Knowledge-Based Systems 227 (2021) 107232

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Enhancement of a multi-dialectal sentiment analysis system by the detection of the implied sarcastic features

Ibtissam Touahri *, Azzeddine Mazroui

Department of Computer Science, Faculty of Sciences, University Mohamed First, B-P 717, Oujda 60000, Morocco

ARTICLE INFO

Article history:
Received 25 June 2020
Received in revised form 5 January 2021
Accepted 10 June 2021
Available online 15 June 2021

Keywords:
Sentiment analysis
Irony
Sarcasm
Offensive language
Deep learning
Classical machine learning

ABSTRACT

Sentiment analysis is an NLP task that gained the interest of many researchers in various languages and recently in the Arabic language. We have encountered several challenges when dealing with this task, including sarcasm detection. In this article, we aim to exploit sarcastic characteristics to improve the accuracy of the sentiment analysis system. Sarcasm is difficult to detect because it is implicit and characterized by the presence of positive words in a negative context. We have then extracted a variety of features to define context incongruity and the opposition between the objective and subjective sentences. Offensive language and hate speech correspond to expressions that hurt others. The detection of offensive language is based on identifying offensive terms that are strongly negative and helpful to detect negative expressions. Thus, we have manually and automatically constructed sentimental, offensive and sarcastic lexicons and collected others. In the same way, many corpora either ironic (sarcastic, offensive) or sentimental (positive, negative) were collected. As sarcasm is a major challenge for the sentiment analysis system, we have built a balanced system that contains positive and negative (sarcastic, offensive) tweets. Since the analyzed corpus is multidialectal, we have used a cross dialect lexicon that retains meaning when passing from one dialect to another. Besides the Arabic dialect common characteristics, the classification was enhanced by the detection of the specificities of some dialects that use negation clitics as well as negation words to negate a term. The experiments prove that the enhancement of a sentiment analysis system by sarcastic features improved the results by 8% to reach 84.17% of accuracy using a classical machine learning approach and 80.36% using a Deep learning approach. The classical machine learning approach is improved afterward based on the expansion of the BOW lexicon and the reduction of the characteristic vector to reach an accuracy of 89.24%. This method is multilingual because the built model can be language independent. Indeed, it is enough to have the corresponding resources to apply the system to other languages.

1. Introduction

Sentiment analysis is an NLP task that aims to detect the implied sentiment within expressions to have an overall idea about people's mood in reaction to a specific subject. Sentiment analysis is applied in various fields such as political and product domains. In politics, it helps to identify supporters of political parties; moreover, it may help to predict the candidate who will be elected. In the world of products and companies, it helps to define people's opinions in reaction to the quality of a given product. Sentiment analysis gained the interest of many researchers in recent years. For the Arabic language, the study by Badaro et al. in [1] surveyed previous works in sentiment analysis, focusing on the progress of NLP tools, sentiment lexicons and corpora, classification models, and opinion mining applications.

Arabic sentiment analysis faces major challenges either linguistic or grammatical, including the absence of diacritical marks, spelling and grammar errors, code-switching besides unstandardized dialectal Arabic. Hussein in [2] cited additional challenges such as distant negation words, domain dependence, bipolar words, spam reviews and abbreviations. These challenges make sentiment feature selection a hard task. Moreover, figurative language devices such as irony can affect tweet polarity. Irony is a very broad concept that includes several notions such as sarcasm, offensive language, satire and parody. Sarcasm is used to express the opposite of what is said without using offensive vocabulary. It is characterized by the presence of opposite polarities within the same expression or the presence of positive phrases in a negative context. Hence, the opinion about a specific topic contrasts

* Correspondence to: Department of computer science faculty of sciences, University Mohamed First, Oujda, Morocco.
E-mail addresses: ibtissamtouahri555@gmail.com (I. Touahri), azze.mazroui@gmail.com (A. Mazroui).

https://doi.org/10.1016/j.knosys.2021.107232
0950-7051/

with the objective aspect. The offensive language tends to be aggressive by using hurtful terms to convey contempt or make fun of other people. It is sharp, rude and harsher than negative expressions. It may target a religion, a person, an organization or a country, which leads to hate speech and thus conflict between different parts. Hence, the spread of such languages on the web can provoke hatred and psychological effects that can destroy a whole community and cause many disastrous political and social effects.

According to the previous definitions of irony, sarcasm and offensive language, various studies have confused these notions by grouping them in one class or by considering the differences between irony and sarcasm. Ghanem et al. [3] used irony as an umbrella term that covers various figurative devices such as sarcasm, satire and parody. Farias and Rosso [4] reported that irony and sarcasm are figurative language devices with different communication purposes since sarcasm is more offensive and aggressive than irony. Whereas Joshi et al. [5] defined sarcasm as a language with opposite surface and negative intention. Rajadesingan et al. [6] considered sarcasm as a nuanced form of expressions used by individuals to state the opposite of the implied intention.

In this article, irony refers to one of the two phenomena sarcasm and offensive language. Thus, an ironic expression can be either offensive or sarcastic.

Irony complicates the sentiment analysis task. The major difficulty lies in detecting sarcasm within a positive expression that implies a negative sentiment. This task is so difficult that even humans find it difficult to accomplish. Since sarcasm is nuanced, its detection has aroused the interest of researchers due to its misleading effect on sentiment analysis. Sarcasm presence is strongly related to the expression context and cross-cultural annotation, and it may affect the polarity of the analyzed expression and so mislead the overall polarity. Hence, the presence of context is very crucial to detect this phenomenon. However, the detection of offensive expressions, whose strong negativity widens the gap between positive and negative polarities, makes it easier to classify them as negative tweets.

In this paper, we build the linguistic resources for sentiment analysis. Some of which are collected from previous works and others are constructed manually. We collect a variety of corpora from which we build various lexicons either manually or automatically. The automatic approach is based on using a custom approach to generate a bag of words. We collected from external resources other lexicons that have proved efficiency in their domain of study. However, we have improved them by correcting some labeling errors at the level of semantic terms and domains. We have incorporated and enhanced the linguistic resources by the extraction of pertinent features that address sentiment, offensive language and sarcasm to carry out the sentiment analysis of tweets. We have tested various approaches to perform classification. Therefore, as we aim to determine the pertinent features selected by our system, we have adopted a classical supervised approach that is more convenient for such a task. We first built a model using various classifiers, two of which namely Random Forest and SVM are reported since they reached pertinent results. Next, we built a deep learning method that addresses semantic terms with the Word2vec model, and preserves long-term information using a combination of LSTM and RNN. In order to perform a fair classification, we evaluate our system using different test options that are % split and Cross-validation. Selected features related to sarcasm improved the efficiency of the sentiment analysis system by achieving promising accuracy.

The reason behind our research is that sarcasm detection is still a challenge in the Arabic language especially when it comes to dealing with different dialects and writing styles. Hence, in this paper, we look for the most pertinent features that will help to detect irony within a multidialectal corpus. The main contribution of this paper lies in multi-task learning that gathers sentiment analysis, sarcasm characteristics besides dealing with multi-dialects and different writing styles. Since the paper tackles multi Arabic dialects, we have extracted their main specificities. We have performed different steps with the focus on the following:

- We have manually extracted from sentimental, sarcastic and offensive corpora a set of lexicons based on their semantics.
- We have taken advantage of foreign linguistic resources using machine translation to obtain cross lingual sentiment resources.
- We have built a Bag of Words (BOW) lexicon whose size is chosen according to custom thresholds to deal with the problems of large time required during the manual construction of the lexicon and cross lingual errors when performing the translation.
- We have addressed the Arabic dialect varieties by extracting their main features with a focus on those of the Moroccan dialect.
- We have performed an in-depth study to extract the main characteristics of sentimental, sarcastic and offensive corpora.
- We have analyzed the effect of sarcastic features on sentiment classification and extracted the best performing set.
- We have explored the effect of lexicon expansion on sentiment classification by altering the BOW size.

The innovation of this paper in comparison to the existing literature is that our study deals with the aforementioned points and combines them in one system.

Our paper is organized as follows. We define related works in the next section, and then we collect linguistic resources, namely lexicons and corpora for the sentiment analysis task. The resources besides other features will be used afterward to enhance a sentiment analysis system by the detection of sarcastic features. Our system will be evaluated then using a variety of approaches, test options and metrics. The obtained results will be compared with other systems to prove the efficiency of the followed approach.

2. Related works

We focus on irony feature extraction and sentiment analysis; hence, we give related works to these two tasks.

Affective computing and sentiment analysis

Cambria [7] considered affective computing and sentiment analysis as a subcomponent technology for other systems since they enhance customer relationship management capabilities. Han et al. [8] provided a comprehensive overview of the application of adversarial training to affective computing and sentiment analysis. They presented emotional artificial intelligence (AI) systems challenges and highlighted future research directions. Cambria et al. [9] Integrated top-down and bottom-up learning via a set of symbolic and subsymbolic AI tools which they applied to text polarity detection. They integrated logical reasoning within deep learning architectures to build a new version of SenticNet [10]. Oueslati et al. [11] carried out an in-depth study of the most important research works in Arabic sentiment analysis and discussed the strengths and limitations of existing approaches. They surveyed also machine translation and transfer learning approaches to adapt English resources to Arabic.

The first step that precedes a sentiment analysis system building is the construction of linguistic resources. In order to construct an annotated corpus for sentiment analysis, the tweets will

be labeled according to subjectivity as subjective or objective. The subjective tweets will be labeled according to polarity as positive, negative, neutral or mixed and a fine-grained polarity may be affected [12]. Then, a set of lexicons may be extracted manually or automatically from the collected corpora, or obtained following an automatic translation of a lexicon from a rich resource language to a low-resource language. The manual approach is time-consuming, however its lexicon terms are semantically verified in comparison to the automatic approach that reduces lexicon construction time and may generate cross-lingual and semantic errors. Dodds et al. [13] observed that natural human language words have a universal positivity bias when evaluating 100,000 words spread over 24 corpora relating to 10 languages of different origin and culture. Kloumann [14] reported that the human-perceived positivity towards more than 10,000 of the most frequently used English words exhibits a clear positive bias.

Lee et al. [15] used word2vec to extract place features. They constructed a place sentiment lexicon based on bag-of-words. Lexicon terms of a specific domain may change meaning in another domain depending on the context. Sense disambiguation is a challenging task for natural language processing. Wang et al. [16] aimed to construct a domain-specific sentiment lexicon by incorporating the existing lexicons and the corpus sentiment information using the TF–IDF algorithm to avoid sentimental ambiguity. Yin et al. [17] sought to resolve the lexical sentiments ambiguity. Thus, they proposed sentiment lexicon construction based on context-dependent part-of-speech chunks. The experiment results indicate that the method is balanced for positive and negative corpora. Rajabi et al. [18] presented a context-based model to solve ambiguous polarity concepts. They evaluated the proposed model by applying product reviews corpus and obtained an accuracy of 82.07%. Goularte et al. [19] proposed an approach to disambiguate word senses of short texts through the use of fuzzy lexico-semantic patterns. Orkphol and Yang [20] proposed a Word2vec method to construct a context sentence vector and give a cosine similarity score to each word sense to compute the similarity between sentence vectors. Their method gave results higher than baselines. Rudkowsky et al. [21] introduced word embedding that captures word meaning similarities for social sciences sentiment analysis.

There are many approaches to classify sentiments based on linguistic resources, among which, the unsupervised approach, the semi-supervised approach, the supervised approach and the approach based on deep learning. The unsupervised approach tags a sentimental text by the major score of the present sentimental terms. Both the supervised and deep learning approaches are based on a labeled corpus. Touahri and Mazroui [22] generated a variety of lexical resources and integrated the stem and lemma morphological notions. They addressed the negation context and evaluated the built supervised models on hotels, products, movies and restaurant reviews to reach an accuracy of 93.43%. Semi-supervised learning algorithms develop patterns with great generalizability from a limited labeled sample. Zheng and Wu [23] considered that predicting users' personality traits can be used to improve personalized service and human psychology research. They have introduced a personality analysis framework based on semi-supervised learning. They extracted linguistic features based on the n-gram model. The experimental results proved that the semi-supervised learning that takes advantage of unlabeled data helps to improve the prediction model accuracy. Han et al. [24] proposed a novel semi-supervised model based on dynamic threshold and multi-classifiers to deal with the insufficient labeled data and reduce the time to label a large amount of data. They evaluated the proposed model performance through experiments on real datasets. The proposed model has achieved the highest sentiment analysis accuracy across datasets in comparison with other existing models. Balaanand et al. [25] proposed an approach to detect fake users from Twitter data based on an enhanced graph-based semi-supervised learning algorithm. The performance of the proposed algorithm has achieved 90.3% accuracy. Li et al. [26] addressed label efficient semi-supervised learning from a graph filtering perspective which offers new insights into the classical label propagation methods and the recently popular graph convolution networks. They performed various experiments to validate their findings. Kim and Cho [27] performed experiments on the open data set from Lending Club using a semi-supervised learning method which increased accuracy by about 10% against the model that uses only labeled data.

We may consider different levels of classification. Sentimental texts or opinions can be classified as positive or negative according to a binary classification, or as positive, negative or mixed in the case of a tripolar classification, or according to a fine-grained classification that considers strongly positive, weak positive, mixed, weak negative and strongly negative polarities. Xin Ye et al. [28] proposed a novel multi-view ensemble learning method to classify microblog sentiments. The experimental results show that their method outperformed other methods in identifying polarity from microblog posts. Akhtar et al. [29] proposed a stacked ensemble method to predict the degree of emotion and sentiment intensity by combining several deep learning and feature-based model outputs that improved performance over the existing state-of-the-art systems. Yang et al. [30] reported that the analysis of online reviews can help potential users make purchase decisions and improve product and service quality. They performed a fine-grained sentiment classification at the segment level.

Multilingual sentiment analysis

Data on social media are multilingual by nature, and analyzing them by official language may lead to miss the overall sentiment of online content. Each language has a subset of dialects, hence addressing dialects specificities may improve sentiment analysis accuracy.

Vilares et al. [31] generated BabelSenticNet for multilingual sentiment analysis based on a method that generates SenticNet automatically for a variety of languages. The resource is available for free in 40 languages. Lo et al. [32] presented a multilingual sentiment analysis review. They identified approaches and tools used for this task as well as this research challenges. They recommended a framework to deal with low-resource languages. Fuadvy and Ibrahim [33] proposed a multilingual sentiment classifier to understand Malaysians reaction to the disaster. The experiment results achieved 0.862 accuracy and 0.864 F1-score. Pessutto et al. [34] addressed the multilingual aspect clustering task based on an unsupervised method and the contextual information. The multilingual classification outperformed monolingual baselines.

Harrat et al. [35] focused their study on giving examples of research areas that deal with Moroccan, Algerian and Tunisian Maghrebi Arabic dialects. Oussous et al. in [36] combined several classification algorithms to improve sentiment analysis system performances. They proved the efficiency of this combination as it is better in performance and time terms in comparison to other approaches. Maghfour and Elouardighi in [37] analyzed Facebook comments for sentiment analysis purpose. Their system uses SVM and NB classifiers to deal with Modern Standard Arabic as well as Moroccan dialect.

Irony detection

The collection of linguistic resources for irony is followed by their annotation as ironic or not. Farias and Rosso [4] cited two

approaches for ironic corpus construction, namely self-tagging and crowdsourcing. The self-tagging approach uses tags such as #irony and #sarcasm to identify ironic tweets, while native speakers annotate crowdsourcing tweets manually. Ghanem et al. [3] used a set of keywords to collect Middle East tweets related to the Maghreb Arab spring and the presidential elections. Two Arabic native speakers have manually annotated the collected tweets. Zhang et al. [38] cited the same annotation approaches which can be either automatic using #irony and #sarcasm hashtags or manually performed by humans.

Al-Ayyoub et al. [39] presented a comprehensive overview of the Arabic sentiment analysis. They reviewed previous work to identify gaps and open problems in order to guide future studies in this area. They highlighted the challenges of sarcasm and offensive language that are involved in the notion of irony.

Irony detection has gained the interest of many researchers in a variety of languages. Farias and Rosso [4] addressed the impact of figurative language devices on sentiment analysis. They exploited syntactic, lexical, and semantic features to detect sarcasm. Kolchinski and Potts [40] represented the dense embedding approach to detect interactions between the author and the text that are considered as pertinent when detecting sarcasm. Rajadesingan et al. [6] employed lexical and linguistic features, besides the relationship between authors current and past tweets. Lunando and Purwarianti [41] proposed two level classification, where they first classified sentiment, and then they employed additional features to conduct sarcasm detection. They calculated the number of interjection words that are not grammatically related to the rest of a sentence and negativity information. Joshi et al. [5] presented a survey in which the mentioned works captured interjections, intensifiers and sarcastic patterns that are characterized by the presence of positive verbs and negative phrases. They explored other features like punctuation marks and emoticons. Van Hee et al. [42] presented a shared task that aims to detect irony in English tweets. The first subtask determines whether a tweet is ironic and the second subtask defines the type of irony. The system was trained on 3,834 tweets and tested on 784 tweets. It was based on various features, which are lexical features such as hashtag and punctuation counts, n-grams and emoji presence besides sentiment features. They employed different classifiers such as Support Vector Machines, Random Forest, Naïve Bayes and Maximum Entropy. Zhang et al. [38] formulated irony detection as a transfer learning by enriching irony supervised learning with external sentiment analysis resources. They focused on identifying the implicit incongruity in text. They reported the usefulness of transfer learning in improving irony detection. Justo et al. [43] reported that semantic information and length information provide significant improvement in sarcasm detection. Zhang and Abdul-Mageed [44] fine-tuned the pre-trained bidirectional encoders from transformers (BERT) for irony detection on gold data in a multi-task setting and obtained 82.4 macro F1 score. Chauhan et al. [45] proposed a multi-task deep learning framework to solve sarcasm, sentiment and emotion detection problems in a multi-modal conversational scenario. Majumder et al. [46] trained a classifier for sarcasm and sentiment detection tasks in a neural network using multi-task learning. The method outperformed the results obtained with separate classifiers.

Irony detection is a challenging task for the Arabic language, so several types of research and shared tasks have been conducted to deal with this phenomenon. Rosso et al. [47] reported a review focusing on the Arabic language about author profiling, as well as deception and irony detection challenges. Karoui et al. [48] presented a supervised approach for irony detection in Arabic tweets. They borrowed features employed in other languages, among these features, sentiments, opinion shifters and contextual features. They reached an accuracy of 72.76%. Ghanem et al. [3] built a dataset of 5,030 Arabic political reviews related to the Middle East and the Maghreb. The tweets are written in Modern Standard Arabic besides its varieties Egypt, Gulf, Levantine and Maghrebi dialects. They employed both classical machine learning based on SVM, Logistic Regression, and Ensemble models and deep learning approaches such as CNN, RNN, and LSTM. The best F-score achieved is 0.844. They reported that models based on classical features outperformed the neural ones. Nayel et al. [49] aimed to detect irony in Arabic tweets. They employed Bag-of-Words for feature extraction. They achieved an accuracy of 82.1% using ensemble learning that uses random forest, SVM and multinomial Bayes. Ranasinghe et al. [50] evaluated the performance of several deep learning models to detect irony in Arabic tweets and explored the effect of text processing on system accuracy.

Burgers and Mulken [51] provided an overview of all irony markers in oral or written texts. They described some irony types that have been treated as irony markers. Liebrecht et al. [52] reported in their study that sarcastic messages in microtexts on Twitter are often signaled by hyperbole and contain intensifiers and exclamations, or they are explicitly marked with the hashtag '#sarcasm'. Kunneman et al. [53] developed a system to detect sarcastic tweets in a realistic sample of 2.25 million Dutch tweets without tags or marked by the hashtags '#sarcasm', '#irony', or '#not'. They found that most sarcastic tweets contain a literally positive message and sarcasm marker types such as exclamations, intensified as well as unintensified words. Hallmann et al. [54] described sarcasm as a concept that changes the polarity of a message implicitly. They found that tweets with a user mention contain fewer irony markers than tweets not addressed to a particular user. They concluded that irony markers are used more often when there is less mutual knowledge between sender and receiver.

Offensive language as well gained the interest of many researchers due to its disastrous effects on person psychology, political level and every aspect targeted by this language. Offensive reviews are strongly negative, which means that not all negative reviews are offensive. Therefore, the detection of offensive language requires an offensive lexicon in addition to lexical markers such as call letters (حروف النداء like يا) which indicate that the speech was addressed to someone. Mubarak et al. [55] produced the largest Arabic corpus that contains 10,000 offensive tweets. They conducted many experiments to detect offensive language and reached 79.7% of F-measure. Mohaouchane et al. [56] applied a variety of neural networks on 15,050 labeled Arabic YouTube comments to detect offensive language in social media. They reached an accuracy of 83.46%.

3. Used technical tools

In this paragraph, we present the software and tools used in our system, specifying the reasons behind these choices.

Weka[1] is an open source software that consists of relevant functionalities among which various classical machine learning algorithms and attribute selection methods. It is one of the most used tools written in Java according to the site.[2] We use the functionalities of Weka using a Java API.

DeepLearning4J[3]: is an open source deep learning Java API that is designed for use in enterprise environments on GPUs and distributed CPUs.

1 https://www.cs.waikato.ac.nz/ml/weka/.

2 https://www.softwaretestinghelp.com/machine-learning-tools/.

3 https://deeplearning4j.org/.

Table 1
Sentiment corpora statistics.

Corpus	Sentiment _corpus1	Sentiment _corpus2	Irony _corpus	Balanced
Positive	141	1824	0	1965
Negative	802	2610	2091	1965
Total	943	4434	2091	3930

Google translate[4] is a well-known good quality machine translation service. It is the most used according to the site.[5]

Almaany[6]: is a multilingual online dictionary that offers for each term its corresponding meaning. It helps us to get meaning of Arabic terms.

We developed our sentiment analysis system using the object oriented programming language Java. We choose this language as our system used tools developed with Java.

4. System linguistic resources

In this section, we describe the linguistic resources and basic knowledge to detect ironic expressions and therefore improve the performance of the sentiment analysis system.

4.1. Corpus resources

4.1.1. Ironic corpus

We aim to extract sarcastic features before proceeding to the classification of sentiments. We use the linguistic resource Irony_corpus that was collected from Twitter by the authors of [3]. The corpus contains political issue tweets that were annotated manually by two native speaker annotators as ironic or not. They reached an inter-annotator agreement score of 76% using Cohen's Kappa which is in line with irony annotation agreement in other languages. The disagreement is due to the miscomprehension of some dialectal words, and the lack of context to understand ironic tweet meaning. The study that describes this corpus defines irony as a combination of sarcasm, satire and parody. According to the given definition, ironic tweets contain positive or negative expressions with a negative intention. Given the annotation difficulties relating to the size of the corpus that contains 2091 ironic tweets and 1933 non-ironic tweets, we have restricted the annotation to ironic tweets (2091 tweets) only. So we have performed a second check of the ironic tweets to define the corresponding sentiment, and hence, we tagged the ironic tweets as negative since they imply a negative sentiment. Table 1 gives statistics of Irony_corpus.

4.1.2. Sentiment corpus

We used four sentiment corpora. The first corpus Sentiment_corpus1 that we have collected and annotated contains tweets related to Moroccan political events. The second corpus Sentiment_corpus2 contains tweets related to the Arabic world revolution and was collected and annotated by Mohab Youssef. This corpus is available at GitHub.[7] The third corpus is balanced and combines the positive tweets of Sentiment_corpus1 and Sentiment_corpus2 and an equal number of Ironic tweets of Irony_corpus that are tagged as negative. Table 1 gives statistics of Irony_corpus and these sentiment corpora.

Pretreatment

The corpora contain non-Arabic words and sentences, and are characterized by the diversity of the spelling styles of some tweets, which makes polarity detection a hard task. We pretreat the corpora by removing non-Arabic reviews, numbers, elongations, and diacritics.

Table 2
External lexicons statistics.

Class	Lex1	Lex2	Lex3	Lex4	Lex5	Lex6	Total
Positive	566	1666	107	1278	221	296	2836
Negative	414	2500	118	2226	136	220	3690

4.2. Lexical resources

4.2.1. Sentiment lexicon

External lexicons: Our goal in this section is the collection of external sentimental resources to test their effect and performance in areas other than their own.

Lex1 (SemEval[8]*)*: A sentimental lexicon whose terms are given with their corresponding sentiment intensity.

Lex2 (MPQA[9]*)*: An Arabic translation of the MPQA English terms. This version contains terms with their equivalent Buckwalter transliteration as well as the corresponding sentimental tag.

Lex3 (Seeds[10]*)*: A lexicon that contains the Arabic translation of English seed words set.

Lex4 (ENG_AR): An English lexicon built by Liu et al. [57]. The English lexicon was translated into Arabic by [22] and then filtered following a meticulous manner by the same system.

Lex5 (HAPP): A lexicon collected in [22] by browsing a sentimental dataset. The lexicon contains both positive and negative terms.

Lex6 (LABR): A lexicon created in [58] and cleaned by [22] based on term semantic. The system keeps only general sentimental terms.

The lexicons Lex4, Lex5 and Lex6 have been already pretreated in [22] by removing redundancy and verifying semantic and polarity. We followed the same steps of [22] to pretreat Lex1, Lex2 and Lex3. Afterward, all the lexicons have been gathered (Total), and the redundancies between lexicon terms were removed. The statistics related to lexicons are given in Table 2.

Specific lexicons: We have manually extracted political specific lexicons from Sentiment_corpus1 and Irony_corpus.

Offensive lexicon: The offensive lexicon will be useful in detecting offensive tweets. The semantic consideration is crucial when annotating terms as offensive or not since the offensive lexicon is sharper than the negative sentiment lexicon. We browse the corpus built in [59], which contains political offensive tweets, to extract 1558 terms manually among which 1120 are offensive and 438 are negative. The results are presented in Table 3.

4.2.2. Lexicons summary

We pretreated each lexicon by eliminating numbers, non-Arabic and dialectal words, and redundancies between words. We summarize the obtained results statistics in Table 3. The last column of Table 3 (Overall sentiment lexicon) presents the combination of all these lexicons without redundancy.

We have constructed two lexicons, positive and negative.

- The positive lexicon contains positive terms such as هامة /hAmp/ (important), مطمئن/mTm}n/ (reassured), تمام /tmAm/ (all right).

4 https://translate.google.com/.
5 https://www.makeuseof.com/tag/best-online-translators/.
6 https://www.almaany.com/ar/dict/ar-ar/.
7 https://github.com/Mohabyoussef09/Arabic-Sentiment-Analysis.
8 http://www.saifmohammad.com/WebPages/SCL.html .
9 http://www.purl.org/net/ArabicSA .
10 http://saifmohammad.com/WebPages/WebDocs/arabicSA-JAIR.pdf .

Table 3
Lexicons statistics.

Corpus	External	Sentiment_corpus1	Irony_corpus	Offensive_corpus	Total	Overall sentiment lexicon
Lexicon	External_lexicon	Sentiment_lexicon	Irony_lexicon	Offensive_lexicon		
Positive	2836	530	954	Not reported	4320	3948
Negative	3690	1248	1638	1558	8134	7066

Table 4
Bag of words statistics.

	Balanced corpus		
Type	Number of tweets	Number of words	Vocabulary
Positive	1,965	16,879	9,331
Negative	1,965	25,629	13,660

Table 5
Bag of words iterations.

Iteration	I0	I1	I2	I3	I4	I5
Positive	9,331	7,363	1,072	355	100	37
AOT	1.27	2.84	4.52	7.64	10.89	14.69
Negative	13,660	9,453	1,862	399	118	37
AOT	1.45	3.26	6.99	12.53	21.32	34.17

- The negative lexicon contains negative terms such as ينزف /ynzf/ (bleeds), كارثى /kArvY/ (disastrous) and offensive terms that are strongly negative such as حربايه /HrbAyp/ (chameleon), بلطجي /blTjy/ (Bandit), الخائن (bully), /AlxA}n/ (the traitor), الأحمق /Al>Hmq/ (the crazy) terms.

In contrast to what has been reported in some studies ([13] and [14]) for languages other than Arabic, most words of these lexicons are negative. This phenomenon is partly explained by the political nature of the corpora used to extract tweets and the strong presence of negative discourse in political debate in the Arab world.

4.2.3. Bag of words (BOW) lexicon

In order to minimize the hard process of the lexicon manual construction, we proceed with an automatic lexicon construction.

The BOW lexicon is an automatic lexicon extracted from the balanced corpus. It is made up of the most frequent words and ignores grammar and word order. In order to widen the gap between positive and negative tweets, we follow these pretreatment steps:

- We split the tweets based on their tags as positive or negative.
- We tokenize the sentences of each split into terms to obtain positive and negative lexicons.
- We remove stop words from the lexicons (the stop words list contains 794 terms collected by browsing various corpora. The negation words also are included in this list).
- We remove also the overlap between the lexicons.

As a result, we obtained a clean lexicon that contains 9331 positive and 13660 negative terms (see Table 4).

To reduce the BOW sizes, we iterate five times by performing the following operations:

- Calculate the average occurrence of terms (AOT) of each lexicon.
- Keep in the new lexicon only the terms whose occurrence is greater than the AOT.

The results obtained are presented in Table 5. We choose the lexicon of iteration 4 (I4) to represent the corpus since it gives an acceptable size to represent the political tweets.

We note that each of the 100 positive terms of the lexicon I4 appears in the initial corpus of positive tweets at least 8 times (AOT = 7.64). Likewise, the 118 negative terms of I4 appear in the initial corpus on average more than 22 times (AOT = 21.32).

5. Sarcasm linguistic features

In order to enrich the existing linguistic resources, we have manually collected the following lexicons from the Irony_corpus, which seem to be useful for improving sentiment analysis system accuracy.

Hashtag lexicons: we collected all present hashtags in Irony_corpus, and then we annotated them as positive or negative. The terms of hashtags may be words or phrases.

- Positive Hashtags #نهضة /nhDp/[11] (renaissance), #شكرا_روسيا /$krA_rwsyA/ (thank you Russia)
- Negative Hashtags #جهل /jhl/ (ignorance), #كلاب /klAb/ (dogs), #الحرب /AlHrb/ (war)

Exclamation mark: the exclamation mark (!) is used to express strong emotions either surprise, excitement or astonishment.

Laugh indicators: such as هههه /hhh/ هع /hE/ هاها /hAhA are expressions that follow positive expressions to mock them.

Sarcasm indicators: we also collected sarcasm indicators from Irony_corpus. These indicators indicate that the expressions to which they belong contain sarcastic intentions. A sample of these indicators are: مسخرة /msxrp/ (sarcasm), هزار /hzAr/ (kidding), هزلت /hzlt/ (joke), ساخر /sAxr/ (sarcastic), تريقه /tryqh/ (sarcasm), استهزاء /AsthzA'/ (mockery).

Mixed indicators: sarcasm can be identified by the presence of incongruity in a specific sentence. Hence, within sarcastic expressions we find positive and negative content. Thus, we collected a set of mixed indicators such as لكن /lkn/ (but), رغم /rgm/ (despite), مع ذلك /mE*lk/ (although), إنما /<nmA/ (but), which are very useful in identifying sarcastic sentences.

Table 6 gives the statistics of these resources.

6. Sentiment classification

In this section, we aim to build a supervised approach to classify tweets as positive or negative and enhancing the classification by sarcastic features.

In order to classify the tweets, we adopt the approach developed in [22]. This approach has been applied to multi-domain corpora, namely the domains of hotels, products, restaurants and movies, and it showed pertinent results in all these domains characterized by the presence of reviews of different lengths, short, medium and long.

Our system that addresses sentimental reviews nature and dialect specificities is an improvement of the approach described in [22] that is based on a set of pertinent features such as

[11] http://www.qamus.org/transliteration.htm.

Table 6
Sarcastic features statistics.

Indicators	Hashtag lexicon		Exclamation mark	Laugh indicators	Sarcasm indicators	Mixed indicators
	+#	-#				
Statistiques	202	454	1	4	121	46

positive and negative term occurrences besides addressing the negation context. The analyzed corpus is multidialectal, we have used a cross dialect lexicon that retains meaning when passing from one dialect to another. Besides the Arabic dialect common characteristics, the classification was enhanced by the detection of the specificities of some dialects that use negation clitics as well as negation words to negate a term. We have constructed a set of negation words that will help to improve our sentiment analysis system accuracy. The negation words that precede the sentimental terms were detected based on a list that contains 189 negation terms such as {لا /lA/ (No), عديم /Edym/ (He is not), دون/dwn/ (without)}. The Moroccan dialect is distinguished besides the negation words by the presence of negation clitics, such as the proclitic م and the enclitic ش. Thus, we collected seven negation proclitics and two negation enclitics.

We consider $L_P = (P_i)_{1\leq i\leq n}$ a positive lexicon and $L_N = \left(N_j\right)_{1\leq j\leq m}$ a negative lexicon and whose respective sizes are n and m. The characteristic vector of a tweet T is composed of the following features.

$p_i(1 \leq i \leq n)$ is the occurrence of the positive term P_i in T.
$n_j(1 \leq j \leq m)$ is the occurrence of the negative term N_j in T.
$p_T = \sum_{i=1}^{n} p_i$ is the score of positive terms that belong to T.
$n_T = \sum_{j=1} n_j$ is the score of negative terms that belong to T.
$\overline{p}_T$ is the number of the negated positive terms of the tweet T.
$\overline{n}_T$ is the number of the negated negative terms of T.
The characteristic vector V_T of each tweet T is given by:

$$V_T = (p_1, \ldots, p_n, n_1, \ldots, n_m, p_T, n_T, \overline{p}_T, \overline{n}_T) .$$

The approach is enhanced by the following features extracted from the corpus of study either the ironic or the sentimental:

- Presence of incongruity: it is detected by the presence of mixed indicators (P_{MI}) to indicate that an expression carries both positive and negative tags. The part on the left of the indicator will be opposite to the part on the right in terms of polarity.
- Presence of hashtags ($P_{hashtag}$): The feature defines the relation between the object and the corresponding subjective sentences. Indeed, writing positive reviews on an object designated by a negative hashtag may prove that the reviews are sarcastic.
- Presence of exclamation mark ($P_{exclamation}$): this feature is important as it expresses astonishment which means that the opinion holder is not satisfied with the situation.
- Presence of laugh indicator (P_{laugh}) (هههه /hhh/ هع /hE/ هاها /hAhA/…): these indicators are useful because they follow positive expressions to mock them.
- Presence of sarcasm indicators (P_{SI}) which are explicit indicators that determine whether an expression is sarcastic or not. Sarcasm indicators are pertinent features that contribute in identifying sarcastic expressions.

$$V_{ST} = \left(P_{MI}, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{laugh}, P_{SI}\right) .$$

In the sentiment analysis system corpus, a given expression is positive or implies a negative sentiment regardless of the polarity of the used terms. The negativity is implicit in the sarcastic expressions (positive sentences within negative context) and explicit in the offensive ones (offensive terms). In order to detect sarcastic expressions, we select the aforementioned features to enhance the sentiment analysis system. Offensive language detection is automatically included within the sentiment analysis system since the used lexicon resources contain offensive terms.

7. System presentation

7.1. Text representation

Bag of words (BOW): The Bag of words model is used in natural language processing and information retrieval to represent a text as a bag of the most frequent words, it ignores grammar and word order.

Word2vec : It has two main methods of contextualizing words, the Continuous Bag of Words (CBOW) that uses source words to predict target words and the Skip-Gram model that uses target words to predict the source.

7.2. Classifiers

Support Vector Machine: An SVM is a supervised learning method based on the notion of support vectors that are points involved in identifying the maximum margin which corresponds to the hyperplanes separating data from classes.

Random Forest (RF): RF is an ensemble learning method that works by building a multitude of decision trees at the time of learning, and classification will be done by majority vote between the decisions of each tree. Random forest corrects errors in individual decision trees and provides the big picture by adapting rules to training sets.

Recurrent Neural Network (RNN): The idea behind RNNs is to use sequential information. RNNs are called recurrent because any output depends on previous calculations.

Long Short-term memory (LSTM): Since RNNs are short-term memory (they do not support long-term dependency), the LSTM architecture addresses this problem by retaining long-term information. The LSTM system works in three stages:

- Forget Gate: decide how much of the past we need to remember.
- Update Gate/input gate: decides how much of this unit is added to the current state.
- Output Gate: decide which part of the current cell goes to the output.

We now present the classical machine learning and the deep learning approaches that we have used for classification.

Classical machine learning: from the extracted features, we generated a model using either Support vector machine or Random forest classifiers. Experiments were also carried out with Naïve Bayes and KNN classifiers but were not reported since they did not perform well compared to other classifiers.

Deep learning: we created a word2vec model with a layer size equal to 300 based on Skip-gram from a training corpus, and then we fed it into a deep neural network that combines both RNN and LSTM. This method is applied besides the classical machine learning approach since we aim to analyze the effect of sarcastic features that is not evident with a neural network.

Table 7
Resources and features of sentiment analysis systems.

	Sentiment corpora	Sentiment_corpus1		Sentiment_corpus2		Irony_corpus	Positive tweets	Negative tweets
		Positive	Negative	Positive	Negative	Negative		
Linguistic resources	Sentiment_corpus1	×	×				141	802
	Sentiment_corpus2			×	×		1824	2610
	Balanced	×		×		×	1965	1965
	Lexicon	The SL lexicon composed of 150 positive terms and the 150 negative terms of Overall lexicon most frequent in the sentiment corpora and the BOW lexicon (BOW) for balanced corpus						
System	Features	Sentiment features (Sentiment lexicon occurrence + negation context) and Sarcastic features (hashtags, exclamation mark, laugh indicator, mixed indicator, sarcasm indicator)						
	Characteristic vector	We use the vector V = (V_T, V_{ST}) that is the combination of the sentiment features $V_T = (p_1, \ldots, p_n, n_1, \ldots, n_m, p_T, n_T, \overline{p}_T, \overline{n}_T)$ and the sarcastic features $V_{ST} = \left(P_{MI}, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{laugh}, P_{SI}\right)$						
	Classifier	Classical machine learning (RF, SVM), deep learning (RNN, LSTM)						
Metrics		% split, 10 folds cross-validation						

7.3. Test options

Percentage split (%): we randomly extracted 80% from the corpus for use in the training phase and the remaining 20% for the test.

Cross validation (CV): cross validation is not sensitive to the training set. Indeed, this approach randomly decomposes the corpus into k subsets of the same size, and at each iteration chooses (k-1) subsets for the training and the k^{th} for the test. The accuracy of the system is the average of the accuracies of the k tests. In our tests, we have chosen k equal to 10.

Supplied test set: we build a classifier using a training set and then we supply a specific test set to evaluate the model.

7.4. Classification metrics

We evaluate the classification using accuracy, precision, recall and F-measure metrics defined by:

$$\text{Accuracy} = \frac{\text{Number of correctly classified tweets}}{\text{Number of tweets in the test corpus}}$$

$$\text{Precision } C_i = \frac{\text{Number of tweets in } C_i \text{ correctly classified}}{\text{Number of tweets classified as } C_i}$$

$$\text{Recall } C_i = \frac{\text{Number of tweets in } C_i \text{ correctly classified}}{\text{Number of tweets of the class } C_i}$$

$$F - \text{measure } C_i = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{precision} + \text{Recall}}$$

C_i can correspond to the class of positive tweets or that of negative tweets.

8. System evaluation

8.1. Sentiment experiments

In this section, we combine sentiment (positive, negative and offensive) and sarcastic features to build a robust sentiment analysis system that addresses irony. Feature extraction is based on the overall sentiment lexicon (Table 3) that contains positive, negative and offensive terms besides sarcastic features including hashtags, mixed and sarcasm indicators (Table 6). We perform experiments on the following corpora: Sentiment_corpus1, Sentiment_corpus2 and the combination of these corpora defined by Balanced. The usefulness of the balanced corpus lies in keeping the balance between the positive and negative tweets within the corpus and hence their distribution when performing classification.

Table 8
The occurrences of lexicon terms in the studied corpora.

Corpus	Sentiment_corpus1	Sentiment_corpus2	Balanced	
			SL	BOW
OPT	383	1683	1842	751
ONT	302	1143	768	1479

Table 9
Sentiment classification results.

Corpus	Lexicon	Test option			
		%		CV	
		SVM	RF	SVM	RF
Sentiment_corpus1	$\mathbf{V_{SL}}$	91.00	87.30	86.21	87.49
Sentiment_corpus2	$\mathbf{V_{SL}}$	63.59	63.92	64.59	64.95
Balanced	$\mathbf{V_{SL}}$	74.93	74.68	75.47	75.39
	$\mathbf{V_{Bow}}$	82.32	83.59	82.52	84.05

The classification in these corpora uses a part of the overall sentiment lexicon (150 occurring positive terms and 150 occurring negative terms) of Table 3. For the balanced corpus, we also tested the classification with the BOW lexicon. This lexicon is automatically extracted from the balanced corpus as described in Section 4.2.3. The obtained BOW from this corpus at iteration 4 contains 100 terms from the positive split and 118 terms from the negative split.

We summarize Table 7 in the used resources in this section and their corresponding statistics as well as the employed features to classify sentiments.

The tests based on the overall lexicon are characterized by a vector that contains 150 positive terms and 150 negative terms.

The occurrences of positive and negative terms (OPT and ONT respectively) in each corpus are summarized in Table 8. The results of the classification are given in Table 9. The V_{SL} represents the characteristic vector V defined in Table 7 based on SL and V_{Bow} is the vector V based on the BOW.

The classification of Sentiment_corpus1 gives promising results. The used lexicon for classification gives a good representation to this corpus since the difference between OPT and ONT is not large. The results of Sentiment_corpus2 are degraded although the external lexicons covered many terms in the corpus, this may be due to the used lexicon for classification that generated a wide difference between OPT and ONT terms, which may cause an imbalance between the present terms and the overall polarity. The use of the BOW lexicon in the balanced corpus achieves better results than the SL lexicon. The difference in accuracy between the SL-based classification and the BOW-based classification may be due to the selected SL terms causing

an imbalance with the overall polarity. An exploration of the BOW shows that even not all terms are pure sentimental, they contain named entities that distinguish the positive and the negative splits. In general, the manual extraction of the lexicon that addresses the term semantic is more accurate than an automatic approach; however, the obtained results show the usefulness of the custom bag of words approach introduced in this paper.

8.2. Identification of pertinent sarcastic features

We aim in this section to identify the most pertinent sarcastic features within the constructed sentiment classification system. Thus, we perform several experiments on the Balanced corpus with the RF classifier that gives the best results and the following vectors as features:

V_1: uses the BOW lexicon.

$$V_1 = V_T = (p_1, \ldots, p_n, n_1, \ldots, n_m, p_T, n_T, \overline{p}_T, \overline{n}_T)\,.$$

V_2: uses the BOW lexicon besides hashtag features.

$$V_2 = \left(V_T, P_{hashtag+}, P_{hashtag-}\right)$$

V_3: uses the BOW lexicon besides the exclamation feature.

$$V_3 = (V_T, P_{exclamation})$$

V_4: uses the BOW lexicon besides laugh indicator (هههه) feature.

$$V_4 = \left(V_T, P_{laugh}\right)$$

V_5: uses the BOW lexicon besides sarcasm indicators.

$$V_5 = (V_T, P_{SI})$$

V_6: uses the BOW lexicon besides mixed indicators.

$$V_6 = (V_T, P_{MI})$$

V_7: uses the BOW lexicon besides hashtag and exclamation.

$$V_7 = \left(V_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}\right)$$

V_8: uses V_7 besides laugh indicator هههه.

$$V_8 = \left(V_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{laugh}\right)$$

V_9: uses V_7 besides sarcasm indicators.

$$V_9 = \left(V_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{SI}\right)$$

V_{10}: uses V_7 besides mixed indicators.

$$V_{10} = \left(V_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{MI}\right)$$

V_{11}: uses V_7 besides laugh indicator هههه and sarcasm indicators.

$$V_{11} = \left(V_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{laugh}, P_{SI}\right)$$

V_{12}: uses V_7 besides laugh indicator هههه and mixed indicators.

$$V_{12} = \left(V_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{laugh}, P_{MI}\right)$$

V_{13}: uses V_7 besides sarcasm and mixed indicators.

$$V_{13} = \left(V_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{SI}, P_{MI}\right)$$

V_{14}: uses V_7 besides hashtag, exclamation, laugh indicator هههه, sarcasm and mixed indicators.

$$V_{14} = V_{Bow} = \left(V_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{laugh}, P_{SI}, P_{MI}\right)$$

We have limited ourselves to the aforementioned features since when browsing the corpus we have found that these features distinguish the sarcastic reviews, moreover hyperbole and metaphor have not been considered since they present a study themselves and to perform these tasks we will need additional resources and features [60].

The results of the classification are given in Table 10 using % split test option.

The use of hashtags and exclamation features (V_2 and V_3 respectively) shows an improvement of more than 2% and 4%. Moreover, their joint use further improves performance (V_7 compared to V_2 and V_3). On the other hand, the use of the laugh هههه, sarcasm and mixed indicators shows no improvement (V_4, V_5 and V_6 respectively). However, results improve when certain combinations of these three indicators (laugh هههه, sarcasm, and mixed indicators) are added to the hashtags and exclamation features (V_{11} and V_{14}). The best performance is obtained by using hashtags, exclamation features, laugh هههه, and sarcasm indicators together (V_{11}).

The comparison between V_{11} and V_{14} in Table 11 shows a slight difference. In addition, the F-measure of the positive and negative classes are very close given the balanced nature of the corpus.

We give in Table 12 , the results based on a deep learning approach (DL) that reaches 80.36% of accuracy and is surpassed by the classical machine learning approach (RF). The classical machine learning approach is still efficient in comparison to the deep learning approach that uses a vector with a size of 300, as it necessitates a reduced characteristic vector, which leads to a gain in term of model construction time.

8.3. Error analysis

In this section, we intend to analyze the reviews that were misclassified by the RF classifier and give the misclassification causes. The reviews are represented by numeric vectors. Since the API Weka used by our system takes randomly 80% vectors for training and 20% for testing (from which 16.16% were misclassified), we retained the misclassified vectors and the corresponding reviews. We have a pair of (reviews, vectors) that allow the analysis represented in Table 13 .

The analysis of the above examples shows that the lack of lexicon terms and the representation of short reviews by long vectors are among the error causes. In the following, we try to deal with this by increasing the size of the BOW and decreasing the length of representing vectors.

8.4. Lexicon size increase effect on classification

The results of Table 12 are based on a BOW of reduced size (100 terms from the positive corpus and 118 from the negative one), the results of the following Table 14 are based on an extended BOW (EBOW) with increased size (500 terms from the positive corpus and 500 terms from the negative one). In order to construct the second Bag of Words (EBOW), we followed the pretreatment steps described in Section 4.2.3.

Table 14 shows that the performance of the system has improved significantly by using the EBOW lexicon instead of the BOW lexicon. Moreover, these results confirm the relevance of the use of sarcastic features (results of V_{11} compared to those of V_1) and that of the mixed indicators (results of V_{14} compared to those of V_1). Finally, the V_{11} and V_{11} representations perform similarly with a slight advantage to V_{11}.

8.5. Characteristic vector size reduction

We have performed information gain feature selection under Weka tool. The results showed that the top ranked features are, sarcasm features (the positive and negative hashtags,

Table 10
Effect of the sarcastic features on the sentiment analysis system accuracy.

Vectors	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9	V_{10}	V_{11}	V_{12}	V_{13}	V_{14}
Accuracy	76.08	78.50	80.79	76.08	76.08	76.08	83.46	79.77	83.33	83.46	83.84	79.77	83.33	83.59

Table 11
Evaluation metrics related to classical machine learning using RF.

Vectors	V_{11}				V_{14}			
Metric	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
Positive	83.84	0.78	0.93	0.85	83.59	0.78	0.94	0.85
Negative		0.92	0.74	0.82		0.93	0.73	0.82

Table 12
Evaluation metrics related to the classification using classical machine learning and deep learning.

Metric	RF				DL			
	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
Positive	83.84	0.78	0.93	0.85	80.36	0.74	0.93	0.82
Negative		0.92	0.74	0.82		0.90	0.68	0.78

Table 13
Samples of misclassified reviews.

Review	Tag given by annotators	Tag given by our system	Representation	Error cause
أرض الأحرار....هههههه	1	−1	Contains laugh indicator	Annotation error
أنظر في قلبك ستراني !	1	−1	Contains exclamation indicator	A short review whose terms do not belong to the BOW Ambiguity error
أحذر يا #بشار: فأمة العرب اذا غضبت غردت بالتويتر وغيرت بالبروفايل وقلبت بالصور :) يا سوريا #عرب# D: احنى	−1	1	A vector of zeros	The sentimental terms do not belong to the BOW
ألجالية النوبية فى مصر قامت بدور عظيم فى الحرب	−1	1	A vector of zeros	The sentimental terms do not belong to the BOW
دوق طعم مفهوم الإرهاب فى مصر .. شكل جديد .. بنفس الطعم الرائع .. cpd	−1	1	A vector of zeros	The sentimental terms do not belong to the lexicon

Table 14
Effect of lexicon size on classification.

Test option	%split			CV		
Lexicon	V_1	V_{11}	V_{14}	V_1	V_{11}	V_{14}
BOW	76.08	**83.84**	83.59	76.23	**84.17**	84.05
EBOW	88.17	88.68	**89.19**	85.52	**88.24**	88.07

the exclamation mark, and sarcasm indicator) besides the sentiment features ($p_T, n_T, \overline{p}_T, \overline{n}_T$) (see Sections 5 and 6). Hence, we use only these features to present the tweets in the Balanced dataset. Thus, in order to perform classification we use the reduced characteristic vector.

$$V_R = \left(p_T, n_T, \overline{p}_T, \overline{n}_T, P_{hashtag+}, P_{hashtag-}, P_{exclamation}, P_{laugh}, P_{SI}\right).$$

In Table 15, we extract V_{11} features using the BOW (V_{11Bow}) and the EBOW (V_{11EBow}) and V_R features using the EBOW. We give the results using CV test option and precision, recall and F-measure evaluation metrics.

We note that the use of EBOW helps not only to improve the classification results by more than 4 points but it keeps the balance between the positive and the negative classes results as well (comparison between the precision, recall and F-measure of the positive and negative classes).

The obtained results using the reduced vector show improvement in comparison to V_{11EBow} results. Using the reduced characteristic vector correctly classifies some tweets that were misclassified when using the occurrence vector of the EBOW. Moreover, using the reduced characteristic vector helps to reduce the classification cost and simplifies the interpretation of the obtained results.

9. Systems comparison

This section aims to situate our work in relation to previous works. We present resources, approaches and results of different sarcasm detection systems.

The comparison of our sentiment analysis system that contains sarcastic tweets with other systems of Table 16 proved the efficiency of the extracted features to detect sarcasm, since it outperformed sarcasm detection tasks. By exploiting the sarcastic features, we were able to overcome the problem of sentiment analysis systems relating to the difficulty of assigning the negative tag to sarcastic tweets that contain positive expressions.

Conclusion and further work

The strength of this paper lies in overcoming different challenges. The first one concerns the nature of the used corpus for

Table 15
The CV results using V_{11Bow}, V_{11EBow} and V_R represented by various evaluation metrics.

Features	V_{11Bow}				V_{11EBow}				Reduced vector V_R			
Metric	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure	Accuracy	Precision	Recall	F-measure
Positive	84.17	0.794	0.922	0.854	88.24	0.873	0.895	0.884	89.24	0,867	0,927	0,896
Negative		0.907	0.761	0.828		0.892	0.87	0.881		0,922	0,858	0,888

Table 16
Sarcastic and sentiment analysis systems comparison.

Authors	Resources	Approaches	Evaluation metrics
Van Hee et al. [61]	Training corpus of 3,834 ironic and non-ironic tweets, test set of 784 tweets.	Lexical features such as hashtag, punctuation counts, n-grams, emoji presence besides sentiment features	F1 = 0.71
Ghanem et al. [3]	5030 ironic tweets	Word2Vec, BERT, FastText, SVM, Logistic Regression, RNN, LSTM	F1 = 0.84
Karoui et al. [48]	1733 ironic tweets and 1,733 non-ironic tweets	Surface, sentiment, shifter and contextual features.	F1 = 0.72
Our system	1965 positive tweets 1965 sarcastic tweets 1000 lexicon terms	Sentiment and sarcastic features	Accuracy = 89.24%

classification. The multidialectal corpus contains besides Modern Standard Arabic many varieties such as Egypt, Gulf, Levantine and Maghrebi dialects. In order to overcome this challenge, we used a cross dialect lexicon that retains meaning when passing from one dialect to another. Besides the common characteristics, the classification was enhanced by the detection of the specificities of some dialects that use negation clitics as well as negation words to negate a term. The second challenge relates to the automatic construction of a lexicon for classical machine learning classification. In order to deal with this challenge, we selected a bag of words based on a custom approach which helped to improve the system accuracy. The third challenge concerns the management of irony to improve the sentiment analysis system. Irony implies both sarcasm and offensive language. The difficulty in detecting sarcasm lies in using positive terms to convey a negative feeling, which complicates the task of distinguishing sarcastic tweets from positive tweets. We have performed a multi-task learning that combines two systems that analyze sentiments and address sarcasm. We have thus exploited sarcastic features and trained a model based on sarcastic and positive tweets to reduce the classification steps to one task and improve the sentiment classification system. Offensive terms are useful in detecting negative expressions since they widen the gap between positive and negative tweets by increasing the negativity of a sentence. From the obtained results, sentiment analysis system has been improved when enhanced by sarcastic features. Our system achieved promising results by an improvement of 8% to reach 84.17% of accuracy using classical machine learning and 80.36% using deep learning. The classical machine learning approach has been improved afterward based on the BOW lexicon expansion and the characteristic vector reduction to reach an accuracy of 89.24%.

Besides the cited characteristics, our system can be cross lingual as the constructed model is language independent. We believe that the availability of a labeled corpus, the automatic construction of a lexicon (BOW) and the identification of sarcastic features are sufficient to apply the system to other languages.

As further work, we intend to transfer the extracted knowledge and features from the used political corpora to other domains. We will also analyze the caveats of our approach by addressing the challenges of sarcasm detection such as language type, the used figurative devices, and the used style in social media. Moreover, since the proposed approach is multidialectal, we intend to apply it to other languages to explore the strength of the built system.

CRediT authorship contribution statement

Ibtissam Touahri: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing - original draft, Writing - review & editing. **Azzeddine Mazroui:** Conceptualization, Methodology, Formal analysis, Writing - review & editing, Supervision, Research administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] G. Badaro, et al., A survey of opinion mining in arabic: A comprehensive system perspective covering challenges and advances in tools, resources, models, applications, and visualizations, ACM Trans. Asian Low-Resour. Lang. Inf. Process. 18 (3) (2019) 1–52, http://dx.doi.org/10.1145/3295662.
[2] D.M.E.-D.M. Hussein, A survey on sentiment analysis challenges, J. King Saud Univ., Eng. Sci. 30 (4) (2018) 330–338, http://dx.doi.org/10.1016/j.jksues.2016.04.002.
[3] B. Ghanem, J. Karoui, F. Benamara, V. Moriceau, P. Rosso, Idat at fire2019: Overview of the track on irony detection in arabic tweets, in: Proceedings of the 11th Forum for Information Retrieval Evaluation, 2019, pp. 10-13.
[4] D.I.H. Farias, P. Rosso, et al., Irony, sarcasm, and sentiment analysis, in: Sentiment Analysis in Sentiment Analysis in Social Networks, Elsevier, 2017, pp. 113–128.
[5] A. Joshi, P. Bhattacharyya, M.J. Carman, et al., Automatic sarcasm detection: A survey, 2016, arXiv:1602.03426 [cs], févr, http://arxiv.org/abs/1602.03426.
[6] A. Rajadesingan, R. Zafarani, H. Liu, et al., Sarcasm detection on Twitter: A behavioral modeling approach, in: Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15, Shanghai, China, 2015, pp. 97–106, http://dx.doi.org/10.1145/2684822.2685316.
[7] E. Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2) (2016) 102–107.
[8] J. Han, Z. Zhang, N. Cummins, B. Schuller, et al., Adversarial training in affective computing and sentiment analysis: Recent advances and perspectives [review article], IEEE Comput. Intell. Mag. 14 (2) (2019) 68–81, http://dx.doi.org/10.1109/MCI.2019.2901088.
[9] E. Cambria, Y. Li, F.Z. Xing, S. Poria, K. Kwok, et al., Senticnet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event Ireland, 2020, pp. 105–114, http://dx.doi.org/10.1145/3340531.3412003.
[10] E. Cambria, S. Poria, D. Hazarika, K. Kwok, et al., SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, p. 8.

[11] O. Oueslati, E. Cambria, M.B. HajHmida, H. Ounelli, et al., A review of sentiment analysis research in Arabic language, Future Gener. Comput. Syst. 112 (2020) 408–430, http://dx.doi.org/10.1016/j.future.2020.05.034.
[12] M. Nabil, M. Aly, A. Atiya, et al., ASTD: Arabic sentiment tweets dataset, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 2015, pp. 2515–2519, http://dx.doi.org/10.18653/v1/D15-1299.
[13] P.S. Dodds, E.M. Clark, S. Desu, M.R. Frank, A.J. Reagan, J.R. Williams, L. Mitchell, K.D. Harris, I.M. Kloumann, J.P. Bagrow, K. Megerdoomian, Human language reveals a universal positivity bias, Proc. Natl. Acad. Sci. 112 (8) (2015) 2389–2394.
[14] I.M. Kloumann, C.M. Danforth, K.D. Harris, C.A. Bliss, P.S. Dodds, Positivity of the English language, PLoS One 7 (1) (2012) e29484.
[15] Y. Lee, S. Park, K. Yu, J. Kim, Building place-specific sentiment lexicon, in: Proceedings of the 2nd International Conference on Digital Signal Processing, 2018, pp. 147-150.
[16] Y. Wang, F. Yin, J. Liu, M. Tosato, et al., Automatic construction of domain sentiment lexicon for semantic disambiguation, Multimedia Tools Appl. 79 (31–32) (2020) 22355–22373, http://dx.doi.org/10.1007/s11042-020-09030-1.
[17] F. Yin, Y. Wang, J. Liu, L. Lin, et al., The construction of sentiment lexicon based on context-dependent part-of-speech chunks for semantic disambiguation, IEEE Access 8 (2020) 63359–63367, http://dx.doi.org/10.1109/ACCESS.2020.2984284.
[18] Z. Rajabi, M.R. Valavi, M. Hourali, et al., A context-based disambiguation model for sentiment concepts using a bag-of-concepts approach, Cogn. Comput. (2020) 1–19, http://dx.doi.org/10.1007/s12559-020-09729-1.
[19] F.B. Goularte, D. Sorato, S.M. Nassar, R. Fileto, H. Saggion, et al., MSC+: Language pattern learning for word sense induction and disambiguation, Knowl.-Based Syst. 188 (2020) 105017, http://dx.doi.org/10.1016/j.knosys.2019.105017.
[20] K. Orkphol, W. Yang, et al., Word sense disambiguation using cosine similarity collaborates withword2vec and wordnet, Future Internet 11 (5) (2019) http://dx.doi.org/10.3390/fi11050114.
[21] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, M. Sedlmairand, et al., More than bags of words: Sentiment analysis with word embeddings, Commun. Methods Meas. 12 (2–3) (2018) 140–157, http://dx.doi.org/10.1080/19312458.2018.1455817.
[22] I. Touahri, A. Mazroui, et al., Studying the effect of characteristic vector alteration on Arabic sentiment classification, J. King Saud Univ.-Comput. Inf. Sci. (2019) http://dx.doi.org/10.1016/j.jksuci.2019.04.011.
[23] H. Zheng, C. Wu, et al., Predicting personality using facebook status based on semi-supervised learning, in: ACM International Conference Proceeding Series, Part F1481, 2019, pp. 59–64, http://dx.doi.org/10.1145/3318299.3318363.
[24] Y. Han, Y. Liu, Z. Jin, et al., Sentiment analysis via semi-supervised learning: a model based on dynamic threshold and multi-classifiers, Neural Comput. Appl. 32 (9) (2020) 5117–5129, http://dx.doi.org/10.1007/s00521-018-3958-3.
[25] M. BalaAnand, N. Karthikeyan, S. Karthik, R. Varatharajan, G. Manogaran, C.B. Sivaparthipan, et al., An enhanced graph-based semi-supervised learning algorithm to detect fake users on Twitter, J. Supercomput. 75 (9) (2019) 6085–6105, http://dx.doi.org/10.1007/s11227-019-02948-w.
[26] Q. Li, X.M. Wu, H. Liu, X. Zhang, Z. Guan, et al., Label efficient semi-supervised learning via graph filtering, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019, pp. 9574–9583, http://dx.doi.org/10.1109/CVPR.2019.00981.
[27] A. Kim, S.B. Cho, et al., An ensemble semi-supervised learning method for predicting defaults in social lending, Eng. Appl. Artif. Intell. 81 (2019) 193–199, http://dx.doi.org/10.1016/j.engappai.2019.02.014.
[28] X. Ye, H. Dai, L. Dong, X. Wang, et al., Multi-view ensemble learning method for microblog, Expert Syst. Appl. (2020) 113987, http://dx.doi.org/10.1016/j.eswa.2020.113987.
[29] M.S. Akhtar, A. Ekbal, E. Cambria, et al., How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes], IEEE Comput. Intell. Mag. 15 (1) (2020) 64–75, http://dx.doi.org/10.1109/MCI.2019.2954667.
[30] Q. Yang, Y. Rao, H. Xie, J. Wang, F.L. Wang, W.H. Chan, et al., Segment-level joint topic-sentiment model for online review analysis, IEEE Intell. Syst. 34 (1) (2019) 43–50, http://dx.doi.org/10.1109/MIS.2019.2899142.
[31] D. Vilares, H. Peng, R. Satapathy, E. Cambria, et al., Babelsenticnet: A commonsense reasoning framework for multilingual sentiment analysis, in: Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018, 2019, pp. 1292–1298, http://dx.doi.org/10.1109/SSCI.2018.8628718.
[32] S.L. Lo, E. Cambria, R. Chiong, D. Cornforth, et al., Multilingual sentiment analysis: from formal to informal and scarce resource languages, Artif. Intell. Rev. 48 (4) (2017) 499–527, http://dx.doi.org/10.1007/s10462-016-9508-4.
[33] M.J. Fuadvy, R. Ibrahim, et al., Multilingual sentiment analysis on social media disaster data, in: ICEEIE 2019 - International Conference on Electrical, Electronics and Information Engineering: Emerging Innovative Technology for Sustainable Future, 2019, pp. 269–272, http://dx.doi.org/10.1109/ICEEIE47180.2019.8981479.
[34] L.R.C. Pessutto, D.S. Vargas, V.P. Moreira, et al., Multilingual aspect clustering for sentiment analysis, Knowl.-Based Syst. 192 (2020) 105339, http://dx.doi.org/10.1016/j.knosys.2019.105339.
[35] S. Harrat, et al., Maghrebi Arabic dialect processing: an overview to cite this version: HAL Id: hal-01873779 Maghrebi Arabic dialect processing: an overview, 2018.
[36] A. Oussous, A.A. Lahcen, S. Belfkih, et al., Improving sentiment analysis of moroccan tweets using ensemble learning, 2, 2018, pp. 91–104, http://dx.doi.org/10.1007/978-3-319-96292-4.
[37] M. Maghfour, A. Elouardighi, Standard and dialectal arabic text classification for sentiment analysis, in: International Conference on Model and Data Engineering, Springer, Cham., 2018, pp. 282–291.
[38] S. Zhang, X. Zhang, J. Chan, P. Rosso, et al., Irony detection via sentiment-based transfer learning, Inf. Process. Manage. 56 (5) (2019) 1633–1644, http://dx.doi.org/10.1016/j.ipm.2019.04.006.
[39] M. Al-Ayyoub, A.A. Khamaiseh, Y. Jararweh, M.N. Al-Kabi, et al., A comprehensive survey of arabic sentiment analysis, Inf. Process. Manage. 56 (2) (2019) 320–342, http://dx.doi.org/10.1016/j.ipm.2018.07.006.
[40] Y.A. Kolchinski, C. Potts, et al., Representing social media users for sarcasm detection, 2018, arXiv:1808.08470 [cs], août http://arxiv.org/abs/1808.08470.
[41] E. Lunando, A. Purwarianti, et al., Indonesian social media sentiment analysis with sarcasm detection, in: 2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Sanur Bali, Indonesia, 2013, pp. 195–198, http://dx.doi.org/10.1109/ICACSIS.2013.6761575.
[42] C. Van Hee, E. Lefever, V. Hoste, et al., Semeval-2018 task 3: Irony detection in english tweets, in: Proceedings of the 12th International Workshop on Semantic Evaluation, New Orleans, Louisiana, 2018, pp. 39–50, http://dx.doi.org/10.18653/v1/S18-1005.
[43] R. Justo, T. Corcoran, S.M. Lukin, M. Walker, M.I. Torres, et al., Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web, Knowl.-Based Syst. 69 (2014) 124–133, http://dx.doi.org/10.1016/j.knosys.2014.05.021.
[44] C. Zhang, M. Abdul-Mageed, et al., Multi-task bidirectional transformer representations for irony detection, in: CEUR Workshop Proceedings, Vol. 2517, 2019, pp. 391–400.
[45] D.S. Chauhan, D.S. R., A. Ekbal, P. Bhattacharyya, et al., Sentiment and emotion help sarcasm? A multi-task learning framework for multi-modal sarcasm, Sentiment Emot. Anal. (2020) 4351–4360, http://dx.doi.org/10.18653/v1/2020.acl-main.401.
[46] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, A. Gelbukh, et al., Sentiment and sarcasm classification with multitask learning, IEEE Intell. Syst. 34 (3) (2019) 38–43, http://dx.doi.org/10.1109/MIS.2019.2904691.
[47] P. Rosso, F. Rangel, I.H. Farías, L. Cagnina, W. Zaghouani, A. Charfi, et al., A survey on author profiling deception and irony detection for the Arabic language, Lang. Linguist. Compass 12 (4) (2018) e12275, http://dx.doi.org/10.1111/lnc3.12275.
[48] J. Karoui, F.B. Zitoune, V. Moriceau, et al., SOUKHRIA: Towards an irony detection system for arabic in social media, Procedia Comput. Sci. 117 (2017) 161–168, http://dx.doi.org/10.1016/j.procs.2017.10.105.
[49] H.A. Nayel, W. Medhat, M. Rashad, BENHA@ IDAT: Improving irony detection in arabic tweets using ensemble approach, in: FIRE, 2019, pp. 401–408, (Working Notes).
[50] T. Ranasinghe, H. Saadany, A. Plum, S. Mandhari, E. Mohamed, C. Orasan, R. Mitkov, RGCL at IDAT: deep learning models for irony detection in Arabic language, 2019, p. 10.
[51] S. Attardo (Ed.), The Routledge Handbook of Language and Humor, Taylor & Francis, 2017.
[52] C.C. Liebrecht, F.A. Kunneman, A.P.J. van Den Bosch, The perfect solution for detecting sarcasm in tweets# not, 2013.
[53] F. Kunneman, C. Liebrecht, M. Van Mulken, A. Van den Bosch, Signaling sarcasm: From hyperbole to hashtag, Inf. Process. Manage. 51 (4) (2015) 500–509.
[54] K. Hallmann, F.A. Kunneman, C.C. Liebrecht, A.P.J. van den Bosch, M.J.P. van Mulken, Sarcastic soulmates: Intimacy and irony markers in social media messaging, 2016.
[55] H. Mubarak, A. Rashed, K. Darwish, Y. Samih, A. Abdelali, et al., Arabic offensive language on Twitter: Analysis and experiments, 2020, arXiv:2004.02192 [cs], avr., Consulté le: avr. 16, http://arxiv.org/abs/2004.02192.
[56] H. Mohaouchane, A. Mourhir, N.S. Nikolov, et al., Detecting offensive language on arabic social media using deep learning, in: 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 2019, pp. 466–471, http://dx.doi.org/10.1109/SNAMS.2019.8931839.

[57] B. Liu, M. Hu, J. Cheng, et al., Opinion observer, in: Proceedings of the 14th International Conference on World Wide Web - WWW '05, Vol. 342, 2005, http://dx.doi.org/10.1145/1060745.1060797.
[58] M. Aly, A. Atiya, et al., LABR: A large scale arabic book reviews dataset, in: The 51st Annual Meeting of the Association for Computational Linguistics, 2013, pp. 494–498.
[59] H. Mubarak, K. Darwish, W. Magdy, et al., Abusive language detection on arabic social media, in: Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 2017, pp. 52–56, http://dx.doi.org/10.18653/v1/W17-3008.
[60] I. Alsiyat, S. Piao, Metaphorical expressions in automatic arabic sentiment analysis, 2020.
[61] C. Van Hee, E. Lefever, V. Hoste, et al., Semeval-2018 task 3: Irony detection in english tweets, in: Proceedings of the 12th International Workshop on Semantic Evaluation, 2018, pp. 39–50, http://dx.doi.org/10.18653/v1/S18-1005.