Research on Multi-head Self-attention Aspect Term Extraction with Multi-level Features Encoding

Zhang Penghui[1,3(✉)] and Yang Peng[1,2,3]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 210000, China
{zhangpenghui,pengyang}@seu.edu.cn
[2] School of Computer Science and Engineering, Southeast University, Nanjing 211189, China
[3] Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 211189, China

Abstract. With the development of social media and e-commerce, aspect-based sentiment analysis technique is more and more urgently needed. Aspect term extraction is an important part of aspect-based sentiment analysis, and the quality of the extracted aspect words directly determines the quality of aspect-based sentiment analysis. To solve the problems of current research, in this paper, we design and implement a multi-head self-attention aspect term extraction fusing with multi-level features model (MSA-FMF). In addition, we conduct experiments on three public datasets, and the proposed model outperforms the comparison models, proving the effectiveness of our model.

Keywords: Aspect-Based Sentiment Analysis · Aspect Term Extraction · Msa-Fmf

1 Introduction

In recent years, with the development of social media and e-commerce, aspect-based sentiment analysis techniques have become particularly important. The aspect term extraction (ATE) task is a sub-task of aspect-based sentiment analysis, and the quality of the extracted aspect words directly affects the results of sentiment analysis. As shown in Fig. 1, aspect term extraction aims to extract aspect words in comment sentences to provide support for the aspect-based sentiment classification task. Researchers have done in-depth research on aspect term extraction, and the research methods can be roughly divided into unsupervised-based methods and supervised-based methods.

In the supervised-based research method, Li et al. [4] used the words in the text as nodes to construct a graph, voted and ranked each node in the graph, and used the ranking result as the basis for aspect word extraction. Vorontsov et al. [10] captured the relationship between aspect words and text words based on semantic information, and used this method to obtain the semantic relationship between words and review texts for aspect term extraction. Mehri et al. [7] carried out statistics and weight calculation on

K. Subramanian et al. (Eds.): CTMCD 2022, ACSR 99, pp. 380–389, 2023.
https://doi.org/10.2991/978-94-6463-046-6_45

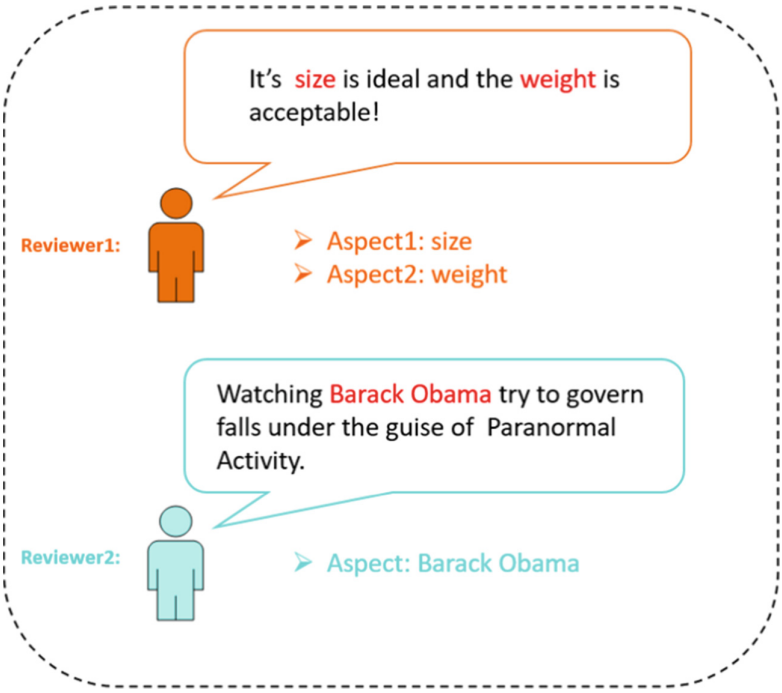

Fig. 1. Aspect Term Extraction Data Samples

the words in the comments, and used the words with large weight calculation results as aspect words. Supervised-based methods require manual labeling of data, and manual labeling of data brings cost issues that unsupervised-based methods can avoid. The latent Dirichlet distribution proposed by Blei et al. [9] was a classical model based on unsupervised methods to solve the problem of aspect term extraction. It regards the obtained probability distribution as the topic distribution and takes it as the basis for aspect word extraction. Yin et al. [12] used a model that can automatically learn the features of aspect words, combined with the dependencies between words, to complete the extraction of aspect words with the same meaning but different syntactic functions.

At present, the problem of aspect term extraction task mainly focuses on two aspects:

Firstly, semantic association features and character-level features between words in a sentence are ignored, resulting in incomplete aspect words finally extracted by the model. Besides, the association between the aspect word in the sentence and the contextual information in the sentence is not fully learned, resulting in the wrong extraction of the aspect word.

Aiming at the above two problems in the current research, this paper introduces the method of integrating multi-level features encoding, and combines the multi-head self-attention mechanism to complete the extraction of aspect words in sentences. The specific research ideas are as follows:

(1) Introduce the pre-training model Roberta, which uses word, position and segment multi-level features embedding information as the input of the model, and combines the

encoding results of character embedding information to use four different granularity features as the final representation of the sentence.

(2) Introduce a multi-head self-attention mechanism downstream of the model to fully learn the correlation information between the aspect words in the sentence and the contextual information, so as to alleviate the model extraction errors caused by the mismatch between the aspect words and the contextual information.

2 Methods

This section mainly introduces the multi-head self-attention aspect term extraction fusing with multi-level features model that proposed in this study. Moreover, the architecture of MSA-FMF is shown in Fig. 2. It includes an encoding layer that incorporates multi-level features, a context encoding layer based on a bidirectional long short-term memory network, a global context information extraction layer based on a multi-head self-attention mechanism, and a sequence decoding layer based on conditional random fields. In this study, the aspect word extraction task is transformed into a sequence labeling task, and the data is annotated using the IOB [8] labeling method. "B-ASP" represents the first word of the aspect word, "I-ASP" represents the non-first word of the aspect word, and "O" represents the non-aspect word, the labeled data is shown in Fig. 3.

2.1 Task Definition

Suppose the word sequence of the sentence input to the model is $w^s = \{w_1, w_2, ..., w_m\}$, where m represents the number of words in the sentence. The aspect word sequence is $asp = \{asp_1, asp_2, ..., asp_t\}$, and t is the number of aspect words in the sentence. Wherein, the i-th aspect word can be expressed as $asp_i = \{w_1^i, w_2^i, ..., w_{tl}^i\}$, and tl represents the length of the word sequence included in the i-th aspect word. In short, the goal of aspect term extraction is to accurately extract the tl aspect words from the sentence sequence w^s.

2.2 Multi-level Features Encoding Layer

(1) Encoding of words, positions and segments.
The encoding layer that fuses multi-level features mainly fuses embedding information of four granularities of characters, words, positions and segments, encodes these embedding information, and extracts the preliminary features of sentences.

After the sentence is entered into the model, Roberta will add "[CLS]" and "[SEP]" to the beginning and end of the sentence respectively. Assume that the sequence of words in the input model sentence after Roberta adds the token at the start and end positions is $w^s = \{w_1, w_2, ..., w_m\}$, where m denotes the length of the sequence. First, the embedding of words, positions and segments is carried out. The feature embedding information of these three granularities can be expressed as:

$$E_w = E_T + E_P + E_S \tag{1}$$

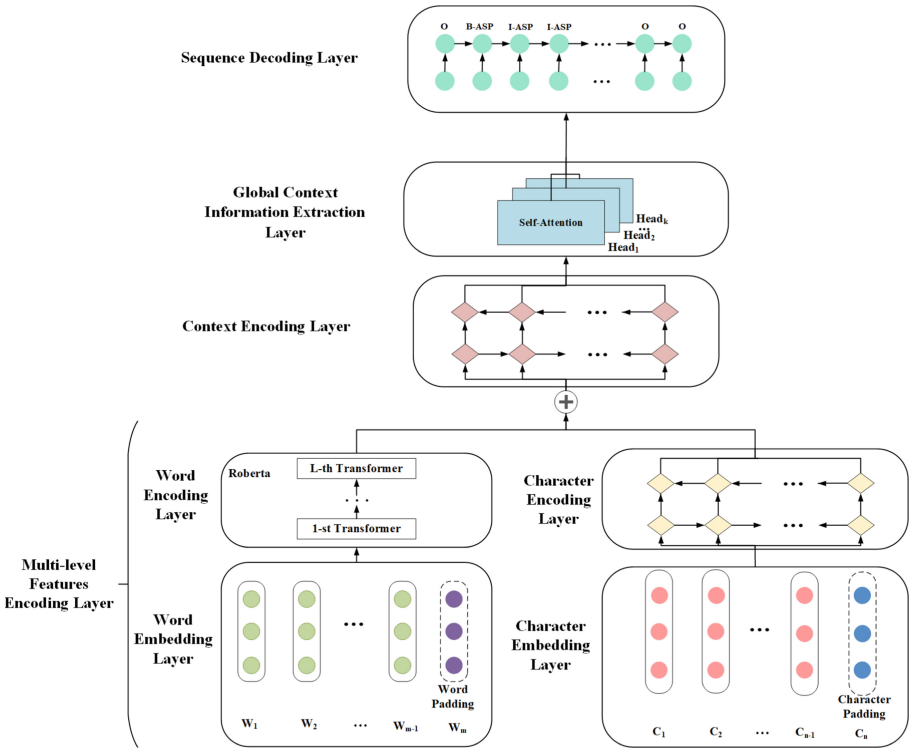

Fig. 2. The architecture of MSA-FMF.

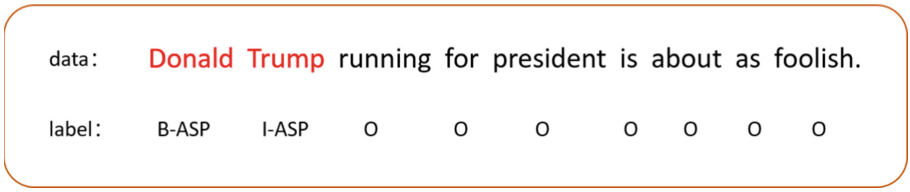

Fig. 3. Aspect term extraction data labeling example.

In the above formula, E_T represents the token embedding, E_P represents the positional embedding of the word in the sequence, E_S represents the segment embedding, which is used to distinguish different clauses of a complete sentence, and $E_w = \{e_1, e_2, ..., e_m\}$ represents the final embedding result.

After that, the multi-layer Transformer will encode the result of the embedding. The initial input is $H_0 = E_w$, and the encoding result of the sentence is obtained through the encoding of the L-layer Transformer. This calculation process can be expressed as:

$$H_i = \text{Transformer}(H_{i-1}),\ i \in [1,L] \tag{2}$$

Among them, H_i represents the output of the i-th layer Transformer, $H_L = \{h_1^L, h_2^L, ..., h_m^L\}$ represents the final output of Roberta.

(2) Encoding of character.

Suppose the filled character sequence is $C = \{c_1, c_2, ..., c_n\}$, and n represents the number of characters. Assuming that Emb_c is the character embedding matrix, the character embedding process can be expressed as:

$$E_c = Emb_c \cdot C \tag{3}$$

In the above formula, E_c is the embedding matrix of characters. In the character encoding stage, using the bidirectional long short-term memory network as the character encoder, the encoding process can be expressed as:

$$H_C = \overrightarrow{lstm}(E_c) \oplus \overleftarrow{lstm}(E_c) \tag{4}$$

$\overrightarrow{h_c}$ represents the forward hidden state output of the bidirectional LSTM, $\overleftarrow{h_c}$ represents the backward hidden state output of the bidirectional LSTM, H_C represents the final output of the bidirectional LSTM, and $\oplus$ represents the concatenation operation of the vector.

The encoding vector that fuses four different granular features of word, position, segment and character can be expressed as:

$$H_{CW} = H_L \oplus H_C \tag{5}$$

$H_{CW} = \{h_1^{cw}, h_2^{cw}, \ldots, h_m^{cw}\}$ represents a vector representation that fuses four features of word, position, segment and character.

2.3 Context Encoding Layer

After the coding layer of multi-level features is fused, a vector representation of four different granular features is obtained. In the context coding layer based on bidirectional long short-term memory network, the context of the sentence is coded, and the context information is integrated into the vector representation. The context encoding process based on long short-term memory network can be expressed as:

$$H_{ctx} = \overrightarrow{lstm}(H_{CW}) \oplus \overleftarrow{lstm}(H_{CW}) \tag{6}$$

$\overrightarrow{lstm}(H_{CW})$ represents the output of the bidirectional LSTM forward hidden layer, $\overleftarrow{lstm}(H_{CW})$ represents the output of the bidirectional LSTM backward hidden layer, $H_{ctx} = \{h_1^{ctx}, h_2^{ctx}, ..., h_m^{ctx}\}$ represents the final output of the bidirectional LSTM, and $\oplus$ represents the connection operation of the vector.

2.4 Global Context Information Extraction Layer

Aiming at the difficulty in matching aspect words and contextual information, a multi-head self-attention mechanism is used in the downstream of the model to learn the

contextual information of the context. This layer is composed of K self-attention mechanism layers. Each self-attention layer performs self-attention calculation. After connecting the calculation results of the K self-attention layers, the output of the linear layer is H_{att}, which is also based on multi-head self-attention. The calculation method of the self-attention mechanism can be described as, the first stage transforms the input vector through a linear layer:

$$H_{ctx}^{lin} = W_0^{lin} H_{ctx} + b_0^{lin} \tag{7}$$

H_{ctx}^{lin} is the eigenvector calculated by the linear layer for the input, W_0^{lin} and b_0^{lin} are the weight matrix and the bias value respectively.

In the second stage, the eigenvectors are multiplied by the three weight matrices W^Q, W^K and W^V respectively to obtain q_i, k_j and v_j. The calculation process is as follows:

$$q_i = W^Q h_i^{lin} \tag{8}$$

$$k_j = W^K h_j^{lin} \tag{9}$$

$$v_i = W^V h_i^{lin} \tag{10}$$

In the third stage, multiply the transposition q_i^T of q_i and k_j to get the attention score, then divide the attention score by $\sqrt{d_k}$, and finally get the weight matrix w_{ij} after normalization by the softmax function. The calculation process is as follows:

$$w_{ij} = soft\max(\frac{q_i^T \cdot k_j}{\sqrt{d_k}}) \tag{11}$$

In the fourth stage, multiply v_i by the weight w_{ij}, and then accumulate the output vector h_i^k of the self-attention layer. The calculation process is as follows:

$$h_i^k = \sum_{j=1}^{m} v_i \odot w_{ij} \tag{12}$$

The vector connection process of the multi-head self-attention mechanism can be expressed as:

$$H_{att}^{'} = concat(head_1^{att}, head_2^{att}, ..., head_K^{att}) \tag{13}$$

$head_k^{att}$ represents the output of the k-th self-attention head. In the fifth stage, the final output of the multi-head attention mechanism will be obtained through the operation of the linear layer. The calculation process is as follows:

$$H_{att} = W_1^{lin} H_{att}^{'} + b_1^{lin} \tag{14}$$

H_{att} is the final output of this layer, W_1^{lin} is the weight matrix, b_1^{lin} is the bias.

2.5 Sequence Decoding Layer

In the last layer of the model, the conditional random field is used as the sequence decoder, and the extraction of aspect words in the sentence is completed by sequence annotation. Assuming that the input of the sequence decoding layer is $X = \{x_1, x_2, ..., x_m\}$ and the label sequence is $Y = \{y_1, y_2, ..., y_m\}$, the calculation process of prediction can be expressed as:

$$s(X, Y) = \sum_{i=1}^{m} A_{y_i, y_{i+1}} + \sum_{i=1}^{m} H_{i, y_{i+1}} \tag{15}$$

$$P(Y|X) = \text{soft}\max(s(X, Y)) \tag{16}$$

$s(X, Y)$ represents the score of label prediction, and A represents a randomly initialized matrix to represent the correlation between adjacent labels y_i and y_{i+1}. H represents the output of the previous layer, and $H_{i,y_{i+1}}$ represents the score of the i-th label. $P(Y|X)$ represents the conditional probability of X appearing under the condition of Y, and softmax is the activation function. Finally, the Viterbi algorithm is used to calculate the label sequence with the highest score and use it as the final prediction result $\widehat{Y}$. The calculation process is as follows:

$$\widehat{Y} = \arg\max_{Y} s(X, Y) \tag{17}$$

The loss function of the model is:

$$\text{loss} = -\sum_{X,Y} \ln p(Y|X) \tag{18}$$

3 Test Results and Discussions

3.1 Datasets and Experimental Settings

In this study, a total of three datasets are used for experiments, namely the ACL14 Twitter dataset [1], the SemEval2014 Restaurant dataset [6] and the SemEval2014 Laptop dataset [6], all of which are labeled with IOB annotation method. During the training process, the training set and the validation set are divided according to 8:2. The statistics of these three data sets are shown in Table 1.

Table 1. The data distribution of the aspect word extraction experimental dataset.

Dataset	Training	Validation	Test
Twitter	4997	1250	691
Restaurant	2881	721	1119
Laptop	1861	466	691

The experiment uses Macro-F1 as evaluation matric, and its calculation method is as follows:

$$P = \frac{TP}{TP + FP} \tag{19}$$

$$R = \frac{TP}{TP + FN} \tag{20}$$

$$F_1 = \frac{2 * P * R}{P + R} \tag{21}$$

In the above formula, P represents the precision rate, R represents the recall rate, F_1 represents $Macro - F1$ score, TP represents true positive, FP represents false positive, FN represents false negative.

During the experiment, some experimental parameters are set, the type of Roberta is Roberta-base, the number of layers of Transformer is 12, and the dimension of the vector obtained by Roberta is 768. The character embedding uses a random initialization method, the dimension of the character embedding is 100, the maximum length of the characters in the word is set to 20, the number of LSTM hidden units in the character encoding layer is 100, and the number of LSTM hidden units in the context encoding layer is 300. In the global information extraction layer based on the multi-head self-attention mechanism, the number of attention heads is set to 4. The batch size of training is set to 8, the learning rate is set to 3e-5, and the number of training rounds is 20. After each round of training, it is verified on the validation set and reserved the best performing model. In addition, the Adam optimizer is used to optimize the network during training, and dropout is used to prevent the model from overfitting, and the value of dropout is set to 0.5.

3.2 Main Results and Analysis

In this study, the MSA-FMF model was tested on three data sets respectively. The results obtained by the model MSA-FMF model proposed in this study and the model of the comparison group are shown in Table 2. The experimental results prove that the effectiveness of the model.

Table 2 records the comparison results of the model proposed in this study with the other groups model on the Laptop and Restaurant datasets. The data show that the model proposed in this study achieved the best results on both datasets. On the Laptop dataset, MSA-FMF improves the Macro-F1 by 0.59 percentage points compared to the best performing model (BAT) in the comparison group. On the Restaurant dataset, MSA-FMF improves the Macro-F1 by 2.59 percentage points compared to the best performing model in the comparison group (BiDTreeCRF).

Table 3 records the experimental results of the proposed model MSA-FMF and the three models in the comparison group on Twitter. The evaluation indicators include precision rate, recall rate and F1 value. The experimental results show that compared with the LSTM-CRF model, the model in this study improves the accuracy, recall and F1 value by 5.20, 4.18 and 4.69 percentage points, respectively; compared with Bi-LSTM, it improves by 3.55, 3.08 and 3.31 percentage points. Compared with the BERT-PT model, the improvements are 2.84, 1.84, and 2.35 percentage points, respectively.

Table 2. Comparison of experimental results between MSA-FMF model and comparison models.

Models	Laptop	Restaurant
LSTM-CRF [2]	73.43	81.80
BiLSTM-CRF [2]	76.10	82.38
BiLSTM-CNN [5]	78.97	83.87
BERT-PT [11]	84.26	—
BAT [3]	85.57	—
MSA-FMF	**86.16**	**87.90**

Table 3. Experimental results of MSA-FMF model on Twitter dataset.

Models	Twitter		
	P(%)	R(%)	F1
LSTM-CRF	93.03	94.50	93.76
BiLSTM-CRF	94.68	95.60	95.14
BERT-PT	95.39	96.84	96.10
MSA-FMF	**98.23**	**98.68**	**98.45**

4 Conclusion

The current research on aspect word extraction suffers from the problems that character-level features are ignored and the association between aspect words and contextual information in sentences is not sufficiently learned. In this paper, a novel model to tackle these problems by fusing multi-level feature encoding and multi-head attention mechanism, and propose the MSA-FMF model. In order to verify the effectiveness of the model proposed in this paper, we selected several groups of comparative models, and carried out experimental verification on a public dataset. The experimental results prove the superiority of our proposed model.

Acknowledgments. This work was supported in part by the Consulting Project of Chinese Academy of Engineering under Grant 2020-XY-5, 2018-XY-07, and in part by the Fundamental Research Funds for the Central Universities and the Academy-Locality Cooperation Project of Chinese Academy of Engineering under Grant JS2021ZT05.

References

1. Dong L, Wei F, Tan C, et al. Adaptive recursive neural network for target-dependent twitter sentiment classification[C]//Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 2: Short papers). 2014: 49–54.

2. Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv 2015[J]. arXiv preprint arXiv:1508.01991, 2015.
3. Karimi A, Rossi L, Prati A. Adversarial training for aspect-based sentiment analysis with BERT[C]//2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021: 8797–8803.
4. Li W, Zhao J. TextRank algorithm by exploiting Wikipedia for short text keywords extraction[C]//2016 3rd International Conference on Information Science and Control Engineering (ICISCE). IEEE, 2016: 683–686.
5. Ma X, Hovy E. End-to-end sequence labeling via bi-directional lstm-cnns-crf[J]. arXiv preprint arXiv:1603.01354, 2016.
6. Manandhar S. Semeval-2014 task 4: Aspect based sentiment analysis[C]//Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014). 2014.
7. Mehri A, Jamaati M, Mehri H. Word ranking in a single document by Jensen–Shannon divergence[J]. Physics Letters A, 2015, 379(28-29):1627-1632.
8. Sang E F, Veenstra J. Representing text chunks[J]. arXiv preprint cs/9907006, 1999.
9. Sommeria-Klein G, Zinger L, Coissac E, et al. Latent Dirichlet Allocation reveals spatial and taxonomic structure in a DNA-based census of soil biodiversity from a tropical forest[J]. Molecular Ecology Resources, 2020, 20(2): 371-386.
10. Vorontsov K, Potapenko A, Plavin A. Additive regularization of topic models for topic selection and sparse factorization[C]//International Symposium on Statistical Learning and Data Sciences. Springer, Cham, 2015: 193-202.
11. Xu H, Liu B, Shu L, et al. BERT post-training for review reading comprehension and aspect-based sentiment analysis[J]. arXiv preprint arXiv:1904.02232, 2019.
12. Yin Y, Wei F, Dong L, et al. Unsupervised word and dependency path embeddings for aspect term extraction[J]. arXiv preprint arXiv:1605.07843, 2016.