, which favors our comparative study. To compare different recognition systems, we use a compatible Dell, Intel (R) Core i5- CPU 2.50 GHz, and 4 GB of RAM.

4.1 Experimentations setup

A. Dataset

The following benchmarking datasets, as English corpora, are used to realize this work:

- BBCSport as multi labeled data contains news classified in five classes. The CSV file, associated with BBCSport, is composed of two columns (News and Classes) [28].
- For business analysis sentiment, we employ Amazon reviews data (available on Kaggle site web).

B. The Classification parameters

We start our experimentation by the data pre-processing step. This process permits a considerable improvement in the classification phase. Next, the matrix embedding presents the inputs of a selection of Machine learning classifiers. Each classifier has its parameters, which allow the improvement of the tasks according to the problem.

In this paper, for the SVM model, we use the polynomial kernel as a kernel function.

To classify with the feedforward neural network (FNN), we test with the DL4J library, where the SoftMax is the activation function. The epoch's number fixed on 10.

To combine the set of used classifiers, we use as Hybrid ML the vote technic with the majority combiner functions [27].

The inputs, for the proposed classifiers, are calculated based on the unsupervised Doc2vec model.

We aim to prove, in the previous sections, that the change of Neural network Doc2vec parameters change the score of the classification using the chosen classifiers. Also, by the presented results, we try to find the optimal size of the generated vectors and epoch numbers.

4.2 Results and Discussion

A. Performance Measures and Models learning

- To evaluate the performance of classification systems, we use the precision, recall, and accuracy measures [29], as shown in Table 1.

Table 1: Attributes of Cleveland dataset

Measure	Formula
Recall	TP/(TP+FP)
Precision	TP/(TP+FN)
Accuracy	(TP+TN)/(TP+FP+TN+FN)
TP: True Positive; FP: False Positive. FN: False Negative; TN: True Negative.	

- The K-Cross-validation is a way of predicting a model's ability on a possible validity system, when a separate and explicit validation set is not available, or when the problem is overfitting [30]. In these experiments, we use 10-Cross-validation to learn and test the used classifiers on the classification phase.

B. Multi labeled data classification Results and discussion

Using the News data, the giving Tables (2, 3, 4, and 5) show the results of the classification systems based on PV-DM, as NUMBER version, and a set of classifiers (SVM, FNN and Logistic function), combined, in the following, by the Majority voting. The comparison is based on three primary parameters:

- The projection layer type of the model. (Add, Concatenate, and Average)
- The epoch numbers.
- The vector word size.

The mentioned parameters have a significant impact on the compressed dimensionality of the given inputs and the systems' response time.

Table 2: classification results using PV-DM representation (with epoch number=1 and vector size=100), a set of combination methods and classifiers.

	Epoch Number=1 & vector size=100								
	Doc2_Average			Doc2_Add			Doc2v_concat		
	Precision %	Recall%	Accuracy%	Precision %	Recall%	Accuracy %	Precision %	Recall%	Accuracy %
SVM	90.8	90.8	90/77	91.0	90.6	90.6	75	74.4	74.3
LF	91.9	91.7	92.1	92.1	92.1	76	75.1	75.3	75.3
FNN	89.4	89.1	98.1	90.3	90.4	90.3	75.6	75.4	75
Maj vote	90	90	90	94	94	94	76.7	75.3	75.3

Table 3: classification results using PV-DM representation (with epoch number=5 and vector size=100), a set of combination methods and classifiers.

	Epoch Number=5 & vector size=100								
	D2V_Average			D2V_Add			D2V_concat		
	Precision %	Recall%	Accuracy%	Precision %	Recall%	Accuracy %	Precision %	Recall%	Accuracy %
SVM	97.5	97.4	97.4	96.4	96.2	96.2	91.5	91.4	91.4
LF	96.1	96.1	96	91.2	91.2	91.1	82.3	82.9	82.9
FNN	97.3	97.3	97.2	97.2	97.1	97.1	90.6	90.7	90.6
Maj vote	96.2	96.2	96.2	92.4	92.3	92.2	89.9	89	89

Table 4: classification results using PV-DM representation (with epoch number=1 and vector size=300), a set of combination methods and classifiers.

	Epoch Number= 1 & Vector size= 300								
	D2V_Average			D2V_Add			D2V_concat		
	Precision %	Recall%	Accuracy%	Precision %	Recall%	Accuracy %	Precision %	Recall%	Accuracy %
SVM	89.1	89	89	86.4	86.4	86.4	75 .1	74.9	74.9
LF	91.9	91.7	92.1	92.1	92.1	76	75.1	75.3	75.3
FNN	88.3	88.2	88.1	86	85.9	85.5 5	74.2	74.4	74.2
Maj vote	94.3	94.3	92.4	92.4	92.3	92.3	78	78	78

Table 5: classification results using PV-DM representation (with epoch number=5 and vector size=300), a set of combination methods and classifiers.

	Epoch Number= 5 & Vector size= 300								
	D2V_Average			D2V_Add			D2V_concat		
	Precision %	Recall%	Accuracy%	Precision %	Recall%	Accuracy %	Precision %	Recall%	Accuracy %
SVM	97.7	97.7	97.6	97	97	97	91	90	90
LF	98.1	98.1	98.1	97.7	97.6	97.6	79.6	79.6	80
FNN	98.1	98.1	98.1	98.1	96	96	90	90	90
Vote	98.1	98.1	98.1	97	97	97	91	90	90

The change of NUMBER parameters changes the classification performance. Due to the low accuracy given by the concatenate method, it is eliminated in this comparison. Besides, PV-DM architecture, which uses the average method in the projection phase, gives the most results. Optimal epoch numbers permit to reduce the iteration number, which produces a small algorithm complexity.

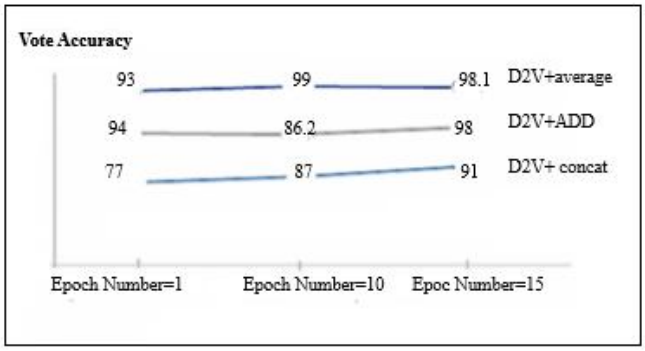

Figure 3: voting Accuracy based on the variation of PV-DM parameters (with a fixed vector size = 100, and a change in the number of epochs).

Figure 3 shows that the number of epochs has an important influence on Neural embedding and classification performances. The presented confusion Matrix, Figures 4 and 5 prove that a good choice of the epoch number, reduce the False positive rate, from 0,025 to 0.002, in the classification phase.

Class 1	Class 2	Class 3	Class 4	Class 5	
101	0	0	0	0	**Class 1**
0	123	1	0	0	**Class 2**
2	0	263	0	0	**Class 3**
1	0	1	145	0	**Class 4**
0	0	0	0	100	**Class5**

Figure 4: Confusion Matrix of a system based on PV-DM with a good choice of epoch Numbers, using BBC Sport dataset.

Class 1	Class 2	Class 3	Class 4	Class 5	
99	1	0	0	1	**Class 1**
1	116	1	6	0	**Class 2**
3	1	247	8	6	**Class 3**
2	6	13	123	3	**Class 4**
5	1	2	2	90	**Class5**

Figure 5: Confusion Matrix of a system based on PV-DM with a bad choice of epoch Numbers, using BBC Sport dataset.

In Table 6, we foxed on the PV-DM with the average method to define the best classification system. The set of values, i.e., percentages, correspond to the majority voting accuracy. Several vector size and epoch numbers tested in order to define the best PV-DM representation.

Table 6: Summary of the multi labeled data classification results using the voting classifier and a variation PV-DM neural embedding parameter.

Parameters	Voting Accuracy
Vector size=100 & Epoch Number=1	93%
Vector size= 100 & Epoch Number= 10	99%
Vector size= 100 & Epoch Number=15	98.1%
Vector size=300 & Epoch Number=5	97.6%
Vector size= 300& Epoch Number=10	98.8%
Vector size= 300& Epoch Number=15	99.1%
Vector size= 500& Epoch Number=6	97.6%

As shown, the system based on PV-DM+ average, as a combination method with vector size=100 and epoch number=10, to classify BBC sports data with the majority vote, gives an excellent accuracy = 99%.

The choice of the vector size =100 depends on the optimal response time, where the response time is reduced by 40% when we generate vectors of size = 100, compared to the size vector = 300. Also, the given selection assures a significant reduction of dimensionality to work with the relevant features.

Another way to visualize the classification's performance is the receiver operating characteristic (ROC) curve [29]. Thus, we propose the ROC curve for the best-given results employing the voting classifier. Looking to Figure 6, the shape of the curve is perfect, as the measure Under Roc Area (UAC) [29] exceeds 99%, which means the efficiency of the classification and a good impact of the choice of the characteristics of the D2V representation.

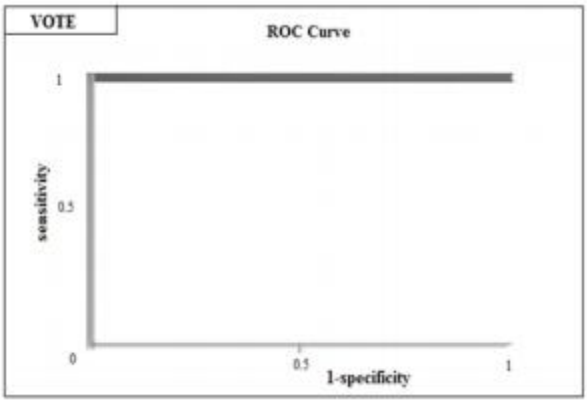

Figure 6: ROC curve for voting technique, using the best D2V characteristics and multi labeled data set.

C. Amazon business sentiment data Results and discussion

The famous Kaggle competition launched the challenge to analyze a million of Amazon reviews customers as the relevant content for developing e-commerce requests. Seen that our material does not support the whole base, we have to test on a portion of the cited database, i.e., 4002 Amazon customer reviews (2001 positive reviews and 2001 negative reviews). The partition of amazon data is good enough to confirm the neural parameter impact on the representation and the classification of the document as we will discover in this part.

Practicing always the average method, for the PV-DM doc2vec version, we suggest a variety of values for the rest of the parameters, i.e., vector size and Epoch number.

Table 7: The amazon data classification results using the voting classifier and a variation PV-DM neural embedding parameter

Parameters	Voting Accuracy
Vector size=1000 & Epoch Number=10	90%
Vector size= 1000& Epoch Number=15	82%
Vector size=300 & Epoch Number=10	90%
Vector size=100 & Epoch Number=5	94%
Vector size=100 & Epoch Number=1	92%
Vector size=500 & Epoch Number=10	82%

Accordingly, to classify amazon's business sentiment data, we need to follow the standard classification system architecture. Moreover, to choose the adequate neural parameters for the PV-DM representation method, which influence the sentiment analysis accuracy as we chow in Table 7.

positif	negatif	
1803	195	Positf
192	1808	negatif

Figure 7: Confusion Matrix of a system based on PV-DM with a good choice of epoch Numbers, using Amazon data.

positif	negatif	
1685	316	Positf
312	1689	negatif

Figure 8: Confusion Matrix of a system based on PV-DM with a bad choice of epoch Numbers, using Amazon data

Each parameter, i.e., epoch number, vector size, and the projection layer type of the PV-DV, controls the performance of the analysis system. Including the Confidence Matrixes (Figures 7 and 8), we demonstrate the transition of reviews customers to the appropriate class. They are hence reducing the False positive rate when, for example, the choice of epoch member is successful.

Also, the given ROC curve in Figure 9 illustrates the performance of a correct choice of D2V parameters using the business data, where the UAC score is 95%.

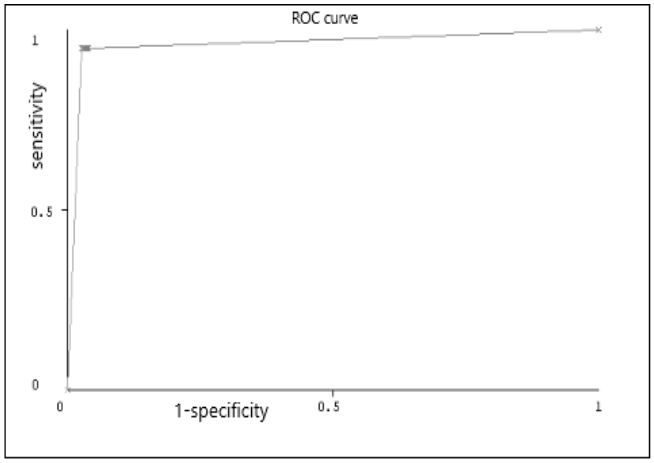

Figure 9: ROC curve for voting technique, using the best D2V characteristics and business sentiment data set

Besides, Figure 10 proves that for perfect sentiment data analyses, it is necessary to adjust the all mentioned neural embedding parameters. The curve also confirms that adjusted PV-DM architecture enhances the categorization quality using the hybrid ML model's voting technique.

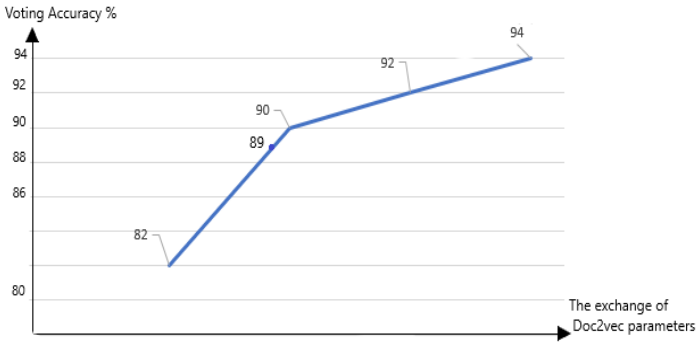

Figure 10: Evolution curve for vote accuracy applying several neural PV-DM parameters.

Therefore, we have illustrated the impact of doc2vec neural characteristics on the document classification quality as a task of text mining from the given examples. Moreover, an optimal selection of these parameters decreases the used algorithms complexity, the response time of the system, and deploy as well as the memory optimization. As illustrate in table 7, 100 vectors are sufficient to present a corpus of 4000 instances. Furthermore, we can insert the selection features [31] algorithms to enhance the classification quality, and to reduce more the descriptors dimensionality.

It should be noted that there is no general case for the choice of parameters; each database requires its case study. The given methods and analyses could also apply to other contexts for NLP as the Scientific publication topic [32].

5. CONCLUSION

The main objective is to present a detailed analysis that allows Neural embedding users to understand the principle of Doc2vec and to profit from these advantages. Differently from the existing, we offer a set of parameters in this paper, which control the effectiveness of the PV-DM document representation, i.e., epoch number, vector size, and the PV-DM combination method projection layer. Also, an optimal alternative of the given parameters allows to decrease the complexity of the used algorithms, reduce the response time, and reduce memory loss. Notably, using the benchmarked dataset and a set of ML classifiers, and especially the Vote technique, we confirm the impact of the neural embedding characteristics on the text categorization task. As shown in the results section, after a variety of parameter selection, the tested systems' recognition rate is satisfied. The accuracy for multi labeled data is 99% and 94% for business sentiment analyses purpose. Hence, we propose to automate the neural parameters selection according to the employed dataset, as future works, to benefit from the word embedding methods with the deep neural networks and manage the big data.

REFERENCES

1. Lee, Sang-Gi, et al. **A study of intelligent recommendation system based on naive bayes text classification and collaborative filtering**. *Journal of information management* 41.4 (2010): 227-249.
2. Sinoara, Roberta A., et al. **Knowledge-enhanced document embeddings for text classification**. *Knowledge-Based Systems* 163 (2019): 955-971.
3. Bounabi, M., Moutaouakil, K. E., & Satori, K. **A comparison of text classification methods using different stemming techniques**. *International Journal of Computer Applications in Technology*, (2019), 60(4), 298-306.
4. Wang, G., Li, C., Wang, W., Zhang, Y., Shen, D., Zhang, X., ... & Carin, L. **Joint embedding of words and labels for text classification,** 2018, arXiv preprint arXiv:1805.04174.
5. Hamon, D. **System and method providing a binary representation of a web page**. *U.S. Patent No 9,298,679, 29* mars 2016.
6. Tao, Y., Cui, Z., & Wenjun, Z. (2018, October). **A Multi-Label Text Classification Method Based on Labels Vector Fusion**. In 2018 *International Conference on Promising Electronic Technologies (ICPET*) (pp. 80-85). IEEE.
7. Zhai, C. **Probabilistic topic models for text data retrieval and analysis**. (2017, August) *In Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval* (pp. 1399-1401). ACM.
8. Xie, F., Wu, X., & Zhu, X. **Efficient sequential pattern mining with wildcards for keyphrase extraction.** *Knowledge-Based Systems*, 2017, 115, 27–39. doi: 10.1016/j. knosys.2016.10.011.
9. Baroni, M. Dinu, G., & Kruszewski, G. Don't count, predict! **A systematic comparison of context-counting vs. context-predicting semantic vectors**. *In Proceedings of the fifty-second annual meeting of the association for computational linguistics*, 2014, 1 (pp. 238–247). (Long papers)
10. Chaturvedi, I., Ong, Y., Tsang, I. W., Welsch, R. E., & Cambria, E. **Learn- ing word dependencies in text by means of a deep recurrent belief network.** 2016, Knowledge-Based Systems, 108, 144–154. doi: 10.1016/j.knosys.2016.07.019.
11. Pennington, J., Socher, R., & Manning,C. **Glove: Global vectors for word representation**. *In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, (pp. 1532–1543).
12. Le, Q., & Mikolov, T. **Distributed representations of sentences and documents**. *In International conference on machine learning*, 2014, (pp. 1188-1196).
13. Dai, A. M., Olah, C., & Le, Q. V. **Document embedding with paragraph vectors**. *arXiv preprint arXiv*:1507.07998, 2015.
14. Dietterich, T. G. **Ensemble methods in machine learning**. *In International workshop on multiple classifier systems*, 2000, (pp. 1-15). Springer, Berlin, Heidelberg.
15. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. **Distributed representations of words and phrases and their compositionality**. *In Advances in neural information processing systems*, 2013, (pp. 3111-3119).
16. Bounabi, M., El Moutaouakil, K., & Satori, K. **A comparison of Text Classification methods Method of weighted terms selected by different Stemming Techniques**. *In Proceedings of the 2nd international Conference on Big Data, Cloud and Applications*, 2017, (p. 43). ACM.
17. Bounabi, M., El Moutaouakil, K., & Satori, K. **A Probabilistic Vector Representation and Neural Network for Text Classification**, *In International Conference on Big Data, Cloud and Applications* (pp. 343-355), 2018, Springer, Cham.
18. Trieu, L. Q., Tran, H. Q., & Tran, M. T. **News classification from social media using twitter-based doc2vec model and automatic query expansion**. In Proceedings of the Eighth International Symposium on Information and Communication Technology, 2017, (pp. 460- 467). ACM
19. Le, Q., & Mikolov, T. **Distributed representations of sentences and documents.** *In International conference on machine learning*, 2014, (pp. 1188-1196).
20. Thanda, A., Agarwal, A., Singla, K., Prakash, A., & Gupta, A. **A Document Retrieval System for Math Queries**. *In NTCIR*, 2016.

21. Dynomant, E., Darmoni, S. J., Lejeune, É., Kerdelhué, G., Leroy, J. P., Lequertier, V., ... & Grosjean, J. **Doc2Vec on the PubMed corpus : study of a new approach to generate related articles**. *arXiv preprint arXiv:*1911.11698, 2019.
22. Lau, J. H., & Baldwin, T. **An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation**, July 2016.
23. PATEL, Priti S. et DESAI, S. G. **A comparative study on data mining tools**. *International Journal of Advanced Trends in Computer Science and Engineering*, 2015, vol. 4, no 2.
24. N. Aharrane, K. El moutaouakil, and K. Satori. **A comparison of supervised classification methods for a statistical set of features:Application**, *IEEE Amazigh OCR. In Intelligent Systems and Computer Vision (ISCV)*, pp. 1-8, March 2015.
25. Zahid, F. M., & Tutz, G. (2013). **Ridge estimation for multinomial logit models with symmetric side constraints**. *Computational Statistics*, 28(3), 1017-1034.
26. Svozil, D., Kvasnicka, V., & Pospichal, J. **Introduction to multilayer feed-forward neural networks**. *Chemometrics and intelligent laboratory systems*, 1997, 39(1), 43-62
27. Dietterich, T. G. **Ensemble methods in machine learning**. *In International workshop on multiple classifier systems*, 2000, (pp. 1-15). Springer, Berlin, Heidelberg.
28. D. Greene and P. Cunningham. **Practical Solutions to the Problem of Diagonal Dominance in Kernel Document Clustering**, Proc. ICML 2006.
29. M. Sokolova, N. Japkowicz and S. Szpakowicz: Beyond **Accuracy, FScore and ROC: A Family of Discriminant Measures for Performance Evaluation**, *Lecture Notes in Computer Science*, Vol. 4304, 2006, pp. 1015-102.
30. Krstajic, D., Buturovic, L. J., Leahy, D. E., & Thomas, S. **Cross-validation pitfalls when selecting and assessing regression and classification models**. *Journal of cheminformatics*, 2014,6(1), 1-15.
31. Kumar, K., Kumar, G., & Kumar, Y. **Feature selection approach for intrusion detection system**. *International Journal of Advanced Trends in Computer Science and Engineering (IJATCSE)*, 2013, 2(5), 47-53.
32. Mifrah, S., EL habib benlahmer , Y., Mifrah, & Ezeouaty, M. **Toward a Semantic Graph of Scientific Publications**: **A Bibliometric Study.** *International Journal of Advanced Trends in Computer Science and Engineering*, 2020, vol. 9, no 3.