

A sentiment analysis method of objects by integrating sentiments from tweets

Huyen Trang Phan^a, Ngoc Thanh Nguyen^{b,d}, Van Cuong Tran^c and Dosam Hwang^{a,*}

^a*Department of Computer Engineering, Yeungnam University, Gyeongsan, Republic of Korea*

^b*Faculty of Computer Science and Management, Wroclaw University of Science and Technology, Wroclaw, Poland*

^c*Faculty of Engineering and Information Technology, Quang Binh University, Dong Hoi, Vietnam*

^d*Faculty of Information Technology, Nguyen Tat Thanh University, Ho Chi Minh city, Vietnam*

Abstract. Sentiment analysis has been gaining importance in many applications such as recommendation systems, the decision making support and prediction models. Sentiment analysis helps to understand and evaluate public opinion regarding social events, product services, and political trends, especially the feelings expressed through comments by users in social networks such as Twitter, Facebook, and Instagram. There have been a lot of research attempts to address the tweets sentiment analysis problem with high accuracy, particularly in case of tweets that express a single sentiment towards a single object. However, the results of the classification are not highly accurate in cases such as the following: a user expresses multiple sentiments towards a single object in a tweet; a user presents multiple sentiments towards multiple objects; and a user indicates a single sentiment towards multiple objects. Furthermore, the previous studies only analyze the sentiment of each tweet without considering the objects and the sentiment towards each object from an entire set of tweets. This study attempts to deal with the limitations of the previous methods; an approach is proposed herein, based on integrating the sentiment towards a particular object from all tweets related to that object. The proposed method focuses on determining the objects and indicating the sentiment towards the specific objects by combining the sentiments related to each object from the entire set of tweets. On experimental evaluation, the proposed method is observed to have achieved a fairly good result in terms of the error ratio and achieved information.

Keywords: Sentiment-analysis, sentiment-integration, object-determination

1. Introduction

In recent years, social network services have become widespread and necessary communication platforms in everyday life. Twitter is among the most popular social networks worldwide. As of the third quarter of 2018, Twitter averaged at 326 million

monthly active users¹. Twitter is very convenient to use; and celebrities, corporations, and organizations create and manage their Twitter accounts to interact with their fans and the public. Moreover, Twitter is probably the primary communication channel used by many people. Twitter users can freely publish tweets to express their feelings, find their friends, follow famous people they are interested in, and access information at any given period of time. Hence, the number of tweets is increasing rapidly, and their rate of generation is remarkably high. Therefore, Twitter has become a source of meaningful information to anyone who knows how to use it. The sentiments

*Corresponding author. Dosam Hwang, Department of Computer Engineering, Yeungnam University, Gyeongsan, Republic of Korea. E-mail: dosamhwang@gmail.com.

¹<https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>

of users expressed through tweets can be used as reviews and survey responses of user opinions and social media, for applications ranging from marketing to customer service. Therefore, sentiment analysis is a potential topic of interest to many researchers, as it can be used to exploit the useful information on social networks [10].

Sentiment analysis is defined as the process of extracting subjective information from a document, sentence, or term [19]. Not only does it include extracting the polarity [24], stance [18], target [8], and the holder of opinion [13] but also it involves in analyzing the implicatures or effects to indicate aspects of the goal that people like or dislike [11].

The methods used for sentiment analysis can be divided into three levels namely the document level, sentence level, and the entity and aspect level [14]. The document level provides the polarity to classify a document as either positive, negative, or neutral. The sentence level provides the polarity for classifying a sentence into either of the classes objective or subjective; and the sentiment of the subjective sentences is in turn classified into two classes: positive or negative. The entity and aspect level explain what people like or dislike. In this study, the second and the third levels of sentiment analysis are examined.

There have been several published methods to solve the sentiment analysis problem. The methods used in previous studies for analyzing sentiment can be divided into three types [23]. The first approach relies on the famous machine learning algorithms. However, this approach has the following limitations: a large number of labeled training tweets are needed for the supervised methods, while the unsupervised techniques have to seek unlabeled training tweets [16]. The second approach is based on the lexicon-based methods [1, 6]. This approach depends on finding the sentiment lexicon. The limitation of the lexicon-based methods is that it is very time consuming, and cannot be used alone; it must be combined with other automated methods as a final check to avoid mistakes. The third approach is the hybrid methods by combining both above approaches [2, 12]. This method is widespread and play a pivotal role in the majority of sentiment analysis methods.

In general, the previous methods could achieve accurate results only in the case of simple tweets that reflect a single sentiment for a single object; for instance, the tweet “*I have the worst anxiety because of the weather*” presents a *negative* sentiment for a single object, *weather*. However, they often generate

inaccurate classification results in cases such as the following:

- i) Tweets expressing two different sentiments for two different objects: For example, a tweet such as “*I do not like rain, but I like the sunshine*” would be classified as *neutral* by mentioned methods for sentiment analysis. However, this result is inaccurate because the user expresses a *negative* sentiment for the object “*rain*” and a *positive* sentiment for the object “*sunshine*.”
- ii) Tweets that present multiple sentiments for a single object: For example, a tweet such as “*Rain is nothing but a thorn of roses*” would be classified as *neutral* by mentioned methods for sentiment analysis although the actual expression is *negative*.
- iii) Tweets that express a single sentiment for two different objects: For example, a tweet such as “*I like rain, and I like the sunshine*” would be classified as *positive* by mentioned methods for sentiment analysis. Although this is correct, it does not extract the user’s opinion fully.

Additionally, most of the previous studies implemented the following two steps: The first step was to determine whether a tweet expresses an opinion or not, and the second step classified the opinion tweets as positive and negative classes. The result of this process provides only the polarity scores of the tweets, while the intended objects remain unknown. For example, regarding the iPhone, one user tweeted: “*The color is good,*” while another user tweeted: “*The sound is not good.*” A sentiment analysis using the previous methods would classify the first tweet as positive and the second tweet as *negative*. Therefore, the manufacturer would not know which part of the iPhone received a *positive* response and which part received a *negative* one.

Therefore, a mechanism is required to address the above problems; and this motivates us to propose a method that would perform a sentiment analysis by integrating the sentiments from tweets with respect to a particular object. The proposed method involves the following steps: First, a large number of tweets that belong to a specific topic from several users are collected. Second, the objects and the sentiments with respect to each object are determined by using the Bidirectional Long Short-Term Memory (BiLSTM) and Conditional Random Field (CRF) algorithms. Third, the sentiment of each tweet is determined by combining the objects and sentiments related to the

objects in each tweet. Finally, the sentiment for each object from all tweets is integrated by computing the sentiment score of that object based on the frequency of the positive and negative components in all tweets.

The difference between the proposed method and the previous methods is that not only does this method classify the sentiment of each tweet such as negative, positive, or neutral but it also provides the objects related to the sentiments. In particular, this method integrates the overall sentiment for each object from tweets. The presented differences are also the main contributions of this study.

The rest of the paper is organized as follows: In Section 2, we briefly summarize the previous work on sentiment analysis approaches. The research problem of this study is described in Section 3. Section 4 presents the proposed method and the experimental results are presented in Section 5. The conclusion and future works are presented in Section 6.

2. Related works

There have been many published studies regarding the sentiment analysis of tweets, which can be divided into the following three methods: lexicon-based, machine learning-based, and a hybrid method. The machine learning-based method focuses on extracting various features related to lexical, syntactic, semantic, and sentiment polarity of tweets such as context information, Part-Of-Speech (POS) tags, n-grams, sentiment words, and word embeddings. A classifier is then built based on these features by using one of the machine learning algorithms, for example, the Naive Bayes [7], Support Vector Machine [3]. The lexicon-based method indicates the overall sentiment tendency of a given text by employing lexicons of words weighted with their sentiment, such as SentiWordNet (SWN) [4]. The hybrid approach combines the speed of the lexical approach and the accuracy of machine learning-based approach; and moreover, deep learning methods are known as the state-of-art methods in the area of sentiment analysis [20, 27]. Thus, the existing research has focused on classifying tweets into three classes namely positive, negative, and neutral. In this section, we analyze some specific cases to understand the motivation for our proposal.

Atanu Dey et al. [6] presented a methodology to create a lexicon known as the Senti-N-Gram. In this approach, the n-gram sentiment scores were extracted

by using the rules. These scores were then compared with the analysis made by human annotators and found to be statistically equivalent. Finally, a sentiment classification methodology was proposed by counting the number of positive and negative sentences. The paper also proposed a ratio-based sentence-level sentiment classification approach by counting the number of positive and negative sentences. This approach has the following advantages: i) It is fully automatic and domain independent. ii) Human annotators are eliminated, thereby reducing human bias as well as the cost. iii) The computation time is further reduced in case of n-gram based sentiment analysis as the score from the stored lexicon can be reused. However, the method also has certain limitations: i) It cannot process all types of sentences; for example, the compound, conditional, and irony sentences. ii) The object related to the sentiment remains unknown. iii) The sentiment of all tweets related to each object is not extracted.

Asghar et al. [1] proposed a lexicon-based method to classify user sentiments from online communities. They combined different lexicons (e.g., SentiWordNet (SWN), user-defined dictionaries) and different classifiers (e.g., an SWN-based classifier, negation classifier, domain-specific classifier). The advantage of this approach is that not only the relationships among words but also domain-specific words are extracted to resolve the issue of domain dependency. However, this approach has the following drawbacks: i) Compound sentences and irony sentences are not considered. ii) The object of the sentiment is not identified. iii) The sentiment of all tweets related to each object is not extracted.

In study [2], the authors focused on presenting a unified framework for classifying tweets using a hybrid classification scheme. This method aims to improve the performance of Twitter-based sentiment analysis systems by combining four classifiers namely a slang classifier, an emoticon classifier, the SWN classifier, and an improved domain-specific classifier. This approach has the following advantages: i) Classification results are better than the baseline methods. ii) The framework is generalized and can classify tweets in any domain. However, the method has as the following limitations: i) The results are inaccurate with tweets containing compound sentences. ii) The method only focuses on analyzing the sentiment of each tweet. iii) The object of the sentiment is not extracted.

In the paper [27], the authors introduced a word embeddings method using unsupervised learning

based on contextual latent semantic relationships and co-occurrence characteristics of words in tweets. These word embeddings were combined with n-grams features and the sentiment scores for words to create a set of sentiment features of tweets. This features set was integrated into a deep convolutional neural network for training and predicting sentiment labels. The authors compared the performance of their model to the baseline model on five Twitter data sets, and the results indicated that their model performed better for tweet sentiment analysis. However, there were some limitations: they could only classify tweets into two sentiment groups, i.e., positive and negative; they could not decide the object of the sentiment; and could not integrate sentiment for each object from all tweets.

In general, the previous studies have been able to classify tweets in any domain with high accuracy. However, the methods produced inaccurate results for tweets that expressed multiple sentiments; for example, a tweet such as “wind, wind, wind, I hate wind, I only like snow” would be labeled as *neutral* in their systems, while actually the result is *negative* for *wind* and *positive* for *snow*. Moreover, the accuracy of the classification results decreased in case of compound tweets; for example, a tweet such as “Rain is a kind of devil” is labeled *neutral* in their system although it has to be labeled as *negative*. In addition, the object of the sentiment is not considered; for example, in the cases of tweets such as “I like the most wind and hate the most sunshine,” which contains two objects, “wind” and “sunshine,” corresponding to two sentiments, *positive* and *negative*. Moreover, the previous methods only indicated the sentiment of each tweet without presenting the general sentiment of all tweets towards each object.

Through the above analysis of the advantages and limitations of several previous methods, in this study, we focus on addressing, in a complex way, some of the issues including the extraction of the object of the sentiment, analyzing the sentiment of each tweet, and integrating the sentiment of all tweets towards each object by combining the BiLSTM and CRF algorithms as well as using the rating technique based on sentiment scores.

3. Research problem formalization

This section presents the basic concepts and definitions related to tweets such as objects, the sentiment of each tweet, and the sentiment related to each object

in the entire set of tweets. The research problems are stated at the end of this section.

3.1. Definitions

We assume a finite set of tweets, T , that represents the opinions of users about a specific topic, and is represented as follows:

$$T = \{t_1, t_2, \dots, t_n\},$$

where n is the number of gathered tweets.

Let Pt be a set of positive words and Nt be a set of negative words. To determine necessary elements in tweets, each tweet has to be separated into a set of tokens. Let $t_i = \{w_1, w_2, \dots, w_g\}$ be a set of tokens of tweet t_i from T .

Definition 1. A sentiment relation between token w_k and w_h ($w_h \in Pt \cup Nt$) is defined by function Θ , as represented by the following expression:

$$\Theta(w_h, w_k) = \begin{cases} 1, & \text{if } w_k \text{ referring to } w_h \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

A tweet can have many objects. Let $O_i = \{o_1, o_2, \dots, o_m\}$ be a set of objects of t_i in which o_j ($j = 1, 2, \dots, m$) be an object assigned to the sentiment related to the chosen topic.

Definition 2. An object o_j of sentiments in a tweet is a token w_k that satisfies the two conditions simultaneously such as, w_k has to be a noun or noun phrase, and related to at least one sentiment word existing in that tweet. o_j is defined as follows:

$$o_j = \{w_k | \text{tag}(w_k) = 'NOUN', \\ \exists w_h \in t_i : \Theta(w_k, w_h) = 1\}.$$

Assume that $S_T = \{s_{t_1}, s_{t_2}, \dots, s_{t_n}\}$ be a set of sentiments from T where s_{t_i} be a sentiment of tweet t_i .

Definition 3. The sentiment of the tweet is represented by a set of sentiments associated with objects. The sentiment of the object is denoted by an object variable respective if the sentiment is positive and by a negation symbol (\neg) stand is placed before the variable representing the object if the sentiment is negative. s_{t_i} is defined as follows:

$$s_{t_i} = \{e_{o_j}, j = 1, \dots, m\} \quad (2)$$

where e_{o_j} is the sentiment of the object o_j and is determined as

$$e_{o_j} = \begin{cases} \neg o_j, & \text{if } ((\exists w_k \in Nt) \wedge (\Theta(o_j, w_k) = 1)) \\ o_j, & \text{if } ((\exists w_k \in Pt) \wedge (\Theta(o_j, w_k) = 1)) \end{cases} \quad (3)$$

Assume that $S_O = \{s_{o_1}, s_{o_2}, \dots, s_{o_m}\}$ be a set of sentiments according to objects in entire tweets, and elements in S_O can be repetitive. Let o_j^+ and o_j^- be the positive and negative sentiment components of the object o_j , respectively [15].

Definition 4. A sentiment of the object o_j in T , denoted by s_{o_j} , is determined by integrating sentiments from S_T in which those sentiments associated with object o_j . Let se_{o_j} be a sentiment degree of object o_j , s_{o_j} is represented by a double $\langle o_j, se_{o_j} \rangle$, and se_{o_j} is assessed as follows:

$$se_{o_j} = \begin{cases} \text{very negative,} & \text{if } score(o_j) < -0.5 \\ \text{negative,} & \text{if } -0.5 \leq score(o_j) < -0.1 \\ \text{neutral,} & \text{if } -0.1 \leq score(o_j) \leq 0.1 \\ \text{positive,} & \text{if } 0.1 < score(o_j) \leq 0.5 \\ \text{very positive,} & \text{if } score(o_j) > 0.5 \end{cases} \quad (4)$$

where

$$score(o_j^+) = \frac{frequency(o_j^+)}{frequency(o_j^+) + frequency(o_j^-)} \quad (5)$$

$$score(o_j^-) = \frac{frequency(o_j^-)}{frequency(o_j^+) + frequency(o_j^-)} \quad (6)$$

$$score(o_j) = score(o_j^+) - score(o_j^-).$$

3.2. Problems

In this study, we focus on finding a method to answer the main question as follows: *How can determine the sentiment of objects by integrating sentiments of those objects from tweets?* This question is partitioned in the set of three following sub-questions:

1. For given tweets (T), how can the determination objects and their sentiments in each tweet?
2. How to express the sentiment of each tweet based on the sentiment of objects in that tweet?
3. How can a sentiment integration process be performed to determine the sentiment of objects from entire tweets?

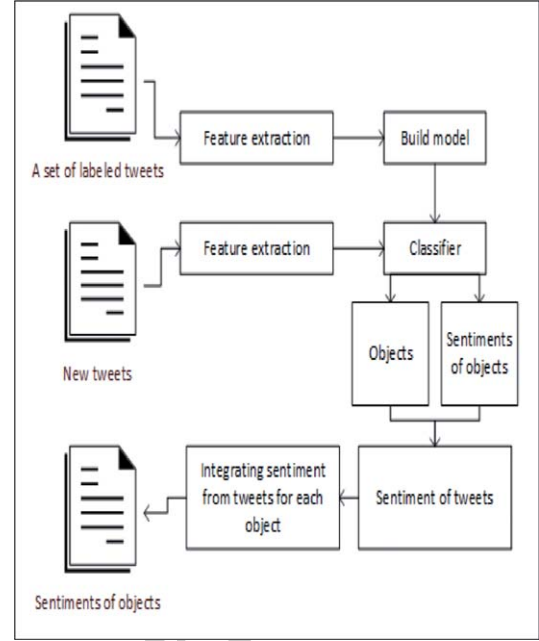


Fig. 1. Workflow of sentiment analysis of objects by integrating sentiment from tweets.

4. Methods

This section presents the method to solve problems identified in Section 3.2. The proposed method consist of three main steps: Firstly, determining objects referring to sentiments; Secondly, indicating sentiment of each tweet; Finally, analyzing sentiment according to each object by integrating sentiment from tweets. The workflow of method is shown in Fig.1. The following subsections explain details of the proposed method.

4.1. Feature extraction

To determine objects related to sentiments, the features are extracted from in each tweet. In this work, the information referring to the lexical, syntactic, semantic, and polarity sentiment of words are employed as features.

n-grams: n-grams includes 1-gram, 2-grams, and 3-grams. Each type of n-grams is created based on the term frequency-inverse document frequency (TF-IDF) values. Each n-grams appearing in a tweet becomes an entry in the feature vector with the corresponding feature value of TF-IDF.

POS tags of words: POS tags of words are used as features for indicating words that are adjectives, verb, adverbs, nouns and noun phrases. These

words support to determine sentiments and objects of sentiments existing in each tweet. The NLTK toolkit [5] is used to annotate the POS tags and implement tokenization for all the tweets. POS tags with their corresponding TF-IDF values are the syntactic features and feature values, respectively.

Word embeddings: A significant challenge when handling with tweets is that the lexicons and syntactics used in a tweet are informal and much different from normal sentences. Hence, it is challenging to correctly capture the properties related to the lexical and syntactic features. This problem is dealt by applying the word embeddings approach to compute tweet embedding representations.

In this study, the 300-dimensional pre-trained word embeddings from Glove² is used to compute a tweet embedding as the average of the embeddings of words in the tweet.

The Glove model is applied because it is an extension to the Word2vec model. This approach combines both the global statistics of matrix factorization techniques namely latent semantic analysis and social context-based learning in Word2vec [25]. Moreover, rather than using a window to define the local context, the glove model constructs an explicit word-context or a co-occurrence word matrix using statistics across the entire text corpus. Therefore, in general, the model may result in better word embeddings.

Special words: Special words include negation, intensifier, and diminisher words. Special words can reverse the polarity of sentiment words; for example, if a negation word appears before a positive word then the sentiment of the tweet is changed to negative and vice versa. These features are extracted using a window of 1 to 3 words before a sentiment word and search for these kinds of words.

Sentiment words: Sentiment words have a significant role in determining the sentiment polarity of tweets. Sentiment words consist of positive and negative words. A number of sentiment words in each tweet are used as a feature. The sentiment dictionaries provided by Hu and Liu [8] are employed for determining the positive and negative words in a tweet.

4.2. Determining objects of sentiments in each tweet

A combination of BiLSTM and CRF models [9] is used to determine objects and their sentiments in each tweet. The combination leverages the advantages of

both models: the creative ability to extract the features of the LSTM model and the steady predictability of the CRF model [22].

The above model operates according to the following steps: Word embedding of each word is entered into the BiLSTM layer to extract the features presented in section 4.1. Then the CRF layer treats the above features to predict about label for each word. In addition to the information received from the BiLSTM layer, the CRF layer also relies on information from previously anticipated labels; for example, if the previous word was an object, it is highly probable that the present word is a sentiment word.

The parameters of the model include the word-embedding matrix, the weight matrix of the BiLSTM layer, and the transposition matrix of the CRF layer. All these parameters are updated during training on the labeled tweets via the back-propagation algorithm with Stochastic Gradient Descent (SGD). Moreover, during training, the dropout technique is applied to avoid over-fitting.

In addition, to increase the accuracy of identifying sentiments and the objects of sentiments in each tweet, we produce a set of rules and based on these rules to label the training data as follows:

Rule 1 (r_1): If the tweet contains sentiment words complementary for a noun or noun phrase (denoted by n) in a tweet, then n is marked as a word indicating the object; for example, “*I have a nice bag.*” or “*She has nice eyes.*”, then “*eyes*” and “*bag*” are identified as objects of sentiments.

Rule 2 (r_2): If the tweet contains sentiment words and n stands after preposition, then n is a word determining the object in tweet. e.g., “*iPhone is good at sound*”, then “*sound*” is one object in this tweet.

Rule 3 (r_3): If the tweet contains sentiment words and n stands after normal verb containing sentiment, then n is a word determining the object in tweet. e.g., “*I like sunshine*”, then “*sunshine*” is determined as one object.

Rule 4 (r_4): If the tweet contains sentiment words and n stands before “to be” verb, then n is determined as a word indicating the object. e.g., “*The flower is nice.*” Follow this rule, we have one object as “*flower.*”

The order of priority for the rules is r_1, r_2, r_3, r_4 ; this means that r_1 takes precedence over r_2 , and so on; for example, when in a tweet, n satisfies both r_2 and r_4 then r_2 is preferred.

The following example provides a detailed analysis to clearly understand the steps to determine the objects of sentiments in tweets.

²<http://nlp.stanford.edu/projects/glove/>

Example 1: Given a set of tweets, $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$,

- t_1 : “I do not like rain, but I like the sunshine.”
 t_2 : “Rain is nothing but a thorn of roses.”
 t_3 : “I like rain, and I like the sunshine.”
 t_4 : “Sunshine is lovely, but the wind is uncomfortable.”
 t_5 : “The rain is sad, and the sunshine is also sad.”
 t_6 : “Sunshine is good for me.”

a) For t_1 :

$\text{tag}(\text{rain}) = \text{'NOUN'}$, $\text{not like} \in Nt$,
 and $\Theta(\text{not like}, \text{rain}) = 1$. Thus $o_1 = \text{rain}$.
 $\text{tag}(\text{sunshine}) = \text{'NOUN'}$, $\text{like} \in Pt$,
 and $\Theta(\text{like}, \text{sunshine}) = 1$. Thus $o_2 = \text{sunshine}$.
 From o_1 and o_2 as a result $O_1 = \{\text{rain}, \text{sunshine}\}$.

b) For t_2 :

$\text{tag}(\text{rain}) = \text{'NOUN'}$, $\text{nothing} \in Nt$,
 and $\Theta(\text{nothing}, \text{rain}) = 1$. Thus $o_1 = \text{rain}$.
 From o_1 as a result $O_2 = \{\text{rain}\}$.

c) For t_3 :

$\text{tag}(\text{rain}) = \text{'NOUN'}$, $\text{like} \in Pt$,
 and $\Theta(\text{like}, \text{rain}) = 1$. Thus $o_1 = \text{rain}$.
 $\text{tag}(\text{sunshine}) = \text{'NOUN'}$, $\text{like} \in Pt$,
 and $\Theta(\text{like}, \text{sunshine}) = 1$. Thus $o_2 = \text{sunshine}$.
 From o_1 and o_2 as a result $O_3 = \{\text{rain}, \text{sunshine}\}$.

d) For t_4 :

$\text{tag}(\text{sunshine}) = \text{'NOUN'}$, $\text{lovely} \in Pt$,
 and $\Theta(\text{lovely}, \text{sunshine}) = 1$. Thus $o_1 = \text{sunshine}$.
 $\text{tag}(\text{wind}) = \text{'NOUN'}$, $\text{uncomfortable} \in Nt$,
 and $\Theta(\text{uncomfortable}, \text{wind}) = 1$. Thus $o_2 = \text{wind}$.
 From o_1 and o_2 as a result $O_4 = \{\text{sunshine}, \text{wind}\}$.

e) For t_5 :

$\text{tag}(\text{rain}) = \text{'NOUN'}$, $\text{sad} \in Nt$,
 and $\Theta(\text{sad}, \text{rain}) = 1$. Thus $o_1 = \text{rain}$.
 $\text{tag}(\text{sunshine}) = \text{'NOUN'}$, $\text{sad} \in Nt$,
 and $\Theta(\text{sad}, \text{sunshine}) = 1$. Thus $o_2 = \text{sunshine}$.
 From o_1 and o_2 as a result $O_5 = \{\text{rain}, \text{sunshine}\}$.

f) For t_6 :

$\text{tag}(\text{sunshine}) = \text{'NOUN'}$, $\text{good} \in Pt$,
 and $\Theta(\text{good}, \text{sunshine}) = 1$. Thus $o_1 = \text{sunshine}$.
 From o_1 as a result $O_6 = \{\text{sunshine}\}$.
 From O_1, O_2, O_3, O_4, O_5 , and O_6 as a result $O = \{\text{rain}, \text{sunshine}, \text{wind}\}$

4.3. Determining sentiment of each tweet

The sentiment of each tweet is determined by implementing the two following steps: First, sentiments of objects in each tweet are determined by using the combination of BiLSTM and CRF models

Table 1
Description of r_5

Rule	Example
Negation + Negative \rightarrow Positive	“not bad” \rightarrow Positive
Negation + Positive \rightarrow Negative	“not good” \rightarrow Negative
Negation + Neutral \rightarrow Positive	“no problem” \rightarrow Positive

Table 2
Description of r_6

Rule	Example
Intensifier + Negative \rightarrow Negative	“so bad” \rightarrow Negative
Intensifier + Positive \rightarrow Positive	“too good” \rightarrow Positive
Intensifier \rightarrow Positive	“absolutely” \rightarrow Positive

Table 3
Description of r_7

Rule	Example
Diminisher + Negative \rightarrow Negative	“quite bad” \rightarrow Negative
Diminisher + Positive \rightarrow Positive	“slight good” \rightarrow Positive
Diminisher + Neutral \rightarrow Positive	“slight problem” \rightarrow Positive

Table 4
Description of r_8

Rule	Example
Negative + Negative \rightarrow Negative	“sad, angry” \rightarrow Negative
Positive + Positive \rightarrow Positive	“nice, good” \rightarrow Positive

as explained in Section 4.2. Second, objects and sentiments related to these objects in each tweet are combined to create the sentiment of tweet. In addition, to support the determination sentiments of objects, the following set of rules are devised to label the training data:

Rule 5 (r_5): Rule related to negation words. This rule is used to identify the sentiments of objects when negation words appear before sentiment words in the tweet (see Table 1).

Rule 6 (r_6): Rule related to intensifier words. This rule is applied to decide the sentiments of objects when the intensifier words appear before sentiment words in the tweet (see Table 2).

Rule 7 (r_7): Rule related to diminisher words. This rule is employed to determine the sentiments of objects when the diminisher words precede the sentiment words in the tweet (see Table 3).

Rule 8 (r_8): Rule related to sentiment words. This rule is used to identify the sentiments of objects when consecutive sentiment words appear in the tweet (see Table 4).

The order of priority of the rules r_5, r_6, r_7 and r_8 is from left to right, meaning that in the tweet, if the rule to the left is satisfied first then that rule takes precedence. After objects and sentiments of objects

are extracted, the sentiment of each tweet is analyzed by applying Definition 3 in Section 3.1. For greater clarity, the following example is executed and analyzed based on the objects extracted in Ex.1.

Example 2: From the result of Ex.1, the sentiment of each tweet is identified as follows:

a) For t_1 :

$o_1 = \text{rain, not like} \in Nt, \Theta(\text{not like, rain}) = 1.$

Hence, $e_{o_1} = \neg \text{rain}.$

$o_2 = \text{sunshine, like} \in Pt, \Theta(\text{like, sunshine}) = 1.$

Hence, $e_{o_2} = \text{sunshine}.$

From e_{o_1}, e_{o_2} as a result $s_{t_1} = \{\neg \text{rain, sunshine}\}.$

b) For t_2 :

$o_1 = \text{rain, nothing} \in Nt, \Theta(\text{nothing, rain}) = 1.$

Hence, $e_{o_1} = \neg \text{rain}.$

From e_{o_1} as a result $s_{t_2} = \{\neg \text{rain}\}.$

c) For t_3 :

$o_1 = \text{rain, like} \in Pt, \Theta(\text{like, rain}) = 1.$

Hence, $e_{o_1} = \{\text{rain}\}.$

$o_2 = \text{sunshine, like} \in Pt, \Theta(\text{like, sunshine}) = 1.$

Hence, $e_{o_2} = \text{sunshine}.$

From e_{o_1}, e_{o_2} as a result $s_{t_3} = \{\text{rain, sunshine}\}.$

d) For t_4 :

$o_1 = \text{sunshine, lovely} \in Pt,$

$\Theta(\text{lovely, sunshine}) = 1.$

Hence, $e_{o_1} = \text{sunshine}.$

$o_2 = \text{wind, uncomfortable} \in Nt,$

$\Theta(\text{uncomfortable, wind}) = 1.$

Hence, $e_{o_2} = \neg \text{wind}.$

From e_{o_1}, e_{o_2} as a result $s_{t_4} = \{\text{sunshine, } \neg \text{wind}\}.$

e) For t_5 :

$o_1 = \text{rain, sad} \in Nt, \Theta(\text{sad, rain}) = 1.$

Hence, $e_{o_1} = \neg \text{rain}.$

$o_2 = \text{sunshine, sad} \in Nt, \Theta(\text{sad, sunshine}) = 1.$

Hence, $e_{o_2} = \neg \text{sunshine}.$

From e_{o_1}, e_{o_2} as a result $s_{t_5} = \{\neg \text{rain, } \neg \text{sunshine}\}.$

f) For t_6 :

$o_1 = \text{sunshine, good} \in Pt, \Theta(\text{good, sunshine}) =$

1.

Hence, $e_{o_1} = \text{sunshine}.$

From e_{o_1} as a result $s_{t_6} = \{\text{sunshine}\}.$

Therefore, $S_T = \{\{\neg \text{rain, sunshine}\}, \{\neg \text{rain}\}, \{\text{rain, sunshine}\}, \{\text{sunshine, } \neg \text{wind}\}, \{\neg \text{rain, } \neg \text{sunshine}\}, \{\text{sunshine}\}\}.$

The results of Ex.2 show that, unlike previous methods, this method yields more detailed results. For example, if $s_{t_2} = \{\neg \text{rain}\}$, it means that t_2 has a negative sentiment, which is assigned to object *rain*. In others words, the object of the sentiment is also identified in addition to the sentiment of the tweet.

4.4. Integrating sentiments according to each object from tweets

This section focus on determining the overall sentiment of each objects by integrating sentiments from all tweets. For each object, we compute the sentiment score of positive and negative components by applying Equations 5 and 6 based on the frequency of these components in all tweets. Then the sentiment score of objects is calculated by using Equation 7. Furthermore, to analyze the sentiment of each object in more detail, we give a scale to divide the sentiment of each object into different levels based on the sentiment score of that object. In this work, the thresholds for the sentiment score are chosen in an interval $[-1, +1]$ based on empirical and some previous method [21, 26]. The steps to integrate sentiments according to each object from tweets are explained below in Ex.3.

Example 3: Based on a set of objects and a set of sentiments of objects extracted in Ex.1 and Ex.2, the sentiment of each object is calculated as follows:

a) For $o_1 = \text{rain}$:

$\text{frequency}(\text{rain}^+) = 1, \text{frequency}(\text{rain}^-) = 3,$

$\text{score}(\text{rain}^+) = 0.25, \text{score}(\text{rain}^-) = 0.75,$

$\text{score}(\text{rain}) = 0.25 - 0.75 = -0.5.$

Thus $se_{\text{rain}} = \text{negative}.$

Therefore, $s_{\text{rain}} = \langle \text{rain, negative} \rangle.$

b) For $o_2 = \text{sunshine}$:

$\text{frequency}(\text{sunshine}^+) = 4,$

$\text{frequency}(\text{sunshine}^-) = 1,$

$\text{score}(\text{sunshine}^+) = 0.8,$

$\text{score}(\text{sunshine}^-) = 0.2,$

$\text{score}(\text{sunshine}) = 0.8 - 0.2 = 0.6,$

Thus $se_{\text{sunshine}} = \text{very positive}.$

Therefore $s_{\text{sunshine}} = \langle \text{sunshine, very positive} \rangle.$

c) For $o_3 = \text{wind}$:

$\text{frequency}(\text{wind}^+) = 0, \text{frequency}(\text{wind}^-) =$

1,

$\text{score}(\text{wind}^+) = 0, \text{score}(\text{wind}^-) = 1,$

$\text{score}(\text{wind}) = 0 - 1 = -1,$

Thus $se_{\text{wind}} = \text{very negative}.$

Therefore $s_{\text{wind}} = \langle \text{wind, very negative} \rangle.$

Hence, $S_O = \{\langle \text{rain, negative} \rangle, \langle \text{sunshine, very positive} \rangle, \langle \text{wind, very negative} \rangle\}.$

5. Experiment

Focusing on three research problems presented in Section 3.2, we proposed a method and verified it through experimentation on a real dataset. This section presents and analyzes the experimental results of the proposed method.

5.1. Data acquisition

In this study, the Python package Tweepy³ is used to collect 4758 tweets related to *Weather* topic in English. The data is pre-processed to remove unnecessary elements such as punctuation marks, re-tweet symbols, URLs, hashtags, and query term. In addition, a describing text is replaced for an emoji icon in a tweet by using the Python emoji package⁴. Since tweets are usually informal in which users can use acronyms as well as wrong spellings, the accuracy of the results can be affected. Therefore, the Python-based Aspell library⁵ is employed to correct the spelling of tweets. We scan all words in tweets and search for the words belonging to the two available sentiment dictionaries. This research does not consider tweets that do not contain any word belongs to these dictionaries. Since each tweet represents a person's opinion on a topic, it is difficult to exist a tweet that contains two opposing sentiments directed at the same object. Therefore, we do not consider tweets that indicate two opposing sentiments for the same object. The number of tweets after pre-processing is 2057 tweets. These tweets are then divided and stored into two separate database files for training and testing. The training data consists of 1440 tweets, and the testing data includes 617 tweets. The tweets are assigned labels. The data was annotated with three labels (i.e., *Positive*, *Negative*, and *Object*) by five manual annotators. The training data file contains two columns separated by a tab space. Each token has placed in a separate line such that the first item on each line is a token and the second item is a label. The tokens that do not belong to factors which need to interest were annotated as *Other*. We annotated the test set as the gold standard to assess the performance. There are 849 tokens indicating the object, 557 tokens determining positive sentiment, 292 tokens indicating negative sentiment which need to be predicted in the test set.

5.2. Evaluation

The effectiveness of the proposed method is evaluated by using different evaluation metrics namely Precision (P), Recall (R), F-measure (F_1), and Accuracy as follows:

$$P = \frac{TP}{TP + FP} \quad (8)$$

$$R = \frac{TP}{TP + FN} \quad (9)$$

$$F_1 = 2 \times \frac{P \times R}{P + R} \quad (10)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (11)$$

where *True Positive* (TP) is the number of exactly assigned labels (as *positive*, *negative*, *object*); *False Positive* (FP) is the number of misassigned labels; *True Negative* (TN) is the number of exactly assigned *Other* labels; *False Negative* (FN) is the number of misassigned *Other* labels.

In addition, the mean absolute error MAE_m [17] is applied to estimate error ratio of objects polarity method. MAE_m is calculated as follows:

$$MAE_m = \frac{1}{2m} \sum_{i=1}^m |v_i - v_i^*| \quad (12)$$

where m is the number of actual objects; v_i is the predicted average polarity of o_j ; v_i is based on the classes assigned to the occurrences of o_j by the proposed method. v_i^* is the actual average polarity of o_j ; v_i^* is based on the classes assigned to the occurrences of o_j by the human annotators.

5.3. Result and discussion

a. Determining objects

In the testing set, there are twelve objects that are described by at least one sentiment word including: *rain*, *wind*, *sunshine*, *snow*, *fog*, *ice*, *cloud*, *hail*, *rainbow*, *frost*, *temperature*, *storm*. Because of the limitation of space, the objects are denoted as o_j ($j = 1, \dots, 12$). The result of the object detection process is shown in Table 5.

Using the confusion matrix in Table 5 and Equations 8, 9, and 10, the performance of the object detection in the tweets was calculated.

Table 6 shows that the performance of the object detection is fairly good, in which, the average P is 72.82%; The average R is 80.05%, and the average F_1 score is 76.03%. According to our assessment, one of the main reasons to achieve this performance is that the combination of BiLSTM and CRF models has promoted the advantages of both models in

³<https://pypi.org/project/tweepy/>

⁴<https://pypi.org/project/emoji/>

⁵<https://pypi.org/project/aspell-python-py2/>

Table 5
Confusion Matrix of object detection

		Predicted classes											
		o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}	o_{11}	o_{12}
Actual classes	o_1	70			1		2		2	10	11		
	o_2		72				1		1				13
	o_3			71						10			
	o_4		1		57		7		2		6	1	
	o_5	6		1		30		4		1			
	o_6	2					28				6		
	o_7	2						70		9			
	o_8	3		4					22				
	o_9	11				10				88			
	o_{10}				2		3				18		
	o_{11}			10	1		3					80	1
	o_{12}	7	5			1		1		2			80

Table 6
Performance of detecting objects

	o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}	o_{11}	o_{12}
TP	70	72	71	57	30	28	70	22	88	18	80	80
TN	722	756	753	771	796	797	763	815	708	803	753	739
FP	31	6	15	4	11	16	5	5	32	23	1	14
FN	26	15	10	17	12	8	11	7	21	5	15	16
P	69.3	72.3	82.6	73.4	73.2	63.6	73.3	81.5	73.3	43.9	78.8	85.1
R	72.9	82.8	87.7	77.0	71.4	77.8	86.4	75.9	80.7	78.3	84.2	83.3
F_1	71.1	77.2	85.1	75.2	72.3	70.0	79.3	78.6	76.8	56.3	81.4	84.2

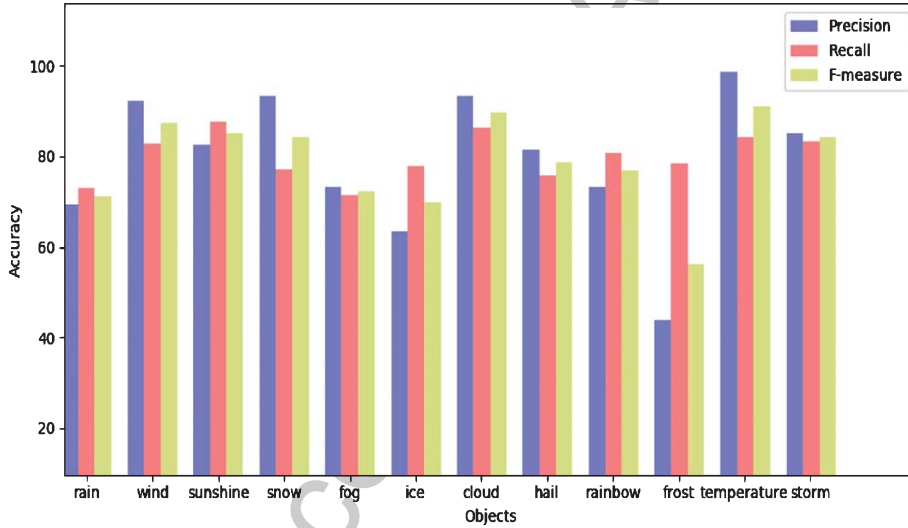


Fig. 2. Compare performance in specific objects.

detecting objects and their sentiments. In addition, the rules provided help the training model to determine, more accurately, the location of words that indicate objects and words that define sentiments of objects.

Fig.2 shows that determining of the specific objects generally gives well result on classifying objects

as o_2 , o_3 , o_8 , o_{11} , o_{12} . However, the performance of classification object o_{10} does not obtain as desirable. Concretely, the F_1 score is only 56.3%. Intuitively, one of the main reasons for the low performance is that the training data has less words indicating object o_{10} . We believe that the construction of a large data warehouse and the balance between words indicating

Table 7
Result of sentiment of each tweet

	1s-1o	1s-2o	1s-3o	2s-2o	2s-3o
# t_i	393	87	11	86	40
# t_i contains o_j	219	35	4	0	0
# t_i contains $\neg o_j$	174	52	7	0	0
# t_i contains both o_j and $\neg o_k$	0	0	0	86	40
Accuracy	81.3	77.1	75.4	80.0	73.6

factors are satisfied, this difference can be significantly improved.

b. Analyzing sentiment of each tweet

From 617 tweets in the testing set, we have extracted the sentiment of each tweet, which is presented in Table 7. where 1s – 1o means that the tweet expresses a single sentiment for one object, and the remaining symbols have similar meanings.

The results indicate that tweets presenting a single sentiment for a single object and two different sentiments for two objects achieved better performance than other cases. Since such tweets have a balance between the words indicating objects and the sentiment words, it is easy to detect and assign a label by using the algorithm. Besides, unlike previous methods, our method analyzes the sentiment of each tweet by combining the sentiment of objects existing in each tweet. The previous methods only extract sentiments of users while the objects at which the sentiments are directed remained unknown. Moreover,

most of the other methods only mention sentiment of each tweet without explicitly specifying the sentiment for each object existing in each tweet. The effectiveness of the proposed method can be further clarified by the data in Table 7 and the result of Ex.2, we not only know the sentiments of the speaker but also grasp objects of those sentiments. (e.g., “I do not like rain” shows the *negative* sentiment with “rain” object), and objects sentiment on the entire tweet is also expressed (e.g., for *weather* topic, most of the opinions give *positive* sentiment with “wind” object, *very positive* with “snow”, etc.). Therefore, the result of the proposed method contains more information.

c. Analyzing sentiment from tweets according to each object

To evaluate the performance of the proposed method, we requested five humans as experts defining sentiment of twelve objects by integrating sentiments from 617 tweets. Then, these results are used to compare to the results which our method has given by using the mean absolute error. The actual and predicted result of sentiment integration from entire tweets according to each specific object are shown in Table 8.

From Table 8 and by applying Equation 12, we have the mean absolute error of the proposed method as $MAE_{12} = \frac{1}{2 \times 12} \times 5.06 = 0.21$, which is an acceptable error ratio.

Table 8
The actual and predicted result of sentiment integration from tweets according to specific objects

		o_1	o_2	o_3	o_4	o_5	o_6	o_7	o_8	o_9	o_{10}	o_{11}	o_{12}
Actual	total	96	87	81	74	42	36	81	29	109	23	95	96
	o_j^+	90	50	74	60	20	24	21	17	97	5	76	23
	o_j^-	6	37	7	14	22	12	60	12	12	18	19	73
Score	o_j^+	0.94	0.57	0.91	0.81	0.48	0.67	0.26	0.59	0.89	0.22	0.80	0.24
	o_j^-	0.06	0.43	0.09	0.19	0.52	0.33	0.74	0.41	0.11	0.78	0.20	0.76
	$o_j^+ - o_j^-$	0.88	0.14	0.82	0.62	-0.04	0.34	-0.48	0.18	0.78	-0.56	0.6	-0.52
	total	70	72	71	57	30	28	70	22	88	18	80	80
Predicted	o_j^+	6	50	50	47	14	15	31	17	80	15	61	9
	o_j^-	64	22	21	10	16	29	44	10	8	3	19	71
	$o_j^+ - o_j^-$	0.09	0.69	0.70	0.82	0.47	0.34	0.41	0.63	0.91	0.83	0.76	0.11
Score	o_j^+	0.91	0.31	0.30	0.18	0.53	0.66	0.59	0.37	0.09	0.17	0.24	0.89
	o_j^-	-0.82	0.38	0.4	0.64	-0.06	-0.32	-0.18	0.26	0.82	0.66	0.52	-0.78
	$o_j^+ - o_j^-$												

Table 9
The result of integrating sentiments from tweets according to specific objects
(Very positive, Positive, Neutral, Negative, and Very Negative are denoted by VP, P, NE, N, and VN, respectively)

	Rain	Wind	Sunshine	Snow	Fog	Ice	Cloud	Hail	Rainbow	Frost	Temperature	Storm
Actual	VP	P	VP	VP	NE	P	N	P	VP	VN	VP	VN
Predicted	VN	P	P	VP	NE	N	N	P	VP	VP	VP	VN

Table 9 represents the final result of sentiment integration from sentiments of tweets according to objects. The result shows that the method generally did not perform well with the very positive sentiment (the true predicted ratio is $\frac{3}{5}$) and the very negative sentiment (the true predicted ratio is $\frac{1}{2}$); however, it has performed well with the two remaining sentiments (the true predicted ratio is 1 in both cases). Moreover, the average accuracy (78.9% by applying Equation 11) can prove that this method is relatively good in determining the sentiment of each object by integrating sentiments from entire tweets.

From the above analysis, we find that the results of the sentiment analysis for each object using the proposed method is fairly good in terms of the error ratio and achieved information.

6. Conclusion and Future Work

This study proposed a method for analyzing the overall sentiment related to each object from the entire set of tweets by sentiment integration of each object from different tweets. The method includes the following steps: First, the proposed method extracted features to determine the objects and the sentiments of the objects. Second, the sentiment in each tweet was determined in detail by clearly presenting some information as objects and the sentiment related to these objects. Finally, the sentiment related to a specific object was analyzed by integrating the sentiments from all the tweets. The experimental results show that the proposed method significantly improved the performance in terms of the error ratio and achieved information. Such results can help, for example, a manufacturer grasp the users' opinion about parts of the product; using this, the manufacturer can come up with the right development strategy for the product, which can create products that meet the needs of the users.

There are however some limitations of the proposed approach, which include the impact of sarcastic tweets and tweets containing conditional sentences that cannot be considered in the sentiment analysis. In future, we plan to address the limitations of the method proposed in this study; especially, we hope to analyze the impact of sarcastic tweets and tweets containing conditional sentences on sentiment analysis, which would be another interesting research direction.

Acknowledgment

This work was supported by the 2018 Yeungnam University Research Grant. And this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2017R1A2B4009410).

References

- [1] M.Z. Asghar, A. Khan, S. Ahmad, M. Qasim and I.A. Khan, Lexicon-enhanced sentiment analysis framework using rulebased classification scheme, *PloS One* **12**(2) (2017), e0171649.
- [2] M.Z. Asghar, F.M. Kundi, S. Ahmad, A. Khan and F. Khan, T-SAF: Twitter sentiment analysis framework using a hybrid classification scheme, *Expert Systems* **35**(1) (2018), <https://doi.org/10.1111/exsy.12233>.
- [3] M. Ahmad, S. Aftab and I. Ali, Sentiment analysis of tweets using svm, *Int J Comput Appl* **177**(5) (2017), 25–29.
- [4] S. Baccianella, A. Esuli and F. Sebastiani, Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2010*, 1723 May 2010, Valletta, Malta, 2010, <http://www.lrec-conf.org/proceedings/lrec2010/summaries/769.html>.
- [5] S. Bird and E. Loper, Nltk: The natural language toolkit. In: *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions*. p.31. Association for Computational Linguistics, 2004.
- [6] A. Dey, M. Jenamani and J.J. Thakkar, Senti-N-Gram: An n-gram lexicon for sentiment analysis, *Expert Syst Appl* **103** (2018), 92–105.
- [7] P. Gamallo and M. García, Citius: A naive-bayes strategy for sentiment analysis on english tweets. In: *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@ COLING 2014*, Dublin, Ireland, 2014. pp. 171–175. <http://aclweb.org/anthology/S/S14/S14-2026.pdf>.
- [8] M. Hu and B. Liu, Mining and summarizing customer reviews. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2004, pp. 168–177.
- [9] Z. Huang, W. Xu and K. Yu, *Bidirectional lstm-crf models for sequence tagging*, arXiv preprint arXiv:1508.01991, 2015.
- [10] J. Jankowski, P. Ziemba, J. Watrobski and P. Kazienko, Towards the Tradeoff Between Online Marketing Resources Exploitation and the User Experience with the Use of Eye Tracking, Intelligent Information And Database Systems, ACIIDS 2016, Edited by: N.T. Nguyen, B. Trawinski, H. Fujita and T.P. Hong, *Book Series: Lecture Notes in Artificial Intelligence*, Volume:962, 2016, pp. 330–343. DOI: 10.1007/978-3-662-49381-6_32
- [11] Y. Jo and A.H. Oh, Aspect and sentiment unification model for online review analysis, In: *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, ACM, 2011, pp. 815–824.

- [12] A.Z. Khan, M. Atique and V. Thakare, Combining lexicon-based and learning-based methods for twitter sentiment analysis, *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)* (2015), 89.
- [13] S.M. Kim and E. Hovy, *Extracting opinions, opinion holders, and topics expressed in online news media text*. In: Proceedings of the Workshop on Sentiment and Subjectivity in Text, Association for Computational Linguistics, 2006, pp. 1–8.
- [14] B. Liu, Sentiment analysis and opinion mining, *Synthesis Lectures on Human Language Technologies* **5**(1) (2012), 1–167.
- [15] N.T. Nguyen, Advanced methods for inconsistent knowledge management, Springer Science & Business Media, 2007.
- [16] A.G. Pablos, M. Cuadros and G. Rigau, W2VLDA: Almost unsupervised system for aspect based sentiment analysis, *Expert Syst Appl* **91** (2018), 127–137. <https://doi.org/10.1016/j.eswa.2017.08.049>.
- [17] I. Pavlopoulos, *Aspect based sentiment analysis*, Athens University of Economics and Business, 2014.
- [18] R. Remus, U. Quasthoff and G. Heyer, Sentiws – a publicly available germanlanguage resource for sentiment analysis, In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10) European Language Resources Association (ELRA)*, Valletta, Malta, 2010.
- [19] L.M. Rojas-Barahona, Deep learning for sentiment analysis, *Language and Linguistics Compass* **10**(12) (2016), 701–719.
- [20] C. dos Santos and M. Gatti, Deep convolutional neural networks for sentiment analysis of short texts. In: *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 69–78.
- [21] Z.W. Saldan a, *Sentiment analysis for exploratory data analysis*, Programming Historian, 2018.
- [22] D. Tang and M. Zhang, Deep learning in sentiment analysis, In: *Deep Learning in Natural Language Processing*, Springer, 2018, pp. 219–253.
- [23] H. Thakkar and D. Patel, *Approaches for sentiment analysis on twitter: A state-of-art study*, arXiv preprint arXiv:1512.01043, 2015.
- [24] P.D. Turney, Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2002, pp. 417–424.
- [25] V.C. Tran, N.T. Nguyen, H. Fujita, D.T. Hoang and D. Hwang, A combination of active learning and self-learning for named entity recognition on twitter using conditional random fields, *Knowledge-Based Systems* **132** (2017), 179–187.
- [26] L. Tian, C. Lai and J.D. Moore, Polarity and intensity: The two aspects of sentiment analysis. arXiv preprint arXiv:1807.01466, 2018.
- [27] J. Zhao, X. Gui and X. Zhang, Deep convolution neural networks for twitter sentiment analysis, *IEEE Access* **6** (2018), 23253–23260. <https://doi.org/10.1109/ACCESS.2017.2776930>.