



Bidirectional GRU for Targeted Aspect-Based Sentiment Analysis Based on Character-Enhanced Token-Embedding and Multi-Level Attention

Esther Irawati Setiawan^{1,2,3*} Ferry Ferry³ Joan Santoso^{1,3} Surya Sumpeno^{1,2}
 Kimiya Fujisawa⁴ Mauridhi Hery Purnomo^{1,2*}

¹Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Indonesia

²Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Indonesia

³Department of Informatics, Institut Sains dan Teknologi Terpadu Surabaya, Indonesia

⁴School of Media Science, Tokyo University of Technology, Tokyo, Japan

* Corresponding author's Email: hery@ee.its.ac.id, esther16@mhs.ee.its.ac.id

Abstract: The user's feedback on healthcare services is usually based on ratings from post-service questionnaires. However, in order to get a clear view of the user's perspective, online text reviews need to be analyzed. We combined targeted and aspect-based sentiment analysis by multi-level attention to get a specific user sentiment on a target of an aspect. The multi-level attention consists of Target-level and Sentence-level attention. Our proposed framework is based on Bidirectional Gated Recurrent Unit. Bi-GRU is commonly known to have comparable results compared to LSTM while having lesser computational complexity. We also utilized Bidirectional LSTM based Character-Enhanced Token-Embedding to handle out of vocabulary words and misspelling to avoid error in detecting sentiment. We created a dataset of online healthcare reviews from 2018-2020, targeting the name of the hospital or department, with ten aspects: cleanliness, cost, doctor, food, nurse, parking, receptionist and billing, safety, test and examination, and waiting time. To improve the results of our proposed method, we calculated polarity weight to handle imbalanced aspects in the dataset. We classified these reviews into three polarities, which are positive, negative, and neutral. Based on our experiments, we achieved the best F1-Score of 88%.

Keywords: Targeted aspect-based sentiment analysis, Bidirectional GRU, Character-enhanced token-embedding, Multi-level attention.

1. Introduction

Analyzing public opinion currently is essential to gain insight from society on product or brand of companies. Opinion may be obtained from sites, social media tweets, reviews from search engines, and online review platforms. As technology rapidly grows, automated opinion analysis is required to obtain the optimal analysis. The widely used approach is with Sentiment Analysis and Affective Computing [1, 2].

Sentiment Analysis is a task in natural language processing to achieve polarity of opinion, which is positive, negative, or neutral [3]. Sentiment Analysis has been utilized in various domains such as restaurant reviews [4], public opinions [5], stock

market prediction [6, 7], and hotel [8, 9]. Sentiment Analysis has also been performed in various languages, such as English [10], Chinese [11], Arabic [12], and Indonesian [4].

Generally, traditional analysis of opinion in Sentiment Analysis only gets a whole view of comment, without considering aspects inside the sentence-level. Aspect Based Sentiment Analysis identifies sentiment expressed for an aspect [13]. However, public opinion usually consists of more than one aspect. Incorrect polarity detection happens due to the user's opinion text, which is aimed at two aspects with different polarities. Therefore, it is a challenge to extract sentiment polarity correctly from every aspect. A sentence with multiple aspects is displayed in Fig. 1. The target in this review is HospitalX, which could be the name of the hospital

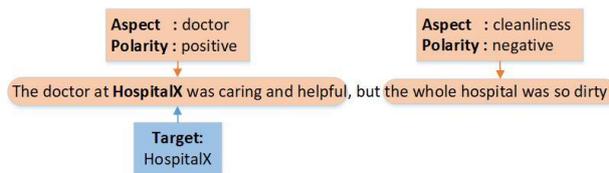


Figure. 1 Extracted Review with multi-aspect on a target

or department. The aspects are doctor, which have positive polarity, and cleanliness with negative polarity.

Deep Learning has been widely used for Sentiment Analysis as a better alternative for classical machine learning since it can reduce manually handcrafting features. The recent approach is with Bidirectional LSTM [14] and Bidirectional Gated Recurrent Unit [15].

Attention has been proven to improve the performance of several sentiment analysis approaches by focusing on the aspect [16]. Combining these several models of attention to multi-level attention has shown an increase in results [17]. Word Embedding is used in sentiment analysis since it has an advantage that can include the semantic meaning of words in a text. There are a variety of word embedding models, such as Glove [18]. Glove focuses on co-occurring words. Its embeddings refer to the probability of two terms occurring together. A more novel approach to embedding for Text Classification is Character Embedding, which may overcome misspelling and out of vocabulary words [19]. Another method is Aspect Embedding for Sentiment Analysis [10].

Thus our research effort is to classify sentiment polarity with Targeted Aspect Based Sentiment Analysis. Targeted Aspect Based Sentiment Analysis is the task to analyze sentiments to achieve a polarity of sentiment aimed at a given target [10]. This is one of the fine-grained aspect-level approach to sentiment analysis [20].

In this Targeted Aspect-Based Sentiment Analysis task, suppose that there's a sentence consist of multiple word $s = [w_1, w_2, w_3, \dots, w_n]$, the target of the sentiment $t = [t_1, t_2, t_3, \dots, t_n]$ was one or more terms in the sentence s that represent the name of department or hospital in the sentences and aspect a was a part of a hospital or department that was already predefined. Our goal is to predict the sentiment polarity p for aspect a toward target t that exists in the sentence s where the polarity of sentiment could be defined as three category $p = \{positive, negative, none\}$. For example, the sentiment polarity in the sentence "XYZ doctors was caring and helpful, unlike the ABC doctors who would throw you out of the room as fast as possible"

with target "ABC" toward aspect "doctor" have polarity negative while for target "XYZ" it would be the "positive".

Our proposed framework, as displayed in Fig. 2 has several contributions as follows:

- Targeted Aspect Based Sentiment Analysis based on Bidirectional GRU on a given target and multi-aspect sentiment polarity prediction.
- We used character-enhanced token-embedding [21] and multi-level attention [10] to achieve better results and analyzed their performance on the Sentic LSTM and Bidirectional GRU model.
- A dataset for Targeted Aspect-Based Sentiment Analysis with multi-target and multi-aspect. Our dataset was taken from online healthcare reviews with ten aspects: cleanliness, cost, doctor, food, nurse, parking, receptionist and billing, safety, test and examination, and waiting time.
- Handling imbalanced dataset by adding polarity weight. Thus we multiplied the loss with the weight of its sentiment class.

Our paper is divided into six parts and organized as follows. We describe the introduction in this section. Section 2 describes the previous study in Targeted Aspect-Based Sentiment Analysis. Section 3 discusses our proposed framework. Section 4 explains how to construct a dataset for Targeted Aspect-Based Sentiment Analysis. Section 5 describes the experiments and analysis in this study. Section 6 outlines about the conclusion and future works.

2. Related works on targeted aspect-based sentiment analysis

In this section, we present some of the previous works relevant to our model: aspect-based sentiment analysis, targeted aspect-based sentiment analysis, and character embedding.

Aspect-based sentiment analysis is an enhancement in the sentiment analysis task, which is motivated by the need to get more fine-grained information in the text. The earlier work of this task consisted of hand-crafted feature engineering [22-24], which requires a technical foundation in linguistics.

Neural networks have changed how Artificial Intelligence works. Now, most AI tasks use neural networks and deep learning as its core, and this includes the Aspect-Based Sentiment Analysis task as well. Many models have been used in experiments on how to solve this task. Research in [25] uses Recursive Neural Network, which processes text in

the tree kind of ways. On the other hand, [26] uses CNN as the base of the model.

Aside from these methods, many of the Aspect-Based Sentiment Analysis tasks use a recurrent network, such as LSTM [27]. The recurrent network was chosen because of its ability to represent things sequentially without losing the past information, which is important in analyzing text. Attention as one of the most popular methods that have also been used in the ABSA task. Typically, LSTM was utilized for the aspect model to gain better representation [16].

Targeted Aspect-Based Sentiment Analysis is a task with the aim to classify sentiment polarity on a set of aspects owned by each entity in a text. This task combines two common sentiment classification tasks, which are aspect-based sentiment analysis and targeted sentiment analysis, which was first introduced by the article [20]. Authors contribute by sharing the TABSA dataset they used. The manuscript also contrasts the classification result of the hand-engineered features and the neural-network based models, which surprisingly revealed that the hand-engineered feature model is better than neural-network model. The process of determining the target could utilize tools such as Named Entity Recognition [28][29] or obtaining noun phrases for the aspect that consists of the multi-word [30][31]. However, in this proposed model, since performance validation is needed by assuming that the input is accurate, the process of determining the target sentiment is carried out manually.

One of the deep learning approaches on the TABSA task was shown in the paper Sentic Long Short Term Memory (LSTM) [28], which seeks to improve LSTM by incorporating common sense thus the model could capture the sentiment implicitly expressed in the text. To incorporate common sense, SenticLSTM first attempts to capture the concept presented in the text and then converts it to vector representation using affectiveSpace. AffectiveSpace enables to process of semantic features associated with sentiments being used [32]. However, it does not focus on out of vocabulary words and misspelling.

The sequential model is one of the basic principles in Natural Language Processing because the context of the preceding word is needed to determine the meaning of the current word. Several models have been created to handle this sequential problem, which GRU is one of those models. GRU has similarities with LSTM in the way they process the text. The difference between these two models was that GRU doesn't have a cell state. Bi-GRU also has less complex computing compared to LSTM while maintaining comparable performance. Since

Bi-GRU has a less complicated structure, Bi-GRU could train faster compared to LSTM.

Since it was published, many researches use GRU for many different tasks, including sentiment analysis. Although not as popular as LSTM, Bi-GRU was also a model that can be found in the task of sentiment analysis and its variant. The types of sentiment analysis are Aspect-based sentiment analysis [33, 34], Targeted Sentiment Analysis [35], or even Targeted Aspect-based Sentiment Analysis [36].

Another work on TABSA has been published [36], where authors use external "memory chains" equipped with delayed memory update mechanism to track each entity in the text. Unlike other sentiment analysis models that use LSTM or GRU to represent a word, there are three parts: key vector, memory vector, and delay vector for its mechanism of gating. The paper reported that the model manages to represent an entity quite well, which shown by a quite high accuracy compared to even the SenticLSTM.

Deep Learning algorithms need scalar values or matrices in order to process words. The words are converted into vectors while maintaining a semantic context. This representation is known as word embedding [37]. Moreover, character embedding attempts to represent a character in the form of vector. Character embedding is needed because the lesser vocabulary needs to be represented and its ability to solve out of vocabulary words, even though word embedding existed.

Many studies use character embedding to solve several different tasks as well, i.e., POS Tagging [38], Named Entity Recognition [39], machine translation [40], or language modeling task [41]. There are two methods of using character embedding. The first method is using character embedding on its own when constructing sentence representation as in [38, 40-41]. Another way is to combine it with word embedding and use the combined vector as the final representation for each word. Some examples of the paper that used this type of method are in [39] and [21].

Character embedding also affects sentiment analysis [42]. For example, word and character embedding is used to predict sentiment from a short text. Another work using character embedding is also shown in [19] and [43].

3. Targeted-aspect based sentiment analysis with bidirectional GRU

This section will discuss the details of the proposed Targeted Aspect Based Sentiment Analysis model. The architecture of the proposed model is displayed in Fig. 2. The pseudocode of our proposed

framework's training phase is shown in Algorithm 1, and the testing phase is displayed in Algorithm 2. In our model, we assume that each sentence $s = [w_1, w_2, w_3, \dots, w_n]$ consists of n words. The target of the sentiment $t = [t_1, t_2, t_3, \dots, t_p]$ is one or more words in the sentence \underline{s} that represent the department's name or hospital in the sentences obtained from the function *getTarget*. For each word in the sentence s , m, n size of character $ch = [ch_{1,1}, ch_{1,2}, ch_{1,3}, ch_{2,1}, \dots, ch_{m,n}]$ would be obtained.

Algorithm 1 Pseudocode for Training Targeted Aspect-Based SA

Input : D_{train} : dataset training, Y : target class, A : list of aspects, T_POS : Target positions, cw : class weight, dp : dropout probability, epoch: total maximum epoch

Output : model

```

1. for I = 1 to epoch
2.   for data in  $D_{train}$ 
3.     for T in getTarget( $T\_POS, D_{train}$ )
4.       S = tokenize(data)
5.       for aspect in A
6.         for w in S
7.            $X_{we} = \text{wembedding\_lookup}(w, W_{we})$ 
8.           for ch in w
9.              $X_{ch} = \text{BiLSTM}(ch)$ 
10.          end for
11.          X.append( $[X_{we}, X_{ch}]$ )
12.        end for
13.        X = dropout(X, dp)
14.         $H_{GRU} = \text{BiGRU}(X)$ 
15.         $V_t = \text{target\_attention}(H_{GRU}, T)$ 
16.         $V_s = \text{sentence\_attention}(H_{GRU}, V_t, \text{aspect})$ 
17.         $h_{out} = V_s W_{out} + b_{out}$ 
18.        output = softmax( $h_{out}$ )
19.        loss = compute_loss(output, Y, cw)
20.        DO Backpropagation(loss)
21.      end for
22.    end for
23.  end for
24. end for

```

X contains a set of inputs from words in the dataset. Each element of X is the result of a concatenation of word embedding stored in X_{we} and character embedding of the word formed with the help of the character BiLSTM stored in X_{ch} . *wembedding_lookup* (w, W_{we}) is explained in section 3.2 Eq. (1).

HGRU is the hidden output of BiGRU. BiGRU is a Bidirectional GRU model which consists of forward and backward GRU, whereas the GRU cell is explained in section 3.3. W_{out} is the weight of the fully connected layer used to determine the class polarity of data, i.e., pos, neg, and neutral.

Algorithm 2 Pseudocode for Testing Targeted Aspect-Based SA

Input : D_{test} : dataset testing, A : list of aspects, T_POS : Target positions, cw : class weight, dp : dropout probability

Output : c : class

```

1. for data in  $D_{test}$ 
2.   for T in getTarget( $T\_POS, D_{test}$ )
3.     S = tokenize(data)
4.     for aspect in A
5.       for w in S
6.          $X_{we} = \text{wembedding\_lookup}(w, W_{we})$ 
7.         for ch in w
8.            $X_{ch} = \text{BiLSTM}(ch)$ 
9.         end for
10.        X.append( $[X_{we}, X_{ch}]$ )
11.      end for
12.      X = dropout(X, dp)
13.       $H_{GRU} = \text{BiGRU}(X)$ 
14.       $V_t = \text{target\_attention}(H_{GRU}, T)$ 
15.       $V_s = \text{sentence\_attention}(H_{GRU}, V_t, \text{aspect})$ 
16.       $h_{out} = V_s W_{out} + b_{out}$ 
17.      output = softmax( $h_{out}$ )
18.      loss = compute_loss(output, Y, cw)
19.      c = argmax(output)
20.    end for
21.  end for
22. end for

```

b_{out} is the bias of the fully connected layer, and h_{out} is the output of the fully connected layer. *output* is the softmax result of the fully connected layer. V_t is a result vector of target attention, and V_s is a vector result of sentence attention. c is the output class, which is the polarity output of each aspect. *Backpropagation* (*loss*) is a function for updating the weight of the proposed neural network.

target_attention (H_{GRU}, T) function takes an input of hidden result from BiGRU, and the Target of sentiment in T . Calculation of *target_attention* is explained in section 3.4 Eq. (5). The *sentence_attention* (H_{GRU}, V_t, A) will take an input of hidden result from BiGRU, a vector from target_level_attention, and aspect embedding. The calculation is explained in section 3.4 Eq. (8).

The overall design of the model could be divided into three parts, which are character representation, word representation, and attention model, as displayed in Fig. 3. Character level information would map each character to its unique representation, while word would generate word representation. Our model's attention is inspired by the model of [28] where the attention model consists of target-level attention and sentence-level attention.

First, the model takes as input the words and character of each word in the sentence. Not only the sentence, for each text, the model also requires an

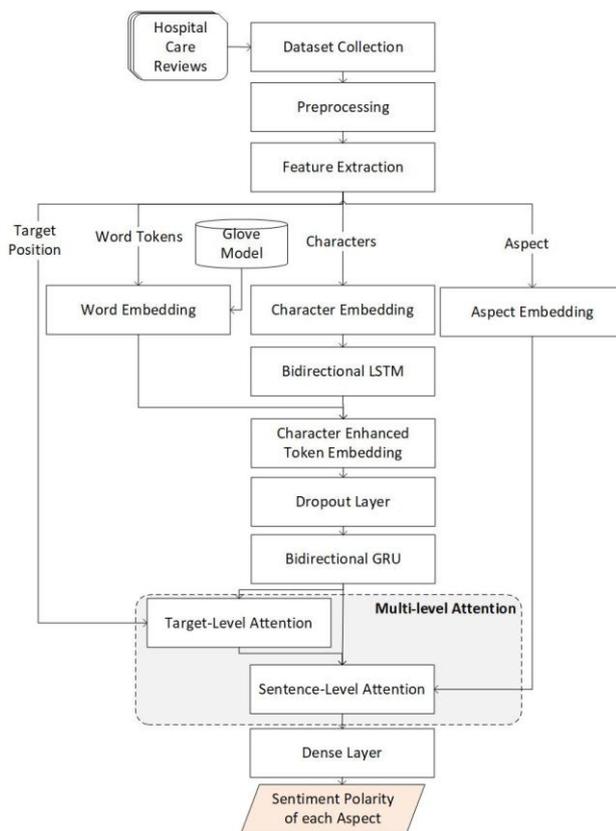


Figure. 2 Proposed framework

aspect that needs to be predicted its sentiment polarity and target word position in the sentence as an input. Each input needs to be converted into its respective vector representation to process the model. This unique vector representation is called embedding.

3.1 Preprocessing and feature extraction

Preprocessing is the extraction of every aspect and sentiment owned by each target in a review. From each sentence, tokenization obtains word tokens, characters, aspect, and target position.

The data will then be further processed by indexing it. In this study, indexing will be performed on the words in each review, the letters of the word, aspects that exist in all reviews in the dataset, and sentiment polarity. Indexing begins with getting every aspect, word, letter, and polarity. Each unique letter, word, aspect, and polarity will be given a unique number that represents the letter, word, aspect, and polarity. The final step of indexing is changing each word, letter, aspect, and polarity contained in each review into an index that represents the word, letter, aspect, and polarity. The sentiment polarity included in the dataset is positive and negative, so it needs to be added to neutral or none polarity at this stage.

When implementing the sentiment analysis model, a token may not be found in the dataset, called the out

of vocabulary (OOV) problem. In this study, we address this problem with a combination of a special index method, the unknown index, and the use of representations of each character that compiles the OOV word. The representation is obtained through the character enhanced token embedding method. An unknown index will be used to represent a token that is not in the dataset. In this study, the unknown index is represented by the number 0.

Apart from a word, each letter in the sentence will also be indexed. The letters obtained will then be used in the character embedding process. Indexing of letters will produce a 3-dimensional array containing each letter in each token for each review. Tokens in a review have different lengths from one another. Thus we need to make the length of the sequence equal. Padding is necessary so that the final length of the array of letters is the largest token length in a sentence. Apart from letters and words, indexing of the existing aspects and polarity will also be carried out. The aspects that will be used in this study are cleanliness, cost, doctor, food, nurse, parking, receptionist and billing, safety, test and examination, and waiting time. These ten aspects will be indexed so that they can be used in the machine learning model created.

Finally, indexing will be conducted on sentiment polarity, which can contain positive, negative, or neutral values. Neutral is an additional polarity given because every review contained in the dataset will only show aspects that have good sentiment is positive and negative sentiment.

After indexing, the final step in the preprocessing phase of this study is to get a ratio of comparison between each type of sentiment polarity owned by each aspect. The first step is to find the number of targets in each dataset. After the target number is reached, the next step is to get the number of positive and negative sentiments.

The number of neutral sentiments is derived from reducing the number of targets by the number of positive and negative sentiments. Then, the sum of each type of sentiment polarity will be divided by the number of targets so that each sentiment polarity ratio is obtained.

3.2 Embedding layer

Character includes rich details that would usually be missed in word embedding, such as the word's morphological part [38]. Although word embedding alone wouldn't have a problem for representing general words such as 'happy' or 'sad', word embedding alone could miss much information on word that has strong morphological connection such as 'sonographer' and 'sonography' which might lead



Figure. 3 Layers in targeted aspect based sentiment analysis

the model to be unable to detect the sentiment in the sentence for specific target and aspect.

The word embedding in our model is calculated based on Eq. (1).

$$wembedding_{lookup(w, W_{we})} = \begin{cases} x_{oh} W_{we} & \text{IF } x \text{ IN vocabulary}(W_{we}), \\ unk_embed & \text{else} \end{cases} \quad (1)$$

x_{oh} is a one hot encoding representation of word w . W_{we} is an embedding matrix from Glove [18]. unk_embed is a vector initialized randomly if the word is not available in the pre-trained Glove model.

The second step of the model is to perform character embedding. The input character needs to be converted into d dimensional vector representation. This mapping process is called embedding. In this model, the character embedding was first initialized randomly, and the representation would then be trained along with the model to achieve better representation.

The embedding for each character in the sentences will then be fed into Bi-LSTM layer. This process was required since the embedding represents the character only and does not consider any other character that has also contributed to the forming of the sentences. This Bi-LSTM layer will generate a

new vector representing each character in relation to the other characters in the sentences X_{ch} .

The next process in the model was to find the final representation of the words. We concatenated the result of the character Bi-LSTM layer and word embedding to get the model's final representation. This process was inspired by a character-enhanced token embedding [21]. Using this method, the representation for each word doesn't only consider word-level information that usually has better syntactic and semantic representation, but also takes into account character-level information that is normally better used in handling morphological representation. Using this method also allows the connection between an OOV word and another word that existed in a dictionary, as long as word's sequence is similar.

3.3 Bidirectional GRU

In our framework, the word representation would then be fed into a dropout layer with a dropout probability of 0.5 to avoid overfitting. The result of dropout would then be fed into the Bi-GRU layer H_{GRU} . Like the character information, the word representation generated from the dropout layer would still only represent the word and not take into account its relationship to another word that composed a sentence. That's why the dropout layer

result needs to be fed into another layer, which is Bidirectional GRU layer. The Bidirectional GRU layer was similar to the Bi-LSTM layer in which the word is processed in forward and backward sequences to generate the relationship between each word. The difference between these two models was that unlike Bi-LSTM, Bidirectional GRU does not use the cell state and uses the hidden state to transfer information. Bidirectional GRU is also widely considered to have similar performance compared to LSTM while having less computational complexity. The Bidirectional GRU layer may be split into two gates, which are update gate and reset gate. The update gate is responsible for determining how much of the past information needs to be passed along to the future. The update gate could be formulated as in Eq. (2).

$$Z_t = \sigma(W_z \cdot X_t + U_z \cdot h_{t-1} + b_z) \quad (2)$$

Where Z_t is the update gate for timestep t , and W_z and U_z is the gate's weight, while b_z was the bias. Like LSTM, the gate takes into account both the current input X_t and the output of the previous step h_{t-1} . The result of the calculation would then be fed into sigmoid activation.

The second gate in the GRU was the reset gate. The reset gate in GRU has a different task from the update gate. If the update gate handles which information should be stored for the future, the reset gate handles how much the past information needs to be forgotten in the future calculation. The reset gate is formulated as in Eq. (3).

$$r_t = \sigma(W_r \cdot X_t + U_r \cdot h_{t-1} + b_r) \quad (3)$$

The reset gate has a similar formula to the update gate. The difference is only in the weight used and the function of the gate itself. GRU uses memory content to make the reset gate handle its responsibilities. The GRU memory content could be formulated as in Eq. (4).

$$h'_t = \tanh(W_{h'} \cdot X_t + r_t \odot (U_{h'} \cdot h_{t-1})) \quad (4)$$

The memory value for the timestep t is h'_t . The formula used for memory content is similar to the GRU gate layer. The two components that separated it from the other GRU layer were the activation unit and the Hadamard calculation. \odot is the hadamard product operation, which is an element-wise multiplication. The Hadamard calculation was made between the result of the reset gate and the calculation between the weight and the previous output. Unlike

the gate, the memory content uses the tanh activation unit.

The final equation of each timestep was to generate the current timestep output to be used in the next step. The output equation could be formulated as in Eq. (5).

$$h_t = Z_t \odot h_{t-1} + (1 - Z_t) \odot h'_t \quad (5)$$

3.4 Multi-level attention layer

One of the drawbacks of a sequential model, such as Recurrent Neural Networks is the inability of the model to recognize information longer. The attention mechanism is one of the methods that can be used to deal with these problems. The idea of attention mechanism is to focus the model on certain parts of the input associated with the goals to be accomplished by the model. For example, in predicting sentiment from the cost aspect, we do not need to pay attention to every word in the sentence, but only the words such as "price" or "cost". In contrast, things related to the taste of food or other items can be ignored or not taken into account in the calculation. The model must be able to determine the weight of each word in the data.

In our work, the result of the Bidirectional GRU layer would then be used in the two different attention processes, which are target-level attention V_t and sentence-level attention V_s . Target-level attention would represent the target, while sentence-level attention would be used as the final representation of the text. Fig. 4 displays examples of attention weights. The depth of color shows the significance of a word in a sentence. Our model would concatenate the target attention vector and the hidden representation of each word in the sentence to incorporate target in the text.

$$\begin{aligned} target_{attention(H,T)} &= V_t = \\ H_t \alpha &= \sum_{j=1} \cap j \in T \alpha_j h_{tj} \end{aligned} \quad (6)$$

The target representation vector is determined by variable H and α , as shown in Eq. (6). H is the output of BiGRU of the target, α is an attention weight vector for every target word. $H_t = [h_{t1}, h_{t2}, \dots, h_{tm}]$ is the hidden output from timestep of BiGRU in T . $T = [t_1, t_2, \dots, t_n]$ consists of a target position of n words which are the target in the dataset. α of a target needs to be obtained first before the target representation vector. α is a target attention vector where $\alpha = [\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n]$. α_j is an attention vector of target t_j , which is the target position. α from a target is calculated as in Eqs. (7) - (8).

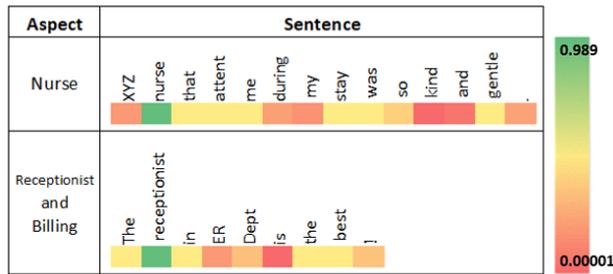


Figure. 4 Examples of attention weights of two sentences

$$h_{tatt} = \tanh(V_a H_t) \tag{7}$$

$$\alpha = \text{softmax}(W_a h_{tatt}) \tag{8}$$

V_a is a vector for attention in the first layer with size $l \times d_{va}$, where d_{va} is the dimension size of attention hidden layer. W_a is the weight matrix in the output layer of attention model. After the multiplication of the weight, the attention vector will be passed to softmax. The attention weight is obtained for each word on the target with a total amounting to 1 for each word.

Unlike attention at the target level, attention at the sentence level takes into account all the words in the sentence as its input. Attention at the sentence level also takes into account aspects that will be sought for its sentiment and attention to the target as input. Attention at the sentence level considers aspects by aspects embedding which is initialized randomly and will have more accurate representations as training is carried out on the model. The result of attention at the sentence level is the final representation as input for the fully-connected layer. The attention vector at the sentence level can be formulated as follows as Eq. (9).

$$\begin{aligned} & \text{sentence_attention}(H, V_t, A) \\ & = V_s = H \beta = \sum_{j=1} \beta_j h_j \end{aligned} \tag{9}$$

Similarly, with target level attention, the sentence level attention vector is influenced by two factors, namely H and β . H is the output of Bidirectional GRU from every word in the review, whereas β will be obtained from the calculation of target level formula. $H=[h_1, h_2, h_3, \dots, h_m]$ is the hidden output of every timestep in BiGRU with length m . β is the sentence attention vector where β_j describes salience of j -th word with respect to aspect A and target T . β_j is calculated using a multilayer neural network in Eq. (10) - (12)

$$x_j = [h_j; V_t] \tag{10}$$

$$out = \tanh(W_m x_j) \tag{11}$$

$$\beta_j = \text{softmax}(V_a^T out) \tag{12}$$

Eq. (8) is used for calculating attention weight for each word in the review sentence. x_j is the input for calculating attention vector for the j -th word, which contains a vertical concatenation of hidden output BiGRU h_j from timestep j and target attention vector V_t .

W_m is the weight from the first layer of multilayer neural network with size $m \times h$, where m is the number of timestep from BiGRU, and h is the first hidden layer size of multilayer neural network. V_a is the aspect embedding of aspect a calculated from Eqs. (13) - (14). V_a^T is the transposed result from the aspect embedding vector.

$$V_a = \text{aspect_embedding_lookup}(a, W_{aembed}) \tag{13}$$

$$\begin{aligned} & \text{aspect_embedding_lookup}(a, W_{aembed}) \\ & = x_{aoh} W_{aembed} \end{aligned} \tag{14}$$

where x_{aoh} is one hot vector representation from aspect a and W_{aembed} is an embedding matrix of all aspects. This matrix is initialized randomly and trained together with a neural network.

Unlike attention at the target level that only uses the target word as input in the process of attention, attention at the sentence level takes into account every word and the target and related aspects. Specifically, the intended target is a vector obtained from the target level attention V_t , while the intended aspect is obtained from the vector representation of the aspect V_a . The first thing to do in obtaining attention vectors at the sentence level is to concatenate the results of the BiGru on each word with the vector of the results of the attention process at the target level. The objective is to get each word's attention weight in the sentence to the target being sought. The concatenation result is the final representation vector of each word, where the vector will then be multiplied by the weight W_m .

The multiplication results will then be activated using tanh. The aspect embedding vector will then be multiplied by the word vector representation with the weight variable to produce a final vector representation containing each weight of each word. Then softmax is applied to the final representation vector.

3.5 Fully-connected layer

The final representation of sentence-level attention needs to be fed into a layer of fully

connected layer to predict the sentiment polarity. This would result in the probability score of each sentiment class. *Argmax* was used to select the most likely sentiment class of the aspect. Backpropagation is needed to update the value of each variable in the model. The model needs to find the loss score first while doing backpropagation itself.

Each aspect in the dataset will be trained for the model. If the sentence doesn't have an aspect, the model will label the sentiment as 'None', which would make the polarity for each class imbalanced as the 'None' sentiment would cover most of the sentiment data. To help to balance this, we multiply the loss with the weight of its sentiment polarity as in Eqs. (15) - (16).

$$\begin{aligned} & \text{compute}_{loss}(y, t, pw) \\ & = pw \sum_{i=1}^{total_{class}} y_i \log(t_i) \end{aligned} \quad (15)$$

$$pw(i, v_{ip}, v_{pw}) = \left(\frac{1}{\left(\frac{v_{ip} v_{pw}}{n} \right)^i} \right) + 1 \quad (16)$$

Where *pw* is the polarity weight, V_{ip} is the vector index polarity containing which sentiment polarity of the current aspect of the data has. V_{pw} is the vector polarity weight per class sentiment in the dataset, and *n* is the number of class sentiment polarity in the dataset. V_{pw} is obtained from the percentage of class sentiment polarity in the dataset. *i* is the parameter that could be tuned to get a better weight comparison. The more imbalanced the data, the greater difference the weight would give, which could lead to too much extreme or too little difference in the weight. To reduce the difference, the formula adds one to the calculation.

4. Hospital reviews dataset

The dataset used in this paper was fetched from Google Review and Careopinion. The dataset consists of 9,859 reviews from five cities in United States, which are Alabama, Alaska, Arizona, Arkansas, and California. The reviews cover various aspects of 347 hospitals. There are several types of targets in this dataset, which are hospitals, clinics, and departments.

The dataset format is inspired by Semeval 2015 task 12 [44] with some adjustments. The tags are reviews, sentences, opinions. In Fig. 5, the target is "minor injuries" department with two aspects "nurse" and "receptionist & billing".

Then the dataset is then divided into three datasets for each different purpose. The three datasets mentioned are training, development, and testing, as

```
<Review id="DS001241">
<Sentences> I went to the minor injuries, I
had cut my finger with a bread knife, I had
dressed it quite well but I believed I needed
to receive professional care. Yes the waiting
time was 90 minutes, fair enough, I am
diabetic and wanted to see if I had time to
get something to eat and drink. I was told it
would be better to stay put. There was a major
incident which occurred after I had been there
30 minutes. after it was resolved, I thought
perhaps they needed time to recover themselves
so I asked if they had a time frame so I could
get something to eat or drink, the
receptionist who by the way had eaten and
drunk constantly in the time I was there
(apart from the incident time) looked at me
with fury when I asked if they knew when I was
likely to be seen. and said in a hostile and
very unkind way that they had a major incident
and they didnt know when I would be seen. I
said I was only asking a simple question. Then
they got a head of steam up and really told me
almost yelling that they were telling me they
did not know. ...</Sentences>
<Opinions>
  <Opinion Category="NURSE" Polarity="negative"
    Target="minor injuries" />
  <Opinion Category="RECEPTIONIST&
    BILLING" Polarity="negative"
    Target="minor injuries"/></Opinions>
</Review>
```

Figure. 5 Dataset example

shown in Table 1. The training dataset contains 6,662 reviews, development dataset contains 1,652 reviews and testing 1,545 reviews. The training dataset is used for the training phase of the model. Since the purpose was to train the model, this dataset has the most significant data in the dataset. Many aspects of the training dataset have more positive sentiment than negative sentiment. The exception for this would be aspect cost, parking, and safety, which has more negative reviews. Unlike other aspects in the dataset with a relatively significant disparity between positive and negative sentiments, the receptionist and billing aspect have an equal amount of positive and negative sentiment.

The development dataset is used to measure the performance of the model after being trained. This dataset would be used to obtain the best epoch with the best model. Aside from the training and development dataset, another dataset is used, which is testing dataset. The testing dataset is used for validation purposes to avoid overfitting.

5. Experiments and analysis

This section will discuss how experiments were conducted. The results of each experiment will be evaluated to see the performance of the proposed model.

Table 1. Dataset statistics: (a) training, (b) development, and (c) testing

(a)

Aspect	Training		
	Count	+	-
Cleanliness	1,404	946	458
Cost	1,288	333	955
Doctor	2,842	1,955	887
Food	986	615	371
Nurse	2,088	1,512	576
Parking	912	261	651
Receptionist & Billing	1,622	807	815
Safety	1,288	228	1,060
Test & Examination	1,443	1,202	241
Waiting Time	2,277	1,430	847

(b)

Aspect	Development		
	Count	+	-
Cleanliness	292	227	65
Cost	173	39	134
Doctor	753	517	236
Food	201	149	52
Nurse	455	347	108
Parking	178	47	131
Receptionist & Billing	377	183	194
Safety	183	33	150
Test & Examination	294	252	42
Waiting Time	448	313	135

(c)

Aspect	Testing		
	Count	+	-
Cleanliness	363	252	111
Cost	146	61	85
Doctor	608	458	150
Food	307	195	112
Nurse	450	362	88
Parking	288	108	180
Receptionist & Billing	355	161	194
Safety	160	27	133

Test & Examination	251	228	23
Waiting Time	483	338	145

5.1 Experiments scenario

In this section, we discuss our experiment scenario. The basic method of Bidirectional GRU could handle both Aspect Based Sentiment Analysis (ABSA) and Targeted Aspect Based Sentiment Analysis (TABSA) task. Specifically, if there is no target present in the sentence, the vector attention target could be represented in a specific way, which in our system, we only represent it using a zero-based vector. Thus we compare our model based on both ABSA and TABSA models. We compare our model with the following baselines:

- Sentic LSTM (SLSTM): This approach is intended to integrate common sense knowledge into the calculation. To gain common sense knowledge, we need to get the concepts in the sentence that are then converted to a vector representation using AffectiveSpace. The model takes the target position, aspect, concept, and word in the sentence as an input, and predicts the sentiment for an aspect that belongs to a particular target in the sentence [10].
- Char Sentic LSTM (CSLSTM): This model is based on Sentic LSTM [10]. The difference between the two models is on the input of the model. Apart from using word embedding alone, we incorporated the result of character Bi-LSTM [21] into Sentic LSTM.
- GCAE [26]: This model is a CNN based model for the ABSA task. Since our base model was created for the TABSA task, which could consist of multiple targets with the same aspect but different sentiment. First, we remove data from datasets that have conflicted sentiment. The model takes word embedding and an aspect as an input and model the sentiment relation from those inputs. When using this model, we ignore the target part of the dataset. If there are multiple targets with multiple aspects, all aspects would be combined into one as if they belong to a single general target.

We have evaluated our model on three datasets: hospital review, SentiHood, and Indonesian Restaurant dataset. We will elaborate on the detail of these three datasets. The first dataset was hospital review dataset, as explained in Section 4. Overall, the data collection has 1195 data with multiple targets and 8,664 data with a single target, as displayed in

Table 2. Number of targets of hospital review dataset

Target	Training	Development	Testing
Multiple Target	978	77	140
Single Target	5,684	1,575	1,405

Table 3. Number of targets of sentihood dataset

Target	Training	Development	Testing
Target with Sentiment	2,473	619	1,241
Total Aspect	2,872	701	1,396

Table 4. Number of targets of hospital review dataset

Singapore Hospital Names

Farrer Park Hospital	Ng Teng Fong General Hospital
Geneagles Hospital	Singapore General Hospital
Mount Alvernia Hospital	Tan Tock Seng Hospital
Mount Elizabeth Hospital	Thomson Medical Centre
National University Hospital	

Table 2. For the data with a sentiment, but has no target in the sentence, it would be classified as *NULL*.

The second dataset we use is SentiHood dataset. SentiHood was a common dataset used in the TABSA task. The SentiHood dataset consists of 5,215 data with 3,862 sentences containing a single target and 1,353 sentences containing multiple targets. This dataset was divided into three datasets, which are train, development, and testing. From the overall 5,215 data, we cleaned the data without sentiment from each dataset. The result after filtering is shown in Table 3.

The third dataset is the Indonesian Restaurant Dataset that was built by the authors. This is an ABSA dataset with four aspects, which are Food, Facility, Service, and Location. The data was obtained from Google Review and TripAdvisor from 75 restaurants. This dataset contains 2,735 reviews which were obtained in August, 2019

We also evaluated our model by deploying it to reviews from Google Review from Singapore Hospitals. The pandemic of Covid19 has affected hospital services. We choose ten hospitals from Singapore that have more than 100 reviews which is shown in Table 4. The dataset would then be divided into two parts based on the date the review published, before and after corona pandemic. Specifically, the date we use to divide them was 23 January, which was known as the first date a patient had been confirmed to have Corona in Singapore. Since Google Reviews doesn't publish the exact date, we

divided it from how many months ago the review was published. From review before corona, we get 1396 reviews belong to those ten hospitals while from after corona, we get 157 reviews.

The performance of our proposed model will be evaluated using F1-Score, which can be seen in Eqs. (17) - (22).

$$MacroF1 = \frac{2(MaP) \times (MaR)}{(MaP) + (MaR)} \quad (17)$$

$$MaP = \frac{1}{|D|} \sum_{t \in D} \frac{|Y_t \cap Y'_t|}{Y'_t} \quad (18)$$

$$MaR = \frac{1}{|D|} \sum_{t \in D} \frac{|Y_t \cap Y'_t|}{Y_t} \quad (19)$$

$$MicroF1 = \frac{2(MiP) \times (MiR)}{(MiP) + (MiR)} \quad (20)$$

$$MiP = \frac{\sum_{t \in D} |Y_t \cap Y'_t|}{\sum_{t \in D} Y'_t} \quad (21)$$

$$MiR = \frac{\sum_{t \in D} |Y_t \cap Y'_t|}{\sum_{t \in D} Y_t} \quad (22)$$

D is the dataset, $t \in D$ is all targets in dataset D . Y_t is the actual label being predicted, and Y'_t is the sentiment label from the prediction.

5.2 Experiments results and discussion

A comparison was made with the base model Sentic LSTM and the bi-GRU based model to obtain a better understanding of the model. The result was analyzed on the testing dataset. In this evaluation, we skipped on the aspect with none label.

The result of the comparison is shown in Table 5. We displayed the comparison between models that uses character information (CSLSTM) and not using it (SLSTM). The model that uses character embedding manages to get a significant boost on the recall performance, which suggests that considering character embedding results better in aspect detection. The accuracy of the sentiment classification has improved, which is shown by the fact that F1-score of our model have a better result for most aspects.

The comparison output between SLSTM and our model shows that although using character embedding could improve the recall, it greatly impacts precision performance. This does not happen with our model as much as with CLSTM. However, this does not occur with our model as much as with the SLSTM model that uses character embedding. In addition to sentiment classification, the model also measures the performance of aspect detection, which

Table 5. Experiments compared with previous models

Aspect	Model	Precision	Recall	F1-Score
Cleanliness	SLSTM	94.191	94.979	94.583
	CSLSTM	90.688	99.115	94.715
	Our Model	95.000	95.798	95.397
Cost	SLSTM	78.846	95.349	86.316
	CSLSTM	80.000	100	88.889
	Our Model	78.571	96.491	86.614
Doctor	SLSTM	96.084	84.840	90.113
	CSLSTM	90.385	95.190	92.725
	Our Model	92.661	95.962	94.282
Food	SLSTM	93.567	73.733	82.474
	CSLSTM	83.258	90.196	86.588
	Our Model	96.324	62.381	75.723
Nurse	SLSTM	90.071	84.950	87.435
	CSLSTM	85.933 %	94.613 %	90.064 %
	Our Model	89.677 %	92.667 %	91.148 %
Parking	SLSTM	86.232	80.952	83.509
	CSLSTM	81.921	97.973	89.231
	Our Model	80.226	89.308	84.524
Receptionist & Billing	SLSTM	97.222	81.395	88.6085
	CSLSTM	90.667	87.179	88.889
	Our Model	87.368	91.209	89.247
Safety	SLSTM	86.957	76.923	81.633
	CSLSTM	63.158	100	77.419
	Our Model	62.162	92.000	74.194
Test & Examination	SLSTM	96.939	93.596	95.238
	CSLSTM	93.182	98.204	95.627
	Our Model	96.482	93.659	95.050
Waiting Time	SLSTM	92.444	85.950	89.079
	CSLSTM	89.224	94.521	91.796
	Our Model	92.395	95.294	93.822

Table 6. Evaluation with baseline models

Model	Macro F1-Score	Micro F1-Score
SLSTM	85.415	83.599
CSLSTM	85.797	83.983
Our Model	86.162	84.378

Table 7. Comparison with GCAE model

Aspect	Model	Precision	Recall	F1-Score
Cleanliness	GCAE	95.24	84.21	89.39
	Our Model	91.97	97.86	94.82
Cost	GCAE	92.54	92.54	92.54
	Our Model	81.97	92.59	86.96
Doctor	GCAE	81.51	89.47	85.31
	Our Model	91.41	91.18	91.3
Food	GCAE	83.52	78.35	80.85
	Our Model	85.58	91.54	88.46
Nurse	GCAE	76.67	89.61	82.64
	Our Model	87.79	91.1	89.41
Parking	GCAE	82.94	91.56	87.04
	Our Model	82.25	86.34	84.24
Receptionist & Billing	GCAE	91.60	89.55	90.57
	Our Model	90.45	86.1	88.22
Safety	GCAE	96.55	95.46	96
	Our Model	63.16	80	70.59
Test & Examination	GCAE	82.35	87.5	84.85
	Our Model	96.32	95.81	96.06
Waiting Time	GCAE	89.89	93.02	91.43
	Our Model	89.03	94.85	91.85

was done in the same way as in the paper of Sentic LSTM [10]. We compare our model with the base model and our model. The measurement was done on the testing dataset. Although our model manages to achieve better recall, it still manages to achieve better precision performance on some aspects compared to the SLSTM. It also obtained better precision performance in most aspects compared to the CSLSTM.

CSLSTM has a relatively lower precision. However, our model has a better balance between precision, recall, and F1-Score than the other models.

In addition to sentiment classification, the model also measures the performance of aspect detection.

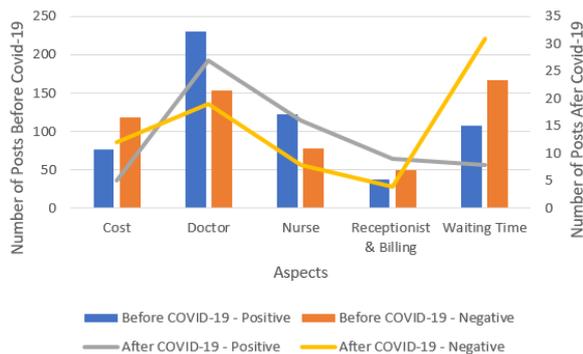


Figure. 9 Graph of sentiment before and after the pandemic

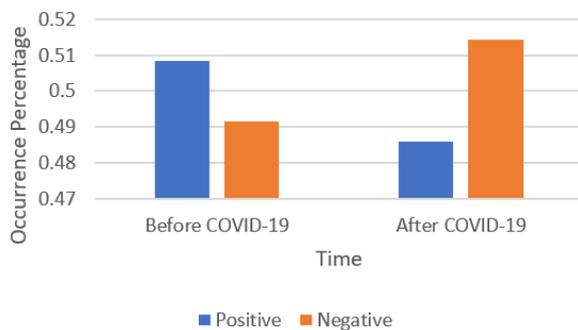


Figure. 10 Graph of sentiment of the top 5 aspects

The graph sentiment in Fig. 10 shows the outcome of the deployment of our model to the study of ten hospitals in Singapore. There are ten aspects that we use, but we select the top five of those aspects that are frequently used in the text. These five aspects were cost, doctor, nurse, receptionist, billing, and waiting time. Fundamentally, there is not much difference between the time before and after COVID-19, although there might be a difference between their pre-and post-COVID-19 ratios, a more positive aspect, such as doctor and nurse, still dominated by positive results even in the middle of the pandemic. While the aspect that is more negative after the pandemic, such as expense, receptionist and billing, and waiting time.

6. Conclusions and future work

The experiments conducted proves an excellent performance in Targeted Aspect Based Sentiment Analysis. Compared with the previous works, our model can achieve better performance with an average F1-Score of 88% for sentiment classification. The performance of aspect detection obtained an F1-Score of 86%, which exceeds the performance of baseline models. Without external knowledge, our model could also perform well on English and Indonesian Dataset. Bidirectional GRU could classify sentiments of each aspect well in an

imbalanced dataset with polarity weight. Adding polarity weights to the model influences the accuracy and the ability of the model to learn characters. Moreover, the use of the character-enhanced token embedding approach improves the sentiment classification performance and helps resolve spelling errors that occur in sentences. Multi-level attention is also working well, where target-level attention is focused on the target while sentence-level attention presents a relevant representation of the text.

For future research, a comparison could be made for different word embedding methods such as FastText, BERT, or Elmo.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, Mauridhi H. Purnomo, Suryo Sumpeno, and Kimiya Fujisawa; methodology, Esther I. Setiawan and Joan Santoso; software, Ferry; validation, Joan Santoso; formal analysis, Esther I. Setiawan, Suryo Sumpeno and Mauridhi H. Purnomo; writing—original draft preparation, Esther I. Setiawan, Joan Santoso and Ferry; writing—review and editing, Mauridhi H. Purnomo and Surya Sumpeno; visualization, Joan Santoso and Surya Sumpeno; supervision, Surya Sumpeno, Kimiya Fujisawa and Mauridhi H. Purnomo; project administration, Esther I. Setiawan; funding acquisition, Esther I. Setiawan.

Acknowledgments

The research for this paper was financially supported by the Ministry of Research, Technology, and Higher Education of Indonesia and the Ministry of Finance, as well as Indonesia Endowment Fund for Education (LPDP) through BUDI-DN scholarship scheme.

References

- [1] E. Cambria, "Affective computing and sentiment analysis," *IEEE Intell. Syst.*, Vol. 31, No. 2, pp. 102–107, 2016.
- [2] M. H. Purnomo, Y. Kristian, E. Setyati, U. D. Rosiani, and E. I. Setiawan, "Limitless possibilities of pervasive biomedical engineering: Directing the implementation of affective computing on automatic health monitoring system", In: *Proc. of 2016 8th International Conf. on Information Technology and Electrical Engineering (ICITEE)*, Yogyakarta, Indonesia, pp. 1–4, 2016.

- [3] E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *A Practical Guide to Sentiment Analysis*, Springer, Switzerland, 2017.
- [4] F. Nurifan, R. Sarno, and K. R. Sungkono, “Aspect Based Sentiment Analysis for Restaurant Reviews Using Hybrid ELMO-Wikipedia and Hybrid Expanded Opinion Lexicon-SentiCircle”, *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 6, pp. 47–58, 2019.
- [5] O. A. M. Ghaleb and A. S. Vijendran, “An Enhancement of the Public Sentiment Analysis on Social Networking by Improving Sentiment Analysis Tools”, *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 3, pp. 68–78, 2018.
- [6] S. Bharathi, A. Geetha, and R. Sathiyarayanan, “Sentiment analysis of twitter and RSS news feeds and its impact on stock market prediction”, *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 6, pp. 68–77, 2017.
- [7] S. Bharathi and A. Geetha, “Sentiment analysis for effective stock market prediction”, *International Journal of Intelligent Engineering and Systems*, Vol. 10, No. 3, pp. 146–153, 2017.
- [8] R. A. Priyantina and R. Sarno, “Sentiment Analysis of Hotel Reviews Using Latent Dirichlet Allocation, Semantic Similarity and LSTM”, *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 4, pp. 142–155, 2019.
- [9] D. A. K. Khotimah and R. Sarno, “Sentiment Analysis of Hotel Aspect Using Probabilistic Latent Semantic Analysis, Word Embedding and LSTM”, *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 4, pp. 275–290, 2019.
- [10] Y. Ma, H. Peng, T. Khan, E. Cambria, and A. Hussain, “Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis”, *Cognit. Comput.*, Vol. 10, No. 4, pp. 639–650, 2018.
- [11] J. S. Lee, D. Zuba, and Y. Pang, “Sentiment Analysis of Chinese Product Reviews using Gated Recurrent Unit”, In: *Proc. of 2019 IEEE Fifth International Conf. on Big Data Computing Service and Applications (BigDataService)*, California, United States, pp. 173–181, 2019.
- [12] M. Al-Smadi, B. Talafha, M. Al-Ayyoub, and Y. Jararweh, “Using long short-term memory deep neural networks for aspect-based sentiment analysis of Arabic reviews”, *International Journal of Machine Learning and Cybernetics International Journal of Machine Learning and Cybernetics*, Vol. 10, No. 8, pp. 2163–2175, 2019.
- [13] M. Nissim and V. Patti, “Semantic aspects in sentiment analysis”, in *Sentiment Analysis in Social Networks*, Elsevier, United States, pp. 31–48, 2017.
- [14] J. Zhou, Y. Lu, H.-N. Dai, H. Wang, and H. Xiao, “Sentiment analysis of Chinese microblog based on stacked bidirectional LSTM”, *IEEE Access*, Vol. 7, pp. 38856–38866, 2019.
- [15] Y. Miao, Y. Ji, and E. Peng, “Application of CNN-BiGRU Model in Chinese Short Text Sentiment Analysis”, In: *Proc. of the 2019 2nd International Conf. on Algorithms, Computing and Artificial Intelligence*, Sanya, China, pp. 510–514, 2019.
- [16] Y. Wang, M. Huang, X. Zhu, and L. Zhao, “Attention-based LSTM for aspect-level sentiment classification”, In: *Proc. of the 2016 Conf. on Empirical Methods in Natural Language Processing*, Texas, USA, pp. 606–615, 2016.
- [17] C. Yu, K. S. Barsim, Q. Kong, and B. Yang, “Multi-level attention model for weakly supervised audio classification,” *arXiv Prepr. arXiv1803.02353*, 2018.
- [18] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation”, In: *Proc. of the 2014 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532–1543, 2014.
- [19] S. Vosoughi, P. Vijayaraghavan, and D. Roy, “Tweet2vec: Learning tweet embeddings using character-level cnn-lstm encoder-decoder”, In: *Proc. of the 39th International ACM SIGIR Conf. on Research and Development in Information Retrieval, Tuscany, Italy*, pp. 1041–1044, 2016.
- [20] M. Saeidi, G. Bouchard, M. Liakata, and S. Riedel, “Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods”, *arXiv Prepr. arXiv1610.03771*, 2016.
- [21] F. Deroncourt, J. Y. Lee, O. Uzuner, and P. Szolovits, “De-identification of patient notes with recurrent neural networks”, *Journal of the American Medical Informatics Association Journal of the American Medical Informatics Association*, Vol. 24, No. 3, pp. 596–606, 2017.
- [22] T. T. Thet, J.-C. Na, and C. S. G. Khoo, “Aspect-based sentiment analysis of movie reviews on discussion boards”, *Journal of Information Science*, Vol. 36, No. 6, pp. 823–848, 2010.
- [23] C. Brun, D. N. Popa, and C. Roux, “XRCE: Hybrid Classification for Aspect-based Sentiment Analysis”, In: *SemEval@ COLING*,

- Dublin, Ireland, pp. 838–842, 2014.
- [24] W. Zhang, H. Xu, and W. Wan, “Weakness Finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis”, *Expert System with Appl.*, Vol. 39, No. 11, pp. 10283–10291, 2012.
- [25] T. H. Nguyen and K. Shirai, “PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis”, In: *Proc. of 2015 Conf. Empir. Methods Nat. Lang. Process.*, Lisbon, Portugal, pp. 2509–2514, 2015.
- [26] W. Xue and T. Li, “Aspect based sentiment analysis with gated convolutional networks”, In: *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, Australia, pp. 2514–2523, 2018.
- [27] S. Ruder, P. Ghaffari, and J. G. Breslin, “A hierarchical model of reviews for aspect-based sentiment analysis”, *arXiv Prepr. arXiv1609.02745*, 2016.
- [28] Y. Ma, H. Peng, and E. Cambria, “Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM”, In: *Proc. of Thirty-second AAAI conf. on artificial Intelligence*, New Orleans, United State, pp. 5876–5883, 2018.
- [29] J. Santoso, E. I. Setiawan, E. M. Yuniarno, M. Hariadi, and M. H. Purnomo, “Hybrid Conditional Random Fields and K-Means for Named Entity Recognition on Indonesian News Documents,” *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 233–245, 2020.
- [30] J. Santoso, Gunawan, H. V Gani, E. M. Yuniarno, M. Hariadi, and M. H. Purnomo, “Noun phrases extraction using shallow parsing with C4.5 decision tree algorithm for Indonesian Language ontology building”, In: *Proc. of 2015 15th International Symposium on Communications and Information Technologies (ISCIT)*, Nara, Japan, pp. 149–152, 2015.
- [31] S. Byrne, “Aspect Identification and Sentiment Analysis in Text-Based Reviews,” Lehigh University, United States, 2017.
- [32] E. Cambria, J. Fu, F. Bisio, and S. Poria, “AffectiveSpace 2: Enabling affective intuition for concept-level sentiment analysis”, In: *Proc. of Twenty-ninth AAAI Conf. on Artificial Intelligence*, Texas, United States, 2015.
- [33] S. Gu, L. Zhang, Y. Hou, and Y. Song, “A position-aware bidirectional attention network for aspect-level sentiment analysis”, In: *Proc. of the 27th International Conference on Computational Linguistics*, New Mexico, USA, pp. 774–784, 2018.
- [34] Y. Han, M. Liu, and W. Jing, “Aspect-Level Drug Reviews Sentiment Analysis Based on Double BiGRU and Knowledge Transfer”, *IEEE Access*, Vol. 8, pp. 21314–21325, 2020.
- [35] D. Ma, S. Li, and H. Wang, “Joint learning for targeted sentiment analysis”, In: *Proc. of the 2018 Conf. on Empirical Methods in Natural Language Processing*, Brussels, Belgium, 2018, pp. 4737–4742.
- [36] F. Liu, T. Cohn, and T. Baldwin, “Recurrent entity networks with delayed memory update for targeted aspect-based sentiment analysis”, *arXiv Prepr. arXiv1804.11019*, 2018.
- [37] A. Firmanto and R. Sarno, “Aspect-Based Sentiment Analysis Using Grammatical Rules, Word Similarity and SentiCircle”, *Int. J. Intell. Eng. Syst.*, Vol. 12, No. 5, pp. 190–201, 2019.
- [38] C. D. Santos and B. Zadrozny, “Learning character-level representations for part-of-speech tagging”, In: *Proc. of the 31st International Conf. on Machine Learning (ICML-14)*, Beijing, China, pp. 1818–1826, 2014.
- [39] C. N. dos Santos and V. Guimaraes, “Boosting named entity recognition with neural character embeddings,” *arXiv Prepr. arXiv1505.05008*, 2015.
- [40] W. Ling, I. Trancoso, C. Dyer, and A. W. Black, “Character-based neural machine translation,” *arXiv Prepr. arXiv1511.04586*, 2015.
- [41] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, “Character-aware neural language models”, In: *Proc. of Thirtieth AAAI Conf. on Artificial Intelligence*, Arizona, United States, pp. 2741–2750, 2016.
- [42] C. Dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts”, In: *Proc. of COLING 2014, the 25th International Conf. on Computational Linguistics: Technical Papers*, Dublin, Ireland, pp. 69–78, 2014.
- [43] J. Wehrmann, W. Becker, H. E. L. Cagnini, and R. C. Barros, “A character-based convolutional neural network for language-agnostic Twitter sentiment analysis”, In: *Proc. of 2017 International Joint Conf. on Neural Networks (IJCNN)*, Alaska, United State, pp. 2384–2391, 2017.
- [44] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, “Semeval-2015 task 12: Aspect based sentiment analysis”, In: *Proc. of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 486–495, 2015.