Scientific paper classification using Convolutional Neural Networks

Conference Paper · October 2019

DOI: 10.1145/3372938.3372951

CITATION
1

READS
512

3 authors:

Monir Ech-Chouyyekh
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes
1 PUBLICATION **1** CITATION
SEE PROFILE

Hicham Omara
Abdelmalek Essaâdi University
17 PUBLICATIONS **48** CITATIONS
SEE PROFILE

Mohamed LAZAAR
Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes
71 PUBLICATIONS **421** CITATIONS
SEE PROFILE

Scientific paper classification using Convolutional Neural Networks

Monir ECH-CHOUYYEKH †
Department of Computer Science
ENSIAS, Mohammed V
University in Rabat, Morocco
monirechchouyyekh@gmail.com

Hicham OMARA
Department of Computer Science
FS/Abdelmalek Essaadi University
Tetouan, Morocco
1monirechchouyyekh@gmail.com

Mohamed LAZAAR
Department of Computer Science
ENSIAS, Mohammed V
University in Rabat, Morocco
m.lazaar@um5s.net.ma

ABSTRACT

The Convolutional Neural Network (CNN), a class of artificial neural networks, has become dominant in various fields, including analysis and text processing. It designed to learn, automatically and adaptively, the spatial hierarchies of backscattered entities using multiple building blocks, such as convolutional layers, grouping layers, and fully connected layers. This article aims to present an approach based on CNN to classify scientific articles by their domains (7 different domains) from their abstracts, the process will base on several features extracted automatically from their summaries. The proposed approach has shown remarkable results for the automatic classification of text compared to other usual automatic learning algorithms.

KEYWORDS

Text classification, Convolutional Neural Network (CNN), Deep Networks, Natural Language Processing (NLP).

1 Introduction

For now, the size of digital documents continues to grow and the overload on the web continues to increase; and subsequently, the automatic classification of these documents becomes a necessity.

Documents classification is a classic problem in the field of Natural Language Processing (NLP) [1]–[3]. Its purpose is to mark the category to which the document belongs. The classification of documents has a wide range of applications, such as thematic tags [4], classification of feelings [5], and so on. The classification of texts aims to group similar texts, thematically close, within a single set. The advantage of this approach is to organize the knowledge so that it can subsequently perform an effective search or retrieval of information [6].

BDIoT'19, October 23–24, 2019, Rabat, Morocco

ACM ISBN 978-1-4503-7240-4/19/10…$15.00
https://doi.org/10.1145/3372938.3372951

With advances in computer vision, CNNs find their place in the field of classification, particularly in the field of natural language processing [2]; they have proven very successful for text classification tasks. NLP involves the use of these algorithms to analyze or synthesize language, whether written or spoken.

In general, CNN networks trained in text classification tasks process word-level sentences representing individual words as vectors. While this approach may seem consistent with the way humans treat language, recent studies have shown that CNNs that process character-level sentences can also achieve remarkable results [7].

In 2011, Ronan et al. developed an end-to-end neural network model with convolution and clustering layers that can be used to solve a variety of fundamental natural language processing problems [2]. Yoon Kim presented in 2014 a paper on the use of CNN for sentence level classification, and formed a simple CNN with a convolution layer on word vectors obtained from an unsupervised neural language model [8]. Zhang and Wallace conducted a single-layer CNN sensitivity analysis in 2015 to examine the effect of architectural components on model performance. They then distinguished between important and relatively unimportant design decisions for sentence classification, and focused on single-layer CNNs (excluding more complex models) because of their comparative simplicity and robust empirical performance, making them a modern standard basic method similar to the support vector machine (SVM) and logistic regression [9]. Xiang et al. proposed in 2016 an empirical exploration of the use of character-level convolution networks (ConvNets) for text classification. They showed that character-level convolution networks could produce state-of-the-art or competitive results. They also proposed comparisons with traditional models such as the word bag, n-grams and their TFIDF variants, and deep learning models such as word-based ConvNets and recurring neural networks [10].

At the end of this paper, we quote the proposal of Collobert et al. in [11]. They considered that the neural representation of a word is called word embedding and is a re-evaluated vector. Word embedding allows us to measure the relationship between words by simply using the distance between two embedding vectors.

In this paper, we use the convolutional neural network, for the classification of scientific articles, based on the processing and analysis of these summaries, and we compare the results obtained by CNN, with other results obtained by other machine learning algorithms such as the Bayesian network, linear regression, Support vector machines (SVM).

The rest of this paper will be organized as follows: sections 2 and 3 describe the machine learning algorithms we used for the comparison, and we quote the architecture and functionality of the deep CNN, and the algorithm we propose for classifying scientific articles. In section 4, a discussion of the results obtained when applying the proposed model to classify scientific articles according to their field. Finally, we conclude the document.

2 Machine Learning

Machine learning, a sub-domain of artificial intelligence (AI), aims to understand the structure of data and integrate them into models understandable by everyone.. In machine learning, different types of classifiers are installed to obtain the highest degree of accuracy and efficiency. These classifiers share common characteristics but, in general, each has advantages and disadvantages, depending on the domain and data developed.

In this article, we will present the machine learning algorithms used for the classification of scientific articles. In the next section, we will introduce other machine learning algorithms with which to compare our approach.

2.1 Naïf Bayes classifier

Naive Bayes Classifier (NBC) is a generative model widely used in information research. Many researchers have approached and developed this technique for their applications. We start with the most basic version of NBC that has been developed using the technique of extracting term frequency characteristics (word count) by counting the number of words in documents.

In simple terms, a naive Bayesian classifier assumes that the existence of a characteristic for a class is independent of the existence of other characteristics. Depending on the nature of each probabilistic model, naive Bayesian classifiers can be trained effectively in a supervised learning context. In many practical applications, parameter estimation for naive Bayesian models is based on maximum likelihood [12].

The advantage of the naive Bayesian classifier is that it requires relatively little training data (documents) to estimate the parameters necessary for classification, i. e. means and variances of the different variables. Indeed, the hypothesis of the independence of the variables makes it possible to be satisfied with the variance of each of them for each class, without having to calculate a covariance matrix. Unlike short documents, long documents are a problem for the classifier Naïf Bayes; a rich vocabulary promotes dependencies between descriptors (words).

2.2 Vector support machines (SVM)

Vector support machines (SVM) are at the origin of new categorization methods, although the first publications on the subject date back to the 1960s. The SVM is a so-called linear classifier, which means that, in the perfect case, the data (text document in our case) must be linearly separable [13]. Thus, our corpus is represented as a vector space, where each text document is represented by a point in it. The problem now is to find the best separator (line, plane or hyper-plane) that divides our corpus into two categories (the case of two classes). The space between these two categories is called the margin, which is defined by the points (Support Vectors) closest to the separator on either side. The main aim is to maximize this margin, so the larger the margin, the better the result.

SVM are well suited for text classification because a large size does not affect them since they protect against overlearning. In other words, he argues that few attributes are useless to the classification task and that SVM avoid aggressive selection that would result in a loss of information [14]. We can afford to keep more attributes. Also, a feature of textual records is that when they are represented by vectors, a majority of entries are null and void [15]–[17].

However, if the data are not linearly separable, the SVM can be modified to tolerate a minimum number of errors. From now on, the goal is to maximize the margin and minimize the classification error. Another alternative to address the problem of data inseparability is to access a higher dimension.

2.3 Regression techniques

Regression technique is a supervised classification algorithm. It model the relationship between predictive variables and a target variable by a mathematical prediction function. The simplest case is univariate linear regression. It will find a function in the form of a line to estimate the relationship. Multivariate linear regression occurs when several explanatory variables are involved in the prediction function. And finally, polynomial regression makes it possible to model complex relationships that are not necessarily linear [18].

Logistic regression is a statistical method for performing binary classifications. It takes qualitative and/or ordinal predictive variables as inputs and measures the probability of the output value using the sigmoid function. Multi-class classification can be performed (for example, as our objective to classify materials into several classes (domains).

2.4 Convolutional neural network

Text classification is the process of grouping documents into classes and categories according to their content. This process is becoming increasingly important due to the enormous textual information available online. The main problem with text classification is how to improve the accuracy of the classification.

Most classification algorithms require the construction of a multidimensional characteristic vector used as the input to the algorithm. Experts are therefore essential to determine the characteristic vectors of the desired operation. A different and innovative approach is not to use an expert to build the characteristic vector, but to extract it automatically using a learning algorithm. Deep learning algorithms have the ability to learn functions useful for the classification task automatically. Convolutional neural networks (CNN) are variants of these learning algorithms [19], [20].

Convolutional neural networks are very similar to ordinary neural networks; they are composed of neurons whose weights and bias can be learned. Each neuron receives some inputs, executes a point product and can follow it with a non-linear function. The entire system always expresses a single differentiable score function: from inputs to classes in outputs. Moreover, they always have a loss function on the last layer.

CNN is typically a sequence of layers, and the outputs of each layer are the inputs of the next layer (Figure 1). These layers are: convolutional layers, pooling layers and fully bonded layers [21]. In the following section, we will see in detail the process of each layer in the classification of scientific articles.

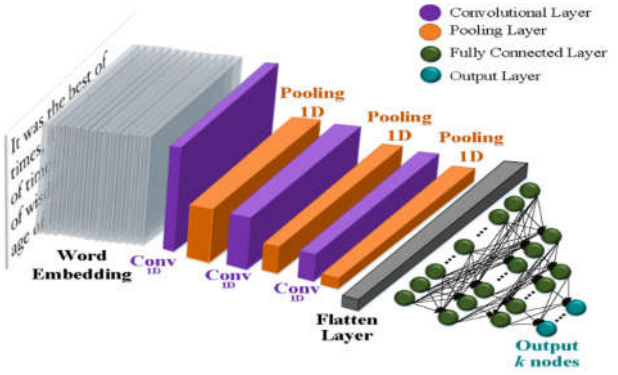

Figure 1: CNN architecture for document classification

3 Methodology

The proposed method consists of using the supervised machine-learning algorithm, CNN, for the classification of scientific articles based on their abstracts. Firstly, we use text-preprocessing principles for the construction of data compatible with the CNN. In this step, we used the methods called "Word Embedding and TF-IDF" [22], [23], this choice is based on many experiences and comparison with another approach. Then, we use CNN to train our model with specific parameters to make the classification of scientific articles, we chose a network contains a one-dimensional convolution layer, with an equal nucleus size 3 that specifies the length of the 1D convolution window.

The figure 2 presents the flow chart of our algorithm. In *Pre-Processing and Representation of documents*, we applied the basic functions for preprocessing which include: delete blank words, delete foreign characters, delete punctuation, and delete numbers... Among these cleaning functions, we used the Tokenization function that breaks down a text flow into words, sentences, symbols or other significant elements called tokens [24]. In addition, there are many functions for extracting the characteristics of the text, which transform each summary into vectors, components of these vectors, in the form of weights. Each weight corresponds to the word frequency of each article. They are many transformation functions in literature as Count Vectors as features, Term Frequency-Inverse Document Frequency (TF-IDF) [22], and Word Embedding.

In this work, we used word embedding as a function of extracting text characteristics where each word or sentence in the vocabulary is associated with a dimensional N vector of real numbers. Various word integration methods have been proposed to translate uni-grams into understandable inputs for machine learning algorithms. This work focuses on Word2Vec, GloVe, and FastText. These three methods are the most used in deep learning.

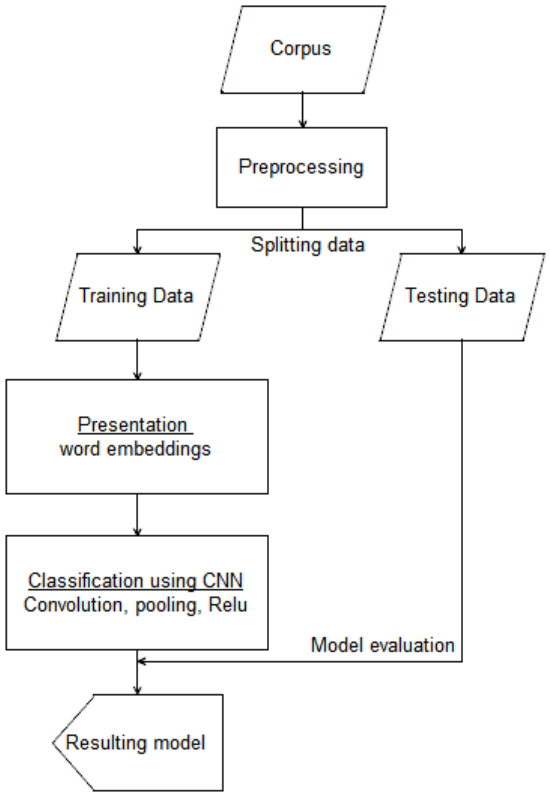

Figure 2: System architecture

After this long pretreatment of document, we are going to the learning stage, *Classification using CNN*. In our model, the typical convolutional layer, the pooling layer and the fully connected layer are included and used as follow:

Input matrix. Let s be the number of summaries and d the size of the word vector, hence an input matrix of the form $s \times d$. We have set the entry to 35238 abstract and the size of the word vectors is 5000. Which gives us a size matrix $35238\ x\ 5000$.

Filters. A filter size was set for all filters to match the word vectors and the region size was changed from one filter to another. The size of the region corresponds to the number of lines in the input matrix representing the words s. We have chosen here to use three sizes of region "h", each region recovers respectively two, three and four summaries at time. In addition, three filters were selected for each region. In general, there are nine filters.

Feature map. In this step, we explain how convolutions/filtering are performed by CNN. We have filled the input matrix with some results from the pre-treatment step, and the filter matrix is filtered with some numbers for clarity and explanation.

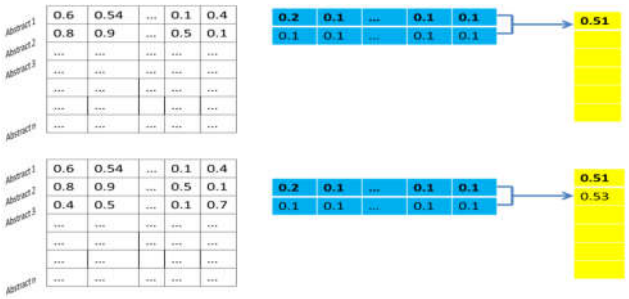

Figure 3: How Convolutions work (filtering)

At the beginning, the two-summary filter, represented by the yellow matrix (2×5000) in figure 3, is superimposed on the vectors of the words summary 1 and summary 2, then performs a product by elements for all its elements (2×5000), then adds them together and obtains a number $(0.6\times0.2 + 0.5\times0.1 +... + 0.4\times0.1 = 0.51)$. 0.51 is recorded as the first element of the output sequence. To obtain the 'c' feature maps, we add a bias term and apply the activation function ReLU (Rectified Linear Unit function).

Max-pooling. The dimensionality of c depends on both s and h, in other words, it changes through filters of different region sizes. To deal with this problem, we used the max-pooling function and extracted the largest number of each vector c.

Softmax. After max-pooling, the resulting vector has a fixed length of 9 elements (number of filters). This resulting vector can then be introduced into a fully connected vector (softmax), to perform the classification task.

4 Experience and results

This work was done on a 6th generation hp omen i7 computer, 8GB RAM, hard disk: HDD 1000GB.

Our objective is to evaluate the performance of the convolutional neural network algorithm in scientific document classification systems. Experimental studies consist in comparing the performance of our proposed algorithm with other machine learning algorithms like as machine support vector (SVM), Naive Bayes, linear classification.

In this work, we used a database published in 2017 by "Web of Science Dataset" [25]. This database consists of $35,238$ scientific articles. Each element is defined by:

- X: an input data that includes text sequences.
- Y: the target value (label).
- Domain: we have seven domains: computer science, psychology, medical, civil, physics, biochemistry and computer vision.
- Keywords: defining each article.
- Summary of each scientific paper.

We used the “Keras” libraries to implement our work [26]. Keras is an open source library written in python and allowing to interact with deep neural network algorithms and machine learning. This library is characterized by its ability to process large databases.

We use the Accuracy Rate (Acc) to evaluate proposed method. It is defined as percentage of correct predictions. That is the case regardless of the number of classes. His formula is the following:

$$Acc = \frac{correctly\ predicted\ class}{total\ testing\ class} \times 100\ (\%)$$

We compared the accuracy of the algorithms with themselves according to the type of pre-treatment, and chose among these results the best accuracy and compare them with the results obtained by CNN. The table 1 below show the results obtained.

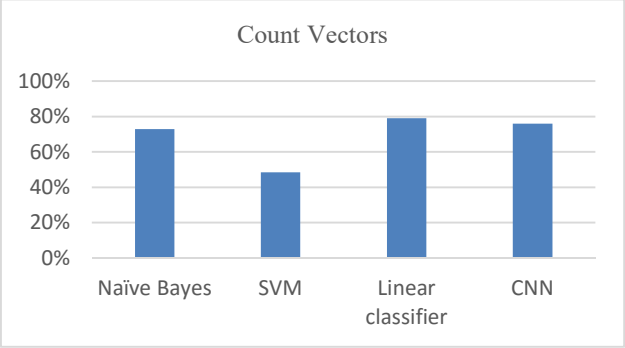

Figure 4. Classification accuracy using Count Vectors pre-treatment of each model

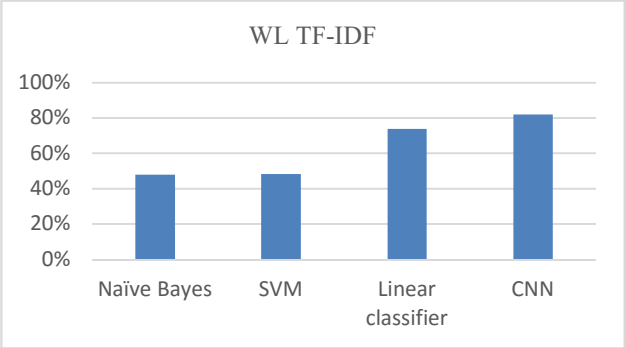

Figure 5. Classification accuracy using Word level TF-IDF vectors pre-treatment of each model

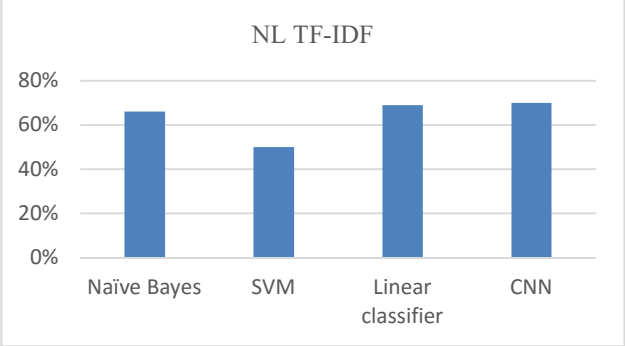

Figure 6. Classification accuracy using N-gram level TF-IDF vectors pre-treatment of each model

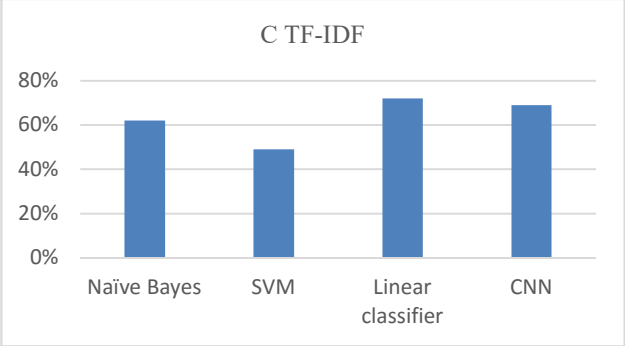

Figure 7. Classification accuracy using Character level TF-IDF vectors pre-treatment of each model

The Figure 4 shows that using the Count Vectors, as a pretreatment, gives a satisfactory result for all classifiers whose classification rate exceeds 73%, with the exception of the SVM classifier, which gave us a value below the average 48,4%. Therefore, Count Vectors pretreatment works very well with the Linear Classifier.

From Figure 5, almost the same result obtained using "Word level TF-IDF vectors" as pretreatment, except this time, in addition to the SVM classifier, the Naïve Bayes classifier with always below average flow rates. We also noticed an improvement with CNN; we had a rate of 82%.

Figure 6 and Figure 7 show that the use of "N-gram level TF-IDF vectors" or "Character level TF-IDF vectors" gives almost the same results, with the exception of the SVM classifier, which always has a lower than average value.

Table 1 summarizes the results obtained.

Table 1: Classification accuracy using different pre-treatment of each model

Algorithms	Type of pre-treatment			
	CV	WL TF-IDF	NL TF-IDF	C TF-IDF
Naïve Bayes	73%	48%	66%	62%
SVM	48.4%	48.44%	50%	49%
Linear classifier	79%	74%	69%	72%
CNN	76%	**82%**	70%	69%

With:
CV: Count Vectors
WL TT-IDF: Word level TF-IDF vectors
NL TF-IDF: N-gram level TF-IDF vectors
C TF-IDF: Character level TF-IDF vectors

Table 1 clearly shows that the classification rate of the algorithms varies according to the pretreatment used and the chosen architecture. Indeed, we obtained a rate close to 82% for the CNN algorithm with the chosen architecture, a rate of 50% for the SVM algorithm with NL TF-IDF preprocessing, a rate of 79% for the algorithm linear regression with CV pretreatment, and a 73% rate for the NB algorithm with CV pretreatment.

Conclusion

In this paper, we present a method using the convolutional neural network algorithm to classify scientific articles according to their domains from their abstracts, the process will base on several features extracted from their summaries. From the result obtained, we can conclude that the CNN algorithm with our chosen architecture is the best classifier of scientific articles, with a classification rate of 82% comparing it with other classical classification algorithm.

The work done shows the CNN's ability to solve the problem of article classification, which encourages us to start the recommendation systems by always using the CNN with the introduction of the other neuron network algorithms.

REFERENCES

[1] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," *ArXiv170802709 Cs*, Aug. 2017.

[2] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *J Mach Learn Res*, vol. 12, pp. 2493–2537, Nov. 2011.

[3] Y. Afoudi, M. Lazaar, and M. Al Achhab, "Collaborative Filtering Recommender System," in *Advanced Intelligent Systems for Sustainable Development (AI2SD'2018)*, Cham, 2019, pp. 332–345.

[4] S. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *50th Annual Meeting of the Association for Computational Linguistics, ACL 2012 - Proceedings of the Conference*, 2012, vol. 2, pp. 90–94.

[5] B. Pang and L. Lee, "Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2005, pp. 115–124.

[6] S. Jaillet, M. Teisseire, J. Chauché, and V. Prince, "Classification Automatique de Documents," in *INFORSID'03: INFormatique des Organisations et Systèmes d'Information et de Décision*, Nancy (France), 2003, pp. 87–102.

[7] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative Study of CNN and RNN for Natural Language Processing," *ArXiv170201923 Cs*, Feb. 2017.

[8] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1746–1751.

[9] Y. Zhang and B. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," *ArXiv151003820 Cs*, Oct. 2015.

[10] X. Zhang, J. Zhao, and Y. LeCun, "Character-level Convolutional Networks for Text Classification," *ArXiv150901626 Cs*, Sep. 2015.

[11] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent Convolutional Neural Networks for Text Classification," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, Austin, Texas, 2015, pp. 2267–2273.

[12] H. Zhang, "The Optimality of Naïve Bayes," in *In FLAIRS2004 conference*, 2004.

[13] H. Hilali, "Application de la classification textuelle pour l'extraction des règles d'association maximales," masters, Université du Québec à Trois-Rivières, Trois-Rivières, 2009.

[14] S. El Mrabti, M. Al Achhab, and M. Lazaar, "Comparison of Feature Selection Methods for Sentiment Analysis," in *Big Data, Cloud and Applications*, Cham, 2018, pp. 261–272.

[15] A. Labiad, "Sélection des mots clés basée sur la classification et l'extraction des règles d'association," masters, Université du Québec à Trois-Rivières, Trois-Rivières, 2017.

[16] O. Hicham, L. Mohamed, and T. Youness, "IMPROVED SELF-ORGANIZING MAPS BASED ON DISTANCE TRAVELLED BY NEURONS," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 5, 2018.

[17] H. Omara, M. Lazaar, and Y. Tabii, "Effect of Feature Selection on Gene Expression Datasets Classification Accuracy," *Int. J. Electr. Comput. Eng. IJECE*, vol. 8, no. 5, pp. 3194–3203, Oct. 2018.

[18] N. Usunier, M.-R. Amini, and P. Gallinari, "Résumé automatique de texte avec un algorithme d'ordonnancement," *Ingénierie Systèmes Inf.*, vol. 11, no. 2, pp. 71–91, Apr. 2006.

[19] S. Boukil, M. Biniz, F. E. Adnani, L. Cherrat, and A. E. E. Moutaouakkil, "Arabic Text Classification Using Deep Learning Technics," *Int. J. Grid Distrib. Comput.*, vol. 11, no. 9, pp. 103–114, Sep. 2018.

[20] S. Laroui, H. Omara, M. Lazaar, and O. Mahboub, "Comparative study of performing features applied in CNN architectures," presented at the Third International Conference on Computing and Wireless Communication Systems, ICCWCS 2019, April 24-25, 2019, Faculty of Sciences, Ibn Tofaïl University -Kénitra- Morocco, 2019.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, USA, 2012, pp. 1097–1105.

[22] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *Int. J. Comput. Appl.*, vol. 181, Jul. 2018.

[23] J. Sahoo, "Modified TF-IDF Term Weighting Strategies for Text Categorization," 2018.

[24] G. Gupta and S. Malhotra, "Text Document Tokenization for Word Frequency Count using Rapid miner," 2015.

[25] K. Kowsari, D. Brown, M. Heidarysafa, K. Jafari Meimandi, M. Gerber, and L. Barnes, "Web of Science Dataset," vol. 6, Mar. 2018.

[26] J. Moolayil, "Keras in Action: A Fast-Track Approach to Modern Deep Learning with Python," 2019, pp. 17–52.