International Journal of Electrical and Computer Engineering (IJECE)
Vol. 13, No. 2, April 2023, pp. 1827~1836
ISSN: 2088-8708, DOI: 10.11591/ijece.v13i2.pp1827-1836

Enhanced sentiment analysis based on improved word embeddings and XGboost

Amina Samih, Abderrahim Ghadi, Abdelhadi Fennan
Department of Computer Sciences, Data and Intelligent Systems Team, Faculty of Science and Technology, Abdelmalek Essaadi University, Tetouan, Morocco

Article Info

Article history:

Received May 13, 2022
Revised Sep 19, 2022
Accepted Oct 14, 2022

Keywords:

Machine learning
Sentiment analysis
Sentiment2Vec
Word2Vec
XGboost

ABSTRACT

Sentiment analysis is a well-known and rapidly expanding study topic in natural language processing (NLP) and text classification. This approach has evolved into a critical component of many applications, including politics, business, advertising, and marketing. Most current research focuses on obtaining sentiment features through lexical and syntactic analysis. Word embeddings explicitly express these characteristics. This article proposes a novel method, improved words vector for sentiments analysis (IWVS), using XGboost to improve the F1-score of sentiment classification. The proposed method constructed sentiment vectors by averaging the word embeddings (Sentiment2Vec). We also investigated the Polarized lexicon for classifying positive and negative sentiments. The sentiment vectors formed a feature space to which the examined sentiment text was mapped to. Those features were input into the chosen classifier (XGboost). We compared the F1-score of sentiment classification using our method via different machine learning models and sentiment datasets. We compare the quality of our proposition to that of baseline models, term frequency-inverse document frequency (TF-IDF) and Doc2vec, and the results show that IWVS performs better on the F1-measure for sentiment classification. At the same time, XGBoost with IWVS features was the best model in our evaluation.

Corresponding Author:

Amina Samih
Department of Computer Sciences, Computing Systems and Telecommunications Laboratory, Faculty of Science and Technology, Abdelmalek Essaadi University
Tangier, Morocco
Email: aminamonasamih@gmail.com

1. INTRODUCTION

Sentiment analysis [1] is a practical technique that allows companies, researchers, governments, politicians, and organizations to learn about people's emotions, happiness, sadness, anger, or a relatively neutral emotion, which play a significant part in decision making. Until now, sentiment analysis in the scientific literature has depended almost entirely on the bag-of-words method. Mozetič *et al.* [2] analyzed several sentiment categorization apps for Twitter data and discovered that practically all use it. Researchers depend on existing sentiment dictionaries [3] or construct customized and context-sensitive dictionaries [4]. The third set of investigations uses machine learning applications [5]. Some researchers have found that assessing sentiment at the level of articles or speeches yields good results [6]. However, the assumptions and simplifications inherent in the bag-of-words method, such as loss of grammatical structure or context-dependent word meanings, have been frequently highlighted [7]. Turney and Pantel [8] offer a review of previous vector space model development. Mikolov *et al.* [9] presented a more efficient architecture for constructing reusable word vector representations from big text corpora, which drew much attention to the

word embeddings technique. Word embedding is a deep learning method for creating vector representations of words and texts. Because of their ability to capture the syntactic and semantic links between words, these approaches have gained much attention in text and sentiment analysis. Word2Vec [10] is the most successful deep learning method for word embeddings.

Le and Mikolov [11] proposed the Word2Vec technique for obtaining word vectors by training a text corpus. The concept of Word2Vec (word embeddings) evolved from the distributed representation of words [12]. Several researchers have employed this strategy in their sentiment analysis study [13]–[15]. Alshari *et al.* [16] expanded word embeddings to sentence embeddings by averaging the word vectors in a sentiment review statement. Their results demonstrated that word embeddings outperformed the bag-of-words model in sentiment classification. In this work, the Word2Vec technique on sentiment analysis was explored; To make our first step in the sentiment analysis research field.

There are two types of sentiment categorization techniques: Polarized lexicon-based methods and machine learning methods like deep learning [17]. The polarized lexicon-based sentiment analysis approach is often based on lists of positive and negative meanings for words and phrases [18]. This method necessitates using a dictionary of words with negative and positive sentiment values. These approaches are easy to use, scalable, and fast to compute. As a result, they are utilized to handle generic sentiment analysis challenges [17]. However, lexicon-based approaches in human-labeled documents rely on human effort [18]. They also rely on locating the sentiment lexicon used to analyze the text [16]. Each word is assigned a polarity score in the polarized lexicon. This score, represented by an accurate value, reflects the degree (or intensity) of positivity or negativity according to a scale with integer or actual values. For example, according to the scale [-2,2], the real -2 represents absolute negativity, and the natural 2 represents absolute positivity. The numbers less than 0 designate different intensities of negativity: the farther the real is from 0 (and closer to 2), the stronger (or intense) the negativity.

Moreover, the reals greater than 0 represent different intensities of positivity: positivity is more vital when the reality is closer to 2. The accurate 0 means that the word is neither positive nor negative: it reflects a neutral polarity. Due to the increased accuracy of text classification, polarity-based approaches have lately been coupled with machine learning algorithms. Several authors reported machine learning approaches to be more accurate than polarity methods [19]. By merging lexicon-based and support vector machine (SVM) approaches, [20] improve sentiment analysis accuracy. For sentiment categorization of Twitter data, Zhang *et al.* [21] effectively integrated a lexicon-based technique with a binary classifier. Basari *et al.* [22] integrated the particle swarm optimization (PSO) approach for sentiment analysis of movie reviews with the SVM method. Machine learning approaches enhanced the accuracy of text classifications in all of these scenarios.

Word embeddings [23] approaches such as Word2Vec are continuous vector representations of word technologies that can turn words into meaningful vectors. Text categorization, grouping, and information retrieval can benefit from vector representations of words. The size of the text corpus impacts the pertinence of the Word2Vec. In other words, as the text corpus increases, the quality increases. Stojanovski *et al.* [24] used the Word2Vec method to learn word embeddings from 50 million tweets and then fed the obtained vectors into a deep learning network. Lauren *et al.* [25] have suggested a discriminant document embeddings method based on skip-gram for generating clinical text word embeddings. For word embeddings in the English and Chinese Wikipedia datasets, Fu *et al.* [26] used Word2Vec to analyze the sentiment approach by using word embeddings as inputs to a recursive autoencoder. Researchers prefer to employ word embeddings vectors as inputs to machine learning models because of the limits and restrictions in some corpora. As a result, improving the quality of word embeddings is critical and plays a crucial role in sentiment categorization algorithms. Kamkarhaghighi and Makrehchi [27] had a low F1-score; by using Word2Vec vectors in their deep learning model. That means that in some datasets, using Word2Vec reduced the F1-score of sentiment categorization. In addition, Fauzi *et al.* [28] suggested an approach to improve the F1-score of sentiment classification. Their technique was evaluated on two datasets, and the proposed algorithm reduced the F1-score on one of them.

This work aims to improve the F1-score of sentiment analysis; we proposed to generate vectorial representations of sentiments based on the combination of two approaches, Sentiment2Vec and Polarized Lexicon; and made classification via the XGboost algorithm. In the next section, we provide our suggested proposition, methodology, and the used classifiers to compare our method in detail. Experiment results show that the approach improves the F1-score of sentiment analysis. The organization of this paper is as: section 2 presents our proposed method and algorithm; section 3 reports our experiments, showing results along with evaluations and discussions; section 4 is the conclusion and future works.

2. METHOD

In our proposed method, improved word vector for sentiment analysis (IWVS) via XGboost, we have increased the F1-score of sentiment embeddings classification by combining natural language processing techniques, polarized lexicon, and Sentiment2Vec (based on Word2Vec). We made the classification using the XGboost model. The main architecture of the proposed approach is shown in Figure 1.

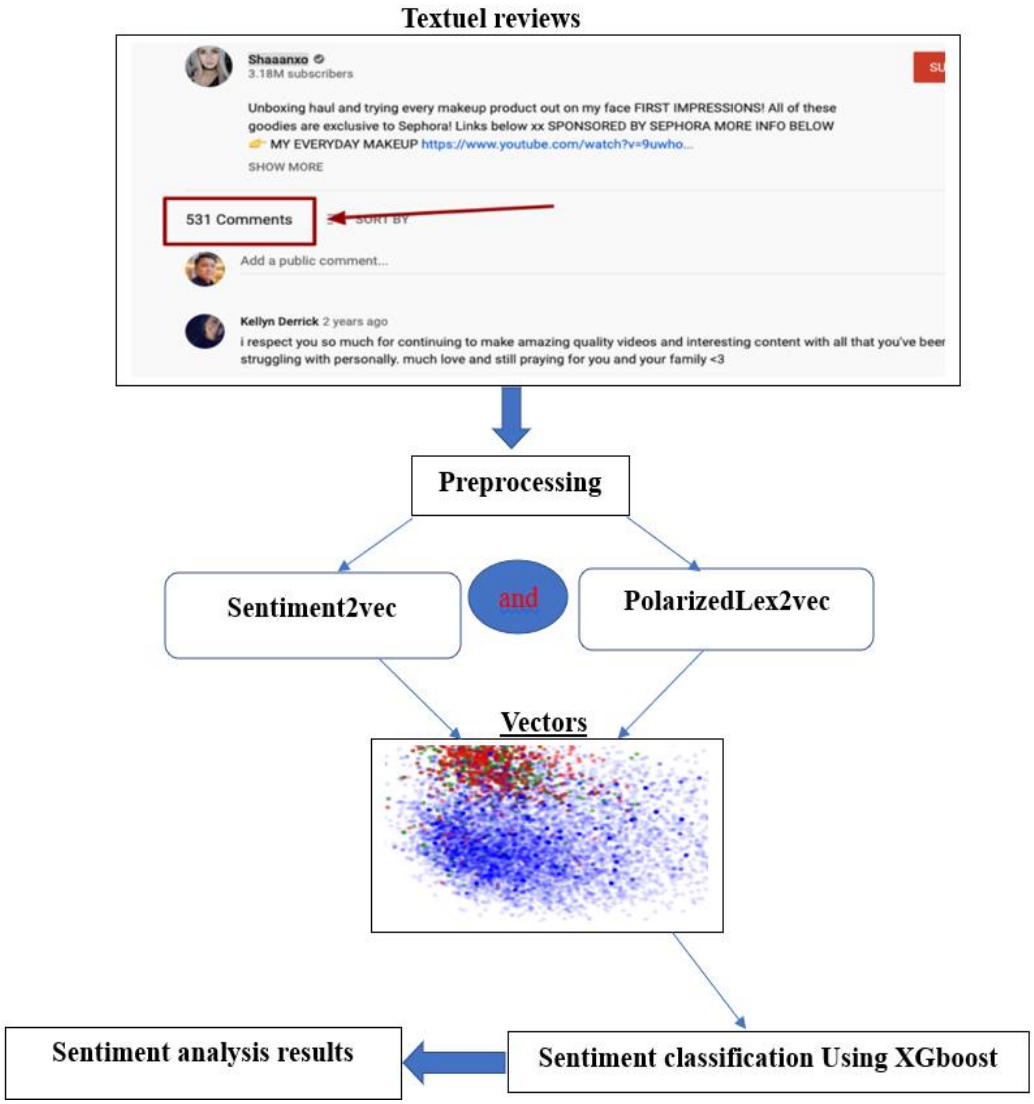

Figure 1. Overview of the proposed approach

2.1. Preprocessing

Preprocessing occurs before the start of the approach technique. Tokenization, case folding, and cleaning are some of the steps conducted during this stage. During tokenization, each review is broken down into smaller components known as tokens or words [29]. Case folding is the technique of converting all of the characters in a review text lowercase [29]. Meanwhile, non-alphabetic letters such as punctuation and numerals are utilized in cleaning. Because stemming and filtering have not been found to improve sentiment analysis performance in previous studies, they are not used in this study.

2.2. Sentiment2Vec

We built sentiment embeddings based on word embeddings in our work. To obtain sentiment embeddings, we averaged the vectors of the words in one sentiment (Sentiment2Vec). The critical task in this stage is to figure out the word embedding matrix W_w:

$$V_{Sentiment2vec}(\omega) = \frac{1}{n} \Sigma W_w^{xi} \quad (1)$$

where W_w $(w = w_1, w_2 \ldots w_n >)$ is the word embedding for word *xi*, which might be learned using the traditional Word2Vec technique [9], [30], [31].

Word2Vec is a popular way of creating word embeddings. It was first introduced by [9]. Skip-gram and CBOW are two Word2Vec versions proposed for learning word embeddings. The context words in the CBOW architecture are used to forecast the current word, whereas the skip-gram uses the current word to anticipate the surrounding words. Figure 2 depicts the two Word2Vec variations. Each of these architectures has three layers: an input layer, a hidden layer, and an output layer. The output layer is made up of neurons with SoftMax activation functions. In this work, we use the Skip-gram architecture.

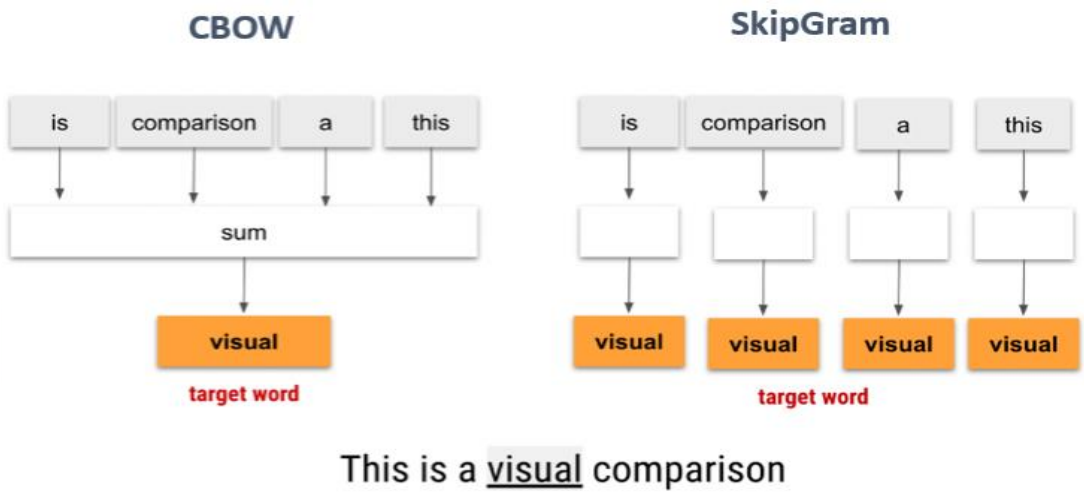

Figure 2. The architectures of Word2Vec

2.3. Polarizedlex2Vec

We define a sentiment as a judgment an individual makes about an object or subject; a polarity characterizes this judgment. For us, a polarity is either positive or negative. In our approach, sentiment is a particular type of opinion endowed with a polarity. Sentiment and emotion lexicons are collections of phrases and words with polarity scores that can be used to evaluate texts. Each lexicon has words and their values, which are sentiment scores for those terms.

2.4. Sentiment classification using XGboost

In the last stage, extreme gradient boosting [32] (XGboost) is used for sentiment embeddings classification. XGboost is a more complex variation of the gradient boosting approach. It includes tree learning algorithms as well as a linear model solver. It is speedy because of its ability to execute parallel processing on a single processor. It also has tools for cross-validation and detecting critical variables. Several parameters must be adjusted to optimize the model. The following are some of XGboost's [32] primary advantages: Regularization: aids in reducing overfitting.

- Parallel processing: XGboost uses parallel processing significantly faster.
- Missing values: It provides a built-in routine for dealing with missing values.
- Cross-validation built-in: allows the user to perform cross-validation at each iteration of the boosting process.

XGboost as shown in Figure 3 is an iterative multi-decision tree algorithm. Every tree learns from the residuals of the trees that came before it. XGboost's predicted output is the sum of all the results.

$$yi = \sum_{k=1}^{n} f\kappa(xi)\,, \quad f\kappa \epsilon F \quad (1)$$

where F denotes the space of trees, $f\kappa$ denotes a tree, so $f\kappa(xi)$ is the outcome of tree *k*, and yi is the predicted value of the i^{th} instance xi.

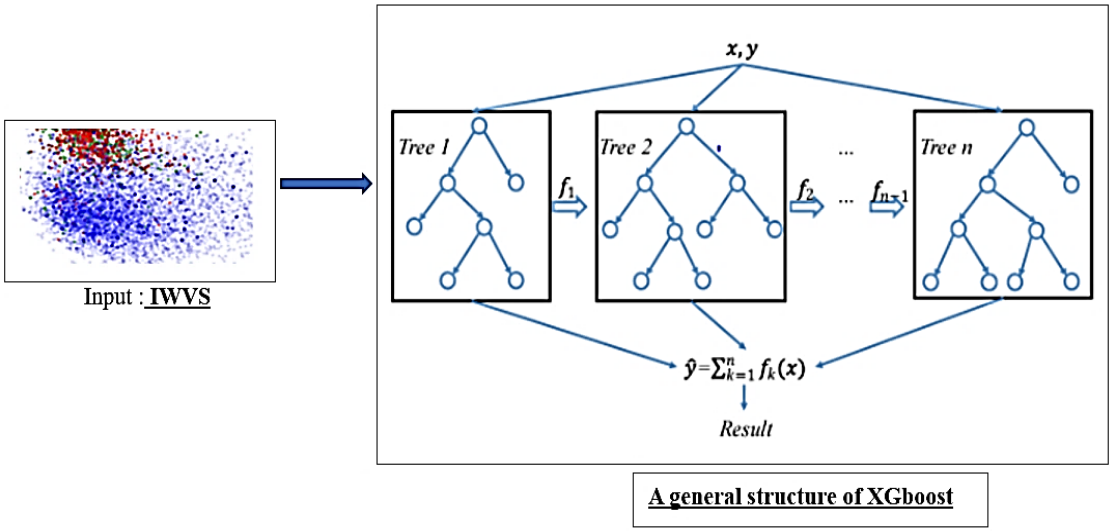

Figure 3. Classification of improved word vectors for sentiments (IWVS) via XGboost

2.5. Algorithm

The algorithm shows that Sentiment2Vec taken sentiment as input and returned vectors of words of the sentiment by using Word2Vec [21]. So, in the second step, each word vector of the input sentiment is extracted from Word2Vec [31], then the mean of vectors of words is calculated. In the third step, sentiment scores of each word are extracted from lexicon-based polarity (negative or positive).and we will normalize them. If a word does not exist in this couple (negative, positive), its score will be zero. The generated vectors from each step will be concatenated with other vectors from previous steps. We fit the XGboost [32] model on the generated vectors.

Algorithm 1. Improved words vector for sentiments analysis using XGboost

```
Inputs:
   S={W1, W2,……., Wn} ,  Input Sentiment S contains n words
   W2V=Word2Vec
   PolaLex2vec (P, N): Polarized
   Lexicon
   P: Positive, N: Negative
Output:
   IWVS: Improved Words vectors of sentiment S.
1.   for j=1 to m do
2.      VTj       GenerateVector (Tj)
3.       Tj        <Tj, VTj>
4.    end for
5.
6.    for each Wi in S do
7.          If Wi exist in W2V then extract VecWi
8.                MVi       VecWi
9.          endif
10.         for k=1 to h do
11.               If Wi in PolaLex2vec then
12.                  Sik        FindVector (Wi)
13.               end if
14.            ADD Sik into MVi
15.         end for
16.   ADD MVi into IWVS
17.   Return IWVS
18.   end for
19.   //fit the XGboost Classifier on IWVS
```

We chose the powerful machine learning technique XGBoost model to assess our generated vectors on well-known datasets. The model XGboost [32] ,as mentioned in section (3.4), is a gradient boosting library developed to be highly efficient, adaptable, and portable. It uses the Gradient Boosting framework to create machine learning algorithms. XGBoost offers parallel tree boosting to address numerous data science tasks quickly and accurately. Every tree learns from the residuals of all preceding trees; XGboost's predicted

output is the sum of all the results. We used two baseline models, Doc2vec and TF-IDF, for comparing our improved generated vectors. For classification, we proposed using four machine learning algorithms (XGboost, random forest, support vector machine, logistic regression) to evaluate our generated vectors on well-known datasets.

3. RESULTS AND DISCUSSION

Sentiment analysis is the process of determining the opinion, judgment, and emotion behind the natural language. For example, when people leave online reviews, comment on a brand or respond to market research, their assessments are necessarily colored by positive, negative, or neutral feelings. This section presents the datasets (from Kaggle [33]) and experimental assessments that we used to demonstrate the efficacy of our suggested strategy for sentiment analysis.

3.1. Datasets

Sentiment analysis is an analytical technique that consists of extracting meaning from many textual sources, such as reviews of e-commerce products, online movie reviews, or comments on social networks (Twitter). A score is then applied based on the sentiment expressed. For example, 0 for negative and 1 for positive. This transcription is done using natural language processing (or NLP for natural language processing), in this section, we present the reel datasets used for our approach evaluation:

- TtD [33]: Twitter tweets data is classified into positive and negative tweets.
- ArP[33]: Amazon reviews polarity dataset is created by classifying review scores 1 and 2 as negative and 4 and 5 as positive. Samples with a score of 3 are disregarded. Class 1 is the negative, and class 2 is the positive in the dataset. Each class comprises 1,800,000 training samples and 200,000 testing samples.
- IMDB [33]: IMDB dataset consists of 50,000 comments, with equal positive and negative comments. Comments are of different lengths and are given in sentences.

3.2. Evaluation metrics

Precision, recall, and F1-score are metrics mainly used to measure the performance of classification algorithms. In this study, we use F1 since this metric reflects both Recall and Precision. F1-score is being used as the evaluation metric. It is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. It is suitable for uneven class distribution problems. The crucial components of the F1-score are:

- True positives sentiments (TPS): These are the correctly predicted positive values which means that the value of the actual class of sentiments is yes, and the value of the predicted class is also yes.
- True negatives sentiments (TNS): These are the correctly predicted negative values, which mean that the actual class of sentiments value is no and the value of the predicted class of sentiments is also no.
- False positives sentiments (FPS): When the actual class of sentiments is no, the predicted class of sentiments is yes.
- False negatives sentiments (FNS): The actual class of sentiments is yes, but the predicted class of sentiments is no.

$$Precision = TPS/TPS + FPS;$$

$$Recall = TPS/TPS + FNS;$$

$$F1\ Score = 2(Recall\ *\ Precision)/(Recall + Precision)$$

3.3. Other word representations

We implemented two alternative baseline models in the literature for comparison: i) Doc2Vec is a Word2Vec extension that works on the entire document rather than individual words. Le and Mikolov developed this paradigm to provide a numerical representation of a document rather than a word representation [12]. Doc2Vec is based on the premise that the meaning of a word is influenced by the context in which it appears. The only difference between the Doc2vec and Word2Vec algorithms is the addition of a document ID. At this point, we treat each sentiment as a document and run the doc2vec algorithm; and ii) TF-IDF a frequency-based method because it considers the occurrence of a word not just in a single document (sentiment in our study) but across the entire corpus. TF-IDF penalizes common words by assigning them lower weights while emphasizing rare words in the entire corpus that appear in large numbers in a few documents (sentiments). The following are the component terms associated with TF-IDF.

- $TF =$ (Number of times word d appears in a document (sentiment))/(Number of words in the document (sentiment));
- $IDF = log(M/m)$, where M is the number of documents (sentiments) in which a word d appears, and m is the number of documents (sentiments) in which a word t appears; and
- $IDF - TF = IDF * TF$.

3.4. Results

We have compared our approach with two baseline models (Doc2vec and TF-IDF) via four machine learning classifiers, using various sentiment datasets with different features as shown in Figure 4. We used the scikit-learn library for implementing and training all the machine learning models in our research. All reports are based on the F1-score calculated overruns of models (XGboost, SVM, random forest, and logistic regression). We have predefined train and test sets for each dataset.

Our implementation starts by cleaning data. The text in the datasets used for evaluation is a highly unstructured form of data. Various types of noise are present in it. The data is not easily parsed without any preprocessing. For this, we applied a set of cleaning processes and text normalization, making it noise-free and ready for analysis. The Word2Vec model is trained on datasets to obtain vector representations for each sentiment's unique words. Then Sentiment2Vec is applied to create a vector for each sentiment by taking the average vectors (from Word2Vec) of the words present in the sentiment. The length of the resultant vector will be the same "200". We will repeat the same process for all textual sentiments available in mentioned datasets to obtain their vectors.

As already mentioned before, two sentiment lexicons based on polarity (positive sentiment and negative sentiment) were used to extract and generate the lexicon polarity vectors. The pre-modeling stages are required to get the data in the proper form and shape. We built models on the datasets with feature sets prepared-our created vectors (IWVS). The following algorithms are used to build models and to make a comparison with our chosen classifier, XGboost:

- Logistic regression (LR) [34]: is a collection of independent variables that predicts a binary outcome (1/0, Yes/No, True/False). Logistic regression can be thought of as a subset of linear regression in which the outcome variable is categorical, and the dependent variable is the log of probability. In other words, it forecasts the likelihood of an event occurring by fitting data to a logit function.
- Support vector machine (SVM) [35]: is commonly used to solve classification difficulties. In this algorithm, each data item is represented as a point in n-dimensional space (where n is the number of features), with the value of each feature being the value of a specific coordinate.
- Random forest (RF) [36]: is a multi-purpose machine learning technique that can solve regression and classification problems. It is an ensemble learning method in which several weak models merge to generate a robust model.

Let us review everything we have learned thus far. We cleaned our raw text data first and then learned about three different types of feature-sets extracted from TtD, ArP, and IMDB datasets: (Sentiment2Vec+PolarizedLex2vec), (Doc2vec), and (TF-IDF). We then used these feature sets to create sentiment analysis models (which can be retrieved from any text data). The Tables 1 to 3 show F1-scores for various models and feature sets (XG, SVM, LR, and RF).

As can be seen Tables 1 to 3, our IWVS with the XGboost model has the highest F1-score, while the RF has the lowest; The F1-score on classifiers used indicates that Doc2Vec vectors are not capturing the correct signals. In comparison, the best model for this problem was XGBoost with IWVS. On all classification algorithms, IWVS outperformed Doc2vec and TF-IDF; this clearly demonstrates the power of our word embeddings proposition in dealing with NLP problems.

Table 1. F1-score results using TtD dataset

Model	IWVS	TF-IDF	Doc2vec
LR	0.61	0.51	0.37
SVM	0.61	0.52	0.20
RF	0.51	0.50	0.07
XG	0.65	0.55	0.34

Table 2. F1-score results using ArP dataset

Model	IWVS	TF-IDF	Doc2vec
LR	0.59	0.52	0.38
SVM	0.61	0.53	0.21
RF	0.54	0.49	0.10
XG	0.69	0.56	0.38

Table 3. F1-score results using IMDB dataset

Model	IWVS	TF-IDF	Doc2vec
LR	0.59	0.50	0.36
SVM	0.55	0.51	0.22
RF	0.52	0.47	0.09
XG	0.62	0.55	0.41

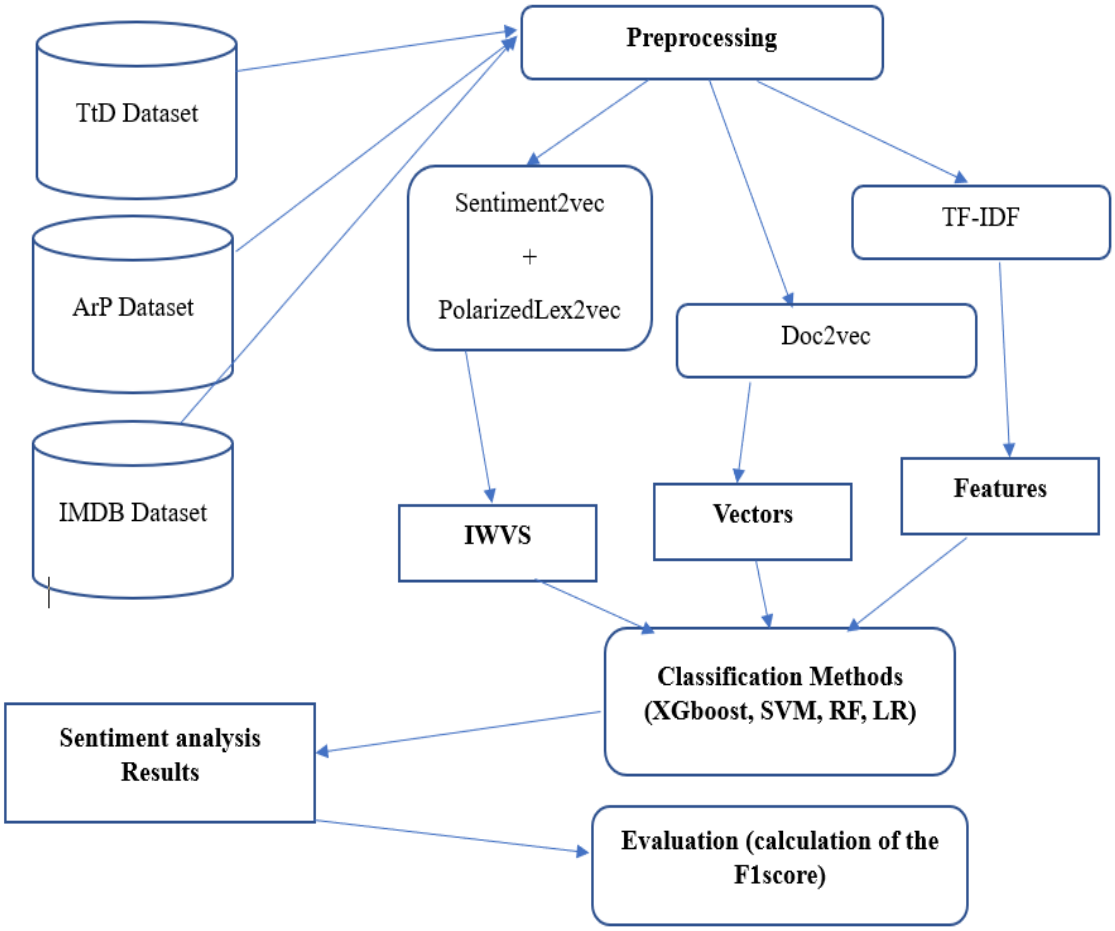

Figure 4. Process of the proposed approach's evolution

4. CONCLUSION

This paper proposed a new method to improve the F1-score of sentiment analysis based on word embeddings using the XGboost algorithm (IWVS). Our method has improved the F1-score of sentiment classification by combining the polarized lexicon approach and the Sentiment2Vec approach. To ensure the F1-score, we have used three sentiment analysis datasets and four machine learning classifiers; we investigated IWVS and features derived from TF-IDF and doc2vec (XG, SVM, LR, and RF); to ensure the F1-score of sentiment analysis. According to the trial results, IWVS outperformed doc2vec and TF-IDF; IWVS in the XGboost algorithm increased the F1-score of sentiment categorization tasks across all datasets.

Briefly, the main advantages of the proposed method are: i) According to previous research, because of the F1-score of trained vectors from Word2Vec, adding any vector to them decreased the F1-score. However, our proposed method (IWVS) has increased the F1-score of trained vectors in sentiment analysis for the first time; ii) The use of polarized lexicons in our research increased the F1-score of our proposed approach in all datasets; and iii) Any improvements in trained word embeddings (add tags embeddings, emojis embeddings). Will increase the F1-score in the future. As a result, our proposed method can be the basis for sentiment analysis techniques using machine learning algorithms. We plan in the future to enhance sentiments embeddings by introducing other information like emojis embeddings and tags embeddings using more sophisticated machine learning algorithms to improve the quality of evaluation.

REFERENCES

[1] A. Samih, A. Adadi, and M. Berrada, "Towards a knowledge based explainable recommender systems," in *Proceedings of the 4th International Conference on Big Data and Internet of Things*, Oct. 2019, pp. 1–5, doi: 10.1145/3372938.3372959.

[2] I. Mozetič, M. Grčar, and J. Smailović, "Multilingual Twitter sentiment classification: the role of human annotators," *PLOS ONE*, vol. 11, no. 5, Art. no. e0155036, May 2016, doi: 10.1371/journal.pone.0155036.

[3] J. Kleinnijenhuis, F. Schultz, D. Oegema, and W. van Atteveldt, "Financial news and market panics in the age of high-frequency sentiment trading algorithms," *Journalism*, vol. 14, no. 2, pp. 271–291, Feb. 2013, doi: 10.1177/1464884912468375.

[4] L. Aaldering and R. Vliegenthart, "Political leaders and the media. Can we measure political leadership images in newspapers using computer-assisted content analysis?," *Quality and Quantity*, vol. 50, no. 5, pp. 1871–1905, Sep. 2016, doi: 10.1007/s11135-015-0242-9.
[5] W. van Atteveldt, J. Kleinnijenhuis, N. Ruigrok, and S. Schlobach, "Good news or bad news? conducting sentiment analysis on dutch text to distinguish between positive and negative relations," *Journal of Information Technology and Politics*, vol. 5, no. 1, pp. 73–94, Jul. 2008, doi: 10.1080/19331680802154145.
[6] D. J. Hopkins and G. King, "A method of automated nonparametric content analysis for social science," *American Journal of Political Science*, vol. 54, no. 1, pp. 229–247, Jan. 2010, doi: 10.1111/j.1540-5907.2009.00428.x.
[7] J. Grimmer and B. M. Stewart, "Text as data: the promise and pitfalls of automatic content analysis methods for political texts," *Political Analysis*, vol. 21, no. 3, pp. 267–297, 2013, doi: 10.1093/pan/mps028.
[8] P. D. Turney and P. Pantel, "From frequency to meaning: vector space models of semantics," *Journal of Artificial Intelligence Research*, vol. 37, pp. 141–188, Feb. 2010, doi: 10.1613/jair.2934.
[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," Jan. 2013, [Online]. Available: http://arxiv.org/abs/1301.3781.
[10] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems*, vol. 26, 2013.
[11] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," in *31st International Conference on Machine Learning*, May 2014, vol. 4, pp. 2931–2939.
[12] A. Samih, A. Ghadi, and A. Fennan, "Hybrid movie recommender system based on word embeddings," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 147, 2023, pp. 454–463, doi: 10.1007/978-3-031-15191-0_43.
[13] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart, "Distributed representations," in *Parallel Distributed Processing: Explorations in The Microstructure of Cognition*, vol. 1, 1986, pp. 77–109.
[14] M. Al-Amin, M. S. Islam, and S. Das Uzzal, "Sentiment analysis of Bengali comments with Word2Vec and sentiment information of words," in *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Feb. 2017, pp. 186–190, doi: 10.1109/ECACE.2017.7912903.
[15] E. M. Alshari, A. Azman, S. Doraisamy, N. Mustapha, and M. Alkeshr, "Effective method for sentiment lexical dictionary enrichment based on Word2Vec for sentiment analysis," in *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, Mar. 2018, pp. 1–5, doi: 10.1109/INFRKM.2018.8464775.
[16] E. M. Alshari, A. Azman, S. Doraisamy, N. Mustapha, and M. Alkeshr, "Improvement of sentiment analysis based on clustering of Word2Vec features," in *28th International Workshop on Database and Expert Systems Applications (DEXA)*, Aug. 2017, pp. 123–126, doi: 10.1109/DEXA.2017.41.
[17] A. Sadeghian and A. R. Sharafat, "Bag of words meets bags of popcorn," *CS224N Proj*, pp. 4–9, 2015.
[18] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational Linguistics*, vol. 37, no. 2, pp. 267–307, Jun. 2011, doi: 10.1162/COLI_a_00049.
[19] K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: Tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, Nov. 2015, doi: 10.1016/j.knosys.2015.06.015.
[20] A. Mudinas, D. Zhang, and M. Levene, "Combining lexicon and learning based approaches for concept-level sentiment analysis," in *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining*, 2012, pp. 1–8, doi: 10.1145/2346676.2346681.
[21] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu, "Combining lexicon-based and learning-based methods for twitter sentiment analysis," *HP Laboratories*, HPL-2011-89, 2011.
[22] A. S. H. Basari, B. Hussin, I. G. P. Ananta, and J. Zeniarja, "Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization," *Procedia Engineering*, vol. 53, pp. 453–462, 2013, doi: 10.1016/j.proeng.2013.02.059.
[23] A. Samih, A. Ghadi, and A. Fennan, "Translational-randomwalk embeddings-based recommender systems: a pragmatic survey," in *Advanced Intelligent Systems for Sustainable Development*, Springer International Publishing, 2022, pp. 957–966.
[24] D. Stojanovski, G. Strezoski, G. Madjarov, and I. Dimitrovski, "Twitter sentiment analysis using deep convolutional neural network," in *Lecture Notes in Computer Science*, Springer International Publishing, 2015, pp. 726–737.
[25] P. Lauren, G. Qu, F. Zhang, and A. Lendasse, "Discriminant document embeddings with an extreme learning machine for classifying clinical narratives," *Neurocomputing*, vol. 277, pp. 129–138, Feb. 2018, doi: 10.1016/j.neucom.2017.01.117.
[26] X. Fu, W. Liu, Y. Xu, and L. Cui, "Combine HowNet lexicon to train phrase recursive autoencoder for sentence-level sentiment analysis," *Neurocomputing*, vol. 241, pp. 18–27, Jun. 2017, doi: 10.1016/j.neucom.2017.01.079.
[27] M. Kamkarhaghighi and M. Makrehchi, "Content tree word embedding for document representation," *Expert Systems with Applications*, vol. 90, pp. 241–249, Dec. 2017, doi: 10.1016/j.eswa.2017.08.021.
[28] M. A. Fauzi, D. C. Utomo, B. D. Setiawan, and E. S. Pramukantoro, "Automatic essay scoring system using N-Gram and cosine similarity for gamification based E-learning," in *Proceedings of the International Conference on Advances in Image Processing*, Aug. 2017, pp. 151–155, doi: 10.1145/3133264.3133303.
[29] E. S. Pramukantoro and M. A. Fauzi, "Comparative analysis of string similarity and corpus-based similarity for automatic essay scoring system on e-learning gamification," in *International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, Oct. 2016, pp. 149–155, doi: 10.1109/ICACSIS.2016.7872785.
[30] A. Samih, A. Ghadi, and A. Fennan, "Deep graph embeddings in recommender systems: a survey," *Journal of Theoretical and Applied Information Technology*, vol. 99, no. 15, pp. 3812–3823, 2021.
[31] A. Samih, A. Ghadi, and A. Fennan, "ExMrec2vec: explainable movie recommender system based on Word2Vec," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 8, 2021, doi: 10.14569/IJACSA.2021.0120876.
[32] T. Chen and T. He, "xgboost: extreme gradient boosting," *R package version 0.71-2*, 2018.
[33] "Kaggle." https://www.kaggle.com/datasets (accessed Jun. 25, 2022).
[34] T. Rymarczyk, E. Kozłowski, G. Kłosowski, and K. Niderla, "Logistic regression for machine learning in process tomography," *Sensors*, vol. 19, no. 15, Aug. 2019, doi: 10.3390/s19153400.
[35] V. Jakkula, "Tutorial on support vector machine (SVM)," *Northeastern University*. Accessed: Apr. 20, 2022. [Online]. Available: http://www.ccs.neu.edu/course/cs5100f11/resources/jakkula.pdf
[36] M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, Jan. 2005, doi: 10.1080/01431160412331269698.

BIOGRAPHIES OF AUTHORS

Amina Samih received the Degree in computer master from the National School of Applied Sciences of Fez in 2017; she is a Ph.D. student in degree in computer sciences in Faculty of Science and Technology, Tangier, Morocco. Her current research interests include recommender systems, big data, graph embeddings, and sentiment analysis. She can be contacted at email: aminamonasamih@gmail.com.

Abderrahim Ghadi Professor in Faculty of Science and Technology, Department of Informatics, Tangier, Morocco. Publication topics graph and complex networks, competitive intelligence systems, and computer network security. He can be contacted at email: ghadi05@gmail.com.

Abdelhadi Fennan Professor in Faculty of Science and Technology, Department of Informatics Tangier, Morocco. Topics, competitive intelligence, decision making, big data, web services, data mining, data visualization, graph theory. He can be contacted at email: afennan@gmail.com.