Cognitive Computation (2024) 16:158–176
https://doi.org/10.1007/s12559-023-10191-y

O2-Bert: Two-Stage Target-Based Sentiment Analysis

Yan Yan[1] · Bo-Wen Zhang[2] · Guanwen Ding[1] · Wenjie Li[3] · Jie Zhang[1] · Jia-Jing Li[1] · Wenchao Gao[1]

Received: 7 July 2022 / Accepted: 12 August 2023 / Published online: 1 September 2023

Abstract
Target-based sentiment analysis (TBSA) is one of the most important NLP research topics for widespread applications. However, the task is challenging, especially when the targets contain multiple words or do not exist in the sequences. Conventional approaches cannot accurately extract the (target, sentiment) pairs due to the limitations of the fixed end-to-end architecture design. In this paper, we propose a framework named **O^2-Bert**, which consists of **O**pinion target extraction (OTE-Bert) and **O**pinion sentiment classification (OSC-Bert) to complete the task in two stages. More specifically, we divide the OTE-Bert into three modules. First, an entity number prediction module predicts the number of entities in a sequence, even in an extreme situation where no entities are contained. Afterwards, with predicted number of entities, an entity starting annotation module is responsible for predicting their starting positions. Finally, an entity length prediction module predicts the lengths of these entities, and thus, accomplishes target extraction. In OSC-Bert, the sentiment polarities of extracted targets from OTE-Bert. According to the characteristics of BERT encoders, our framework can be adapted to short English sequences without domain limitations. For other languages, our approach might work through altering the tokenization. Experimental results on the SemEval 2014-16 benchmarks show that the proposed model achieves competitive performances on both domains (restaurants and laptops) and both tasks (target extraction and sentiment classification), with F1-score as evaluated metrics. Specifically, OTE-Bert achieves 84.63%, 89.20%, 83.16%, and 86.88% F1 scores for target extraction, while OSC-Bert achieves 82.90%, 80.73%, 76.94%, and 83.58% F1 scores for sentiment classification, on the chosen benchmarks. The statistics validate the effectiveness and robustness of our approach and the new "two-stage paradigm". In future work, we will explore more possibilities of the new paradigm on other NLP tasks.

Keywords O^2-Bert · OTE-Bert · OSC-Bert · Entity number prediction · Entity starting annotation · Entity length prediction

Introduction

Target-based sentiment analysis (TBSA), has raised a growing concern among natural language processing researchers due to applications in various scenarios. The purpose is to extract opinion targets with corresponding sentiment polarities from sentences, usually user-generated contents, such as product reviews and tweets.

TBSA is a challenging task since in most cases, the sentences are not as simple as "the apple tastes good". Below we show some special examples in Table 1. The examples ① to ③ show that the targets which correspond to the opinion words do not exist in sentences, known as "null targets", illustrated as "T: NULL". In ② to ③, as well as ⑤ to ⑦, there are multiple targets in a sentence, illustrated as "T: Multiple". In ④ and ⑤, the targets in the sentences are made up with several words, such as *"lava cake dessert"* and *"French onion soup"*, illustrated as "T: Lengthy". Moreover in ⑥ and ⑦, there are several targets existing in a sentence, which correspond to different sentiment polarities. Even worse, there are opposite opinions corresponding to a same target, like "rolls" in ⑥. In the last example ⑧, the sentiment is implicit since there are no opinion words and the sentiment can be inferred from the contextual information. Therefore,

Bo-Wen Zhang contributed equally to the first author.

✉ Yan Yan
yanyanustb@126.com

[1] Department of Computer Science and Technology, School of Mechanical Electronic and Information Engineering, China University of Mining and Technology Beijing, Beijing, China

[2] Beijing Academy of Artificial Intelligence, Beijing, China

[3] Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong

Table 1 Eight typical examples of TBSA from SemEval datasets. (colored for better comprehension), T: NULL for the existence of "NULL" targets, T: Multiple for multiple targets within a sentence, T: Lengthy for the targets represented in several words, S: Inconsistent for the different polarities, S: Implicit for the polarity inferred from context rather than polarity words

Examples	Ground Truth	T: NULL	T: Multiple	T: Lengthy	S: Inconsistent	S: Implicit
①It looked like shredded cheese partly done-still in strips	(NULL, pos)	✓				
②Slightly on the **pricy side** but worth it!	(NULL, pos) (NULL, , neg)	✓	✓			
③The ambience is pretty and nice for conversation, so a casual lunch here would probably be best	(NULL, pos) (NULL, pos)	✓	✓			
④The lava cake dessert was incredible and I recommend it.	(lava cake desserts, pos)			✓		
⑤The food was average to above-average; the French Onion soup filling yet not overly impressive, and the desserts not brilliant in any way.	(food, pos) (French Onion soup, pos) (French Onion soup, neg) (desserts, neg)		✓	✓	✓	
⑥those rolls were big, but not good and sashimi wasn't fresh.	(rolls, neu) (rolls,neg) (sashimi, neg)		✓		✓	
⑦I like the somosas, chai, and the chole, but the dhosas and dhal were kinda dissapointing.	(somosas, pos) (chai, pos) (chole, pos) (dhosas, neg) (dhal, neg)		✓		✓	
⑧ Get the tuna of gari.	(tuna of gari, pos)					✓

an increasing number of scientific research organizations, such as the International Workshop on Semantic Evaluation (SemEval), have focused on TBSA, providing benchmarks to organize TBSA challenges.

Conventionally, there are two groups of mainstream research works to solve the TBSA task: the one-stage approach and the two-stage approach. The two-stage approach usually divides the TBSA task into two subtasks: opinion target extraction (OTE) and opinion sentiment classification (OSC). OTE extracts the opinion targets from the review sentences, in which there could be one or more, or none aspect terms and each target might contain one or multiple words. OSC classifies the sentiment which corresponds to each opinion target. The one-stage method is an end-to-end approach which simultaneously completes the two subtasks through a model.

One-stage methods are mainly based on multi-task learning (MTL) [1], which often shares weights between subtasks and co-trains the subtasks with a joint loss. However, some problems remain in MTL methods which cannot be modeled clearly, such as difficulties in the design of loss functions and setting up learning rate during training process.

For most two-stage methods, OTE is often regarded as a particular sequence labeling task, resulting in a general trend of using conditional random fields (CRFs) [2]. Recent works in OTE include span-based models like SpanMlt, question-answer based models like ASTE, and post-processing models like DE-CNN. OSC is usually based on part-of-speech or single classifiers, such as a multilayer perceptron (MLP) [3] or a support vector machine (SVM). Recent works in OTE include capsule network models like IACapsNet, GCN models like GP-GCN, and contextual attention network models like CGAT. However, there are still some remaining problems. For instance, the challenge to capture the semantic information of an opinion target when it is a phrase, or classify the sentiment despite the long distance between opinion words and targets.

To solve the above issues, some researchers proposed deep networks with attention mechanism to encode the sentence and capture more accurate embeddings, which can help CRF achieve high accuracy on OTE tasks. Additionally, the attention mechanism are also effective in OSC tasks.

The main contributions of this work are summarized as follows:

- In this paper, we propose a two-stage framework for target-based sentiment analysis, namely O^2-Bert, which consists of two modules, respectively for Opinion Target Extraction (OTE-Bert) and Opinion Sentiment Classification (OSC-Bert). Compared to the end-to-end approaches, the proposed framework can make full use of the supervision signals and achieve better-trained models.
- Secondly, the designed standalone modules, respective for entity number prediction, starting position annotation, as well as entity length prediction, are effective to solve the unusual samples, for example, samples with no entities, or with multiple-word entities. Moreover, in the entity starting position module, we introduce an innovative model to combine BERT and GCN to learn contextual relationships among words.
- The proposed approach achieves competitive performances on open benchmarks, SemEval datasets, which demonstrates the effectiveness and robustness through various comparison experiments.

Related Work

We summarize the main matrices employed by models in the "Introduction" and "Related Work" sections to facilitate improved analysis and comparison in Table 2.

Table 2 Overview of matrices and models in "Introduction" and "Related Work"

Task	Method			
	Preprocessing	Network	Attention mechanism	Other
OTE	Word2Vec [4]	RNN [5]	Self [6]	MTL [1]
	BERT [4, 7, 8]	CNN [9–11]	Other [1, 12]	CRF [2, 13, 5]
	GPT [14]	BiGRU [6, 12]		SPR [15]
	Other [10, 16]	Capsule network [12]		LDA [17, 13]
		LSTM [18, 19]		Post-processing [9]
				Span-based [20–22, 18]
				RL [23]
OSC	BERT [24–27]	LSTM [28, 19, 29]	Multi-head [24, 25]	MTL [1]
		Capsule network [30, 29]	Other [1, 31, 32]	MLP [3, 33]
				Dictionary-based [31]
				GCN [24, 32, 25, 26, 34, 35]
				Distance-rule [29]
				Dependency-rule [36, 37]
One-stage	MTL [1]			
	BERT [38]			
	Span-based [18]			

Opinion Target Extraction

In prior research, the conventional approach for the OTE task involved manually selecting features to build a model using frequency, rule-based, or machine learning techniques. Rana and Cheah [15] proposed TF-RBM, a two-layer rule-based model that leverages sequential patterns from customer reviews to define rules. It enhances opinion target extraction accuracy by integrating frequency- and similarity-based approaches. While rule-based methods are effective, their design poses challenges. Creating rules necessitates specialized linguistic expertise and knowledge, and capturing all rules for natural languages proves exceedingly difficult. Shams and Baraani-Dastjerdi [17] designed a Latent Dirichlet Allocation (LDA) topic model that leverages co-occurrence relations as prior domain knowledge to uncover more precise aspects. Jochim and Deleris [13] proposed a CRF-based technique for extracting named entities from medical literature. The study explored the impact of constraints to enhance accuracy. While the assumptions of CRF are generally well-defined, they may not always align with the problem at hand. CRF models face challenges in scalability and fail to effectively capture contextual information over long distances. As a result, achieving high performance in scenarios involving extensive data content can be challenging.

In recent years, researchers have primarily focused on integrating deep learning methods with CRF models to automatically extract opinion targets, as deep learning models excel at extracting high-level features. Guo et al. [5] employed a combination of CRF and recurrent neural network (RNN) to automatically extract opinion targets and opinion words, eliminating the need for manual feature selection. Wang et al. [9] addressed the issue of opinion target number and boundary errors in sequence tagger extraction by proposing a post-processing method. They aimed to control the number of extracted opinion targets and correct their boundaries. On the other hand, Lu and Liu [6] employed two bidirectional gated recurrent unit (BiGRU) networks to extract semantic features and incorporated a self-attention mechanism to capture global dependencies. Su et al. [12] proposed the XLNetCN model as a solution for the TBSA task, incorporating a capsule network with a dynamic routing algorithm. This approach effectively captures local and spatial hierarchical relations within the text sequence, leveraging its proficiency in multilabel text classifications. Pour and Jalili [10] proposed an innovative data preprocessing technique and a deep convolutional neural network that accurately classifies each word in a given sentence as either an aspect or non-aspect word.

Furthermore, researchers have also explored enhancing information encoding alongside feature extraction optimization. Zhang et al. [4] and Bravo-Marquez utilized BERT and Word2Vec in their embedding layer, while Li et al. [16] leveraged the position information of the opinion target during sentence encoding, resulting in improved performance. Kang et al. [7] proposed RABERT, a targeted opinion word extraction method, which encodes target information into BERT by incorporating target markers within the sentence. Additionally, a target-sentence relation network is integrated into RABERT to account for neighboring words. Moreover, various span-based models have been introduced. To

handle cases where a token belongs to multiple entities, Gao et al. [20] utilizes a span-based tagging scheme as opposed to traditional sequence labeling models that can assign only one label per token. Hu et al. [21] proposed a span-based framework that identifies opinion targets based on their span boundaries and determines their polarities using span representations. They treated the task of extracting opinion targets and their sentiment polarities as a sequence tagging problem, addressing challenges related to the extensive search space and sentiment inconsistencies. Xu et al. [22] proposed a span-level model to extract opinion targets comprising multiple words, leveraging interactions between the spans of opinion targets and sentiment words. They also devised a dual-channel span pruning strategy that combines supervision from aspect term extraction and opinion term extraction tasks. In contrast, Yu Bai Jian et al. [23] proposed a novel paradigm called ASTE-RL, which first extracts sentiment words and then identifies their opinion targets, considering the mutual interactions among aspect terms, associated sentiments, and opinion terms.

Opinion Sentiment Classification

For OSC, the most straightforward approach is based on a dictionary, which is dependent on sentimental words marked in the dictionary. Xu et al. [31] designed a dictionary-based method, whose dictionary can update the words newly joined and the compound words using machine learning algorithms. Dictionary-based approaches require human effort to gather opinion words. They necessitate manual removal or correction of words containing errors, and they lack the ability to tailor the thesaurus to a specific application scenario or context. Subsequently, machine learning techniques such as Support Vector Machine (SVM), Multilayer Perceptron (MLP), and others were employed for OSC tasks. Almaghrabi and Chetty [33] developed a Multilayer Perceptron model for sentiment analysis in Arabic and English languages. However, a major drawback of the aforementioned approaches was the manual selection and tuning of rules or features.

In recent research, attention mechanisms have emerged as a prominent method for deep learning. Attention mechanisms enable the learning of varying levels of importance for sequential words, enhancing information capture. Wang et al. [28] previously utilized the attention mechanism in conjunction with a long short-term memory (LSTM) network to acquire aspect embeddings and perform aspect-level sentiment classification. Similarly, Du incorporated SenticNet into an LSTM network, considering user expression patterns through the application of attention mechanisms. Although LSTM is adept at sequence modeling, it encounters challenges such as information loss over long distances and the inability to effectively capture constraints, such as the typical association of adjectives with nouns, thereby limiting its performance. Moreover, researchers have increasingly explored the integration of attention mechanisms with other deep learning techniques, yielding improved outcomes. Du et al. [30] developed a model that primarily leverages a capsule network and attention mechanism, employing a bidirectional recurrent neural network (BiRNN) to extract features. Li et al. [24] proposed a model incorporating a hierarchical multi-head attention mechanism and a graph convolutional network to effectively consider both syntactic dependencies and the relationship between opinion targets and their context. Miao et al. [32] proposed a contextual graph attention network that consists of two graph attention networks and a contextual attention network to capture aspect-sensitive text features and proposed a novel syntactic relative distance-based syntactic attention mechanism for enhanced attention towards opinion targets while reducing computational complexity. In contrast, Wei et al. [25] developed GP-GCN, which simplifies global features by addressing potential noise introduced by contextual information. They employed orthogonal projection and graph convolutional networks to reduce inter-node and node-global feature dependencies.

Recent Advances Promoted by Pretrained Language Models

Tiwari et al. [39] conducted a comparative analysis of various BERT-based approaches for solving the ABSA task, including fine-tuned BERT models, adversarial training with BERT, and the integration of disentangled attention in BERT or DeBERTa. Chouikh et al. [38] focused on contrasting different versions of BERT specifically designed for the Arabic language in their model. Zhu et al. [40] developed the BERT-pair-ABSA model, which employed semantic expansion of auxiliary sentences and sentiment polarity calculation to gain insights into netizens' changing concerns, emotional states, and evolving trends across different stages. Gutierrez et al. [14] examined the few-shot performance of GPT-3 in-context learning compared to fine-tuning smaller PLMs (similar to BERT) on two significant biomedical information extraction tasks: named entity recognition and relation extraction.

O^2-Bert Model

The O^2-Bert model utilizes BERT as the encoder and employs attention pooling to predict entity number. It leverages GCN for annotating entity starts, CRF for entity prediction, and incorporates attention mechanisms and capsule networks to determine entity polarities.

The O^2-Bert model separates the TBSA task into two stages. The first stage is opinion target extraction (OTE), which consists of three modules. These modules are respectively in charge of predicting the number of entities, annotating the start position of each entity, and predicting the lengths of the entities. The second stage is opinion sentiment classification (OSC), which aims to classify the emotional tendency of the entity in the sentence. Figure 1 shows the architecture of the proposed model. In this section we firstly introduce the difficulties in the task, and then describe the methodologies in detail.

Example 1:
"The food here is rather good, but only if you like to wait for it."
Target = "Food" Polarity = "positive"
Target = "NULL" Polarity = "negative"

Example 1 A case of "NULL target" problem, with an "invisible" entity — waiting time

Challenges and Design Motivation

Typically, conventional approaches for named entity recognition (NER) in the context of opinion target extraction (OTE) rely on BIO-based or span-based models. However, as previously mentioned, a significant challenge is addressing the issue of "NULL" targets, where the target entity does not exist within the sentence, as illustrated in Example 1. In this example, the sentiment polarities of the target "food taste" and "waiting time" are respectively positive and negative. However, the mention of "waiting time" is absent in the sentence. Both BIO-based and span-based NER approaches fail to address this issue. To overcome this challenge, we introduce a dedicated entity number prediction module to enhance entity annotation. By utilizing this module, we can determine the count of entities, referred to as the "prediction number". Alternatively, the number of annotated entity start positions, known as the "annotation number", can also be used to infer the entity count. By comparing the difference between these two numbers, we can identify the presence of "NULL entities". In this specific example, two entities are predicted, while only one entity start is annotated, indicating the presence of one "NULL entity". Furthermore, in our experiments, the occurrence of the prediction count being lower than the annotation count is rare, primarily due to the training set's distribution.

Given the three potential scenarios for the difference between the two numbers, it is feasible that the "prediction number" may be lower than the "annotation number." In such cases, we can rely on the "prediction number" and generate the top N (equivalent to the "prediction number") probable entities from the entity annotation. The entity number prediction module typically employs a classification model, utilizing the encoding information from the BERT model as input, coupled with a dense layer. Additionally, to address the imbalanced data distribution, the focal loss is employed.

Another challenge is to process the multiple entities as well as entities with multiple words. Traditional approaches, usually struggle to determine whether there is a multiple-word entity or several single-word entities. In Example 2, the number of entities cannot be determined. Moreover, in Example 3, there are nested entities. As a consequence, we propose to separate the entity annotation into two modules, in which, an entity starting annotation module marks the start positions of entities, while an entity length prediction module predicts the lengths of entities based on their start positions. In Example 2, *"prix"* is marked as the start position and the length of entity is predicted as 3, hence the annotated entity is *"prix fixe menu"*. Similarly, in Example 3, the annotated results are not influenced by nested entities.

Example 2:
"The Prix Fixe menu is worth every penny and you get more than enough."
Target = "Prix Fixe menu"
Assuming the annotation result: "0" is the non-entity "1" is the entity
The Prix Fixe menu is worth every penny and you get more than enough .
0 1 1 1 0 0 0 0 0 0 0 0 0 0

Example 2 A case of "multiple-word entity" problem, with the entity — Prix Fixe menu

Example 3:

"The only thing I moderately enjoyed was their Grilled Chicken special with Edamame Puree."

Possible target:

① Target = "their Grilled Chicken special with Edamame Puree"

② Target = "Edamame Puree"

③ Target = "Grilled Chicken special"

This system: "0" is the non-entity "1" is the beginning of entity

Step 1: Annotate the beginning of entity

The only thing I moderately enjoyed was their Grilled Chicken special with Edamame Puree.

0 0 0 0 0 0 0 1 1 0 0 0 1 0

Step 2: The length of entities prediction model to predict length

After inputting the sample and the starting position of entity (sorted by probability):

① The length of the entity begin with "their" is **7**

② The length of the entity begin with "Edamame" is **2**

③ The length of the entity begin with "Grilled" is **3**

Example 3 A case of "nested entity" problem, with the entities "their Grilled Chicken special with Edamame Puree", "Edamame Puree", and "Grilled Chicken special"

OTE-Bert Framework

Pretrained language models show great abilities to capture contextual semantics. In this work, BERT [41] is utilized as the encoders. Specifically, we use pretrained Roberta-base [42][1] as the initial checkpoint and perform continue-pretraining on domain corpus. Each input sentence is firstly processed into a sequence like "[CLS] $token_1$, $token_2$, ..., $token_n$, [SEP]". Afterwards, these tokens are mapped to token embeddings, as well as segment embeddings and position embeddings, according to the basic usage of BERT. These embeddings are fed into the BERT transformer encoders to formulate the contextual embeddings of the tokens. The three modules share the contextual embeddings to make full use of pretraining. The following is the detailed descriptions of the three modules.

Entity Number Prediction Module

This module predicts the number of entities from the input sentences and the structure is shown in Fig. 1.

This module employs an attention pooling [8] layer to reduce the dimensionality of contextual embeddings and determine the weights assigned to each token. Attention pooling utilizes a dense network activated by softmax, enabling the acquisition of token weights. Consequently, an aggregated representation is obtained by calculating the weighted sum of contextual embeddings. Unlike average pooling, attention pooling takes into account the interdependencies among tokens.

A dense layer is utilized for classification on the aggregated embeddings. To address the issue of imbalanced distribution, we introduce focal loss to the loss function. By reducing the weights of simple negative samples during training, we enhance the accuracy of a limited number of positive samples. The Nadam optimizer, an extension of Adam with Nesterov momentum, is employed for the training process. Nadam [11] effectively utilizes previous momentum to constrain the current momentum and impacts every gradient during backpropagation, leading to improved sensitivity and slower gradient updates.

Entity Starting Annotation Module

The predicted entity numbers are encoded in a format of "entity start embedding", in which there is ES_x for the start position of xth entity or E_O for non-entity positions. For example $N = 2$, the entity start embedding is initialized with random ES_1, ES_2 and E_O. With shared contextual embeddings and the entity start embeddings, this module marks the starting positions of entities. The module regards the entity start annotation as a node classification problem based on a

[1] https://huggingface.co/roberta-base

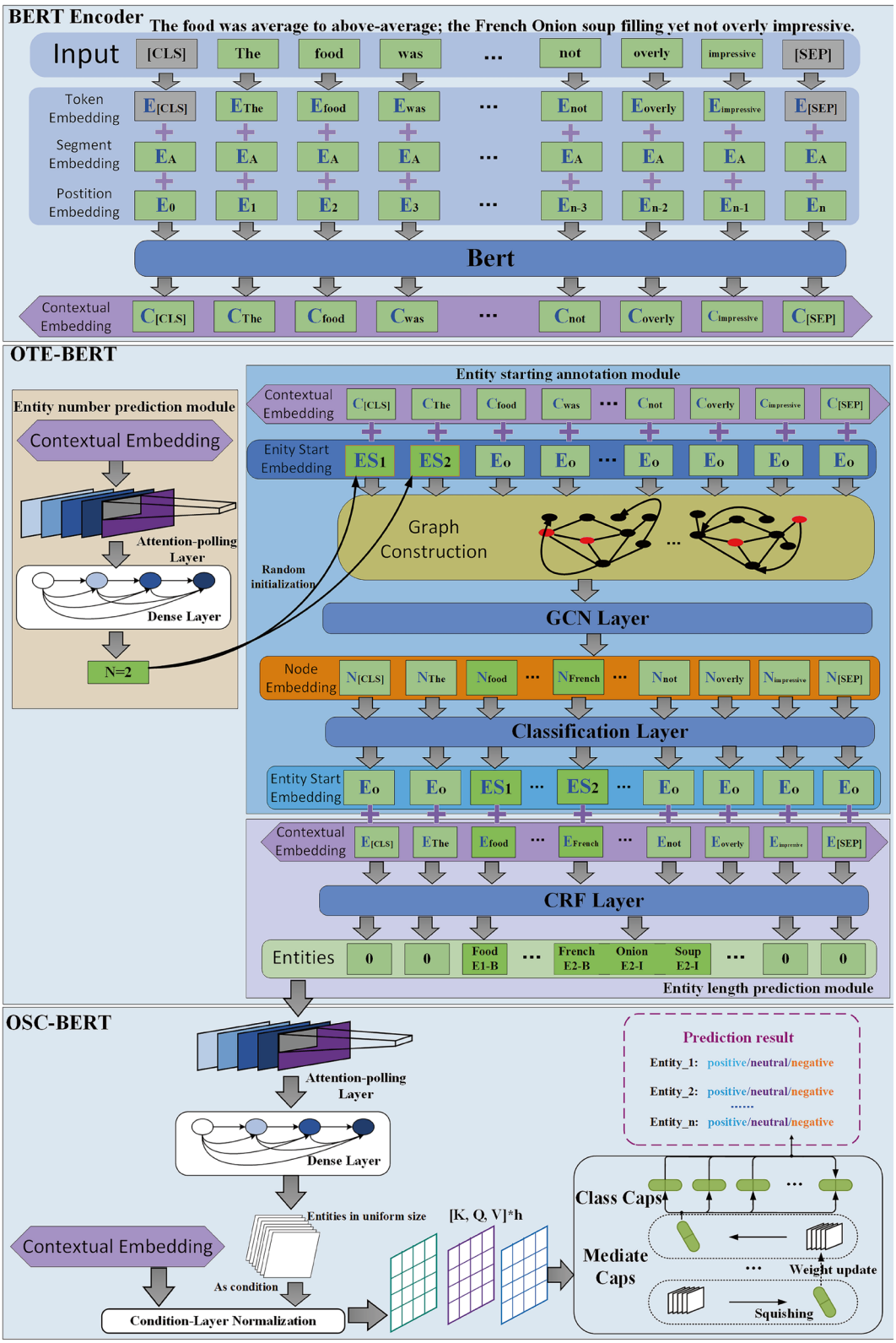


Fig. 1 The two-step architecture of O^2-Bert model where ESx is for the start position of xth entity; Eo is for non-entity positions; K, Q, V represents key matrix, query matrix, value matrix in the attention mechanism respectively (take the sentence 'The food was... not overly impressive' as example)

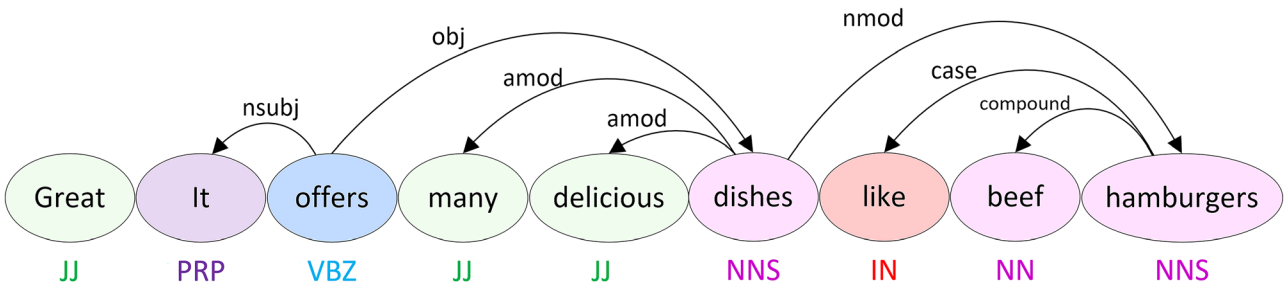


Fig. 2 The dependency parsing of the sentence *"Great! It offers many delicious dishes like beef hamburgers"*

graph convolutional network (GCN), where nodes represent the tokens in a sentence. It perceives the context around entities through considering both syntactic dependencies and long-term relationships.

Graph convolutional network [43] is a method to extract features from graph data. These features are used for classification and prediction. In Fig. 1 (starting annotation module), each word in the sentence is a node, and each node is composed of a D-dimensional vector. The characteristics of these nodes form an N × D-dimensional matrix X, where N is the number of words in the sentence. Then, the relationship between each node creates an N × N-dimensional matrix A, also known as an adjacency matrix. Take an example, the sentence*"Great! It offers many delicious dishes like beef hamburgers."*, we first construct a dependency tree in Fig. 2 where each node denotes the hidden state of words, and each edge represents syntactic dependency. We can see that there is a relation called "amod" between the word *"dishes"* and *"delicious"*. Then, we turn this graph into an adjacency matrix in Fig. 3, so it is "1" at the intersection of the word *"dishes"* and *"delicious"*.

Node characteristics (X) and an adjacency matrix (A) are the inputs of the GCN. The hidden layer node in the middle is calculated as follows:

$$h^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}h^{(l)}w^{(l)}\right) \tag{1}$$

where $\widetilde{A} = \mathrm{A} + \mathrm{I}$, I is the identity matrix; $\widetilde{D}$ is the degree matrix of $\widetilde{A}$; h (l) is the characteristic of layer L. For the

	Great	It	offers	many	delicious	dishes	like	beef	hamburgers
Great	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
It	**0**	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
offers	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**0**
many	**0**	**0**	**0**	**1**	**0**	**1**	**0**	**0**	**0**
delicious	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**0**
dishes	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**1**
like	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**1**
beef	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**
hamburgers	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**

Fig. 3 The adjacency matrix of the sentence *"Great! It offers many delicious dishes like beef hamburgers"*

input layer, h is X; σ is a nonlinear activation function; w (l) is the weight of layer L.

After simplifying the formula normalization operation, the characteristics of each node are calculated as follows:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N_i} \frac{1}{c_{ij}} h_j^{(l)} w^{(l)}\right) \quad (2)$$

where c_{ij} is the normalization factor and N_i is all neighbors of node I, including node I itself.

We utilize the default parameters and recommended hyperparameters as presented in the original GCN paper. The node classification output consists of logits from the classification model, which are further processed using softmax to obtain the final classification result. Specifically, the size of node features matches the word embedding size, and the number of nodes corresponds to the sequence length. Each node is connected to five adjacent word nodes in the context. Our training process involved five epochs with an initial learning rate of 1e-4, along with a warm-up phase consisting of 2000 steps.

The node classification model is trained by computing the cross-entropy loss function for all node embeddings, predicting the entity starting position based on the output entity start embeddings. Each node collects independent information from its neighboring nodes, including their characteristics, to obtain its own characteristic information.

The Entity Length Prediction Module

This section employs a CRF with entity start embeddings, utilizing the starting position and entity number as constraints to predict entity lengths. While CRF can extract entities independently, its accuracy is influenced by the selection of starting positions. Incorrect or incomplete selection of starting positions may result in a chain of errors. Furthermore, empirical evidence suggests that the recall of a standalone CRF is lower compared to incorporating an additional annotation module for labeling start positions. This module procedure is shown in Algorithm 1.

Algorithm 1: entity length prediction

Input: input layer pretrained with BRET s, predicted entity number n, entity beginning b
Output: the score $score_{l|s}$, the possibility $p_{l|s}$
Note: the label sequence l, characteristic functions f and its weight λ

Restraint s with n and b
Initialize $score_{l|s} \leftarrow 0$ **foreach** characteristic functions f_j in f **do**
  foreach tokens s_i in s **do**
    $score_{l|s} \leftarrow score_{l|s} + \lambda_j f_j(s, i, l_i, l_{i-1})$
  end
end
$p_{l|s} \leftarrow \exp(scores_{l|s}) \,/\, \sum_{l'} \exp(scores_{l'|s})$

During CRF training, the loss function enables the learning of word relations. In contrast, traditional neural networks output results through softmax without considering associations. If there is a need to learn such relations, previous layers of the model must account for it. The models' ability to establish connections is significantly impacted by the choice of loss function and tag settings.

CRF computes the probability of each tag at every position, taking into account the likelihood of a tag sequence. It evaluates the probabilities for each position and selects the method with the highest probability as the final outcome. For example, there is a sample with a length of eight, and every word may be one of the three categories, such as noun (n), adjective (a), and verb (v). In our CRF, we use the label system that includes '<start>' (the beginning of a sentence), '<stop>' (the ending of a sentence), 'B' (the beginning of entity), 'E' (the rest words of entity), 'O' (nonentity) and '<mask>' (other objects).

CRF assumes the position of every word as n (noun), a (adjective) or v (verb) and adds all the probability of every position as the probability of a situation. Finally, it selects the maximum from all conditions, as in Example 4.

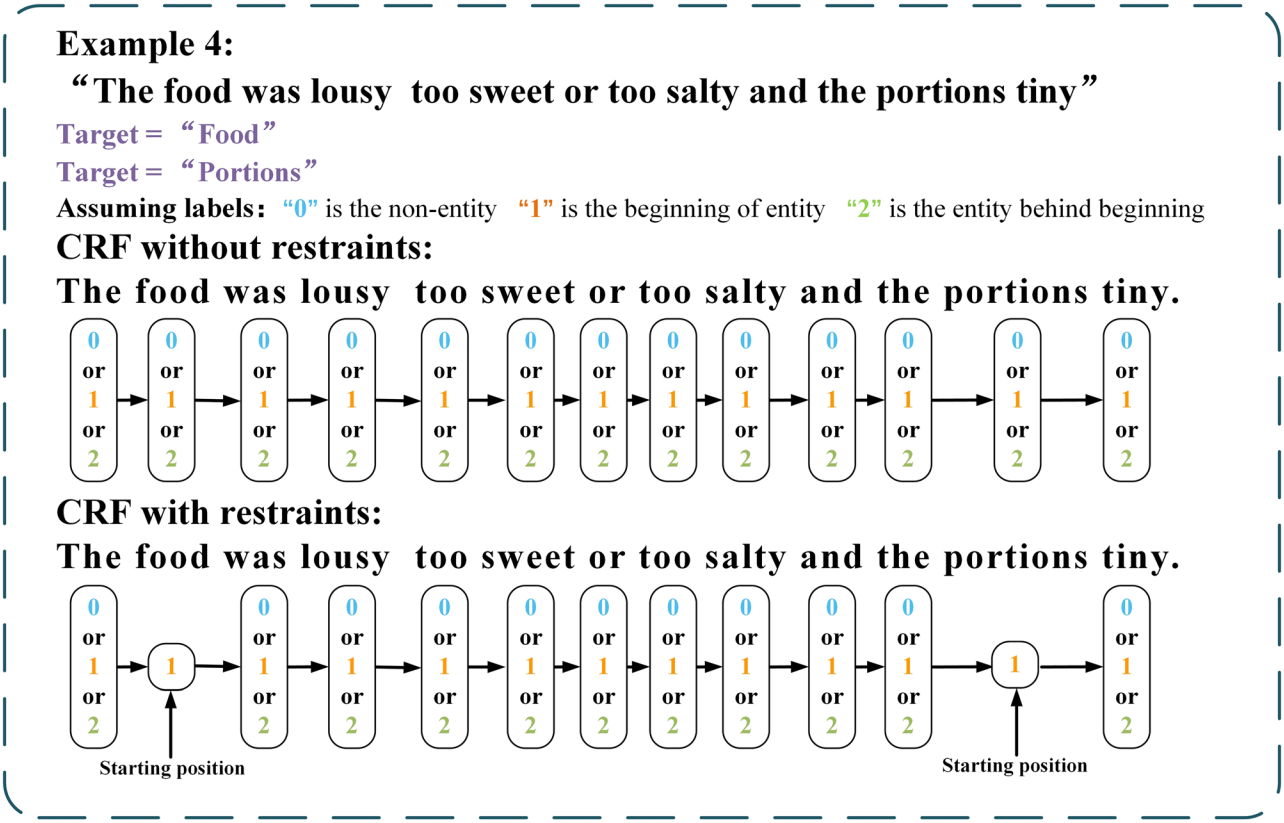
The loss computation in the OTE involves a weighted sum of the losses from three submodules, where the weights are determined as hyperparameters.

OSC-Bert Framework

In ⑥ in Table 1, different "rolls" correspond to different polarities, while for most approaches result in similar polarities. Therefore, it is challenging for traditional sentiment analysis models to distinguish the contradictory polarities of multiple targets. The phenomenon is common in real-world applications, accounting for 21% on SemEval.

We utilize an attention mechanism to derive the sentiment polarity associated with entities. Moreover, the condition layer normalization algorithm [44] normalizes the input sentence into various normal distributions by incorporating an "input condition" within the range of (0,1) distribution from layer normalization.

The procedure of OSC is depicted in Algorithm 2. This model takes the entities as the conditions of condition layer normalization, which can normalize the different entities in the same sentence into different distributions. Finally, this model uses capsule networks to solve the "contradictory-polarity problem". Capsule networks offer a more comprehensive determination by considering the relevance between outputs, in contrast to softmax-based methods.



Example 4 A case of CRF with the starting position and entity number as the restraint

Algorithm 2: attention mechanism and routing-by-agreement in capsule network

f **Input:** input layer pretrained with BRET s, entities layer from OTE y, prediction vectors $u_{j|i}$, current capsule layer l, routing iteration times r
Output: vector output v_j of capsule j in l layer
Note:coupling coefficients c_{ij}, input vector of v_j, input dimension d, in the attention mechanism key matrix K, query matrix Q, value matrix V, the output of the attention mechanism Z

$y \leftarrow$ attention-pooling(y) and dense
Normalize s with y as condition
Initialize K,Q,$V \leftarrow s$
$Z \leftarrow \text{softmax}\left(\frac{Q \times K^T}{\sqrt{d}}\right)$, put Z into capsule
foreach capsule i in layer l and capsule j in layer $(l+1)$ **do**
  $b_{ij} \leftarrow 0$
end
foreach r iterations **do**
  foreach capsule i in layer l **do**
    foreach capsule j in layer $(l{+}1)$ **do**
      $c_{ij} \leftarrow \text{leaky-softmax}(b_{ij})$
    end
  end
  foreach capsule j in layer $(l{+}1)$ **do**
    $s_j \leftarrow \sum_i c_{ij} u_{j|i}$
    $v_j \leftarrow \text{squashing}(s_j)$
  end
  foreach capsule i in layer l **do**
    foreach capsule j in layer $(l{+}1)$ **do**
      $b_{ij} \leftarrow b_{ij} + u_{j|i} \cdot v_j$
    end
  end
end
return v_j

To mitigate the impact of irrelevant paddings, we employ an attention pooling layer and a dense layer. Additionally, the use of attention helps prevent the capsule network from disregarding the contextual-opinion target relationship.

The capsule network utilizes a trained clustering algorithm to classify the sentiment polarity associated with each target. Detailed rule analysis ensures accurate target extraction and differentiation of various polarities. Ultimately, a category vector, encompassing polarity markings, is optimized and generated during network training.

The O^2-Bert model is suitable for short sentences (usually less than 128 tokens) in English corpuses, and is intended for English corpuses after tokenization. The model is not domain-specifically designed, therefore, both tweets and news texts are suitable. However, document-level journalistic news may not be suitable due to its length.

Experiment

This paper carries out experimental comparisons from two perspectives. From the first perspective, we present our idea for how to solve the two difficulties in OTE , the “NULL” targets and the “Lengthy” targets. Comparison experiments

are conducted to analyze the influence of the proposed model on these two problems.

Bi-LSTM, Bi-LSTM+CRF, BERT+CRF, and BERT+Bi-LSTM+CRF are chosen as baseline models in OTE stage. The results are cited from SpanMlt [18]. ATSE and DE-CNN are also chosen as baseline models. We also perform ablation studies to verify the effectiveness of each module through four comparison groups.

From the second perspective, this paper extracts the opinion target separately and predicts the sentiment analysis based on the opinion target, thus, comparing the approach based on one stage. By comparing Bert-based, Graph-based and some classical annotation methods with the current popular sentiment classification models, the performances of different models are analyzed from the overall perspective. All the codes for this experiment are available on GitHub (https://github.com/SuperCornly/TBSA).

Dataset and Experiment Settings

We use the SemEval2014-16 dataset in the experiment. This dataset comments contain the laptops and restaurants domain, presented in Table 3. The number of entities in each sentence is uneven. In the SemEval2014 dataset, none of the sentences include NULL, but the composition of words is more complex. The approximate distribution is presented in Table 4. In SemEval2015 & SemEval2016, most samples contain two entities, while the number of samples containing more entities, such as six or eight, is very limited, and some entities are null.

Ten cross-validations are used in the experiment. We divided the official training set into a training set and a validation set, respectively 70% and 30%, and kept the distribution of the samples in the training set. And we take the official test set as our test set. The datasets division is presented in Table 5.

We utilize F1-score to evaluate the OTE task, as it shares similarities with information retrieval. For assessing aspect-specific sentiment polarity, accuracy (Acc) and F1-score are commonly used to represent true results in evaluating the OSC task. Acc and F1-score are calculated as follows:

$$\begin{cases} Acc = \dfrac{TP + TN}{TP + TN + FP + FN} \\ R = \dfrac{TP}{TP + FN} \\ P = \dfrac{TP}{TP + FP} \\ F1 = \dfrac{2 * P * R}{P + R} \end{cases} \tag{3}$$

where TP is true positive sample, FP is false positive sample, TN is true negative sample, and FN is false negative sample.

Table 3 Description of the datasets

Sentiment	Laptop2014		Restaurant2014		Restaurant2015		Restaurant2016	
	Train	Test	Train	Test	Train	Test	Train	Test
Positive	1002	348	2216	737	1178	341	1618	596
Neural	471	167	643	197	48	34	88	36
Negative	885	134	834	200	380	328	708	189

Table 4 Probability distribution of the number of entities

Number	Laptop2014		Restaurant2014		Restaurant2015	
	Train	Test	Train	Test	Train	Test
1	930	266	1023	298	801	311
2	354	105	572	186	212	127
3-6	199	50	417	128	67	15
>6	5	0	9	2	25	3

Table 5 Datasets division

Division	Laptop2014	Restaurant2014	Restaurant2015	Restaurant2016
Train	1840	2585	1124	1690
Validation	788	1108	482	724
Test	649	1134	703	821

The O^2-Bert model designed in this paper uses Nadam as the optimizer, and the initial learning rate is 0.15. With the increase in epochs, the power index of the learning rate decreases, and it closes to 1E-5 after 500 epochs. The weight attenuation coefficient of the optimizer is 1E-2, and the momentum value is 0.9. To improve the robustness of the model parameters, Gaussian noise is randomly added to 1/30th of the data during the training of each epoch. At the same time, slight perturbation of the backpropagation gradient is performed with a probability of 0.001.

Comparative Methods

Baseline Models

- **Bi-LSTM** is the abbreviation of "Bidirectional Long Short-Term Memory," which is made up of a forward LSTM and a backward LSTM. However, there is a problem with modeling sentences using LSTM alone. It cannot encode information from back to front. For more grain-fined classification, especially when there are several entities and opinion words in a sentence, Bi-LSTM may figure out the "pair" better by catching more context information.
- **Bi-LSTM+CRF**: The input sequence undergoes embedding transformation into a vector sequence. Two bidirectional LSTM units process the input. The forward and backward outputs are concatenated via a fully connected layer, resulting in a vector with dimensions corresponding to the output tags defined by the CRF feature function. The output is then normalized using softmax to obtain label probabilities.
- **BERT+CRF** uses BERT as the pretraining model; the output hidden states of input words are taken as the features for CRF.
- **BERT+Bi-LSTM+CRF**: BERT, functioning as a pretraining model resembling the transformer encoder, generates an embedding vector that captures contextual and word information at the current position. Bi-LSTM extracts the embedding feature and passes it to CRF for classification outcomes.
- **SpanMlt** [18] is a framework that is a span-based multi-task, and the task contains pairwise aspect and opinion term extraction.
- **ATSE** [20] treats the opinion target extraction task as a question-answering machine reading comprehension task. It utilizes a span-based tagging scheme to handle cases where a token can belong to multiple entities.
- **DE-CNN** [9] is a post-processing method to control the number of extracted opinion targets and correct the boundary of the extracted opinion targets. They proposed aspect number determining module and aspect boundary modifying module to better address the errors in extracted opinion targets.
- **TextCNN** [19]: It is an enhanced CNN-based model for text tasks, comprising four layers: input, convolution, pooling, and fully connected softmax layer. The input of the model is the word embedding of each word, and after the fully connected softmax layer, it can output the classification probability.
- **Distance-rule** [29] summarizes customers' reviews in three steps: first they mine product features that appear as nouns or noun phrases in comments by part-of-speech (POS) tagging and association mining. Then the model regards adjectives as opinion words and determines their polarities by employing the synonym and adjective synonym set in WordNet [45]. Moreover, for infrequent feature, the model regards the nearest noun or noun phrase as the opinion target for an opinion word. Finally, it predicts the polarity of a sentence by analyzing all the opinion words in it and generates the final result by summing up all product features.
- **Dependency-rule** [36] proposes dependency tree based templates to identify opinion pairs, making use of the POS tag of opinion targets and opinion words and the dependency path between them.
- **ATAE-LSTM** [19]: ATAE-LSTM encompasses two primary features: aspect embedding and attention mechanism. The former involves learning embedding vectors for each aspect, while the latter focuses on determining the weights that indicate the significance of each word through attention.
- **TransCap** [46] proposes a transfer capsule network model to transfer document-level knowledge to aspect-level representations.
- **IACapsNet** [30] utilizes a capsule network to capture vector-based feature representation. Through the incorporation of an interactive attention EM-based capsule routing mechanism, IACapsNet effectively learns the semantic correlation between opinion targets and opinion words.
- **SGGCN+BERT** [26] employs a gate vector to leverage the representations of the opinion words. Additionally, leveraging opinion words information, it modulates the hidden vectors of graph-based models.
- **CapsNet+BERT** [47]: CapsNet is a fusion of a conventional CNN and a unique fully connected Capsule layer. It comprises three layers: a standard CNN as the first layer, a primarycaps layer as the second, and a digitcaps layer as the third. In Capsule, each cap neuron establishes connections with all cap neurons in the subsequent layer.
- **MHAGCN (BERT)** [24] is a graph convolutional network with a hierarchical multi-head attention mechanism. It aims to leverage the relationship between opinion targets and their context by incorporating semantic information and syntactic dependencies.

- **GP-GCN (BERT)** [25] simplifies the global feature by utilizing orthogonal projection in the process of GCN. It captures the local dependency structure of sentences by syntactic dependency structure and sentence sequence information. Moreover, it proposed a percentage-based multi-headed attention mechanism to better represent the critical output of GCN.
- **ASGCN-DT** [34] **and ASGCN-DG** [34]: Based on directional and un-directional graph respectively, ASGCN-DT and ASGCN-DG both build a GCN, extracting syntactical information and word dependencies over the dependency tree.
- **BiGCN** [35]: It involves constructing a global vocabulary graph from the training corpus, along with local syntactic and vocabulary graphs for each sentence. Additionally, a conceptual hierarchy is employed to differentiate various types of dependent or symbiotic relationships. To extract comprehensive sentence features, the HiarAgg module facilitates interaction between the vocabulary graph and the syntactic graph. By utilizing a mask and gate control mechanism, contextual information is obtained, leading to improved performance in predicting target polarity.
- **CGAT** [32] proposes a contextual attention network which contains two graph attention networks and a contextual attention network to capture aspect-sensitive text features. Furthermore, a novel syntactic attention mechanism based on relative distance is introduced to enhance focus on opinion targets while mitigating computational complexities.

Ablation Models

w/o n and **w/o n+s** is where "w/o" indicates that this component is not included in the O^2-Bert model. "n" represents entity number, "n+s" represents entity number with entity starting. In the starting position module experiment, the influence of the GCN and ensemble learning on the starting position prediction component is compared with the **w/o s** and **w/o se** methods, respectively. "s" represents entity starting, "se" represents using ensemble instead of starting.

Results and Discussion

The Experimental Result of Extracting the Opinion Target in OTE-Bert

The result is exhibited in Table 6. According to this table, the O^2-Bert model is superior to most of other models, especially in the SemEval2015 and SemEval2016.

Comparing Bi-LSTM with Bi-LSTM+CRF, and ATSE There is a defect when just using Bi-LSTM. For example, in the "BIO" tag system, "I" must follow "B," and it is impossible that "O" appears in the middle of two "I". Without the restraint of transmission probability, the model's output may be wrong. CRF can fully consider the order of the labeling sequence to obtain the best global sequence result. For this reason, due to the lack of CRF, it is possible for a neuron network to split a completed entity into several pieces. The feature function of CRF exists to observe and learn various features (N-gram) of a given sequence, which are the relations between multiple words under the limited window size. ATSE incorporates two binary classifiers to accurately identify the starting and ending positions of each opinion target. This allows it to effectively handle scenarios where a token is associated with multiple distinct entities.

Compare BERT+CRF with Bi-LSTM+CRF, BERT+Bi-LSTM+CRF, and DE-CNN BiLSTM combines forward and backward LSTMs to enhance the contextual understanding of text sequences. BERT improves word representation through

Table 6 F1-score for opinion target extraction (OTE-Bert) on four datasets

Method	Lap2014 (%)	Rest2014 (%)	Rest2015 (%)	Rest2016 (%)
Bi-LSTM [18]	55.25	51.90	53.28	51.83
Bi-LSTM+CRF [18]	69.80	78.03	66.27	70.43
BERT+CRF [18]	56.38	54.37	57.01	55.83
BERT+Bi-LSTM+CRF [18]	56.99	54.08	55.85	55.18
SpanMlt [18]	84.51	87.42	81.76	85.62
ATSE [20]	82.47	87.85	77.72	83.34
DE-CNN [9]	**84.89**	88.41	73.47	78.83
O^2-Bert (ours)	84.63	**89.20**	**83.16**	**86.88**
w/o n	81.53	85.31	78.03	80.02
w/o s	75.83	79.38	77.26	78.34
w/o n+s	72.14	80.27	73.72	76.69
w/o se	75.02	78.14	77.56	76.30

Bold indicates best performance

pretraining and simplifies entity segmentation by outputting only the maximum score for each word. Consequently, Bi-LSTM+CRF outperforms BERT+CRF. In contrast to BERT, DE-CNN introduces a post-processing approach that regulates the number of extracted opinion targets and rectifies their boundaries. They define positive and negative samples for both subtasks, enabling the post-processing modules to learn from multiple perspectives and achieve superior outcomes.

Comparison O^2-Bert with SpanMlt It can be concluded from Table 6 that the experimental result of the O^2-Bert model in the OTE task obtains the best results. The O^2-Bert model presented in this paper leverages an entity number prediction module, entity starting annotation module, and entity length prediction module to capture in-depth semantic information for each word. By incorporating these components, the O^2-Bert model effectively identifies NULL cases in sentences, leading to improved performance in the OTE task. Moreover, the entity starting annotation module and entity length prediction module enhance the recognition of multi-word entities, resulting in higher overall accuracy and enabling the model to extract richer semantic meaning, ultimately yielding superior experimental outcomes.

Comparing the w/o n with the O^2-Bert Model shown in SemEval 2015 and 2016 is much improved. We summarized and analyzed the results of the output samples. It was found that it performed better because the model predicts the number of entities in a NULL case in the sentence, which can improve the accuracy of entity extraction. There is no "NULL" entity in the 2014 dataset, so it does not highlight the advantages of the entity number module. However, in the 2015 and 2016 datasets, the accuracy of the entity's number prediction is 100%. The entity number prediction module utilizes attention pooling to extract meaningful information from the dataset and reduce the dimension, which verifies the importance of this module.

Compare w/o s with O^2-Bert The starting position module effectively distinguishes short entity words that are overwritten by long entities, such as Example 3 (Target1 and Target3). In the GCN graph network, w/o s represents the data based on BERT pretraining as a graph structure and predicts the starting position of output entities through the connection of nodes between layers. For this submodule, a comparative study is designed based on ensemble learning that combines DGCNN and multi-head attention methods to predict the start position of entity words. (The experimental code of this paper is posted on GitHub for discussion.) Ensemble learning means that in the same samples, models have different classification effects. Ensemble learning combines the results of multiple models to let the models complete each other. This paper compares the influence of GCN and ensemble learning on the starting position annotation module, and the practical effect is worse than that of the GCN-based method. Based on ensemble learning (w/o se), the weight coefficients of participating models need to be adjusted. After training the single classifier, the number of top N needs to be set manually, and artificial rules and noises are introduced.

Comparing w/o n+s with O^2-Bert w/o n+s is equivalent to extracting the opinion target of the OTE task based on CRF only. CRF can complete entities' "BIO" labeling, but the recall value is relatively low. The starting location marker and the number of entities can be used as restrictions to improve the accuracy of the final entity prediction, as shown in "The Entity Length Prediction Module" section. This is why this paper introduces these two submodules.

The Experimental Result of Opinion Sentiment Classification in OSC-Bert

The OSC task is based on the attention network to predict sentiment analysis, and we select TextCNN, Distance-rule, Distence-rule, LSTM, ATAE-LSTM, TransCap, IACapsNet, BERT, SGGCN+BERT, CapsNet+BERT, MHAGCN, GP-GCN, ASGCN-DT, ASGCN-DG, BiGCN, and CGAT as comparative models. The result is exhibited in Table 7.

We conduct a comparative analysis of our methods with alternative approaches, classified into three distinct groups.

Network

Compare with TextCNN The TextCNN model utilizes convolution to capture n-gram features from sentences, effectively extracting shallow text features. However, it heavily relies on the filter window for feature extraction, posing limitations and insensitivity to longer texts. Consequently, the model may not perform well in classifying emotional evaluations expressed in inverted sentences.

Compare with Distance-rule and Dependency-rule The distance-rule and dependency-rule are notable rule-based techniques that identify opinion words and determine their associated opinion targets based on distance or dependency tree analysis. However, they encounter difficulties in handling complex sentence structures and implicit opinion targets.

Comparing LSTM with ATAE-LSTM The conventional LSTM model lacks the ability to identify the crucial aspect-level information in sentiment classification. However, by

Table 7 Acc and F1-score for opinion sentiment classification (OSC-Bert) on four datasets

Model		Lap2014		Rest2014		Rest2015		Rest2016	
		Acc (%)	F (%)	Acc (%)	F (%)	Acc (%)	F (%)	Acc (%)	F (%)
Network	TextCNN [19]	55.16	48.81	47.69	42.58	-	-	-	-
	Distance-rule [29]	58.39	49.92	50.13	40.42	54.12	45.97	61.90	51.83
	Dependency-rule [36, 37]	64.57	58.04	45.09	37.14	65.49	55.98	76.03	64.62
	LSTM [19]	52.64	58.34	55.71	56.52	57.27	58.93	62.46	65.33
	ATAE-LSTM [19]	77.32	66.57	69.14	63.14	75.43	56.34	83.25	63.85
	TransCap [46]	79.29	70.85	73.87	70.10	-	-	-	-
	IACapsNet [30]	81.79	73.40	76.80	73.29	-	-	-	-
Bert	BERT [27]	84.11	76.68	77.59	73.28	83.48	66.18	90.10	74.16
	SGGCN+BERT [26]	87.20	82.50	82.80	80.20	82.72	65.86	**90.52**	74.53
	CapsNet+BERT [47]	-	76.37	-	73.58	-	70.56	-	76.36
	MHAGCN+BERT [24]	79.06	75.70	82.57	75.83	-	-	-	-
	GP-GCN+BERT [25]	83.89	75.09	83.90	66.89	**87.78**	72.89	75.90	73.90
Graph	ASGCN-DT [34]	80.86	72.19	74.14	69.24	79.34	60.78	88.69	66.64
	ASGCN-DG [34]	80.77	72.02	75.55	71.05	79.89	61.89	88.99	67.48
	BiGCN [35]	81.97	73.48	74.59	71.84	81.16	64.79	88.96	70.84
	CGAT [32]	86.25	80.38	81.41	76.48	-	-	-	-
	O^2-Bert (ours)	**88.43**	**82.90**	**86.81**	**80.73**	86.94	**76.94**	89.83	**83.58**

Bold indicates best performance

incorporating the attention mechanism from ATAE, the model effectively captures the pivotal aspect based on the opinion target. This enables the model to leverage aspect information and learn the inherent relationship between words and input aspects, resulting in enhanced accuracy in classification.

Compare with TransCap The TransCap combines aspect-level and document-level data through aspect-based routing to generate semantic capsules. However, extracting multi-aspect sentences still poses a challenge.

Compare with IACapsNet The IACapsNet utilizes a capsule network to create vector-based feature representations and employs EM routing algorithms for feature clustering. Additionally, it incorporates an interactive attention mechanism to model the semantic relationships between opinion targets and contexts. However, its performance is relatively subpar when dealing with the "NULL" case.

Bert

Compare with BERT This paper follows a similar data processing approach as previous studies, wherein the input sentence and opinion target are transformed into embedding vectors. The opinion target embedding is represented as an average value. In O^2-Bert, the vector is passed through a bidirectional GRU with a residual connection to obtain the contextual representation. The central capsule is then derived from the contextual representation, incorporating the aspect ratio embedding. This process leverages prior knowledge of the emotion category to enhance the routing process. During the classification of capsules, aspect awareness is employed to normalize the weights and guide the routing, ultimately leading to the calculation of the final capsule for classification.

Compare with SGGCN+BERT, CapsNet+BERT, MHAGCN (BERT), and GP-GCN (BERT) In contrast to O^2-Bert, CapsNet+BERT and GP-GCN+BERT utilize BERT instead of embedding and encoding layers, SGGCN+BERT, a graph-based deep learning model, considers the opinion targets and make use of the overall contextual importance scores obtained from the dependency tree. MHAGCN+BERT reconstruct the given context and target fed into BERT to facilitate the training and fine-tuning. In addition, based on the attention mechanism, O^2-Bert in this paper can give different weights to different words in this aspect. The sentence representation obtained through the weighted sum can improve the accuracy and better analyze the emotional polarity.

Graph

Compare with ASGCN-DT, ASGCN-DG, BiGCN, and CGAT With the dependency trees, ASGCN-DT and ASGCN-DG build a GCN to exploit syntactic information and word dependencies. CGAT reconstructs dependency trees connecting target aspects with context words and utilizes a graph attention network to aggregate sentiment information. Both BiGCN and

our approach employ GCN networks, but our method, which incorporates graph and syntax fusion, demonstrates superior performance compared to most methods. This indicates that mining dependency relation is more valuable in identifying sentiment polarity (refer to the "Entity Starting Annotation Module" section). However, BiGCN struggles to accurately identify and assign polarity to the "NULL" case.

During training, O^2-Bert utilizes the focal loss and Nadam optimizer to optimize the model and achieve superior outcomes. While the focal loss was initially designed for binary classification, we extend its functionality to enable multiclassification for the named entity relation task. The Nadam optimizer is an enhanced version of Nadam that incorporates Nesterov momentum. Unlike standard degradation, each Nadam update considers both the current gradient and the accumulation of previous momentum, thereby facilitating more effective updates.

The Effect of the Training Data Size on the Model's Performance

Moreover, we have investigated how the training data size affects our model's performance. As in Fig 4, we can see that with the increasing number of training data size, our model is achieving better performance.

Case Study

O^2-Bert effectively addresses challenges related to the "NULL" entity and overlapping entities, as previously stated. To verify that, we pick some sentences and list the results in Table 8. In Sentences 2 and 5, we can see that through the entity number prediction module, a "NULL" entity with its syntactic dependency word *"worth"* can be found. And the results of Sentences 3 and 4 indicate that the attention mechanism plays an important role in recognizing entities. Moreover, capturing the entity *"beef hamburgers"* in Sentence 5 shows that the entity starting annotation module and the entity length prediction module do work. However, there are still bad case. In Sentence 6, our model extracts the entity *"fish"* with the syntactic dependency words *"not…came"* , and thus thinks it negative. In Sentence 7, it's too hard for our model to capture the description *"Eight out of ten!"*, so it simply finds some useless words like *"comments"*, *"not"*, and *"true"* our model fails to understand the statement *"Eight out of ten!"*, and make a judgement through "comments", "not so true"..

To showcase the efficacy of the attention mechanism in the OSC task, we choose Sentences 1 and 4 as illustrative examples. We visualize the attention weights between entities and other words by heat map in Fig 5. The darker the color, the greater the attention weights. For Case 1, we can see that the word *"incredible"* has the highest attention weight among all

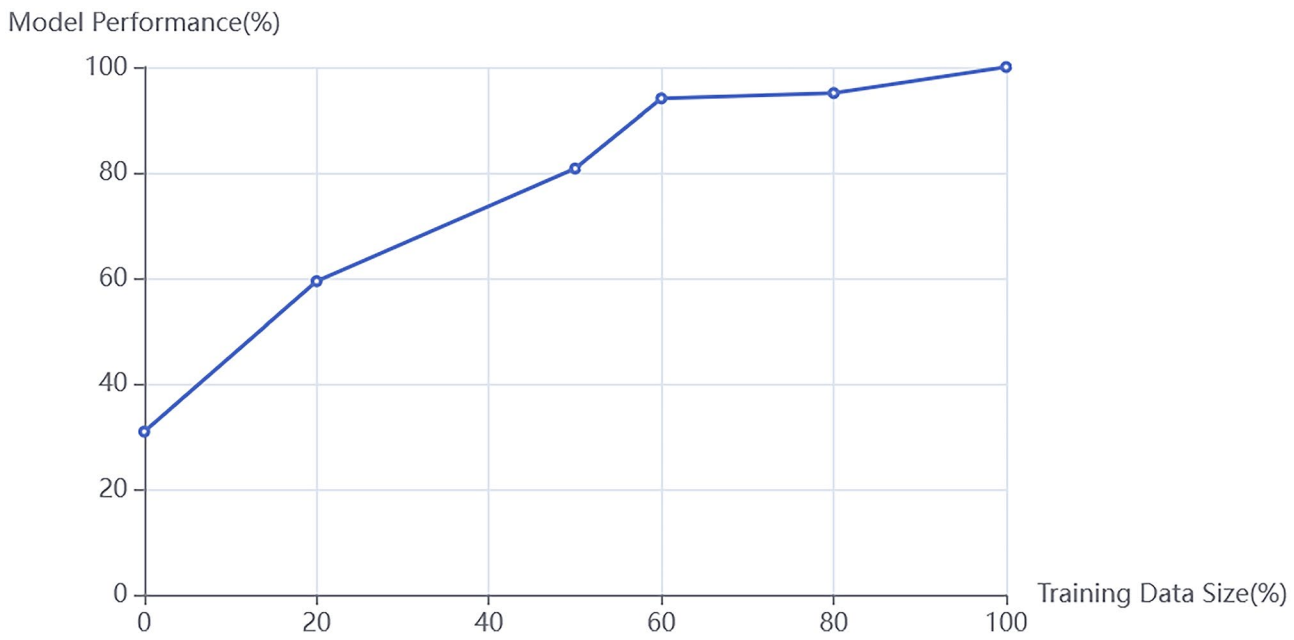
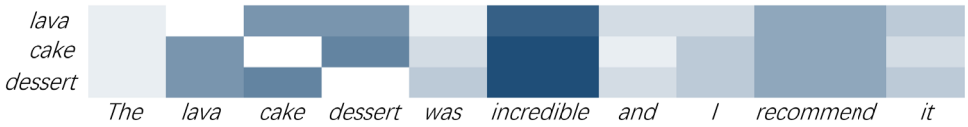


Fig. 4 The effect of the training data size on the model's performance

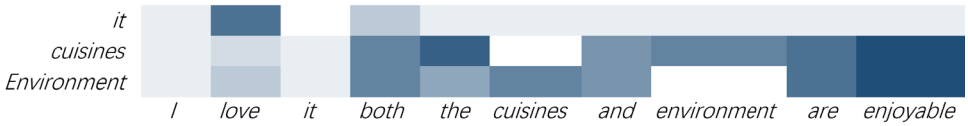
Table 8 Entity and polarity captured by O^2-Bert

Sentence	O^2-Bert	
	Entity	Polarity
1. The lava cake dessert was incredible and I recommend it.	*lava cake dessert*	Positive✓
2. Slightly on the pricy side but worth it.	NULL	Positive✓
	pricy side	Negative✓
3. I like the somosas, chai, and the chole,but the dhosas and dhal were kinda disappointing.	*somosas*	Positive✓
	chai	Positive✓
	chole	Positive✓
	dhosas	Negative✓
	dhal	Negative✓
4. I love it, both the cuisines and environment are enjoyable.	*cuisines*	Positive✓
	environment	Positive✓
5. Great! It offers many delicious dishes like beef hamburgers.	NULL	Positive✓
	dishes	Positive✓
	beef hamburgers	Positive✓
6. Not until the fish came did it worth a five-star.	*fish*	Negative✘
7. Eight out of ten! I think the comments about it are not so true.	*comments*	Negative✘

The opinion targets and the opinion words are colored blue and red respectively. The correct predictions are marked with ✓ and incorrect predictions are marked with ✘



(a) Case 1: *The lava cake dessert was incredible and I recommend it.* (Sentence 1)



(b) Case 2: *I love it, both the cuisines and environment are enjoyable.* (Sentence 4)

Fig. 5 Visualization of the attention weights over two cases

words, and that's because it has a syntactic dependency with *"lava cake dessert"*. Also, as an adjective, it expresses a strong sentiment. Similarly, in Case 2, our model's heatmap demonstrates its heightened focus on the correlation between entities and their corresponding syntactic dependency words, aligning with our expectations.

Conclusion

In this paper, we propose a novel approach to address the TBSA task by decomposing it into two stages. We introduce the O^2-BERT model, which tackles the "two problems" in a sequential manner. Specifically, OTE-Bert is designed for opinion target extraction, while OSC-Bert focuses on sentiment classification. Our statistical analysis confirms the efficacy and resilience of our work utilizing the "two-stage paradigm." Through solving the challenges of "NULL entities" and multiple-word entities, the evaluation results on SemEval2014-16 show that our framework achieves better or comparable performance compared to the sota models, with a superior F1-score performances (0.4% on 2014 Laptop, 0.53% on 2014 Restaurant, 4.05% on 2015 Restaurant, and 7.22% on 2016 Restaurant respectively). In future work, we might explore more possibilities of the new paradigm on other NLP tasks and we may also attempt to introduce Large Language Model and prompt engineering into TBSA tasks.

Funding This study was funded by Fundamental Research Funds for Central Universities and National foreign specialized projects (G2022123009L).

Data and Code Availability The dataset SemEval2014-16 we used is publicly available and the code that supports the results of this study is openly available in https://github.com/SuperCornly/TBSA.

Declarations

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

Conflict of Interest The authors declare no competing interests.

References

1. Wang D, Fan H, Liu J. Learning with joint cross-document information via multi-task learning for named entity recognition. Inf Sci. 2021;579:454–67.
2. Tang H, Ji D, Zhou Q. End-to-end masked graph-based crf for joint slot filling and intent detection. Neurocomputing. 2020;413:348–59.
3. Ni J, Huang Z, Hu Y, Lin C. A two-stage embedding model for recommendation with multimodal auxiliary information. Inf Sci. 2022;582:22–37.
4. Zhang Y, Du J, Ma X, Wen H, Fortino G. Aspect-based sentiment analysis for user reviews. Cogn Comput. 2021;13(5):1114–27.
5. Guo L , Jiang S , Du W , Gan S. Recurrent neural crf for aspect term extraction with dependency transmission. In: CCF International Conference on Natural Language Processing and Chinese Computing. Springer; 2018 p. 378–90.
6. Lu J, Liu W. Automatic information extraction for financial events by integrating bigru and attention mechanism. J Phys Conf Ser. 2022;2171.
7. Kang T, Lee M, Yang N, Jung K. RABERT: Relation-aware BERT for target-oriented opinion words extraction. New York, NY, USA: Association for Computing Machinery; 2021. p. 3127–31.
8. Bi Q, Zhang H, Qin K. Multi-scale stacking attention pooling for remote sensing scene classification. Neurocomputing. 2021;436:147–61.
9. Wang R, Liu C, Zhao R, Yang Z, Zhang P, Wu D. Post-processing method with aspect term error correction for enhancing aspect term extraction. Appl Intell. 2022;52:15751–63.
10. Pour AAM , Jalili S. Aspects extraction for aspect level opinion analysis based on deep cnn. In: 2021 26th International Computer Conference, Computer Society of Iran (CSICC). 2021. p. 1–6.
11. Dozat T. Incorporating Nesterov momentum into Adam. 2016.
12. Su J, Yu S, Luo D. Enhancing aspect-based sentiment analysis with capsule network. IEEE Access. 2020;8:100551–61.
13. Jochim C, Deleris L. Named entity recognition in the medical domain with constrained CRF models. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain: Association for Computational Linguistics; 2017. p. 839–49.
14. Gutierrez BJ, McNeal N, Washington C, Chen Y, Li L, Sun H, Su Y. Thinking about gpt-3 in-context learning for biomedical IE? Think again. In: Conference on Empirical Methods in Natural Language Processing. 2022.
15. Rana TA, Cheah Y-N. A two-fold rule-based model for aspect extraction. Expert Syst Appl. 2017;89:273–85.
16. Li L, Liu Y, Zhou A. Hierarchical attention based position-aware network for aspect-level sentiment analysis. In: Proceedings of the 22nd Conference on Computational Natural Language Learning. Brussels, Belgium: Association for Computational Linguistics; 2018. p. 181–9.
17. Shams M, Baraani-Dastjerdi A. Enriched lda (elda): Combination of latent dirichlet allocation with word co-occurrence analysis for aspect extraction. Expert Syst Appl. 2017;80:136–46.
18. Zhao H, Huang L, Zhang R, Lu Q, Xue H. SpanMlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online. Association for Computational Linguistics; 2020. p. 3239–48.
19. Wang Y, Huang M, Zhu X, Zhao L. Attention-based lstm for aspect-level sentiment classification. In: Proceedings of the 2016 conference on empirical methods in natural language processing. 2016. p. 606–15.
20. Gao L, Wang Y, Liu T, Wang J, Zhang L, Liao J. Question-driven span labeling model for aspect-opinion pair extraction. In: AAAI. 2021.
21. Hu M, Peng Y, Huang Z, Li D, Lv Y. Open-domain targeted sentiment analysis via span-based extraction and classification. arXiv:1906.03820 [Preprint]. 2019. Available from: http://arxiv.org/abs/1906.03820.
22. Xu L, Chia YK, Bing L. Learning span-level interactions for aspect sentiment triplet extraction. arXiv:2107.12214 [Preprint]. 2021. Available from: http://arxiv.org/abs/2107.12214.
23. Yu Bai Jian S, Nayak T, Majumder N, Poria S. Aspect sentiment triplet extraction using reinforcement learning. New York, NY, USA: Association for Computing Machinery; 2021. p. 3603–7.
24. Li X, Ran L, Liu P, Zhu Z. Graph convolutional networks with hierarchical multi-head attention for aspect-level sentiment classification. J Supercomput. 2022;78:14846–65.
25. Wei S, Zhu G, Sun Z, Li X, Weng TH. Gp-gcn: Global features of orthogonal projection and local dependency fused graph convolutional networks for aspect-level sentiment classification. Connect Sci. 2022;34:1785–806.
26. Veyseh APB, Nour N, Dernoncourt F, Tran QH, Dou D, Nguyen TH. Improving aspect-based sentiment analysis with gated graph convolutional networks and syntax-based regulation. arXiv:2010.13389 [Preprint]. 2020. Available from: http://arxiv.org/abs/2010.13389.
27. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 [Preprint]. 2018. Available from: http://arxiv.org/abs/1810.04805.
28. Wang Y, Huang M, Zhu X, Zhao L. Attention-based LSTM for aspect-level sentiment classification. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, Texas: Association for Computational Linguistics; 2016. p. 606–15.
29. Hu M, Liu B. Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '04. New York, NY, USA: Association for Computing Machinery; 2004. p. 168-77.
30. Du C, Sun H, Wang J, Qi Q, Liao J, Xu T, Liu M. Capsule network with interactive attention for aspect-level sentiment classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China: Association for Computational Linguistics; 2019. p. 5489–98.
31. Xu G, Yu Z, Yao H, Li F, Meng Y, Xu W. Chinese text sentiment analysis based on extended sentiment dictionary. IEEE Access. 2019;7:43749–62.
32. Miao YQ, Luo R, Zhu L, Liu T, Zhang W, Cai G, Zhou M. Contextual graph attention network for aspect-level sentiment classification. Mathematics. 2022.
33. Almaghrabi M, Chetty G. Improving sentiment analysis in Arabic and English languages by using multi-layer perceptron model (mlp). In: 2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA). 2020. p. 745–6.
34. Zhang C, Li Q, Song D. Aspect-based sentiment classification with aspect-specific graph convolutional networks. arXiv:1909.03477 [Preprint]. 2019. Available from: http://arxiv.org/abs/1909.03477.

35. Zhang M, Qian T. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online. Association for Computational Linguistics; 2020. p. 3540–9.
36. Zhuang L, Jing F, Zhu XY. Movie review mining and summarization. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, CIKM '06. New York, NY, USA: Association for Computing Machinery; 2006. p. 43-50.
37. Fan Z, Wu Z, Dai X-Y, Huang S, Chen J. Target-oriented opinion words extraction with target-fused neural sequence labeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). Minneapolis, Minnesota: Association for Computational Linguistics; 2019. p. 2509–18.
38. Chouikhi H, Alsuhaibani MA, Jarray F. Bert-based joint model for aspect term extraction and aspect polarity detection in Arabic text. Electronics. 2023.
39. Tiwari A, Tewari K, Dawar S, Singh A, Rathee N. Comparative analysis on aspect-based sentiment using bert. In: 2023 7th International Conference on Computing Methodologies and Communication (ICCMC). 2023. p. 723–7.
40. Zhu YC, Li L, Li CB, Zhang W. Challenges confronting the sustainability of anti-epidemic policies based on the bert-pair-absa model. Oppor Challenge Sustain. 2023.
41. Jawahar G, Sagot B, Seddah D. What does BERT learn about the structure of language? In: ACL 2019–57th Annual Meeting of the Association for Computational Linguistics. Italy: Florence; 2019.
42. Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692 [Preprint]. 2019. Available from: http://arxiv.org/abs/1907.11692.
43. Tran MP, Nguyen MV, Nguyen TH. Fine-grained temporal relation extraction with ordered-neuron LSTM and graph convolutional networks. In: Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021), Online. Association for Computational Linguistics; 2021. p. 35–45.
44. Zhang Z, Li X, Li Y, Dong Y, Wang D, Xiong S. Neural noise embedding for end-to-end speech enhancement with conditional layer normalization. In: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2021. p. 7113–7.
45. Miller George A, Beckwith Richard, Fellbaum Christiane, Gross Derek, Miller Katherine J. Introduction to WordNet: An on-line Lexical database*. Int J Lexicograph. 1990;3(4):235–44.
46. Chen Z, Qian T. Transfer capsule network for aspect level sentiment classification. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics; 2019. p. 547–56.
47. Jiang Q, Chen L, Xu R, Ao X, Yang M. A challenge dataset and effective models for aspect-based sentiment analysis. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics; 2019. p. 6280–5.