DOI: https://doi.org/10.24297/ijct.v20i.8827

# SSAAR: An enhanced System for Sentiment Analysis of Arabic Reviews

[1]Manal Nejjari, Abdelouafi Meziane[2]

[1]Computer Science Laboratory, Science College, Mohammed 1st University, Oujda, Morocco

[2]Computer Science Laboratory, Science College, Mohammed 1st University, Oujda, Morocco

[1]manal.nejjari10@gmail.com, [2]abdelouafi_meziane@yahoo.fr

## Abstract

Sentiment Analysis, or Opinion Mining, has recently captivated the interest of scientists worldwide. With the increasing use of the internet, the web is becoming overloaded by data that contains useful information, which can be used in different fields. In fact, many studies have shed light on Sentiment Analysis of online data in different languages. However, the amount of research dealing with the Arabic language is still limited. In this paper, an empirical study is led to Sentiment Analysis of online reviews written in Modern Standard Arabic. A new system called SSAAR (System for Sentiment Analysis of Arabic Reviews) is proposed, allowing computational classification of reviews into three classes (positive, negative, neutral). The input data of this system is built by using a proposed framework called SPPARF (Scraping and double Preprocessing Arabic Reviews Framework), which generates a structured and clean dataset. Moreover, the provided system experiments two improved approaches for sentiment classification based on supervised learning, which are: Double preprocessing method and Feature selection method. Both approaches are trained by using five algorithms (Naïve Bayes, stochastic gradient descent Classifier (SGD), Logistic Regression, K-Nearest Neighbors, and Random Forest) and compared later under the same conditions. The experimental results show that the feature selection method using the SGD Classifier performs the best accuracy (77.1%). Therefore, the SSAAR System proved to be efficient and gives better results when using the feature selection method; nevertheless, satisfying results were obtained with the other approach, considered consequently suitable for the proposed system.

**Keywords**: Arabic Language, Machine Learning, Natural Language Processing, Opinion mining, Sentiment Analysis.

## 1. Introduction

Using the internet to express opinions or share information about different topics has become essential for people all around the world in their daily lives, especially among people speaking Arabic language. In the Middle East and north Africa, the number of internet users has reached on June 30, 2019, more than 286 million users who represent about 6% of the world's internet surfers who use the Arabic language. For that reason, an increasing interest has been shown recently by many researchers in analyzing this huge amount of information available on the web, including public opinions expressed on blogs, websites and social network, etc.By the way, for decision making and improving their trading activities, many companies resort to opinion mining. It is the same case for governments, which resort to public citizens' opinions to improve the quality of services and information offered to them. Individuals are also interested in others' opinions because it could help them to have an idea about a wanted product or service. However, studies aiming opinion mining or sentiment analysis from Arabic websites or social media stay limited and more research is required to solve many issues related to this topic, due to challenges in processing natural Arabic language with its complex morphology (a Morphologically Rich Language), and also the use of different types of dialectical Arabic, Arabizi (Arabic written in Latin script combined with Arabic numerals).

In this work, we are interested in producing a new system for sentiment analysis of online reviews and comments written in Modern Standard Arabic, including a framework for scraping and preprocessing data. This paper is organized as follows. Section 2 gives an overview of the latest works in sentiment analysis in Arabic and other languages. Section 3 presents the architecture of the proposed system. Section 4 outlines the experiment done and discusses the results obtained by testing two approaches of our system. In Section 5 the paper is concluded and the future work is presented.

## 2. Related work

### 2.1. Sentiment Analysis of Non-Arabic texts

To the best of our knowledge, most of the studies on sentiment analysis are conducted in the English language, due to the lack of resources in other languages. In fact, there are many issues related to sentiment analysis such as subjectivity detection, opinion extraction, irony detection and so on.

B. Pang and al [1] have focused in their work on two tasks of sentiment analysis, the first one is subjectivity classification, it is about classifying a text, into subjective or objective classes. The second task is classifying the subjective text into positive or negative one. However, the most studied task is sentiment polarity classification, to which many approaches have been applied.

Pang and al. [2] used the machine learning approach for text sentiment analysis, they applied naive Bayes, maximum entropy and SVM algorithms for sentiment analysis of reviews. According to experiments, the best performance was obtained using the SVM algorithm. Another approach used for sentiment analysis is the approach based on Semantic Orientation (SO) [3].

Using also the machine learning approach, Wang and al. [4] worked on sentiment analysis of short texts. They build a high-dimensional mixed feature sentiment analysis model based on SVM.

In his literature [5], M. Taboada used an approach based on sentiment dictionary to identify sentiment words and values. This approach does not need manual annotation of samples and is easy to implement but its performance depends essentially on the quality of the sentiment dictionary.

Kiritchenko et al. [6] used different sentiment features extracted from the high-coverage sentiment dictionary, which was generated from sentiment tweets, in order to work on supervised statistical text sentiment classification.

### 2.2. Sentiment Analysis of Arabic texts

Despite of the complexity the Arabic language processing, many scientists have chosen to challenge sentiment analysis in Arabic and deal with its different issues .

El-Halees's work [7] deals with Arabic sentiment analysis using a combined classification approach which consists of three methods (lexicon-based classification, Maximum Entropy and K-Nearest Neighbors) applied in sequence.

Abdul-Mageed and al. [8] worked on sentiment analysis in Arabic language using a machine learning approach. They built a sentence-level system for subjectivity and sentiment analysis. They also presented a newly developed manually annotated corpus of Modern Standard Arabic (MSA) together with a new polarity lexicon and described an automatic SSA tagging system that exploits the annotated data.

In their work [9], Abdul-Mageed and al. Proposed a supervised machine learning system for Arabic SSA (Subjectivity and Sentiment Analysis), called 'SAMAR'. In fact, they built a binary classifier to distinguish objective from subjective cases, then positive sentiment from negative one. SVM was used as a learning algorithm together with language specific and general features.

Ahmad and Almas [10] worked on sentiment analysis of financial texts. They tried to find some grammars or rules describing common and frequent Arabic patterns that are usually used in financial news to report object values changes. Then, they used these rules to visualize the changes in the sentiments contained in the news text.

Nabil and al. [11] presented an Arabic social sentiment analysis dataset collected from twitter, and described their used method for collecting and annotating the dataset. Then they tested a four-way sentiment classification that classifies texts in four classes: objective, subjective, subjective positive and subjective negative.

## *3.* Proposed System : SSAAR

SSAAR (System for Sentiment Analysis of Arabic Reviews) is a machine learning based system for sentiment analysis of Modern Standard Arabic reviews and comments available on the web, it allows a classification of our input comments into three classes (positive, negative and neutral), using a supervised approach. In this proposed system, we develop a new framework called SPPARF for scraping data from the web and gathering it into structured shape, which will be detailed later. We also test two different approaches for comment classification, and evaluate their performance so as to find the ultimate one for our case of study.

### 3.1. Building Dataset

### 3.1.1. Information resources

In this work, we have collected more than 8153 Arabic comments from two information resources:

✓ The Arabic version 'booking.com': it is one of the best-known travel websites, established in 1996, it helps connect travellers with the world's largest selection of places to stay in, including everything from apartments to 5 star luxury resorts. The booking website and mobile application are available in over 40 languages including Arabic, offer 29.006.647 total reported listings and cover more than 153,519 destinations in 225 countries and territories worldwide1.

✓ The Arabic website 'blogs.aljazeera.net': it is a free online news platform, created by Aljazeera Media Network where users can post comments, share photographs and post videos related to the world's news or other interesting content on the web. It is very popular ad interactive website in the Arab world, and the majority of its bloggers are from North Africa and the Middle East.

Those two websites are not chosen at random, it is specially due to the fact that the use of modern standard Arabic in those websites is compulsory for writing reviews. A thing, which will help us reduce noise from our data and concentrate on other tasks than removing expressions written with dialectical Arabic or Arabizi and other non-MSA expressions.

### 3.1.2. SPPARF: Scraping and double preprocessing Arabic Reviews Framework

In this section, we will describe how we reduced scraped HTML files from our information resources into a structured form containing the relevant insight for our work using a new framework called SPPARF.

Most of the files in our collected data are scraps of those websites in raw HTML format. Each website selected has a specific HTML page structure, thus we developed a customized parser for each one. The semi-structured form of HTML and the specific tags for each field allowed us to find easily information we needed on each page. In the Booking website we extracted the couple (Hotel name; Comments), and from Aljazeera Blog we extracted the couple (Topic – title – ; Comments).

Some parsed pages containing advertisements are very noisy, and contain some symbols and numbers, so, we applied web and data preprocessing techniques to clean our data. Otherwise, our work did not stop here, we built a data scraping and preprocessing framework called SPPARF where we can scrap automatically Arabic reviews and apply to them different Arabic preprocessing techniques in order to obtain cleaned and structured data.

The flow diagram of the proposed framework is illustrated in Fig.1, in which we could notice how all components are connected, and understand how the overall developed system works. The main purpose of the proposed

[1] https://www.booking.com/index.fr.html

system is to scrap and preprocess in double steps the Arabic comments from HTML pages (Websites, blogs, etc.). The framework contains three basic units:

i. Text Scraper;
ii. Processing the web;
iii. Preprocessing comments.

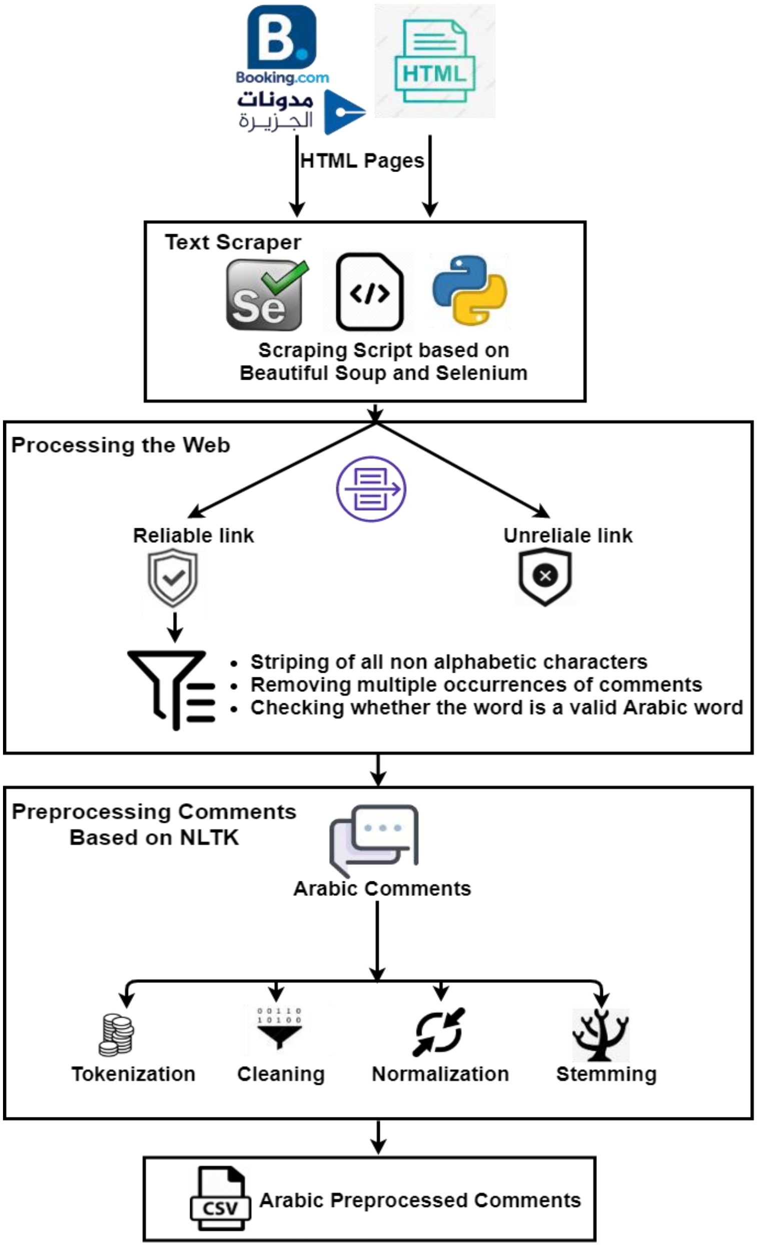

**Fig. 1:** The Scrapping and double Preprocessing Arabic Reviews Framework ( SPPARF*)*

In the following, we will give more details about each component.

➢ Text Scraper:

In this module, we used the Python programming language which contains libraries that are open source and very useful, here we used Beautiful Soup and Selenium Libraries. The input webpage goes through a series of operations, which facilitates the process of text extraction. For this purpose, we use Beautiful Soup, which is a Python library for pulling data out of HTML and XML files. We also use Selenium library, which is a package used to automate web browser interaction from Python. The objective of using those tools is developing a powerful web scraper based on Python language, and which allows text extraction from different types of webpages.

In order to get a global overview of our scrapped data, a word cloud representation is used to identify the most frequent words. The word cloud result of booking website is shown in Fig. 2. In fact, the most frequent words are those with the biggest font size and when font gets smaller the frequency decreases. For example, in the booking website's cloud, the word cloud shows that ‘الفندق’ and some words like ‘كان’;’في’ ; ’من’ ;’ على’ are the most frequent ones, this means that our scraped data is very noisy and contains some words which are useless for sentiment analysis and should be deleted, that is why we will resort later to data preprocessing.

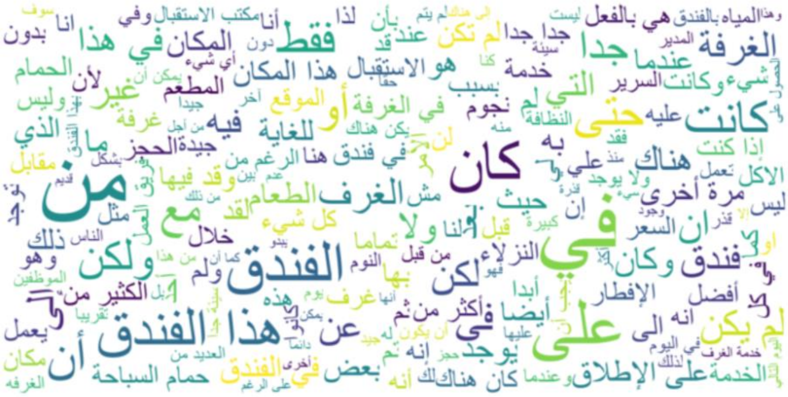

**Fig. 2:** Word cloud of our Dataset

➢ Processing the Web

In this step, and before applying Web-processing techniques, the scraped data is provided as an input to our framework, which has to classify in primary time the fetched results into a reliable and unreliable data. Then, only the reliable links are scraped and undergo some preprocessing techniques.

In the algorithm presented below, we describe the process:

```
Procedure ProcessTheLink (text)
    Link (GetFetchedPages (Text)
    Reliable (GetReliableLink (Link)
    For each link in Reliable
ScrapedPage (ScrapLink (Link)
    Return ScrapedPage
end procedure
```

After selecting a reliable link, the second task of the system is to extract only Arabic text and specific attributes like titles and comments in Modern Standard Arabic (MSA). Furthermore, other operations are executed by the system as the following :(1) striping of all non-alphabetic characters like emoji's (2) Removing multiple occurrences of comments, (3) checking whether the word is a valid Arabic word.

➢ Preprocessing Comments

In this step 'Preprocessing comments' of our proposed framework, we applied some preprocessing techniques using the Natural Language Toolkit (NLTK), which is a comprehensive Python library for natural language preprocessing (NLP) and text analytics [12]. In order to get a robust and less noisy data, the NLTK library provides us with an extremely powerful tool for performing the following tasks we need in our framework:

*Tokenization:* it is the process of breaking a flow of text into words, sentences, symbols, or other significant elements called token. The purpose of tokenization is the exploration of words in a sentence. Tokens lists represent a set of words used as an input for processing.

*Cleaning*: Text Cleanup means the removal of all unnecessary or unwanted information, eliminate redundant lines, filled the empty values and remove diacritics, punctuation and special characters, numbers, Latin letters etc. This step gives a reduced and clear dataset.

*Normalization*: Consists of converting some letters that have different forms in the same word to the same form. For example, the normalization of ‘إ’ ‘آ’ ‘أ’ is ‘ا’, also consists of eliminating diacritics.

*Stemming:* Stemming is the process of transforming expressions into their roots, for example, the expressions ’السرقات ’,’السرقة ’,’مسروقة’ become’ سرق’. This step is used for removing repetitive features and it is useful because it reduces significantly the size of the vocabulary.

### 3.2. Dataset Annotation

After using the first and the second modules of SPPARF to obtain an improved data, and before applying preprocessing techniques, we have to annotate our data because our proposed system relies on a supervised machine learning approach. Therefore, we have chosen two MSA native speakers who are in charge of manually annotating our data. Each annotator was given guidelines. In fact, they should specify the polarity of each comment written by a blogger, if it is positive, negative or neutral. Thus, the annotator assigns each comment with three possible labels: positive, negative, and neutral as illustrated in Table 1.

**Table 1:** Comments annotation

| **Comments** | **Polarity** |
|---|---|
| اشكر خدمات الغرف خديجة أحسن موظفة في الفندق | Positive |
| ....ضجيج مستمر خصوصا فترة الصباح من موظفات | Negative |
| ...لم يعجبني الفندق بتاتا ولا انصح بالسكن فيه | Negative |
| تجربة مختلطة، يمكن أن تكون أفضل | Neutral |
| متوسط | Neutral |
| ...خدمة العملاء والموظفين يعملون على راحتك | Positive |
| ...موظفة الاستقبال (فاطمة) إنسانة رائعة جدا | Positive |
| الافطار مقبول مقابل القيمة | Neutral |

### 3.3. Dataset Exploration

*Dataset size*

In order to better understand the nature of the various collected datasets, the number of comments for each class (positive, negative, neutral) is given below in Table 2.

**Table 2:** Number of comments for each class

| **Class** | **Number of comments** |
|---|---|
| Positive | 3353 |
| Negative | 2647 |
| Neutral | 2153 |
| **Total** | **8153** |

From Table 2, we notice clearly that there is not a big difference between the number of comments for each class, so we can say that our dataset is a balanced one. It is not a coincidence, we tried to build a balanced data in order to avoid overfitting problems during the training of our model.

That is why in the next step, we propose to calculate the comments length, because we notice in our dataset that the neutral comments are longer than the positive and negative ones. This is due to the fact that it is difficult to find a neutral comment; most comments which are classified as neutral,are a kind of a combination between positive and negative ones.

*Comments length*

In order to explore more our dataset and have a general idea about its components, we present below in Fig.3 a box plot, which enables us to study the distributional characteristics of the comments length in our dataset. By using a box plot, we can better understand our data by understanding its distribution, outliers, means, median and variance.

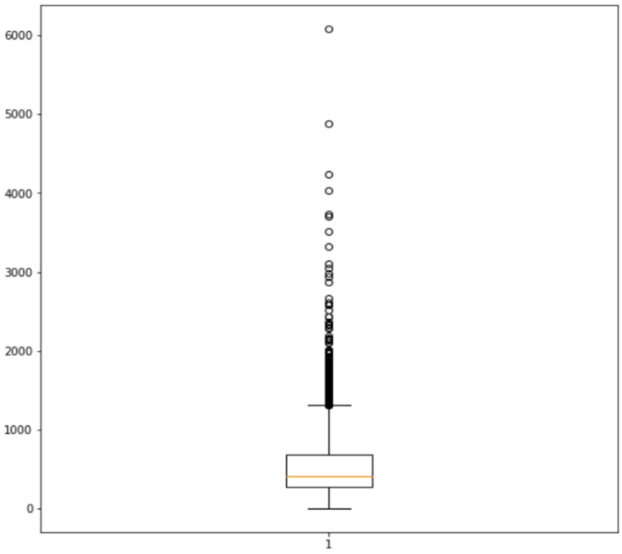

**Figure 3:** Box plot showing the distribution of comments length in our Dataset

We notice that the box plot is comparatively short. This suggests that overall comments have approximately the same length. Furthermore, some of the reviews are way more than 1000 characters long. Here we can say that the majority of the neutral comments are longer than 1000 characters long. Else, note that in this box plot we define comments length as the number of its characters as shown in Table 3.

**Table 3:** Example of comments length for each class.

| **Comment** | **Polarity** | **Comment length** |
|---|---|---|
| متوسط | Neutral | 8 |
| ....ضجيج مستمر خصوصا فترة الصباح من موظفات | Negative | 85 |
| غرفة رائعة شاطئ ساحر محمية طبيعية | Positive | 33 |
| خدمة ممتازة | Positive | 13 |
| ...يسرنا إبلاغكم أن المناسبة الخاصة بنا والتي | Positive | 368 |

## 3.4. Classification

After preparing and analyzing our dataset, we can use it now for the classification. Since we are working under a Machine Learning approach, we used five supervised classification algorithms to classify our dataset into three classes (positive, negative, neutral) which are: Naïve Bayes; SGD Classifier; Logistic Regression, K-Nearest Neighbors and Random Forest. The choice of those algorithms is based on literature. To the best of our knowledge, those are the most frequent machine learning algorithms used in the case of sentiment classification for different languages.

### 3.5. System Architecture

As explained in our work [13], getting sentiment from text can be solved through three approaches, machine-learning approach, lexicon based approach and hybrid approach. In this work, we use two different techniques for sentiment analysis of Arabic comments, which rely on supervised machine learning approach: Sentiment analysis using a double preprocessing method and Sentiment analysis using Feature Selection.

In the supervised machine learning based approach, sentiment analysis needs an annotated corpus, and here we have already annotated our dataset as seen in section 3.2.

Our proposed System Architecture is presented in Fig.4, which gives an overview of the complete process pipeline, including the SPPARF framework presented in section 3.1.2. Our proposed system tests two different approaches: the double preprocessing method and the feature selection method.

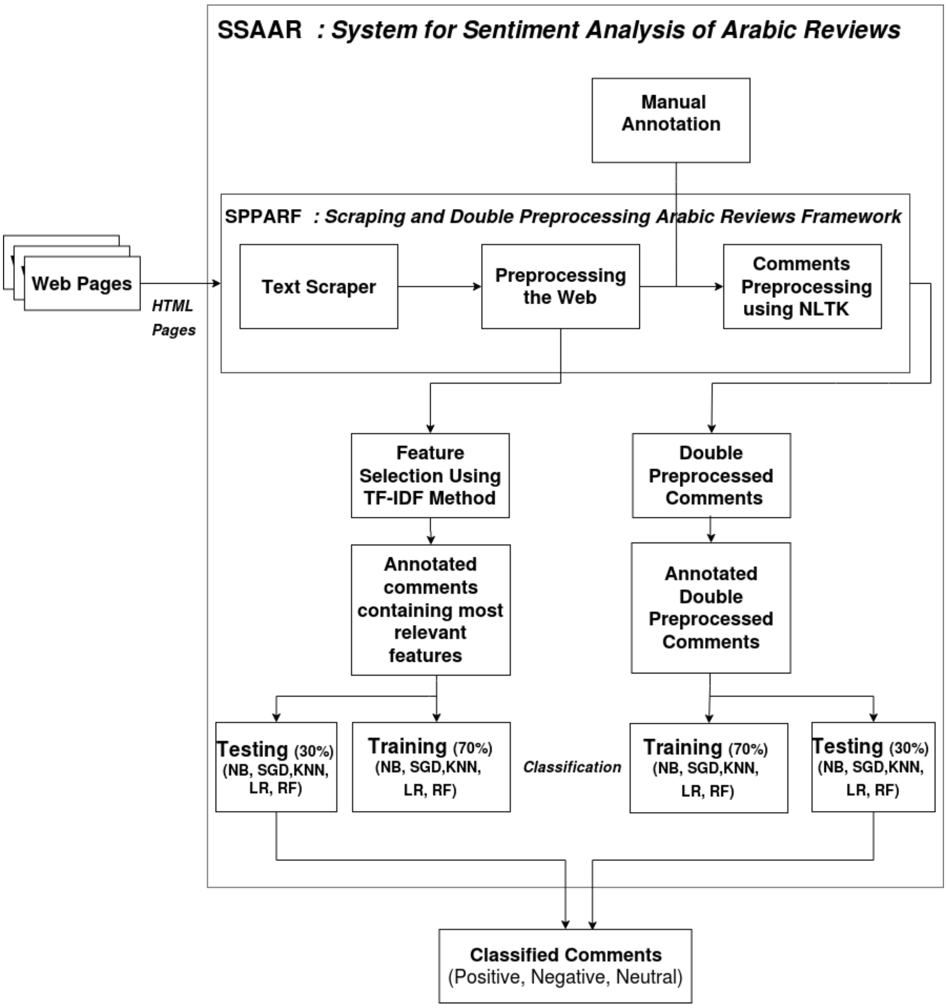
SSAAR : System for Sentiment Analysis of Arabic Reviews
Manual Annotation
SPPARF : Scraping and Double Preprocessing Arabic Reviews Framework
Web Pages
HTML Pages
Text Scraper
Preprocessing the Web
Comments Preprocessing using NLTK
Feature Selection Using TF-IDF Method
Double Preprocessed Comments
Annotated comments containing most relevant features
Annotated Double Preprocessed Comments
Testing (30%) (NB, SGD,KNN, LR, RF)
Training (70%) (NB, SGD,KNN, LR, RF)
Classification
Training (70%) (NB, SGD,KNN, LR, RF)
Testing (30%) (NB, SGD,KNN, LR, RF)
Classified Comments (Positive, Negative, Neutral)

**Figure 4:** SSAAR System Architecture

#### 3.5.1. Double preprocessing method

This method is based mainly on the double preprocessing conducted by the SPPARF described in section 3.1.2.

To prepare our dataset, we applied all SPPARF preprocessing techniques for both training and testing sets. We trained our model using five learning algorithms in order to see which one will perform better in our case. So the double preprocessed data is used to feed our five learning models.

We perform learning by using these double preprocessed comments as features. After applying tokenization, cleaning, normalization and stemming; we get a well preprocessed comments, where the tokens are significant words fed as features. Then the classification is done using the five supervised classifiers mentioned in section 3.4, in order to get three classes: negative, positive and neutral.

### 3.5.2. Feature Selection Method using TF-IDF

Features in the context of opinion mining are the words, terms or phrases that strongly express the opinion as positive or negative [14]. When we set out to classify comments in our dataset, we generally start with a very large number of words that need to be considered, even though very few of the words are actually significant and suitable for our case. In our Arabic sentiment classification case, we are interested in words related to expressing sentiment.

The presence of extra features have drawbacks. The first is that they make document classification slower, since there are far more words than there really need to be. The second is that they can reduce our model performance, since the classifier must consider these words when classifying a document.

Feature selection is the process of removing some of the unnecessary features in order to take advantage of using fewer features. This allows the classifier to fit a model to the problem set more quickly since there is less information to consider, and thus allows it to classify items faster.

Several methods are used to assess the importance of each feature by attaching a certain weight in the text, such as: feature frequency (FF), Term Frequency Inverse Document Frequency (TF-IDF), and feature presence (FP) [15].

TF-IDF is a common metric used in text categorization tasks, it is composed of two scores, term frequency inverse document frequency. Term frequency is found by simply counting the number of times that a given term has occurred in a given document, and inverse document frequency is found by dividing the total number of documents by the number of documents that a given word appears in. When these values are multiplied together, we get a score that is highest for words that appear frequently in a few documents, and low for terms that appear frequently in every document, allowing us to find terms that are important in a document [15].

In fact, in our second proposed approach, we consider feeding our model with Modern Standard Arabic comments after applying on them the first and second tasks of our framework SPPARF. The feature selection Method is used to pick out discriminating terms for training and classification by computing a score for each individual feature.

By the way, TF-IDF (term frequency-inverse document frequency) or Term weighting can be as simple as binary representation or as detailed as a mix of term and dataset existence probabilities stemmed from complex information theoretic underlying concepts [16].

In TF-IDF terms weighting, the text documents are represented as transactions. Selecting the keyword for the feature selection process is the main preprocessing step necessary for the indexing of documents. In our work, we used TF-IDF to weight the terms in term-document matrices of our evaluation datasets.

Then, supposing that 'D' is a document collection, 'W' is a word, and an individual document d. D, the weight 'w' is calculated using Equations 1 and 2 presented in [17].

$$TF = TF \times IDF \qquad (1)$$

$$W_d = f_{w,d} \times \log |D| / f_{w,d} \qquad (2)$$

where $f_{w,d}$or TF is the number of times 'w' appears in a document 'd', |D| is the size of the dataset, $f_{w,D}$or IDF is the number of documents in which 'w' appears in D. The result of TF-IDF is a vector with the various terms along with their term weight as shown in Table 4.

**Table 4:** Term weight in TF-IDF method

| Word | Frequency |
|---|---|
| جميل | 1275 |
| جيد | 1719 |
| سيئة | 1021 |
| رديئة | 1390 |

## 4. Experiments and results

### 4.1. Experimental Setting

In this work, experiments are carried out on our dataset built in section 3.1, composed of Arabic reviews written in modern standard Arabic and collected from two different websites. After manual annotation, we found out that our labelled data is constructed from 3353 positive comments, 2647 negative and 2153 Neutral as shown in Table 2. For Arabic reviews classification, we adapt and apply five learning models trained using the Supervised machine learning classifiers seen in section 3.4. All the algorithms were implemented using Python Language.

Note also that supervised learning is based on labelled dataset and thus the labels are provided to the model during the process. These labelled dataset are trained to produce reasonable outputs when encountered during decision-making. To help us to understand the sentiment analysis in a better way, in all our experiments, we have split our labelled dataset in a 70/30 ratio: 70% for training and 30% for testing as shown in Table 5, which is a common practice in data science, found in the literature. This also relies on the size of the data set and whether it is balanced or not.

**Table 5:** Dataset splitting

| Class | Dataset | Training set | Testing set |
|---|---|---|---|
| Positive | 3353 | 2347 | 1006 |
| Negative | 2647 | 1852 | 795 |
| Neutral | 2153 | 1507 | 646 |
| **Total** | **8153** | **5706** | **2447** |

### 4.2. Results and evaluation

To evaluate the performance of the proposed methods and several classifiers, the F1 score [18] is used after calculating the precision and recall. The accuracy shows the global correctness of the model by averaging the correct classifications on the total number of classifications [19]. The precision measures the accuracy of the classifier in regards to the specific predicted class. The recall or sensitivity of the classifier is the percentage of the correct predicted classes among the actual class in the data.

$$Recall = \frac{Number\ of\ correct\ predictions}{Number\ of\ examples}$$

$$Precision = \frac{Number\ of\ correct\ predictions}{Number\ of\ Predictions}$$

$$F1\_Score = \frac{2 * Recall * Precision}{(Recall + Precision)}$$

### 4.2.1. Double preprocessing method

As seen in section 3.5.1, this method relies on thorough cleaning of the dataset by applying all SPPARF preprocessing techniques for both training and testing sets. Then feeding our five models with the resulted data. Table 6 shows the result of accuracy, precision, recall, and f1-score for each class in the testing dataset and for each algorithm.

**Table 6:** Results of the Double preprocessing method

| Algorithm | Class | Accuracy | Precision | Recall | f1_score |
|---|---|---|---|---|---|
| **Naive Bayes Classifier** | ***Positive*** | 0.678 | 0.96 | 0.66 | 0.78 |
| | ***Neutral*** | | 0.02 | 0.91 | 0.05 |
| | ***Negative*** | | 0.84 | 0.70 | 0.76 |
| **SGD Classifier** | ***Positive*** | 0.770 | 0.90 | 0.82 | 0.85 |
| | ***Neutral*** | | 0.47 | 0.67 | 0.55 |
| | ***Negative*** | | 0.85 | 0.76 | 0.80 |
| **Random Forest** | ***Positive*** | 0.613 | 0.77 | 0.56 | 0.73 |
| | ***Neutral*** | | 0.16 | 0.46 | 0.24 |
| | ***Negative*** | | 0.77 | 0.70 | 0.65 |
| **Logistic Regression** | ***Positive*** | 0.713 | 0.91 | 0.70 | 0.79 |
| | ***Neutral*** | | 0.27 | 0.70 | 0.39 |
| | ***Negative*** | | 0.82 | 0.74 | 0.78 |
| **K Nearest Neighbors** | ***Positive*** | 0.612 | 0.70 | 0.75 | 0.72 |
| | ***Neutral*** | | 0.41 | 0.44 | 0.43 |
| | ***Negative*** | | 0.67 | 0.58 | 0.62 |

We notice that the SGD Classifier produces the best accuracy (0.77) which is slightly higher than Logistic Regression. (0.71). This means that in our case SGD Classifier and Logistic Regression are both suitable for Arabic sentiment analysis.

For more details, we present in Fig. 5, 6 and 7 the performance of the used classifiers for each class (positive, negative and neutral).

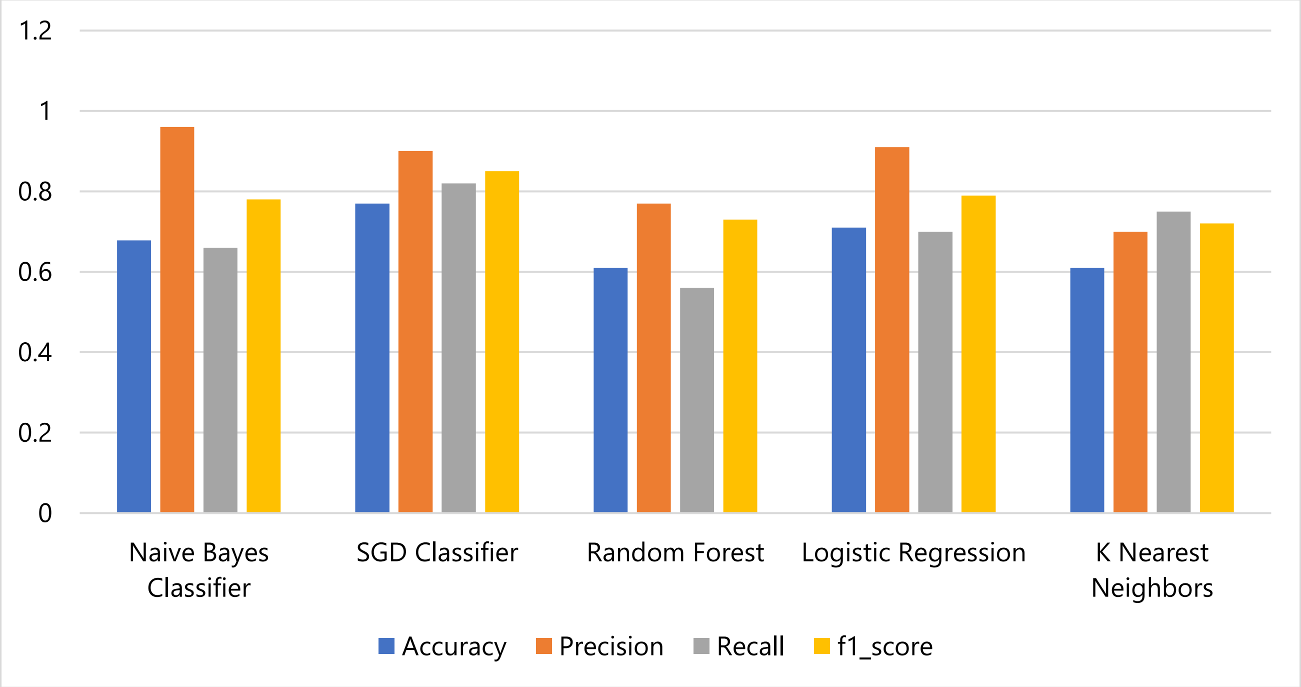

**Figure 5:** Performance of classifiers for positive comments using double preprocessing method

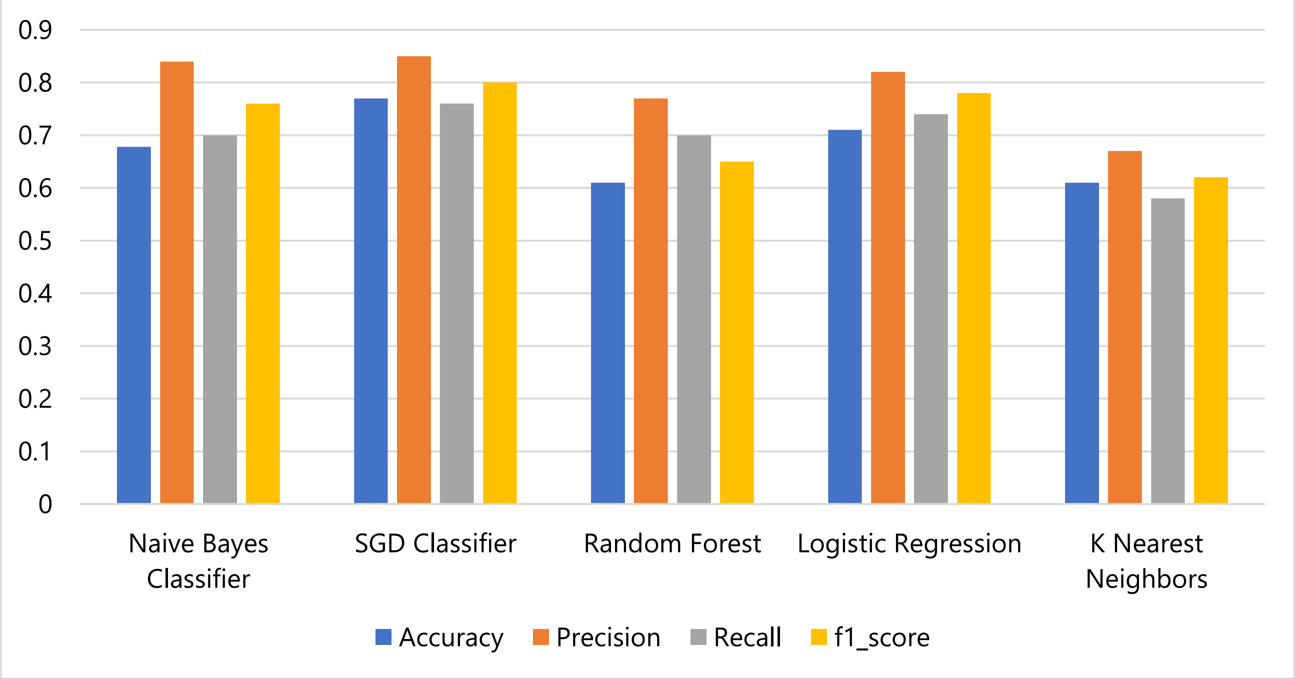

**Figure 6:** Performance of classifiers for negative comments using double preprocessingmethod

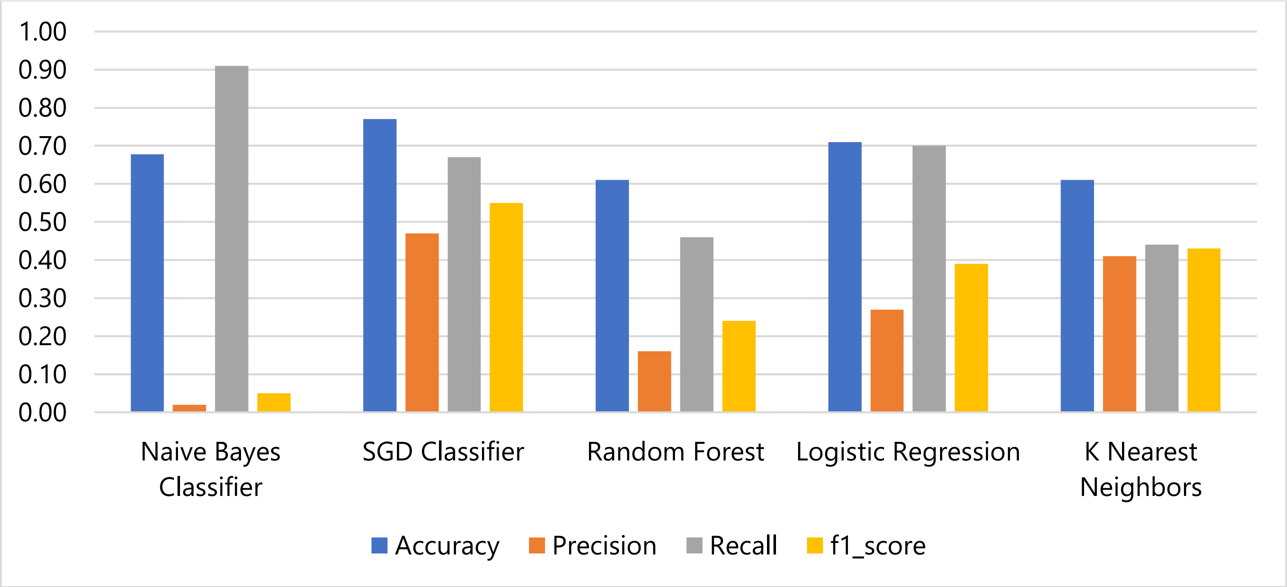

**Figure 7: Performance of classifiers for neutral comments using double preprocessing method**

### 4.2.2. TF-IDF method

In the TF-IDF feature selection method, Table 4 shows that the SGD Classifier performs the best across almost learning algorithms. Its accuracy is 0.77, which is one percent larger than Logistic Regression (0.76) and seven percent larger than Naïve Bayes Classifier (0.70).

The first remark is that feeding the model with selected features can give better results than feeding the learning model with the entire text. Another reflection is that feature selection method (TF-IDF) trained using the five chosen learning classifiers (i.e., SGD, KNN, LR, RF, and NB) exhibits good and robust performance.

**Table 7:** Performance of the classifiers using the TF-IDF method

| **Algorithm** | **Accuracy** | **Precision** | **Recall** | **f1_score** |
|---|---|---|---|---|
| **Naive Bayes Classifier** | 0.703 | 0.719 | 0.639 | 0.575 |
| **SGD Classifier** | 0.771 | 0.748 | 0.744 | 0.742 |
| **Random Forest** | 0.642 | 0.608 | 0.603 | 0.589 |
| **Logistic Regression** | 0.760 | 0.739 | 0.725 | 0.720 |
| **K Nearest Neighbors** | 0.666 | 0.642 | 0.643 | 0.641 |

### 4.2.3. Evaluation

TABLE 8 reports a comparison between accuracy results for the two proposed methods trained with the five aforesaid classifiers. This finding shows that SGD classifier is substantially more accurate than NB, RF, LR, KNN classifiers for unigram based feature weights, and still performing well for other types of features.

**Table 8: Performance comparison between the two used methods**

| | **NB** | **SGD** | **Random Forest** | **Logistic Regression** | **KNN** |
|---|---|---|---|---|---|
| **TF-IDF based approach** | 0.703 | 0.771 | 0.642 | 0.760 | 0.666 |

| | | | | | |
|---|---|---|---|---|---|
| **Double preprocessing based approach** | 0.678 | 0.770 | 0.613 | 0.713 | 0.612 |

From Tables 6, 7 and 8 presented above, we can draw the following conclusions:

- The tests and experiments were conducted using two types of input data: First using TF-IDF method for feature selection, and second using double preprocessing method.
- Each dataset was applied to five different classification algorithms, which are k-Nearest Neighbor, Naïve Bayes, SGD, random forest, and logistic regression.
- Performance Metrics: This research statistically measured the performance of the classification (positive, negative and neutral) tests that were conducted. Recall, precision, F1-score, and accuracy were calculated and analyzed.
- Using the Feature selection method: From Table 7 and based on the tests and experiments conducted for feature selection using TF-IDF algorithm, the SGD Classifier performs the best (77.1%). There is a slightly different performance between SGD and Logistic Regression.
- Using the double preprocessing method: From TABLE 6, Fig.5, 6 and 7, performance results were best achieved by SGD Classifier. In addition, Naïve Bayes achieved the poorest performance.
- Furthermore, although the features were reduced in the double preprocessing method, good performance results could still be achieved by performing feature selection.
- We also noticed that the neutral class gives unsatisfying performance, due to the structure of the neutral comment and the lack of neutral features that could train the model in a correct way.

## 5. Conclusion and future work

In this work, we conducted an empirical study of sentiment Analysis of online reviews written in Modern Standard Arabic and then built a System called SSAAR (System for Sentiment Analysis of Arabic Reviews). To obtain a performing system, we by ourselves construct the SPPARF Framework which is specific scraper and preprocessing framework adapted for MSA Arabic data collection and advanced preprocessing, used as an input framework for our proposed system. In fact, SPPARF was used to collect and preprocess MSA Arabic text, which consists of Arabic comments collected from Booking website and Aljazeera blog.

Secondly, we tested and compared between two methods to feed our learning algorithms. The first one is the double preprocessing method, and the second one is the TF-IDF method based on feature selection method. We found out that in both methods, SGD Classifier exhibits the best performance for sentiment classification for both positive and negative class but not for the neutral class.

Thirdly, the experimental results indicate that the larger our dataset is the best the learning algorithm performs; it is the case for positive and negative classified data verses (VS) the neutral ones.

Finally, we notice that sentiment classifiers are severely dependent on the type, distribution and size of features and dataset. We also conclude that improving the feature selection approach could automatically improve the classification precision and efficiency.

With the consideration of conclusion, our future effort is to investigate in the points below:

→ The influence of comments length on the learning process to train a sentiment classifier.

→ Combining the two proposed methods to get better results

→ Training our model using Cross Validation

→ Developing a web tool with a user interface based on our system SSAAR allowing sentiment analysis of online Reviews on different websites and Blogs.

## 6. Conflicts of Interest

The authors declare no conflicts of interest associated with this manuscript.

## 7. References

1. Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Now Publishers. https://doi.org/10.1561/9781601981516
2. Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. https://doi.org/10.3115/1118693.1118704
3. Turney, P. D. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. https://doi.org/10.3115/1073083.1073153
4. Wang, Y. Z., Zheng, X., Hou, D., & Hu, W. (2018). Short text sentiment classification of high dimensional hybrid feature based on SVM. Comput. Technol. Develop., 28(2), 88-93.
5. Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. Computational linguistics, 37(2), 267-307. https://doi.org/10.1162/COLI_a_00049
6. Kiritchenko, S., Zhu, X., & Mohammad, S. M. (2014). Sentiment analysis of short informal texts. Journal of Artificial Intelligence Research, 50, 723-762. https://doi.org/10.1613/jair.4272
7. El-Halees, A. M. (2011). Arabic opinion mining using combined classification approach. Arabic opinion mining using combined classification approach.
8. Abdul-Mageed, M., Diab, M., & Korayem, M. (2011, June). Subjectivity and sentiment analysis of modern standard Arabic. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (pp. 587-591).
9. Abdul-Mageed, M., Diab, M., & Kübler, S. (2014). SAMAR: Subjectivity and sentiment analysis for Arabic social media. Computer Speech & Language, 28(1), 20-37. https://doi.org/10.1016/j.csl.2013.03.001
10. Ahmad, K., & Almas, Y. (2005, July). Visualising sentiments in financial texts?. In Ninth International Conference on Information Visualisation (IV'05) (pp. 363-368). IEEE.
11. Nabil, M., Aly, M., & Atiya, A. (2015, September). Astd: Arabic sentiment tweets dataset. In Proceedings of the 2015 conference on empirical methods in natural language processing (pp. 2515-2519).. https://doi.org/10.18653/v1/D15-1299
12. Perkins, J. (2010). Python text processing with NLTK 2.0 cookbook. Packt Publishing Ltd.
13. Nejjari, M., & Meziane, A. (2019, March). Overview of Opinion Detection Approaches in Arabic. In Proceedings of the 2nd International Conference on Networking, Information Systems & Security (pp. 1-5). https://doi.org/10.1145/3320326.3320410
14. Haddi, E., Liu, X., & Shi, Y. (2013). The role of text preprocessing in sentiment analysis. Procedia Computer Science, 17, 26-32. https://doi.org/10.1016/j.procs.2013.05.005
15. O’Keefe, T., & Koprinska, I. (2009, December). Feature selection and weighting methods in sentiment analysis. In Proceedings of the 14th Australasian document computing symposium, Sydney (pp. 67-74).
16. Lewis, D. D. (1998, April). Naive (Bayes) at forty: The independence assumption in information retrieval. In European conference on machine learning (pp. 4-15). Springer, Berlin, Heidelberg. https://doi.org/10.1007/BFb0026666
17. Boulaiche, A., & Adi, K. (2018). An auto-learning approach for network intrusion detection. Telecommunication Systems, 68(2), 277-294. https://doi.org/10.1007/s11235-017-0395-z
18. Blair, D. C. (1979). Information Retrieval, CJ Van Rijsbergen. London: Butterworths; 1979: 208 pp. Price: $32.50. Journal of the American Society for Information Science, 30(6), 374-375. https://doi.org/10.1002/asi.4630300621

19. Alotaibi, S. S. (2015). Sentiment analysis in the Arabic language using machine learning. 2000-2019-CSU Theses and Dissertations.