

Sentiment Analysis of Algerian Dialect Using Machine Learning and Deep Learning with Word2vec

Ahmed Cherif Mazari^{1,*} and Abdelhamid Djeflal²

E-mail: mazari.ac@gmail.com, abdelhamid.djeflal@gmail.com

¹ LSEA Laboratory, Mathematics and Computer Science Department, University of Médéa, Algeria

² LESIA Laboratory, Computer Science Department, University of Biskra, Algeria

Keywords: sentiment analysis, machine learning, deep learning, cnn, rnn, word2vec, opinion classification, Algerian dialect, Arabic dialect

Received: March 10, 2022

This paper deals with the issue of sentiment analysis on dialectal comments extracted from social media. These comments concern the Algerian spoken language, written in Arabic and/or Latin characters, which could be either Modern Standard Arabic, French or local dialect. This complexity gives rise to a large number of text processing issues. The contributions of this work are fourfold. First, the building of the Algerian dialect sentiment dataset of 11760 comments collected from diverse social media platforms. Second, the creation of the Skip-Gram and CBOW models by word2vec from a corpus containing 466424 comments, these latter are used to enhance the sentiment dataset by semantically similar words. Third, the proposal of a set of preprocessing steps adapted to deal with dialectal texts. Finally, implementation and testing of different machine learning classifiers (SVM, Naive Bayes via its three variants (Bernoulli NB, Gaussian NB and Multinomial NB)) and two deep learning architectures (CNN, RNN) to evaluate and compare the dataset in original version, in a transcribed to Latin character version and then in a semantically-enhanced version by word2vec models. Experiments reach performances of sentiment classifiers applied on "dataset transcribed to Latin characters" of accuracies = (MNB:84.21%, CNN:64.11%) and on "transcribed dataset and enhanced by word2vec models" of accuracies = (SVM:83.70%, RNN:65.21%).

Povzetek: Ta članek obravnava vprašanje analize sentimenta komentarjev alžirskega narečja, napisanih v arabščini in / ali latinici, pridobljenih iz družbenih medijev. Izvedbo izvajajo različne klasifikatorje strojnega učenja (SVM, Naive Bayes skozi njegove tri različice (Bernoulli, Gaussian in Multinomial)) in dve arhitekturi globokega učenja (CNN, RNN) za ovrednotenje in primerjavo nabora podatkov v prvotni različici, v različici, ki je prepisana v latinščini nato v semantično izboljšani različici modelov word2vec. Poskusi dosejajo izvedbe klasifikatorjev sentimentov, ki se uporabljajo za "nabor podatkov, prepisan v latinične znake" natančnosti = (MNB: 84,21%, CNN: 64,11%) in na "prepisan nabor podatkov in izboljšan z modeli word2vec" natančnosti = (SVM: 83,70%, RNN : 65,21%).

1 Introduction

As internet access grows and becomes more prevalent in the last two decades, an increasing number of people are sharing online their views through different ways and languages, for example, posts, comments, reviews and tweets through blogs or social networks. Sentiment analysis is an increasing task, the aim of which is to classify the opinions and views expressed by human users [1]. This diffusion is due to the fact that views and opinions are fundamental to the incentives and even affect the actions and beliefs of humans. The made choices are greatly influenced by how people view and evaluate the world. Thus, in order to make a decision, it is often necessary to seek out others' views. This is not only right for individuals but also right for organizations and political systems. For example, recently, opinionated postings have mobilized masses of people for political

changes, as have happened in some countries [2]. Thereby, it has become a requirement for opinions to be collected and deeply examined.

Since early 2010, Sentiment Analysis (SA) is widely studied in natural language processing, text mining, web mining, opinion mining and information retrieval. Existing studies have developed various approaches that involve both supervised and unsupervised methods for different sentiment analysis tasks. In the supervised setting, the first articles have applied all types of supervised machine learning (ML) algorithms, such as k-Nearest Neighbors (KNN), Naive Bayes (NB), Support Vector Machines (SVM), etc. Unsupervised models thus included various methods that exploit regular expressions, linguistic patterns and sentiment lexicons. Hence, several

* Corresponding author

survey books and papers covering those early techniques and applications have been published.

More recently, Deep Learning (DL) algorithms such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) have demonstrated very good results for text categorization, opinion mining and sentiment classification. The book of [3] presents a detailed survey on almost all recent deep learning models applied to analyze sentiment on document or sentence level. It concludes that the overwhelming majority of studies explored LSTM, CNN, GRU and attention mechanism, and their algorithms are vying for performance without any distinct outcome edge.

On the other side, the word embedding techniques are the applied methods of DL approaches that permit to capture successfully meaning of words from contextual texts. One of the most popular model to learn word embedding using DL is Word2vec [4]. Given enough data and contexts, Word2vec aims to group vectors of similar words together in vector space for detecting semantic similarities, by comparing these vectors using mathematical models. Thus, based on past appearances of words, the Word2vec model can detect precisely the meaning of target word. Hence, word2vec model and its implementations have recently garnered a great deal of attention in the ML and DL community.

Despite major progress has been made of techniques and methods for sentiment analysis, the field remains an active research area for many applications and languages worldwide. In particular, Arabic is the fifth most spoken language and has become the fourth most frequently used language on the Internet. With the growing research activity in sentiment analysis and opinion classification, several researchers have published studies and surveys to present progress in the area. Therefore, the fast pace of advances in DL and NLP needs continuous up-date of literature surveys [5].

Moreover, the complexity in orthography, morphology, grammar of Arabic and its dialects make sentiment analysis more challenging. Social media are taking a big margin of Arab world daily using Facebook, YouTube and Twitter. On which, the dialectal processing is more complicated task with writing and frequent switching between Arabic and Latin letters. Lately, researchers have made good attempts to resolve the problem of Arabic sentiment analysis. Although their works have more concentrated on Arabic dialects than on the Algerian dialect.

With this massive expansion of the use of social networks, many researchers have taken advantage to collect Algerian dialect's dataset. However, there is a significant lack in many aspects, including corpus processing and selection of the appropriate classification techniques. Hence, the contributions of this paper are fourfold. First, creating a new sentiment dataset from social media. The second contribution is the building a new corpus to create word2vec models which used to enhance the sentiment dataset. Third, proposing and evaluating new preprocessing steps to deal with related problems of dialectal language. Finally, four ML

algorithms, namely, three variants of Naïve Bayes model (Bernoulli Naïve Bayes (BNB), Multinomial Naïve Bayes (MNB) and Gaussian Naïve Bayes (GNB)) and SVM, and also, two well-known DL algorithms, namely, CNN and RNN are applied for sentiment classification to investigate obstacles and challenges of sentiment analysis related to the Algerian dialect (AlGD).

The remainder of the paper is organized as follows: Section 2 discusses related work on deep learning-based sentiment analysis, word embedding by word2vec model and challenges of sentiment analysis in Arabic and Algerian dialect. In section 3, a detailed description of the proposed approach for Arabic dialect sentiment analysis is presented. Section 4 is devoted to experimental tests of classifications by ML and DL algorithms. The conclusion is presented in section 5.

2 Related work

2.1 Deep learning-based sentiment analysis

Several ML algorithms that have been implemented and tested for sentiment analysis have shown good results. Such as [6], by using weblog corpus, the researchers compared the performance of the KNN and the SVM traditional ML algorithms for sentiment classification. They demonstrated that SVM provides a good result in; first, classification with a higher precision, second, a higher computational speed.

Thereafter, DL algorithms for sentiment analysis have become very popular. They provided automatic feature extraction and both richer representation capabilities and good achievement than traditional feature lexicon-based techniques. The basic idea of these techniques involves measuring sentiment from the semantic sense of words or sentences in the comment or the document. Several works tried to improve the performance of sentiment analysis by DL techniques like [7] who sought to improve the learning by integrating traditional surface methods based on manually feature extraction. To compare the output of proposed models with the DL baseline, the authors performed several experiments, where they indicated that the performance of these proposed methods exceeds the original baseline one. In [8], authors analyze the use of various neural networks for text classification task and demonstrate that the CNN can perform best at the level of words, and does not require knowledge of the syntactic or semantic structure of the language. On the other hand, a RNN can effectively classify the text for the sequential data representation. Authors in [9] implemented and tested traditional logistic regression and random forest algorithm to classify sentiment of tweets; the achieved results are around 80% of accuracy. [10] presented a three-way enhanced CNN model named 3W-CNN founded on the ensemble of DL algorithms and traditional ML algorithms based on the features. In [11], Researchers review DL algorithms that have been applied by researchers to different sentiment analysis tasks, and also offer the performance analysis of applied models on a specific dataset. [12] presented a DL model for aspect extraction

from opinions and sentiment scoring method of customers. Hence, they proposed to use seven-layer deep (CNN) to tag each aspect in the sentence. In the review paper of [13], the authors highlighted latest studies regarding the implementation of popular DL architectures such as CNN, RNN through its variant as LSTM and many more for solving different problems of sentiment analysis. Where they concluded that LSTM give better results compared to other DL algorithms, they also discussed the implementation of sentiment analysis on different languages such as English, Arabic, Chinese, Spanish, etc. [14] published a survey that examined SA techniques using several recent DL models. In [15], researchers proposed a Word2vec textual representation for the CNN model to analyze the sentiments of network public opinions, thus, the proposed method enhanced the accuracy within 3%-8% for emotion recognition. [16] presented a study, named *Muatheer*, helping to find influential users on social platforms by applying SA techniques, the experimentation of which has obtained very successful results.

2.2 Word2vec

In 2013 by [4], the vector representations of words, Word2vec, was published for the first time. As creators of word2vec model note, “there can be many different types of similarities”. This opens a question: what kinds of semantic similarities does Word2vec uncover? The answer is that Word2vec can generate ranked lists showing which words are closer and which words are further away in a semantic model. A growing number of works and researches have experimented Word2vec model or its derived methods, over 17943 researchers have cited Word2vec since its publication (*Google Scholar September 22, 2020*). The authors had introduced the Word2vec model for training distributed word embedding which allows word representation with similar meaning to be understood by algorithms of ML. This model produces a vector space of hundreds of dimensions, with each word of the vocabulary in the corpus being attributed a corresponding vector in the space. In which, words sharing common contexts are located near each other in this vector space. Word2vec model can be created using two algorithms based on neural networks; Continuous-Bag-Of-Words (CBOW) and Skip-Gram. The representation of the word is the vector of weights extracted from the last layer before the output of the neural network. Thus, the CBOW model takes the context of each word as the input in the neural network and tries to predict the word corresponding to the context by producing the representations. In contrast, the Skip-Gram model is applied to predict a context or a sentence given an input word in the neural network. Many works tried to apply the Word2vec in sentiment analysis field. Such as [17], the researchers proposed a technique to analyze sentiment based on feature set created by Word2vec. In [18], authors showed the challenges of applying word2vec to analyze twitter posts by using tweet length and weighted average for sentiment classification. [19] reviewed the applying of Word2vec and Doc2vec for sentiment analysis by using

SentiWordNet to establish clinical discharge summaries. A Word2vec sentiment analysis model was presented by the work [20] with additional semantic features to fit short Chinese texts on Weibo (China's Twitter-like service).

2.3 Arabic sentiment analysis challenges

Several sentiment analysis studies on English and some other languages are available, however Arabic language research has not been properly developed and that can be related to the lack of datasets. Although there has been a recent progress in the field of sentiment analysis, most of the resources are either restricted in size, not publicly accessible or domain-specific. In [21], Researchers address this problem by generating large multi-domain sentiment analysis datasets collected from various websites of annotated reviews for restaurants, hotels, movies and products, they consist a total of 33K documents. In addition, the researchers built multi-domain lexicons from the generated datasets. In [22], a scheme of sentiment classification was studied, which analyzes and identifies the polarity of sentiment in social media and reviews. In several architectures, they investigated the creation of a quality neural word embedding using 3.4 billion words of a corpus collected through the web. In the work of [23], efforts have been made to provide informative surveys of trends and challenges in sentiment analysis which focused on detecting and analyzing Arabic tweets. The study investigated the previous works to determine how the sentiment analysis was applied to a big volume of tweets. According to [24], furthermore the complexities of Arabic, the key problems of sentiment analysis include the representation of subjectivity. A trim distinction in a sentence being subjective or objective gets new challenges. Various challenges are highlighted; for example, if a negative text holds a part of words from the positive vocabulary, it is then generally explored as a positive opinion. When a word is part of a vocabulary that is both positive and negative, it poses new challenges in the analysis. For example, words like {(بغرف بارد)}, (cold, disgusting)} have in both negative and positive opinion. In [25], two models of deep RNN and SVM are implemented and tested on an Arabic dataset collected from reviews of Hotels along with lexical, syntactic, morphological, and semantic features. The authors demonstrated that the SVM exceeds the RNN algorithm in task efficiency (sentiment polarity identification, category identification and extraction of opinion target expression). [26] showed the results of using deep learning models to classify sentiments applied on dataset of Arabic tweets. The proposed models are based on pre-trained word vector representations without using feature extraction, in which they achieved promising results, reaching an accuracy of 64.30% and 64.75% for both CNN and LSTM respectively.

More recently, the study of [27] presented and tested an approach for automation building of the annotated sentiment corpus of messages written in Arabic and its dialects. They also summarized some sentiment analysis articles' result for Modern Standard Arabic (MSA). The work of [28] showed that the constraints of current

methods are mainly in feature selection and generation, preprocessing steps, and sentiment classification methods.

2.4 Sentiment analysis for Algerian dialect

The spoken language in Algeria, named Algerian Dialect (AlgD), is mainly derived from Arabic language, several words of French and some words from Turkish and Spanish, along with these foreign words are sometimes conjugated by Arabic grammar rules. AlgD is Also written by Arabic and/or Latin characters [29]. The work of [30] presented a particular study of Algerian, Tunisian, Egyptian dialect and standard Arabic through the phonological, morphological and orthographic variations. [31] proposed an opinion mining technique founded on a lexical base to deal with the specific features of AlgD used in social media. The Authors collected a hybrid corpus (in French and Algerian dialect written in French) of messages from social media and tested different approaches and methods, as well as specific techniques for AlgD preprocessing. Thus, they proposed a hybrid method for automatically annotating the corpus with the lexicon-based method and then applied ML algorithms for classification. Accuracy of the obtained results varies between 82% and 93%. In [32], the authors tackled the problem of AlgD sentiment analysis of Facebook comments published on pages belonging to various companies. They performed many steps. First, they studied the AlgD specificity and the linguistic behavior of Algerian Facebookers. Then they built a manually annotated corpus of 25475 comments and evaluated it using Naive Bayes classifier via divers preprocessing tasks, which reached an accuracy of 71.37% as best performance. The multilayer perceptron algorithm (MLP) and the CNN algorithm are tested to classify posts as negative, positive or neutral, reaching an accuracy of 81.6% 89.5% respectively, but unfortunately their results do not actually show the performance of the classification, as the dataset was not balanced between classes (Positive =75.82%, Negative=14.95% and Neutral=9.23%). Recently, [33] presented "TWIFIL" an open platform for public annotation of Twitter comments to create annotated corpus and dialectal lexicon. This platform offers a useful benchmark for sentiment analysis and opinion mining tools targeting Algerian dialect. As experimentation, the researchers create an AlgD annotated corpus of 9000 tweets, and then they tested set of DL models to classify a given tweet as negative, positive or neutral. LSTM and CNN model achieved the best accuracy of 75% and 76%, respectively.

More lately, the work of [34] shows the positive effectiveness of CNN and RNN to classify texts of opinions written in Algerian dialect retrieved from social media about Hirak_19, the experiment is carried out on a corpus consists of 7800 comments.

3 Proposed Approach

This section describes different phases followed to design the proposed approach, giving a general diagram and then explaining the details of each phase.

3.1 Global Architecture

This work involves the use of social networks for collecting texts to create sentiment dataset and word2vec corpus, preprocessing steps, sentiment analysis using various classification algorithms, and finally comparing obtained results of classifiers. The Figure 1 above presents the general diagram of this approach.

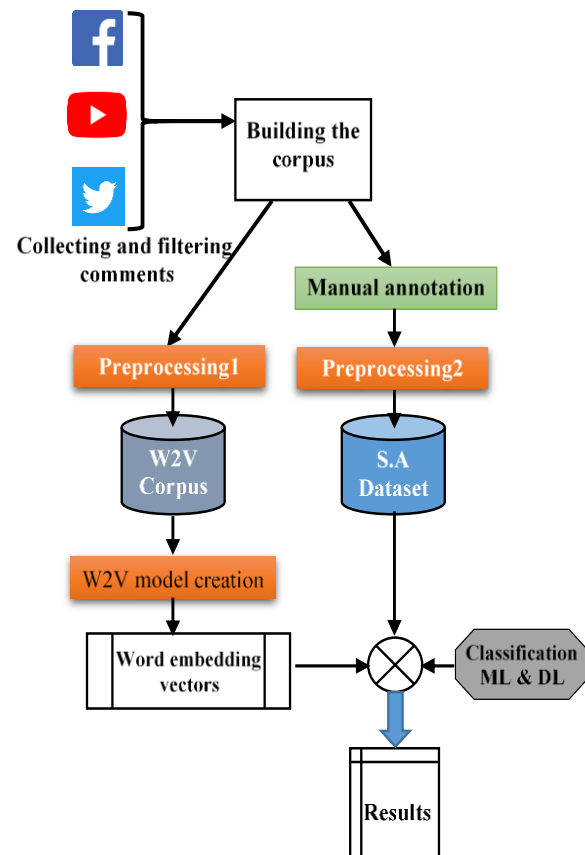


Figure 1: Design of the proposed approach.

3.2 Building the corpus

3.2.1 Collecting comments

It is the collecting of the corpus which may contain texts written in Arabic, French and/or in Algerian dialect transcribed in Arabic or/and Latin characters. This study based on the three more-known social media such as: Facebook, Twitter and YouTube. In which, the process is performed automatically via their platforms as follows:

- Facebook is used to allow for retrieving posts of a selected page and related comments to a specific post.
- On Twitter; by collecting messages exchanged (called Tweets) by users of discussed topics that grouped by Hashtag << # >>.
- YouTube is a video hosting platform; it is by retrieving comments posted relating to a topic of a given video.

3.2.2 Filtering comments

The filtering process is based on topics of FB-pages, Hashtags on twitter and titles of YouTube videos. Then it starts by deleting all the comments written in other languages. A language detection function is used for this purpose and for the Tweets that start with 'RT' must be deleted to eliminate redundancy of reTweets.

3.3 Manual annotation of comments

To create the sentiment dataset, the comment annotation procedure is performed manually in two classes:

- Positive: positive annotated comments are those that carry a positive overall opinion.
- Negative: negative annotated comments are those that carry a negative overall opinion.

3.4 Preprocessing steps

Two types of preprocessing are achieved, Preprocessing1, to prepare the corpus for vector embedding by word2vec models, Preprocessing2, to prepare the dataset for sentiment analysis.

- Preprocessing1 steps: Tokenization, Normalization and Removing stop words.
- Preprocessing2 steps: Tokenization, Normalization, removing stop words, Transcription of Arabic characters to Latin and Grouping.

3.4.1 Tokenization

Tokenization is the process of dividing text into a set of meaningful pieces. Social media texts may pose some challenges; thus, HTML tags, URLs, emoticons, # Hashtags and @user are considered.

3.4.2 Normalization

In this step, the number of unnecessary words of the vocabulary that do not provide any information about the polarity of texts is decreased, by removing the HTML tags, usernames which are detected by the <@> tag, the Hashtags detected by the <#> tag, and the URLs. Since this study is based on textual forms, therefore emoticons are removed, repeated letters more than twice in the same word are deleted by reducing them to a single letter, and all punctuation (.?!<>{}'"/-) and special characters are also removed. Furthermore, all letters are converted to lowercase.

Example 1:

- **AlgD** (*mzltto tamno ra7 diro ataghyr? mdrrrrrr jabtoli da7k http://uses.com*).
- **English** (*do you believe that you make changes? Lol you make me laugh http://uses.com*).
- **List of tokens**: {'mzltto', 'tamno', 'ra7', 'diro', 'ataghyr', '?', 'mdrrrrrr', 'jabtoli', 'da7k',

'http://uses.com'}.

- **List of normalised tokens**: {'mzltto', 'tamno', 'ra7', 'diro', 'ataghyr', 'mdr', 'jabtoli', 'da7k'}.

3.4.3 Removing stop words

In this step, stop words are removed. First, deletion all single letters as they are considered an empty word. Second, removing of empty words for French and Arabic (modern standard Arabic -MSA) by using predefined lists. For the Algerian dialect, a special list of stop words has been created related to this work, for example ['3lih', 'win', 'fi', 'هوما', 'مي', 'تنا', '...]

3.4.4 Transcription Arabic characters to Latin

Using this option to allow testing again the dataset in the same type of written characters, because some speakers write in Arabic and others in Latin. Therefore, all comments must be unified in the same Latin script.

Example 2:

- **AlgD** (*جاوني ضياف وانا يد تحط القهوة عين للتليفزيون حبيبت تشوف المظاهرات. صراحة نورمالمو الضيافة بيدلوهما الناس تولى في (وسط السمانة عندنا واجب وطني بالجمعة*).
- **English** (*Guests came to me, while I was holding the coffee, an eye on the television, and I wanted to see the demonstrations. Frankly, the visit is supposed to be changed by people to become in the middle of the week, we have a national duty on Friday.*).
- **Arabic to Latin script result**: (*jaawnee dayaaf wanaa yad taht alqahwa 'ayn lalteeleefeezyoun habeet nashouf almadhaahraat saraaha nawrmaalmou addayaafa yabdlouhaa annaas tawlee fee wast assamaana 'andnaa wajb wannee baaljm'a*).

This work uses the computational transcriber tool: Intellibe¹ (Arabic-to-Latin text transcriber), when it is vocalized, it provides the nearest corresponding phonetic tone.

3.4.5 Grouping

This step is interested on comments written in Algerian dialect. In fact, the problem of the AlgD is a language without orthographic rules, in addition, several orthographies can be found for the same word in the same text.

Example 3:

- **AlgD** (*nchalah, nchlh, nchallah, nchalh, nshallah, nshalah, incha2 alah*).
- **English** (*"on God's will"*).

To solve this problem, the authors suggest using, first the phonetic grouping, then the grouping by similarity. As shown in the study of [31] that was performed on the Algerian dialect.

a. Phonetic grouping: The authors use the 'Soundex' library [35] implementing an algorithm that returns a

¹ <http://www.intellaren.com/intellibe/doc>

phonetic code for a given word, as presented in Algorithm1. Soundex works by converting the input string to an output of four or more characters that can be compared to the calculated Soundex value for the other string as presented in Algorithm1 and Table 1.

Algorithm1:

Step1 - Save the first letter. Remove all occurrences of a, e, i, o, u, y, h, w.

Step2 - Replace all consonants (include the first letter) with digits.

Step3 - Replace all adjacent same digits with one digit.

Step4 - If the saved letter's digit is the same the resulting first digit, remove the digit (keep the letter).

Step5 - Append 3 zeros if result contains less than 3 digits. Remove all except first letter and 3 digits after it.

Table 1: Association code-letters of Soundex.

Code	Letters
1	B, F, P, V
2	C, G, J, K, Q, S, X, Z
3	D, T
4	L
5	M, N
6	R
SKIP	A, E, H, I, O, U, W, Y

b. Grouping by similarity: the authors propose to use a second grouping within each group of words in the phonetic dictionary to overcome the problem of many words that do not mean the same term are together grouped. This work uses the orthographic similarity grouping using the fuzzy-wuzzy library² which calculates the similarity between two words and provides a percentage of similarity, this library uses the Levenshtein distance [36]. This distance (Lev_distance) is equal to the minimum number of characters to delete, insert or replace to move from one string to another. The formula 1 is used to calculate the similarity degree between two words (w1 and w2).

$$\text{Similarity}(w1, w2) = 1 - \frac{\text{Lev_distance}(w1, w2)}{(\text{length}(w1) + \text{length}(w2))} \quad (1)$$

3.5 Word2vec model

3.5.1 Word2vec model creation

This step needs to create word-embedding vectors by word2vec DL algorithms in order to enhance the sentiment dataset, by adding similar words to the original comment words.

The word2vec model provides two kind of algorithms; the Continuous-Bag-Of-Words (CBOW) algorithm is to predict the target word from the words within the window of context (Figure 2.). The Skip-Gram algorithm is the opposite of CBOW and it is used to predict contextual words of the target word (Figure 2.). This work creates two models; CBOW and Skip-Gram by training the collected W2V corpus.

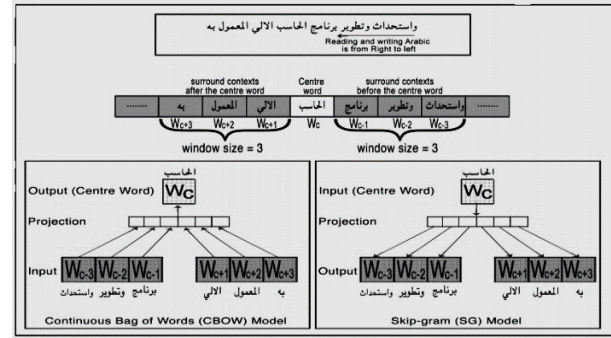


Figure 2: CBOW and Skip-Gram Word2vec model [5].

3.5.2 Dataset enhancement

This study uses these two created models (CBOW and Skip-Gram) to enhance the dataset by adding semantically similar words to original dataset texts.

- **CBOW:** the context is defined by several words for a given target word. Therefore, to enhance the dataset, each comment is input to the model as a context, and then any similar words returned are added to that original comment.
- **Skip-Gram:** the surrounding words returned by the model of each dataset word are added to the original comment. For example, if the word “car” is the target word then “drive” and “road” are its similar words.

4 Experimentation

The experimentation is achieved using Python programming language version 3.6 and its packages as mainly PANDAS to organize the DataFrame, NLTK for preprocessing (Tokenization, removing stop words,..), FuzzyWuzzy for string matching by measuring the similarity between words (based on Levenshtein Distance), Scikit-learn library³ for training and testing dataset by ML algorithms, and KERAS Api⁴ for DL algorithms.

4.1 Corpus and dataset

According to the proposed approach, first stage is to build the Word2vec model corpus then the sentiment dataset.

4.1.1 Word2vec model corpus

Comments and posts are collected from YouTube, Facebook and Twitter via their own APIs to build the given corpus, it consists 466424 texts (31.9Mo) for both Arabic and Latin characters. After preprocessing, the size of 25.4Mo is obtained.

4.1.2 Sentiment dataset

The dataset consists of comments about political topics; they are semi-automatically collected from Facebook platform and via, API-YouTube and API-Twitter. The

² <https://pypi.org/project/fuzzywuzzy/>

³ <https://scikit-learn.org/stable/>

⁴ <https://keras.io/>

annotation of this dataset is performed by three annotators (PhD students), where for each comment, the value '1' is attributed for positive review and the value '0' for negative review (at least two PhD students among the three reviewed it as positive/negative), along with neutral comments are eliminated. Hence, the created dataset is composed of 11760 comments⁵, which has 6111 positive comments and 5649 negative comments. Table2 presents the overall dataset statistics.

Table 2: Sentiment dataset.

Dataset comments				
Polarity	Facebook	YouTube	Twitter	Total
Positive	1300	4561	250	6111 (51.96%)
Negative	1700	3749	200	5649 (48.04%)
Total	3000 (25.51%)	8310 (70.66%)	450 (3.83%)	11760

4.1.3 Preprocessing

- The French and Arabic (MSA) stop word list is used for preprocessing after being extracted from the Python-NLTK package.
- For Latin character transcription: Furthermore the use of the transcriber tool Intellibe (Arabic-to-Latin text transcriber), the authors also developed a new transcription function that is specified for AlgD.
- The MSA's stemmer is also used, to remove certain prefixes for Arabic words, example removing 'ال', 'وال' ('the', 'and the').
- For grouping: First, the Soundex library provides sound lists of AlgD words or grouped words by their pronunciation.

Example 4:

{'N240': ['nqol', 'ngol', 'nchalah', 'nchallah', 'nshalah', 'nchalah', 'nqolo']}

Then, the similarity between two words is calculated using Formula 1 and the Levenshtein distance via fuzzywuzzy library.

Example 5:

The similarity of ('nchallah', 'nshalah') = 0.86 and the similarity of ('nchallah', 'ngol') = 0.50.

For each dictionary word list given in the 'phonetic grouping' step, the most common word is selected in that list and calculated its similarity with the rest elements. When the degree of similarity is above an indicated threshold equals to 70% (the value -70% is determined experimentally), the words are gathered together. This provides an initial set, which designates the first group. The same operation is applied for remaining words in the list until being emptied.

Example 6:

[['nchalah', ['nchallah', 'nshalah']], ['nqol', ['ngol', 'nqolo']]]].

4.2 Word2vec creation and enhancement of the dataset

Based on the collected corpus of 466424 comments, the word2vec algorithm is trained to create the two models: Skip-gram and CBOW.

For **Skip-Gram**. First, sentiment dataset words are chosen whose occurrences are equal to or greater than $K = 50$ (the value of $k = 50$ is determined experimentally), therefore, similar words are obtained by the Skip-Gram model. Second, those returned words are added to the original comments to enhance the sentiment dataset. The Gensim library⁶ is used to implement the Word2vec model.

Figure 3 shows examples of similar words to given words (named entities <مهدي, Mahdi> and <ربي, Rabi>) returned by Skip-Gram model.

```
Most similar to words :
مهدي ['غانسي', 0.7668360471725464], ('ينتميلين', 0.6495689749717712), ('احرار', 0.6144911050796509), ('بقرنها', 0.6156458258628845), ('أجدر', 0.61926960945129), ('منازع', 0.6005415916442871), ('وغانسي', 0.6045782566070557)

Most 3 similar words to rabi :
rabi [('yahafdek', 0.8156903386116028), ('ychadalkom', 0.8125947117805481), ('flmima', 0.8094107508659363)]
```

Figure 3: Examples of similar words by Skip-Gram model.

For **CBOW model**. Each comment is considered an input for the model, so this model returns words related to the context of that comment. Hence, these returned words are added to the original comment.

Example 7:

Original comment:

- **AlgD** (*ma3a masira wrabi ykoun m3ana n7a9o hadaf ta3na ya rabi naj7in.*).
- **English** (*with the march, and may God be with us to achieve our goal, successful on God's will.*).

Enhanced comment: Six (6) returned similar words by CBOW model are added to the original comment.

- **AlgD** (*ma3a masira wrabi ykoun m3ana n7a9o hadaf ta3na ya rabi naj7in wnanj7o matalibna nanj7o mana7absouch 7afdo silmiya*)

4.3 Tests of sentiment classifiers and results

The testing of each algorithm's sentiment classification is carried out, first on the original dataset in two versions, raw texts and transcribed texts to Latin characters, then on the dataset enhanced by word2vec models.

⁵ <https://sourceforge.net/projects/alg-dialect-sentiment-dataset/files/>

⁶ <https://pypi.org/project/gensim/>

4.3.1 Test 1: Sentiment classification of the original dataset

a. Machine Learning classification

The authors opted to use the Naïve-Bayes and SVM (Support Vector Machine) classifier, because they are based on the technique of Bag-Of-Words and the word frequency. In addition, these classifiers are widely applied in text classification and sentiment analysis. Concerning the Naïve Bayes model, the authors applied three variants of algorithms; Bernoulli Naïve Bayes (BNB), Multinomial Naïve Bayes (MNB), and Gaussian Naïve Bayes (GNB).

The dataset is divided in two parts, test and training (test part represents 10% and 90% of train part), the accuracy metric of cross validation is applied to measure the performance. (“The accuracy is the ratio of the number of correct predictions to the total number of input samples”).

b. Deep Learning classification

The classification is performed by the CNN and RNN algorithms as deep learning classifiers. Thus, the dataset is divided into 10% of test part and 90% of train part for both CNN and RNN. Both of them have a batch size = 128 and epochs =10.

- **CNN:** Several models have been tested, varying the numbers of layers, the numbers of neurons on each layer and the activation functions. The Rectified Linear Unit (ReLU) activation function yielded the best result for five hidden layers; each Conv1D layer (one dimension convolution layer) has 100 feature maps. In addition, sigmoid is the activation function on the output layer to make zero (0 Negative) or one (1 Positive) predictions.
- **RNN:** The parameters have been tested and chosen as follows; two LSTM layers, both of them contains 128 memory units (called smart neurons), two Dropout layers of 0.3 (0.3 is the best value to decrease the overfitting determined experimentally), and four dense layers with (ReLU) as an activation function, along with sigmoid, as the activation function on the output layer.

The following Table 3 and Figure 4 present the obtained results of classification valued by accuracy metric.

Table 3: Numerical results of test1.

Classifier	(Accuracy) Original dataset	(Accuracy) Transcribed dataset	Percentage of improvement
BNB	67.10	79.76	18.87%
MNB	67.15	84.21	25.50%
GNB	22.70	77.40	240.97%
SVM	68.07	83.02	21.96%
CNN	62.96	64.11	1.83%
RNN	62.11	61.60	-0.82%

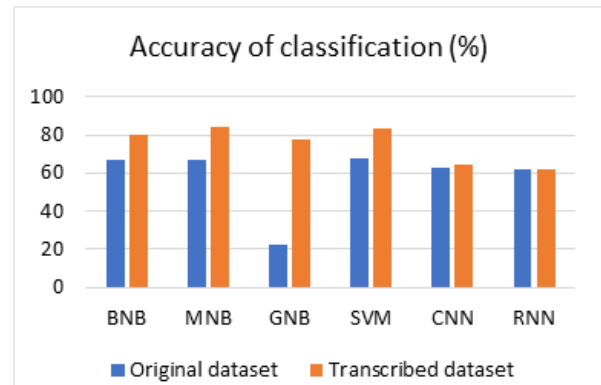


Figure 4: Graphical results of test1.

Discussion1

The MNB algorithm is noticed which is the most adapted compared to the other Naïve Byes models; it provides the best result with an accuracy of 67.15% for original dataset and 84.21% for transcribed dataset.

The SVM’s accuracy is also almost similar to MNB in both datasets (68.07% in the original dataset and 83.02% in the transcribed dataset).

As conclusion, the MNB and SVM algorithms are the best sentiment classifiers for traditional machine learning models.

The GNB algorithm is bad classifier in original dataset.

Concerning deep learning, the CNN and the RNN classifiers almost give the same performance, with CNN getting a small improvement over RNN in both datasets.

Finally, data transcription is noticed to have significantly improved the accuracy for each traditional classifier, and this due to the dataset word unification, by transcribing letters into Latin characters.

4.3.2 Test 2: Sentiment classification of the dataset enhanced by word2vec

The same algorithms for ML and DL is used with the same parameters as the test1.

By this experimentation, different sentiment classifiers are tested on the enhanced dataset; by Skip-Gram and CBOW of word2vec models, and then the accuracy is calculated according the window of the context N (N =3, 6, 9: number of similar words).

The following Table 4 presents the obtained results of classification applied on the enhanced dataset.

Table 4: Numerical results of test2.

Classifier	Original dataset						Transcribed dataset					
	Skip-Gram			CBOW			Skip-Gram			CBOW		
	N=3	N=6	N=9	N=3	N=6	N=9	N=3	N=6	N=9	N=3	N=6	N=9
BNB	69.30	65.39	58.79	67.65	67.88	68.34	81.55	81.34	82.50	83.21	81.38	81.30
MNB	69.17	69.90	70.83	67.38	67.81	67.76	82.41	81.55	81.90	83.28	82.14	81.00
GNB	33.22	38.72	41.49	28.81	29.47	31.23	78.72	78.20	79.10	78.03	78.23	77.74
SVM	69.99	70.66	72.04	68.30	68.52	68.96	82.92	82.51	82.32	83.70	83.21	83.11
CNN	62.25	53.91	55.71	60.97	62.64	61.48	62.51	64.05	64.26	62.72	63.15	64.56
RNN	59.43	61.48	59.69	62.25	60.97	60.71	61.74	62.90	60.92	65.21	61.71	64.18

Discussion2

The best result is also achieved in the transcribed dataset for both Skip-Gram and CBOW model.

Regarding the original dataset, every classifier achieved almost the same results with different number (N) of the similar added words. As conclusion, the CBOW and Skip-Gram model have the same effect for the classification in both datasets.

Machine Learning:

The best result is obtained by SVM in both dataset, (Original Dataset, Skip-gram, 72.04%, N=9) and (Transcribed dataset, CBOW, 83.70%, N=3).

Deep Learning:

The best result for CNN in transcribed dataset CBOW (64.56%, N=9).

The best result for RNN in transcribed dataset CBOW (65.21%, N=3).

The following Table 5 and Table 6 present the percentage change between Skip-Gram and CBOW for the original dataset and the transcribed dataset.

By comparing the CBOW and Skip-Gram model, the following points are noticed:

For original dataset

Since GNB is a bad classifier, its results should be ignored. The other traditional algorithms applied on enhanced dataset by Skip-Gram give better results compared to CBOW, Unlike the CNN algorithm which provides the best results, when applied on a dataset enhanced by CBOW, it increases the results towards (10% -16% in N = 6,9).

For transcribed dataset

Comparing CBOW and Skib-Gram, Table 6 shows that all models almost give the same results except RNN.

As conclusion

In original dataset: Skip-Gram is better for traditional algorithms than CBOW. CBOW is better for DL Algorithms.

In transcribed dataset: Enhancing of dataset by Skip-gram or CBOW has no effect except for RNN. In which the RNN increase results with CBOW for both datasets (original and transcribed) around 4-6%.

The results obtained are summarized in Table 7 and Figure 5 below.

Discussion3

Finally, the conducted experiments summarize the following conclusion points:

1. This study is different from previous work. First, the dataset of work is transcribed by mixing characters (Arabic and Latin) and texts of dataset are retrieved and built from several platforms unlike other works that have used either Facebook or Twitter. Moreover, this study has experimented various algorithms, classic models through machine learning and recent models through deep learning.
2. Dataset transcription has significantly increased the performance of all classification algorithms (except for RNN).
3. SVM achieves the best results of sentiment classification. Thus, this supports the assessment obtained in previous study of [25] where they

Table 5: Improvement in original dataset.

Classifier	Percentage change between CBOW to Skip-Gram in original dataset		
	N=3	N=6	N=9
BNB	-2.38%	3.81%	16.24%
MNB	-2.59%	-2.99%	-4.33%
GNB	-13.28%	-23.89%	-24.73%
SVM	-2.41%	-3.03%	-4.28%
CNN	-2.06%	16.19%	10.36%
RNN	4.75%	-0.83%	1.71%

Table 7: Improvement in transcribed dataset.

Classifier	Percentage change between CBOW to Skip-Gram in transcribed dataset		
	N=3	N=6	N=9
BNB	2.04%	0.05%	-1.45%
MNB	1.06%	0.72%	-1.10%
GNB	-0.88%	0.04%	-1.72%
SVM	0.94%	0.85%	0.96%
CNN	0.34%	-1.41%	0.47%
RNN	5.62%	-1.89%	5.35%

Table 6: Final numerical results.

Classifier	Original dataset							Transcribed dataset						
	Without W2V	Enhanced by Skip-Gram			Enhanced by CBOW			Without W2V	Enhanced by Skip-Gram			Enhanced by CBOW		
		N=3	N=6	N=9	N=3	N=6	N=9		N=3	N=6	N=9	N=3	N=6	N=9
BNB	67.10	69.30	65.39	58.79	67.65	67.88	68.34	79.76	81.55	81.34	82.50	83.21	81.38	81.30
MNB	67.15	69.17	69.90	70.83	67.38	67.81	67.76	84.21	82.41	81.55	81.90	83.28	82.14	81.00
GNB	22.70	33.22	38.72	41.49	28.81	29.47	31.23	77.40	78.72	78.20	79.10	78.03	78.23	77.74
SVM	68.07	69.99	70.66	72.04	68.30	68.52	68.96	83.02	82.92	82.51	82.32	83.70	83.21	83.11
CNN	62.96	62.25	53.91	55.71	60.97	62.64	61.48	64.11	62.51	64.05	64.26	62.72	63.15	64.56
RNN	62.11	59.43	61.48	59.69	62.25	60.97	60.71	61.60	61.74	62.90	60.92	65.21	61.71	64.18

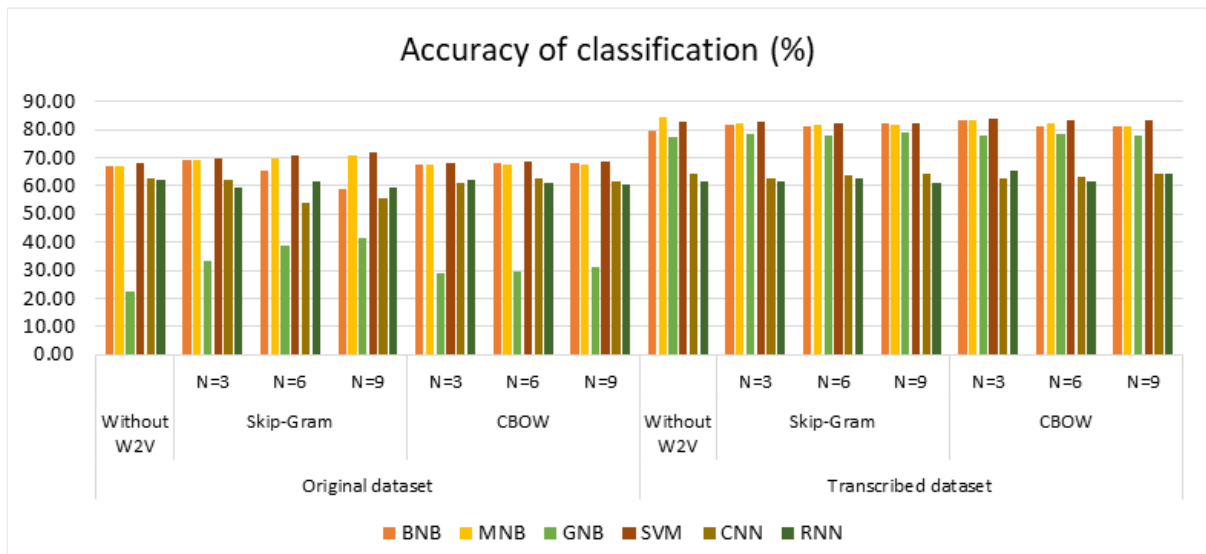


Figure 5: Final graphical results.

- conclude that “SVM has outperformed RNN in Arabic Hotels’ Review”.
4. RNN increases results with CBOW for both datasets (around 4-6%).
 5. CNN is better than RNN in both original and transcribed dataset without word2vec but the RNN is better in dataset enhanced by word2vec through CBOW and Skip-Gram model.
 6. Accuracy of the RNN classifier is increased on dataset enhanced by (CBOW, N=3) (61.60% to 65.21%) (6%).
 7. Number N of words in the context for either CBOW or Skip-Gram does not really affect the accuracy of classification.
 8. Adding directly similar words extracted by word2vec to the original texts in the dataset usually does not improve classification results.
 9. This work achieved better results about CNN and RNN compared to [34] due to the quality and size of the dataset.
 10. Table 8 below summarizes the best algorithm of each dataset.

Table 8: Best Algorithm vs Dataset type.

Model	Original dataset	Original dataset + word2vec	Transcribed dataset	Transcribed dataset + word2vec
Machine Learning	SVM:68.07%	SVM:72.04%	MNB:84.21%	SVM:83.70%
Deep Learning	CNN:62.96%	CNN:62.64%	CNN:64.11%	RNN:65.21%

5 Conclusion

In this paper, the authors presented an experimental study to deal with AlgD sentiment analysis through machine learning and deep learning classifiers. First, they created a sentiment dataset of 11760 comments⁷ (6111 positive/ 5649 negative comments) collected from Facebook, YouTube and Twitter. In addition, they have gathered 466424 comments from social media to build a corpus in order to involve as many words as possible to create word2vec models, for which to exploit the benefit of word embedding of these models in order to retrieve similar words and then to enhance the sentiment dataset.

Several preprocessing steps were implemented, besides grouping the same words that are written differently in the corpus. Thus, the dataset is transcribed to Latin characters and its results have been compared to the original one for each algorithm.

Different ML classifiers (SVM, Naive Bayes with its three variants (Bernoulli, Gaussian and Multinomial)) and two DL architectures (CNN, RNN) were implemented and tested. As result, an improved accuracy is reached for sentiment classification using "Transcribed dataset" (MNB:84.21% and CNN:64.11%) and "Transcribed dataset enhanced by word2vec" (SVM:83.70% and RNN:65.21%) compared to "Original dataset" (SVM: 68.07%, CNN: 62.96%).

The conducted experiments have proved that appropriate preprocessing tasks and transcription of dialectal texts improve results of sentiment classifications. Furthermore, traditional machine learning models as SVM algorithm that provides better performance probably due of bi-classification (Positive, Negative) in average size of dataset. Unlike the deep learning algorithms, which they need a very large size of data.

⁷ <https://sourceforge.net/projects/alg-dialect-sentiment-dataset/files/>

The Word2vec models have been used to enhance semantically the dataset by blindly adding similar words to comments, but the experimental results have shown that this technique did not really improve the performance of different classification algorithms. Another limitation is that a comment has two opinions, which leads to believe that it has decreased the accuracy.

As a future work, the authors plan to study the way to analyze one opinion from comments that have two opinions. Moreover, they plan to study embedding building using Doc2vec, Sec2vec, Elmo or Bert algorithms, and performing more evaluations to investigate which embedding are more suitable for dialectal sentiment classification tasks.

References

- [1] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge university press, 2020.
<https://doi.org/10.1017/9781108639286>
- [2] L. Zhang, S. Wang, and B. Liu, “Deep Learning for Sentiment Analysis: A Survey,” *Lang. Linguist. Compass*, vol. 10, no. 12, pp. 701–719, Jan. 2018.
<https://doi.org/10.1002/widm.1253>
- [3] B. Agarwal, R. Nayak, N. Mittal, and S. Patnaik, *Deep Learning-Based Approaches for Sentiment Analysis*. Springer, 2020.
https://doi.org/10.1007/978-981-15-1216-2_6
- [4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, pp. 1–12, 2013.
<https://doi.org/10.48550/arXiv.1301.3781>
- [5] A. M. Alayba, V. Palade, M. England, and R. Iqbal, “Improving Sentiment Analysis in Arabic Using Word Representation,” *2nd IEEE Int. Work. Arab. Deriv. Scr. Anal. Recognition, ASAR 2018*, pp. 13–18, 2018.
<https://doi.org/10.1109/asar.2018.8480191>
- [6] C. Alfaro, J. Cano-Montero, J. Gómez, J. M. Moguerza, and F. Ortega, “A multi-stage method for content classification and opinion mining on weblog comments,” *Ann. Oper. Res.*, vol. 236, no. 1, pp. 197–213, 2016.
<https://doi.org/10.1007/s10479-013-1449-6>
- [7] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. A. Iglesias, “Enhancing deep learning sentiment analysis with ensemble techniques in social applications,” *Expert Syst. Appl.*, vol. 77, pp. 236–246, 2017.
<https://doi.org/10.1016/j.eswa.2017.02.002>
- [8] M. Amjad, I. Voronkov, A. Saenko, and A. Gelbukh, “Comparison of text classification methods using deep learning neural networks,” in *Proceedings of the 20th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*, 2019.
- [9] H. M. PANDEY, P. Tiwari, A. Khamparia, and S. Kumar, “Twitter-based opinion mining for flight service utilizing machine learning,” *Inform.*, vol. 43, no. 3, pp. 381–386, 2019.
<https://doi.org/10.31449/inf.v43i3.2615>
- [10] Y. Zhang, Z. Zhang, D. Miao, and J. Wang, “Three-way enhanced convolutional neural networks for sentence-level sentiment classification,” *Inf. Sci. (Ny)*, vol. 477, pp. 55–64, 2019.
<https://doi.org/10.1016/j.ins.2018.10.030>
- [11] O. Habimana, Y. Li, R. Li, X. Gu, and G. Yu, “Sentiment analysis using deep learning approaches: an overview,” *Sci. China Inf. Sci.*, vol. 63, no. 1, p. 111102, 2019.
<https://doi.org/10.1007/s11432-018-9941-6>
- [12] P. Ray and A. Chakrabarti, “A mixed approach of deep learning method and rule-based method to improve aspect level sentiment analysis,” *Appl. Comput. Informatics*, 2020.
<https://doi.org/10.1016/j.aci.2019.02.002>
- [13] A. Yadav and D. K. Vishwakarma, “Sentiment analysis using deep learning architectures: a review,” *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4335–4385, 2020.
<https://doi.org/10.1007/s10462-019-09794-5>
- [14] W. Etaïwi, D. Suleiman, and A. Awajan, “Deep Learning Based Techniques for Sentiment Analysis: A Survey,” *Informatica*, vol. 45, no. 7, 2021.
<https://doi.org/10.31449/inf.v45i7.3674>
- [15] J. Li, Y. Wang, and J. Wang, “An Analysis of Emotional Tendency Under the Network Public Opinion: Deep Learning,” *Informatica*, vol. 45, no. 1, 2021.
<https://doi.org/10.31449/inf.v45i1.3402>
- [16] S. Al-Otaibi *et al.*, “Finding Influential Users in Social Networking using Sentiment Analysis,” *Informatica*, vol. 46, no. 5, 2022.
<https://doi.org/10.31449/inf.v46i5.3829>
- [17] E. M. Alshari, A. Azman, S. Doraisamy, N. Mustapha, and M. Alkeshr, “Improvement of Sentiment Analysis Based on Clustering of Word2Vec Features,” in *2017 28th International Workshop on Database and Expert Systems Applications (DEXA)*, 2017, pp. 123–126.
<https://doi.org/10.1109/dexa.2017.41>
- [18] K. A. Djaballah, K. Boukhalfa, and O. Boussaid, “Sentiment analysis of Twitter messages using word2vec by weighted average,” *Sixth IEEE Int. Conf. on Social Networks Analysis, Management and Security, SNAMS 2019*, pp. 223–228, 2019.
<https://doi.org/10.1109/snams.2019.8931827>
- [19] Q. Chen and M. Sokolova, “Word2Vec and Doc2Vec in Unsupervised Sentiment Analysis of Clinical

- Discharge Summaries.,” *CoRR*, vol. 1805.00352. 2018.
<https://doi.org/10.48550/arXiv.1805.00352>
- [20] B. Shi, J. Zhao, and K. Xu, “A Word2vec Model for Sentiment Analysis of Weibo,” in *2019 16th International Conference on Service Systems and Service Management (ICSSSM)*, 2019, pp. 1–6.
<https://doi.org/10.1109/icsssm.2019.8887652>
- [21] H. ElSahar and S. R. El-Beltagy, “Building large arabic multi-domain resources for sentiment analysis,” in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2015, pp. 23–34.
https://doi.org/10.1007/978-3-319-18117-2_2
- [22] A. Dahou, S. Xiong, J. Zhou, M. H. Haddoud, and P. Duan, “Word embeddings and convolutional neural network for arabic sentiment classification,” in *Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers*, 2016, pp. 2418–2427.
<https://aclanthology.org/C16-1228>
- [23] M. Abdullah and M. Hadzikadic, “Sentiment analysis on arabic tweets: Challenges to dissecting the language,” in *International Conference on Social Computing and Social Media*, 2017, pp. 191–202.
https://doi.org/10.1007/978-3-319-58562-8_15
- [24] S. Siddiqui, A. A. Monem, and K. Shaalan, “Evaluation and enrichment of Arabic sentiment analysis,” in *Intelligent Natural Language Processing: Trends and Applications*, Springer, 2018, pp. 17–34.
https://doi.org/10.1007/978-3-319-67056-0_2
- [25] M. Al-Smadi, O. Qawasmeh, M. Al-Ayyoub, Y. Jararweh, and B. Gupta, “Deep recurrent neural network vs. support vector machine for aspect-based sentiment analysis of Arabic hotels’ reviews,” *J. Comput. Sci.*, vol. 27, pp. 386–393, 2018.
<https://doi.org/10.1016/j.jocs.2017.11.006>
- [26] M. Heikal, M. Torki, and N. El-Makky, “Sentiment analysis of Arabic Tweets using deep learning,” *Procedia Comput. Sci.*, vol. 142, pp. 114–122, 2018.
<https://doi.org/10.1016/j.procs.2018.10.466>
- [27] I. Guellil, F. Azouaou, and F. Chiclana, “ArAutoSenti: automatic annotation and new tendencies for sentiment classification of Arabic messages,” *Soc. Netw. Anal. Min.*, vol. 10, no. 1, p. 75, 2020.
<https://doi.org/10.1007/s13278-020-00688-x>
- [28] A. Ghallab, A. Mohsen, and Y. Ali, “Arabic Sentiment Analysis: A Systematic Literature Review,” *Appl. Comput. Intell. Soft Comput.*, vol. 2020, 2020.
<https://doi.org/10.1155/2020/7403128>
- [29] K. Meftouh, N. Bouchemal, and K. Smaili, “A study of a non-resourced language: an Algerian dialect,” in *Spoken Language Technologies for Under-Resourced Languages*, 2012.
- [30] H. Saadane and N. Habash, “A Conventional Orthography for Algerian Arabic,” in *Proceedings of the Second Workshop on Arabic Natural Language Processing (ANLP)*, 2015, pp. 69–79.
<https://doi.org/10.18653/v1/w15-3208>
- [31] M. Bettiche, M. Z. Mouffok, and C. Zakaria, “Opinion Mining in Social Networks for Algerian Dialect,” in *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2018, pp. 629–641.
https://doi.org/10.1007/978-3-319-91479-4_52
- [32] A. Soumeur, M. Mokdadi, A. Guessoum, and A. Daoud, “Sentiment analysis of users on social networks: overcoming the challenge of the loose usages of the Algerian Dialect,” *Procedia Comput. Sci.*, vol. 142, pp. 26–37, 2018.
<https://doi.org/10.1016/j.procs.2018.10.458>
- [33] L. Moudjari, K. Akli-Astouati, and F. Benamara, “An Algerian Corpus and an Annotation Platform for Opinion and Emotion Analysis,” in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 1202–1210.
<https://aclanthology.org/2020.lrec-1.151>
- [34] A. C. Mazari and A. Djeflal, “Deep Learning-Based Sentiment Analysis of Algerian Dialect during Hirak 2019,” in *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH)*, 2021, pp. 233–236.
<https://doi.org/10.1109/ihsh51661.2021.9378753>
- [35] D. Holmes and M. C. McCabe, “Improving precision and recall for soundex retrieval,” in *Proceedings. International Conference on Information Technology: Coding and Computing*, 2002, pp. 22–26.
<https://doi.org/10.1109/itcc.2002.1000354>
- [36] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” in *Soviet physics doklady*, 1966, vol. 10, no. 8, pp. 707–710.