Two-Stage Fine-Grained Text-Level Sentiment Analysis Based on Syntactic Rule Matching and Deep Semantic

Weizhi LIAO\(^{(a)}\), Yaheng MA\(^{(b)}\), Members, Yiling CAO\(^{(c)}\), Guanglei YE\(^{(d)}\), and Dongzhou ZUO\(^{(c)}\), Nonmembers

SUMMARY Aiming at the problem that traditional text-level sentiment analysis methods usually ignore the emotional tendency corresponding to the object or attribute. In this paper, a novel two-stage fine-grained text-level sentiment analysis model based on syntactic rule matching and deep semantics is proposed. Based on analyzing the characteristics and difficulties of fine-grained sentiment analysis, a two-stage fine-grained sentiment analysis algorithm framework is constructed. In the first stage, the objects and its corresponding opinions are extracted based on syntactic rules matching to obtain preliminary objects and opinions. The second stage based on deep semantic network to extract more accurate objects and opinions. Aiming at the problem that the extraction result contains multiple objects and opinions to be matched, an object-opinion matching algorithm based on deep semantic network is used to accurately extract the objects and opinions. Finally, the proposed algorithm is evaluated on several public datasets to demonstrate its practicality and effectiveness.

key words: text-level sentiment analysis, fine-grained sentiment elements, syntactic rule matching, deep semantics, lexical separation distance

1. Introduction

Text-level sentiment analysis is the process of analyzing and extracting the emotions of given texts [1], [2]. The technique is applicable in numerous situations including natural language processing, data mining, and information retrieval and extraction [3]–[5]. Recently, sentiment analysis has also extended to the fields of consumption decision-making, public opinion analysis, and personalized recommendation, with high commercial value [6], [7]. Fine-grained methods have become the focus of text-level sentiment analysis due to increasing analytical needs and the diverse and complex language environment [8]–[10]. The main difficulties of fine-grained sentiment analysis are listed as follows:

- Obtaining clear and specific fine-grained emotional elements in text data;
- Identifying and extracting the specific objects, attributes, and emotional tendencies mentioned in the text;
- Matching the extracted object and opinion accurately.

Numerous studies have explored fine-grained sentiment analysis. For example, [2] proposed a double propagation unsupervised learning method to extract objects and opinions. In [11], the extraction of emotional elements were treated as a sequence labeling problem and the Conditional Random Field (CRF) model was applied to fuse the diverse features. Alternatively, a sorting algorithm was used in [12] to extract emotional objects and opinions in text. Such methods are based on named entity recognition, which can identify and extract specific objects and attributes, as well as other entity information. However, such techniques fail to obtain the corresponding emotional tendency.

In [13], regular and unconventional opinion words were identified according to the co-occurrence features of subjects. In [14], an association set between opinions and feature categories was constructed to infer the attributes, where the relevance between attributes and opinions was obtained through co-occurrence information. Word-level and sentence-level emotions were classified and combined with the given subject in [15], while [16] applied the similarity evaluation method to determine the subject according to the composition and location features of noun phrases. [9] improved the performance of end-to-end extraction of opinion targets and corresponding sentiment tendencies in user comments by improving the basic sequence annotation model. [17] combined recursive neural networks and CRF, learning high-level discriminative features and double propagates information between aspect and opinion terms. [18] proposed a MIL method that can predict the emotional polarity of consecutive segments with the help of the label of the document. [19] proposed an end-to-end deep learning model that jointly extracts attributes and opinion words. Above emotion analysis methods are based on automatic keywords extraction, which can extract the subject and keywords in text, but fail to match the extracted objects and opinions.

To solve the above problems, a two-stage fine-grained text-level sentiment extraction method based on syntactic rule matching and deep semantics is proposed. In the first stage, based on the syntactic rules matching, the objects and its corresponding opinions are initially extracted. In the second stage, on the basis of the first stage, the deep semantic network is used to accurately extract the objects and opinions. Aiming at the problem that traditional methods are difficult to match objects and opinions, a novel matching algorithm based on minimum lexical separation distance is proposed. The shortcomings of both named entity recognition and automatic keyword extraction are overcome, effectively capturing the corresponding emotional tendency of objects and accurately matching the opinions and objects.

Copyright © 2021 The Institute of Electronics, Information and Communication Engineers
2. Framework of Two-Stage Fine-Grained Emotional Element Extraction

The principal concept of the proposed framework is to decompose emotion analysis into two stages of emotional element extraction tasks. The first stage is to extract the coarse-grained emotional elements. Based on the results, the second stage works to extract increasingly fine-grained emotional elements and obtain more detailed evaluation objects, as well as their corresponding emotional tendency. In this way, the new method can achieve accurate matching between the extracted objects and opinions.

The first stage is based on syntactic rule matching. The model is trained in an unsupervised manner on unlabeled text data to obtain preliminary evaluation objects and their corresponding opinions, denoted as the labeled knowledge data for the second stage. The utilized syntactic rules include both the Dependency parsing (DP) and subject-verb relationship. According to the DP analysis, the dependent syntactic components of each word are generated. Then, based on the subject-verb relationship, the coarse-grained emotional elements are obtained. In the second stage, the deep semantic extraction algorithm is used to accurately extract the emotional elements, producing more accurate evaluation objects and their corresponding opinions. Based on the minimum lexical separation distance matching algorithm, the obtained objects and opinions can then be accurately matched. The principle framework of two-stage fine-grained emotional element extraction algorithm is summarized in Fig. 1.

3. Algorithm of Two-Stage Fine-Grained Emotional Element Extraction

3.1 First Stage: Extraction Algorithm Based on Dependency Parsing and Subject-Verb Relationship

From the analysis of the part of speech (POS) and syntactic components, evaluation objects contained in the text are usually nouns located in the subject part of the sentence. Opinions are usually adjectives located in the predicate part of the sentence. By extracting the subject and the predicate in the text, the object and its corresponding opinion can be obtained. The first stage algorithm is designed accordingly in Fig. 2. \( \text{Word}_1 \text{Word}_2 \text{Word}_3 \) represents the word sequence after word segmentation, \( \text{POS}_1 \text{POS}_2 \ldots \text{POS}_k \) and \( \text{DP}_1 \text{DP}_2 \ldots \text{DP}_k \) respectively represent the corresponding sequence of part-of-speech (POS) and dependent parsing components, and \( (\text{SB}, \text{V})_1, (\text{SB}, \text{V})_2, \ldots, (\text{SB}, \text{V})_m \) are the extracted emotional elements. DP analysis contains a total of 24 DP relations, of which SBV is the subject-predicate.

In this paper, a multiple deterministic DP analysis method based on the move-reduction algorithm is used, which includes three kinds of analysis actions: Shift, Right-Reduce, and Left-Reduce. Each loop of the analysis process traverses the entire sentence from left to right, establishing a dependent subtree through the Right-Reduce and Left-Reduce actions. Shift is used to advance the analysis window until all the words in the sentence are hung under the same node, which is the root node of the entire dependency tree. According to the representation of the analysis action in [20], the detailed move-reduction algorithm is provided in Fig. 3.

In the figure, \( \text{W} \) is the window of the target node with the basic width of two, indicating that the target of each decision is the two adjacent words in the current sequence. Additionally, \( \text{A} \) is the obtained set of dependency relations and \( \text{i} \) is the word position in current sentence. The same word may be in different positions in multiple loop.
When the window focuses on a certain word pair, the analyzer extracts the contextual features which express the current analysis state and they are sent to the pre-trained classifier model for predicting the analysis action in the current step. The classifier then performs the predicted action to expand the dependency set A and advance the analysis window. The classifier repeats this analysis process on the training sentences with dependent structure to obtain the vector pair of action-feature.

In Fig. 3, the content on the top of the horizontal line indicates the window of the current node, and the one on the bottom of the same line represents the result after performing the corresponding analysis action.

For example, analyze the sentence ‘The master in the store is very careful, but the waiting time is too long.’ Two subject-predicate relations should be denoted as ‘SBV’, which corresponds to the word combinations of ‘master’ and ‘careful’, as well as ‘time’ and ‘long’. Once the two combinations have been extracted, the evaluation objects and corresponding opinions in the sentence can be obtained.

3.2 Second Stage: Extraction Matching Algorithm Based on Deep Semantics and Minimum Lexical Separation Distance

The emotional element extraction algorithm based on deep semantics is composed of deep recurrent neural networks and semantic feature fusion, which is used to extract more accurate evaluation objects and opinions. Commonly deep neural networks include Elman RNN, LSTM, and GRU [21], [22], all of which contain unique advantages for text-level emotional element extraction [22], [23]. The matching algorithm of minimum lexical separation distance is then used to match the extracted objects and opinions. The algorithm schematic diagram is shown in Fig. 4. The proposed emotional element extraction algorithm is composed of word sequence, the vectorization representation and semantic feature layer, the feature coding layer for evaluation text, the fully connected layer, and the final output layer. Here, \( w_1w_2...w_n \) indicates the evaluation sequence and \( e_1e_2...e_n \) is the vectorized representation sequence of evaluation text based on word2vec. Additionally, \( o_1o_2...o_3 \in (B1, I1, O) \) and \( s_1s_2...s_3 \in (B2, I2, O) \) represent the sequences of evaluation action and opinion respectively. POS and DP indicate the POS features and DP features, respectively. FC denotes the fully connected layer, and \( h \) represents the feature coding layer for evaluation text, which are shared to all time steps. By using matching algorithm based on minimum lexical separation distance, the evaluation objects and opinions are extracted and matched. Supposing that \( S_1,S_2,...,S_p \) and \( O_1,O_2,...,O_q \) are the extracted opinions and objects and \( E = \{(S'_1,O'_1),(S'_2,O'_2),..., (S'_K,O'_K)\} \) is their corresponding set, the purpose of matching algorithm is to match \( S_1,S_2,...,S_p \) and \( O_1,O_2,...,O_q \) correctly and produce \( E \).

1) Emotional element extraction algorithm based on deep semantics: As all input vectors share the same neurons in the deep recurrent neural network, the evaluation text-related features at each time step share the same \( h \) and FC layer.

In this paper, Elman RNN, LSTM, and GRU are used as the deep semantic network in the second stage algorithm. The following is a detailed explanation of their training process on the emotional element extraction task.

The word sequence is first converted with one-hot encoding into a sequence of word embedding vectors, so that the encoded word vector contains more semantic information. The word embedding representation can be obtained by Eq. (1), where \( E \) represents the word vector matrix and \( w_i \) is the one-hot code of the \( i \)-th word.

\[
e_i = EW_s, \tag{1}
\]

The role of Elman RNN, LSTM, and GRU is to fuse the word embedding vector, POS features, and DP features for obtaining semantic vectors containing rich contextual information. Here, \( y = g(x) \) is used to represent the internal operating mechanisms of Elman RNN, LSTM, and GRU. The process of obtaining semantic vectors can be denoted according to Eq. (2).

\[
h_i = g(W_e e_i + W_{POS} e_{POS} + W_{DP} e_{DP} + Uh_{i-1}), \tag{2}
\]

here, \( W_e, W_{POS}, W_{DP}, \) and \( U \) are training parameters, \( e_{POS} \) and \( e_{DP} \) are the POS features and DP feature vectors, respectively, \( e_i \) is the obtained word embedding representation by Eq. (1), and \( h_{i-1} \) is the output of the previous time step.

To introduce nonlinear operations, sigmoid is applied as the active function as follows:

\[
y_i = sigmoid(h_i), \tag{3}
\]
At the end of the deep recurrent neuron network, a fully connected layer is added to realize the transformation from the temporal information to the emotional elements. Then, the evaluation object $o_i$ and opinion expression $s_i$ in the text can be obtained.

$$s_i = \frac{\exp(w_i^T h_a)}{\sum_{k=1}^{K} \exp(w_k^T h_a)},$$

(4)

$$o_i = \frac{\exp(w_i^T h_b)}{\sum_{k=1}^{K} \exp(w_k^T h_b)},$$

(5)

2) Matching algorithm for minimum lexical separation distance: For the problem of matching multiple pairs of evaluation objects and opinions in the second stage, a matching algorithm architecture for minimum lexical separation distance is proposed, which is trained in an unsupervised manner without extra annotation data. Let $S_1, S_2, \ldots, S_p$ and $O_1, O_2, \ldots, O_q$ be the extracted opinions and evaluation objects. As the opinion and object in the text are unpaired, their numbers, $p$ and $q$, are not equal here. The pair number $K$ is the smaller one of $p$ and $q$ (denoted as $K = \min(p, q)$), while $E = \{(S'_1, O'_1), (S'_2, O'_2), \ldots, (S'_K, O'_K)\}$ is the pair set of evaluation objects and opinions. The purpose of the algorithm is to correctly match $S_1, S_2, \ldots, S_p$ and $O_1, O_2, \ldots, O_q$, and produce $E$, as shown in Fig. 5. The number of words between $w_i$ and $w_p$ are defined as their lexical separation distance, in which each punctuation mark takes up the space of two words. If $w_i = w_p$, lexical separation distance is denoted as 0. The algorithm flow chart is provided in Fig. 6. For the opinion expression $S_1, S_2, \ldots, S_p$ and evaluation object $O_1, O_2, \ldots, O_q$, the matching algorithm for minimum lexical separation distance first defines an empty set $E$, which is used to store the matched object-opinion pairs. Then, by comparing the sizes of $p$ and $q$, the model decides whether to traverse the opinion sequence or object sequence for generating the final output $E$. The specific algorithm flow is a matching algorithm which requires non-labeling data and solves the matching problem of multiple object-opinion pairs. After this process it is possible to obtain the object-opinion pair with high matching accuracy rate, which is the emotional element. Algorithm flow is shown as follows:

Algorithm 1 Matching algorithm for minimum lexical separation distance

**Input:** Objects $O_1, O_2, \ldots, O_q$ and opinions $S_1, S_2, \ldots, S_p$

**Output:** Set $E = \{(S'_1, O'_1), (S'_2, O'_2), \ldots, (S'_K, O'_K)\}$, $K = \min(p, q)$

1: Determine if there are multiple objects and opinions: if $p = q = 1$, return $(S_1, O_1)$

2: Define an empty set $E$ and compare $p$ and $q$. If $p < q$, carry out Step 3. Otherwise, proceed to Step 4; 3: After traversing all opinions, proceed to Step 5. Otherwise, for the $i$-th opinion $S_i$, find the object $O_i$ with the smallest distance and combine the two as $(S_i, O_i)$. Then, add the combination to the set $E$;

3: After traversing all objects, proceed to Step 5. Otherwise, for the $i$-th object $O_i$, find the opinion $S_i$ with the smallest distance and combine them as $(S_i, O_i)$. Add the combination to the set $E$;

4: Output set $E$, including the object-opinion pairs.

4. Experimental Verification

Before the experiment, the text data and the main work was preprocessed using text segmentation and text vectorization. In this paper, word segmentation and word2vec [24] was used to perform the two tasks, respectively. The following section includes the introduction of datasets, algorithm evaluation metrics, and finally, experimental results and analysis.

4.1 Datasets and Evaluation Metrics

The experimental datasets consisting of public data and actual data are shown in Table 1. The public data is from SemEval-2014 Task 4, including laptop evaluation data (SemEval-2014 Laptop) and restaurant evaluation data.
Table 1  Detailed summary of public dataset and automobile evaluation dataset

| Dataset               | Train Samples | Test Sample | Total Sample |
|-----------------------|---------------|-------------|--------------|
| SemEval-2014 Laptop   | 3,045         | 800         | 3,845        |
| SemEval-2014 Restaurant| 3,041        | 800         | 3,841        |
| Automobile Evaluation | 4,030         | 2,000       | 6,030        |

Table 2  Accuracy, recall, and f1 value on different length text and the number of samples dataset

| Length of Text    | P  | R  | F1  | Sample Number |
|-------------------|----|----|-----|---------------|
| No more than 7    | 52.79 | 53.60 | 53.19 | 2,764         |
| No more than 9    | 53.74 | 53.46 | 53.09 | 3,570         |
| No more than 11   | 51.41 | 47.89 | 49.59 | 3,743         |
| No more than 15   | 46.61 | 44.84 | 45.70 | 4,327         |
| Original data     | 45.58 | 41.43 | 43.41 | 6,030         |

Fig. 7  Accuracy rate, recall rate, F1 value of different length text

(SemEval-2014 Restaurant). The actual text data is comprised of approximately 6,030 evaluation pieces regarding a certain automobile 4S shop.

Precision P, recall R, and F1 value were first applied as the evaluation metrics, and are usually used in classification and sequence labeling tasks.

4.2 Experimental Results and Analysis

The proposed algorithm was evaluated to extract the object-opinion pair in word segmentations with the length of no more than 7, 9, and 15 words, respectively. The comparison results are provided in Table 2.

According to the results in Table 2, the change of experimental results under different lengths is illustrated in Fig. 7. As shown, the accuracy, recall, and F1 values decrease with the increase of word length. As the length of the text is longer, its lexical, syntactic, and grammatical structure is more complex, leading to a decline in the accuracy of DP analysis. When the segmented text includes no more than 7 words, the recall rate and F1 value are the best. The highest accuracy is obtained on the text with no more than 9 words. As the number of the sample with no more than 7 words is too small, the text with no more than 9 words was selected to generate the object-opinion pair by using the extraction algorithm in the first stage.

Statistics concerning the obtained results via deep semantic extraction algorithm in the second stage are provided in Table 3. The SemEval-2014 Laptop was selected as the training data, and the accuracy, recall, and F1 value were employed as evaluation metrics. Elman RNN, LSTM, and GRU were also applied to distill additional features including DP information and POS labeling information.

Figure 8 shows the statistical changes of experimental results according to Table 3. It can be clearly observed that the accuracies and recall rates of GRU and LSTM are higher than that of Elman RNN. By providing the DP information for three networks, the recall rate increases significantly. However, the improvement of accuracy rate is not obvious, and even decreases in Elman RNN and LSTM models. Similar results can be concluded when providing the POS labeling information and using both DP and POS information does not improve results further. The possible reason for this is that a variety of lexical and syntactic features generate a high level noise when introducing rich language information. Based on the comprehensive experimental results and above analysis, the GRU network with dependent parsing features is determined to perform the best on both datasets.

The proposed algorithm was further evaluated on
Table 4  Effect of different recurrent neuron networks when adding diverse features on the semeval-2014 restaurant dataset

| Model       | P        | R        | F1       |
|-------------|----------|----------|----------|
| Elman-RNN   | 80.58    | 75.48    | 77.95    |
| +DP         | 80.32    | 76.23    | 79.29    |
| +POS        | 80.35    | 76.84    | 78.56    |
| +DP+POS     | 80.27    | 75.82    | 78.02    |
| LSTM        | 82.70    | 77.34    | 79.93    |
| +DP         | 81.96    | 78.46    | 80.17    |
| +POS        | 81.02    | 77.45    | 79.19    |
| +DP+POS     | 81.15    | 76.35    | 78.68    |
| GRU         | 81.80    | 77.52    | 79.60    |
| +DP         | 81.66    | 78.52    | 80.26    |
| +POS        | 82.80    | 78.90    | 80.80    |
| +DP+POS     | 82.87    | 77.65    | 78.08    |

Table 5  Effect of different recurrent neuron networks when adding diverse features on the automotive evaluation dataset

| Model       | P        | R        | F1       |
|-------------|----------|----------|----------|
| Elman-RNN   | 74.05    | 69.54    | 71.72    |
| +DP         | 75.70    | 70.40    | 72.92    |
| +POS        | 74.45    | 70.34    | 72.34    |
| +DP+POS     | 75.39    | 70.62    | 72.93    |
| LSTM        | 76.17    | 72.23    | 74.15    |
| +DP         | 77.68    | 72.00    | 74.73    |
| +POS        | 75.70    | 71.80    | 73.70    |
| +DP+POS     | 76.17    | 72.88    | 74.49    |
| GRU         | 76.30    | 72.91    | 74.57    |
| +DP         | 77.37    | 73.70    | 75.49    |
| +POS        | 76.60    | 72.52    | 74.50    |
| +DP+POS     | 76.91    | 72.43    | 74.60    |

Fig. 9  Results of different types of deep recurrent neuron networks with different features on the semeval-2014 restaurant dataset.

Fig. 10  Results of different types of deep recurrent neuron networks with different features on the evaluation data of automobile evaluation.

Chinese data set are 5% to 6% lower than that of the English dataset (SemEval-2014 ResLaptop and Restaurant). For the Chinese set, it was necessary to segment the input first, causing an error that influenced the accuracy of the next stage.

Another possible reason for the decline is due to the data not being structured in the same manner as the public sets. On the actual data, LSTM network with DP information and GRU network with the POS information perform better. However, the improvement of the three networks with POS labeling information is not obvious. Based on the above analysis and experimental results, the GRU network combined with the DP information was selected to perform the deep semantic extraction algorithm in the second stage.

5. Conclusion

Aiming at the problem that text sentiment data can not form tagged knowledge, a two-stage emotional element extraction method based on syntactic matching regulation and deep semantics was proposed, using unsupervised syntactic rule matching to obtain tagged knowledge for the second stage fine-grained sentiment analysis extraction. Aiming at the problem that the extraction result contains multiple evaluation objects and opinions to be matched, an object-opinion matching algorithm based on the minimum lexical separation distance was proposed to achieve accurate pairwise matching. Finally, the effectiveness of the algorithm proposed in this paper is verified through experiments.

Acknowledgments

This work was supported by the National Key R & D Program of China (2017YFB1400301 & 2017YFB1400300).

References

[1] L. Bing, “Sentiment analysis and opinion mining,” Synthesis Lectures on Human Language Technologies, vol.28, no.4, pp.325–344, 2011.
[2] G. Qiu, B. Liu, J. Bu, and C. Chen, “Opinion word expansion and target extraction through double propagation,” Computational Linguistics, vol.37, no.1, pp.9–27, 2011.
[3] N. Kaji and M. Kitsuregawa, “Building lexicon for sentiment analysis from massive collection of html documents,” EMNLP-CoNLL,
[4] H. Kanayama and T. Nasukawa, “Fully automatic lexicon expansion for domain-oriented sentiment analysis,” EMNLP, vol.28, no.4, pp.355–363, 2006.

[5] J.B. Guang Qiu, B. Liu, and C. Chen, “Expanding domain sentiment lexicon through double propagation,” Twenty-First International Joint Conference on Artificial Intelligence, 2009.

[6] H. Shi, W. Zhan, and X. Li, “A supervised fine-grained sentiment analysis system for online reviews,” Intelligent Automation & Soft Computing, vol.21, no.4, pp.589–605, 2015.

[7] Z. Wang, C.S. Chong, L. Lan, Y. Yang, S.B. Ho, and J.C. Tong, “Fine-grained sentiment analysis of social media with emotion sensing,” Future Technologies Conference, pp.1361–1364, 2016.

[8] R.K. Amplayo and M. Song, “An adaptable fine-grained sentiment analysis for summarization of multiple short online reviews,” Data Knowl. Eng., vol.110, pp.54–67, 2017.

[9] G. Chang and H. Huo, “A method of fine-grained short text sentiment analysis based on machine learning,” Neural Network World, vol.28, no.4, pp.325–344, 2018.

[10] Q. Sun, J. Niu, Z. Yao, and H. Yan, “Exploring ewom in online customer reviews: sentiment analysis at a fine-grained level,” Engineering Applications of Artificial Intelligence, vol.81, no.MAY, pp.68–78, 2019.

[11] E.R. Yejin Choi, C. Cardie, and S. Patwardhan, “Identifying sources of opinions with conditional random fields and extraction patterns,” Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, vol.28, no.4, pp.355–362, 2005.

[12] J. O’Neil, “Entity sentiment extraction using text ranking,” SIGIR, p.1042, 2012.

[13] M. Hu and B. Liu, “Mining and summarizing customer reviews,” pp.375–382, 2004.

[14] Q. Su, X. Xu, H. Guo, Z. Guo, X. Wu, X. Zhang, B. Swen, and Z. Su, “Hidden sentiment association in chinese web opinion mining,” Proc. 17th international conference on World Wide Web - WWW '08, pp.959–968, 2008.

[15] S.M. Kim and E. Hovy, “Automatic detection of opinion bearing words and sentences,” Posters/Demos, vol.28, no.4, pp.1433–1443, 2015.

[16] J.H. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, “Sentiment analyzer: extracting sentiments about a given topic using natural language processing techniques,” Third IEEE International Conference on Data Mining, ICDM 2003, pp.427–434, 2003.

[17] W. Wang, S.J. Pan, D. Dahlmeier, and X. Xiao, “Recursive neural conditional random fields for aspect-based sentiment analysis,” Proc. 2016 Conference on Empirical Methods in Natural Language Processing, pp.616–626, 2016.

[18] S. Angelidis and M. Lapata, “Multiple instance learning networks for fine-grained sentiment analysis,” Transactions of the Association for Computational Linguistics, vol.6, pp.17–31, 2018.

[19] D. Dahlmeier and X. Xiao, “Coupled multi-layer attentions for co-extraction of aspect and opinion terms,” AAAI, vol.6, pp.3316–3322, 2017.

[20] J. Nivre, “An efficient algorithm for projective dependency parsing,” Proceedings of the Eighth International Conference on Parsing Technologies, pp.149–160, 2003.

[21] K.C. Chung, C. Gulcehre, and Y. Ben-gio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” vol.28, no.4, pp.365–372, 2014.

[22] P. Liu, S. Joty, and H. Meng, “Fine-grained opinion mining with recurrent neural networks and word embeddings,” Proc. 2015 Conference on Empirical Methods in Natural Language Processing, pp.1433–1443, 2015.

[23] T. Young, D. Hazarika, S. Poria, and E. Cambria, “Recent trends in deep learning based natural language processing,” 2017.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” Computer Science, 2013.