IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 12, No. 1, March 2023, pp. 284~294
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i1.pp284-294

A hybrid composite features based sentence level sentiment analyzer

Mohammed Maree[1], Mujahed Eleyat[2], Shatha Rabayah[3], Mohammed Belkhatir[4]
[1]Department of Information Technology, Faculty of Engineering and Information Technology, Arab American University, Jenin, Palestine
[2]Department of Computer Systems Engineering, Faculty of Engineering and Information Technology, Arab American University, Jenin, Palestine
[3]Department of Computer Science, Faculty of Engineering and Information Technology, Arab American University, Jenin, Palestine
[4]Department of Computer Science, Campus de la Doua, University of Lyon, Lyon, France

Article Info

Article history:

Received Dec 3, 2021
Revised Aug 25, 2022
Accepted Sep 23, 2022

Keywords:

Composite features
Experimental evaluation
Extrinsic semantic resources
Natural language processing pipelines
Sentiment classification

ABSTRACT

Current lexica and machine learning based sentiment analysis approaches still suffer from a two-fold limitation. First, manual lexicon construction and machine training is time consuming and error-prone. Second, the prediction's accuracy entails sentences and their corresponding training text should fall under the same domain. In this article, we experimentally evaluate four sentiment classifiers, namely support vector machines (SVMs), Naive Bayes (NB), logistic regression (LR) and random forest (RF). We quantify the quality of each of these models using three real-world datasets that comprise 50,000 movie reviews, 10,662 sentences, and 300 generic movie reviews. Specifically, we study the impact of a variety of natural language processing (NLP) pipelines on the quality of the predicted sentiment orientations. Additionally, we measure the impact of incorporating lexical semantic knowledge captured by WordNet on expanding original words in sentences. Findings demonstrate that the utilizing different NLP pipelines and semantic relationships impacts the quality of the sentiment analyzers. In particular, results indicate that coupling lemmatization and knowledge-based n-gram features proved to produce higher accuracy results. With this coupling, the accuracy of the SVM classifier has improved to 90.43%, while it was 86.83%, 90.11%, 86.20%, respectively using the three other classifiers.

Corresponding Author:

Mohammed Maree
Department of Information Technology, Faculty of Engineering and Information Technology
Arab American University
P.O. Box 240 Jenin, 13 Zababdeh 00970-4-2418888 ext. 1123, Palestine
Email: mohammed.maree@aaup.edu

1. INTRODUCTION

Sentiment analysis (SA) has become one of the most reliable tools for assisting organizations better understand the perception of their users about the products and services that they offer [1], [2]. Exploiting SA techniques in real-world application domains also serves individuals who are interested in learning more about the various customer perceptions on the products and/or services of interest. Recently, there has been a growing number of SA techniques which can be characterized by a number of strengths and weaknesses; demonstrated by the quality of their prediction accuracy. Lexicon-based and machine learning approaches have been among the most common techniques for this purpose. As far as lexicon-based approaches are concerned, a lexicon containing a set of words with their polarities, such as SentiWordNet is employed to

predict the sentiment orientation of word mentions in sentences. However, depending on a lexicon alone is not sufficient due to the fact that a word's prior polarity doesn't reflect its contextual polarity in a sentence [3]. In addition, various semantic relations and axioms should be incorporated; requiring to explicitly add such entities. On the other hand, machine learning methods rely on training samples to predict the polarity of opinions. This normally involves supervised learning, wherein a group of sentences is classified into several labelled classes, such as positive, neutral and negative. Although this approach can detect polarity of sentences; however, it is time-consuming and can only analyze the sentiments of texts which belong to the same field of the training samples. In addition, the size of the training samples should be large enough in order for this approach to yield good results. Starting from this position, we study the main features that characterize these techniques and evaluate their effectiveness using large-scale real-world datasets that comprise sentiment reviews. We focus on the feature engineering process and explore the impact of using semantic and taxonomic relationships, as well as different priorities for word types (Noun (NN), Adjective (ADJ), Adverb (ADV), Verb (VV)) to perform sentence-level SA. The main contributions of our proposition are summarized:

- Explore existing SA techniques and evaluate their effectiveness, as well as efficiency using three publicly-available datasets.
- Exploiting knowledge triplets encoded in generic knowledge repositories and measuring their impact on the quality of the utilized SA models.

The rest of this paper is organized into four sections. In section 2, we introduce the research method and discuss the related works. We also provide the theoretical details and algorithms used in the proposed natural language processing (NLP). In section 3, we introduce the experimental evaluation steps that we carried out to evaluate a variety of SA pipelines. We also compare our proposed SA model with state-of-the-art techniques and discuss the findings accordingly. In section 4, we present the conclusions and highlight the future directions of our proposed work.

2. RESEARCH METHOD

Sentiment Analysis can be defined as the practice of employing NLP and text processing techniques to identify the sentiment orientation of text [4]. Researchers exploit various NLP techniques; coupling text analysis, in addition to using extrinsic resources to assist in identifying the semantic orientation of sentences. Among these techniques are: (i) Lexicon-based and (ii) Machine learning approaches [5]–[10], [11]–[16]. For instance, in [5], the authors compared between support vector machine (SVM) and artificial neural networks (ANNs). Both models were utilized to classify the polarity of texts at document level. As reported by the authors, the SVM has demonstrated to be less effective than ANNs, especially when using unbalanced data contexts. Nevertheless, experiments showed some limitations, specifically in terms of the high computational cost required for processing and analyzing large-size documents. Therefore, as reported in Reference [6], working on document level is expensive and inaccurate in some scenarios. This is mainly because the whole document is treated as a single unit and the entire content of the document is classified as positive or negative towards a specific issue. This problem also appears when working at the paragraph level, as a paragraph may contain two sentences with different sentiment polarities. Therefore, other researchers have focused on SA at sentence level [7], [8]. In Reference [7], Pak and Paroubek analyzed a dataset of financial reports at sentence level using multiple sentence embedding models. They used models based on Siamese CBow and fastText. However, as reported by Basha and Rajput in [9], using a lexicon with predefined term polarities doesn't reflect any contextual sensitive polarities at the sentence level. To address this issue, Meškelė and Frasincar proposed a hybrid model called ALDONA [10], that combines lexicon based and machine learning approaches. The results showed more accurate results than state-of-the-art models, namely when dealing with sentences with complex structures. In the same line of research, we can also find some researchers who analyzed the sentiment polarity at a word level [11]. For example, Chen *et al.* utilized the reinforcement learning (RL) method and combined it with a long-short term memory (LSTM) model to extract word polarities. This approach succeeded in correctly determining the polarity of the word in most cases but it also failed in some scenarios. When exploring the related works, we can find out that main factor that has a major impact on the quality of SA results is the feature selection as reposted in [12], where the authors compared three different features selection methods, namely feature frequency (FF), term frequency – inverse document frequency (TF-IDF), and feature presence (FP). The Chi-square method was utilized to extract and filter the most relevant features, and the SVM classifier was used to evaluate the system's performance. Results confirmed that the SVM classifier's performance varied significantly depending on input features. In a similar work proposed by Alam and Yao [13], the authors developed a methodology for comparing the performance of different machine learning classifiers (SVM, Naive Bayes (NB), and maximum entropy (MaxE)). The authors used different NLP techniques (stemming, stopwords removal and emoticons removal) to extract significant features. They also used sets of n-gram token types (Uni-grams, Bi-grams, Uni-grams

and Bi-grams, Uni-gram with part of speech tags), in addition to using Word2vec technique for word vectorization, in an attempt to increase the classification accuracy. The results confirmed that the NLP techniques had a significant impact on the quality of the employed sentiment analyzers. Results confirmed that the NB classifier has outperformed the SVM and MaxE classifiers, however, it is not clear which combination of features should be used in what scenarios i.e., sentiment classification tasks. In a recent work proposed in Reference [14] by Sohrabi and Hemmatian, Fatemeh, the authors used a dataset of social media posts obtained from Twitter to conduct a comparative study between several machine learning SA techniques (decision tree, neural network, SVM, and W-Bayes network) at sentence level. The authors used both the RapidMiner software package and Python programing language to make this comparison. In the proposed methodology, researchers used a series of NLP techniques, including normalization, removing stopwords, and tokenization. For feature weighting, the authors used Word2vec and TF-IDF methods. The results confirmed significant variations in the accuracy of the utilized SAs after using the proposed NLP techniques. In [15], Krouska proposed a methodology to apply SA on three well known datasets, namely Obama-McCain debate (OMD), health care reform (HCR) and Stanford twitter sentiment gold standard (STS-Gold). Researchers compared several classification techniques using a variety of NLP techniques, including removing stop words, stimming, and n-gram tokenization. Results confirmed that n-grams captured by extrinsic resources has proved to improve the quality of the SAs.

2.1. Proposed pipeline and algorithm

The process of identifying the sentiment orientation of a given sentence passes through several phases. The first phase starts with data collection, cleaning and preparation. The second phase aim at feature selection and extraction from processes texts. The third phase focuses on training the sentiment classifier and testing its quality. In the next sections, we introduce the details of each of these phases.

2.1.1. Data acquisition and cleansing

Sources of sentiment sentences can vary on the Web. Twitter and other social media websites are among the most commonly referred to sources [2]. For experimental evaluation purposes, we use the well-known internet movie database (IMDB) movie reviews dataset, which is publicly available at Kaggle. In addition, we collected the 10,662 LightSide dataset, and 300 other generic movie reviews from Twitter. The first task after the data acquisition is to clean the raw data as it normally contains some special characters, including hashtags, consecutive white spaces, URLs, and unnecessary symbols. In addition, there is a set of emoticons that we cleaned using a pre-defined set of icon representations. After this step, we proceed to the second phase of data processing, that is tokenization and feature extraction.

2.1.2. Tokenization and feature extraction

At this stage, a set of features is extracted from textual information encoded in sentiment sentences. As we discussed in the literature review section, there are several SA models that employ different feature extraction techniques. For instance, considering the n-gram tokenization process, choosing a specific size for n-gram tokens affects the overall's accuracy of the sentiment analyzer, especially when dealing with sentences contain negations, such as "not good" or "not very good" as reported in [2], [16]. In this context, we can see that using unigram features to process a sentence with "not good", will separate the word "good" from the negation word "not". Similarly, using the same bigram features in the second scenario, the negation word "not" and the word "good" will be separated into two different tokens, these are: "not very" and "very good". In terms of computational cost, if the dictionary size when using unigrams is |D|, it will become |D|2 when using bigrams, and |D|3 for trigrams. Therefore, using n-grams with n greater than 3 is very time consuming. In an earlier study in 2004, Pang and Lee has showed that unigram features are more significant than bigrams when performing emotional classification for movie reviews [17]. On the other hand, other studies showed that coupling bigrams and trigrams based on extrinsic semantic resources provided better results than unigrams alone [2], [18]. As reported by Maree and Eleyat in [2], incorporating high-order n-grams such as trigrams that can be acquired based on external knowledge resources captures a sentence contextual information more precisely than other unigram or bigram-based tokenizers. Accordingly, and in light of these conclusions, it is crucial to experimentally evaluate the utilization of n-gram tokenization as part of the large-scale SA process. In the next sections, we provide details concerning each of the proposed NLP pipeline phase.

2.1.3. Stopwords removal, word stemming and lemmatization

Raw text of sentiment sentences usually contains a lot of words that are frequently repeated. Such words have no significant contribution in determining the sentiment orientation of sentences. Moreover, they may have a negative impact on determining the polarity of sentences as reported in [3]. These words are

referred to as stopwords. Removing such words is a crucial step in the data pre-processing phase. Some researchers divided stopwords into two types, these are: 1) general stopwords and 2) domain-specific stopwords [16]. General stopwords such as (the, a, on, of, …) are considered as stopwords regardless of the domain of the dataset. On the other hand, domain-specific stopwords are those words that appear to be of little significance in deriving the meaning of a given sentence in a particular domain [19]. Such words can be identified through the utilization of the tf.idf model in the same manner as described in [19]. For example, as far as the movie reviews dataset is concerned, words such as, movie, actor, and film appear very frequently in the sentiment sentences. These words are specific stopwords to this field and can be regarded as stopwords when they appear redundantly in sentences [18]. Stemming, on the other hand, is one of the common morphological analysis processes that is mainly concerned with the removal of derivational affixes in the hope of achieving a common base form for a given word. The goal of this process is to reduce inflectional forms and achieve a common base form for words in sentiment sentences. As reported by Haddi *et al*. the importance of this step lies in reducing text dimensions for sentiment classifiers [12]. In this context, the number of dimensions representing different words in the text will be reduced. As such, this allows representing a word, in addition to its inflectional forms, as one word. For instances, the words: product, producer, production and produce will be reduced to produce [6]. This reduction in word dimensions helps also to correctly determine the weights of the words and their importance in the text. In our work, we have exploited and evaluated two of the most common stemming techniques that are employed in the context of sentiment analysis. These are: Porter [20], [21] and Snowball stemmers. Porter stemmer is one of the oldest and most popular stemming algorithms that is still utilized as part of various sentiment analysis pipelines [22], [23]. It is a rule-based stemmer that consists of five phases of word reductions that are applied sequentially. Each rule group is applied to the longest suffix. One of main limitations of this stemmer is that it may return the same form for two words with different sentiment polarities. For instance, the word "Dependability" which has a positive polarity and the word "Dependent" which has a negative polarity are both stemmed to "Depend". Similarly, Snowball stemmer, which was developed based on Porter's stemmer, shares many of the limitations that are inherent in Porter stemmer. However, this technique has demonstrated to produce more promising stemming results against those produced by Porter. In addition, as reported by Rao in [24], the performance of Snowball is higher than Porter and it can be applied on many languages other than English. Similar to stemming, lemmatization is another common morphological analysis process that has been widely utilized for sentiment analysis purposes. It is mainly used to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. One of the main advantages of lemmatization is that a returned lemma can be further processed to obtain synonyms, as well as other semantically-related terms from extrinsic semantic resources such as WordNet and Yago [25].

2.2. Features selection and sentiment classification techniques

Several research works have focused on feature selection from text, namely for sentiment analysis purposes [2], [10], [26], [27]. The main features that have captured the focus of these works are term frequency, inverse document frequency, n-gram based tokens, and POS-tags of words extracted from the text of sentiment sentences. The aim of the feature selection process in this context is to reduce feature vectors in an attempt to improve the computation speed and increase the accuracy of the sentiment classification methods. However, little attention has been given to the latent semantic aspects of words encoded in sentiment sentences [28], [29]. In addition, the composition of the best features that significantly contribute to producing the most accurate sentiment classes has not been given sufficient attention. Accordingly, in the context of our research work, we attempt to investigate the impact on combining multiple features, including semantically-related features, on the quality of a number of machines learning based sentiment classification methods. In particular, we employ the well-known TF-IDF feature weighting technique with the combination of n-gram tokens and semantic features extracted from WordNet. As such, a sentiment sentence is assigned a weight based on (1). The unit we use to represent each sentence is D. We use the variable D to denote sentences in the dataset to maintain consistency with the general terms used to define the TF-IDF equation.

$$\text{TF-IDF}(t) = (1+ \log10\ \text{TF}(t)) * \log10\ (N/\text{DF}(t)), \quad (1)$$

Where:

- N: is the number of sentiment sentences in the corpus.
- DF: represents the number of sentences that contain term t.
- TF: is the number of occurrences of term t in sentence D.

It is important to point out here that the TF-IDF feature weighting technique is applied on composite features extracted from pre-processed text of sentiment sentences. A wide range of machine learning (supervised, semi-supervised and un-supervised) techniques have been developed and utilized for sentiment

analysis purposes [29], [30]. The input to any of the utilized methods is a set of composite features generated based on the techniques discussed in the previous section. As such, the quality of the utilized sentiment classifier will be highly dependent on the quality of the selected features, especially when dealing with highly skewed datasets. Traditionally, there are various types of classifiers that have been commonly used for sentiment analysis. These are: 1) SVM, NB, random forest (RF) and logistic regression (LR). It is important to point out that these classifiers are considered among the most commonly used and robust classification methods that have been successfully applied for sentiment analysis as reported in Reference [6]. They have demonstrated highly accurate classification of sentiment sentences in various application domains [28]. In addition, they tend to be less affected by noisy terms, especially when compared with ANN-based sentiment classifiers when the data imbalance increases. Furthermore, the time required to train an ANN is usually much higher than the that required for training these types of classifiers.

Figure 1 shows the phases of the proposed sentiment analysis pipeline. First, we read the dataset containing sentences associated with their sentiment classes. Then, we perform data cleansing by removing special characters, unwanted signs, case folding and removing numbers. After this step, we apply text tokenization using two types of tokenizers *Wordpunct_Tokenizer* and *Casual_Tokenizer*. Then, the produced tokens are stemmed using two types of stemmers, Porter stemmer and Snowball stemmer. Lemmatization is also another step that we apply on the tokens using WordNet lemmatizer, WordNet knowledge-based n-gram recognition, and synonyms-based enrichment techniques, noting that we use synonyms of nouns and adjectives for term enrichment purposes.

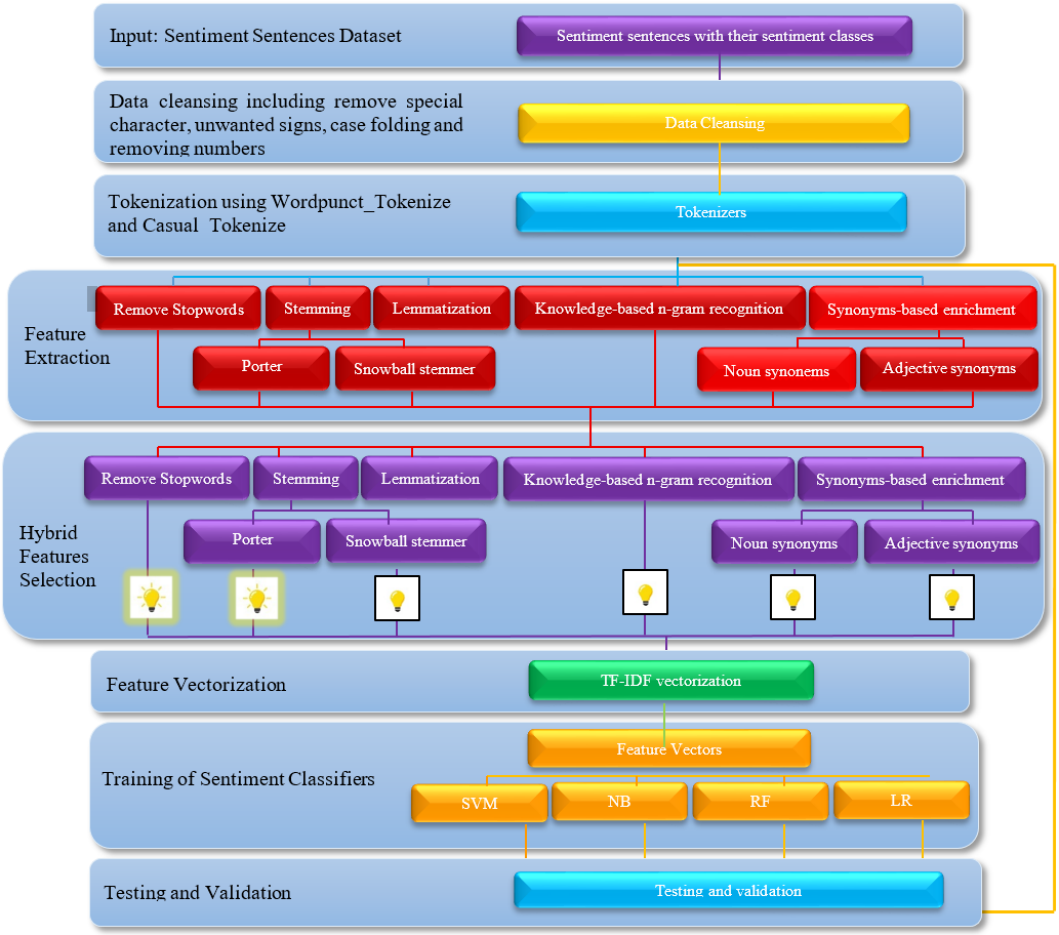


Figure 1. Phases of the proposed sentiment analysis pipeline

Then, in the next step, we select and activate a hybrid set of features for tf-idf vectorization. We use the produced vectorization results for training each of the sentiment classifiers. The details of these step are presented in Algorithms 1. First, using Algorithm1, we perform data cleansing (steps 2 to 6). Step 10 presents

the tokenization function. In step 11, we invoke the feature selection and activation function. Using this function, we select and activate specific feature extraction functions, such as stemming, lemmatization, and n-gram recognition. to compose the hybrid features set. In step 13, we submit the composite features for tf-idf vectorization. Next, at step 18 we train the classifier and at step 19 we the algorithm carries out the testing and validation task and calculates accuracy of the results.

Algorithm 1. Identifying the sentiment orientation algorithm

INPUT: Sentiment sentences with their polarity. S < Stopwords >. And list of unwanted signs (%,#,%,|), L < unwanted signs >
OUTPUT: list of sentiment orientation predictions for test set reviews and algorithm accuracy.
PREDICTSENTIMENT(R,S,L) : sentiment orientation, algorithm accuracy
1: reviews ← R <r_i, p_i> // list of sentiment sentences with their sentiment polarity
2: while reviews ≠ NIL do:
3: for each word in r_i do
4: do **DELETE** (word) if word ∈ (L or numbers) .
5: word ← **CASEFOLDING** (word)
6: end for
7: F ← < > // F: list of features
8: while Next r_i in reviews do
9: T ← < >// T: list of tokens for r_i
10: T ← **TOKENIZE** (r_i) // tokenization using Wordpunct_Tokenize or Casual_Tokenize
//Next function (Step 11) constructs composite feature using the helper functions that start at step 21.
11: F ← **FEATURESELECTIONANDACTIVATION**(T)
12: class ← < p_i>
13: vectoriser ← **TFIDFVECTORIZER**(F)
14: TrainSet ← 0,70 ×(vectoriser, class) //Split vectors in to TrainSet and TestSet
15: TestSet ← **REST** (vectoriser, class)
16: trained classifer← **TRAINCLASSIFER**(TrainSet)
17: P ← < > // P list of sentiment orientation predictions for test set reviews
18: P ← trained classifer. predict (TestSet)
19: accuracy ←**CALCULATEACCURICY** (P, TestSet)
20: return P, accuracy

Next, for identifying the sentiment orientation we SentiWordNet lexicon. In particular, we use this lexicon to find SentiWordNet scores for every token in each review. We accumulate positive scores and negative scores per token. We use the accumulated scores to determine sentence sentiment orientations and produce predictions for the used datasets associated with their accuracy metrics. Our goal of using SentiWordNet in this context is to find out how a sentiment classifier's accuracy can be affected by the incorporation of prior term polarity information.

3. RESULTS AND DISCUSSION

To perform the experiments and develop the proposed NLP pipeline, we used Python program language. This language was also utilized for pre-processing the three publicly-available datasets that are presented in Table 1. Before proceeding with the discussion on the conducted experiments, we present some details about the used datasets.

Table 1. Statistics about the used sentiment review datasets

Dataset	Sentiment Sentences	Positive	Negative	Average Unigrams Per Review	Average Unigrams & Bi-grams Per Review	Average Unigrams without Stopwords Per Review
IMDB	50.000	25.000	25.000	270.82	261.4352	148.5876
Sentiment Sentences from Reference [2]	10.662	5.331	5.3331	22.71264	21.09316	13.75654
LightSide's Movie Reviews	3000	150	150	698.5533	674.4867	396.12

In our experiments, we classified the reviews using four machine learning algorithms. These are: SVMs, NB, LR, and RF classifiers. And we have used accuracy metric to compare between these classifiers. Accuracy in this context is the ratio of correctly classified reviews to the total reviews. (2) explains how to calculate accuracy, where a sentiment prediction is classified into true positive (TP), true negative (TN), false positive (FP) and false negative (FN).

$$\text{accuracy}=\frac{\text{number of correctly classified reviews}}{\text{number of reviews}}=\frac{\text{tp+tn}}{\text{tp+tn+fp+fn}} \quad (2)$$

We also used the natural language toolkit (NLTK) to apply some NLP techniques. Using the NLTK library, we employed two types of tokenizers namely: wordpunct and casual tokenizers. We have also used uni-grams features and n-grams features (uni-grams and bi-grams). Moreover, we have updated NLTK stopword list to filter out sentences. It is important to point out that we removed all negation words from the predefined stopwords list. This is because such words have an important impact on identified correct sentiment orientations. We also enriched the list with additional words that appear frequently in the datasets (domain-specific stopwords). Example of such words are: story, movies, watching, acting, character, and film. Accordingly, we exploited a number of NLP pipelines to study the impact of each pipeline phases on the quality of each classifier. Tables 2-3 illustrate the variations on the accuracy results when utilizing each pipeline.

Table 2. Experimental results using Wordpunct_Tokenize

	Datasets	SVM	NB	LG	RF
Original Text	IMDB	90.27%	85.96%	89.84%	84.42%
	Sentiment_sentences	76.52%	77.61%	75.52%	68.23%
	Movie Review	70.00%	52.22%	67.78%	75.56
Remove unwanted symbols and numbers	IMDB	90.20%	85.87%	89.71%	84.38%
	Sentiment_sentences	76.64%	77.55%	75.45%	68.95%
	Movie Review	71.11%	52.22%	67.78%	68.89%
Stopwords Removal	IMDB	89.81%	86.61%	89.77%	86.22%
	Sentiment_sentences	74.64%	76.27%	74.68%	68.11%
	Movie Review	64.44%	56.67%	61.11%	68.89%
Porter stemmer	IMDB	89.95%	85.25%	89.61%	84.08%
	Sentiment_sentences	77.49%	77.11%	76.17%	69.92%
	Movie Review	70.00%	52.22%	63.33%	72.22%
Porter stemmer + Stopwords Removal	IMDB	89.51%	85.66%	89.37%	85.37%
	Sentiment_sentences	75.23%	76.11%	74.36%	69.45%
	Movie Review	64.44%	56.67%	64.44%	65.56%
Snowball stemmer	IMDB	89.90%	85.24%	89.60%	84.61%
	Sentiment_sentences	78.08%	77.64%	76.49%	70.33%
	Movie Review	71.11%	52.22%	65.56%	75.56%
Snowball stemmer + Stopwords Removal	IMDB	89.44%	85.57%	89.18%	85.12%
	Sentiment_sentences	76.14%	76.64%	75.20%	69.54%
	Movie Review	63.33%	57.78%	63.33%	58.89%
Lemmatizer	IMDB	90.07%	85.63%	89.74%	84.40%
	Sentiment_sentences	76.83%	77.08%	75.52%	69.14%
	Movie Review	71.11%	51.11%	66.67%	72.22%
Lemmatizer + Stopwords Removal	IMDB	89.63%	86.37%	89.51%	85.85%
	Sentiment_sentences	74.89%	76.17%	74.14%	68.51%
	Movie Review	66.67%	55.56%	64.44%	56.67%
Uni-gram &Bi-gram	IMDB	90.39%	86.01%	89.97%	84.88%
	Sentiment_sentences	77.11%	77.55%	75.08%	68.36%
	Movie Review	71.11%	52.22%	65.56%	67.78%
Uni-gram &Bi-gram Stopwords Removal	IMDB	90.17%	86.80%	90.02%	86.27%
	Sentiment_sentences	74.86%	76.95%	74.73%	68.76%
	Movie Review	64.44%	56.67%	61.11%	73.33%
Uni-gram &Bi-gram + Porter stemmer	IMDB	90.15%	85.51%	89.79%	84.30%
	Sentiment_sentences	77.20%	77.24%	76.05%	69.95%
	Movie Review	68.89%	52.222%	66.67%	71.11%
Uni-gram &Bi-gram + Snowball stemmer	IMDB	90.09%	85.35%	89.73%	84.68%
	Sentiment_sentences	77.49%	77.77%	76.49%	69.45%
	Movie Review	68.89%	52.22%	66.67%	77.78%
Uni-gram &Bi-gram + Lemmatizer	IMDB	90.43%	85.99%	89.95%	84.75%
	Sentiment_sentences	76.67%	77.39%	74.92%	69.04%
	Movie Review	70.00%	51.11%	64.44%	72.22%
Uni-gram + adj synonyms	IMDB	90.19%	86.59%	89.98%	84.45%
	Sentiment_sentences	76.45%	77.17%	76.08%	69.23%
	Movie Review	72.22%	54.44%	70.00%	73.33%
Uni-gram + Noun synonyms	IMDB	90.07%	84.75%	89.24%	84.01
	Sentiment_sentences	74.80%	75.30%	73.89%	69.48%
	Movie Review	64.44%	52.22%	62.22%	74.44%
Uni-gram, Bi-grams +Noun synonyms + lemmatization	IMDB	89.99%	84.99%	89.33%	84.28%
	Sentiment_sentences	74.67%	75.39%	73.42%	68.76%
	Movie Review	65.56%	51.11%	65.56%	71.11%
Uni-gram, Bi-grams +Adj synonems + lemmatization	IMDB	90.30%	86.83%	90.11%	84.95%
	Sentiment_sentences	76.08%	77.55%	75.30%	68.54%
	Movie Review	74.44%	51.11%	68.89%	65.56%
Uni-gram, Bi-grams + Adj synonyms	IMDB	90.33%	86.77%	90.01%	85.15%
	Sentiment_sentences	76.42%	77.05%	75.92%	68.89%
	Movie Review	73.33%	52.22%	67.78%	82.22%

Table 3. Experiments results using lexicon-based sentiment analysis

Dataset	SentiWordNet	Wordpunct Tokenize	Casual Tokenize
IMDB	65.00%	90.40%	90.50%
Sentiment Sentences	58.00%	78.08%	77.64%
Movie Review	59.00%	82.22%	75.56%

In our experiments we have used 19 different combinations of NLP techniques. Each combination was applied twice, using Wordpunct tokenizers. Each combination of NLP techniques produced different set of features. Which means different results for the sentiment reviews polarity. Based on the results in Tables 2-3, we notice that all classifiers are positively affected when we used n-grams, snowball stemmer, sentence enrichment with adjective synonyms, lemmatization, and when we removed unwanted signs and numbers. Also, when we have a combination of them. This means that these NLP techniques have an important and positive impact in determining the polarity of reviews. Also, we note that when we do synonyms enrichment to adjectives, we obtain better results than when perform noun-based enrichment. When we used the Wordpunct_tokenizer with the IMDB dataset, we obtained the best result (90.43%). In particular, when we used SVM classifier with uni-grams and bi-grams with lemmatization. Both the NB and LR classifiers produced (86.83%) and (90.11%) accuracy results when we used n-grams feature with sentence enrichment by Adj synonyms and lemmatization. The RF classifier produced the best result (86.20%) when we used n-grams with stopwords removal. Considering the Sentiment Sentences dataset that is used by the authors of [2], we obtained the best result (78.08%) when using the SVM classifier with Snowball stemmer. Also, the RF classifier produced the best predictions (70.33%) when we used Snowball stemmer. As far as the NB and LR classifiers are concerned, they produced the best results (77.77%) and (76.49%), respectively when we using n-grams with Snowball stemmer. For the LightSide's Movie_Reviews dataset, we obtained the best result (82.22%) when we used the RF classifier with n_grams and Adjective synonyms combination. Also, the SVM classifier produced the best result (74.44%) when we used n_grams, sentence enrichment with Adjective synonyms and lemmatization. The NB classifier produced the best result (57.78%) when using Snowball stemmer with Stopwords removal. While the LR produced best result (70.00%) when we used sentence enrichment with Adjective synonyms. When using the Casual_tokenizer, we find that for the IMDB dataset, we got the best result (90.50%) with the SVM classifier when applied on the original text. On the other hand, the NB classifier produced the best result (86.91%) when using n-gram features with sentence enrichment by Adj synonyms and lemmatization. Both the LR and RF classifiers produced the best results (90.06%) and (86.13%) when we used n-grams with stopwords removal. In Sentiment_sentences dataset We got the best result (77.64%) when with the SVM classifier when employing the Snowball stemmer. Also, both the NB and LR classifiers produced the best results (77.58%) and (76.52%), respectively when we used Snowball stemmer. For the RF classifier, it produced the best result (69.95%) when we used the Snowball stemmer with Stopwords removal. For the LightSide's Movie_Reviews dataset, we obtained the best result (75.56%) when using the RF classifier with n_grams and sentence enrichment with Adjective synonyms. Also, the LR classifier produced the best result (70.00%) when we used n_grams with sentence enrichment with Adjective synonyms. On the other hand, the NB classifier produced the best result (58.89%) when we used Stopwords removal. While the SVM classifier produced the best result (70.00%) when we used sentence enrichment with Adjective synonyms, lemmatization or snowball stemmer. As illustrated from the results in Table 3, we can see that the obtained results using the lexicon-based sentiment analysis technique are less precise than their counterparts. This is mainly due to the fact that this approach relies on the accumulative sum of the pre-defined polarity of review tokens. Compared to the results of the previous approaches, we note that this approach ignores a word's contextual polarity in a given review. Additionally, the latent semantic dimensions of tokens are ignored in this model.

3.1. Comparison with other SA models

In this section, we present a comparison between the results that we obtained when using the IMDB dataset with respect to similar previous works, namely [26], [31]. The researchers used a group of supervised learning algorithms such as the SVM, NB, K-nearest neighbor (KNN) and Maximum Entropy. The results obtained by the researchers are shown in Table 4. As shown in this table, the Maximum Entropy classifier proved to outperform the rest of the classifiers as it produced the highest accuracy result which is 83.93%. It was followed by the SVM classifier with an accuracy of 67.60%. The authors of [26] proposed models to classify the IMDB reviews by using six layers in neural network architecture, and they utilized word weights to predict sentiment classes. In their work, the authors used two methods to extract the word polarity (positive or negative). In particular, they used two manually-constructed lists of pre-defined positive and negative words. In addition, they created word ranks by calculating sentiment and measuring domain relevance parameters. The researchers obtained a training accuracy of 91.90% and an accuracy of 86.67% for

validation. In References [32] Sahu and Ahuja employed SentiWordNet and n-grams feature selection to perform feature extraction and ranking. A group of supervised learning algorithms were used where the best accuracy achieved was 88.95% when using the RF classifier. In [27], [33], [34], the authors focused on using a convolutional neural network (CNN) and long short-term memory (LSTM) to examine sentiment polarity. The best sentiment class prediction accuracy that was obtained using this model was 89.50%.

Table 4. Comparison with existing SA models

System	Employed Classifier	Accuracy
Our Result	SVM	90.43%
Sentiment analysis on IMDB using lexicon and neural networks [26]	lexicon and neural networks	86.67%
Sentiment Analysis of IMDB Movie Reviews Using Long Short-Term Memory [27]	Long Short-Term Memory	89.90%
Sentiment analysis of movie reviews: A study on feature selection & classification algorithms [32]	RF	88.95%
Single and Multibranch CNN-Bidirectional LSTM for IMDB Sentiment Analysis [33]	Single and Multi-branch CNN-Bidirectional LSTM	89.54%
Deep CNN-LSTM with combined kernels from multiple branches for IMDB review sentiment analysis [34]	CNN-LSTM	89.50%
Sentiment Analysis on IMDB Movie Reviews Using Hybrid Feature Extraction Method [35]	Maximum Entropy	83.93%

4. CONCLUSION

Current lexicon-based and machine learning based sentiment analysis approaches are still hindered by limitations, such as the insufficient knowledge about words' prior polarities and their contextual polarities in sentences, and the manual training required for predicting the sentiment classes of sentences, which is time consuming, domain-dependent and error-prone. In this article, we experimentally evaluated four probabilistic and machine learning based sentiment classifiers, namely SVM, NB, LR and RFs. We evaluated the quality of each of these techniques in predicting the sentiment orientation using three real-world datasets that comprised a total of 60,962 sentiment sentences. Our main goal was to study the impact of a variety of NLP pipelines on the quality of the predicted sentiment orientations. We also compared that with the utilization of lexical semantic knowledge captured by WordNet. Findings demonstrate that there is an impact of varying the employed NLP pipelines and the incorporation of semantic relationships on the quality of the sentiment analyzers. In particular, we concluded that coupling lemmatization and knowledge-based n-gram features proved to produce higher accuracy rates when applied on the IMDB dataset. With this coupling, the accuracy of the SVM classifier has improved to be 90.43%. For the three other classifiers (NB, RF and LR), the quality of the sentiment classification has also improved to be 86.83%, 90.11%, 86.20%, respectively. It is important to highlight the fact that the used pipeline phases are of less complexity in terms of configuring their hyper-parameters, as well as possibly computational cost when compared with other deep learning models. This is mainly because there are normally several hyper-parameters that require manual configuration and fine-tuning one the one hand, and also the existence of multiple layers and neurons in conventional neural networks; making it require more time to be trained, configured and also for processing the input sentences and assigning labels to sentiment sentences. As such, we conclude that obtaining highly accurate sentiment prediction results can still be achieved using a composition of conventional NLP processes that are comparable and, in some cases, more superior than complex architectures. Nevertheless, in the future work, we will study the impact on varying the composite futures extracted from sentiment sentences on the quality of neural network and pre-trained text processing models. For this purpose, we will utilize the pipeline phase introduced in this article to evaluate the LSTM, Word2Vec and bidirectional encoder representations from transformers (BERT) models.

REFERENCES

[1] S. Amirmokhtar Radi and S. Shokouhyar, "Toward consumer perception of cellphones sustainability: A social media analytics," *Sustainable Production and Consumption*, vol. 25, pp. 217–233, 2021, doi: 10.1016/j.spc.2020.08.012.

[2] M. Maree and M. Eleyat, "Semantic Graph Based Term Expansion For Sentence-Level Sentiment Analysis," *International Journal of Computing*, vol. 19, no. 4, pp. 647–655, 2020, doi: 10.47839/ijc.19.4.2000.

[3] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis," *Computational Linguistics*, vol. 35, no. 3, pp. 399–433, 2009, doi: 10.1162/coli.08-012-R1-06-90.

[4] D. M. E. D. M. Hussein, "A survey on sentiment analysis challenges," *Journal of King Saud University - Engineering Sciences*, vol. 30, no. 4, pp. 330–338, 2018, doi: 10.1016/j.jksues.2016.04.002.

[5] D. M. E.-D. M. Hussein, "Analyzing Scientific Papers Based on Sentiment Analysis," *Information System Department Faculty of Computers and Information Cairo University*, 2016, doi: 10.13140/RG.2.1.2222.6328.

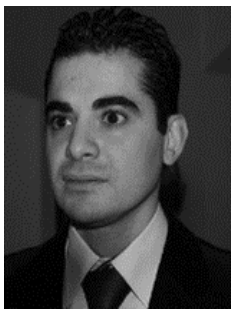
[6] E. Haddi, "Sentiment Analysis: Text Pre-Processing, Reader Views, and Cross Domains," pp. 1–133, 2015, [Online]. Available: http://bura.brunel.ac.uk/handle/2438/11196.
[7] A. Pak and P. Paroubek, "Twitter as a corpus for sentiment analysis and opinion mining," *Proceedings of the 7th International Conference on Language Resources and Evaluation, LREC 2010*, pp. 1320–1326, 2010, doi: 10.17148/ijarcce.2016.51274.
[8] C. H. Du, M. F. Tsai, and C. J. Wang, "Beyond Word-level to Sentence-level Sentiment Analysis for Financial Reports," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2019-May, pp. 1562–1566, 2019, doi: 10.1109/ICASSP.2019.8683085.
[9] S. M. Basha and D. S. Rajput, "Evaluating the impact of feature selection on overall performance of sentiment analysis," *ACM International Conference Proceeding Series*, pp. 96–102, 2017, doi: 10.1145/3176653.3176665.
[10] D. Meškelė and F. Frasincar, "ALDONAr: A hybrid solution for sentence-level aspect-based sentiment analysis using a lexicalized domain ontology and a regularized neural attention model," *Information Processing and Management*, vol. 57, no. 3, 2020, doi: 10.1016/j.ipm.2020.102211.
[11] R. Chen, Y. Zhou, L. Zhang, and X. Duan, "Word-level sentiment analysis with reinforcement learning," *IOP Conference Series: Materials Science and Engineering*, vol. 490, no. 6, 2019, doi: 10.1088/1757-899X/490/6/062063.
[12] E. Haddi, X. Liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," *Procedia Computer Science*, vol. 17, pp. 26–32, 2013, doi: 10.1016/j.procs.2013.05.005.
[13] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," *Computational and Mathematical Organization Theory*, vol. 25, no. 3, pp. 319–335, 2019, doi: 10.1007/s10588-018-9266-8.
[14] M. K. Sohrabi and F. Hemmatian, "An efficient preprocessing method for supervised sentiment analysis by converting sentences to numerical vectors: a twitter case study," *Multimedia Tools and Applications*, 2019, doi: 10.1007/s11042-019-7586-4.
[15] A. Krouska, C. Troussas, and M. Virvou, "The effect of preprocessing techniques on Twitter sentiment analysis," *IISA 2016 - 7th International Conference on Information, Intelligence, Systems and Applications*, 2016, doi: 10.1109/IISA.2016.7785373.
[16] I. Nasra and M. Maree, "On the use of Arabic stemmers to increase the recall of information retrieval systems," *ICNC-FSKD 2017 - 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery*, pp. 2462–2468, 2018, doi: 10.1109/FSKD.2017.8393161.
[17] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *Proceedings of the 42nd ACL*, pp. 271–278, 2004.
[18] M. Maree, "Semantics-based key concepts identification for documents indexing and retrieval on the web," *International Journal of Innovative Computing and Applications*, vol. 12, no. 1, pp. 1–12, 2021, doi: 10.1504/IJICA.2021.113608.
[19] M. Maree, A. Kmail, and M. Belkhatir, "Analysis & Shortcomings of E-Recruitment Systems: Towards a Semantics-based Approach Addressing Knowledge Incompleteness and Limited Domain Coverage," 2020, doi: 10.1177/0165551518811449.
[20] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980, doi: 10.1108/eb046814.
[21] M. F. Porter, "Snowball: A language for stemming algorithms," *Program*, 2001, [Online]. Available: http://snowball.tartarus.org/texts/introduction.html.
[22] T. Renault, "Sentiment analysis and machine learning in finance: a comparison of methods and models on one million messages," *Digital Finance*, vol. 2, no. 1–2, pp. 1–13, 2020, doi: 10.1007/s42521-019-00014-x.
[23] R. Patel and K. Passi, "Sentiment Analysis on Twitter Data of World Cup Soccer Tournament Using Machine Learning," *IoT*, vol. 1, no. 2, pp. 218–239, 2020, doi: 10.3390/iot1020014.
[24] A. V. S. Siva Rama Rao and P. Ranjana, "Empower good governance with public assessed schemes by improved sentiment analysis accuracy," *Electronic Government*, vol. 16, no. 1–2, pp. 118–136, 2020, doi: 10.1504/EG.2020.105252.
[25] M. Maree, R. Hodrob, M. Belkhatir, and S. M. Alhashmi, "A Knowledge-based Model for Semantic Oriented Contextual Advertising," *KSII Transactions on Internet and Information Systems*, vol. 14, no. 5, pp. 2122–2140, 2020, doi: 10.3837/tiis.2020.05.014.
[26] Z. Shaukat, A. A. Zulfiqar, C. Xiao, M. Azeem, and T. Mahmood, "Sentiment analysis on IMDB using lexicon and neural networks," *SN Applied Sciences*, vol. 2, no. 2, 2020, doi: 10.1007/s42452-019-1926-x.
[27] S. M. Qaisar, "Sentiment Analysis of IMDb Movie Reviews Using Long Short-Term Memory," *2020 2nd International Conference on Computer and Information Sciences, ICCIS 2020*, 2020, doi: 10.1109/ICCIS49240.2020.9257657.
[28] R. Moraes, J. F. Valiati, and W. P. Gavião Neto, "Document-level sentiment classification: An empirical comparison between SVM and ANN," *Expert Systems with Applications*, vol. 40, no. 2, pp. 621–633, 2013, doi: 10.1016/j.eswa.2012.07.059.
[29] S. Malviya, A. K. Tiwari, R. Srivastava, and V. Tiwari, "Machine Learning Techniques for Sentiment Analysis: A Review," *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, vol. 12, no. 02, pp. 72–78, 2020, [Online]. Available: www.ijmse.org.
[30] D. Sharma, M. Sabharwal, V. Goyal, and M. Vij, "Sentiment analysis techniques for social media data: A review," *Advances in Intelligent Systems and Computing*, vol. 1045, pp. 75–90, 2020, doi: 10.1007/978-981-15-0029-9_7.
[31] A. Abbasi, S. France, Z. Zhang, and H. Chen, "Selecting attributes for sentiment classification using feature relation networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 3, pp. 447–462, 2011, doi: 10.1109/TKDE.2010.110.
[32] T. P. Sahu and S. Ahuja, "Sentiment analysis of movie reviews: A study on feature selection and classification algorithms," *International Conference on Microelectronics, Computing and Communication, MicroCom 2016*, 2016, doi: 10.1109/MicroCom.2016.7522583.
[33] C. Vielma, A. Verma, and D. Bein, "Single and Multibranch CNN-Bidirectional LSTM for IMDb Sentiment Analysis," *Advances in Intelligent Systems and Computing*, vol. 1134, pp. 401–406, 2020, doi: 10.1007/978-3-030-43020-7_53.
[34] A. Yenter and A. Verma, "Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis," *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference, UEMCON 2017*, vol. 2018-January, pp. 540–546, 2017, doi: 10.1109/UEMCON.2017.8249013.
[35] K. Kumar, B. S. Harish, and H. K. Darshan, "Sentiment Analysis on IMDb Movie Reviews Using Hybrid Feature Extraction Method," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 5, p. 109, 2019, doi: 10.9781/ijimai.2018.12.005.

BIOGRAPHIES OF AUTHORS

Dr. Mohammed Maree received the Ph.D. degree in Information Technology from Monash University. He began his career as a R&D manager with gSoft Technology Solution, Inc. Then, he worked as the Director of research and QA with Dimensions Consulting Company. Subsequently, he joined the Faculty of Engineering and Information Technology (EIT), Arab American University, Palestine (AAUP), as a full-time Lecturer. He is currently the Assistant to VP for Academic Affairs and a Full-time lecturer at the EIT faculty at AAUP. He has published articles in various high-impact journals and conferences, such as ICTAI, Knowledge-Based Systems, and the Journal of Information Science. He is also a Committee Member and Reviewer of several conferences and journals. He has supervised a number of Master's and Ph.D. students in the fields of data analysis, information retrieval, NLP, and hybrid intelligent systems. He can be contacted at email: mohammed.maree@aaup.edu

Dr. Mujahed Elyat is an Assistant Professor of Computer Science at the Arab American University (AAUP) in Palestine. He obtained a PhD scholarship in Norway and received his PhD from Norwegian University of Science and Technology in 2014. During his PhD study, he worked as an employee in a Norwegian company called Miraim AS for three years and did research in the field of high-performance computation. Before that, he obtained a scholarship from USA, called the Presidential Scholarship, to study at the University in Arkansas where he received his master in Computer Science. In addition to teaching at AAUP for more than 10 years, Dr. Eleyat had also been the head of the department of Computer Systems Engineering for 6 years and the Assistant of the Academic VP for one year. In addition, he had been the dean of the Faculty of EIT for two years (2019-2020). He is also a member of High Performance and Embedded Architecture and Compilation (HiPEAC) and his areas of expertise include high performance computing, embedded systems, and NLP. He can be contacted at email: mujahed.eleyat@aaup.edu.

Shatha Rabayah received a B.S. degree in Computer Engineering from An-Najah National University, Nablus, Palestine, in 2014. She is currently pursuing a master's degree in computer science at the Arab American University, Palestine (AAUP). She worked as a Computer Engineering teacher in Tulkarm Industrial Secondary School. She also was a part-time lecturer at the Arab American University for one year (2019/2020). She can be contacted at email: Shatha.Rabaia@aaup.edu.



Dr. Mohammed Belkhatir received the M.Phil. and Ph.D. degrees in computer science from the University of Grenoble, France with research grants supported by the French Ministry of Research. He is currently an Associate Professor with the University of Lyon, France. He can be contacted at email: mohammed.belkhatir@univ-lyon1.fr.