Knowledge Guided Capsule Attention Network for Aspect-Based Sentiment Analysis

Bowen Zhang, Xutao Li, Xiaofei Xu, Ka-Cheong Leung, *Senior Member, IEEE*, Zhiyao Chen, and Yunming Ye

***Abstract*—Aspect-based (aspect-level) sentiment analysis is an important task in fine-grained sentiment analysis, which aims to automatically infer the sentiment towards an aspect in its context. Previous studies have shown that utilizing the attention-based method can effectively improve the accuracy of the aspect-based sentiment analysis. Despite the outstanding progress, aspect-based sentiment analysis in the real-world remains several challenges. (1) The current attention-based method may cause a given aspect to incorrectly focus on syntactically unrelated words. (2) Conventional methods fail to identify the sentiment with the special sentence structure, such as double negatives. (3) Most of the studies leverage only one vector to represent context and target. However, utilizing one vector to represent the sentence is limited, as the natural languages are delicate and complex. In this paper, we propose a knowledge guided capsule network (KGCapsAN), which can address the above deficiencies. Our method is composed of two parts, a Bi-LSTM network and a capsule attention network. The capsule attention network implements the routing method by attention mechanism. Moreover, we utilize two prior knowledge to guide the capsule attention process, which are syntactical and n-gram structures. Extensive experiments are conducted on six datasets, and the results show that the proposed method yields the state-of-the-art.**

***Index Terms*—Aspect-based sentiment analysis, attention mechanism, capsule attention network.**

I. Introduction

ASPECT-BASED sentiment analysis (ABSA)[1] is a fine-grained task in sentiment analysis. It aims to identify the sentiment polarity of the opinion targets in a sentence or document (e.g., negative, neutral, or positive) [1]. Most sentences or documents come from online posts, such as Amazon reviews or Twitter. ABSA has gained increasing popularity in recent years since it has a wide range of applications in the real world [2]. For example, it can help raise perspicacity on consumer needs or their product experience, guiding producers to improve their products.

Aspect-based sentiment analysis can be classified into two subtasks, namely, aspect-category sentiment analysis (ACSA) and aspect-term sentiment analysis (ATSA). ACSA aims to identify the sentiment polarity to a given aspect target, which is one of a few predefined categories, while the goal of ATSA is to predict the sentiment polarity of the aspect term that appears in the text, which can be the word or phrase (multi-word). For example, the sentence “The food price is reasonable although the service is poor” expresses the positive sentiment for the “food price” aspect, but it also conveys the negative sentiment for the “service” aspect. The number of distinct words used as the aspect terms could be more than a thousand, which poses more challenges. Here, we focus on ATSA in this paper.

Existing ATSA methods can be divided into two categories. Traditional approaches mainly leverage the statistical methods to classify the sentiment of the aspect through designing a set of hand-crafted features to train a classifier such as SVM [3]. However, the preparation of massive number of hand-crafted features is labor-intensive and cost expensive. Inspired by the recent performance breakthroughs of employing deep learning in natural language processing, the deep neural networks (say convolutional neural network (CNN) and recurrent neural network (RNN)) have become dominant in the literature [4], [5]. This is because such methods can automatically generate useful low-dimensional representations from both aspects and contexts so as to achieve remarkable results without careful feature engineering [6]. Recently, several studies attempt to deal with ATSA problem based on deep learning methods [7], [8]. Ever since Tang *et al.* [9] raised the challenge of modeling semantic relationships between aspect and context, researchers resorted to RNN models with attention mechanisms for ATSA, which can effectively identify more informative and relevant words to a given aspect in a sentence [10], [11].

Despite the effectiveness of previous studies, it is challenging to apply them in real-life applications: (1) The current attention mechanism may cause a given aspect to incorrectly focus on syntactically unrelated words. Take sentence “The food is delicious and the price is acceptable” as an example. Due to the lack of fine-grained attention mechanism, previous methods tend to take “acceptable” to incorrectly infer the sentiment of the

Manuscript received March 18, 2020; revised July 9, 2020; accepted August 3, 2020. Date of publication August 17, 2020; date of current version September 7, 2020. This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1402500, and in part by the National Science Foundation of China under Grants 61772155, 61802089, 61832004, 61832014, U1836107, 61972111, 61572158, and 61602132. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Taro Watanabe. *(Corresponding authors: Xiaofei Xu; Xutao Li.)*

Bowen Zhang and Xiaofei Xu are with the College of Computer Science and Technology, Harbin Institute of Technology, Harbin 150000, China (e-mail: zhang_bo_wen@foxmail.com; xiaofei@hit.edu.cn).

Xutao Li, Ka-Cheong Leung, Zhiyao Chen, and Yunming Ye are with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen 518055, China, and also with Shenzhen Key Laboratory of Internet Information Collaboration, Shenzhen 518055, China (e-mail: lixutao@hit.edu.cn; kcleung@ieee.org; dyleaf@foxmail.com; yeyunming@hit.edu.cn).

Digital Object Identifier 10.1109/TASLP.2020.3017093

[1] also called aspect-level sentiment analysis.

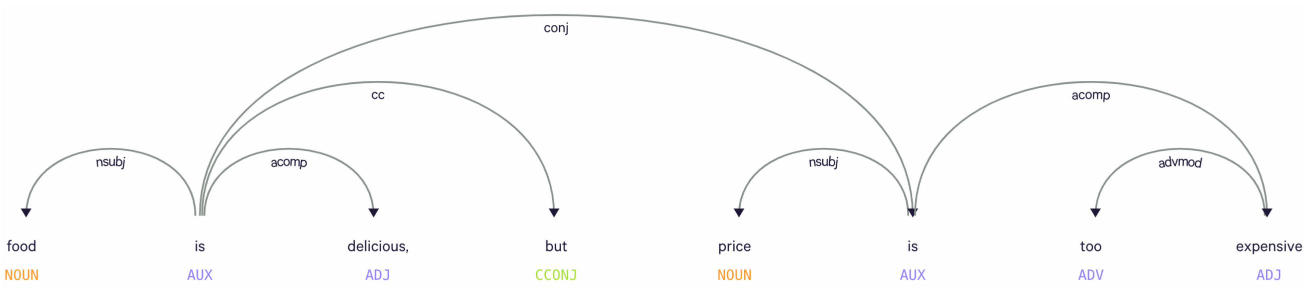


Fig. 1. An example for the illustration of the syntactical dependency.

aspect "food," which indeed is related to the "price" aspect. (2) Conventional methods fail to tackle the sentences with special structures nicely. For example, "the food is not very good". Here, "not very good" should be considered as the whole phrase which implies a negative sentiment, but existing attention mechanisms fail to model such syntactic structures. (3) Existing models rely heavily on the quality of instance representation. Conventional methods mainly represent the context and aspect with a vector. For example, some studies leverage the aspect term as a query to employ the attention method with context to acquire the representation vector [10], [11]. However, utilizing one vector to represent the instance is limited, as the natural languages are delicate and complex.

In this paper, we propose a knowledge guided capsule attention network (KGCapsAN) for ATSA. Our model is motivated by the fact that prior knowledge, such as syntactic knowledge, can help identify the syntactically related words to the aspect and understand the special sentence structure. In Figure 1, we observe that syntactical knowledge can help identify the sentiment related words towards the aspect. In KGCapsAN, we first propose a Bi-LSTM network to model the text. Moreover, we develop a capsule attention network (CAN) to enhance the sentence and aspect representation. Capsule Network was first proposed by [12] and good at modeling the part-whole relationships between the lower-level capsules and higher-level capsules. CapsNet transfers the information by utilizing the dynamic routing mechanism, which updates the coupling coefficients between capsules in lower and upper layers through iterations. It aims to extend the multi-hop attention mechanism [5] by controlling the number of self-loops to achieve multi-step attention in the single attention layer.

CAN draws on the idea of dynamic routing and treats output (matrix) of the hidden layer obtained by Bi-LSTM as lower-level capsules. The attention output of CAN is regarded as a higher-layer capsule obtained through dynamic interactions with these lower-layer capsules. CAN extend the conventional multi-hop attention mechanisms by incorporating high-level information (such as syntactic and sentence structures) as the attention query to guide the attention process and improve the performance of ATSA. Specifically, the **first** query aims to utilize the syntactic knowledge. We first feed each sentence to the syntactical dependency tree to acquire the syntactic relationships. Then, we build a small graph specific to the sentence, where each word is a node and an edge between each node pair indicates the syntactic relationship (such as $0/1$ for indicating the existence of the syntactic relationship). Afterwards, a graph-based convolutional network (GCN) [13] is employed to learn the graph representation, which is an effective graph-based neural network that captures the high order neighborhood information to achieve the graph representation so as to capture the syntactically relevant words. The **second** query is designed to capture the special sentiment carrying phrases (also known as n-grams in natural language processing), say, "not bad" or "should be". To this end, we develop a CNN-based local n-gram layer, which can utilize the informative words (1-gram) or phrases (n-gram) as the second query to guide the attention mechanism. Finally, we propose a CapsAttention network, which simulates the information transfer of dynamic routing by designing multiple knowledge guided attention mechanisms.

The main contributions of this work can be summarized as follows:

- We propose KGCapsAN,[2] a novel framework for ATSA, which simulates the capsule network by utilizing the attention mechanism. KGCapsAN makes use of multi-queries to guide the attention process, and provides the output capsule more information which effectively improves the sentiment classification.
- We propose a multi-knowledge guided capsule attention network, which can dynamically adjust the information transfer of different prior knowledge.
- We collect a unique dataset (SpATSA) for special sentence structures, such as the conditional statement and subjunctive ATSA and release it at http://dwz1.cc/5AWF8Mg.
- To evaluate the effectiveness of our approach, we conduct extensive experiments on five widely used datasets. The experimental results show that our proposed CAN model can make better use of syntactic information to enhance text representation. This hence allows the model to adapt to a more complex sentence structure for ATSA. The results also demonstrate that our model achieves the state-of-the-art results.

II. Related Work

A. Aspect-Level Sentiment Analysis

Previous studies [14]–[16] on sentiment classification have achieved remarkable results at the sentence or document level.

[2] [Online]. Available: http://dwz1.cc/f1zndpU.

However, the methods only produce the sentiment classification on the whole text, which is aspect independent.

Recently, ABSA draws more attention and many methods are developed, which can be classified into conventional machine learning methods and the neural network based methods [17]. The traditional machine learning methods focus on extracting a set of handcraft features like sentiment lexicons to train a sentiment statistic-based classifier [3]. However, such methods rely heavily on hand-crafted features that are labor-intensive and costly.

Driven by the remarkable progress of attention-based deep neural networks, many studies sentiment classifiers of the type have been developed. For example, Tang *et al.* [9] developed a memory network to learn the weights of the context words by utilizing the multi-hop attention mechanism and use the weighted sums to compute the aspect-specific textual representations. Tang *et al.* [6] proposed TD-LSTM to extend the standard structure by using two separate LSTMs to model both left context and right contexts of the target word, respectively. Li *et al.* [18] utilized the hierarchical attention network to identify the informative sentiment words towards the target to guide the classifier. Ma *et al.* [11] proposed IAN to learn the representations of the target and context with two attention networks interactively.

B. Attention-Based Capsule Networks

Capsule network (CapsNet) was first proposed by Hinton *et al.* [19], which has introduced the concept of "capsules" with transformation matrices to let networks learn part-whole relationships automatically. Subsequently, Sabour *et al.* [12] proposed a routing-based method for the capsule network. Each capsule is an aggregation of neurons that represent various attributes of a particular feature. These attributes represent different instantiation parameters, such as relative position. Thus, the capsule network has much stronger text representation capability than conventional deep neural networks. Further studies [20], [21] have extended the routing-based capsule network for the natural language processing applications.

The dynamic routing method is similar to the multi-hop attention method since the lower-level capsules are aggregated to the upper-level through self-iterative coupling coefficient updating. To enhance the operation speed and parallelism capability, some studies extend the dynamic routing based capsule method by using the attention mechanism. Zhou *et al.* [22] introduced a capsule-based attention method for visual question answering task and has achieved remarkable results. Capsule attention utilizes the multi-hop attention mechanism and denotes the attention weight as the coupling coefficient. Wang *et al.* [15] proposed an RNN-based capsule network for sentence-level sentiment analysis. Given a hidden vector encoded by a standard RNN as the attention query, the capsule representation can be acquired by the typical attention mechanism. In [23], an aspect-target level capsule model was devised, which integrates the target information into the single capsule cell and achieves significant progress. Yang *et al.* [24] developed a query-guided capsule network in order to integrate the capsule routing mechanism into the multi-head attention structure for a significant performance improvement in terms of sentiment identification accuracy.

C. Graph Neural Networks

Graph neural networks have received growing interest in NLP tasks recently [25].

With the development of deep learning method, many research studies have extended the deep neural network structure that can be used for the arbitrarily structured graph. Among them, Kipf and Welling [26] proposed a graph convolutional network (GCN). This yields the remarkable results on many benchmark datasets. Subsequently, many other studies extended GCN to various tasks such as machine translation and text classification. Recent studies have explored graph neural networks for textual classification. For example, a graph-CNN method was proposed in [27] to convert text to graph structure which can capture non-consecutive and long-distance semantics. In [28], a heterogeneous graph was constructed by representing documents and words as nodes and then uses the GCN for classification. This approach does not require inter-document relationships, but it can achieve state-of-the-art results for textual classification.

D. Incorporating External Knowledge

Recently, many studies incorporated external resources (such as logic reasoning rules, grammar knowledge and sentiment lexicons, etc.) into deep learning framework to address ATSA [29]–[32].

Among them, the method of fusing syntactical knowledge into deep learning based methods has received extensive attention. For example, Zhang *et al.* [33] proposed C-GCN that first introduced dependency trees knowledge into the neural network. The main structure of C-GCN is a multi-layer GCN, where the input is the embedding vectors of words in a sentence. For GCN layers, the adjacency matrix is constructed based on the syntactical tree structure. Upon the hidden representation from the GCN layers, a softmax layer is appended to deliver the classification result. Subsequently, Zhang *et al.* [34] improve the C-GCN structure by adding an LSTM layer in between the input embedding layer and the first GCN layer. As a result, the word sequential information in the sentence can be exploited. Zuo *et al.* [35] focused on optimizing the syntactic tree structure with heuristic rules, and then employed the GCNs to solve the ATSA.

Another typical method is to incorporate the locality knowledge. In [36], it has been proved that such knowledge is very effective for ASTA. For example, Wei *et al.* [37] proposed to extract the local structures, e.g., k-hop sub-tree, for sentimental analysis. Hu *et al.* [38] proposed to integrate the locality information with first-order logic. With a carefully-designed deep learning network, the logic rules can be effectively combined. Zeng *et al.* [36] developed a locality weighting scheme to leverage the word context around the aspects.

However, the above methods focus either on incorporating syntactical knowledge or exploiting the locality knowledge. None of the studies both simultaneously. In this paper, we propose a CAN method, which can flexibly and effectively integrate

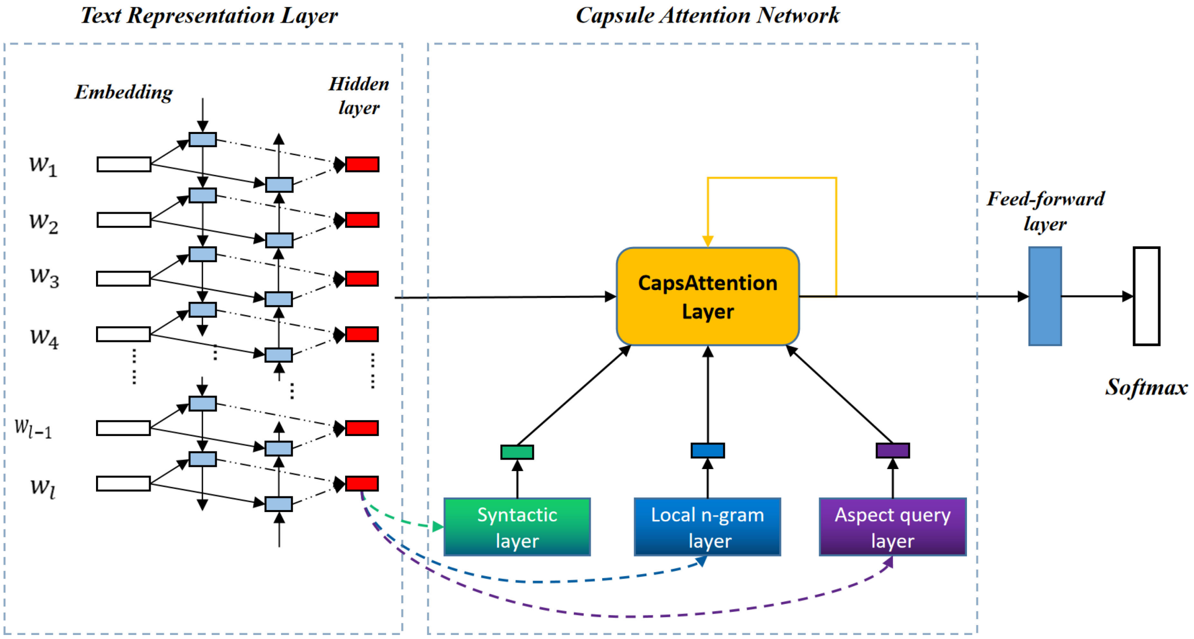


Fig. 2. The overall framework of knowledge guided capsule attention network (KGCapsAN) for aspect-level sentiment classification.

TABLE I
NOTATIONS USED IN THIS PAPER

Notations	Description
x	each input sentence
w	each word in the sentence
w^c,w^a	context word, aspect term word
y	sentiment label
$\overrightarrow{h}$, $\overleftarrow{h}$	forward hidden state, backward hidden state
$\overrightarrow{c}$, $\overleftarrow{c}$	forward memory, backward memory
H	the hidden states of context representation
S_{mask}	output of the syntactic layer
Z_{mask}	output of local n-gram layer
H_{mask}	output of aspect query layer
M	output capsule

different types of prior knowledge with a carefully-designed multi-knowledge guided capsule attention mechanism. Specifically, we leverage the mechanism to exploit the syntactical, locality and lexicon knowledge simultaneously.

III. KGCapsAN Model

KGCapsAN aims to address the deficiencies of the conventional attention-based approach in ATSA. Among them, the Capsule Attention Network (CAN) is a core component of KGCapsAN, which implements the dynamic routing process of the CapsNet structure with a capsule-based attention mechanism. Specifically, CAN uses syntactic knowledge and n-gram information as the query to guide the attention, and then integrates such knowledge with the representation vector to enhance the capabilities of the representation.

KGCapsAN, depicted in Figure 2, consists of two components, namely, a Bi-LSTM Network and a Capsule Attention Network, to improve the performance of ATSA. We are going to give the task definition and the overview of our model in Section III.A and Section III.B, respectively. Then, we describe the details of the Bi-LSTM network and CAN in Section III.C and III.D, respectively. Finally, the training process is discussed in Section III.E.

A. Problem Definition

The ATSA task can be formulated as follows. Given a sentence $x = \{w^c_1, \ldots, w^a_\tau, \ldots, w^a_{\tau+m}, \ldots, w^c_n\}$ contains a corresponding aspect-term words $w^a_\tau, \ldots, w^a_{\tau+m}$, where w denotes the each word in the sentence and m denotes the aspect term length. Each sentence has a sentiment label y. ATSA aims to predict a sentiment label for the input sentence x towards the given aspect term. In this paper, we use superscripts “c,” “a” to indicate a context word and aspect-term word, respectively. The notations used in this paper are summarized in Table 1 for clarity.

B. Framework Overview

As shown in Figure 2, KGCapsAN consists of two main components: *Text Representation Layer* and *Capsule Attention Network (CAN)*. The text representation layer employs a *Vanilla* Bi-LSTM structure trained with textual features. It contains an embedding layer and a Bi-LSTM layer for capturing sequential features of the text. CAN contains four layers. The first layer is the **syntactic layer**, which uses the syntactic graph constructed through the use of the syntactical dependency trees to acquire the syntactic query. The second layer is the **local n-gram layer**, which uses CNN to capture informative n-gram features. The third layer is the **aspect query layer**, which utilizes the aspect term to learn the aspect-specific information of the whole sentence. In CAN, all three layers are denoted as the attention query of the CapsAttention layer, which can effectively guide the attention.

C. Vanilla Bi-LSTM Network

Generally, Vanilla Bi-LSTM networks are employed to encode the input sentence x. Bi-LSTM can capture the left and right contexts of each word in the input. In particular, for the t-th word w_t in the input sequence of the target, we first convert the t-th word into the word embedding layer E to acquire the word embedding representation e_t. Here, the sentence representation can be denoted as $E(x)$. Then, we feed $E(x)$ into Bi-LSTM to compute its forward hidden state $\overrightarrow{h}_t^p$ and backward hidden state $\overleftarrow{h}_t^p$:

$$[\overrightarrow{h}_1, \overrightarrow{c}_1] = \overrightarrow{LSTM}(e_1, \overrightarrow{h}^P, \overrightarrow{c}^P)$$
$$[\overleftarrow{h}_2, \overleftarrow{c}_2] = \overleftarrow{LSTM}(e_n, \overleftarrow{h}^P, \overleftarrow{c}^P) \quad (1)$$

where we concatenate both the forward and backward hidden states to form the final hidden state $h_t = [\overrightarrow{h}_t \oplus \overleftarrow{h}_t]$ for the word w_t at the t-th position of the input target. $h \in \mathbb{R}^d$ and $c \in \mathbb{R}^d$ denote the hidden state and state of the cell in LSTM, respectively. The symbols → and ← represent the forward or backwards path directions. Thus, the final representation of the Bi-LSTM layer can be denoted as $H = \{h_1, h_2, \ldots, h_n\}$.

Here, LSTM cell has three gates: an input gate i_t, a forget gate f_t, an output gate o_t, and a memory cell c_t. Formally, the current hidden state h_t in the LSTM networks are computed as follows:

$$\begin{aligned} i_t^T &= \sigma(W_i^T e_t + U_i^T h_{t-1} + V_i^T c_{t-1}) \\ f_t^T &= \sigma(W_f^T e_t + U_f^T h_{t-1} + V_f^T c_{t-1}) \\ o_t^T &= \sigma(W_o^T e_t + U_o^T h_{t-1} + V_o^T c_{t-1}) \\ \tilde{c}_t^T &= tanh(W_c^T p_t + U_c^T h_{t-1}) \\ c_t^T &= f_t^T \odot c_{t-1}^T + i_t^T \odot \tilde{c}_t^T \\ h_t^T &= o_t^T \odot tanh(c_t^T) \end{aligned} \quad (2)$$

where $W_{\{i,f,o,c\}}^T$, $U_{\{i,f,o,c\}}^T$, and $V_{\{i,f,o,c\}}^T$ are the set of all trainable parameters to be learned, σ denotes the sigmiod function, and $\odot$ represents the element-wise multiplication.

D. Capsule Attention Network

The traditional capsule network was proposed to capture the part-whole relationships in the iterative routing procedure. The capsules in the lower layer are transferred to the higher layer by aggregating their transformations with iteratively updated coupling coefficients. Each capsule is an aggregation of neurons, where each neuron indicates multiple attributes of the special feature present in the text. These attributes can be kinds of instantiation parameters, such as the syntactical relationship between a word and its position in a sentence.

Nevertheless, there are two drawbacks of directly employing such capsule network in ATSA. First, the capsule network cannot focus on the aspect-specific words while inferring the sentiment. Second, the original dynamic routing mechanism is independent of the back propagation stage, which makes it time-consuming and cannot be parallelized.

To alleviate the aforementioned issue, we propose CAN, which makes use of the attention mechanism for realizing the capsule structure. It is reasonable to utilize that capsule-based structure to represent the sentence, since it can obtain more information instead of using only one vector as for the conventional attention-based methods. CAN is developed based on two features: 1) The use of syntactic information can effectively address the problem of incorrectly focusing on syntactically unrelated words in a short or long range. 2) The enhancement of the learning ability of n-gram can help the model to accurately understand complex structures, such as "not bad" can be considered as a whole.

Next, we will introduce each component respectively.

1) Syntactic Layer: The syntactic layer learns syntactically relevant words towards the target aspect through the dependency tree,[3] which is wildly used in the NLP task and can effectively identify the relationships between words. Given a sentence x, we first build the syntactic graph (S-Graph) to model the syntactical relationships about this sentence. The S-graph utilizes the words as nodes. It constructs the weighted edges based on the syntactical relationships. We denote A as the adjacency matrix of S-Graph.

After obtaining the hidden state of the sentence $H \in \mathbb{R}^{n \times d}$, we feed them into a two-layer GCN. The graph representation $\hat{S}^l \in \mathbb{R}^{n \times k}$ can be calculated as:

$$\hat{S}^l = \sigma(A\sigma(AHW_0))W_1^l) \quad (3)$$

where σ represents a non-linear function, and $W_0 \in \mathbb{R}^{d*v}$ and $W_1^l \in \mathbb{R}^{d*k}$ are trainable parameters. Note that, similar to CNN, we use l different weights W_1^l to capture multiple features. Here, $\hat{S}^l$ denotes l-th graph representation.

Aspect-Specific Zero Masking: GCN draws syntactically related words into an aspect term to achieve the syntactical aspect-specific representation. In this layer, we introduce to use the aspect-specific zero masking mechanism, which aims to mask out the graph representation vectors of non-aspect term words and keep merely high-level aspect-specific features.

Formally, given the l-th graph representation $\hat{S}^l = \{s_1^l, \ldots, s_\tau^l, \ldots, s_{\tau+m}^l, \ldots, s_n^l\}$, the output of a mask can be denoted as: $\hat{S}_{mask}^l = \{0,...,s_\tau^l,...,s_{\tau+m}^l,..., 0\}$. After acquiring all $\hat{S}_{mask}$, we compute the weighted sum of all graph representations to achieve the output of the syntactic layer S_{mask}. Note that the dimension of S_{mask} is the same as H. As shown in Figure 3, white blocks are employed to indicate the zero masked hidden states.

2) Local n-Gram Layer: For ATSA, it is important for the network to learn the sentiment-carry n-gram features, such as "not bad". Thus, we develop an n-gram layer to enhance the learning ability of the n-gram features. The n-gram layer consists of two convolutional layers to extract n-gram features of the input sequence through the convolutional operations.

As the two convolutional layers share a similar structure, we only give the details for one convolutional layer. Let $W \in \mathbb{R}^{k \times d}$ be the convolutions filters, where k is the filter width. A filter

[3] We use spaCy toolkit: [Online]. Available: https://spacy.io/.

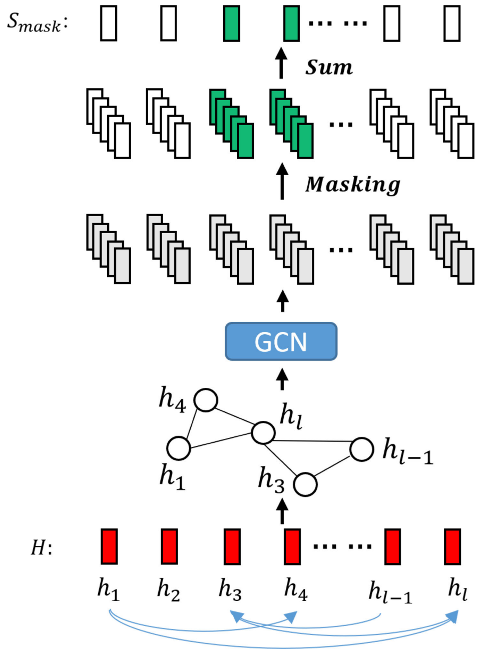


Fig. 3. The structure of the syntactic layer.

of width k allows the convolutional layer to slide over the input sequence and acquire new features. We denote z_i as the new feature obtained from a local window of the word sequence $e_{i:i+k-1}$, which can be computed as:

$$z_i = \sigma(W \odot e_{i:i+k-1} + b) \tag{4}$$

where $\odot$ denotes a convolutional operator, σ is a non-linear function and b is the trainable parameter. This convolution filter is applied to every possible window of words in the input sequence $\{e_{1:k}, e_{2:k+1}, \ldots, e_{m-k+1:m}\}$ so as to produce a feature map $\mathbf{z} \in \mathbb{R}^{n-k+1}$ as exhibited below:

$$\gamma = [z_1, \ldots, z_{m-k+1}] \tag{5}$$

Here, the filter weights and bias terms of each filter are shared between all positions in the input, thereby preserving spatial locality. Finally, we send γ to the second convolutional layer and we can obtain the convolutional representation $\mathbf{Z}$.

We observe that, for ATSA, the sentiment-carry words towards the aspect has the locality properties, where such words appear in a small range around the aspect term. To obtain the important local n-gram filters, we utilize the **aspect-specific zero masking** to select the k-range words. This is because the two-layer convolutional operations can express the n-gram information over an area of size $2k$ into the representation vector of the target aspect. The computational details are illustrated in Figure 4. Finally, the masked hidden vectors are denoted as $\mathbf{Z}_{mask}$.

3) Aspect Query Layer: This layer aims to learn the aspect-specific queries for capsule attention. To better embed the aspect-specific queries into the CapsAttention layer, we also utilize the aspect-specific zero masking to fit the dimension size. Formally, we send H into the masking layer, and the output of the aspect query layer can be represented as $H_{mask} = \{0, \ldots, h_\tau, \ldots, h_{\tau+m}, \ldots, 0\}$.

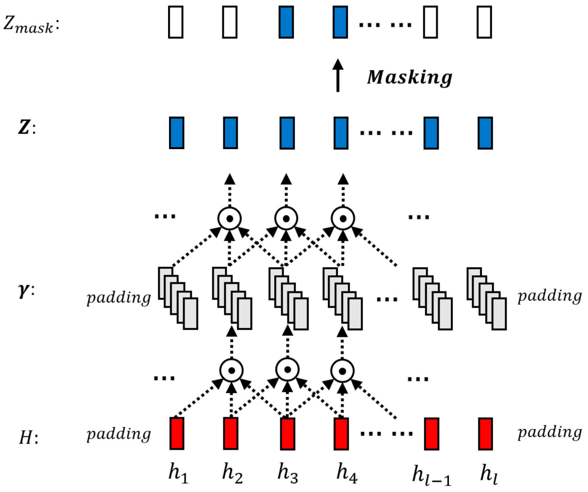


Fig. 4. The Structure of the local n-gram layer.

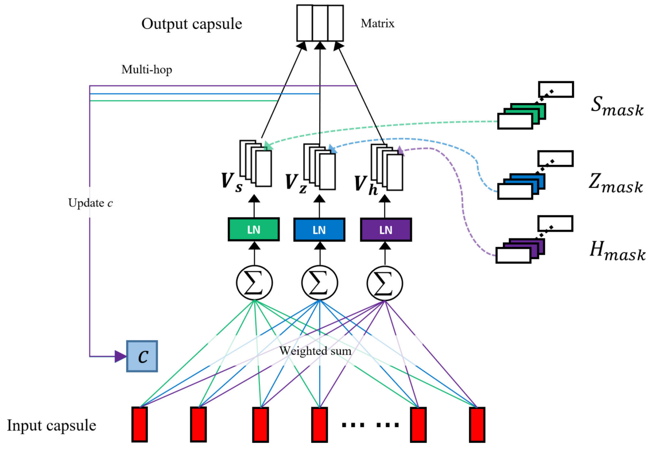


Fig. 5. Structure of the CapsAttention layer.

4) CapsAttention Layer: To achieve the dynamic routing with the attention mechanism, we propose an iterative attention algorithm, known as the CapsAttention layer. Figure 5 gives an example. CapsAttention layer treats each vector in the hidden state of text representation layer as input layer capsules, and the attention output is the output capsule that contains information related to the prediction. Here, the output capsule has of three vectors which can be represented in the matrix form.

In the CapsAttention layer, we utilize three queries to guide the attention in the iterative manner. Specifically, in the first iteration, we initialize the three queries V_s^1, V_z^1, V_h^1 as S_{mask}, Z_{mask} and, H_{mask}, respectively. Given the input capsules $H \in \mathbb{R}^{n \times d}$, the coupling coefficient matrix c can be computed as:

$$\begin{aligned} c_s^1 &= V_s^1 (H_s^1)^{\mathrm{T}} \\ c_z^1 &= V_z^1 (H_z^1)^{\mathrm{T}} \\ c_h^1 &= V_h^1 (H_h^1)^{\mathrm{T}} \end{aligned} \tag{6}$$

where $c_{\{s,z,h\}} \in \mathbb{R}^{n \times n}$. In the first iteration, input capsules are the same for the three attention queries, where $H = H_s^1 =$

Algorithm 1: Knowledge Guided Capsule Attention.

```
Input: $H$, $S_{mask}$, $Z_{mask}$, $H_{mask}$
Output: output capsule $M$
Initialize $V_{\{s,z,h\}}$
for $t$ in $T$ iterations do
    Obtain coupling coefficients: $c^{t-1}_{\{s,z,h\}}$
    Update queries: $V^{t}_{\{s,z,h\}}$
    Update input capsules: $H^{t}_{\{s,z,h\}}$
end for
Obtain the weighted sum feature: $q^{t}_{\{s,z,h\}}$
Obtain the output capsules: $M$
return $M$
```

$H_z^1 = H_h^1$. The next layer $V^2_{\{s,z,h\}}$ can be updated as:

$$\begin{aligned} V_s^2 &= c_s^1 H_s^1 \\ V_z^2 &= c_z^1 H_z^1 \\ V_h^2 &= c_h^1 H_h^1, \end{aligned} \tag{7}$$

where the dimension of the new query $V^2_{\{s,z,h\}}$ is the same as that of the initial query $V^1_{\{s,z,h\}}$. The input capsules of the next iteration can be updated by:

$$\begin{aligned} H_s^2 &= \lambda\ LayerNorm(\sigma(V_s^2)) + H_s^1 \\ H_z^2 &= \lambda\ LayerNorm(\sigma(V_z^2)) + H_s^1 \\ H_h^2 &= \lambda\ LayerNorm(\sigma(V_h^2)) + H_s^1, \end{aligned} \tag{8}$$

where $LayerNorm$ performs the standard layer normalization [39]. Note that, in CapsNet the "squash" activation function is utilized [12]. In this paper, we adopt the widely used non-linear sigmoid function, σ, instead of "squash".

After t iterations, the output capsule M can be found as:

$$\begin{aligned} \mathbf{q}_s &= H_s^t(softmax \sum_i c^t_{s,i}), \\ \mathbf{q}_z &= H_z^t(softmax \sum_i c^t_{z,i}), \\ \mathbf{q}_h &= H_h^t(softmax \sum_i c^t_{h,i}), \\ M &= \{\mathbf{q}_s \oplus \mathbf{q}_z \oplus \mathbf{q}_h\} \end{aligned} \tag{9}$$

where $softmax(f_\gamma) = \frac{e^{f_\gamma}}{\sum_\delta e^{f_\delta}}$, $\oplus$ is the concatenation operator, c_i denotes the i-th vector of the coupling coefficient matrix c, and $M \in \mathbb{R}^{1\times 3d}$.

Compared to the forward propagation algorithm of CapsNet, we replace the dynamic routing by the attention-based mechanism. The updating strategy is kept such that the lower-level capsules are transferred to the higher-level capsules by updating the coupling coefficients. The gradients of CAN can be computed by the standard back-propagation algorithm. The detailed process is presented in Algorithm 1.

E. Sentiment Classification

After acquiring the representation M, it is fed into the feed-forward layer and then the softmax layer to obtain the sentiment probability distribution:

$$P = softmax(WM + b) \tag{10}$$

where W and b are trainable parameters.

Finally, the model parameters are trained to minimize the following loss function:

$$Loss = -\sum^{N} logJ_\beta + \alpha||\theta|| \tag{11}$$

where β is the label, N is the training size, J_β is the β-th element of J, θ denotes the trainable parameter matrix and α represents the coefficient of L2-regularization.

F. Two Variants

It is worth pointing out that the proposed KGCapsAN is a very general framework, which supports flexible variants. For example, in the text representation layer, we can replace the BiLSTM with a pre-trained BERT. In this case, the model takes "[CLS] + sentence + [SEP] + aspect + [SEP]" as input. As a result, KGCapsAN is able to exploit the extra knowledge delivered by BERT. We refer to the variant as KGCapsAN-BERT. Also, the CAN allows incorporating more prior knowledge. For example, word emotional features from lexicon can be exploited as the fourth query. There are many ways to obtain and encode the emotional feature. Here we use the SenticNet [40] to obtain the emotional-related terms for each word, and calculate the average on their embeddings as query input. We name the variants as KGCapsAN-LI.

IV. Experiments

A. Datasets

To compare and evaluate the effectiveness of our proposed method and the existing approaches, we have conducted extensive experiments on five datasets.

- **Twitter corpus**. Twitter dataset is originally built via Twitter[4] [41]. Each sentence contains several aspect terms. Each aspect term is assigned with a sentiment label drawn from "positive," "neutral," or "negative". Twitter corpus includes 1561 positives, 3127 neutrals, and 1560 negatives for training. The test data contains 692 tweets.
- **Lap14** and **Rest14**. Lap14 and Rest14 datasets are taken from SemEval-14 Task 4 in [42], respectively. Lap14 comprises of the laptop reviews, and it is composed of 2328 training samples of 3 sentiment classes (where there are 994 positives, 464 neutrals, and 870 negatives). Its test set contains 638 samples. Rest14 composes of the restaurant reviews, where there are 2164 positives, 637 neutrals, and 807 negatives. Its test set has 1120 samples.

[4] [Online]. Available: https://twitter.com/

TABLE II
STATISTICS OF THE DATASETS

Dataset		Positive	Neutral	Negative
Twitter	Train	1561	3127	1560
	Test	173	346	173
Lap14	Train	994	464	870
	Test	341	169	128
Rest15	Train	912	36	256
	Test	326	34	182
Rest14	Train	2164	637	807
	Test	728	196	196
Rest16	Train	1240	69	439
	Test	469	30	117
SpATSA	Train	1792	1559	1375
	Test	433	412	337

- **Rest15**. Rest15 is collected from SemEval 2015 task 12 in [43], it contains in total 1204 training samples with three sentiment classes and 542 samples for test.
- **Rest16**. Rest16 is taken from SemEval 2016 task 5 in [44]. The data set includes 1748 training samples of three classes, 1240 positives, 69 neutrals, and 439 negatives. Its test set has 616 samples.
- **SpATSA**. Many previous studies have revealed that existing methods fail to adequately predict the sentences with special structures. To examine the performance in such case, we construct an additional dataset by selecting the sentence-aspect pairs from the above four datasets. Here, each sentence contains the special sentence structure, e.g., conditional statement and subjunctive ATSA, etc. For example, "the staff should be a bit more friendly." All the sentences in the dataset contain special sentence structures, and we refer to the data set SpATSA. In SpATSA, there are 4726 training samples with 1792 positive, 1559 neutral and 1375 negative instances. The test set includes 1182 samples.

The statistical information of the five data sets is summarized in Table II. As in [1], [9], [11], [34], [45], we discard the "conflict sentences," each of which has one aspect term labeled with multiple sentiments.[5]

B. Ablation Study

C. Baselines and Experimental Setting

As for a comparison, we adopt thirteen sentiment classification methods as baselines, which can be categorized into three groups: attention-based methods, capsule-based methods and BERT-based methods.[6]

a) Attention-based methods:

- **SVM** [46]: SVM is an effective traditional mechine learning based method for sentiment analysis. Kiritchenko *et al.* [46] devised SVM to solve SemEval 2014 Task 4.
- **LSTM** [6]: This method utilizes the standard LSTM to model the sentiment representation. The last hidden state is forwarded into the softmax layer to obtain the sentiment probability. The method uses Adam Optimizer with a learning rate of 0.001. The dimension of embeddings is 300.
- **IAN** [11]: This method learns the representations of the target and context with two LSTMs, and then utilizes the interactive attention to model the relationship with respect to each other. In the experiment, the hidden size is 300 and learning rate is 0.001.
- **MemNet** [9]: MemNet utilizes the memory network, which is widely used in aspect-level sentiment classification. This benefits from the multi-hop attention mechanism for sentiment inference. The hidden states dimension is 300, the hop for memory is 3, and the learning rate is 0.001.
- **AOA** [1]: The attention-over-attention (AOA) method models the aspects and sentences jointly, and it explicitly captures the interaction between aspects and contexts. Here, the hyper-parameters setting in our experiment are the same with IAN.
- **TNet-LF**[6] [45]: TNet-LF formulates a transformer-based method, known as context preserving transformation, to increase the informative element of contexts. For TNet-LF the hop of the attention block is set to 2, the number of convolutional filters are set to 50, and the learning rate is 0.001.
- **ASGCN**[7] [34]: ASGCN utilizes the external dependency tree to model long-range word dependencies, by making use of GCN to model the dependency tree graph. ASGCN-DT uses the directed graphs while ASGCN-DG employs the unidirectional ones. In the expertment, we utilize spaCy to access the dependency tree, learning rate is 0.001, dimension of hidden size is 300 and the number of GCN is set to 2.

b) Capsule-based methods:

- **TransCap**[8] [47]: TransCap makes use of a dynamic routing based capsule network for aspect-based sentiment analysis. The results on Rest14 and Lap14 are retrieved from [47]. For the reminder data sets, the parameter settings are as follows. The routing iteration in our experiment is set to 3, the learning rate is set to 0.01, and the slack parameter for loss is set to 0.8.
- **RNN-Cap**[9] [15]: RNN-Cap proposes the attention-based capsule structure for aspect-based sentiment analysis, where the hidden states of RNN denote as the lower capsules. Note that the original RNN-Cap was developed for sentence-level sentiment analysis, followed by [23], we replace the capsule query by the embedding of aspect term. The capsule block is set to 2, and learning rate is 0.01, the hidden dimension is 512 with the weight decay rate as 0.0001.

c) BERT-based methods:

- **BERT** [48]: This method fine-tunes from a pre-trained BERT model to perform ATSA. Following [49], we convert the given context and target to "[CLS] + sentence + [SEP]

[5] [Online]. Available: https://github.com/ganeshjawahar/mem_absa.

[6] [Online]. Available: https://github.com/songyouwei/ABSA-PyTorch.

[7] [Online]. Available: https://github.com/GeneZC/ASGCN.

[8] [Online]. Available: https://github.com/NLPWM-WHU/TransCap

[9] [Online]. Available: http://www.wangyequan.com/publications/

TABLE III
EVALUATION RESULTS ON ALL DATASETS. THE BEST RESULT ON EACH DATASET IS IN BOLD. ∗ MARKS RESULTS REPORTED IN THE ORIGINAL PAPERS, † MARKS RESULTS PRODUCED BY USING THE OPEN IMPLEMENTATION AND THE NUMBER IN PARENTHESIS REPRESENTS THE VARIANCE

Model	Twitter (%)		Lap14 (%)		Rest14 (%)		Rest15 (%)		Rest16 (%)		SpATSA (%)	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
SVM*	63.40	63.30	70.49	-	80.16	-	-	-	-	-	-	-
LSTM	69.56	67.70	69.28	63.09	78.13	67.47	77.37	55.17	86.80	63.88	-	-
MemNet	71.48	69.90	70.64	65.17	79.61	69.64	77.31	58.28	85.44	65.99	64.21	63.99
AOA	72.30	70.20	72.62	67.52	79.97	70.42	78.17	57.02	87.50	66.21	66.61	66.56
IAN	72.50	70.81	72.05	67.38	79.26	70.09	78.54	52.65	84.74	55.21	-	-
TNet-LF†	72.98	71.43	74.61	70.14	80.42	71.03	78.47	59.47	**89.07**	70.43	68.70	68.46
ASGCN†	72.15	70.40	75.55	71.05	80.86	72.19	79.89	61.89	88.99	67.48	70.45	70.15
RNN-CAP†	72.41	70.91	73.74	70.13	79.51	71.37	78.41	58.64	86.99	67.78	69.93	69.47
TransCap†	63.87	63.69	73.87	70.10	79.55	71.41	80.44	64.36	87.01	67.83	72.25	72.17
KGCapsAN	**74.13** (±0.83)	**72.52** (±0.88)	**76.96** (±1.11)	**72.89** (±1.39)	**82.05** (±0.78)	**74.04** (±1.72)	**81.86** (±0.85)	**65.60** (±0.25)	88.47 (±0.56)	**70.72** (±1.34)	**73.18**	**72.91**
-KGCapsAN-LI	**74.57**	**72.74**	**77.02**	**72.97**	**82.49**	**74.21**	-	-	-	-	-	-

+ aspect+ [SEP]" structure. The learning rate is 2e-5, the dropout rate is 0.1 and the dimension of BERT is 768.

- **BERT-PT** [50]: The BERT-PT method proposes a novel post-training strategy for the basic BERT model, which can effectively increase the performance for ATSA task. The input structure is the same with "[CLS] + sentence + [SEP] + aspect+ [SEP]," the learning rate is 3e-5, and max length of the post-training is set to 320, the dimension of BERT is 768.
- **LCF-BERT**[6] [36]: LCF-BERT is a method based on BERT fine-tuning. It contains a local context focus (LCF) mechanism method to learn local features from contexts. The hyper-parameter setting is the same as **BERT**. Here, for LCF, the input structure is "[CLS] + sentence + [SEP]".
- **AEN-BERT**[6] [49]: This method incorporates the attentional encoder network into the BERT framework, which can help the BERT-based method to learn target-specific words. The hyper-parameter setting is same as **BERT**.

In all the experiments, the parameters initialized with Xavier uniform [51], we utilize 300-dimensional pre-trained GloVe vectors to initialize the word embeddings and the the optimizer we use is Adam. Other weight parameters are initialized by randomly sampling the values from the uniform distribution $U(-0.01, 0.01)$. The dimensions of the hidden state of Bi-LSTM is 300, and the kernel size for the Local n-gram layer is 3. The scale weight λ is set with 0.1, 0.01 and 0.001, α is set to 0.00001. The model is optimized with the Adam optimization algorithm with the batch size of 32 and the learning rate is 0.001.

As in [34], we use accuracy and Macro-Averaged F1 as the evaluation metrics. We compute the metrics independently for each class and then take the average (hence treating all classes equally), as the final performance.

D. Experimental Results

In the experiments, we apply sentiment classification accuracy and F1 score as evaluation metrics to evaluate our method. To evaluate the stability of the model, following [34], we run the method three times and reported the mean accuracy and standard deviation in Table III. We also utilize the Friedman test to verify the significance of differences between and other approaches with p-value of 0.05.

TABLE IV
FRIEDMAN'S ANOVA TABLE

Source	SS	df	MS	Chi-sq	Prob>Chi-sq
Columns	118.8	6	19.8	25.46	0.0003
Error	21.2	24	0.8833		
Total	140	34			
P-value			**2.8091e-04**		

From the results, we can observe that LSTM has the worst performance because LSTM does not exploit the aspect information for sentiment prediction. The neural networks with the attention mechanism (e.g., MemNet, AOA, IAN, and TNET-LF) significantly perform better than the standard LSTM, since such methods explicitly encode the target information. For example, with the use of the Rest14 dataset, these methods improve 2.17%, 2.95%, 2.62%, and 3.56% in F1, respectively. This demonstrates that the attention mechanism can help the model to capture the aspect-specific information. ASGCN, which is the graph-based neural network, outperforms all other methods in three out of five datasets. For example, compared with the best competitor, ASGCN improves 2.42% on Rest15 and 0.91% on Lap14 for F1 score, respectively. This is because ASGCN utilizes the syntactical dependency tree to enhance the learning ability of the long-range word dependencies.

Our KGCapsAN achieves the best results among all five widely used datasets. For instance, our proposed method outperforms the best existing method under study by 3.71% on Rest15, 1.97% on Rest14, and 1.81% on Lap14 in terms of the F1 score. As we all know, it is difficult to improve 1% of the F1 score on the ATSA task [52]. This thus demonstrates the effectiveness of the proposed model. Also, we have carried out the KGCapsAN-LI model, which utilizes the emotion-related lexicon information to construct an additional query (as introduced in section III F). We observe that KGCapsAN-LI indeed improves KGCapsAN, which validates the effectiveness and flexibility of the proposed method to incorporate other prior knowledge.

It is also interesting to compare our model with the existing methods understudy that utilize the capsule network as the

TABLE V
EVALUATION RESULTS FOR BERT-BASED METHODS. THE BEST RESULTS ARE IN BOLD, † MARKS RESULTS PRODUCED BY USING THE OPEN IMPLEMENTATION

Model	Lap14		Twitter		Rest14	
	Accuracy (%)	F1 (%)	Accuracy (%)	F1 (%)	Accuracy (%)	F1 (%)
BERT	78.84	75.10	74.42	72.89	**85.45**	78.42
BERT-PT*	78.07	75.08	-	-	84.98	76.96
LCF-BERT†	79.62	76.26	73.27	72.35	84.38	78.72
AEN-BERT†	**79.93**	76.31	74.71	73.13	83.12	73.76
KGCapsAN-BERT	79.47	**76.61**	**74.86**	**73.70**	85.36	**79.00**

base sentiment classifier, such as TransCap and RNN-CAP. Compared with most methods under study, TransCap can still yield comparable results. This is because the dynamic routing method can capture the part-whole relationship and the output capsules contain the supplementary information than only one vector (non-capsule based networks utilized). Our model consistently and substantially outperforms RNN-CAP and TransCap in all tasks. This performance improvement can be explained with the following two reasons. First, our method utilizes prior knowledge as the queries to guide the capsule attention network. Second, the output capsules contain supplementary information that contributes to infer the sentiment.

As introduced in Section III F, the proposed method is also able to incorporate the external knowledge conveyed in BERT. Hence, in Table V we report and compare the results of our method and relevant BERT models. We can see that the proposed KGCapsAN-BERT in general outperforms other models. The results demonstrate the effectiveness of our variant model to incorporate BERT knowledge, and also validate our KGCapsAN framework can effectively combine the syntactic and local-n-gram prior knowledge for ASTA, due to its unique multi-knowledge guided capsule attention mechanism.

We have also performed a Friedman's test on five widely used datasets for testing the statistical significance of the performance superiority of the proposed method IV. Test results show that the proposed method is significantly better than the baselines at p value < 0.05.

d) Experimental Analysis on SpATSA: To evaluate the performance of the proposed method for solving the sentence with the special sentence structures (where the conventional methods fail to predict), we select several strong baselines, say, attention-based methods (MemNet, AOA) and syntactic enhance method (ASGCN), and we run several experiments on SpATSA. The results are summarized in Table III. From the results, our method achieves the state-of-the-art performance on the SpATSA dataset. Specifically, our method significantly outperforms other methods with attention-based structure. For example, our method improves 8.86% and 8.8% on MemNet for accuracy and F1 score, respectively. ASCGCN also performs better than that of MemNet. This is because the conventional method may mistakenly take some words into an aspect, but the use of syntactic knowledge can effectively help understand the sentence, especially for special sentence structure.

Compared with ASGCN, our method still yields an improvement. For instance, our method improves 2.62% (accuracy) and 2.72% (F1) for SpATSA. This shows that our methods can better utilize the syntactic information (both syntactic and locality knowledge). Besides, our Capsule Attention Method can provide additional knowledge for text representation which helps for significant performance improvement.

E. Ablation Study

In order to study the influence of each component of our model, we implement the ablation test of KGCapsAN in terms of removing the Capsule Attention network (denoted as w/o Caps), the structure can be regarded as the conventional attention-based method, Syntactic layer (denoted as w/o Syntactic), the local n-gram layer (denoted as w/o Local), the aspect query layer (denoted as w/o Aspect), and the aspect-specific zero masking layer (denoted as w/o Mask).

We construct the network of w/o Caps by utilizing the Bi-LSTM and the standard attention mechanism [53], in which the aspect term denotes the attention query. The model without CapsAttention layer is similar to KGCapsAN with just one iteration by only utilizing the aspect query layer. The results are summarized in Table VI. From the results, we can see that all the proposed parts provide a noticeable improvement to KGCapsAN. In particular, we can find that the CAN has the largest impact on the performance of KGCapsAN. The classification accuracy drops sharply when discarding CAN. This is within our expectation since the capsule attention network is able to capture the part-whole relationship and the output capsule can utilize additional information.

Under the CAN structure, we can observe that all three query layers contribute a lot to the performance. This is because such layers provide the KGCapsAN with multiple high-level knowledge. The syntactic layer can help capture the syntactical relationships between words, especially for the long-range words. The local n-gram layer enhances the learning ability of the local n-gram by using the locality of the aspect. The aspect query layer establishes the correlation between the aspect and sentiment-carry words. The mask strategy is a unique mechanism in our method, which deployed in three query layers. We observe from Table VI that the mask mechanism has a significant impact on the performance. To illustrate why the mechanism is effective, we depict in Figure 6 the attention weights learned by our model on an example sentence with/without the mechanism. We can see that without the mask the model tends to pay mistaken attentions to the non-local sentimental words, e.g., "fresh juice concoctions".

F. Case Study

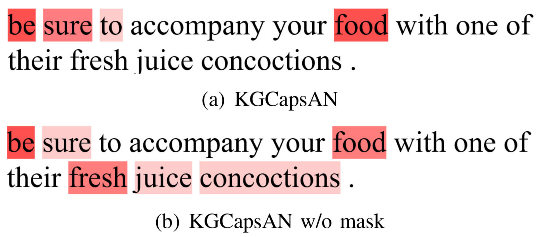
To better understand how our method works, we perform the case study with four examples. Specifically, we visualize the attention weights offered by the strong baseline methods

TABLE VI
Evaluation Results of Ablation Study on All Datasets

Model	Twitter		Lap14		Rest14		Rest15		Rest16	
	Accuracy %	F1 %	Accuracy %	F1 %	Accuracy %	F1 %	Accuracy %	F1 %	Accuracy %	F1 %
Ours	74.28	72.67	76.28	72.47	81.96	73.54	81.12	64.86	88.47	70.72
w/o Caps	**73.88**	**69.63**	**73.07**	**71.52**	**80.63**	**71.47**	**79.58**	**61.84**	88.42	69.35
-w/o Local	74.92	**70.02**	73.27	71.42	81.82	73.32	80.26	**61.76**	**88.10**	**66.18**
-w/o Syntactic	75.44	71.19	73.55	71.53	81.72	73.05	**80.01**	62.87	88.47	68.83
-w/o Aspect	75.29	70.97	**72.74**	**71.15**	**81.52**	**72.98**	81.37	63.63	88.20	68.14
-w/o Mask	**74.12**	71.26	75.08	71.16	81.54	73.12	80.41	63.41	88.25	68.14

TABLE VII
Case Study for Attention Scores of MemNet, IAN, ASGCN and KGCapsAN. The √ Indicates a Correct Prediction While × Indicates an Incorrect Prediction

Model	Prediction	Label		Attention visualization	Aspect
MemNet	positive	negative	×	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	negative	positive	×	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion
	positive	positive	×	We recently spent New Year 's Eve at the restaurant , and had a great experience , from the wine to the dessert menu .	dessert menu
	negative	neutral	×	the power plug has to be connected to the power adaptor to charge the battery but won't stay connected.	power adaptor
IAN	positive	negative	×	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	negative	positive	×	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion
	positive	positive	√	We recently spent New Year 's Eve at the restaurant , and had a great experience , from the wine to the dessert menu .	dessert menu
	negative	neutral	×	the power plug has to be connected to the power adaptor to charge the battery but won't stay connected.	power adaptor
ASGCN	neutral	negative	×	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	negative	positive	×	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion
	neutral	positive	×	We recently spent New Year 's Eve at the restaurant , and had a great experience , from the wine to the dessert menu .	dessert menu
	negative	neutral	×	the power plug has to be connected to the power adaptor to charge the battery but won't stay connected.	power adaptor
KGCapsAN	negative	negative	√	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	positive	positive	√	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion
	positive	positive	√	We recently spent New Year 's Eve at the restaurant , and had a great experience , from the wine to the dessert menu .	dessert menu
	neutral	neutral	√	the power plug has to be connected to the power adaptor to charge the battery but won't stay connected.	power adaptor


be sure to accompany your food with one of their fresh juice concoctions .

(a) KGCapsAN

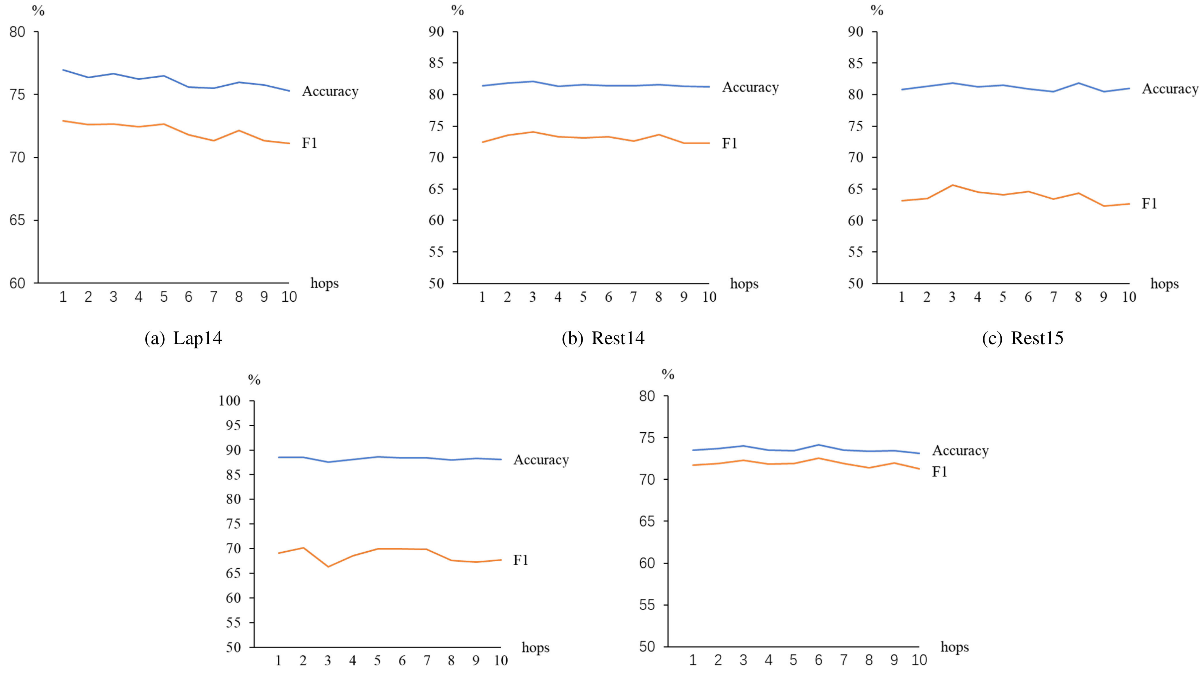
be sure to accompany your food with one of their fresh juice concoctions .

(b) KGCapsAN w/o mask

Fig. 6. A comparison example of attention weights learned with/without mask mechanism. The target of the sentence is "food," and the sentiment is neutral.

(MemNet, IAN, ASGCN) and our model KGCapsAN in Table VII. We also give prediction and the ground truth labels for such examples. The first sample "but, the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.," with the aspect term "filet mignon". This sentence contains the special structure "not very good". It is difficult to model such a structure using the conventional methods. For example, MemNet fails to focus on all informative words for the whole sentiment and it was not able to learn aspect-acrry words. IAN tends to focus on "very good," which may lead to incorrect prediction. ASGCN also fails to make such prediction, as it has a more attention to "good at". However, the KGCapsAN method can effectively predict this sentence. This may contribute by the local n-gram layer, which enables the model to consider "not very good" for the whole phase.

The second one "The Sashimi portion are big enough to appease most people, but I didn't like the fact they used artificial lobster meat.," has multiple aspect terms with different sentiment labels inside the sentence during testing (e.g., sashimi portion and artificial lobster meat). Such the case may lead the conventional attention-based models to align the aspects with their relevant descriptive words incorrectly. For example, given the aspect "sashimi portion," the existing methods tend



(a) Lap14

(b) Rest14

(c) Rest15

(d) Rest16

(e) Twitter

Fig. 7. Results of varying iteration numbers.

to focus on the words which in fact describe “artificial lobster meat ”. KGCapsAN successfully predicts the instance, because the aspect-specific zero masking mechanism can help focus on locality words (multi-words) towards the aspect.

For the third one: “We recently spent New Year’s Eve at the restaurant, and had a great experience, from the wine to the dessert menu.,” there is a long-range words distance between the sentiment-carry words and the aspect, which makes the model hard to detect implicit semantics. For example, MemNet and ASGCN focus on the un-related parts towards the aspect “dessert menu”. IAN can correctly make the prediction for this example. This may be due to the enhanced relationship between words and aspect term by the interactive attention mechanism. KGCapsAN can handle such a sample, as the uses of our capsule attention network and multiple information as queries can guide the attention process and retain more information in the output layer, making classification more accurate.

Finally, we give a neutral example: “the power plug has to be connected to the power adaptor to charge the battery but won’t stay connected.” with the target “power adaptor”. Such a sentence pattern contains a sentiment-carry clause, but does not directly describe the target. MemNet and IAN focus on the long-range sentiment carry words “but,” “won’t “ towards the aspect “price tag,” which results in wrong predictions. ASGCN also fails to make the correct prediction. This may due to the syntactical dependency tree can not handle complex sentence structure nicely, such as “but”-clause. KGCapsAN avoids the excessive focus on emotional words and gives a successful prediction. This benefits from our CAN structure, which allows the model to understand the sentence from different perspectives, instead of the current attention mechanism utilized in existing methods.

G. Number of Routing Iterations

The number of routing iterations is an important hyper-parameter of capsule-based structure, since it helps to model the part-whole representation. Previous studies show that multiple iterations can lead to better results. Here we would like to investigate its impact on the proposed KGCapsAN. Specifically, we report its performance on all the data sets by increasing the number of routing iterations from one to ten. Here we would like to study its impact on the proposed KGCapsAN. Specifically, we give the performance for all datasets by running the experiments with the number of routing interactions from one to ten. Classification accuracy and F1 scores are the average value over 3 runs with random initialization. Figure 7 shows the results. We observe that KGCapsAN can obtain the best performance with the number of iterations within three iterations. After six iterations, the performance tends to decline steadily. The reason may be because our method utilizes the prior knowledge to guide the attentions. As a result, the proposed method can focus accurately on distinct features rapidly, and thus it needs very few iterations to deliver the best performance.

H. Error Analysis

To investigate the limitation of the proposed methods, we further analyze the prediction errors by our method. Specifically, we analyze the error samples when the proposed method runs over the Rest14 dataset. There are several reasons for explaining the findings. First, our method fails to understand some sentences that the sentiment hides in the latent opinions. For example, the correct label for the sentence “you can eat gourmet food at a fast food price.” should be “positive”. Our method tends to classify the sentence as “neutral,” since the model was unable to explore the internal relationship between "fast food price" and "gourmet food. Second, our method fails to classify some sentences that require a deep comprehension such as "unfortunately, unless you live in the neighborhood, it's not in a convenient location but is more like a hidden treasure." The attention weights tend to assign to the negative words such as "unfortunately," "not convenient". This is because of the proposed method unable to understand the special sentence structure, e.g., "but"-clause.

V. Conclusion

In this paper, we propose a knowledge guided capsule network (KGCapsAN) for aspect-level sentiment analysis. Our method consists of two parts, a text representation layer and a capsule attention network (CAN). CAN implements the routing method by attention mechanism, in which multi-prior knowledge is utilized to guide the capsule attention process. Among CAN, we first propose a GCN-based syntactic layer to integrate the syntactic knowledge acquiring from the syntactic dependency tree. Additionally, we propose a local n-gram layer to enhance the ability of the model which can effectively focus on the informative n-gram features. The experimental results demonstrated that the KGCapsAN model significantly outperformed the state-of-the-art methods for aspect-based sentiment analysis. In addition, ablation study, qualitative analysis, and case visualization are provided to further demonstrate the effectiveness of the proposed model. In the future, we prepare to combine the sentiment resources (e.g., sentiment lexicon, emotional tags) into the KGCapsAN, which can supply further comprehensive knowledge for sentiment classification.

References

[1] B. Huang, Y. Ou, and K. M. Carley, "Aspect level sentiment classification with attention-over-attention neural networks," in *Proc. Int. Conf. Soc. Comput., Behavioral-Cultural Model. Predict. Behavior. Represent. Model. Simul.*, 2018, pp. 197–206.

[2] B. Pang *et al.*, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, no. 1/2, pp. 1–135, 2008.

[3] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent twitter sentiment classification," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguist. Human Language Technol.-Volume 1*. 2011, pp. 151–160.

[4] S. Poria, E. Cambria, D. Hazarika, and P. Vij, "A deeper look into sarcastic tweets using deep convolutional neural networks," in *Proc. 26th Int. Conf. Comput. Linguist.: Tech. Papers*, 2016, pp. 1601–1612.

[5] P. Chen, Z. Sun, L. Bing, and W. Yang, "Recurrent attention network on memory for aspect sentiment analysis," in *Proc. Conf. Empirical Methods Natural Language Process.*, 2017, pp. 452–461.

[6] D. Tang, B. Qin, X. Feng, and T. Liu, "Effective LSTMs for target-dependent sentiment classification," in *Proc. 26th Int. Conf. Comput. Linguist. Tech. Papers*, 2016, pp. 3298–3307.

[7] Y. Wang *et al.*, "Attention-based LSTM for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods in Natural Lang. Process.*, 2016, pp. 606–615.

[8] T. H. Nguyen and K. Shirai, "PhraseRNN: Phrase recursive neural network for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2509–2514.

[9] D. Tang, B. Qin, and T. Liu, "Aspect level sentiment classification with deep memory network," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 214–224.

[10] J. Liu and Y. Zhang, "Attention modeling for targeted sentiment," in *Proc. 15th Conf. Eur. Chapter Assoc. for Comput. Linguist. Volume 2, Short Papers*, 2017, pp. 572–577.

[11] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 4068–4074.

[12] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Proc. Adv. Neural Inform. Process. Syst.*, 2017, pp. 3859–3869.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[14] J. Wang, L. Yu, K. R. Lai, and X. Zhang, "Tree-structured regional CNN-LSTM model for dimensional sentiment analysis," *IEEE ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 581–591, 2020.

[15] Y. Wang, A. Sun, J. Han, Y. Liu, and X. Zhu, "Sentiment analysis by capsules," in *Proc. World Wide Web Conf. World Wide Web*. International World Wide Web Conferences Steering Committee, 2018, pp. 1165–1174.

[16] H. Xu, B. Liu, L. Shu, and S. Y. Philip, "Double embeddings and CNN-based sequence labeling for aspect extraction," in *Proc. 56th Annu. Meet. Assoc. Comput. Linguist. (Volume 2: Short Papers)*, 2018, pp. 592–598.

[17] Y. Song, J. Wang, T. Jiang, Z. Liu, and Y. Rao, "Attentional encoder network for targeted sentiment classification," 2019, *arXiv:1902.09314*.

[18] L. Li, Y. Liu, and A. Zhou, "Hierarchical attention based position-aware network for aspect-level sentiment analysis," in *Proc. 22nd Conf. Comput. Natural Lang. Learn.*, 2018, pp. 181–189.

[19] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proc. Int. Conf. Artif. Neural Netw.*, 2011, pp. 44–51.

[20] M. Yang, W. Zhao, L. Chen, Q. Qu, Z. Zhao, and Y. Shen, "Investigating the transferring capability of capsule networks for text classification," *Neural Netw.*, vol. 118, pp. 247–261, 2019.

[21] B. Zhang, X. Xu, M. Yang, X. Chen, and Y. Ye, "Cross-domain sentiment classification by capsule network with semantic rules," *IEEE Access*, vol. 6, pp. 58 284–58 294, 2018.

[22] Y. Zhou, R. Ji, J. Su, X. Sun, and W. Chen, "Dynamic capsule attention for visual question answering," 2019.

[23] Y. Wang, A. Sun, M. Huang, and X. Zhu, "Aspect-level sentiment analysis using AS-capsules," in *Proc. World Wide Web Conf.*, 2019, pp. 2033–2044.

[24] Z. Yang, J. Zhang, F. Meng, S. Gu, Y. Feng, and J. Zhou, "Enhancing context modeling with a query-guided capsule network for document-level translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 1527–1537.

[25] H. Cai, V. W. Zheng, and K. C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1616–1637, Sep. 2018.

[26] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Representations*, Toulon, France, Apr. 24-26, 2017.

[27] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-CNN," in *Proc. World Wide Web Conf.* International World Wide Web Conferences Steering Committee, 2018, pp. 1063–1072.

[28] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proc. 33rd AAAI Conf. Artif. Intell., 31st Innovative Appl. Artif. Intell. Conf., 9th AAAI Symp. Educational Advances Artificial Intell.*, Honolulu, Hawaii, USA, Jan. 27–Feb. 1, 2019, pp. 7370–7377.

[29] B. Zhang, X. Xu, M. Yang, X. Chen, and Y. Ye, "Cross-domain sentiment classification by capsule network with semantic rules," *IEEE Access*, vol. 6, pp. 58 284–58 294, 2018.

[30] M. Dragoni and G. Petrucci, "A fuzzy-based strategy for multi-domain sentiment analysis," *Int. J. Approx. Reason.*, vol. 93, pp. 59–73, 2018.

[31] J. Zhang, P. Lertvittayakumjorn, and Y. Guo, "Integrating semantic knowledge to tackle zero-shot text classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. Human Lang. Technol., Volume 1 (Long and Short Papers)*, 2019, pp. 1031–1040.

[32] Z. Hu, X. Ma, Z. Liu, E. Hovy, and E. Xing, "Harnessing deep neural networks with logic rules," in *Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (Volume 1: Long Papers)*, 2016, pp. 2410–2420.

[33] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proc. Conf. Empir. Methods Nat. Lang. Process.*, 2018, pp. 2205–2215.

[34] C. Zhang, Q. Li, and D. Song, "Aspect-based sentiment classification with aspect-specific graph convolutional networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4560–4570.

[35] E. Zuo, H. Zhao, B. Chen, and Q. Chen, "Context-specific heterogeneous graph convolutional network for implicit sentiment analysis," *IEEE Access*, vol. 8, pp. 37 967–37 975, 2020.

[36] B. Zeng, H. Yang, R. Xu, W. Zhou, and X. Han, "LCF: A local context focus mechanism for aspect-based sentiment classification," *Appl. Sci.*, vol. 9, p. 3389, 08 2019.

[37] W. Wei and J. A. Gulla, "Enhancing the HL-SOT approach to sentiment analysis via a localized feature selection framework," in *Proc. 5th Int. Joint Conf. Natural Lang. Process.*, 2011, pp. 327–335.

[38] Z. Hu, Z. Yang, R. Salakhutdinov, and E. Xing, "Deep neural networks with massive learned knowledge," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1670–1679.

[39] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *Stat*, vol. 1050, p. 21, 2016.

[40] E. Cambria, S. Poria, D. Hazarika, and K. Kwok, "Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018.

[41] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, "Adaptive recursive neural network for target-dependent Twitter sentiment classification," in *Proc. 52nd Ann. Meeting Assoc. Comput. Linguist. (volume 2: Short papers)*, 2014, pp. 49–54.

[42] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *Proc. 8th Int. Workshop Semantic Evaluation (SemEval 2014)*. Dublin, Ireland, Aug. 2014, pp. 27–35. [Online]. Available: https://www.aclweb.org/anthology/S14-2004

[43] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "Semeval-2015 task 12: Aspect based sentiment analysis," in *Proc. 9th Int. Workshop Semant. Eval. (SemEval 2015)*, 2015, pp. 486–495.

[44] M. Pontiki *et al.*, "Semeval-2016 task 5: Aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval-2016)*, 2016, pp. 19–30.

[45] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," in *Proc. 56th Annu. Meet. Assoc. Comput. Linguist. (Volume 1: Long Papers)*, 2018, pp. 946–956.

[46] S. Kiritchenko, X. Zhu, C. Cherry, and S. Mohammad, "Nrc-canada-2014: Detecting aspects and sentiment in customer reviews," in *Proc. 8th Int. Workshop Semantic Evaluation (SemEval 2014)*, 2014, pp. 437–442.

[47] Z. Chen and T. Qian, "Transfer capsule network for aspect level sentiment classification," in *Proc. 57th Annu. Meet. Assoc. Comput. Linguist.*, 2019, pp. 547–556.

[48] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. Human Lang. Technol., NAACL-HLT 2019, Minneapolis, MN, USA, Jun. 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.

[49] Y. Song, J. Wang, T. Jiang, Z. Liu, and Y. Rao, "Attentional encoder network for targeted sentiment classification," *CoRR*, vol. abs/1902.09314, 2019. [Online]. Available: http://arxiv.org/abs/1902.09314

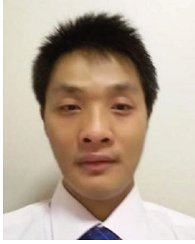
[50] H. Xu, B. Liu, L. Shu, and P. S. Yu, "BERT post-training for review reading comprehension and aspect-based sentiment analysis," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguist. Human Lang. Technol., NAACL-HLT 2019, Minneapolis, MN, USA, Jun. 2-7, 2019, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 2324–2335.

[51] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Thirteenth Int. Conf. Artif. Intell. Statist., AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, ser. JMLR Proceedings, Y. W. Teh and D. M. Titterington, Eds., vol. 9. JMLR.org, 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[52] M. Yang, Q. Jiang, Y. Shen, Q. Wu, Z. Zhao, and W. Zhou, "Hierarchical human-like strategy for aspect-level sentiment classification with sentiment linguistic knowledge and reinforcement learning," *Neural Netw.*, vol. 117, pp. 240–248, 2019.

[53] D. Tang, B. Qin, X. Feng, and T. Liu, "Target-dependent sentiment classification with long short term memory," *CoRR*, vol. abs/1512.01100, 2015. [Online]. Available: http://arxiv.org/abs/1512.01100

Bowen Zhang received the B.Sc. and M.Sc. degrees in computer science and application from the Macau University of Science and Technology, Taipa, China. He is currently working toward the doctorate degree with the the Harbin Institute of Technology, Harbin, China. His research interests are in natural language processing, big data analysis, and data mining.



Xutao Li received the bachelor's degree from the Lanzhou University of Technology, Lanzhou, China, in 2007, and the master's and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 2009 and 2013, respectively. He is currently an Associate Professor with the Shenzhen Graduate School, Harbin Institute of Technology. His research interests include data mining, machine learning, graph mining and social network analysis, especially tensor based learning and mining algorithms.

Xiaofei Xu received the Ph.D. degree in computer science from Harbin Institute of Technology (HIT), Harbin, China, in 1988. He is currently a Professor of computer science with the HIT, and Vice President of HIT, and President of HIT, Weihai. His research interests include service computing and service engineering, cloud services and big services, enterprise computing and enterprise interoperability, supply chain management, software engineering, databases and data mining, business intelligence, smart city services, smart healthcare and elder-care, etc.

Ka-Cheong Leung (Senior Member, IEEE) received the B.Eng. degree in computer science from the Hong Kong University of Science and Technology, Hong Kong, in 1994, the M.Sc. degree in electrical engineering (Computer Networks) and the Ph.D. degree in computer engineering from the University of Southern California, Los Angeles, CA, USA, in 1997 and 2000, respectively. He was with Nokia Research Center, Nokia, Inc., Irving, Texas, USA from 2001 to 2002, Texas Tech University, Lubbock, Texas, USA, from 2002 to 2005, and the University of Hong Kong, Hong Kong, from 2005 to 2019. He is currently an Associate Professor with the School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China. His research interests include smart grid, vehicle-to-grid (V2G), machine learning, future Internet, and wireless communications. He is an Associate Editor for the IEEE SYSTEMS JOURNAL and *ACM/Springer Wireless Networks*. He has also co-guest edited a special issue of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING. Furthermore, he is a Subarea-Chair of the IEEE SIG on Intelligent Internet Edge.

Zhiyao Chen received the B.Sc. degree in computer science and application from the Hefei University of Technology, Hefei, China. He is currently working toward the master's degree with the Harbin Institue of Technology, Harbin, China. His research interests are in natural language processing, big data analysis, and data mining.

Yunming Ye received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China. He is currently a Professor with the Shenzhen Graduate School, Harbin Institute of Technology, Harbin, China. His research interests include data mining, text mining, and ensemble learning algorithms.