Received: February 28, 2022. Revised: May 4, 2022.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Key Challenges and Proposed Solutions to Design Sentiment Analysis System

Ahmed H. Aliwy[1]* **Ayad R. Abbas[2]** **Mustafa J. Hadi[2]**

[1] *University of Kufa, Iraq*
[2] *University of Technology- Iraq, Iraq*
* Corresponding author's Email: ahmedh.almajidy@uokufa.edu.iq

Abstract: Sentiment analysis (SA) is one of the most important tasks in the natural language processing (NLP) field. Many researchers have been trying to build an efficient SA system for many applications such as detection of terrorist activities, customer support management, analyzing customer feedback, market research, competitive research, and many others. Almost all these researches deal with a classification task, and they tried to solve one or two of the challenges that face this task. In this research, a complete SA system was constructed for processing five main challenges in SA and studying the effect of each one on the system. These challenges are the processing of negation, multi-polarity of words, text multi-polarity, semantically ambiguous words (exact meaning), and sarcasm. Some solutions were introduced for these challenges for getting high accuracy. It is the first work for studying the effect of these five challenges collectively with novel solutions to some of them. For aspect-based SA, three different classifiers were used; support vector machine (SVM), maximum entropy (MaxEnt), and long short-term memory (LSTM). The best results for f-measure were 0.833, 0.852, and 0.875 using the LSTM method on Foursquare ABSA, SemEval2014 (laptop and restaurant) datasets, respectively. We induced the most effective negation processing on the SA system from many test scenarios, followed by multi-polarity of words, semantically ambiguous words, text multi-polarity, and sarcasm, respectively. Also, the proposed sarcasm processing technique was evaluated on two annotated corpora with sarcasm, iSarcasm, and Rillof datasets, before applying it on SA.

Keywords: Aspect based sentiment analysis, Multi-polarity detection, Sarcasm detection.

1. Introduction

The text, in general, has two types of information: factual information and opinions information. Factual information is the information that solely deals with facts. On the other side, opinion information is what a person believes or thinks about something. The fact can be proven, but the opinion cannot. Most information processing methods such as web search and text mining work with factual information. The factual statement can imply opinion. In recent decades, many researchers have analyzed this factual text and extracted their opinions. This produced a new field called sentiment analysis, opinion mining, sentiment mining, or subjective analysis, but the famous term is sentiment analysis.

Sentiment analysis (SA) is required for many applications [1]. For example, what the people think about a new phone, device, a candidate, or issue. Also, according to SA, some events can be predicted, such as election outcomes or market trends. Also, it can be used for the enhancement of online learning for students [2]. SA can be a classification problem that has three levels; (i) simple task of positive-negative classification, (ii) range task between two values of most positive and the most negative, and (iii) advanced task that detects the target source and other information. There are many challenges in all three types of sentiment analysis: irony or sarcasm, the negation with its range, semantically ambiguous word, multi-polarity of the word, and text multi-polarities.

The first challenge in SA is iron or sarcasm; it is a sentiment where a person means the opposite of what he says/writes [3, 4]. In the case of speech, it is easy to detect from tonal and gestural clues, but in text, it is very difficult to be detected. The second challenge is negation detection with its range.

Received: February 28, 2022. Revised: May 4, 2022.

Negation is a way of reversing the polarity of words, phrases, and even sentences where the sentiment polarity of an element means the orientation of the expressed opinion, such as positive, negative, and neutral opinion[5]. This scope should be determined because the scope of negation ranges from changing word polarity to complete sentences. It is a challenging task in many situations. The third challenges are the semantically ambiguous word. In this case, the word has multiple meanings (senses); for example, the word “bank” is “official bank” or “riverside”. In such a situation, the exact meaning of this word should be known for getting good results in SA. The fourth challenge is the multi-polarity of words (ambiguous words in polarity). The ambiguous word in polarity has dual aspects according to the context or the subject. For example, the polarities of “high quality” and “high price”, for the word “high”, are positive and negative, respectively. This case is called the multi-polarity of the word. The fifth challenge is the multi-polarity text, where the text has multiple-polarity for multiple entities; hence, the whole SA is misleading. For example, we find positive and negative polarities in the same text. Some researchers tried to solve these challenges as standalone tasks, and others attempted to solve some of them, but not all, as part of the SA system. According to our understanding, all the known works are not complete SA systems because they did not solve or deal with all these challenges and problems.

In this work, a complete SA system will be implemented that solves all these challenges together and studies the effect of each challenge separately on SA performance. The contribution of this work can be summarized by: (i) These five challenges are studied collectively for the first time, (ii) Expanding the text using synonyms and antonyms of the exact meaning of the semantically ambiguous words, (iii) Detection of active negation and the range of active negation using syntactic parse tree, (iv) Using novel methodology for multi-polarity of words by considering all words have a degree of multi-polarity and assigning two vectors for each word for the degree of positivity and negativity, and (v) Using hierarchical structure with a probability distribution for aspect-based SA.

The remainder of this paper is organized as follows. Section 2 reviews in detail prior work in SA. Section 3 presents the proposed system. Section 4 presents all experimental results. The last section presents conclusions.

2. Related works

Many works tried to construct a whole SA system. Some of these works that solved one or more problems are mentioned in this section.

There are many works in the traditional SA system, but samples of these works will be mentioned. A SA [6] was implemented with the assistance of a general-purpose sentiment lexicon. The effect of this lexicon was tested and studied, among five lexicons, on two datasets. They did not deal with any challenges of SA problems. Despite this work on sentence-level and document-level, independent tests were done where document-level does not depend on sentence-level. A deep convolution neural network [7] was used for sentiment analysis with a word embedding feature. The proposed system was evaluated on five twitter datasets. Word embedding with n-grams and word sentiment polarity score were combined as a feature for the SA system. They also did not deal with any challenge except negation. Also, despite using syntactic features, they deal with negation for sentence range but not for the true range as in the parse tree. A bidirectional LSTM [8] was proposed as a sentiment analysis approach where the weights of the words are tf-idf. Skip-Gram model trained the word vectors. They compared the proposed system with four techniques In addition, they took SA as a classification problem without regard to any one of the challenges. A feature ensemble model [9] was presented as a sentiment analysis for tweets with a fuzzy opinion. They used two datasets; the first is a private dataset downloaded from Twitter, and the second is downloaded from the Kaggle website. They focus on features such as word types but not the challenges of SA such as word-ambiguity. Also, they did not deal with any challenge except the negation, where the range of negation was applied to the whole sentence despite using a parse tree. A sentiment analysis [10] was implemented using different versions of recurrent neural networks such as Bi-GRU and Bi-LSTM. The used dataset was the Amazon review dataset. In all of these and many other works, the SA system was implemented as a classification task without any regard for a solution for the main challenges of SA except the negation challenge. Also, they did not solve any one of the SA problems.

For the multi-polarity (text and word) challenge, a BiLSTM model [11] was used with multi-polarity orthogonal attention for implicit sentiment analysis. They used three implicit and explicit sentiment analysis datasets for the evaluation process. A lexicon-based word polarity identification method was proposed on several customer reviews datasets

[12]. The used multi-polarity word identification was based on measuring semantic relatedness at the fragment level. Furthermore, they used synonyms expansion on the target word. A SA [13] was proposed based on multi-person multi-criteria decision-making. They used a deep learning model on the TripR-2020 dataset and tried to solve the text multi-polarity problem for multiple entities. In addition, they did not deal with negation or the sense of the ambiguous word. Also, they take the SA as a task without concentration on the other challenges.For semantic and word sense processing, semantic analysis [14] was used by extracting the synonyms of a word, without regard to its exact meaning, as an assistant to SA of suicide sentiment prediction in social media. The classifiers that were used for SA were SVM, Max-Entropy, and NB using the WEKA tool on a privately collected dataset of 892 tweets. They dealt with semantic challenges only but not the other challenges. A linguistic knowledge [15] was used for language representation as preprocessing for SA. They used SentiWordNet as a lexical resource and Yelp-dataset-Challenge as a dataset. Finally, they dealt with word meaning ambiguity challenge only but not the other challenges. Supervised approaches of five classifiers (decision tree, a linear regression, a Lasso regression, a Ridge regression, and support vector regression) [16] were used for SA. The polarity was a real value between -1 (most negative) to +1 (most positive) with the lexical-semantic feature. They tested the system on two datasets taken from SemEval-2017 and dealt with word meaning ambiguity challenge only but not the other challenges. A representation model [17] was proposed using word-level linguistic knowledge and sentiment polarity to context-aware sentiment attention mechanism. The linguistic knowledge was introduced from SentiWordNet. In addtion, they used the sentence-level and aspect-level SA models and dealt with word meaning ambiguity challenge only but not the other challenges. Also, they did not used aspect-level for whole document level SA.

A sentiment analysis [18] was proposed for entity-level for aspect-based challenges. It tried to solve text multi-polarity by classifying the entities and sentiment. They used a private labeled dataset of 3,000 issue comments derived from 10 open-source projects. In addition, they tried to solve text multi-polarity and negation only without considering the range of negation or the other challenges. A CapsNet and CapsNet-BERT models [19] were used for sentiment analysis of text multi-polarity. They evaluated their approach on two datasets; MAMS and SemEval-14 restaurant reviews. Each sentence contains at least two different opinions on different aspects. They tried to solve text multi-polarity and negation only. A gradual machine learning [20] was proposed for aspect-level SA to solve low resources. They compared the proposed work with a deep neural network on ACSA and ATSA data tasks. They triedto solve text multi-polarity and negation only where the range of negation is the complete sentence. A propositional logic proposed a human-interpretable learning approach [21] for aspect-based sentiment analysis. They used SemEval 2014 dataset for evaluation without significant processing of the SA challenges. They tried to solve text multi-polarity and negation without considering the range of negation or the other challenges.

A deep neural network [22] was used as a multitask learning setting for the sarcasm and sentiment analysis for the sarcasm challenge. The used dataset was small samples annotated with sentiment tag where 35% of these samples were sarcastic. A model based on the Google BERT method [23] was proposed to detect the text's sarcasm as a standalone task. They used Twitter and Reddit conversion datasets. They compared their works with four other techniques. An artificial neural network [24] was used for detecting sarcasm in SA. They used three small datasets of reviews taken from drug, car, and hotel sites. Finally, a memory network [25] was proposed using sentiment semantics to capture sarcasm expressions. They used IAC-V2, IAC-V2, and Twitter datasets for the evaluation. A model based on the attention-based neural model [26] was proposed to detect the text's sarcasm as a standalone task. They used six benchmark datasets from Twitter, Reddit, and the internet argument corpus. They also did not deal with the other SA challenges. ISarcasm dataset [27] was introduced to solve the limitation of other datasets for the sarcasm detection task. They tried to solve the sarcasm problem in SA without regard to the different challenges of SA. All the works [22,23,24,25,26,27] tried to solve sarcasm problems without the other challenges.

All these works did not solve all the mentioned challenges. In our work, we try to solve all these challenges and suggest a solution for each challenge to improve SA performance.

3. The proposed system

Our system is a comprehensive sentiment analysis system that solves the most famous problems such as; (i) sarcasm detection, (ii) negation range detection, (iii) semantically ambiguous word detection, and (iv) multi-polarity detection of text and word. A suggested solution for each problem is explained in this section. The system has three

Received: February 28, 2022. Revised: May 4, 2022.

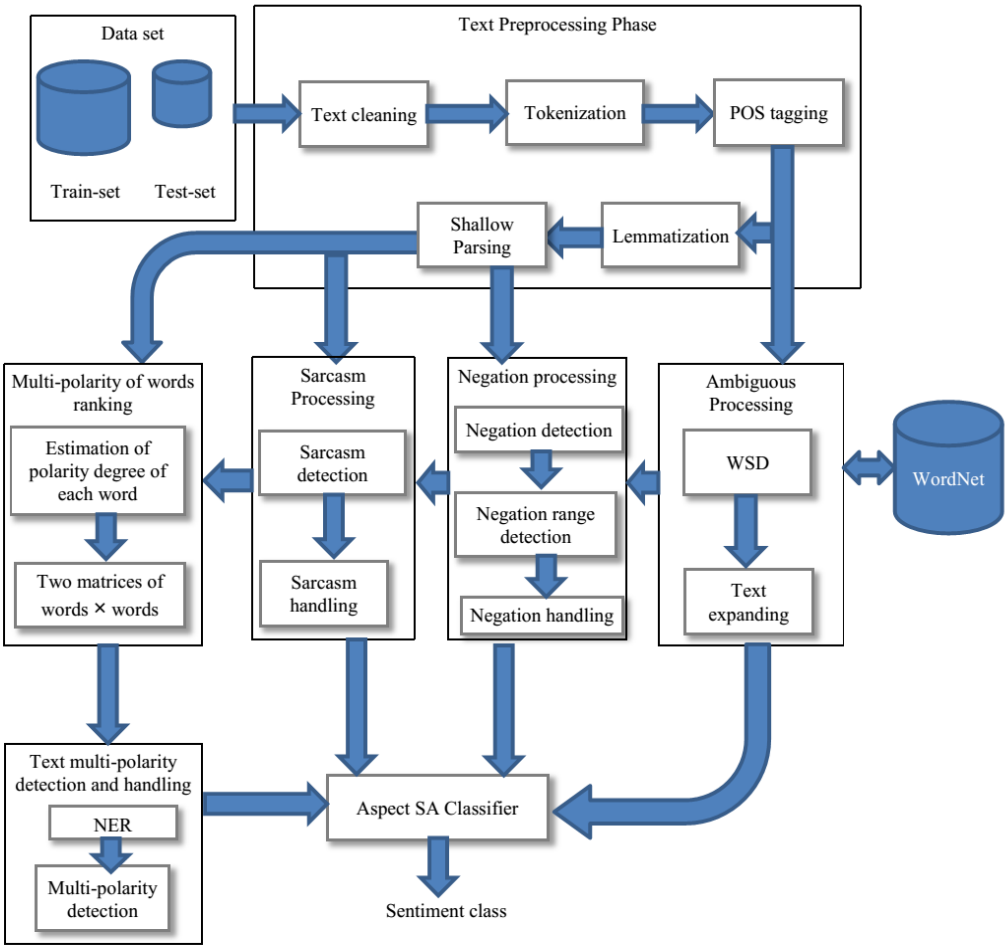

Figure. 1 Block diagram of data initialization and representation phase

distinct phases; (i) text initialization, (ii) challenge processing (detection and handling), and (iii) Sentiment analysis classifier. The whole system is shown in Fig. 1.

3.1 Text preprocessing phase

In most NLP applications, the text should be initialized before further processing. This phase is a very important step with a little difference in the details according to the application. In this work, this phase consists of five steps that are applied sequentially: (i) text cleaning, (ii) tokenization, (iii) POS tagging, (iv) lemmatization, and (v) shallow parsing. From Fig. 1, it is clear that shallow parsing is not used for semantically ambiguous word detection and handling. All these steps are well-known by the researchers in natural language processing (NLP); therefore, they will be explained briefly.

In the first step, text cleaning, the text is cleaned by removing some symbols, empty lines, URLs, repeated whitespaces, etc. This step is very important for reducing the noise in the text and removing some of the useless strings, which may affect the performance of the subsequent steps. In our work, this step is done using regular expressions. After cleaning the text, it is tokenized (second step) into sentences and tokens. In most cases, the tokens will be a valid word. The third step is part of speech (POS) tagging, which assigns a POS tag for each token. POS tags give more information about the words in the context. This step is useful for lemmatization, shallow parsing, and SA classifier. This step is done, in this work, using Stan-ford POS tagger [28]. The fourth step is lemmatization which extracts the lemma from a word. It is very important in many applications that deal with the compositional or lexical-semantic. This step is done using WordNetLemmatizer as part of NLTK library in Python language. The final step of preprocessing phase is shallow parsing. It assigns a parsing tree for the phrases or sentences. Stanford parser is used for this step [29].

3.2 Challenges detection and handling

In this phase, the main challenges that face the SA system are detected and processed for improving the SA system. These challenges are (i) exact meaning of ambiguous word detection and processing, (ii) negation range detection, (iii) multi-polarity detection of text and words, and (iv) sarcasm detection.

3.3 The exact meaning of ambiguous word detection and handling

Many words in natural languages have more than one meaning (multiple senses). These words are called semantically ambiguous words. Detection of these ambiguous words and extracting their exact meaning are useful for many applications such as machine translation, information retrieval, SA, and many others. The SA system can be improved by extracting the exact meaning of the ambiguous words where this task is called word sense disambiguation (WSD). We used the Lesk algorithm with annotated corpus with the assistance of English WordNet as lexical for WSD in the same methodology that was used by [30]. When the exact meaning of the ambiguous word is extracted, the synonyms and antonyms of this sense will be added to the text where the antonyms will be prefixed by the "not" word. This operation will increase the possibility of nearby the true polarity and far from the false polarity. This methodology was introduced by [31].

3.4 Negation range detection and handling

Negation, in most cases, is used to reverse the polarity of a word, phrase, or sentence. Detection of negation is an easy task, but detection of the scope of negation is more difficult because the scope of the negation can be extended to more than one sentence. Processing of negation is the basic step in any sentiment analysis system.

	t_1	t_2	…	t_v
t_m	wp_{m1}	wp_{m2}	….	wp_{mv}

(a)

	t_1	t_2	…	t_v
t_m	wn_{m1}	wn_{m2}	….	wn_{mv}

(b)

Figure. 2 The two polarity vectors for term tm: (a) vector of term t_m from positive texts and (b) vector of term t_m from negative texts

In this work, the negation detection and handling are done by four steps; (i) negation detection, (ii) determining the effective negation and ineffective negation, (iii) detecting the scope of negation, and (iv) negation handling.

Negation detection is done using the negation words such as not, no, nor, neither … etc., or by prefixes such as dis_, un_, ir_ … etc. with very few exceptions. According to the small dictionary, these words will be classified into effective negation or ineffective. For example, the words "not only", "not to mention", and "not just" are samples of ineffective negation, and they will be neglected or excepted in our system. The other cases will be an effective negation, and they should be handled. When the effective negation is identified, the scope of this negation will be done using POS tagging and syntactic parsing for producing the parse tree. This pars tree will limit the scope according to the connection of negation in the parse tree. i.e morphological and syntactic assistance will be used for detection of the scope of negation. The negation will be handled by adding the prefix "not_" to all words in the scope of negation. Also, their synonyms, produced by semantically ambiguous word detection and handling tasks, are prefixed by "not_".

3.5 Multi-polarity detection

There are two types of multi-polarity; multi-polarity of words and text multi-polarity. Multi-polarity of words means that words have dual use as negative and positive depending on the domain and context. For example, the polarities of "high quality" and "high price", for the word "high", are positive and negative, respectively. in Text multi-polarity, the text contains praise and disparage for two entities or aspects. We propose a novel method for detecting and processing the multi-polarity of words.

3.5.1. Multi-polarity of words (MPW) detection

Almost all the researchers that deal with MPWs, tried to extract MPWs from multiple domains and fields and record them for SA. In our research, we will violate this trend and consider that all the words have multi-polarity, where each word will have a degree of polarity according to the context. This method is used for the first time, according to our knowledge. Determining the degree of polarity of the word is done using the adjacent and close words within the negative and positive texts, and each word will have two vectors. The first vector represents and reflects the word's appearance in positive texts, while the second vector represents the appearance of the word in negative texts. For example, if we have a vocabulary containing v words, then the two vectors for word tm will be shown in Fig. 2.

Where wp_{mi} represents the occurrence weight of the term m (tm) with term i (ti) in positive text, this weight can be term frequency, tf-idf, probability, or any other weight. We used the following equations for estimating the weight.

$$wp_{ij} = \frac{\#\ occurance\ of\ terms\ i\ with\ j\ in\ positive\ text}{\#\ occurance\ of\ terms\ i\ in\ positive\ text} \quad (1)$$

$$wn_{ij} = \frac{\#\ occurance\ of\ terms\ i\ with\ j\ in\ negative\ text}{\#\ occurance\ of\ terms\ i\ in\ negative\ text} \quad (2)$$

If all words are considered, two matrices will be constructed for the same words: positive texts and negative ones, as shown in Fig. 3.

We should see that wpii and wnii are neglected, and they take the value of 0 because they have no sense. All these weights are estimated using skip-gram. The weight of each word will be extracted from these two matrices according to the text of the test document. If the test document has a set of words ℓ, then the weight of each word can be estimated by:

	t_1	t_2	…	t_v
t_1	wp_{12}	wp_{12}	…	wp_{1v}
t_2	wp_{21}	wp_{22}	…	wp_{2v}
…	…	…	…	
t_v	wp_{v2}	wp_{v2}	…	wp_{vv}

(a)

	t_1	t_2	…	t_v
t_1	wn_{12}	wn_{12}	….	wn_{1v}
t_2	wn_{21}	wn_{22}	….	wn_{2v}
…	…	…	…	…
t_v	wn_{v2}	wn_{v2}	….	wn_{vv}

(b)

Figure. 3 The two matrices vectors for all words in the vocabulary v.: (a) vectors that are produced from positive texts and (b) vectors that produced from negative texts

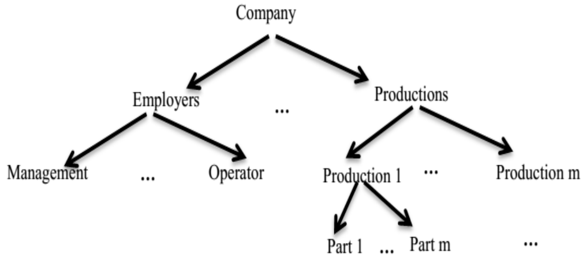

Figure. 4 Hierarchical relations for a company

$$wp_i = \sum_{j\in\ell} wp_{ij} \quad (3)$$

$$wn_i = \sum_{j\in\ell} wn_{ij} \quad (4)$$

Where wp_i, and wn_i are weights of the word i for positive and negative classes, respectively.

The weight of each word will be varied according to test text because it will be estimated according to the summation of all weights of this word that appear with the test word as in Eqs. (3) and (4). Hence the degree of polarity of each word will be included in these weights. These weights are used in SA for predicting the class as positive or negative.

3.5.2. Text multi-polarity detection and processing

In the case of text multi-polarity detection, the task is more difficult than multi-polarity of words detection. This is because the texts may have positive and negative polarities for different aspects in the same text. Hence, estimating or predicting whole text polarity is challenging or impossible in some situations, such as unrelated aspects. For example, if the text has two aspects, such as display and processor for one mobile phone, these are related aspects. Also, if the text reviews an employer in a company and a device produced by the same company, they are associated aspects. In this case, the polarity for the parent thing (maybe the whole text) can be estimated. But if the employer and device are not to the same company, they are none related, and extracting the whole polarity of text is challenging and has no sense in most situations. The second type is out of range of this study.

Firstly, the type of aspect should be known with their hierarchical relations. For example, Fig. 4 shows the hierarchical relation for a company example.

Each node in this hierarchical structure has a weight that reflects the effect of this node on making the final decision. These weights can be recorded manually, estimated automatically, or uniformly distributed probability. In this research, uniform probability distribution was chosen for the weights.

For calculating the text polarity, the following steps will be done:

- Extract the test text aspects by executing named entity recognition (NER).
- The SA classifier will be executed to estimate each aspect's polarity for each aspect. This step should be fed to the multi-polarity of words, negation processing, and sarcasm.
- Each classification of positive polarity for an aspect, the weight of this aspect will be recorded with a positive sign, but it will be assigned a

Received: February 28, 2022. Revised: May 4, 2022.

negative sign if the classification gives negative polarity as a class.

- The positivity degree of a text will result from the multiplication of all positive aspect weights.
- The negativity degree of a text will result from the multiplication of all negative aspect weights (absolute value).

For example, negativity and positivity degrees can be estimated for mobile phones in one text as in Fig. 5.

3.6 Sarcasm detection and processing

In sarcasm, the texts have a meaning different from the apparent meaning. Therefore, detecting sarcasm is one of the most challenging tasks in SA. However, in the case of speech, the problem is relatively easier than text because there are lots of indications such as facial features or tone of voice. In this work, three stages for sarcasm are done; (i) sarcasm detection, (ii) sarcasm range detection (aspect); and (iii) sarcasm processing.

For detecting sarcasm, many features are used; (i) emoji, (ii) emoticons, (iii) special symbols, (iv) conflicts in the text, and (v) compatibility of hashtags with content. All these features are recorded from labeled data with sarcastic. Emoji, emoticons, and special symbols represent the number of positive and negative emoji, emoticons, and special symbols, respectively. The conflicts detection is accomplished, in this work, using antonyms in the same text. It is a number that represents the existing conflict (i.e. number of antonyms) in the text.

The compatibility of hashtags with content is done by splitting the hashtag words using the maximum match algorithm that is used for the Chinese language with simple modification for manipulating underscores and numbers. Two features for hashtags are used where the first represents a number of compatible hashtags and the second represents incompatible hashtags. The compatibility of the hashtag with the text is done by comparing words of hashtags and their synonyms with the content of the text.

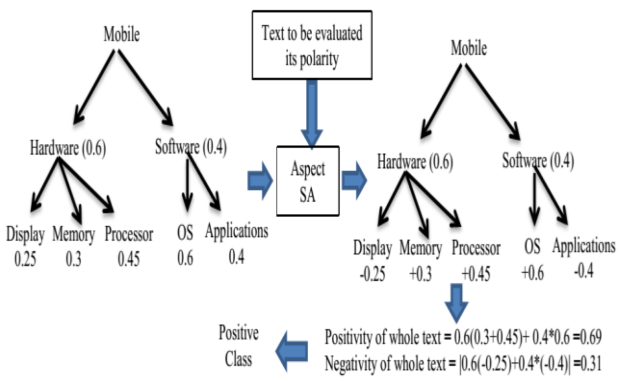

Figure. 5 Negativity and positivity degree of mobile phone example using hierarchal entities

3.7 Learning and classification phase

Three well-known classifiers were used, in this work, as SA classifiers. These classifiers are long short term memory (LSTM), maximum entropy (ME), and support vector machine (SVM). All these classifiers are learned from the same training data set. Maximum entropy (MaxEnt) is a "general and intuitive way for estimating a probability from data and it has been successfully applied in various natural language processing tasks" [32]. It belongs to the family of classifiers called log-linear classifiers.

Long short term memory (LSTM) was introduced by [33]. LSTM is a type of RNN network that can grasp long-term dependence. It is used in many different tasks such as text classification, speech recognition, sentimental analysis, etc. In this work, a deep learning model is built using LSTM for the aspect sentiments analysis task.

SVM is a supervised machine learning algorithm that can be used for both classification and regression challenges. SVM maps training examples to points in space to maximize the width of the gap between the two categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. It was implemented, in this work, based on [34].

4. Experimental results and evaluation

In this section, a brief explanation of the experimental setting, the used dataset, and the final results will be shown. Firstly the used dataset and the experimental setting will be shown then the efficiency of the sarcasm detection, and handling will be evaluated. Finally, the evaluation and results of the overall system will be presented with/without the main challenges of processing. We used Stanford packages for the preprocessing stage, such as tokenization, POS tagging, syntactic parser, and named entity recognition [28, 29]. For lemmatization, WordNetLemmatizer as part of the NLTK library is used.

The evaluation metrics for the proposed system are precision (P), recall (R), and f-measure (F), as shown below.

$$\mathrm{R} = \frac{TP}{TP+FN} \tag{5}$$

$$\mathrm{P} = \frac{TP}{TP+FP} \tag{6}$$

Received: February 28, 2022. Revised: May 4, 2022.

$$F = 2 \times \frac{P \times R}{P+R} \quad (7)$$

Where TP, TN, FP, and FN are true positive, true negative, false positive, and false negative, respectively.

4.1 Experimental setting and dataset

The proposed system was tested on a Windows operating system of 64-bits, 8 GB Ram, and Intel Core i7 processor. Python 3.5 was used as a programing language with two important libraries NLTK and scikit-learn.

We used four datasets; two for testing the sarcasm and two for testing the overall system. ISarcas [27] and Rillof [34] datasets are used for sarcasm evaluation. iSar-casm contains 4,484 tweets, out of which 777 are labeled as sarcastic and 3,707 as non-sarcastic. Rillof dataset [35] includes 3000 tweets annotated with two classes' sarcasm and none-sarcasm. The other two datasets were Foursquare ABSA [36] dataset and the SemEval-2014 dataset [37]. Foursquare ABSA is random samples manually annotated dataset from Foursquare comments. It contains 1006 English sentences for the restaurant domain that are annotated with the SemEval2014 annotation guidelines. It contains 12 semantic classes for the restaurant domain. SemEval-2014 dataset is an aspect-based that is manually annotated. It consists of 3045 and 3041 English sentences from laptop and restaurant reviews.

Also, English WordNet was used as lexical semantics for providing the sense set of each ambiguous word and senses examples for learning the WSD.

4.2 Results and evaluation

The sarcasm detection and handling is evaluated on Isarcasm [27] and Rillof [35]datasets. Table 1 shows the Precision, Recall, and F-measure using 10-fold cross-validation (average) with the three classifiers; LSTM, SVM, and MaxEnt, for our model. In addtion, the results of some related works that used the same data set are shown in Table 1. The highest values are highlighted with bold font. As can be seen from Table 1, using the LSTM method gives the best performance. Moreover, this result proves the effectiveness of our model for the sarcasm detection task.

If all words are considered, two matrices will be constructed for the same words: positive texts and negative ones, as shown in Fig. 3.

For studying the effect of each challenge in SA, the overall system is tested and evaluated using Foursquare ABSA [36] and SemEval-2014 [37]datasets with seven scenarios; (i) simple SA system where none of the challenges will be processed, (ii) overall proposed system without negation processing, (iii) overall proposed system without ambiguous word processing, (iv) overall proposed system without multi-polarity of words processing, (v) overall proposed system without text multi-polarity processing, (vi) overall proposed system without sarcasm processing, and, (vii) overall proposed system with the processing of all the challenges.

Tables 2 to 8 show the precision, recall, and F-measure (average of 10-fold cross-validation) for all these scenarios using four classifiers as aspect-based sentiment classifiers on the two datasets; Foursquare ABSA [36] dataset and SemEval-2014 dataset [37]. The polarity of the whole text was estimated according to the methodology explained in section 3.2. Table 8 contains the results of previous works that used the same datasets. Solving all the problems gives the highest results compared to the other works that solve part of these problems. As a result, the experimental results reveal that our proposed method achieved improved performance of SA compared to that of state-of-the-art so far.

The results show that processing the main challenges collectively gives higher precision than those obtained without processing all the problems. The difference can be seen in Table 2 and Table 8. If we take Table 8 as reference results where all the problems are solved, we can see the effect of neglecting each problem separately. By comparing the results of Tables 3, 4, 5, 6, and 7, with the results of Table 8, it is obvious that the most effective problem is negation processing because the difference between the result of Table 3 and Table 8 is the highest. The second effective problem is the multi-polarity of words processing. The third and fourth effective problems are ambiguous word processing and text multi-polarity processing. The last effect is sarcasm processing, as shown by comparing Table 7 and Table 8. The LSTM techniques gave the highest f–measure in almost all the tests. Also, all the classifiers gave low scores on the ABSA dataset for many reasons, but the main reasons are the dataset itself and the small size. Also, some tests have a reduction in precision more than recall, which means that the false positive is increased and the false negative is decreased, classifying the negative polarity text by positive polarity.

Because most subtasks were processed using packages, accumulative errors negatively affected the results.

The proposed sarcasm detection and handling was evaluated on two annotated corpora with sarcasm and gave good results for one and acceptable results for the other dataset (Table 1).

Also, many errors were recorded from the Stanford POS tagger, lemmatizer, and parser from testing the system steps manually. These errors affect the entire related sub-task and hence the whole system.

5. Conclusion

For constructing an efficient SA system, almost all the challenges that face the system should be solved. In this work, the main challenges in SA were studied. These challenges are multi-polarity of text and words, semantically ambiguous words, negation, and sarcasm. The effect of each one on the SA system was studied. It turns out that the most influential is

Table 1: Evaluation of sarcasm detection and handling on two datasets (average of 10-fold cross-validation)

Reference	Algorithm	Isarcasm Dataset			Rillof Dataset		
		P	R	F	P	R	F
[A]	NBOW	-	-	-	0.713	0.624	0.641
	Vanilla CNN	-	-	-	0.710	0.671	0.686
	Vanilla LSTM	-	-	-	0.673	0.67	0.673
	Attention LSTM	-	-	-	0.688	0.687	0.687
	SIARN	-	-	-	0.663	0.647	0.654
	MIARN	-	-	-	0.698	0.667	0.678
[B]=[33]	LSTM	0.217	0.747	0.336	-	-	0.669
	Att-LSTM	0.260	0.436	0.325	-	-	0.679
	CNN	0.261	0.563	0.356	-	-	0.681
	SIARN	0.219	0.782	0.342	-	-	0.741
	MIARN	0.236	0.793	0.364	-	-	0.712
	3CNN	0.250	0.333	0.286	-	-	-
	Dense-LSTM	0.375	0.276	0.318	-	-	-
Our model	MaxEnt	0.500	0.714	0.588	0.870	0.859	0.864
	SVM	0.420	0.750	0.538	0.844	0.783	0.812
	LSTM	**0.540**	**0.730**	**0.621**	**0.896**	**0.784**	**0.836**

Table 2: Results of implementing SA system without processing any one of the challenges

Algorithm	Foursquare ABSA			SemEval2014					
				Laptop			Restaurant		
	P	R	F	P	R	F	P	R	F
MaxEnt	0.39	0.375	0.382	0.421	0.472	0.445	0.408	0.49	0.445
SVM	0.39	0.406	0.398	0.421	0.496	0.455	0.414	0.479	0.444
LSTM	0.41	0.482	0.443	0.444	0.495	0.468	0.421	0.489	0.452

Table 3: Results of implementing SA system without processing negation only.

Algorithm	Foursquare ABSA			SemEval2014					
				Laptop			Restaurant		
	P	R	F	P	R	F	P	R	F
MaxEnt	0.564	0.572	0.568	0.719	0.769	0.743	0.721	0.736	0.728
SVM	0.55	0.563	0.556	0.713	0.693	0.703	0.746	0.786	0.765
LSTM	0.652	0.67	0.661	0.746	0.776	0.761	0.749	0.759	0.754

Table 4: Results of implementing SA system without ambiguous word processing.

Algorithm	Foursquare ABSA			SemEval2014					
				Laptop			Restaurant		
	P	R	F	P	R	F	P	R	F
MaxEnt	0.68	0.747	0.712	0.819	0.761	0.789	0.852	0.766	0.807
SVM	0.69	0.742	0.715	0.796	0.84	0.818	0.852	0.778	0.813
LSTM	0.73	0.802	0.764	0.816	0.864	0.839	0.875	0.839	0.857

Received: February 28, 2022. Revised: May 4, 2022.

Table 5: Results of implementing SA system without processing Multi-Polarity of words only.

Algorithm	Foursquare ABSA			SemEval2014					
				Laptop			Restaurant		
	P	R	F	P	R	F	P	R	F
MaxEnt	0.63	0.724	0.674	0.799	0.757	0.778	0.839	0.759	0.797
SVM	0.61	0.718	0.659	0.763	0.841	0.8	0.806	0.763	0.784
LSTM	0.71	0.798	0.751	0.796	0.861	0.827	0.836	0.844	0.84

Table 6: Results of implementing SA system without text multi-polarity processing

Algorithm	Foursquare ABSA			SemEval2014					
				Laptop			Restaurant		
	P	R	F	P	R	F	P	R	F
MaxEnt	0.67	0.736	0.702	0.819	0.761	0.789	0.855	0.765	0.807
SVM	0.71	0.747	0.728	0.789	0.839	0.814	0.849	0.779	0.813
LSTM	0.69	0.793	0.738	0.819	0.865	0.841	0.868	0.841	0.854

Table 7: Results of implementing SA system without Sarcasm processing.

Algorithm	Foursquare ABSA			SemEval2014					
				Laptop			Restaurant		
	P	R	F	P	R	F	P	R	F
MaxEnt	0.75	0.781	0.765	0.819	0.769	0.793	0.859	0.772	0.813
SVM	0.76	0.776	0.768	0.809	0.86	0.834	0.865	0.783	0.822
LSTM	0.81	0.802	0.806	0.822	0.862	0.842	0.888	0.852	0.87

Table 8: Results of implementing SA system with processing all the problems

Reference	Algorithm	Foursquare ABSA			SemEval2014					
					Laptop			Restaurant		
		P	R	F	P	R	F	P	R	F
[19]	TextCNN	-	-	-	-	-	-	-	-	0.859
	LSTM	-	-	-	-	-	-	-	-	0.846
	TD LSTM	-	-	-	-	-	-	-	-	0.845
	AT LSTM	-	-	-	-	-	-	-	-	0.812
[21]	CRF & Max-Ent	-	-	0.569	-	-	-	-	-	0. 630
Our model	MaxEnt	0.78	0.804	0.792	0.832	0.774	0.802	0.875	0.778	0.824
	SVM	0.81	0.794	0.802	0.829	0.86	0.844	0.878	0.792	0.833
	LSTM	0.85	0.817	0.833	0.842	0.862	0.852	0.895	0.855	0.875

negation, followed by multi-polarity of words, ambiguous words, text multi-polarity, and sarcasm, respectively. The least effective on the SA system was the sarcasm challenge. The suggested SA system used most NLP basic techniques such as text cleaning, tokenization, POS tagging, lemmatization, and shallow parsing. These processes are very important for getting an efficient SA system because of the nature of natural languages. Also, it is obvious from this study determining the polarity for the whole text (review) can be achieved by determining the polarity of aspects. Using tuned probabilities for the aspects hierarchically was suitable for this purpose. Improving POS tagger and lemmatizer will enhance the performance of WSD and the whole SA system. Also, Improving the shallow parsing will enhance the performance of negation domain detection, sarcasm, and multi-polarity processing and hence the whole SA system.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, Ahmed; methodology, Ahmed; software, Ahmed and Ayad; validation, Ahmed and Mustafa ; formal analysis, Ahmed, Ayad and Mustafa; investigation, Ahmed; resources, Ahmed; data curation, Ahmed; writing original draft preparation, Ahmed, Ayad and Mustafa; writing review and editing, Ahmed,Ayad and Mustafa; visualization, Ahmed; supervision, Ahmed.

Received: February 28, 2022. Revised: May 4, 2022.

References

[1] S. Qaiser, N. Yusoff, F. K. Ahmad, and R. Ali, "Sentiment Analysis of Impact of Technology on Employment from Text on Twitter", *International Journal of Interactive Mobile Technologies*, Vol. 14, No. 3, pp. 83-103, 2020.

[2] E. P. Yildiz, M. Tezer, and H. Uzunboylu, "Student opinion scale related to Moodle LMS in an online learning environment: Validity and reliability study", *International Journal of Interactive Mobile Technologies*, Vol. 12, No. 4, pp. 97-108, 2018.

[3] E. Camp, "Sarcasm, pretense, and the semantics/pragmatics distinction", *Nous*, Vol. 46, No. 4, pp. 587-634, 2012.

[4] S. K. Bharti, B. Vachha, R. Pradhan, K. S. Babu, and S. K. Jena, "Sarcastic sentiment detection in tweets streamed in real time: a big data approach", *Digital Communications and Networks*, Vol. 2, No. 3, pp. 108-121, 2016.

[5] E. Çeltek, "Handbook of Research on smart technology applications in the tourism industry,1st Edition", *Business Science Reference*, USA, 2020.

[6] C. S. Khoo and S. B. Johnkhan, "Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons", *Journal of Information Science*, Vol. 44, No. 4, pp. 491-511, 2018.

[7] Z. Jianqiang, G. Xiaolin, and Z. Xuejun, "Deep convolution neural networks for twitter sentiment analysis", *IEEE Access*, Vol. 6, pp. 23253-23260, 2018.

[8] G. Xu, Y. Meng, X. Qiu, Z. Yu, and X. Wu, "Sentiment analysis of comment texts based on BiLSTM", *IEEE Access*, Vol. 7, pp. 515220-51532, 2019.

[9] H. T. Phan, V. C. Tran, N. T. Nguyen, and D. Hwang, "Improving the performance of sentiment analysis of tweets containing fuzzy sentiment using the feature ensemble model", *IEEE Access*, Vol. 8, pp. 14630-14641, 2020.

[10] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, and P. Nagrath, "Sentiment analysis using gated recurrent neural networks", *SN Computer Science*, Vol. 1, No. 2, pp. 1-13, 2020.

[11] J. Wei, J. Liao, Z. Yang, S. Wang, and Q. Zhao, "BiLSTM with multi-polarity orthogonal attention for implicit sentiment analysis", *Neurocomputing*, Vol. 383, pp. 165-173, 2020.

[12] K. Abdalgader and A. A. Shibli, "Experimental Results on Customer Reviews Using Lexicon-Based Word Polarity Identification Method", *IEEE Access*, Vol. 8, pp. 179955-179969, 2020.

[13] C. Zuheros, E. M. Cámara, E. H. Viedma, and F. Herrera, "Sentiment analysis based multi-person multi-criteria decision making methodology using natural language processing and deep learning for smarter decision aid. Case study of restaurant choice using TripAdvisor reviews", *Information Fusion*, Vol. 68, pp. 22-36, 2021.

[14] M. Birjali, A. B. Hssane, and M. Erritali, "Machine learning and semantic sentiment analysis based algorithms for suicide sentiment prediction in social networks", *Procedia Computer Science*, Vol. 113, pp. 65-72, 2017.

[15] P. Ke, H. Ji, S. Liu, X. Zhu, and M. Huang, "SentiLARE: Sentiment-aware language representation learning with linguistic knowledge", In: *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6975-6988, 2019.

[16] A. Dridi, M. Atzeni, and D. R. Recupero, "FineNews: fine-grained semantic sentiment analysis on financial microblogs and news", *International Journal of Machine Learning and Cybernetics*, Vol. 10, No. 8, pp. 2199-2207, 2020.

[17] P. Ke, H. Ji, S. Liu, X. Zhu, and M. Huang, "Sentilare: Linguistic knowledge enhanced language representation for sentiment analysis" In: *Proc. of the 2020 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6975-6988, 2019.

[18] J. Ding, H. Sun, X. Wang, and X. Liu, "Entity-level sentiment analysis of issue comments", In: *Proc. of the 3rd International Workshop on Emotion Awareness in Software Engineering*, pp. 7-13, 2018.

[19] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A challenge dataset and effective models for aspect-based sentiment analysis", In: *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6280-6285, 2019.

[20] Y. Wang, Q. Chen, J. Shen, B. Hou, M. Ahmed, and Z. Li, "Gradual Machine Learning for Aspect-level Sentiment Analysis", *arXiv preprint arXiv:*1906.02502, 2019.

[21] R. K. Yadav, L. Jiao, O. C. Granmo, and M. Goodwin, "Human-level interpretable learning for aspect-based sentiment analysis", In: *Proc. of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*, 2021.

[22] N. Majumder, S. Poria, H. Peng, N. Chhaya, E. Cambria, and A. Gelbukh, "Sentiment and sarcasm classification with multitask learning", *IEEE Intelligent Systems*, Vol. 34, No. 3, pp. 38-

Received: February 28, 2022. Revised: May 4, 2022.

43, 2019.

[23] A. Kalaivani and D. Thenmozhi, “Sarcasm Identification and Detection in Conversion Context using BERT”, In: *Proc. of the Second Workshop on Figurative Language Processing*, pp. 72-76, 2020.

[24] S. Gupta, R. Singh, and V. Singla, “Emoticon and text sarcasm detection in sentiment analysis”, In: *Proc. of the First International Conference on Sustainable Technologies for Computational Intelligence*, pp. 1-10, 2020.

[25] L. Ren, B. Xu, H. Lin, X. Liu, and L. Yang, “Sarcasm detection with sentiment semantics enhanced multi-level memory network”, *Neurocomputing*, Vol. 401, pp. 320-326, 2020.

[26] Y. Tay, A. T. Luu, S. C. Hui, and J. Su, “Reasoning with sarcasm by reading inbetween”, In: *ACL*, pp. 1010–1020, 2018.

[27] S. Oprea and W. Magdy, “isarcasm: A dataset of intended sarcasm”, In: *Proc. of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1279-1289, 2019.

[28] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, “Feature-rich part-of-speech tagging with a cyclic dependency network”, In: *Proc. of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 252-259, 2003.

[29] D. Chen and C. D. Manning, “A fast and accurate dependency parser using neural networks”, In: *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 740-750, 2014.

[30] A. H. Aliwy, A. R. Abbas, and D. C. Wyld, “Improvement WSD Dictionary using annotated corpus and testing it with simplified lesk algorithm”, In: *Proc. of the Fifth International Conf. on Computer Science and Information Technology*, Vol. 5, No. 4, pp. 89-97, 2015.

[31] A. H. Aliwy, “Exact Meaning of the Ambiguous Word for Improving Sentiment Analysis”, *AIP Conference Proceedings*, Vol. 2386, No. 1, 2021.

[32] A. Jain and J. Mandowara, “Text Classification by Combining Text Classifiers to Improve the Efficiency of Classification”, *International Journal of Computer Application*, Vol. 6, No 2, pp. 2250-1797, 2016.

[33] S. Hochreiter and J. Schmidhuber, “Long short-term memory”, *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, 1997.

[34] V. V. Belle, B. V. Calster, S. V. Huffel, J. A. Suykens and P. Lisboa, “Explaining support vector machines: a color based nomogram”, *PloS One*, Vol. 11, No. 10, 2016.

[35] E. Riloff, A. Qadir, P. Surve, L. D. Silva, N. Gilbert, and R. Huang, “Sarcasm as contrast between a positive sentiment and negative situation”, In: *Proc. of 2013 Conf. on Empirical Methods in Natural Language Processing*, pp. 704-714, 2013.

[36] C. Brun and V. Nikoulina, “Aspect based sentiment analysis into the wild”, In: *Proc. of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pp. 116-122, 2018.

[37] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. A. Smadi, M. A. Ayyoub, Y. Zhao, B. Qin, and O. D. Clercq, “Semeval-2016 task 5: Aspect based sentiment analysis”, In: *Proc. of International Workshop on Semantic Evaluation*, pp. 19-30, 2016.