International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Jul-24)

http://dx.doi.org/10.12785/ijcds/160135

Sentiment Analysis from Texts Written in Standard Arabic and Moroccan Dialect Based on Deep Learning Approaches

Abdellah Ait elouli [1], El mehdi Cherrat [2], Hassan Ouahi [1] and Abdellatif Bekkar[3]

[1]*Department of Computer Sciences, Laboratory of Innovation in Mathematics and Intelligent Systems Faculty of Applied Sciences, Ibn Zohr University, Ait Melloul, Morocco*
[2]*Department of Computer Sciences, Laboratory of Systems Engineering and Information Technology National School of Applied Sciences, Ibn Zohr University, ENSA, Agadir, Morocco*
[3]*Department of Computer Sciences, LIM Laboratory Faculty of Sciences and Technics Hassan II University of Casablanca, Mohammedia, Morocco*

Received 19 Oct. 2023, Revised 19 Apr. 2024, Accepted 21 Apr. 2024, Published 1 Jul. 2024

Abstract: Sentiment analysis plays a crucial role in extracting subjective information from various sources using natural language processing techniques. It involves identifying opinions, attitudes, and emotions towards specific topics or documents. This study focuses on evaluating the performance of machine learning, deep learning, and transfer learning algorithms in accurately classifying positive and negative sentiments in Arabic comments. The study uses different machine learning and deep learning techniques, including the use of Arabert. A transfer learning technique based on the BERT algorithm for Arabic language processing. Arabert is pre-trained on a vast corpus of Arabic data, which allows him to capture complex Arabic-specific linguistic patterns. It is then refined onto a smaller dataset of Arabic comments for sentiment analysis. The study will outline the important steps and processes involved in each approach, highlighting their strengths, and comparing their performance. The utilization of deep learning and transfer learning techniques, such as Arabert, has the potential to enhance sentiment analysis accuracy on Arabic comments. By comparing the performance of different methods, the study aims to identify the most effective approaches for sentiment analysis in Arabic text. The findings of this research have practical implications in improving sentiment analysis accuracy for Arabic language applications, particularly when working with limited labeled datasets. The results can be valuable in fields like market research, customer service, and social media analysis, providing insights into the attitudes, opinions, and emotions expressed by Arabic-speaking users.

Keywords: Natural Language Processing, Sentiment analysis, machine learning, supervise learning, preprocessing.

1. INTRODUCTION

Arabic, one of the important Semitic languages, is spoken by more than 500 million people and is the administrative and official language of over 21 countries [1]. Its word morphology is complex compared to English, and its various dialects add to the language's richness and complexity. Modern Standard Arabic (MSA) is used for formal writing, while dialectical Arabic (DA) is used for informal daily communication. Additionally, Arabizi, a form of Arabic language used on social media, is becoming increasingly popular. Despite its widespread use, research in the field of modern computational linguistics has been limited for the Arabic language, with named entity recognition (NER) and sentiment analysis (SA) being the most challenging tasks in Arabic Natural Language Processing (ANLP) [2], [3].

The Arabic language poses several challenges for natural language processing (NLP), such as morphological complexity, script complexity, ambiguity, diacritics, and dialects. Arabic words can take on many different forms due to the rich system of inflection and derivation, making it challenging to identify the root form of a word [4]. Arabic script is written from right to left, which can be challenging for NLP algorithms designed for left-to-right scripts [5]. Arabic words can have multiple meanings depending on the context in which they are used, and diacritical marks can change the meaning of a word, further complicating the task [6]. Furthermore, each dialect can have its own unique vocabulary, grammar, and pronunciation, making it difficult to develop NLP algorithms that work well across all dialects [7].

However, researchers have made significant progress in recent years, including the development of pre-trained models like ARABERT and other language models [8]. These advancements are beginning to address the complexities of Arabic NLP, signifying a burgeoning area of research with substantial future potential.

E-mail address: abdellah.aitelouli@edu.uiz.ac.ma, e.cherrat@uiz.ac.ma, h.ouahi@uiz.ac.ma

Our objective is to employ a range of machine learning algorithms, including deep learning techniques, on Arabic text comments written in Arabic script. Through this exploration, we aim to underscore the unique advantages of each algorithm. Within this study, we delineate and elucidate the processes and critical stages of each approach we evaluate. Our goal is to identify the methodology that best accomplishes the task of classifying comments accurately into their respective positive and negative categories.

The rest of this article is organized as follows: in the next section we will discuss the related work, we then describe the proposed methods, namely that based on machine learning, deep learning, and transfer learning. The results of the experiments and the evaluation measures are given in sections 5 and 6. A conclusion and some perspectives of this work are presented in the last section.

2. RELATED WORK

Sentiment analysis in Moroccan Arabic, also known as Darija, is an active area of research within the natural language processing field, with numerous challenges presented by the unique features of this dialect. Recent studies look to expand sentiment analysis to cover the breadth of Moroccan linguistic diversity, tackling both Arabic and Latin scripted dialects [9]. Methodologies include different algorithms, such as Naive Bayes and Random Forests, for classifying sentiment in social media content, with one study showing the utility of multiple classifiers in analyzing Twitter comments [10]. In the context of Arabic social media text processing, there are specific challenges due to the diglossia nature of the language, use of roman script, and code-switching [11]. Ensemble learning techniques have been proposed to enhance sentiment analysis accuracy in Moroccan Arabic, exploiting the strength of different feature selection methods and classifiers to improve performance [12].

The research conducted by N. Habbat, H. Anoun, and L. [13] builds upon previous work in sentiment analysis and topic modeling, with a focus on Arabic social media during the COVID-19 pandemic. Their study specifically examines Moroccan Arabic tweets, revealing insights into public sentiment and prevalent topics during this crisis. By achieving high accuracies, particularly with logistic regression classifiers reaching up to 68.80%, their work demonstrates the effectiveness of adapted ensemble approaches for sentiment analysis in Arabic social media. This research underscores the importance of understanding linguistic nuances in dialectal Arabic for accurate sentiment analysis and broader applications like public opinion analysis.

Abdul-Mageed and Kubler [14] developed SAMAR, a hybrid system that combines lexicon-based and machine learning techniques for the sentiment analysis of Arabic social media content. The system showcased promising results in evaluating a dataset composed of diverse Arabic social media texts. They further described the system's architecture, dataset characteristics, and the evaluation methodology which included a comparison with other prominent systems dedicated to Arabic sentiment analysis.

Errami et al. [10] addressed the sentiment analysis of Moroccan dialect using machine learning techniques and social media content. They curated a dataset of Moroccan dialect tweets and applied a range of preprocessing steps before employing various machine learning models, including SVM, NB, and RF to classify the tweets across sentiment classes. The SVM model emerged superior with an impressive accuracy of 80%.

A study by Duwairi and Qarqaz [15] extended this work to a more general set of Arabic social media comments. Exploring different feature representation techniques, they achieved the best precision with an SVM classifier, confirming the efficiency of machine learning approaches in recognizing complex patterns in Arabic sentiment analysis.

3. DATA AND METHODOLOGY

A. *Machine Learning Approach*

Conducting sentiment analysis on Facebook comments written in Modern Standard Arabic (MSA) or Dialectal Arabic (DA) using a machine learning approach involves a series of essential steps. The process is depicted in Figure 1, and in the subsequent sections, we will delve into the primary tasks associated with each step.

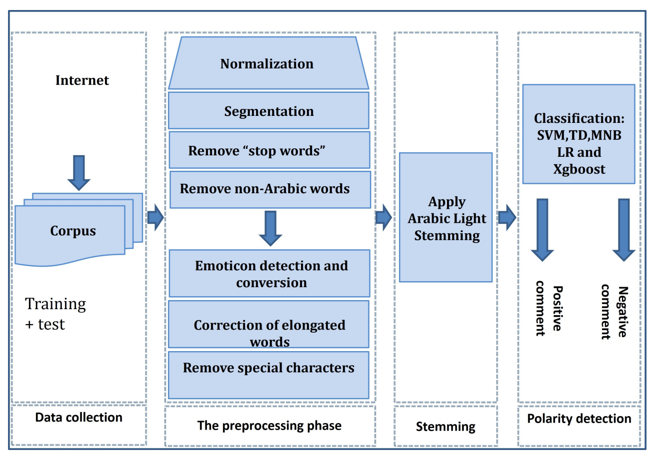


Figure 1. Steps in the proposed process for sentiment analysis

1) *Phase 1: Data collection*

The proposed machine learning process is currently undergoing testing using Facebook comments [16] related to Morocco's parliamentary elections on October 7, 2016. The data was obtained from the Internet and comprises a total of 10,254 comments. Among these, 3,673 comments have been annotated as positive, while 6,581 comments have been annotated as negative (see Figure 2). The annotation of the collected comments was done using crowdsourcing, where a group of professional volunteers manually determined the sentiment polarity of each comment (as shown in Figure 3).

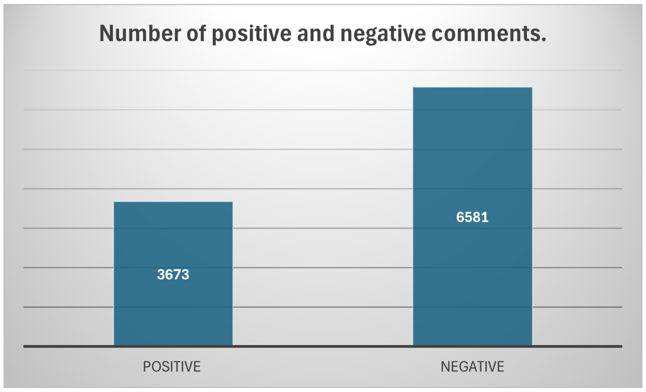


Figure 2. Graph representing the number of positive and negative comments.

To evaluate the model's performance, the dataset is divided into two parts: the training data and the test data. Initially, the model is trained using the training dataset, and once it has learned from this data, its effectiveness is tested using the separate test dataset.

Input variables are automatically extracted from the corpus formed from preprocessed comments using n-gram extraction and TF/TF-IDF weighting schemes. And therefore, we precede a supervised learning sentiment analysis

index	sentence	sentiment
33	واش لكيبيع البيض والنعناع ممزيانش ولكن التعليم ديلكم لناقص من تأثير الحكومات السابقة	P
34	هدرة خاوية بنكيران زمطكم وفضحكم وقال للشعب ها الأعداء وها الشفارا وهذا إنجاز عظيم أما الوضع المغربي لن يغيره بنكيران وحده	P
35	سير شوف رجال الدولة ديال بصح	N
36	كاملين بيوكوم عنستفكوم من فوق شواية الا بنكيران هو الفاخر	N
37	أخويا خودها و ماتمشيش تصوت ههيه، راه تاحنا خاصنا نتعاملو مع السياسيين بالسياسة ، يوليو هما لي كايتشكاو من الشعب ماشي حنا	N
38	جاوبني علي هاد السؤال علاش ما اقتطعش من البرلمانيين و اقتطع غير من الموضفين و ولي تزم و دوي علي راسك	N
39	أي مغربي أو مغربية في السعودية تعنف وتظلم وتعدب وتقهر وتحقر دولة ظالمة لا للظلم للمغاربة الموجودين هناك لماذا يستحقرون المغاربة	N
40	ساعة الخير هناك في الجزائر اخوانا و اخوتنا و احبابنا لي عندنا مدة ما شفناهمش	P
41	تحياتي واحترامي لكل المغاربة الأحرار والشرفاء ربي معاكم	P
42	العدالة و التنمية بغيتو و لا كرهتوا	P

Figure 3. Extract from the database

2) Phase 2: Pre-processing

Text preprocessing plays a pivotal role in the sentiment analysis research process. Its primary objective is to extract variables in the form of words or word sequences for classification purposes. The effectiveness of this preprocessing significantly impacts the performance of classification models and the final outcomes of the analysis.

This initial step starts with text cleaning and standardization, involving the removal of punctuation, symbols, repeated letters, stopwords, or words that do not contribute pertinent information to the subject under investigation. Following this, the text undergoes tokenization, which involves breaking it down into individual lexical units or tokens. In the context of modern or dialectal Arabic text, these units are often more intricate as they frequently consist of multiple words. Therefore, stemming becomes a crucial task. In our approach, we drew inspiration from the "light10 stemmer" proposed by Larkey et al. in 2007 [17], which involves the elimination of common prefixes and suffixes found in tokens within modern Arabic language text. The following table (I) shows an example of preprocessing a comment:

TABLE I. EXAMPLE OF PRE-PROCESSING OF A COMMENT

Treatment	Result
Initial Text	الله يرحم من قراك. راك فسرتي لينا دستور جزيرة الوقواق هاد الشي كاع مكان فروسنا إو دبا صوت انت أو خلينا في التيساع مافينا مايشهد الزووووور. 😕
Tokenization	„الله„ „يرحم„ „من„ „قراك„ „راك„ „فسرتي„ „لينا„ „دستور„ „جزيرة„ „الوقواق„ „هاد„ „الشي„ „كاع„ „مكان„ „فروسنا„ „إو„ „دبا„ „صوت„ „انت„ „أو„ „„„خلينا„ „في„ „التيساع„ „مافينا„ „مايشهد„ „الزووووور „ 😕
Cleaning	الله يرحم من قراك راك فسرتي لينا دستور جزيرة الوقواق هاد الشي كاع مكان فروسنا إو دبا صوت انت أو خلينا في التيساع مافينا مايشهد الزووووور سلبي
Normalization	الله يرحم من قراك راك فسرتي لينا دستور جزيرة الوقواق هاد الشي كاع مكان فروسنا إو دبا صوت انت أو خلينا في التيساع مافينا مايشهد الزوور سلبي

Removing Stop Words	الله يرحم قراك راك فسرتي لينا دستور جزيرة الوقواق هاد الشي كاع مكان فروسنا صوت انت خلينا التيساع مافينا مايشهد الزوور سلبي
Stemming	الله رحم قر راك فسر لين دستور جزير وقوق هاد شي كع مكان فروس صوت انت خلين تيسع مافينا مايشهد زور سلبي

The Cleaning and Standardization part consists of:

- Keep emojis: We do not delete emojis during preprocessing. However, they are replaced by a meaningful word; if it is a positive emoji (e.g., 😄, 😂, 😁) it is replaced by

  ايجابي

  otherwise, if it is a negative emoji (e.g., 😩, ☹️, 😭) we replace it with

  سلبي

- Remove HTML tags, as well as any Latin characters.
- Delete emails and URLs.
- Suppress diacritics(ضمتان، فتحتان، كسرتان، فتحة، كسرة، ضمة، سكون،
- Remove Tatweel (—) « Tatweel is an essential element of Arabic typography, used to elongate characters and add beauty to calligraphy ».
- Insert a space before and after all non-Arabic numerals or the Arabic alphabet and the Latin alphabet or the 2 square brackets, and then insert a blank space between the words and the numbers or the numbers and words.
- Replace the repetition of more than 2 non-numeric characters by 2 of this character.
- Replace the Hindi numbers with the standard caractere. Ex: ٥٩٩١ ← 5991
- Remove punctuation.

3) Phase 3: Extracting and Selecting Variables

From the cleansed and standardized textual data, we obtain terms (simple and compound) extracted from the data collection [18]:

الله رحم قر راك فسر لين دستور جزير وقوق هاد شي كع مكان فروس صوت انت خلين تيسع مافينا مايشهد زور سلبي

Then, to extract the important terms, we calculate the Term Frequency (TF) or Term Frequency-Inverse Document Frequency (TF-IDF) after using the n-gram technique.

n-gram approach

The n-gram approach [19] is a technique used in natural language processing to represent text as a set of contiguous sequences of n items. These items could be words, letters, or other units of text, depending on the application. For example, in the case of $n = 2$, the approach would create a set of all possible pairs of consecutive words in a given text, called bigrams. In the case of $n = 3$, the approach would create a set of all possible sequences of three consecutive words, called trigrams, and so on. The n-gram approach is used for various natural language processing tasks such as language modeling, text classification, and machine translation. It has been widely used in traditional statistical approaches to natural language processing, and it is also utilized in modern deep learning techniques such as convolutional and recurrent neural networks.

Calculate TF and TF-IDF weighting

The TF-IDF [20] (Term Frequency-Inverse Document Frequency) method is a common weighting technique employed in information retrieval, especially in the field of text mining. This statistical measure allows us to gauge the significance of a term within a document (in our context, each comment is treated as a document) relative to a larger collection or corpus. The weight assigned to a term increases as it appears more frequently within the document, and it also varies depending on how often the term appears across the entire corpus. Modified versions of the original formula are frequently used in search engines to evaluate the relevance of a document in response to a user's search criteria.

To rank terms by their importance, the TF-IDF weights of terms within the collection are computed using the following formulas, leading to the creation of a comment-term matrix (as depicted in Figure 4).

The Term Frequency (TF) for a term i in a comment j is calculated as:

$$TF(i, j) = \frac{\text{occurrence of term } i \text{ in comment } j}{\text{number of terms in comment } j} \tag{1}$$

The Inverse Document Frequency (IDF) for a term i is calculated as:

$$IDF(i) = \log\left(\frac{N}{\text{number of comments containing term } i}\right) \tag{2}$$

where N is the total number of comments.

The TF-IDF weight for a term i in a comment j is then calculated as:

$$\text{TF-IDF}(i, j) = TF(i, j) \times IDF(i) \tag{3}$$

```
Name: sentence, Length: 10254, dtype: object
()TfidfVectorizer
0.11374113733967565    (11636 ,0)
0.2634897728534007     (9690 ,0)
0.17376415583259466    (12036 ,0)
0.28712733746201924    (13188 ,0)
0.3239660240398204    (10049 ,0)
0.2589487782297278     (7359 ,0)
0.3436603407827087     (4690 ,0)
0.5150933278923234      (873 ,0)
0.3592925994160964     (2376 ,0)
0.18136173409421277     (8278 ,0)
0.25619375693614505     (1978 ,0)
0.14815360637288472    (13292 ,0)
0.3539728562633428     (9321 ,1)
0.19279354404524138     (5745 ,1)
0.33014028320020034     (8886 ,1)
```

Figure 4. Example comment-term matrix with tf-idf weighting.

Each term has a higher weighting than other weights, which will be important in the document to which it belongs. Finally, candidate key terms are selected, then ordered by importance, and finally, the most important comments are selected as key terms. At the end of this step, we obtained data ready to train the learning model.

4) Phase 4: Classification of Comments

To classify comments, we applied 5 supervised classification algorithms (implemented on Python software): Logistic Regression (LR), Decision Tree (DT), Support Vector Machines (SVM), Multinomial Naive Bayesian, and the famous XGBoost algorithm. We try to explain each one of them:

Logistic Regression (LR)

Logistic regression is a prediction model that associates a vector of random explanatory variables with a binary or multinomial dependent variable. It uses a logistic function (sigmoid) to predict classes rather than continuous numerical values. The model produces a bounded logistic curve between 0 and 1 [21].

Support Vector Machines (SVM)

SVMs are supervised learning techniques used to categorize text. They seek to determine separating hyperplanes that optimize the minimum distance between the training examples and the separation boundary. SVMs also make it possible to transform the data representation space into a larger space through kernel functions [21], [22].

Decision Tree (DT)

Decision trees are simple and interpretable classifiers. They divide the data space into subsets based on attribute-value conditions and are often used for data mining and emotional analysis [21].

Multinomial Naive Bayesian

The naïve Bayes classifier is based on Bayes' theorem and assumes that the effect of an attribute value on a given class is independent of other attributes. It is used to predict the probability of belonging to a specific class [22].

XGBoost

XGBoost, an algorithm introduced in 2015, is known for its power in Machine Learning. It is used for regression and classification. XGBoost is a supervised learning algorithm that incorporates regularization mechanisms to avoid overfitting, while maintaining good performance on large amounts of data [23].

In sum, this article explores various machine learning models, explaining their principles and applications. Each of these models has unique advantages and characteristics that make them suitable for different types of prediction and classification problems.

B. Deep Learning Approach

1) Neural Network approach

Regarding data collection and preprocessing, this model follows the same processes as the previous model, and they differ from the previous model in the Comment Classifications phase, as shown in Figure 5 below.

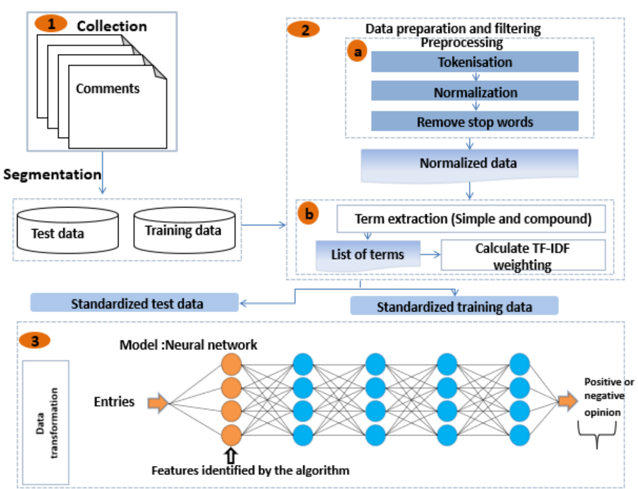


Figure 5. Architecture based on neural networks

Our neural network model consists of an input layer that does not exceed 300 words transformed into a matrix of numbers following the TF-IDF method, an intermediate layer, then a Relu activation function, and the dropout function, then an output layer, and the sigmoid function. (see Figure 6 below):

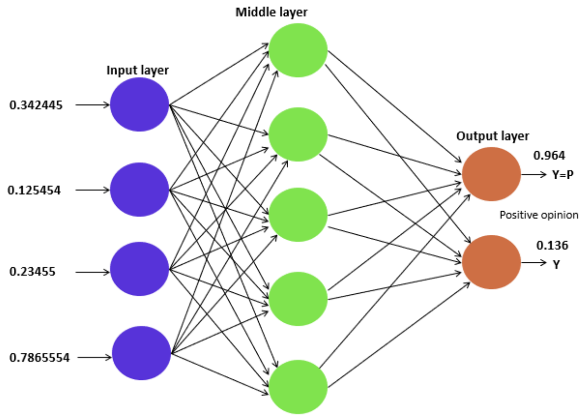


Figure 6. Our model of Neural Networks

2) Transfer learning approach using ARABERT

Since ARABERT is presented as an extension of the famous BERT algorithm, it is therefore obvious to start the latter in the first paragraph before seeing ARABERT in the next paragraph.

a) BERT [24]

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a state-of-the-art deep learning model that has revolutionized natural language processing (NLP) tasks. At the heart of BERT lies the transformer architecture, which has become the backbone of many successful NLP models. Transformers are neural networks designed to capture the contextual relationships and dependencies within a sequence of words or tokens. Unlike traditional recurrent neural networks, transformers employ self-attention mechanisms to model long-range dependencies efficiently. This allows BERT to capture rich contextual information from both the left and right contexts of each word, enabling it to understand the nuances and intricacies of natural language. With its pre-training and fine-tuning approach, BERT has achieved remarkable performance across various NLP tasks, propelling it to become a widely adopted model in the research community.

b) ARABERT [8]

ARABERT is an extension of BERT specifically designed for the Arabic language. Developed by a team of researchers, ARABERT takes advantage of the powerful language modeling capabilities of BERT and tailors them to address the unique characteristics and challenges of Arabic text. Arabizi, a mixture of Arabic and Latin characters commonly used in online communication, poses a significant challenge for Arabic NLP. AraBERT addresses this challenge by incorporating a pre-training phase that utilizes large-scale Arabic text data, allowing the model to learn Arabic language representations effectively. With the ability to capture the nuances of Arabic language, AraBERT has demonstrated remarkable performance on various Arabic NLP tasks, including sentiment analysis, named entity recognition, and text classification. The availability of AraBERT has greatly empowered researchers and practitioners working with Arabic text, enabling them to leverage the power of transfer learning and achieve state-of-the-art results in Arabic NLP applications.

The ARABERT architecture is based on the concept of transfer learning, which consists of completing the training of a machine learning model, previously trained to solve a given task, in order to allow it to solve a similar task. Indeed, the model used is already pre-trained on a large database (for ARABERT 70 GB of data) and our database will only be a last layer where we exploit its training results obtained. (see figure Figure 7 below)

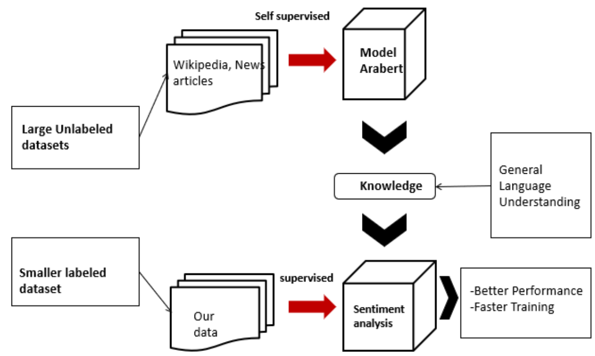


Figure 7. Transfer learning scheme

The final ARABERT architecture is given by the figure 8 below:

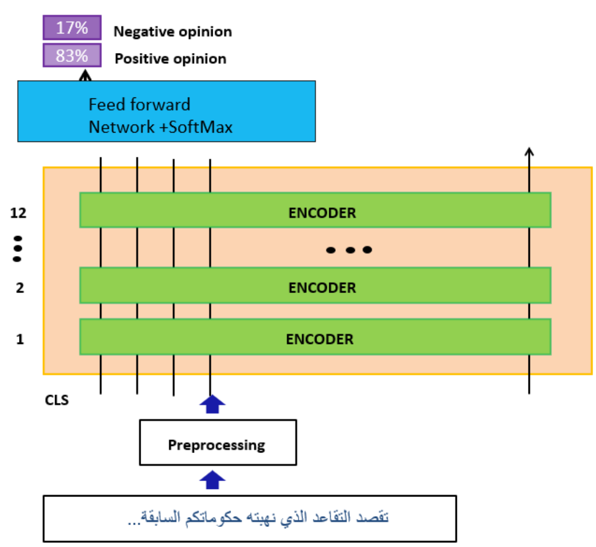


Figure 8. ARABERT architecture

4. Evaluation Measures

The evaluation of sentiment analysis applications is determined through experimental testing using test data. In the case of binary classifiers, various performance metrics are employed, which rely on specific parameters. These metrics are applied after a binary classifier has been used on a test dataset, and the results are typically represented in a format like Figure 9 (for specific details, refer to Table II).

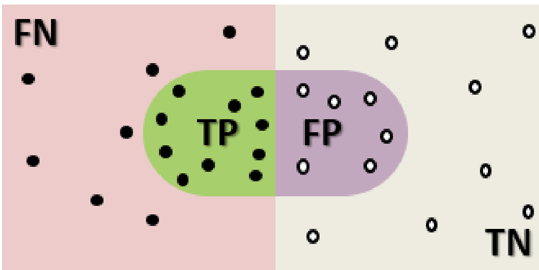


Figure 9. Evaluation measures

The terms "positive" and "negative" in this context refer to the sentiment or prediction orientation made by a classifier. Meanwhile, "true" and "false" indicate whether these predictions align with the actual outcomes, as outlined below:

- True Positive (TP): represents the count of positive instances correctly predicted as positive.
- False Positive (FP) signifies the count of negative instances incorrectly labeled as positive.
- True Negative (TN) indicates the count of negative instances correctly classified as negative.
- False Negative (FN) denotes the count of positive instances erroneously marked as negative.

TABLE II. Table of Contingencies

	Real Positive	**Real Negative**
Positive Test Result	True Positive (TP)	False Positive (FP)
Negative Test Result	False Negative (FN)	True Negative (TN)

Accuracy

In binary classification, accuracy quantifies the proportion of correctly predicted instances compared to the total number of instances predicted. It is calculated as:

$$\text{Accuracy} = \frac{TP}{TP + FP} \tag{4}$$

Recall

Recall, also known as sensitivity, is defined as the count of true instances correctly predicted among all the actual instances, including both true positives and instances incorrectly predicted as positive:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5}$$

Measure F

The F-measure is a combined metric that considers both precision and recall. It can be calculated as follows:

$$F = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

For different emphasis on precision or recall, the F-measure can be adjusted with β:

$$F_\beta = \frac{(1+\beta^2) \cdot \text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{7}$$

5. RESULTS AND DISCUSSION

A. Using Machine Learning Algorithms

The combination of extraction and weighting schemes allowed us to test six different configurations , for which we applied two different approaches: Tf and TF-idf. Each dataset was divided into two subsets: 80% for training, 20% for testing. The table below (Table III) summarizes the results of the experiments carried out. For each configuration, we presented the following metrics: VP, VN, FP, FN, accuracy, Recall, F1score. The following table III represents a summary of the results found:

Results

As shown in Table III, we summarize the classification results by configuration, demonstrating the performance of various classifiers across different feature extraction and weighting techniques.

TABLE III. Summary of Classification Results by Configuration

Configurations	Classifiers	VP(%)	VN(%)	FP(%)	FN(%)	Accuracy	Recall	F1 Score
Unigram/TF	SVM	71	82	29	18	78%	78%	78%
	TD	61	78	39	22	72%	72%	72%
	LR	68	86	32	14	80%	80%	80%
	MNB	72	84	28	16	79%	79%	79%
	Xgboost	57	91	43	9	79%	79%	80%
Unigram/TF-IDF	SVM	67	86	33	14	79%	79%	80%
	TD	60	75	40	25	70%	69%	69%
	LR	62	92	38	8	82%	82%	82%
	MNB	72	83	28	17	79%	79%	79%
	Xgboost	57	90	43	10	79%	79%	80%
Bigram/TF	SVM	53	89	47	11	76%	76%	77%
	TD	80	41	20	59	54%	54%	54%
	LR	48	91	52	9	76%	76%	78%
	MNB	66	74	34	26	71%	71%	71%
	Xgboost	45	91	55	9	75%	75%	77%
Bigram/TF-IDF	SVM	53	88	47	12	76%	76%	77%
	TD	40	85	60	15	51%	51%	53%
	LR	46	91	54	9	75%	75%	77%
	MNB	68	74	32	26	72%	72%	72%
	Xgboost	49	87	51	13	74%	74%	75%
(Unigram+Bigram)/TF	SVM	68	86	32	14	80%	80%	80%
	TD	59	80	41	20	73%	73%	73%
	LR	65	89	35	11	81%	81%	81%
	MNB	71	83	29	17	79%	79%	79%
	Xgboost	57	90	43	10	79%	79%	79%
(Unigram+Bigram)/TF-IDF	SVM	53	88	32	12	76%	76%	76%
	TD	66	68	34	32	67%	67%	66%
	LR	63	91	37	9	81%	81%	82%
	MNB	68	74	32	26	79%	79%	79%
	Xgboost	67	82	33	18	77%	77%	76%

Discussion

- For the **Logistic Regression algorithm**, we observe that the results are relatively stable (between 80% and 82% accuracy) across both TF and TF-IDF approaches for Unigram and (Unigram + Bigram) combinations. However, the Unigram configuration with the TF-IDF approach shows a slight advantage, achieving an accuracy of 82%, even though this model's training success is only 89%. The Bigram configuration, on the other hand, yielded less remarkable results with both approaches.

- The **Decision Tree algorithm** performed best with the (Unigram + Bigram) combination using the TF method, reaching a 73% success rate. Other approaches yielded modest results, especially the (Bigram + TF-IDF) method, which only achieved a 51% success rate, despite the models being well-trained across all configurations (99.9% training success).

- With the **MNB algorithm**, excluding the Bigram approach, all configurations resulted in a 79% accuracy rate. A relatively stable recall rate of 84%, and an F1 score of 84% for negative sentiments and about 70% for positive sentiments were observed. These models were well-trained, with more than 95% training success. Notably, better results across all configurations were obtained for negative sentiments.

- The **SVM algorithm** showed that all models are well trained, with 5 out of 6 configurations exceeding 99% in training success. The best result was obtained with the (Unigram + Bigram) configuration using both approaches, achieving 81% success. As with other algorithms, better outcomes were noted for negative sentiments, with an accuracy of 88%, compared to 68% for positive sentiments.

- Similar to the MNB algorithm, the **Xgboost algorithm** also reported a 79% accuracy rate for both the Unigram and (Unigram + Bigram) configurations with the TF approach. A slight difference in accuracy was noted, with 90% success for negative sentiments and 57% success for positive sentiments in the (Unigram + Bigram) / TF configuration.

B. Using Neural Network algorithm

Using Neural Network algorithm, we obtain those results:

Results

The performance metrics for the Neural Network Algorithm under the TF-IDF configuration are detailed in Table IV, showcasing the algorithm's balanced accuracy, recall, and F1 Score.

TABLE IV. Neural Network Algorithm Results with TF-IDF Configuration

Metric	Value (%)
True Positive (TP)	80
True Negative (TN)	80
False Positive (FP)	20
False Negative (FN)	20
Accuracy	80
Recall	79
F1 Score	79

Discussion

The result found with our neural network model is about 80% accuracy. A very promising result since the model is simple and consists of only one intermediate layer and with only three epochs and a batch size=100.

This promising level of accuracy underscores the efficiency of our neural network model, even with its simplicity and limited training parameters. An example of the predictions made by the neural network model is illustrated in Figure 10, showcasing the model's performance on sample data.

```
#for x in data["comment_message"][:10]:
x="تبارك الله عليك خويا "

tokens = tokenizer.texts_to_matrix([x], mode='tfidf')

c=model.predict(np.array(tokens))
    #cc=model.predict_classes(tokens)
cc = np.argmax(c,axis=-1)
xc = encoder.inverse_transform(cc)


print(c,"= \t",cc,"= \t",xc)
[[0.13392395 0.82614684]] =      [1] =    ['P']
```

Figure 10. NN prediction example.

C. Using ARABERT algorithm

Results

The ARABERT configuration's effectiveness in classification tasks is demonstrated through various metrics presented in Table V, highlighting its strong performance in terms of accuracy, recall, and F1 Score.

TABLE V. BERT Classifier Results with ARABET Configuration

Metric	Value (%)
True Positive (TP)	84
True Negative (TN)	89
False Positive (FP)	16
False Negative (FN)	11
Accuracy	88
Recall	88
F1 Score	87

The comprehensive classification performance of the ARABERT model is encapsulated in the classification report, as depicted in Figure 11, which details the precision, recall, and F1 scores across various categories.

```
              precision    recall  f1-score   support

           0       0.89      0.91      0.90      1308
           1       0.84      0.81      0.82       743

    accuracy                           0.88      2051
   macro avg       0.87      0.86      0.86      2051
weighted avg       0.87      0.88      0.87      2051

Arabert traitement : 0.8751828376401756
```

Figure 11. ARABERT's Classification report.

Discussion

About 88% success, which means that our Arabert-based model was able to correctly predict 1804 reviews out of 2051 in total. The following graph illustrates the correct prediction number among the first 50 comments. Only the dots in Blue or Pink are incorrect. We notice that only 4 comments among these 50 comments are incorrect, which shows the credibility of this algorithm.

The effectiveness of our approach is visually summarized in the prediction chart, shown in Figure 12, which highlights the model's predictive capabilities across a set of test data.

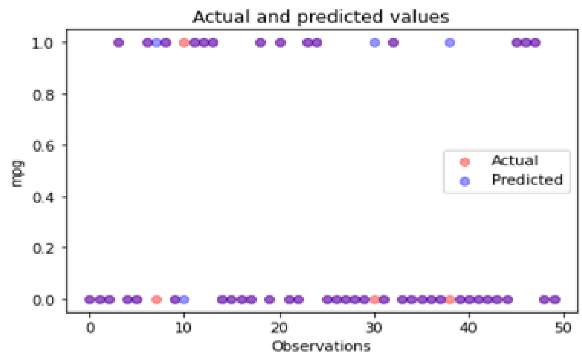


Figure 12. Prediction chart

An illustrative example of an ARABERT prediction is presented in Figure 13, demonstrating the model's capability to accurately classify the sentiment of a given text snippet.

```
texts = ['هه معروفين باسلاء على وزارية هه كعكة للجميع']
predictions = learner.predict_batch(texts)
predictions
```

```
[[('N', 0.9960553646087646), ('P', 0.003944649826735258)]]
```

Figure 13. Example of ARABERT prediction

D. Summary

We tried in this study to detect sentiments from commentary written in Arabic. For this, we used a total of 7 algorithms: 5 in machine learning (logistic regression, decision tree, Bayesian, SVM, and XGBoost) and 2 in deep learning: neuron network and ARABERT. We based ourselves on the TF_IDF method, and TF for the first 6 algorithms, and on BERT for ARABERT. In general, these results at the level of machine learning algorithms show that the best performance was obtained with the combinations [Unigram/TF-IDF] and [Unigram + Bigram/TF-IDF] regardless of the algorithm used. Considering the impact of variable selection for different configurations, we can conclude that Logistic Regression with Unigram extraction and TF-IDF weighting, is the most efficient in terms of precision and accuracy. The following table VI and charts show the summary of the best results obtained for each configuration chosen:

Figure 14 shows the accuracy graph for machine learning algorithms. And at the level of Deep Learning,Figure 15 shows that the best performance was obtained with ARABERT based on the Bert algorithm.

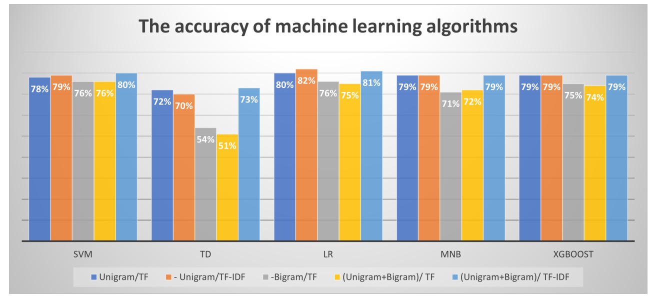


Figure 14. Accuracy graph for machine learning algorithms

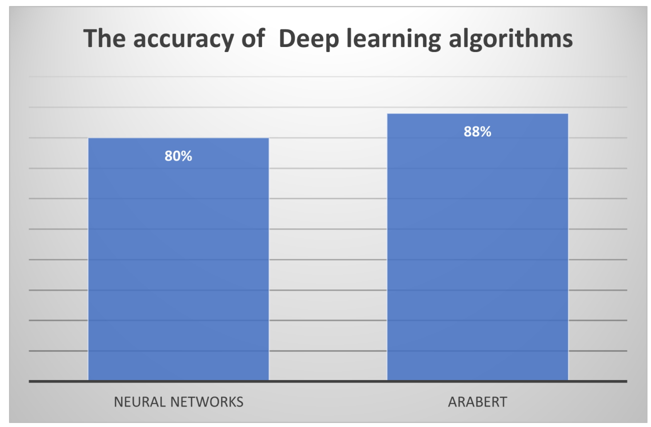


Figure 15. Accuracy graph for deep learning algorithms

TABLE VI. Summary of Results

Configurations	Classifiers	Accuracy	Recall
TF_IDF	LR	82%	82%
TF_IDF	Neuron Network	80%	79%
BERT	Arabert	88%	88%

6. Conclusion

Sentiment analysis has gained significant popularity in the research community as it tackles the challenges of analyzing unstructured text data for determining emotions and opinions. Although extensive research has been conducted in this field, numerous problems persist due to the nature of sentiment analysis. This has attracted the attention of many researchers, resulting in the proposal of numerous models that have demonstrated promising results across various sentiment analysis tasks. In this brief, we first presented the process of sentiment analysis, including its applications, tasks, and sentiment analysis challenges.

Similarly, we presented our methodology which is based on both traditional approaches (different machine learning algorithms) and modern approaches (especially the transfer learning technique used with ARABERT).

Indeed, our main contributions in this work include:

1) Describe the properties of Arabic languages, in particular Dialect Arabic and its challenges for sentiment analysis.
2) Propose a set of techniques to preprocess written comments for sentiment analysis.
3) Build and select entities (words or sequences of words) from comments to get the best sentiment classification model.
4) Realized a comparative study between the different algorithms used.

From this survey, it can be concluded that machine learning approaches offer better accuracy with moderation *[Unigram + Bigram/TF-IDF]* regardless of the algorithm used.

Since sentiment analysis is used to predict the polarity of users' opinions and deep learning models are all about predicting or imitating the human mind, deep learning models offer more accuracy than machine learning models such as deep neural networks and especially the famous Arabert algorithm with which we obtained an accuracy of 88%. A result Better than any other algorithm tested in this study.

To further enhance this brief, several future perspectives are proposed, including:

1) The need for developing a specialized stemmer for Moroccan dialectal Arabic, as the results obtained using existing stemmers have been modest.
2) Adapting the approach to detect sentiment communities in social networks by considering logical links, which would aid in identifying groups of users with similar opinions.
3) Exploring the possibility of automating opinion annotation with a semi-supervised approach, as it was done manually in the current work.

Overall, these perspectives aim to enrich the understanding and application of sentiment analysis in various contexts, paving the way for future advancements in the field.

Acknowledgment

I would like to express my sincere gratitude to all those who have contributed to the completion of this research work. Your support and assistance have been invaluable throughout this journey.

I am deeply thankful to my advisors and mentors for their guidance and expertise. Your insights and feedback have been instrumental in shaping this research.

I would also like to extend my appreciation to my colleagues who provided valuable input and constructive discussions, which enriched the quality of this research.

Lastly, I am grateful to my family and friends for their unwavering encouragement and understanding during this research.

References

[1] S. Mezhoud, "Arabic language and computers. application of computational linguistics to serve the arabic language," *Altralang Journal*, vol. V3, no. 01, 2021. [Online]. Available: https://www.univ-oran2.dz/revuealtralang/index.php/altralang/article/download/108/97

[2] X. Qu *et al.*, "A survey on arabic named entity recognition: Past, recent advances, and future trends," *IEEE Trans. Knowl. Data Eng.*, Feb. 2023.

[3] H. Elsayed and T. Elghazaly, "A named entities recognition system for modern standard arabic using rule-based approach," Apr. 2015.

[4] A. Shaker, A. Aldarf, and I. Bessmertny, "Using lstm and gru with a new dataset for named entity recognition in the arabic language," Apr. 2023.

[5] N. Nassiri, V. Cavalli-Sforza, and A. Lakhouaja, "Mosar: Modern standard arabic readability corpus for l1 learners," in *Proceedings of the 3rd International Conference on Arabic Computational Linguistics*. Doha, Qatar: ACM, Oct. 2019.

[6] H. Bouressace, "Computational analysis of printed arabic text database for natural language processing," *Culture and Society*, vol. 13, no. 3, 2023.

[7] A. Alnawas and N. Arıcı, "The corpus based approach to sentiment analysis in modern standard arabic and arabic dialects: A literature review," *Politeknik Journal*, Jun. 2018. [Online]. Available: https://dergipark.org.tr/en/download/article-file/438471

[8] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," *American University of Beirut*, 2020.

[9] M. Jbel, I. Hafidi, and A. Metrane, "Sentiment Analysis Dataset in Moroccan Dialect: Bridging the Gap Between Arabic and Latin Scripted Dialect," 2023. [Online]. Available: https://arxiv.org/abs/2303.15987

[10] M. Errami, M.-A. Ouassil, R. Rachidi, B. Cherradi, S. Hamida, and A. Raihani, "Sentiment Analysis on Moroccan Dialect Based on ML and Social Media Content Detection," 2023. [Online]. Available: http://thesai.org/Downloads/Volume14No3/Paper_47-Sentiment_Analysis_on_Moroccan_Dialect_based_on_ML.pdf

[11] R. Eskander, M. Al-Badrashiny, N. Habash, and O. Rambow, "Foreign Words and the Automatic Processing of Arabic Social Media Text Written in Roman Script," 2014. [Online]. Available: https://aclanthology.org/W14-3901.pdf

[12] A. Oussous, A. A. Lahcen, and S. Belfkih, "Improving Sentiment Analysis of Moroccan Tweets Using Ensemble Learning," in *Improving Sentiment Analysis of Moroccan Tweets Using Ensemble Learning*, 2018.

[13] N. Habbat, H. Anoun, and L. Hassouni, "Sentiment Analysis and Topic Modeling on Arabic Twitter Data during Covid-19 Pandemic," 2022. [Online]. Available: https://ojs.literacyinstitute.org/index.php/ijias/article/download/432/159

[14] M. Abdul-Mageed and S. Kubler, "SAMAR: A System for Subjec-

tivity and Sentiment Analysis of Arabic Social Media," *Computer Speech & Language*, vol. 28, no. 1, pp. 20–37, 2014.

[15] R. M. Duwairi and I. Qarqaz, "A framework for arabic sentiment analysis using supervised classification," *International Journal of Data Mining, Modelling and Management*, vol. 8, no. 4, pp. 369–381, 2016. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJDMMM.2016.081247

[16] A. Elouardighi, M. Maghfour, and H. Hammia, "Collecting and processing arabic facebook comments for sentiment analysis," in *International conference on model and data engineering*. Springer, 2017, pp. 262–274.

[17] L. S. Larkey, L. Ballesteros, and M. E. Connell, *Light stemming for Arabic information retrieval*. Springer, 2007, pp. 221–243.

[18] N. Boudad, R. Faizi, R. O. H. Thami, and R. Chiheb, "Sentiment analysis in arabic: A review of the literature," *Ain Shams Eng. J.*, 2017, in press.

[19] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1995, pp. 181–184.

[20] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[21] S. B. Kotsiantis, P. E. Pintelas, and I. D. Zaharakis, "Machine learning: a review of classification and combining techniques," *Artificial Intelligence Review*, vol. 26, pp. 159–190, 2006.

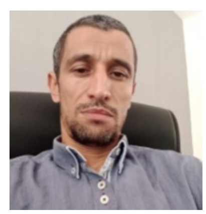
[22] M. Mohammed, M. B. Khan, and E. M. B. Bashier, *Machine learning: algorithms and applications*. CRC Press, 2016.

[23] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1. Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186.

ABDELLAH AIT ELOULI I am passionate about computer science and data analysis, holding a master's degree in data science and big data. I share my extensive expertise with my students as a computer science teacher while simultaneously advancing my own academic journey as a Ph.D. student. My enthusiasm for data exploration and dedication to teaching make me a valuable mentor for those seeking to grasp the intricacies of computer science and data analysis.

EL MEHDI CHERRAT He received his Ph.D degree from ENSA, the University Ibn Zohr Agadir Morocco in 2020. Now he is an assistant professor at department of Computer Engineering and Intelligent Systems, FSAAM, University of Ibn Zohr, Ait Melloul Morocco. He is the Vice-Head of the Department of Computer Engineering and Intelligent Systems. His research interests are Image Analysis, Recognition Biometrics, and Intelligent Systems.



HASSAN OUAHI Ph.D in Computer Science. Currently he hold the position of Assistant Professor in Computer Science at Ibn Zohr University of Morrocco since 2016. Before joining the University,he pursued his studies at the LabSIV Computer Science Laboratory of FSA agadir City, Morocco, from 2011 to 2015. His Research Interests are: Natural Language Processing, Internet of Things Security, Cloud Computing and Virtualization.

ABDELLATIF BEKKAR Received his B.S. in Computer Science from Ibn Zohr University, Morocco, in 2007, and his M.S. in Data Science and Big Data from ENSIAS, the IT College at Mohammed V University in Rabat, Morocco, in 2020. He is currently pursuing his Ph.D. in Computer Science at the LIM Laboratory, Faculty of Sciences and Technics, Hassan II University, Morocco. His research interests span machine learning, data analytics, air pollution modeling, and natural language processing.