IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 5, 2019, accepted May 25, 2019, date of publication May 30, 2019, date of current version June 28, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2920075*

# Deep Learning for Aspect-Level Sentiment Classification: Survey, Vision, and Challenges

**JIE ZHOU[1,2], JIMMY XIANGJI HUANG[3], (Senior Member, IEEE), QIN CHEN[4], QINMIN VIVIAN HU[5], TINGTING WANG[1,2], AND LIANG HE[1,2]**

[1]Shanghai Key Laboratory of Multidimensional Information Processing, East China Normal University, Shanghai 200241, China
[2]Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China
[3]Information Retrieval and Knowledge Management Research Laboratory, York University, Toronto, ON M3J 1P3, Canada
[4]School of Data Science, Fudan University, Shanghai 200433, China
[5]School of Computer Science, Ryerson University, Toronto, ON M5B 2K3, Canada

Corresponding author: Liang He (lhe@cs.ecnu.edu.cn)

This work was supported in part by the Natural Science Foundation of Shanghai under Grant 17ZR1444900, in part by the Science and Technology Commission of Shanghai Municipality under Grant 18511103802, in part by the Xiaoi Research, in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the NSERC CREATE Award in Advanced Disaster, Emergency and Rapid Response Simulation (ADERSIM), in part by the York Research Chairs (YRC) Program, and in part by the Ontario Research Fund-Research Excellence (ORF-RE) Award in BRAIN Alliance. The work of J. Zhou was supported by the NSERC Discovery Grant.

**ABSTRACT** This survey focuses on deep learning-based aspect-level sentiment classification (ASC), which aims to decide the sentiment polarity for an aspect mentioned within the document. Along with the success of applying deep learning in many applications, deep learning-based ASC has attracted a lot of interest from both academia and industry in recent years. However, there still lack a systematic taxonomy of existing approaches and comparison of their performance, which are the gaps that our survey aims to fill. Furthermore, to quantitatively evaluate the performance of various approaches, the standardization of the evaluation methodology and shared datasets is necessary. In this paper, an in-depth overview of the current state-of-the-art deep learning-based methods is given, showing the tremendous progress that has already been made in ASC. In particular, first, a comprehensive review of recent research efforts on deep learning-based ASC is provided. More concretely, we design a taxonomy of deep learning-based ASC and provide a comprehensive summary of the state-of-the-art methods. Then, we collect all benchmark ASC datasets for researchers to study and conduct extensive experiments over five public standard datasets with various commonly used evaluation measures. Finally, we discuss some of the most challenging open problems and point out promising future research directions in this field.

**INDEX TERMS** Aspect based sentiment analysis, aspect-level sentiment classification, attention, convolutional neural network (CNN), deep learning, memory network, neural networks, recurrent neural network (RNN).

## I. INTRODUCTION

Sentiment analysis can be divided into three levels, namely the document level, the sentence level, and the aspect level [1]. The document-level sentiment analysis comes with an assumption that the whole document only contains opinions about one topic. Obviously, this is not reasonable in many cases. The sentence-level sentiment analysis similarly assumes that only one topic is expressed in one sentence. However, it is often the case that one sentence contains multiple topics (i.e., aspects) or that the opinions are opposite within the same sentence. For both the document-level and the sentence-level sentiment analysis, the decided sentiment polarities are based on the whole document/sentence rather than the topics given in the document/sentence. In contrast, aspect-level sentiment analysis aims to judge the sentiment polarity expressed for each aspect being discussed. This allows for a more detailed analysis that makes use of more information given by the review/tweet.

As a fundamental subtask of sentiment analysis [2], aspect-level sentiment analysis has received a lot of attention from both industries and academic communities.

The associate editor coordinating the review of this manuscript and approving it for publication was Arif Ur Rahman.

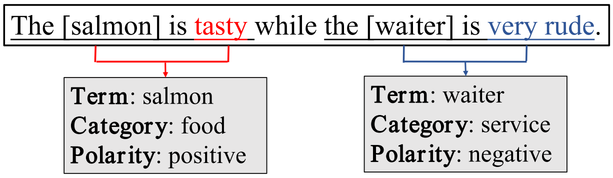


**FIGURE 1.** An example of aspect-level sentiment classification.

Recently, aspect-level sentiment analysis is a central concern in the research communities of semantic Web and computational linguistics [3]–[5]. The goal of aspect-level sentiment analysis is to identify the aspects (aspect extraction) and infer the sentiment expressed for each aspect (also known as aspect-level sentiment classification). In this paper, we focus on the problem of deep learning based aspect-level sentiment classification (ASC). This allows us to cover more recent developments, instead of repeating established insights provided in other surveys [1], [6]–[8]. Evidently, the field of deep learning in ASC is flourishing. The goal of ASC is to determine whether user opinions expressed in reviews/tweets on given aspects (aspect categories or aspect terms) are positive, negative, or neutral [1]. An example in Figure 1 presents a sample sentence taken from the SemEval 2014 [3] dataset (e.g., Restaurants14). An aspect category implicitly describes a general category of the entities. For instance, in the sentence "The salmon is tasty while the waiter is very rude", the user expresses positive and negative sentiments towards two aspect categories "food" and "service" respectively. An aspect term characterizes a specific entity that explicitly occurs in a sentence. For the same sentence, the aspect terms are "salmon" and "waiter", where the user expresses positive and negative sentiments over them respectively. The aspect category is coarse-grained while the aspect term is fine-grained in terms of the aspect granularity [9].

In recent years, several surveys in the field of traditional sentiment analysis have been published. For example, in 2008, Pang *et al.* [6] presented a good review of sentiment analysis. Various techniques and applications were discussed, and the considerations of ethics, theory, and practice were covered. However, they mainly focused on traditional machine learning approaches for document-level sentiment analysis. In 2009, Tang *et al.* [8] introduced a survey which also mainly paid attention to the machine learning methods for document-level sentiment analysis on the consumer reviews domain. In 2011, Tsytsarau and Palpanas [7] published a survey, which focused on document-level sentiment analysis as well and discussed four different approaches for predicting the sentiment polarity, namely machine learning based, dictionary-based, semantic-based and statistical-based respectively. In 2012, Liu [1] presented a survey, which provided an introduction of the entire field of sentiment analysis. A list of sub-problems when implementing an actual solution were given in the section about aspect-level sentiment analysis: 1) the definitions of aspect extraction, including various challenges like solving implicit and explicit entities and opinions; 2) how to identify the aspects and sentiment polarities and linked to each another (a.k.a. ASC). The most related survey work to ours is [10] that focused on aspect-level sentiment analysis. It gave an in-depth overview of the traditional machine learning methods for aspect-level sentiment analysis, including aspect extraction and ASC. However, none of the existing work focuses on deep learning based ASC though it has achieved great success in recent years. Furthermore, a systematic classification of deep learning based approaches for ASC and reports of their performance over benchmark datasets are missing, which are the gaps that this survey is aiming to fill.

This survey aims to thoroughly review the literature on the advances of deep learning based ASC. It provides an overview with which readers can quickly understand and step into the field of deep learning based ASC. This survey serves the researchers, practitioners, and educators who are interested in ASC, with the hope that they will have a rough guideline when it comes to choosing the deep learning models to solve ASC tasks at hand. To summarize, the differences between this survey and former ones include: (1) to the best of our knowledge, this is the first time to well summarize the field of deep learning based ASC and organize existing works and current progress. 2) we collect and analyze almost all of the benchmark datasets of ASC; 3) we implement the classical state-of-the-art models and evaluate them on five classical datasets with widely used evaluation measures.

The key contributions of this paper can be summarized as follows.

1) In light of the significantly increasing number of studies on deep learning for ASC, it is necessary to make a comprehensive summarization of existing literature. To this end, we are the first to provide a detailed review over representative approaches and summarize current work with a classification scheme. The existing methods are categorized into five main groups: RecNN for ASC, RNN for ASC, attention-based RNN for ASC, CNN for ASC and memory network for ASC.
2) This survey provides the most comprehensive overview of modern deep learning techniques for ASC. For each type of ASC models, we provide detailed descriptions on representative algorithms, and make a necessary comparison and summaries the corresponding algorithms.
3) We collect almost all of the standard ASC datasets, including SemEval 2014, SemEval 2015, SemEval 2016, Twitter and others. We also explore these datasets in detail and translate them into a uniform XML/JSON format for researchers to study.[1]
4) Note that the existing models verify the effectiveness on different datasets with different metrics and

[1] All the source codes and collected datasets for ASC are available at https://github.com/12190143/Deep-Learning-for-Aspect-Level-Sentiment-Classification-Baselines and https://www.yorku.ca/jhuang/Deep-Learning-for-Aspect-Level-Sentiment-Classification-Baselines.

the experiment settings of these models are different. We reproduce the classical state-of-the-art deep learning methods for ASC and evaluate the performance of them with the commonly-used metrics (e.g. Accuracy, Precision, Recall, Macro-F1 and so on.) over public benchmark ASC datasets.

5) We discuss the important challenges and open issues in deep learning based ASC, which provides a promising avenue and inspires the vision for further research.

The remaining of this article is organized as follows. Section II introduces the overview of deep neural based ASC and presents our classification framework. Section III gives a detailed introduction to the state-of-the-art methods. Then, the benchmark datasets and widely-used evaluation measures for ASC are described in Section IV and Section V respectively. After that, Section VI reports the experimental results with various metrics over public benchmark datasets and Section VII discusses the challenges and prominent open research issues for ASC. Finally, we summarize our work in Section VIII.

## II. OVERVIEW OF DEEP LEARNING BASED ASC

Before we dive into the details of this survey, we start with a discussion about the reasons and motivations of introducing deep neural networks to ASC. We also introduce the basic terminology and concepts regarding deep learning based ASC techniques.

Aspect based sentiment analysis is a fundamental task in sentiment analysis research field [3], [11], which includes several key sub-tasks: aspect extraction [12]–[14], opinion identification [15], [16] and ASC [17]–[19]. Some previous studies have tried to solve these sub-tasks jointly [20], [21], dedicating most of the research work in dealing with an individual sub-task. In this study, we focus on deep learning methods for solving ASC problem. Different from document-level and sentence-level sentiment classification, ASC considers both the sentiment and the target information, as a sentiment always has a target. As mentioned above, a target is usually an entity or an aspect of an entity. For simplicity, both entity and aspect are usually called aspect. Given a sentence and an aspect, ASC aims to infer the sentiment polarity/orientation of the sentence towards the given aspect.

Traditional methods for ASC are mostly traditional machine learning models based on lexicons and syntactic features [17], [18], [22]. The performance of such models is highly dependent on the quality of the hand-crafted features which is labor intensive. Therefore, recent research has turned its attention to developing end-to-end deep neural network models. To provide insight into the large number of proposed deep learning based methods for ASC, a categorization is made based the types of employed deep learning techniques, dividing all approaches into the following five categories: recursive neural network (RecNN) for ASC, recurrent neural network (RNN) for ASC, attention-based RNN for ASC, convolutional neural network (CNN) for ASC and memory network for ASC. Figure 2 summarizes the classification scheme. Additionally, Table 1 lists all the reviewed approaches, which were organized following the above classification scheme. We give a brief introduction of each category in the following sections and the details of these approaches will be provided in Section III.

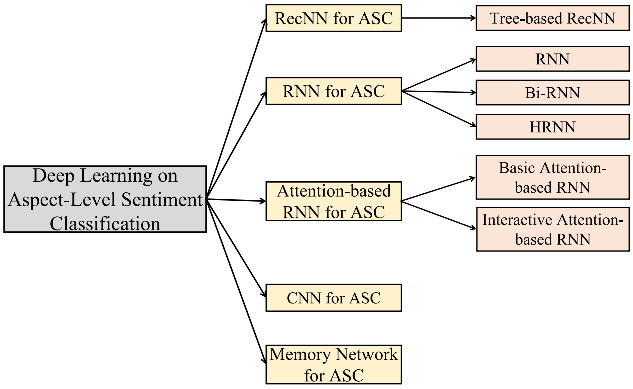

Deep Learning on Aspect-Level Sentiment Classification
RecNN for ASC
Tree-based RecNN
RNN for ASC
RNN
Bi-RNN
HRNN
Attention-based RNN for ASC
Basic Attention-based RNN
Interactive Attention-based RNN
CNN for ASC
Memory Network for ASC

**FIGURE 2. The categorization of deep learning methods for ASC. We divide existing methods into five categories: RecNN for ASC, RNN for ASC, attention-based RNN for ASC, CNN for ASC and memory network for ASC. The RNN for ASC can be further divided into RNN, Bi-RNN, and HRNN based on their architectures. Attention-based RNN models include basic attention-based RNN, interactive attention-based RNN.**

### A. RECNN FOR ASC

Recursive Neural Network (RecNN) [56] is a type of neural network that is applied to learn a directed acyclic graph structure (e.g., a tree structure) from data. It can be regarded as a generalization of the recurrent neural network. Given the structural representation of a sentence (e.g., a parse tree), RecNN recursively generates parent representations in a bottom-up way, by combining tokens to obtain representations for phrases, eventually the whole sentence. The representation of a sentence then is used to make a final prediction (e.g., sentiment classification) for the given input sentence. Tree-based RecNN was introduced into ASC by Dong *et al.* [23] and Nguyen and Shirai [24].

### B. RNN FOR ASC

The recurrent neural network (RNN) has been shown to be powerful in many (language) sequence learning problems. Moreover, most of state-of-the-art methods for ASC are based on RNN [25]–[28]. In this category, models can be divided into three subcategories: RNN, bidirectional-RNN (Bi-RNN), and hierarchical RNN (HRNN). To capture semantic relations between the aspect and its context words in a more flexible way, Tang *et al.* [26] proposed a target-dependent LSTM (TD-LSTM) and a target-connection LSTM (TC-LSTM) to extend LSTM by taking the aspect into consideration, which applied RNN into ASC. In addition, commonsense knowledge of sentiment-related concepts was incorporated into the end-to-end training of an LSTM model for ASC [27]. Zhang *et al.* [25] used gated neural network structures to model the syntax and semantics in sentence and

**TABLE 1.** **Statistics of the existing public methods for ASC. Model represents the type of deep learning methods adopted by the corresponding published paper. Aspect and Position with √ denote the model considering the aspect information and position information respectively. Attention without × means the model using the attention, CA, GA and DPA indicate "Contact Attention", "General Attention" and "Dot-Product Attention" respectively, which will be described in Section III-D.1 in detail.**

| | Method | Model | Attention | Aspect | Position |
|---|---|---|---|---|---|
| RecNN for ASC | | | | | |
| Tree-base RecNN | AdaRNN [23] | RecNN | × | × | × |
| | PhraseRNN [24] | RecNN | × | × | × |
| RNN for ASC | | | | | |
| Bi-RNN | GRNN [25] | Bi-GRU | × | × | × |
| RNN | TD-LSTM/TC-LSTM [26] | LSTM | × | √ | × |
| | Sentic LSTM+TA+SA [27] | Bi-LSTM | CA | √ | × |
| HRNN | H-LSTM [28] | Bi-LSTM | × | × | × |
| Attention-based RNN for ASC | | | | | |
| Basic Attention | ATAE-LSTM/AE-LSTM [29] | LSTM | CA | √ | × |
| | AB-LSTM [30] | Bi-LSTM | DPA+GA | √ | × |
| | LSTM+SynATT+TarRep [31] | LSTM | GA | √ | √ |
| | Word&Clause-Level ATT [32] | HAN+Bi-LSTM | CA | √ | × |
| | AF-LSTM(CONV) [33] | LSTM | CA | √ | × |
| | HEAT [34] | Bi-GRU | CA | √ | × |
| | MGAN [9] | Bi-LSTM | CA | √ | √ |
| | PosATT-LSTM [35] | LSTM | CA | √ | √ |
| | PRET+MULT [36] | LSTM | GA | √ | × |
| | Inter-Aspect Dependencies [37] | LSTM | CA | √ | × |
| Interactive Attention | IAN [38] | LSTM | GA | √ | × |
| | BILSTM-ATT-G [39] | Bi-LSTM | CA | √ | × |
| | MGAN [40] | Bi-LSTM | DPA+GA | √ | √ |
| | PBAN [41] | Bi-GRU | GA | √ | √ |
| | AOA-LSTM [42] | Bi-LSTM | GA | √ | × |
| | LCR-Rot [43] | Bi-LSTM | GA | √ | × |
| CNN for ASC | | | | | |
| CNN | GCAE [44] | CNN | × | √ | × |
| | PF-CNN [45] | CNN | × | × | × |
| | Conv-Memnet [46] | CNN+Memory | GA | √ | × |
| | TNet [47] | LSTM+CNN | CA | √ | √ |
| Memory Network for ASC | | | | | |
| Memory Network | MemNet [19] | Memory | CA | √ | √ |
| | DyMemNN [48] | Memory | CA | √ | × |
| | RAM [49] | LSTM+Memory | CA | √ | √ |
| | CEA [50] | Memory | CA | √ | √ |
| | DAuM [51] | Memory | GA | √ | × |
| | IARM [52] | GRU+Memory | CA | √ | × |
| | TMNs [53] | Memory | GA | √ | × |
| | L2MNS [54] | Memory | CA | √ | × |
| | Cabasc [55] | Memory | CA | √ | √ |

the interaction between the aspect and its surrounding context words by Bi-RNN. HRNN was adopted by Ruder *et al.* [28], who proposed to use a hierarchical bidirectional LSTM model for ASC, which was able to learn both intra- and inter-sentence relations.

### C. ATTENTION-BASED RNN FOR ASC

Attention mechanism [57] has been successfully applied to many natural language processing (NLP) tasks [58], such as neural machine translation [57], [59], question answering [60], [61], and machine comprehension [62], [63]. A variety of attention-based RNN models have recently been introduced to ASC, which can attend to the important parts of the sentence towards the given aspect effectively. Attention-based RNN for ASC can be divided into basic attention-based RNN models and interactive attention-based RNN models. In particular, there were a large number of studies focus on improving the basic attention RNN models [27], [31]–[33], [35]–[37], [41], [55]. Wang *et al.* [29] proposed an attention-based LSTM method with aspect embedding, which was proven to be an effective way to enforce the model to capture the related parts of a sentence. In addition, interactive attention based models were widely used for ASC [38]–[40], [42], [43]. For instance, Ma *et al.* [38] proposed an interactive attention mechanism, which interactively learned attentions from the specific aspect and the context.

### D. CNN FOR ASC

Convolutional neural network (CNN) [64] is good at capturing local patterns and plays an important role in NLP [65]. CNN is able to extract the local and global representations from a sentence. Some work adopted CNN for ASC [45]–[47]. To be specific, Huang and Carley [45] incorporated aspect information into CNN leveraging parameterized filters and parameterized gates. Li *et al.* [47] adopted a proximity strategy to scale the input of the convolutional layer with positional relevance between a word and an aspect.

Fan *et al.* [46] proposed a convolutional memory network which incorporated an attention mechanism to capture both words and multi-words expressions in sentences for ASC. Furthermore, Xue and Li [44] proposed a model based on convolutional neural networks and gating mechanisms.

### *E. MEMORY NETWORK FOR ASC*

Memory network [61] obtained great success in ASC [19], [48]–[55], [66]. Tang *et al.* [19] first introduced an end-to-end memory network for ASC, which employed an attention mechanism with an external memory to capture the important information of the sentence with respect to the given aspect. Chen *et al.* [49] proposed a recurrent attention mechanism based on memory network for each aspect to capture sentiment information separated by a long distance. In [55], the sentence-level content attention mechanism was proposed to overcome the short-sight problem of the memory models.

## III. DEEP LEARNING BASED ASC

In this section, problem definition and notations are given first. Then, we highlight the state-of-the-art research prototypes for deep learning based ASC to identify the most notable and promising advancement in recent years.

### *A. PROBLEM DEFINITION AND NOTATIONS*

Given a sentence-aspect pair $(S, A)$, where the aspect $A = \{w_{start}, t_{start+1}, \ldots, w_{end-1}, w_{end}\}$ is the subsequence of the sentence $S = \{w_1, w_2, \ldots, w_n\}$ that consists of n words, *start* and *end* are starting and ending indices of the aspect $A$ that consists of $m = end - start + 1$ words. The goal of ASC is to predict sentiment polarity $c \in C = \{N, O, P\}$ for the sentence $S$ towards the aspect $A$, where N, O, and P denote the "negative", "neutral" and "positive" sentiment polarities respectively.

For the sentence $S = \{w_1, w_2, \ldots, w_n\}$ and the aspect words $A = \{w_{start}, t_{start+1}, \ldots, w_{end-1}, w_{end}\}$, we map each word into its embedding vector $X = \{x_1, x_2, \ldots, x_n\}$ and $V = \{v_{start}, v_{start+1}, \ldots, v_{end-1}, v_{end}\}$. It maps the word representation from a high-dimensional sparse vector space (e.g. one-hot encoding vector space) to a lower-dimensional dense vector space. One commonly used word embedding approach is Word2Vec,[2] which contains Continuous Bags-of-Words model (CBOW) [67], and Skip-Gram model (SG) [68]. Another frequently used learning method is Global Vector (GloVe)[3] [69], which is trained on the non- zero entries of a global word-word co-occurrence matrix. We summarize the commonly used notations in Table 2.

### *B. RECNN FOR ASC*

In this section, we first introduce the basic RecNN model and then we describe the studies of tree-based RecNN for ASC in detail.

**TABLE 2. Commonly used notations.**

| Notations | Descriptions |
|---|---|
| $\lvert \cdot \rvert$ | The length of a set. |
| $\odot$ | Element-wise product. |
| $[A; B]$ | Concatenation of A and B. |
| $S$ | A sentence |
| $A$ | An aspect |
| $w_i$ | A word $w_i \in S$ |
| $x_i$ | The embedding of word $w_i$ |
| n,m | The length of the sentence S and aspect A |
| start, end | The starting and ending indices of aspect A in the sentence S |
| $dim_w, dim_h$ | The dimensions of word embedding and the hidden states |
| c | The sentiment polarity of the sentence $S$ towards the aspect $A$ |
| C | The collection of classes |
| $a_r$ | The representation of the aspect $A$ |
| $s$ | The representation of the sentence $S$ |

#### 1) RECNN

We first give a brief description of recursive neural network (RecNN) [56]. RecNN [56] is a class of architecture that can learn a directed acyclic graph structured input (e.g., a tree structure). As a generalization of the recurrent neural network [70], RecNN has a specific kind of tree structure. RecNN has been successfully employed to model compositionality in NLP via parse-tree-based structural representations, such as sentence-level sentiment analysis [71], [72] and paraphrase detection [73]. Based on RecNN and the parsing tree, Socher *et al.* [56] proposed a phrase-level sentiment analysis approach, where each node in the parsing tree was assigned a sentiment label. Given the structural directed acyclic graph of a sentence (e.g., a parse tree), RecNN visits the nodes in topological order and recursively generates parent representations in a bottom-up strategy, which combines tokens to generate representations for phrases, eventually the whole input sentence. Then the sentence representation is used to make a final classification (e.g., sentiment classification). Figure 3 presents an example process of vector composition in RecNN. The vector representation of node "very good" is generated from the vector representations of the node "very" and the node "good". Similarly, the node "not very good" is generated from the phrase node "very good" and the word node "not".

#### 2) TREE-BASED RECNN FOR ASC

RecNN was firstly applied into ASC by Dong *et al.* [23], they proposed an adaptive recursive neural network (AdaRNN) for

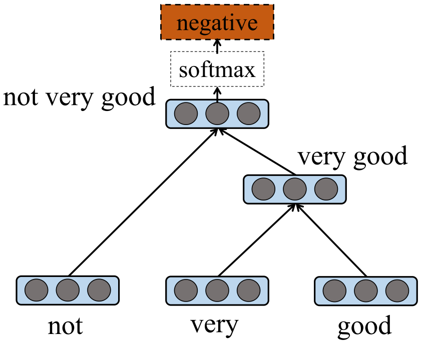


**FIGURE 3. The framework of the RecNN model.**

[2] https://code.google.com/archive/p/word2vec/

[3] https://github.com/stanfordnlp/GloVe

target-dependent twitter sentiment classification. AdaRNN learned to predict the sentiment polarities of words towards the aspect based on the context and syntactic structure. The representation of the root node was fed into the softmax classifier to predict the distribution of sentiment polarities. A binary dependency tree was built from a dependency tree of the sentence for a given sentence containing a target aspect. Intuitively, it represented syntactic relations associated with the aspect. Each word (leaf) or phrase (internal node) in the binary dependency tree was represented as a d-dimensional vector. The representation of a parent node v was computed by combining the left child vector representation $v_l$ and the right child vector representation $v_r$ from bottom to up via a global function g in RecNN:

$$v = f(g(v_l, v_r)) = f(W\begin{bmatrix} v_l \\ v_r \end{bmatrix} + b), \tag{1}$$

where $v_l$, $v_r$ were the vector representations of its left and right child, g and f were the composition function and the non-linearity function (e.g., tanh, sigmoid, softsign.) respectively. $W \in \mathbb{R}^{dim_w \times 2dim_w}$ was the parameter matrix and b denoted the bias vector.

Instead of using only a global function g, AdaRNN selected n compositional functions $G = \{g_1, \ldots, g_n\}$ based on the linguistic tags and combined vectors as follows:

$$v = f\left(\sum_{i=1}^{n} P(g_i|v_l, v_r, e)g_i(v_l, v_r)\right) \tag{2}$$

where $P(g_i|v_l, v_r, e)$ represented the probability of function $g_i$ given the external feature vector e and vector representations of child $v_l$, $v_r$. The probabilities were calculated as follows:

$$\begin{bmatrix} P(g_1|v_l, v_r, e) \\ \cdots \\ P(g_n|v_l, v_r, e) \end{bmatrix} = softmax\left(\beta R \begin{bmatrix} v_l \\ v_r \\ e \end{bmatrix}\right), \tag{3}$$

where $\beta \in \mathbb{R}$ was a hyper-parameter, and $R$ indicated the parameter matrix.

The vector representation of the root node of the binary dependency tree (as a representation of the target aspect) was fed to a softmax function to infer the sentiment polarity of the given aspect.

In addition, Nguyen and Shirai [24] proposed a PhraseRNN model to judge the sentiment of the sentence towards the given aspect, which demonstrated that the RecNN can obtain sentence representations from the recursive structure effectively. This model obtained the representation of an aspect from a "target dependent binary phrase dependency tree", which was constructed by a combination of the dependency and constituent trees. Different from AdaRNN, instead of using a list of global functions G, PhraseRNN used two types of composition functions $G = \{g_1, \ldots, g_n\}$ in inner-phrase and $H = \{h_1, \ldots, h_m\}$ in outer-phrase, where n and m were the number of functions in G and H, respectively. To be specific, the dependency tree was first transformed into a phrase dependency tree. Then the phrase dependency tree was transformed into a target dependent binary phrase dependency tree.

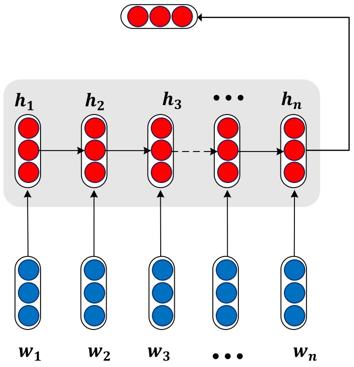


**FIGURE 4.** The framework of the RNN model.

However, these existing RecNN models for ASC may suffer from syntax parsing errors which were common in practice [22], [25].

### C. RNN FOR ASC

In this section, we describe the details of the RNN model, including standard RNN, LSTM, and GRU. We also review that most primitive methods about RNN for ASC, which can be divided into RNN based ASC, Bi-RNN based ASC and HRNN based ASC. Note that we only present the most classical RNN methods for ASC here for RNN is widely used in ASC. To better understand the RNN methods used in the existing work, we list the model type (such as LSTM, GRU, Bi-GRU, and Bi-LSTM) of each work in Table 1.

#### 1) RNN

We first provide a brief description of a basic recurrent neural network (RNN) [70] model. RNN models sequential inputs (e.g., sequences of words in a sentence) and the basic framework of an RNN is shown in Figure 4. RNN is a classical type of neural network that has recurrent connections, which allows a form of memory. This also makes it suitable for sequential prediction problems with arbitrary spatiotemporal dimensions. Thus, many NLP tasks adopt the structure of RNN by regarding the interpretation of a sentence as analyzing a sequence of tokens. Given a sentence $S$, we can obtain a sequential hidden states $H = [h_1, h_2, \ldots.h_n] \in R^{n \times dim_h}$ by feeding the input $X = [x_1, x_2, \ldots, x_n] \in R^{n \times dim_w}$ through RNN, where $dim_w$ and $dim_h$ denote the dimensions of word embedding and the hidden states respectively. In particular, RNN can be divided into the following three categories.

1) **Standard RNN** Standard RNN [70] is a basic framework of RNN. The transition function of standard RNN is a linear layer followed by a non-linear layer (e.g., tanh). The input of the network at time step $t$ is $x_t$ and $h_t$ represents the hidden state at the same time step. Calculation of $h_t$ is as follows:

$$h_t = f(Ux_t + Wh_{t-1}), \tag{4}$$

Thus, $h_t$ is calculated via the current input $x_t$ and the hidden state of previous time step $h_{t-1}$. The function f represents a non-linear transformation function (such as tanh, ReLU). U, V, W are standard RNN's weights that are shared across time. $x_t$ is the vector representation of words typically in NLP. As stated before, it can be considered as the network's memory element that accumulates information from other time steps. However, this standard RNN is really hard to learn and tune the parameters in practice since it suffers from the infamous vanishing gradient problem.

2) **LSTM** Long Short Term Memory (LSTM) network [74] is a special type of RNN, which is able to learn long-term dependencies. Similarly, the hidden layer $h_t$ at time step t is computed form a non-linear transformation function of the current input $x_t$ and the previous hidden state $h_{t-1}$. Then the output $y_t$ is calculated using the hidden state $h_t$. $h_t$ can be regarded as a representation summarizing the past, which is used to make a final decision on the current input. Apart from the hidden state vector, LSTM has a memory cell structure, which consists of three gates: an input gate, a forget gate and an output gate. The input gate is used to dictate the extent to which the memory cell will be influenced by the new input; the forget gate controls the extent to which previous information in the memory cell will be forgotten; and the output gate controls the extent to which the memory cell will influence the current hidden state. All three of these gates depend on the previous hidden state and the current input. Specifically, LSTM cell is calculated as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}; x_t] + b_i), \tag{5}$$

$$f_t = \sigma(W_f \cdot [h_{t-1}; x_t] + b_f), \tag{6}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}; x_t] + b_o), \tag{7}$$

$$g_t = tanh(W_r \cdot [h_{t-1}; x_t] + b_r), \tag{8}$$

$$c_t = i_t \odot g_t + f_t \odot c_{t-1}, \tag{9}$$

$$h_t = o_t \odot tanh(c_t), \tag{10}$$

where $\odot$ stands for element-wise multiplication and $\sigma$ is sigmoid function. $W_i$,$b_i$ are parameters of the input gate, $W_f$, $b_f$ are the parameters of the forget gate and $W_o$, $b_o$ are the parameters of the output gate. See Graves [75] and Greff *et al.* [76] for more details of LSTM.

3) **GRU** Gated Recurrent Unit (GRU) [59] is a more recent framework, which is similar to the LSTM model but simpler with fewer parameters. Empirically, GRU has observed to perform comparably to LSTM, despite its comparative simplicity [77]. Different from LSTM, instead of the memory cell, GRU uses an update gate to control how much the hidden gate will be updated, and a reset gate to control how the information is updated to the hidden state and control how much the previous hidden state will influence the current hidden state. The GRU state can be computed as follows:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{11}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \tag{12}$$

$$\hat{h}_t = tanh(W_h x_t + r_{t-1} \odot (U_h h_{t-2}) + b_h), \tag{13}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t, \tag{14}$$

where $W_z$,$U_z$, $b_z$, $W_r$, $U_r$, $b_r$ are the parameters of update and reset gates.

### 2) RNN BASED ASC

RNN plays a significant role in ASC. Tang *et al.* [26] first introduced LSTM into ASC for it can capture semantic relations between the aspect and its context words in a more flexible way. They proposed target-dependent LSTM (TD-LSTM) and target-connection LSTM (TC-LSTM) to extend LSTM by taking the aspect target into account. As shown in Figure 5, TD-LSTM learned representations from the left and right context with respect to the given aspect by making use of two LSTM networks, namely $LSTM_L$ and $LSTM_R$ respectively. After that, they concatenated the last hidden vectors of $LSTM_L$ and $LSTM_R$, and fed them to a softmax layer to predict the sentiment polarity of the sentence towards the aspect. To capture the interactions between the aspect and its contexts, TC-LSTM was proposed. It extended TD-LSTM by incorporating an aspect connection component, which explicitly utilized the connections between the aspect and each context word when constructed the representation of a sentence. The given aspect was regarded as a feature and was concatenated with the context features for ASC.

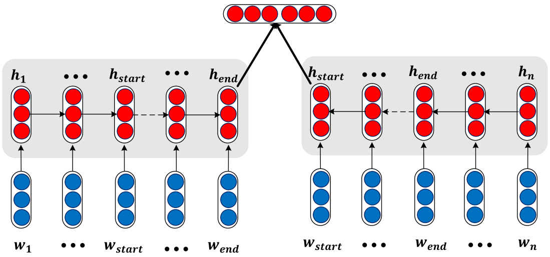


**FIGURE 5. The framework of the TD-LSTM model (From [26]).**

Ma *et al.* [27] incorporated commonsense knowledge of sentiment-related concepts into the end-to-end training of an LSTM model for ASC. The LSTM model was extended by integrating commonsense knowledge into gate mechanisms. They assumed that the sentiment concepts were meaningful to control the information of word-level information. For instance, a multi-word aspect "rotten fish" might suggest that the word "rotten" was a sentiment-related qualifier of the word "fish" so that less information need to be filtered out at the next time step. Thus, to filter the information, knowledge concepts were incorporated into the forget, input, and output gate of standard LSTM. The input gate used the sentiment concepts to prevent the memory cell from being affected by input tokens conflicting with knowledge. Similarly, such knowledge was utilized by the output gate to filter out the irrelevant information stored in the memory.

### 3) BI-RNN BASED ASC

Bidirectional RNN (Bi-RNN) is based on idea that output at time step t should depend on previous and future contents in a sentence. As shown in Figure 6, the Bi-RNN consists of two RNNs: a forward $\overrightarrow{RNN}$ which reads the sentence $S$ from $w_1$ to $w_n$ and a backward $\overleftarrow{RNN}$ from $w_n$ to $w_1$.

$$\overrightarrow{h}_i = \overrightarrow{RNN}(x_i, \theta_{RNN}), i \in [1, n] \quad (15)$$

$$\overleftarrow{h}_i = \overleftarrow{RNN}(x_i, \theta_{RNN}), i \in [n, 1] \quad (16)$$

where $\theta_{RNN}$ represents the parameters of the RNN model. The final context-aware representation for the word is obtained by concatenating the two hidden state vectors, namely $h_i = [\overrightarrow{h}_i, \overleftarrow{h}_i]$. As two typical categories of Bi-RNN, bidirectional GRU (Bi-GRU) and bidirectional LSTM (Bi-LSTM) [75] are widely used and achieved great success in ASC.

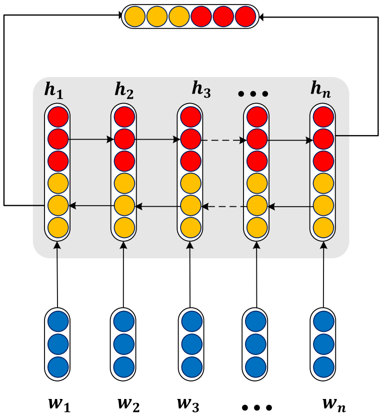


**FIGURE 6.** The framework of the Bi-RNN model.

A gated recurrent neural network (GRNN) was proposed by Zhang *et al.* [25] to model the syntax and semantics in the sentence and the interaction between the aspect and its surrounding contexts. The framework of GRNN is shown in Figure 7. This model adopted Bi-RNN (e.g., Bi-GRU) to overcome the weakness of pooling functions. To achieve that, two gated neural networks were presented. First, it leveraged a Bi-GRU to connect the words in a sentence so that pooling functions were applied over the hidden states instead of word embeddings for better representing the aspect and its contexts. Second, a three-way gated neural network structure was used to model the interaction between the aspect mentioned in the sentence and its surrounding contexts. Gated neural networks have been shown to reduce the bias of standard Bi-GRU towards the ends of a sentence by better propagation of gradients.

### 4) HRNN BASED ASC

Hierarchical RNN (HRNN) models have been used predominantly for representation learning of paragraphs and documents. Ruder *et al.* [28] also proposed to use a hierarchical bidirectional LSTM (H-LSTM) model for ASC, which was able to leverage both intra- and inter-sentence relations. As shown in Figure 8, word embeddings were fed into a sentence-level Bi-LSTM. Final hidden states of the forward LSTM and backward LSTM were concatenated together with the aspect embedding and fed into a review-level Bi-LSTM. At every time step, the outputs of the forward LSTM and backward LSTM was concatenated and fed into a softmax layer, which generated a probability distribution over sentiment polarities.

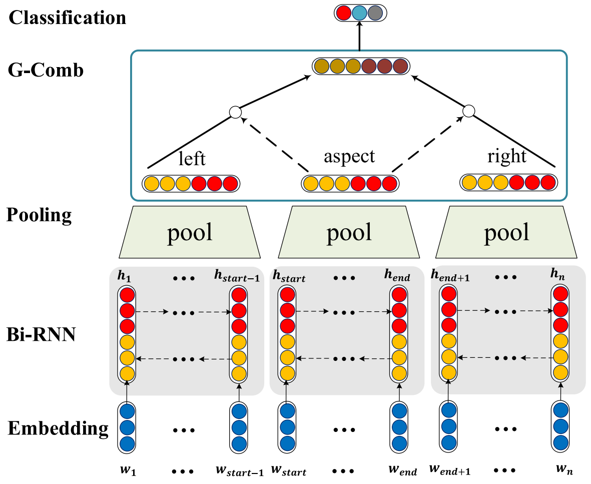


**FIGURE 7.** The framework of the GRNN model (From [25]).

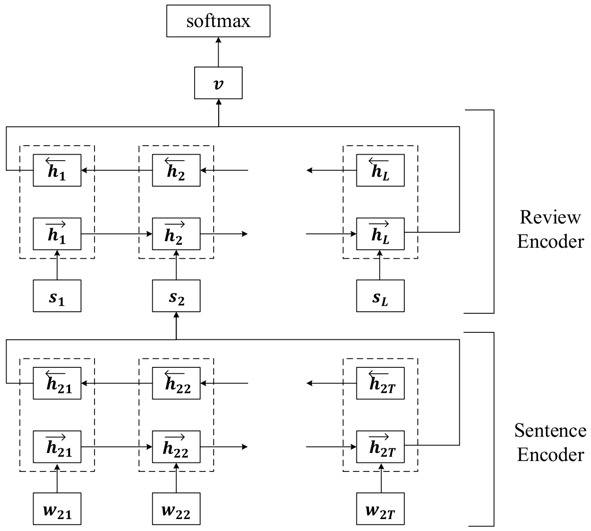


**FIGURE 8.** The framework of the H-LSTM model.

## D. ATTENTION-BASED RNN FOR ASC

In this section, we first introduce the standard attention-based RNN briefly. Then, we split the work of attention-based RNN for ASC into basic attention-based RNN for ASC and interactive attention-based RNN for ASC, and describe them in detail.

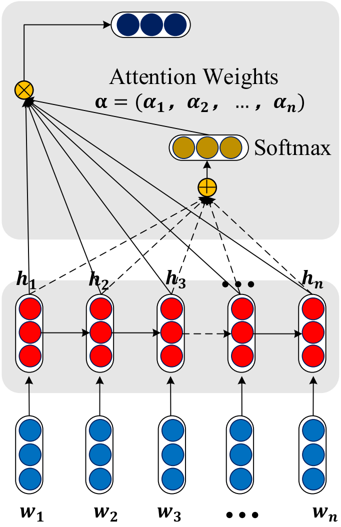


**FIGURE 9.** The framework of the attention-based RNN model.

### 1) ATTENTION-BASED RNN

Attention mechanism was first proposed in the neural machine translation task by Bahdanau *et al.* [59]. Since not all the information in the sequence is important, attention mechanism is proposed to enforce the RNN model to focus on the important parts of the sequence. The notion of attention has lately attracted a large amount of interest from neural networks researchers for its ability to capture the important parts of a text (in contrast, e.g., depending on the final hidden state vector). Attention mechanism has been successfully applied to many NLP tasks [58], such as neural machine translation [57], [59], question answering [60], [61], and machine comprehension [62], [63]. Figure 9 shows the framework of attention-based RNN. Specifically, the context vector is computed as a weighted sum of these annotations $h_i$:

$$s = \sum_i \alpha_i h_i \tag{17}$$

The weight $\alpha_i$ of each annotation $h_i$ is computed by:

$$\alpha_i = \frac{exp(score(h_i, a_r)}{\sum_j exp(score(h_j, a_r)} \tag{18}$$

Here, *score*() is referred as an aspect-aware function for which we consider three different alternatives [57]:

1) **Dot-Product Attention (DPA)**

$$score(h_i, a_r) = {h_i}^T a_r \tag{19}$$

2) **Concat Attention (CA)**

$$score(h_i, a_r) = v_a tanh(W_a[h_i; a_r]) \tag{20}$$

3) **General Attention (GA)**

$$score(h_i, a_r) = {h_i}^T W_a a_r \tag{21}$$

### 2) BASIC ATTENTION-BASED RNN FOR ASC

As mentioned earlier, most of the neural network models for ASC do not take into consideration the relationships between the specific aspect and its context words. Thus such models easily suffer from the semantic mismatching problem. To solve this problem, a series of attention-based neural network models have recently been proposed for they can automatically identify the relevant information with respect to a specific aspect in a sentence, which can be directly used for improving the quality of the features extracted by the neural network models [57]. Some of the representative basic attention-based RNN models proposed for ASC task are discussed below.

Wang *et al.* [29] proposed a single-hop attention based LSTM (named ATAE-LSTM) model with aspect embedding, which took the concatenations of the aspect representation and the word embeddings as input and the hidden states of LSTM were used for attention computation. Figure 10 shows the framework of ATAE-LSTM. For this model, "Concat Attention" was used to capture the important parts of the sentence towards the given aspect. It was proven to be an effective way to enforce the neural model to attend to the related part of a sentence in response to a specific aspect. Likewise, Yang *et al.* [30] proposed two kinds of attention-based bidirectional LSTM (AB-LSTM) models to improve classification performance. Zeng *et al.* [35] proposed a PosATT-LSTM model, which took the importance of context words into consideration and incorporated the position-aware vectors that represented the explicit position context between an aspect and its context words.

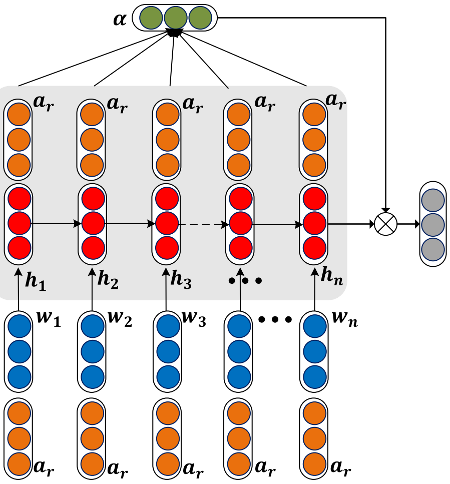


**FIGURE 10.** The framework of the ATAE-LSTM model (From [29]).

In addition, He *et al.* [36] transferred the knowledge from document-level sentiment classification dataset to ASC via pre-training and multi-task learning (PRET+MULT) based on attentive LSTM model. The existing ASC benchmark datasets were relatively small, which largely limited the performance of neural network models. Despite the lack of labeled ASC data, large-scale document-level sentiment classification labeled data were easily available online (e.g., Amazon and Yelp reviews). These reviews come with rating labels naturally and contained substantial linguistic patterns. The performance of ASC was improved by employing knowledge gained from document-level sentiment classification datasets. Li *et al.* [9] proposed a novel framework named Multi-Granularity Alignment Network (MGAN) to simultaneously align aspect granularity and aspect-specific feature representations across domains.

He *et al.* [31] proposed an approach to obtain better aspect representation by capturing the semantic information of the given aspect. Then, they incorporated syntactic information into the attention mechanism to obtain a better representation of the sentence. The framework of the proposed "LSTM + SynATT + TarRep" model is shown in Figure 11. The representation of each aspect was obtained by a mixture of m embeddings of aspect terms so that each embedded aspect can represent a combination of closely related aspect terms. An autoencoder structure was adopted to learn the aspect embeddings and the representation of the aspect which was a weighted summation of the aspect embeddings. Second, syntactic information was integrated into attention, namely a syntax-based attention model. In previous work, all words in a sentence were of equal importance for the attention models. Therefore, the attentive weight entirely depended on the semantic relationship between the specific aspect and its context words. However, it may not be sufficient to capture related opinion words for different aspects. Thus, a dependency parser was applied on the review sentence to obtain the syntactic path and then the syntax-based attention mechanism was designed to selectively capture the most related sentiment words that were close to the aspect on the syntactic path.

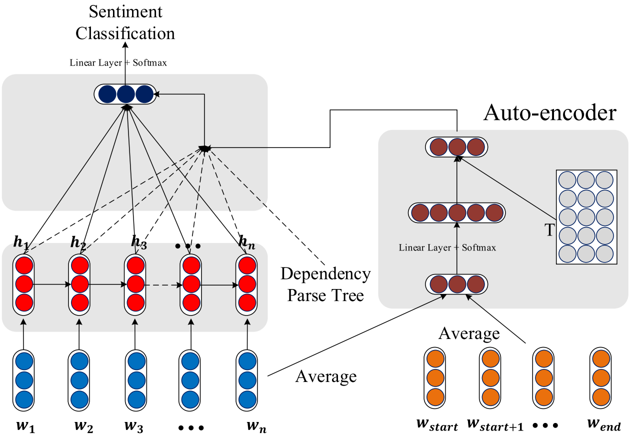


**FIGURE 11. The framework of the LSTM+SynATT+TarRep model (From [31]).**

Tay *et al.* [33] proposed an Aspect Fusion LSTM (AF-LSTM) model to integrate aspect information into the neural network model by modeling relationships of word-aspect. To capture the correct words towards a given aspect adaptively, AF-LSTM learned to attend based on associative relationships between the aspect and the sentence words. This addressed the limitations of other state-of-the-art methods that modeled word-aspect similarity via naive concatenations. Instead, to model the similarity between the aspect and its content words, this model developed circular convolution and circular correlation and incorporated them into a differentiable attention-based neural network model.

Hazarika *et al.* [37] predicted the sentiment polarities of all aspects in the same sentence to capture the inter-aspect dependencies and learned temporal dependency of their corresponding sentence representations utilizing RNN model. To be specific, the proposed model first inputted a sentence along with all of its aspects and then generated the sentence representations relative to each aspect to gain better aspect-aware representations [26]. An attention-based LSTM network was used for the attention mechanism enabled the model to capture key parts of the content words with regard to the given aspect. The same as [29], aspect representations were concatenated with each word embedding so that the attention mechanism can enable the model to capture aspect information. Finally, to capture the inter-aspect dependencies, the aspect-aware sentence representations were ordered as a sequence and fed into another LSTM to model the temporal dependency. Each time step of this LSTM corresponded to a specific aspect. Then the output of hidden state for each aspect was fed to a dense layer and a softmax layer to judge the sentiment polarities of each given aspect.

Wang *et al.* [32] adopted a hierarchical attention network model [78] for ASC. They proposed a hierarchical network with both word-level and clause-level attentions (namely Word&Clause-Level ATT) for aspect sentiment classification to take into account the importance degrees of both words and clauses inside a sentence. The overall architecture of the proposed model is shown in Figure 12. Specifically, they first utilized sentence-level discourse segmentation to divide a sentence into several clauses. Then, they leveraged one Bi-LSTM to model all clauses in the sentence and designed a word-level attention mechanism to capture the important words in each clause for not all the words inside a clause are meaningful. Finally, they adopted another Bi-LSTM to model the attentive representation of each clause and designed a clause-level attention mechanism to capture the important clauses in a sentence for not all the clauses in a sentence are meaningful.

### 3) INTERACITVE ATTENTION-BASED RNN FOR ASC

For this category of attention-based RNN methods for ASC, the interaction between the given aspect and its content words is taken into account. Ma *et al.* [38] proposed an Interactive Attention Network (IAN) that considered both attention mechanisms on the aspect and the full context. As shown in Figure 13, it used two attention-based LSTM to interactively capture the key words of the aspect terms and the important words of its context. Word embeddings of a given aspect and its context were inputted into two LSTM to obtain the hidden states of words respectively. To focus on the important information in the context and the specific

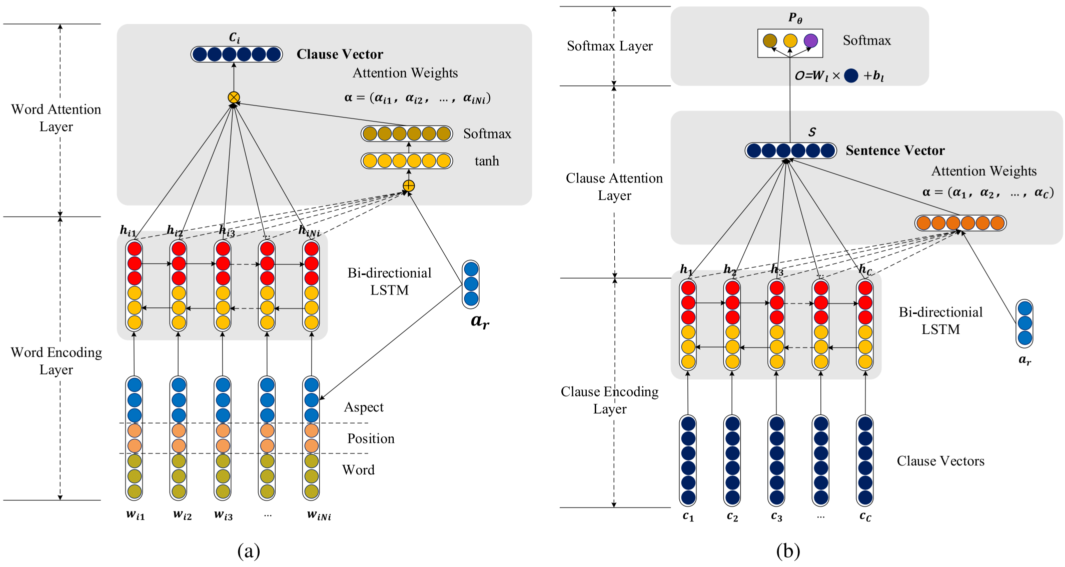


**FIGURE 12.** The frameworks of Word&Clause-Level ATT (From [78]). (a) Word-level Attention. (b) Clause-level Attention.

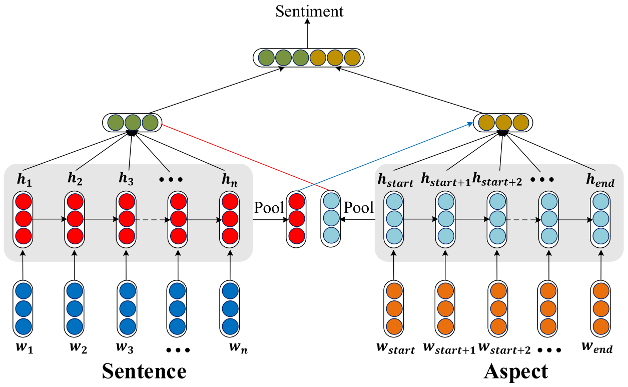

Sentiment
$h_1$ $h_2$ $h_3$ ••• $h_n$
Pool
Pool
$h_{start}$ $h_{start+1}$ $h_{start+2}$ ••• $h_{end}$
$w_1$ $w_2$ $w_3$ ••• $w_n$
Sentence
$w_{start}$ $w_{start+1}$ $w_{start+2}$ ••• $w_{end}$
Aspect

**FIGURE 13.** The framework of IAN model (From [38]).

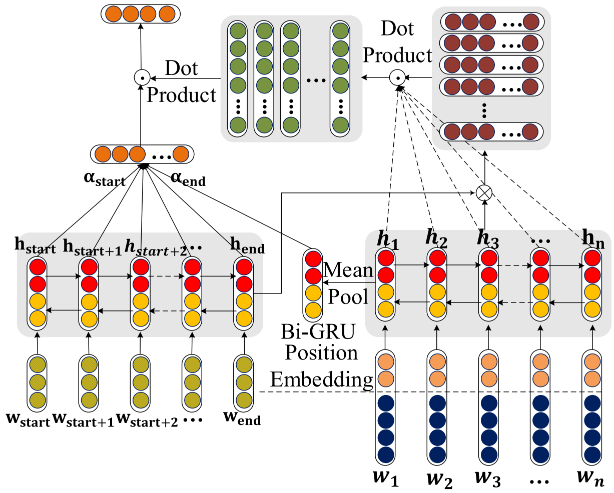

Dot Product
Dot Product
$\alpha_{start}$
$\alpha_{end}$
$h_{start}$ $h_{start+1}$ $h_{start+2}$ ... $h_{end}$
Mean Pool
Bi-GRU
Position Embedding
$h_1$ $h_2$ $h_3$ ... $h_n$
$w_{start}$ $w_{start+1}$ $w_{start+2}$ ... $w_{end}$
$w_1$ $w_2$ $w_3$ ... $w_n$

**FIGURE 14.** The framework of the PBAN model (From [41]).

aspect, the attention mechanism was adopted and the average values of the hidden states of the aspect and the hidden states of its context were adopted to guide the generation of attentive weights. Thus, the aspect and the full context can influence the generation of their representations interactively. Finally, the final representation of the sentence was obtained by concatenating the representations of the aspect and its context and inputted to a softmax layer for inferring the sentiment class.

To efficiently obtain the representation of the aspect especially when the aspect was multi-word and use the interaction among the aspect, its left context and its right context to focus on the key words in them, Zheng and Xia [43] proposed a left-center-right separated neural network with rotatory attention mechanism (LCR-Rot). Specifically, they developed a left-center-right separated LSTMs that consisted of three LSTMs (i.e., left-, center- and right- LSTM) to model the left context, aspect and right context of a sentence. Furthermore, a rotatory attention mechanism was introduced to take into consideration the interaction between aspects and its left/right contexts to better represent the given aspects and its contexts. They adopted a target2context attention to focus on the most related sentiment words in left/right contexts. At the same time, a context2target attention was designed to focus on the important words in the aspect so that a two-side representation of the aspect was obtained, namely left-aware aspect and right-aware aspect. Finally, the final representation of the sentence towards the given aspect obtained by concatenating the component representations was fed into a softmax function to predict the sentiment polarity.

Gu *et al.* [41] proposed a position-aware bidirectional attention network (PBAN) based on Bi-GRU. PBAN took the position information of aspect terms into account and mutually modeled the relevance between the aspect terms and the context by employing a bidirectional attention mechanism. To be specific, as shown in Figure 14, the proposed model

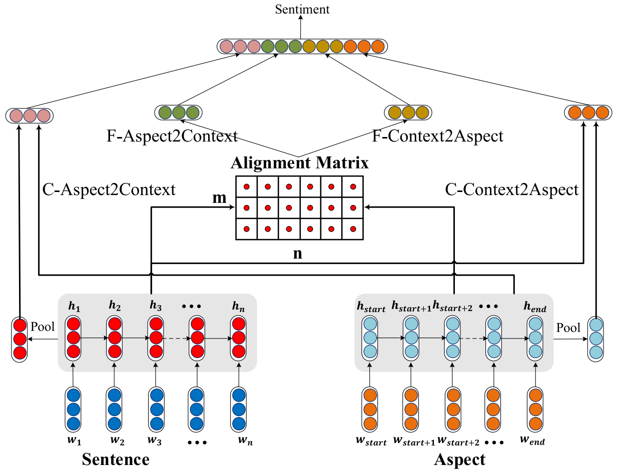

Sentiment
F-Aspect2Context
F-Context2Aspect
Alignment Matrix
C-Aspect2Context
C-Context2Aspect
m
n
Pool
$h_1$ $h_2$ $h_3$ ••• $h_n$
$h_{start}$ $h_{start+1}$ $h_{start+2}$ ••• $h_{end}$
Pool
$w_1$ $w_2$ $w_3$ ••• $w_n$
Sentence
$w_{start}$ $w_{start+1}$ $w_{start+2}$ ••• $w_{end}$
Aspect

**FIGURE 15.** The framework of MGAN model (From [40]).

consisted of three steps. First, the position information of the words in the sentence with respect to the given aspect term was obtained and converted into position embedding. Then, two Bi-GRU networks were adopted to extract the features of the specific aspect and its context respectively. Finally, the bidirectional attention mechanism was used to model the relevance between the aspect terms and its content words. Inspired by [79], [80], they appended position representation into word embedding to obtain the aspect-specific embedding.

Huang *et al.* [42] introduced an attention-over-attention (AOA) neural network for ASC to model the aspect and the sentence simultaneously to capture the interaction between the given aspect and its context words explicitly. Furthermore, the representations of the aspect and its context generated from LSTMs interacted with each other through the AOA module. It was observed that not all the words play a significant role in a sentence towards a given aspect. The opinion words in the sentence were highly relative to the specific aspect. Taking the sentence ''the appetizers are ok, but the service is slow.'' as an example, there were two aspects ''appetizers'' and ''service''. According to the language experience, the positive word ''ok'' described ''appetizers'' rather than the ''service''. The AOA module was introduced to generate mutual attentions from both aspect-to-context and context-to-aspect and capture the most important part of both the specific aspect and its corresponding context.

Fan *et al.* [40] proposed a fine-grained attention mechanism to model the interaction between the aspect and its context on the word-level. As shown in Figure 15, the MGAN framework consisted of two compositions, namely fine-grained attention mechanism and coarse-grained attention mechanism respectively. In particular, a fine-grained attention mechanism (i.e. F-Aspect2Context and F-Context2Aspect) was introduced to model interaction between the given aspect and its corresponding context words on the word-level, and reduce the information loss caused by coarse-grained attention mechanism. Furthermore, the bidirectional coarse-grained attention (i.e. C-Aspect2Context and C-Context2Aspect) were developed and combined with fine-grained attentive vectors to construct the MGAN framework for predicting the sentiment polarity of the sentence with respect to the given aspect. In addition, to utilize the aspect-level interaction information, an aspect alignment loss was adopted in the loss function to strengthen the difference of the attentive weights for the aspects in the same sentence which had different sentiment polarities.

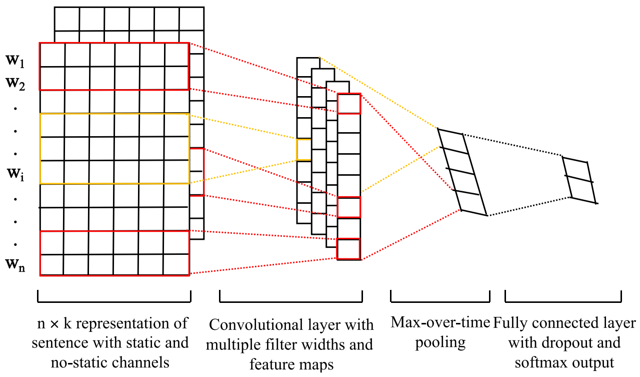

$w_1$
$w_2$
$w_i$
$w_n$
n × k representation of sentence with static and no-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

**FIGURE 16.** The framework of CNN model (From [65]).

Liu and Zhang [39] utilized the attention mechanism to calculate the importance level of each word with regard to sentiment polarities of the given aspect. They extended the attention mechanisms by enhancing the difference of the attentive weights obtained from the left and right contexts of a specific aspect. Multiple gates were introduced to further control the attention contribution. Specifically, a Bi-LSTM was adopted to model word embeddings over a sentence and then attention mechanism was employed over the hidden states to compute the contribution of each word in the sentence towards a specific aspect.

### *E. CNN FOR ASC*

In this section, a brief introduction of CNN is provided. Then we review the CNN based methods for ASC in detail.

#### 1) CNN

Convolution neural network (CNN) [81] is powerful in processing unstructured multimedia data with convolution and pool operations. CNN can be used for feature representation learning. It utilizes word embedding to map the sentence into a lower-dimensional semantic representation as well as maintain the sequences information of the words. The extracted representation of the sentence then passes through a convolutional layer with multiple filters, a max-pooling layer, and a fully-connected layer consecutively. Figure 16 presents the framework of CNN.

Specifically, let $x_{i:i+j}$ represents the concatenation of vectors $x_i, x_{i+1}, \ldots, x_j$. Convolution operation is performed on this input embedding layer. To generate a new feature, a filter

$k \in \mathbb{R}^{h \cdot dim_w}$ is adopted to a window of h words. For instance, a feature $c_i$ is calculated over the window of h words $x_{i:i+h-1}$ as follows:

$$c_i = f(x_{i:i+h-1} \cdot k^T + b) \tag{22}$$

Here $b \in \mathbb{R}$ is the bias term and f is a non-linear activation function (e.g., tanh, ReLU). The filter k is employed to all possible window of h words utilizing the same weights to produce the feature map.

$$c = [c_1, c_2, \ldots, c_{n-h+1}] \tag{23}$$

In a CNN, several kernels (also called convolutional filters) are used with different widths slide over the entire word embedding matrix X. Each kernel extracts a specific pattern of n-gram. After the convolution layer, a max-pooling strategy is usually adopted over the feature map and the maximum value $\hat{c} = max\{c\}$ is taken as feature corresponding to this particular kernel. The max operation on each kernel is adopted to subsample the input typically. The idea is to capture the most important n-gram feature - one with the highest value. This pooling strategy naturally solves variable sentence lengths by mapping the input to a fixed-size output.

#### 2) CNN BASED ASC

CNN was adopted for ASC for its ability to extract the local and global representations from text [45]–[47]. Huang *et al.* [45] incorporated aspect information into CNN by applying parameterized filters and parameterized gates. In particular, two simple CNN based models which incorporated aspect information were proposed. They introduced two neural units that took aspects into consideration, namely parameterized filter and parameterized gate. These units were designed for learning aspect-specific features. Then, two model variants Parameterized Filters for CNN (PF-CNN) and Parameterized Gated CNN (PG-CNN) were introduced.

Xue and Li [44] proposed a model based on CNN and gating mechanisms. The proposed model included two separate convolutional layers over the input embedding layer, whose outputs were combined by gating units. Convolutional layers with multiple filters were applied to generate n-gram features efficiently. Two non-linear gates were designed and connected to the two convolutional layers respectively. Given the aspect information, they selectively captured aspect-aware sentiment information for ASC. Since the proposed model could be easily paralleled, much less training time was costed than the models that were based on LSTM and attention mechanisms. When the aspect consisted of multiple words, another convolutional layer was adopted for obtaining the aspect representation.

Fan *et al.* [46] proposed a convolutional memory network for ASC which was inspired by the convolutional operation and based on the memory network. This model incorporated an attention mechanism to learn both words and multiple words information in the sentences. The proposed memory network was able to capture long-distance dependency by storing the context information into a fixed-size window at the same time.

Li *et al.* [47] adopted a method to scale the input of the convolutional layer with position information between the specific aspect and its context words. After re-examining the disadvantages of attention mechanism and the obstacles that block good performance of CNN, a TNet model was developed for ASC. Instead of the attention mechanism, a CNN layer was employed to generate important features from the hidden states obtained by the Bi-LSTM layer. To be specific, to incorporate aspect information into the representation of the word better, an aspect-aware transformation component was introduced. In addition, CNN was employed as the feature extractor, and the context-preserving and positional information were applied to overcome the disadvantages of CNN model.

### F. MEMORY NETWORK FOR ASC

In this section, we first introduce the detail of the memory network, which has been widely used in NLP. Then, we present an overview of the deep memory network for ASC.

#### 1) MEMORY NETWORK

Memory Network has achieved great success in NLP. Specifically, given a sentence $s = \{w_1, w_2, \ldots, w_n\}$ and the aspect words $\{w_{start}, w_{start+1}, \ldots, w_{end-1}, w_{end}\}$ , each word is mapped into its embedding vector. These word embedding vectors are divided into two parts, namely aspect representation and context representation. Aspect representation is the embedding of aspect word when the aspect is a single word (such as "food" and "service"). Aspect representation is the average value of its word embedding vectors when the aspect is a multi-word phrase (e.g., "battery life"). Context word vectors $\{m_1, m_2, \ldots, m_{start\text{âĹ}'1}, m_{end+1}, \ldots, m_n\}$ are stacked and converted into the external memory $m$. The internal state $u$ is set as the aspect representation. The match between $u$ and each memory $m_i$ is computed by taking the inner product followed by a softmax layer:

$$p_i = Softmax(u^T m_i). \tag{24}$$

where p is a probability vector over the inputs.

The output vector from the memory o is then a summation of the transformed inputs $m_i$, weighted sum by the probability vector from the input:

$$o = \sum_i p_i m_i. \tag{25}$$

Then the model is extended to handle multi-hop operations. The memory layers are stacked as follows: the input to layers above the first $u_{k+1}$ is calculated as the sum of the output $o_k$ and the input $u_k$ of layer k:

$$u_{k+1} = u_k + o_k \tag{26}$$

#### 2) MEMORY NETWORK BASED ASC

Tang *et al.* [19] first developed a deep memory network based on a multi-hop attention mechanism for ASC, which

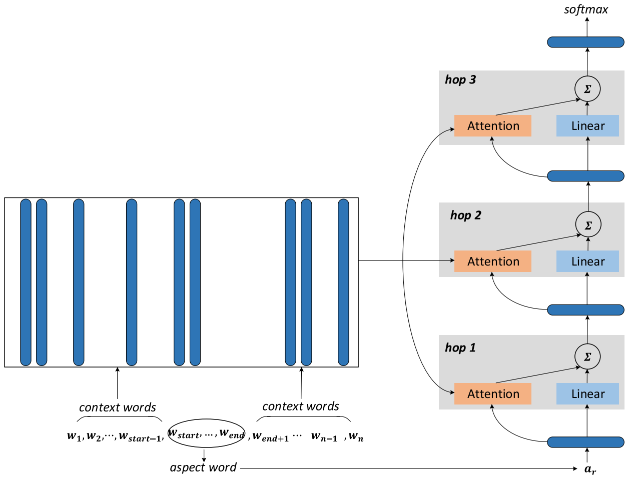


**FIGURE 17.** The framework of memory network model (From [19]).

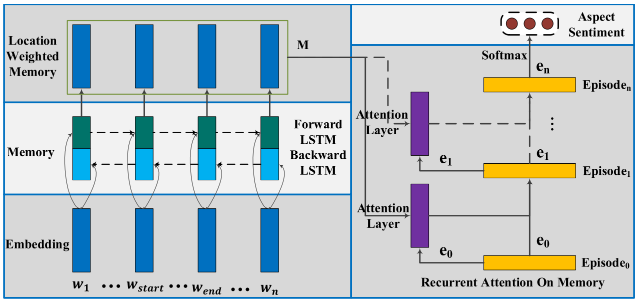


**FIGURE 18.** The framework of RAM model (From [49]).

was effective and computationally inexpensive. As shown in Figure 17, it adopted an multi-hop attention mechanism over an external memory to focus on the importance level of the context words w.r.t. the given aspect. The proposed approach explicitly captured the important information of context words for inferring the sentiment polarity of the specific aspect. Such importance degree and text representation were calculated by multiple computational layers, which were attention-base neural models with an external memory.

Tay *et al.* [48] introduced a dyadic memory network (DyMemNN) to capture rich dyadic interactions between the given aspect and its context words by incorporating parameterized neural tensor compositions and holographic compositions into the memory selection operation. Two kinds of dyadic memory networks, namely the Tensor DyMemNN and Holo DyMemNN were developed to focus on rich dyadic interaction between the aspect and the sentence.

Chen *et al.* [49] proposed a recurrent attention mechanism based on a memory network (RAM) for each aspect to extract sentiment information separated by a long distance. As shown in Figure 18, to achieve that, the proposed model utilized a recurrent/dynamic attention structure and learned a nonlinear combination of the attention in GRUs. However, attention weights were calculated only based on local sentence information due to the character of the LSTM employed in the proposed model.

In [55], they proposed a context attention mechanism for ASC, which explicitly took into account the correlation between the given aspect and each context word. This model consisted of two attention-enhancing mechanisms, namely sentence-level content attention mechanism and content attention mechanism. Sentence-level content attention mechanism was able to capture the important information of the sentence towards the given aspect from a global perspective and overcome the short-sight problem of the deep memory network model.

To better model interaction between aspect and sentiment, Li and Lam [66] incorporated the aspect detection task into sentiment prediction task. They achieved sentiment identification via an end-to-end method, in which two tasks were learned simultaneously via a deep memory network. In such a way, signals generated in aspect detection provided feedback for sentiment classification, and reversely, the predicted polarity provided clues to the aspects identification.

In addition, Wang *et al.* [53] proposed target-sensitive memory networks (TMNs) for ASC. TMNs can capture the sentiment interaction between the aspect and its context words. In addition, six techniques were introduced to construct the TMNs. Majumder *et al.* [52] presented a method which integrated the neighboring aspects related information into predicting the sentiment polarity of the aspect via memory networks.

## IV. DATASETS

In this section, we describe the standard datasets for ASC in detail. The most popular datasets were released by the international workshop on semantic evaluation for the aspect-level sentiment analysis task [3]–[5], namely SemEval 2014 [3], SemEval 2015 [4] and SemEval 2016 [5]. In addition, datasets such as Twitter [23], Sentihood [93] and Mitchell [100] are also used for ASC. An overview of statistics on the usage of each standard dataset is shown in Table 3. We summarize the representative publications of each dataset in this table. It is evident from the Table 3, as far as the data source is concerned, a lot of work has been done on SemEval 2014 [3], SemEval 2015 [4], SemEval 2016 [5] and Twitter [23]. At the same time, a small number of papers used Sentihodd [93], Mitchell [100], MPQA [101], tripAdvisor [98] and other datasets. It is notable that all datasets are ground truth data from different domains (e.g., Restaurant, Laptop, Twitter). To make it easier for researchers, we collect all benchmark datasets for everyone to study. In order to unify the format, we translate all datasets into the XML form as shown in Fig. 19. In addition, we explore all the datasets and describe them in detail in the following sections.

### A. SEMEVAL 2014

SemEval 2014 task4[4] [3] is concerned with aspect based sentiment analysis and the goal of this task is to detect the aspects

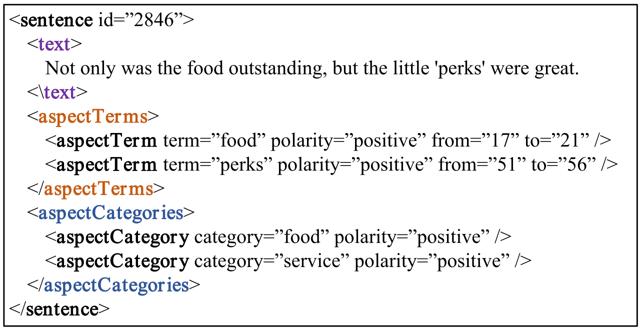
[4] http://alt.qcri.org/semeval2014/task4/

**TABLE 3.** The Datasets of ASC.

| Dataset | Publications |
|---|---|
| SemEval-2014 Task4 | [43], [55], [46], [42], [83], [51], [31], [41], [47], [53], [37], [36], [44], [45], [33], [38], [49], [48], [34], [84], [29], [19], [24], [18], [85], [3], [52], [9], [35], [86], [87] |
| SemEval-2015 Task12 | [32], [31], [36], [27], [33], [48], [34], [4], [88] |
| SemEval-2016 Task5 | [83], [31], [36], [48], [34], [5], [89], [90], [28], [91] |
| Twitter | [43], [55], [46], [47], [49], [92], [39], [30], [26], [25], [22], [23], [9], [86] |
| MPQA | [39], [25], [93] |
| Hindi | [83], [27] |
| SentiHood | [94], [95] |
| Mitchell | [39], [25], [96] |
| tripAdvisor | [97], [98], [99] |
| openTable | [100] |

**TABLE 4.** The statistical information of SemEval-2014 Task4: Reataurant14, Laptop14. #Samples, #AvgLen, #TermSet, #AvgTermLen, #ATPS represent the number of samples, the average length of samples, the number of term set, the average length of the term and the average number of terms for each sample respectively. Neg./Neu./Pos. indicate the number of negative samples, neutral samples, and positive samples.

| Dataset | | #Samples | #AvgLen | #TermSet | #AvgTermLen | #ATPS | Neg./Neu./Pos. |
|---|---|---|---|---|---|---|---|
| Restaurants14 | train | 1,978 | 16.2856 | 1,191 | 2.0722 | 1.8210 | 805/633/2,164 |
| | test | 600 | 15.4167 | 520 | 1.9942 | 1.8667 | 196/196/728 |
| Laptop14 | train | 1,462 | 18.5855 | 939 | 1.9191 | 1.5821 | 866/460/987 |
| | test | 411 | 14.9562 | 389 | 1.9434 | 1.5523 | 128/169/341 |



```
<sentence id=”2846”>
  <text>
    Not only was the food outstanding, but the little 'perks' were great.
  <\text>
  <aspectTerms>
    <aspectTerm term=”food” polarity=”positive” from=”17” to=”21” />
    <aspectTerm term=”perks” polarity=”positive” from=”51” to=”56” />
  </aspectTerms>
  <aspectCategories>
    <aspectCategory category=”food” polarity=”positive” />
    <aspectCategory category=”service” polarity=”positive” />
  </aspectCategories>
</sentence>
```

**FIGURE 19.** The unified format of the above datasets.

of the given target entities and determine the sentiment polarity expressed upon each aspect. There are two domain-specific datasets for laptops and restaurants, namely Restaurants14 and Laptop14, consisting of over 6,000 sentences with aspect-level human-authored labels for training. To be specific, each single or multi-word aspect term is assigned one of the following polarities based on the sentiment that is expressed in the sentence towards it: (1) positive; (2) negative; (3) neutral (means neither positive nor negative sentiment). (4) conflict (means both positive and negative sentiment). We remove the data with conflict sentiment polarity and Table 4 shows the statistical information of Restaurants14 and Laptop14. We report the details of datasets Restaurants14 and Laptop14 in the following sections.

### 1) RESTAURANTS14

Restaurants14 consists of over 3,000 English sentences extracted from the restaurant reviews/comments of Ganu *et al.* [102] as the training dataset. Extra reviews of the restaurant are labeled in the same way as the test dataset. After removing the data with conflict sentiment polarity or without aspect term, there are 1,978 training samples and 600 test samples remained. The dataset includes annotations for coarse aspect categories, aspect terms, aspect term-specific polarities, and aspect category-specific polarities. From Table 4, we observe that the average number of the aspects in the same sentence is about 1.8 and the average length of the aspect is about 2. These indicate that one sentence usually contains more than one aspect and the aspect usually contain more than one words. It is also notable that the sample number of each class is unbalanced in this data. In addition, the top-10 aspect terms of this dataset are shown in Table 5.

**TABLE 5.** The top-10 aspect terms of Restaurants14.

| | Restaurants14 | | | |
|---|---|---|---|---|
| | Train | | Test | |
| 1 | food | 360 | food | 125 |
| 2 | service | 225 | service | 74 |
| 3 | prices | 63 | menu | 22 |
| 4 | place | 59 | atmosphere | 21 |
| 5 | dinner | 56 | staff | 21 |
| 6 | staff | 55 | place | 19 |
| 7 | menu | 55 | prices | 18 |
| 8 | pizza | 51 | meal | 14 |
| 9 | atmosphere | 46 | sushi | 14 |
| 10 | price | 41 | drinks | 13 |

### 2) LAPTOP14

This dataset consists of over 3,000 English sentences obtained from customer laptops reviews. Part of this dataset is divided as test data. After removing the data with conflict sentiment polarity or without aspect term, there are 1462 training samples and 411 test samples remained. The dataset only includes annotations for aspect terms of the sentences and

**TABLE 6. The top-10 aspect terms of Laptop14.**

| | Laptop14 | | | |
|---|---|---|---|---|
| | Train | | Test | |
| 1 | screen | 60 | performance | 15 |
| 2 | price | 56 | price | 15 |
| 3 | use | 53 | works | 14 |
| 4 | battery life | 52 | os | 13 |
| 5 | keyboard | 50 | features | 11 |
| 6 | battery | 47 | windows 8 | 11 |
| 7 | programs | 37 | use | 9 |
| 8 | features | 35 | screen | 9 |
| 9 | software | 33 | size | 8 |
| 10 | warranty | 31 | keyboard | 7 |

**TABLE 7. The top-10 aspect terms of Restaurants15.**

| | Restaurants15 | | | |
|---|---|---|---|---|
| | Train | | Test | |
| 1 | null | 375 | null | 248 |
| 2 | food | 158 | food | 74 |
| 3 | service | 117 | place | 49 |
| 4 | place | 82 | service | 30 |
| 5 | restaurant | 29 | restaurant | 20 |
| 6 | staff | 27 | staff | 12 |
| 7 | pizza | 26 | waiter | 10 |
| 8 | atmosphere | 21 | waitress | 8 |
| 9 | sushi | 20 | atmosphere | 7 |
| 10 | decor | 16 | meal | 7 |

their polarities. Similarly, from Table 4, it is observed that one review usually consists of more than one aspects, the aspect usually consists of more than one words, and the sample number of each class is unbalanced for Laptop14. All these observations largely limit the performance of the deep learning model. Furthermore, Table 6 shows the top-10 aspect terms of Laptop14.

## B. SEMEVAL 2015

SemEval-2015 task12[5] [4] is a continuation of SemEval-2014 task4. The goal of this task is to identify all the aspects and their overall polarities. In particular, the input datasets of SemEval 2015 task12 contain entire reviews rather than isolated sentences. For training, two datasets of about 500 reviews of restaurants and laptops annotated with aspects and their polarities are provided. For test dataset, additional datasets are provided. Since the laptop dataset does not contain the aspect term information, we process the restaurant dataset as Restaurants15.

### 1) RESTAURANTS15

This dataset consists of 254 and 96 restaurant reviews annotated with aspects and their sentiment polarities for training and testing respectively. Each review may contain multiple sentences, and each sentence includes annotations for category, aspect term and aspect term polarity. After removing the data of conflict sentiment polarity, there are 1,120 sentences for training and 582 for testing. Here we show the statistical information of the dataset Restaurants15 at sentence-level in Table 8. We find that the sample number for sentiment negative, neutral and positive is 749, 98 and 1,652 respectively, which shows the unbalance of each class. Note that the sentence in Restaurants15 usually contains multiple aspects and the aspect usually contains multiple words. Moreover, Table 7 and Table 9 show the top-10 aspects and aspect categories of Restaurants15 respectively.

## C. SEMEVAL 2016

SemEval-2016 task5[6] [5] is similarly to the SemEval-2015 task12, the dataset consists of entire reviews. In addition, the dataset contains five domains and covers eight languages. Participants are free to choose the languages and domains as they wish, here we consider English dataset of restaurant domain, namely Restaurants16.

### 1) RESTAURANTS16

This dataset consists of 350 restaurant reviews annotated with aspect terms, aspect categories and polarities for training and 92 for testing. After removing the data with conflict sentiment polarity, there are 1,708 annotated sentences for training and 587 for testing. Detailed statistical information can be seen in Table 10. The similar as Restauarnts14, Laptop14, and Restaurants15, the sample number of each class is unbalanced in Restaurants16. Also, multiple aspects are given in one sentence and multiple words compose the aspect in most case. In addition, we present the top-10 aspects and aspect categories of Restaurants16 in Table 11 and Table 12 respectively.

## D. TWITTER

Dong *et al.* [23] introduced a manually annotated dataset for target-dependent twitter sentiment analysis. This is the largest target-dependent twitter sentiment classification dataset which is annotated manually. The training data has 6,248 tweets, and the testing data consists of 692 tweets with a sentiment class balance of 25% negative, 50% neutral and 25% positive. As shown in Figure 20, the original corpus has only annotated one target per tweet. The detailed statistical information is shown in Table 13. Different from the datasets mentioned above, one tweet only contains one aspect and the sample number of each sentiment is relatively balanced, even though an aspect also usually consists of multiple words. Furthermore, Table 14 reports the top-10 aspects of dataset Twitter.

## E. OTHERS

### 1) MITCHELL

Mitchell dataset[7] [100] consists of about 30,000 Spanish tweets and 10,000 English tweets labeled for named entities (NE) in Begin, Inside, Outside (BIO) encoding as shown in Figure 21: the start of an NE is labeled with B-NE and

[5] http://alt.qcri.org/semeval2015/task12/

[6] http://alt.qcri.org/semeval2016/task5/

[7] http://www.m-mitchell.com/code/index.html

**TABLE 8.** The statistical information of SemEval-2015 Task12: Restaurants15. #Samples, #AvgLen, #TermSet, #AvgTermLen, #ATPS represent the number of samples, the average length of samples, the number of term set, the average length of the term and the average number of terms for each sample respectively. Neg./Neu./Pos. indicate the number of negative samples, neutral samples, and positive samples.

| Dataset | #Samples | #AvgLen | #TermSet | #AvgTermLen | #ATPS | Neg./Neu./Pos. |
|---|---|---|---|---|---|---|
| train | 1,120 | 13.1009 | 492 | 2.0163 | 1.4768 | 403/53/1,198 |
| test | 582 | 14.3728 | 252 | 1.8968 | 1.4519 | 346/45/454 |

**TABLE 9.** The top-10 aspect categories of Restaurants15.

| | Train | | Test | |
|---|---|---|---|---|
| 1 | FOOD#QUALITY | 581 | FOOD#QUALITY | 271 |
| 2 | RESTAURANT#GENERAL | 269 | SERVICE#GENERAL | 175 |
| 3 | SERVICE#GENERAL | 268 | RESTAURANT#GENERAL | 147 |
| 4 | AMBIENCE#GENERAL | 183 | AMBIENCE#GENERAL | 77 |
| 5 | FOOD#STYLE_OPTIONS | 93 | FOOD#STYLE_OPTIONS | 40 |
| 6 | RESTAURANT#MISCELLANEOUS | 62 | RESTAURANT#MISCELLANEOUS | 38 |
| 7 | FOOD#PRICES | 54 | RESTAURANT#PRICES | 35 |
| 8 | RESTAURANT#PRICES | 48 | FOOD#PRICES | 31 |
| 9 | DRINKS#QUALITY | 34 | DRINKS#QUALITY | 12 |
| 10 | DRINKS#STYLE_OPTIONS | 26 | LOCATION#GENERAL | 8 |

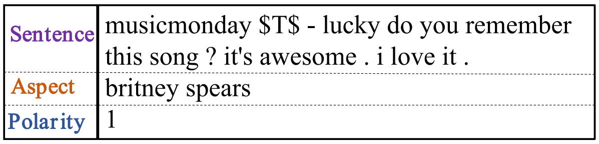
**TABLE 10.** The statistical information of SemEval-2016 Task5: Restaurants16. #Samples, #AvgLen, #TermSet, #AvgTermLen, #ATPS represent the number of samples, the average length of samples, the number of term set, the average length of the term and the average number of terms for each sample respectively. Neg./Neu./Pos. indicate the number of negative samples, neutral samples, and positive samples.

| Dataset | #Samples | #AvgLen | #TermSet | #AvgTermLen | #ATPS | Neg./Neu./Pos. |
|---|---|---|---|---|---|---|
| train | 587 | 13.5427 | 671 | 2.0596 | 1.4678 | 749/101/1,657 |
| test | 587 | 13.4957 | 289 | 1.8581 | 1.4634 | 204/44/611 |

**TABLE 11.** The top-10 aspect terms of Restaurants16.

| | Restaurants16 | | | |
|---|---|---|---|---|
| | Train | | Test | |
| 1 | null | 627 | null | 209 |
| 2 | food | 233 | food | 69 |
| 3 | service | 148 | service | 44 |
| 4 | place | 129 | place | 32 |
| 5 | restaurant | 49 | sushi | 21 |
| 6 | staff | 40 | restaurant | 15 |
| 7 | pizza | 31 | atmosphere | 11 |
| 8 | atmosphere | 28 | pizza | 11 |
| 9 | sushi | 26 | menu | 10 |
| 10 | decor | 23 | staff | 9 |

the rest of the NE is labeled with I-NE. 7,105 Spanish tweets contained 9,870 name entities and 2,350 English tweets contained 3,577 name entities after removing retweets. To obtain sentiment labels (positive, negative, or no sentiment), crowdsourcing was used through Amazon's Mechanical Turk. For 10-fold cross-validation, the English data is divided into folds. The statistics of English dataset of Mitchell is shown in Table 15 and the top-10 aspect terms are shown in Table 16. More details of this dataset are described in [100].



| | |
|---|---|
| Sentence | musicmonday $T$ - lucky do you remember this song ? it's awesome . i love it . |
| Aspect | britney spears |
| Polarity | 1 |

**FIGURE 20.** The origin format of Twitter.

### 2) SENTIHOOD

SentiHood [93] is a benchmark dataset that is annotated for the targeted aspect-based sentiment analysis task in the domain of urban neighborhoods. It is based on the questions relating to neighborhoods of the city London, which is obtained by filtering the text from Yahoo! Answers' question answering platform. SentiHood consists of 5,215 sentences with 3862 sentences containing a single location and 1,353 sentences containing two locations. Figure 22 shows an example of Sentihood with JSON format. Location entity names are masked by location1 and location2 in the whole dataset, so this task does not involve the named entities identification. The more details of Sentihood is presented by Saeidi *et al.* [93].

### 3) MPQA

MPQA[8] [101] contains news articles and other text documents annotated for opinions and other states (such as emotions, beliefs, sentiments and speculations.). In MPQA 3.0, the entity-target and event-target (eTarget) annotations are added. Note that the previous span-based target annotations in MPQA 2.0 are retained in this new corpus, which are renamed as sTarget (span-based target). In particular, the current dataset contains 70 documents, which consists of 1,029 expressive subjective elements (ESEs), 1,287 attitudes, and 1,213 target spans of attitudes from MPQA 2.0. In addition, 1,366 eTargets are added to the ESEs and 1,608 eTargets are added to the target spans.

[8] http://mpqa.cs.pitt.edu/corpora/

**TABLE 12.** The top-10 aspect categories of Restaurants16.

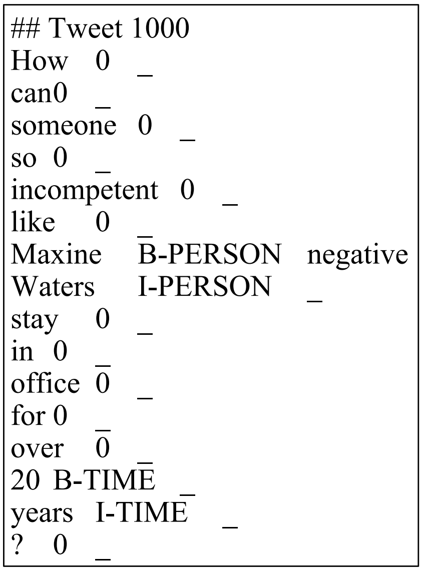
| | Train | | Test | |
|---|---|---|---|---|
| 1 | FOOD#QUALITY | 849 | FOOD#QUALITY | 313 |
| 2 | SERVICE#GENERAL | 449 | SERVICE#GENERAL | 155 |
| 3 | RESTAURANT#GENERAL | 422 | RESTAURANT#GENERAL | 142 |
| 4 | AMBIENCE#GENERAL | 255 | AMBIENCE#GENERAL | 66 |
| 5 | FOOD#STYLE_OPTIONS | 137 | FOOD#STYLE_OPTIONS | 55 |
| 6 | RESTAURANT#MISCELLANEOUS | 98 | RESTAURANT#MISCELLANEOUS | 33 |
| 7 | FOOD#PRICES | 90 | FOOD#PRICES | 23 |
| 8 | RESTAURANT#PRICES | 80 | DRINKS#QUALITY | 22 |
| 9 | DRINKS#QUALITY | 47 | RESTAURANT#PRICES | 21 |
| 10 | DRINKS#STYLE_OPTIONS | 32 | LOCATION#GENERAL | 13 |

**TABLE 13.** The statistical information of dataset Twitter. #Samples, #AvgLen, #TermSet, #AvgTermLen, #ATPS represent the number of samples, the average length of samples, the number of term set, the average length of the term and the average number of terms for each sample respectively. Neg./Neu./Pos. indicate the number of negative samples, neutral samples, and positive samples.

| Dataset | #Samples | #AvgLen | #TermSet | #AvgTermLen | #ATPS | Neg./Neu./Pos. |
|---|---|---|---|---|---|---|
| train | 6,248 | 18.8078 | 113 | 1.7965 | 1.0 | 1,560/3127/1,561 |
| test | 692 | 18.8671 | 82 | 1.8049 | 1.0 | 173/346/173 |

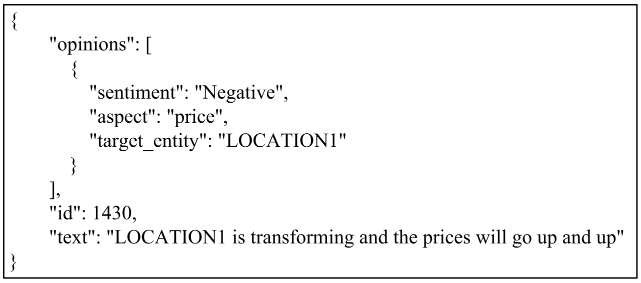
**TABLE 14.** The top-10 aspect terms of Twitter.

| | Twitter | | | |
|---|---|---|---|---|
| | Train | | Test | |
| 1 | britney spears | 924 | britney spears | 96 |
| 2 | lindsay lohan | 400 | lindsay lohan | 44 |
| 3 | harry potter | 324 | harry potter | 36 |
| 4 | madonna | 233 | ipod | 25 |
| 5 | barack obama | 222 | lady gaga | 24 |
| 6 | wii | 195 | windows 7 | 24 |
| 7 | sarah palin | 191 | barack obama | 23 |
| 8 | lady gaga | 181 | psp | 20 |
| 9 | ipod | 180 | wii | 19 |
| 10 | windows 7 | 171 | madonna | 19 |



```
## Tweet 1000
How    0   _
can0   _
someone   0    _
so  0   _
incompetent   0   _
like    0   _
Maxine   B-PERSON   negative
Waters   I-PERSON    _
stay    0   _
in  0   _
office  0   _
for 0   _
over    0   _
20  B-TIME   _
years   I-TIME   _
?   0   _
```

**FIGURE 21.** An example of dataset Mitchell with BIO encoding.



```
{
    "opinions": [
      {
        "sentiment": "Negative",
        "aspect": "price",
        "target_entity": "LOCATION1"
      }
    ],
    "id": 1430,
    "text": "LOCATION1 is transforming and the prices will go up and up"
}
```

**FIGURE 22.** An example of dataset Sentihood.

## V. EVALUATION MEASURES

How to evaluate the model is another important problem in ASC. In the field of ASC, there are no recognized evaluation measures. The authors always utilize different evaluation measures in different papers, making it hard for comparison. In this section, we present the public metrics for ASC in detail. Statistics on the usage of each metric are shown in Table 17. Some measures that are most-used to compare and evaluate the classification method mainly include: 1) Accuracy; 2) Precision and recall; 3) F-measure; 4) Macro average and micro average. The details of these metrics are given as follows.

### A. ACCURACY

Accuracy is the most basic evaluation measure of classification. The evaluation measure accuracy represents the proportion of the correct predictions of the model, it can be calculated as:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+FP+TN+FN} \\ &= \frac{TP+TN}{N}, \end{aligned} \tag{27}$$

where $N$ is the total number of testing samples. $TP$ and $TN$ are true predictions for positive, negative examples respectively, $FP$ and $FN$ mean false predictions for negative, positive examples respectively, which are described in Table 18.

**TABLE 15.** The statistical information of dataset Mitchell-en. #Samples, #AvgLen, #TermSet, #AvgTermLen, #ATPS represent the number of samples, the average length of samples, the number of term set, the average length of the term and the average number of terms for each sample respectively. Neg./Neu./Pos. indicate the number of negative samples, neutral samples, and positive samples.

| #Samples | #AvgLen | #TermSet | #AvgTermLen | #ATPS | Neg./Neu./Pos. |
|---|---|---|---|---|---|
| 2,350 | 18.4404 | 2,355 | 1.7970 | 1.3715 | 269/2,259/695 |

**TABLE 16.** The top-10 aspect terms of Mitchell-en.

| | Mitchell-en | | | |
|---|---|---|---|---|
| | Train | | Test | |
| 1 | facebook | 133 | facebook | 133 |
| 2 | twitter | 113 | twitter | 113 |
| 3 | google | 37 | apple | 37 |
| 4 | apple | 37 | google | 37 |
| 5 | youtube | 25 | youtube | 25 |
| 6 | un | 21 | un | 21 |
| 7 | obama | 16 | obama | 16 |
| 8 | forex | 14 | forex | 14 |
| 9 | amazon | 11 | reuters | 11 |
| 10 | reuters | 11 | amazon | 11 |

Though accuracy can be a good measure of the effectiveness of a classifier in most cases, once the positive and negative examples are uneven the high accuracy does not necessarily mean good classification performance. Therefore, precision, recall, and F-measure are to be introduced.

### B. PRECISION AND RECALL

Classification effectiveness is usually evaluated in terms of precision and recall. The precision is the proportion of correct predictions among all predictions with the positive label, it indicates how many of the instances that are positively predicted are true positive instances. The regular precision is calculated as:

$$Precision = \frac{TP}{TP + FP}. \tag{28}$$

The recall is the proportion of correct predictions among all positive instances, it denotes how many of positive instances are predicted positively. The regular recall is calculated as:

$$Recall = \frac{TP}{TP + FN}. \tag{29}$$

### C. F-MEASURE

The metrics precision and recall are a reciprocal relationship. The purpose of classification is to obtain precision and recall. F-measure is the harmonic mean of precision and recall, the traditional F-measure is computed as:

$$\begin{aligned} F &= \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \\ &= 2\frac{Recall \times Precision}{Recall + Precision} \\ &= \frac{2 \times TP}{2 \times TP + FP + FN}, \end{aligned} \tag{30}$$

which is also known as $F1$ measure since here the weight of precision and recall is equal. It is also a specific case of general $F_\beta$.

$$F_\beta = \left(1 + \beta^2\right) \cdot \frac{precision \times recall}{\beta^2 \cdot precision + recall}, \tag{31}$$

where $\beta$ is a non-negative real value.

### D. MACRO AVERAGE AND MICRO AVERAGE

Evaluation measures apart from accuracy mentioned previously are enough to evaluate the effectiveness of two-class classification tasks, while ASC is a multi-label classification problem for there are more than three possible values for sentiment polarity. Precision, recall and F1 are aimed at a class with only local significance. Hence, we need to calculate precision and recall for each class, then apply corresponding macro-average and micro-average as final result respectively.

#### 1) MACRO-AVERAGE

The macro-average measures take evaluations of each class into consideration. The macro precision and macro recall are computed as:

$$\begin{aligned} MacroPrecision &= \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i}{TP_i + FN_i} \\ &= \frac{1}{|C|}\sum_{i=1}^{|C|} P_i, \end{aligned} \tag{32}$$

$$\begin{aligned} MacroRecall &= \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{TP_i}{TP_i + FP_i} \\ &= \frac{1}{|C|}\sum_{i=1}^{|C|} R_i, \end{aligned} \tag{33}$$

where $|C|$ means the amount of classes, $P_i$, $R_i$ is corresponding precision, recall of class i respectively. The corresponding $F1$ measure, used in [32], [46], is computed as:

$$Macro - F1 = \frac{2 \times MacroPrecision \times MacroRecall}{MacroPrecision + MacroRecall}. \tag{34}$$

#### 2) MICRO-AVERAGE

The micro-average evaluation measures focus on each sample of the dataset. The micro precision and micro recall are computed as:

$$MicroPrecision = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FN_i}, \tag{35}$$

$$MicroRecall = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} TP_i + FP_i}, \tag{36}$$

**TABLE 17.** **The evaluation measures of ASC.**

| Metrics | Publications |
|---|---|
| Accuracy | [43], [55], [46], [42], [83], [32], [51], [31], [41], [47], [37], [36], [44], [27], [33], [38], [49], [92], [39], [34], [30], [5], [29], [19], [28], [94], [26], [91], [25], [22], [24], [24], [93], [18], [85], [3], [23], [98], [100], [17], [45], [52], [9], [35], [86], [87], [95] |
| Macro-F1 | [46], [32], [51], [31], [47], [53], [36], [27], [92], [39], [48], [5], [26], [25], [22], [23], [9], [54], [86], [87] |
| Precision/Recall/F1 | [51], [30], [89], [90], [84], [88], [24], [96], [93], [3], [98], [97], [100], [104], [99] |
| Micro-F1 | [53], [27], [4] |

**TABLE 18.** **The contingency table.**

| | | True Label | |
|---|---|---|---|
| | | Yes | No |
| Classifier Label | Yes | TP | FP |
| | No | FN | TN |

The corresponding $F1$ measure of micro-averaging is computed as:

$$Micro - F1 = \frac{2 \times MicroPrecision \times MicroRecall}{MicroPrecision + MicroRecall}. \quad (37)$$

## VI. EXPERIMENT IMPLEMENTATION

In this section, we first describe the datasets and the evaluation measures in Section VI-A. Then, the implementation details are shown in VI-B. After that, we present the implemented methods in Section VI-C. Finally, the experimental results and analyses are introduced in Section VI-D.

### A. DATASETS AND EVALUATION MEASURES

As shown in Table 19, we statistic the experimental results of almost all the existing methods. We find that most of the experimental results of various public metrics (e.g., Accuracy, Macro-F1) on the most-used benchmark datasets (e.g., Restaurants14, Laptop14, Restaurants15, Restaurants16, and Twitter) are missing. In addition, unfortunately, some existing work did not use development dataset. Thus, for better comprehension of the reported performances, we implement several classical public state-of-the-art baselines and release the codes on the website. To be specific, we run the experiments for most-used benchmark datasets (e.g., Restaurants14, Laptop14, Restaurants15, Restaurants16, and Twitter) with various metrics (e.g., Accuracy, Precision, Recall, F1, Marco-average and Micro-average). Note that we use the training set and test set released by the providers [3]–[5], [23] for a fair comparison. We randomly sample 10% from the original training data as the development data which is used to tune algorithm parameters. Accuracy and Macro-Average Precision/Recall/F1, Micro-Average Precision/Recall/F1, Precision, Recall, F1 are adopted to evaluate the model performance, which are the primary metrics used in ASC [36], [47].

### B. IMPLEMENTATION DETAILS

In our experiments, we show the details of the configurations and used hyper-parameters in Table 20. In particular, word embedding vectors are initialized with 300-dimension GloVe [69] vectors and fine-tuned during the training, which are the same as [19]. The dimension of hidden state vectors and position embedding are 300 and 100 respectively. Words out of vocabulary GloVe [69] and weight matrices are initialized with the uniform distribution $U(-0.1, 0.1)$, and the biases are initialized to zero. Adam [104] is adopted as the optimizer. For the multiple-hops models, we set the hop number to 3 following the previous study [61]. To avoid overfitting, dropout is used in our training model and we search the best dropout rate from 0.4 to 0.7 with an increment of 0.1. We obtain the best hyper-parameter learning rate and mini-batch size from $\{0.001, 0.0005\}$ and $\{4, 8, 16, 32\}$ respectively via grid search. We implement our neural networks with Pytorch.[9] We keep the optimal parameters based on the best performance on the development set and the optimal model is used for evaluation in the test set.

### C. IMPLEMENTED METHODS

The classical state-of-the-art methods implemented by us are as follows:

- **ContextAvg:** the average of the word embeddings is fed to a softmax layer for sentiment prediction, which was adopted as a baseline in [19].
- **AEContextAvg:** the concatenation of the average of the word embeddings and the average of the aspect vectors is fed to a softmax layer for sentiment prediction, which was adopted as a baseline in [19].
- **LSTM:** the last hidden vector obtained by LSTM [74] is used for sentence representation and sentiment prediction.
- **GRU:** the last hidden vector obtained by GRU [59] is used for sentence representation and sentiment prediction.
- **BiLSTM:** the concatenation of last hidden vectors obtained by BiLSTM is used for sentence representation and sentiment prediction.

[9] https://pytorch.org/

**TABLE 19.** The results obtained from the published papers.

| Method | Restaurants14 | | Laptop14 | | Restaurants15 | | Restaurants16 | | Twitter | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | Marco-F1 | Accuracy | Marco-F1 | Accuracy | Marco-F1 | Accuracy | Marco-F1 | Accuracy | Marco-F1 |
| RecNN for ASC | | | | | | | | | | |
| AdaRNN | - | - | - | - | - | - | - | - | 66.30 | 65.90 |
| PhraseRNN | 66.20 | - | - | - | - | - | - | - | - | - |
| RNN for ASC | | | | | | | | | | |
| GRNN | - | - | - | - | - | - | - | - | - | - |
| TD-LSTM | - | - | - | - | - | - | - | - | 70.80 | 69.00 |
| TC-LSTM | - | - | - | - | - | - | - | - | 71.50 | 69.50 |
| AE-LSTM | 76.60 | - | 68.90 | - | - | - | - | - | - | - |
| H-LSTM | - | - | - | - | - | - | - | - | - | - |
| Attention-based RNN for ASC | | | | | | | | | | |
| ATAE-LSTM | 77.20 | - | 68.70 | - | - | - | - | - | - | - |
| AB-LSTM | - | - | - | - | - | - | - | - | 72.60 | 72.20 |
| BILSTM-ATT-G | - | - | - | - | - | - | - | - | 73.60 | 72.10 |
| IAN | 78.60 | - | 72.10 | - | - | - | - | - | - | - |
| AF-LSTM(CONV) | 75.44 | - | 68.81 | - | - | - | - | - | - | - |
| HEAT | - | - | - | - | - | - | - | - | - | - |
| Sentic LSTM+TA+SA | - | - | - | - | - | - | - | - | - | - |
| MGAN | 81.49 | 71.48 | 76.21 | 71.42 | - | - | - | - | 74.62 | 73.53 |
| PosATT-LSTM | 79.40 | - | 72.80 | - | - | - | - | - | - | - |
| PRET+MULT | 79.11 | 79.73 | 71.15 | 67.46 | 81.30 | 68.74 | 85.58 | 79.76 | - | - |
| PBAN | 81.16 | - | 74.12 | - | - | - | - | - | - | - |
| LSTM+SynATT+TarRep | 80.63 | 71.32 | 71.94 | 69.23 | 81.67 | 66.05 | 84.61 | 67.45 | - | - |
| MGAN | 81.25 | 71.94 | 75.39 | 72.47 | - | - | - | - | 72.54 | 70.81 |
| Inter-Aspect Dependencies | 79.00 | - | 72.50 | - | - | - | - | - | - | - |
| AOA-LSTM | 81.20 | - | 74.50 | - | - | - | - | - | - | - |
| LCR-Rot | 81.34 | - | 75.24 | - | - | - | - | - | 72.69 | - |
| Word&Clause-Level ATT | - | - | - | - | 80.90 | 68.50 | - | - | - | - |
| CNN for ASC | | | | | | | | | | |
| GCAE | 77.28 | - | 69.14 | - | - | - | - | - | - | - |
| PF-CNN | 79.20 | - | 70.06 | - | - | - | - | - | - | - |
| Conv-Memnet | 78.26 | 68.38 | 76.37 | 72.10 | - | - | - | - | 72.11 | 70.80 |
| TNet | 80.69 | 71.27 | 76.54 | 71.75 | - | - | - | - | 74.97 | 73.60 |
| Memory Network for ASC | | | | | | | | | | |
| MemNet | 80.95 | - | 72.21 | - | - | - | - | - | - | - |
| DyMemNN | - | 58.82 | - | 60.11 | - | - | - | - | - | - |
| RAM | 80.23 | 70.80 | 74.49 | 71.35 | - | - | - | - | 69.36 | 73.85 |
| CEA | 80.98 | - | 72.88 | - | - | - | - | - | - | - |
| DAuM | 82.32 | 71.45 | 74.45 | 70.16 | - | - | - | - | 72.14 | 60.24 |
| IARM | 80.00 | - | 73.8 | - | - | - | - | - | - | - |
| TMNs | - | 68.84 | - | 67.23 | - | - | - | - | - | - |
| Cabasc | 80.89 | - | 75.07 | - | - | - | - | - | 71.53 | - |

- **BiGRU:** the concatenation of last hidden vectors obtained by BiGRU is used for sentence representation and sentiment prediction.
- **TD-LSTM:** a target-dependent LSTM model which modeled the preceding and following contexts surrounding the target for sentiment classification [26].

- **TC-LSTM:** this model extends TD-LSTM by incorporating a target connection component, which explicitly utilizes the connections between target word and each context word when composing the representation of a sentence. [26].
- **AT-LSTM:** it uses an LSTM to model the sentence and a basic attention mechanism is applied for sentence representation and sentiment prediction. [29].
- **AT-GRU:** it uses a GRU to model the sentence and a basic attention mechanism is applied for sentence representation and sentiment prediction.
- **AT-BiLSTM:** it uses a BiLSTM to model the sentence and a basic attention mechanism is applied for sentence representation and sentiment prediction.
- **AT-BiGRU:** it uses a BiGRU to model the sentence and a basic attention mechanism is applied for sentence representation and sentiment prediction.
- **ATAE-LSTM:** the aspect representation is integrated into attention-based LSTM for sentence representation and sentiment prediction [29].
- **ATAE-GRU:** the aspect representation is integrated into attention-based GRU for sentence representation and sentiment prediction.
- **ATAE-BiLSTM:** the aspect representation is integrated into attention-based BiLSTM for sentence representation and sentiment prediction.
- **ATAE-BiGRU:** the aspect representation is integrated into attention-based BiGRU for sentence representation and sentiment prediction.
- **IAN:** the attention mechanisms in the context and aspect were learned interactively for context and aspect representation [38].
- **LCRS:** it contains three LSTMs, i.e., left-, center- and right- LSTM, respectively modeling the three parts of a review (left context, aspect and right context) [43].
- **CNN:** The sentence representation obtained by CNN [81] is used for ASC.
- **GCAE:** it has two separate convolutional layers on the top of the embedding layer, whose outputs are combined by gating units [44].
- **MemNet:** the content and position of the aspect is incorporated into a deep memory network [19].
- **RAM:** a multi-layer architecture where each layer contains an attention-based aggregation of word features and a GRU cell to learn the sentence representation [49].
- **CABASC:** two novel attention mechanisms, namely sentence-level content attention mechanism and context attention mechanism are introduced in a memory network to tackle the semantic-mismatch problem [55].

### D. EXPERIMENTAL RESULTS AND ANALYSES

Table 21, Table 22, Table 23, Table 24 and Table 25 report the experimental results of classical state-of-the-art methods across Restaurants14, Laptop14, Restaurants15, Restaurants16 and Twitter respectively. We implement the typical RNN, attention-based RNN, CNN and memory network based state-of-the-art methods for ASC and evaluate these models in terms of accuracy, macro-average (Marco-Precision, Marco-Recall, Marco-F1), micro-average (Mirco-Precision, Mirco-Recall, Mirco-F1) and precision, recall, F1 for each class (negative, neutral and positive).

**TABLE 20.** The details of the configurations and used hyper-parameters.

| | |
|---|---|
| Word embedding | Glove [69] |
| Dimension of word embedding | 300 |
| Dimension of hidden state vectors | 300 |
| Dimension of position embedding | 100 |
| Initializer of weight matrices | Uniform distribution U(-0.1, 0.1) |
| Initializer of weight matrices | zero |
| Initializer of words out vocabulary | Uniform distribution U(-0.1, 0.1) |
| Optimizer | Adam [105] |
| Multiple-hops | 3 |
| Dropout | Search from {0.4, 0.5, 0.6, 0.7} |
| Learning Rate | Search from {0.001, 0.00005} |
| Mini-batch size | Search from {4, 8, 16, 32} |
| Deep learning framework | PyTorch |

From the tables, the following observations are found: **1)** The models taking aspect representation into account always perform better than the ones without considering aspect representation. In particular, we find that AEContextAvg usually performs better than ContextAvg and TC-LSTM outperforms basic LSTM model in most cases. **2)** The performance of BiRNN is better than the corresponding RNN model. For example, BiLSTM and BiGRU outperform LSTM and GRU respectively. **3)** Attention mechanism can improve the performance of the models effectively. For example, AT-LSTM and AT-GRU perform better than LSTM and GRU respectively in most cases. **4)** The simple attention-based BiRNN usually obtains good performance. It is observed that the performance of the AT-BiLSTM is comparable to the best results across Restaurants14, Laptop14, Restaurants15, and Restaurants16. **5)** As a state-of-the-art model, IAN always perform well, which indicates the interactive attention mechanism can capture the important information of the aspect and the sentence effectively. **6)** For datasets Restaurants14, Laptop14, Restaurants15, and Restaurants16, the classes are unbalanced, especially for the neutral samples, which have a side effect on the performance of ASC. For example, for datasets Restaurants15 and Restaurants16, the F1 of the neutral samples is near to 0.0 for most of the models. This problem largely limits the performance of the existing models. **7)** RNN models always outperform CNN models for ASC. For instance, the performance of GRU is better than CNN over all five datasets in terms of accuracy.

## VII. FUTURE DIRECTIONS AND CHALLENGES

As we can see from our discussion before, the existing work has established a solid foundation for deep learning based ASC research. In this section, we will present several promising future research directions and discuss some of the most challenging open problems.

**TABLE 21.** Experimental results of Restaurants14.

| | Accuracy | Macro | | | Micro | | | Precision | | | Recall | | | F1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 | Precision | Recall | F1 | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. |
| ContextAvg | 73.48 | 62.92 | 58.44 | 59.58 | 73.48 | 73.48 | 73.48 | 56.48 | 51.79 | 80.49 | 55.61 | 29.59 | 90.11 | 56.04 | 37.66 | 85.03 |
| AEContextAvg | 75.27 | 66.30 | 61.47 | 63.10 | 75.27 | 75.27 | 75.27 | 62.09 | 55.47 | 81.36 | 57.65 | 36.22 | 90.52 | 59.79 | 43.83 | 85.70 |
| LSTM | 77.23 | 67.54 | 64.34 | 65.51 | 77.23 | 77.23 | 77.23 | 63.35 | 54.55 | 84.73 | 61.73 | 39.80 | 91.48 | 62.53 | 46.02 | 87.98 |
| GRU | 78.75 | 70.51 | 65.61 | 67.11 | 78.75 | 78.75 | 78.75 | 67.36 | 59.84 | 84.35 | 66.33 | 37.24 | 93.27 | 66.84 | 45.91 | 88.58 |
| BiGRU | 77.14 | 67.61 | 63.55 | 65.15 | 77.14 | 77.14 | 77.14 | 64.94 | 53.69 | 84.19 | 57.65 | 40.82 | 92.17 | 61.08 | 46.38 | 88.00 |
| BiLSTM | 78.30 | 69.11 | 66.01 | 67.12 | 78.30 | 78.30 | 78.30 | 65.13 | 56.64 | 85.55 | 64.80 | 41.33 | 91.90 | 64.96 | 47.79 | 88.61 |
| TD-LSTM | 78.66 | 70.84 | 67.56 | 68.98 | 78.66 | 78.66 | 78.66 | 72.88 | 54.55 | 85.09 | 65.82 | 45.92 | 90.93 | 69.17 | 49.86 | 87.92 |
| TC-LSTM | 77.41 | 69.06 | 65.18 | 66.72 | 77.41 | 77.41 | 77.41 | 67.78 | 55.70 | 83.69 | 62.24 | 42.35 | 90.93 | 64.89 | 48.12 | 87.16 |
| AT-LSTM | 78.04 | 70.84 | 61.52 | 63.37 | 78.04 | 78.04 | 78.04 | 70.06 | 61.25 | 81.23 | 63.27 | 25.00 | 96.29 | 66.49 | 35.51 | 88.12 |
| AT-GRU | 78.30 | 70.74 | 64.76 | 66.58 | 78.30 | 78.30 | 78.30 | 67.91 | 61.21 | 83.11 | 64.80 | 36.22 | 93.27 | 66.32 | 45.51 | 87.90 |
| AT-BiGRU | 77.77 | 69.51 | 64.74 | 66.18 | 77.77 | 77.77 | 77.77 | 65.13 | 59.84 | 83.56 | 64.80 | 37.24 | 92.17 | 64.96 | 45.91 | 87.66 |
| AT-BiLSTM | 78.84 | 72.84 | 63.67 | 65.66 | 78.84 | 78.84 | 78.84 | 68.45 | 67.82 | 82.27 | 65.31 | 30.10 | 95.60 | 66.84 | 41.70 | 88.44 |
| ATAE-GRU | 76.79 | 68.68 | 63.49 | 65.32 | 76.79 | 76.79 | 76.79 | 69.49 | 54.62 | 81.92 | 62.76 | 36.22 | 91.48 | 65.95 | 43.56 | 86.44 |
| ATAE-LSTM | 76.79 | 67.93 | 62.74 | 63.72 | 76.79 | 76.79 | 76.79 | 64.53 | 57.00 | 82.25 | 66.84 | 29.08 | 92.31 | 65.66 | 38.51 | 86.99 |
| ATAE-BiGRU | 76.34 | 65.95 | 63.26 | 63.82 | 76.34 | 76.34 | 76.34 | 63.77 | 50.41 | 83.67 | 67.35 | 31.63 | 90.80 | 65.51 | 38.87 | 87.09 |
| ATAE-BiLSTM | 75.98 | 67.01 | 61.71 | 63.43 | 75.98 | 75.98 | 75.98 | 66.29 | 53.28 | 81.46 | 60.20 | 33.16 | 91.76 | 63.10 | 40.88 | 86.30 |
| IAN | 76.70 | 68.29 | 63.69 | 65.12 | 76.70 | 76.70 | 76.70 | 64.25 | 58.06 | 82.57 | 63.27 | 36.73 | 91.07 | 63.75 | 45.00 | 86.61 |
| LCRS | 76.25 | 68.71 | 60.85 | 63.03 | 76.25 | 76.25 | 76.25 | 69.82 | 56.44 | 79.88 | 60.20 | 29.08 | 93.27 | 64.66 | 38.38 | 86.06 |
| CNN | 75.18 | 68.45 | 58.44 | 60.25 | 75.18 | 75.18 | 75.18 | 60.44 | 65.79 | 79.12 | 56.12 | 25.51 | 93.68 | 58.20 | 36.76 | 85.79 |
| GCAE | 77.41 | 68.58 | 64.80 | 65.06 | 77.41 | 77.41 | 77.41 | 64.86 | 57.43 | 83.44 | 73.47 | 29.59 | 91.35 | 68.90 | 39.06 | 87.21 |
| MemNet | 73.39 | 62.74 | 61.13 | 61.09 | 73.39 | 73.39 | 73.39 | 52.56 | 52.38 | 83.29 | 62.76 | 33.67 | 86.95 | 57.21 | 40.99 | 85.08 |
| RAM | 77.41 | 68.38 | 65.67 | 66.76 | 77.41 | 77.41 | 77.41 | 67.20 | 53.25 | 84.68 | 64.80 | 41.84 | 90.38 | 65.97 | 46.86 | 87.44 |
| CABASC | 77.68 | 69.01 | 67.18 | 68.02 | 77.68 | 77.68 | 77.68 | 65.59 | 55.68 | 85.75 | 62.24 | 50.00 | 89.29 | 63.87 | 52.69 | 87.48 |

**TABLE 22.** Experimental results of Laptop14.

| | Accuracy | Macro | | | Micro | | | Precision | | | Recall | | | F1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 | Precision | Recall | F1 | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. |
| ContextAvg | 66.93 | 63.47 | 59.98 | 58.19 | 66.93 | 66.93 | 66.93 | 46.41 | 67.65 | 76.35 | 65.62 | 27.22 | 87.10 | 54.37 | 38.82 | 81.37 |
| AEContextAvg | 66.46 | 61.64 | 59.56 | 58.04 | 66.46 | 66.46 | 66.46 | 47.40 | 61.54 | 75.97 | 64.06 | 28.40 | 86.22 | 54.49 | 38.87 | 80.77 |
| LSTM | 66.14 | 62.37 | 60.20 | 55.35 | 66.14 | 66.14 | 66.14 | 48.08 | 62.79 | 76.23 | 78.12 | 15.98 | 86.51 | 59.52 | 25.47 | 81.04 |
| GRU | 67.71 | 64.31 | 61.50 | 58.60 | 67.71 | 67.71 | 67.71 | 49.47 | 66.67 | 76.80 | 73.44 | 23.67 | 87.39 | 59.12 | 34.93 | 81.76 |
| BiGRU | 69.44 | 65.61 | 63.83 | 61.49 | 69.44 | 69.44 | 69.44 | 49.22 | 67.11 | 80.49 | 74.22 | 30.18 | 87.10 | 59.19 | 41.63 | 83.66 |
| BiLSTM | 68.81 | 63.41 | 63.56 | 62.09 | 68.81 | 68.81 | 68.81 | 50.28 | 59.05 | 80.90 | 69.53 | 36.69 | 84.46 | 58.36 | 45.26 | 82.64 |
| TD-LSTM | 68.50 | 62.66 | 62.98 | 61.87 | 68.50 | 68.50 | 68.50 | 47.70 | 57.63 | 82.66 | 64.84 | 40.24 | 83.87 | 54.97 | 47.39 | 83.26 |
| TC-LSTM | 67.08 | 62.02 | 62.66 | 61.11 | 67.08 | 67.08 | 67.08 | 46.52 | 57.76 | 81.79 | 67.97 | 39.64 | 80.35 | 55.24 | 47.02 | 81.07 |
| AT-LSTM | 69.44 | 64.23 | 65.02 | 63.16 | 69.44 | 69.44 | 69.44 | 51.91 | 58.88 | 81.90 | 74.22 | 37.28 | 83.58 | 61.09 | 45.65 | 82.73 |
| AT-GRU | 70.85 | 66.57 | 66.21 | 63.58 | 70.85 | 70.85 | 70.85 | 54.21 | 64.63 | 80.87 | 80.47 | 31.36 | 86.80 | 64.78 | 42.23 | 83.73 |
| AT-BiGRU | 69.28 | 64.44 | 64.36 | 63.28 | 69.28 | 69.28 | 69.28 | 48.86 | 62.61 | 81.84 | 67.19 | 42.60 | 83.28 | 56.58 | 50.70 | 82.56 |
| AT-BiLSTM | 71.94 | 66.36 | 66.80 | 66.42 | 71.94 | 71.94 | 71.94 | 55.48 | 59.06 | 84.55 | 63.28 | 52.07 | 85.04 | 59.12 | 55.35 | 84.80 |
| ATAE-GRU | 69.75 | 64.43 | 63.46 | 62.45 | 69.75 | 69.75 | 69.75 | 52.76 | 61.22 | 79.31 | 67.19 | 35.50 | 87.68 | 59.11 | 44.94 | 83.29 |
| ATAE-LSTM | 67.40 | 65.16 | 62.18 | 58.47 | 67.40 | 67.40 | 67.40 | 47.39 | 69.64 | 78.44 | 78.12 | 23.08 | 85.34 | 59.00 | 34.67 | 81.74 |
| ATAE-BiGRU | 70.38 | 67.00 | 66.20 | 64.12 | 70.38 | 70.38 | 70.38 | 49.25 | 68.82 | 82.95 | 76.56 | 37.87 | 84.16 | 59.94 | 48.85 | 83.55 |
| ATAE-BiLSTM | 70.53 | 66.84 | 65.99 | 63.43 | 70.53 | 70.53 | 70.53 | 50.75 | 67.47 | 82.30 | 78.91 | 33.14 | 85.92 | 61.77 | 44.44 | 84.07 |
| IAN | 68.50 | 64.11 | 62.69 | 60.90 | 68.50 | 68.50 | 68.50 | 51.12 | 63.41 | 77.78 | 71.09 | 30.77 | 86.22 | 59.48 | 41.43 | 81.78 |
| LCRS | 66.46 | 63.15 | 60.84 | 59.50 | 66.46 | 66.46 | 66.46 | 46.70 | 66.67 | 76.08 | 66.41 | 33.14 | 82.99 | 54.84 | 44.27 | 79.38 |
| CNN | 66.93 | 65.95 | 59.91 | 57.75 | 66.93 | 66.93 | 66.93 | 45.99 | 76.36 | 75.51 | 67.19 | 24.85 | 87.68 | 54.60 | 37.50 | 81.14 |
| GCAE | 65.83 | 60.95 | 60.34 | 59.20 | 65.83 | 65.83 | 65.83 | 43.72 | 60.00 | 79.14 | 62.50 | 37.28 | 81.23 | 51.45 | 45.99 | 80.17 |
| MemNet | 64.42 | 59.08 | 59.36 | 58.01 | 64.42 | 64.42 | 64.42 | 43.01 | 54.87 | 79.35 | 62.50 | 36.69 | 78.89 | 50.96 | 43.97 | 79.12 |
| RAM | 67.55 | 62.25 | 60.78 | 59.73 | 67.55 | 67.55 | 67.55 | 49.09 | 60.44 | 77.23 | 63.28 | 32.54 | 86.51 | 55.29 | 42.31 | 81.60 |
| CABASC | 70.06 | 66.14 | 63.05 | 62.94 | 70.06 | 70.06 | 70.06 | 50.98 | 69.79 | 77.63 | 60.94 | 39.64 | 88.56 | 55.52 | 50.57 | 82.74 |

## A. PRE-TRAINING FOR ASC

The research on pre-training models has become a research hotspot recently. Some existing issues need to be studied, such as pre-training on what granularity (e.g., word, subword, character), training in what structure language model (such as LSTM, Transformer [105], etc.) and how to apply

**TABLE 23.** Experimental results of Restaurants15.

| | Accuracy | Macro | | | Micro | | | Precision | | | Recall | | | F1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 | Precision | Recall | F1 | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. |
| ContextAvg | 72.31 | 65.35 | 50.18 | 49.80 | 72.31 | 72.31 | 72.31 | 74.91 | 50.00 | 71.15 | 58.67 | 2.22 | 89.65 | 65.80 | 4.26 | 79.34 |
| AEContextAvg | 73.37 | 49.87 | 50.22 | 49.17 | 73.37 | 73.37 | 73.37 | 78.54 | 0.00 | 71.06 | 59.25 | 0.00 | 91.41 | 67.55 | 0.00 | 79.96 |
| LSTM | 77.99 | 51.77 | 54.60 | 53.14 | 77.99 | 77.99 | 77.99 | 75.49 | 0.00 | 79.84 | 78.32 | 0.00 | 85.46 | 76.88 | 0.00 | 82.55 |
| GRU | 76.80 | 51.87 | 53.01 | 51.96 | 76.80 | 76.80 | 76.80 | 80.97 | 0.00 | 74.64 | 67.63 | 0.00 | 91.41 | 73.70 | 0.00 | 82.18 |
| BiGRU | 77.28 | 51.48 | 53.70 | 52.44 | 77.28 | 77.28 | 77.28 | 76.99 | 0.00 | 77.46 | 72.54 | 0.00 | 88.55 | 74.70 | 0.00 | 82.63 |
| BiLSTM | 78.34 | 52.34 | 54.36 | 53.14 | 78.34 | 78.34 | 78.34 | 79.18 | 0.00 | 77.84 | 72.54 | 0.00 | 90.53 | 75.72 | 0.00 | 83.71 |
| TD-LSTM | 77.28 | 64.53 | 57.65 | 59.04 | 77.28 | 77.28 | 77.28 | 78.06 | 37.50 | 78.04 | 71.97 | 13.33 | 87.67 | 74.89 | 19.67 | 82.57 |
| TC-LSTM | 74.44 | 62.62 | 53.41 | 54.10 | 74.44 | 74.44 | 74.44 | 76.51 | 37.50 | 73.84 | 65.90 | 6.67 | 87.67 | 70.81 | 11.32 | 80.16 |
| AT-LSTM | 80.00 | 53.32 | 55.82 | 54.48 | 80.00 | 80.00 | 80.00 | 79.88 | 0.00 | 80.08 | 78.03 | 0.00 | 89.43 | 78.95 | 0.00 | 84.50 |
| AT-GRU | 79.41 | 52.87 | 55.48 | 54.11 | 79.41 | 79.41 | 79.41 | 78.78 | 0.00 | 79.84 | 78.32 | 0.00 | 88.11 | 78.55 | 0.00 | 83.77 |
| AT-BiGRU | 77.99 | 61.04 | 54.64 | 54.30 | 77.99 | 77.99 | 77.99 | 81.88 | 25.00 | 76.24 | 70.52 | 2.22 | 91.19 | 75.78 | 4.08 | 83.05 |
| AT-BiLSTM | 79.88 | 53.11 | 55.88 | 54.45 | 79.88 | 79.88 | 79.88 | 78.41 | 0.00 | 80.93 | 79.77 | 0.00 | 87.89 | 79.08 | 0.00 | 84.27 |
| ATAE-GRU | 78.58 | 85.80 | 55.40 | 54.88 | 78.58 | 78.58 | 78.58 | 79.33 | 100.00 | 78.06 | 75.43 | 2.22 | 88.55 | 77.33 | 4.35 | 82.97 |
| ATAE-LSTM | 79.53 | 53.15 | 55.34 | 54.09 | 79.53 | 79.53 | 79.53 | 80.62 | 0.00 | 78.85 | 75.72 | 0.00 | 90.31 | 78.09 | 0.00 | 84.19 |
| ATAE-BiGRU | 78.70 | 69.08 | 56.30 | 56.29 | 78.70 | 78.70 | 78.70 | 77.46 | 50.00 | 79.80 | 77.46 | 4.44 | 87.00 | 77.46 | 8.16 | 83.25 |
| ATAE-BiLSTM | 78.34 | 52.21 | 54.59 | 53.29 | 78.34 | 78.34 | 78.34 | 78.14 | 0.00 | 78.47 | 75.43 | 0.00 | 88.33 | 76.76 | 0.00 | 83.11 |
| IAN | 79.41 | 86.18 | 56.74 | 56.82 | 79.41 | 79.41 | 79.41 | 78.82 | 100.00 | 79.72 | 77.46 | 4.44 | 88.33 | 78.13 | 8.51 | 83.80 |
| LCRS | 75.50 | 59.03 | 53.63 | 53.73 | 75.50 | 75.50 | 75.50 | 76.28 | 25.00 | 75.81 | 68.79 | 4.44 | 87.67 | 72.34 | 7.55 | 81.31 |
| CNN | 69.35 | 64.71 | 47.47 | 46.93 | 69.35 | 69.35 | 69.35 | 77.46 | 50.00 | 66.67 | 47.69 | 2.22 | 92.51 | 59.03 | 4.26 | 77.49 |
| GCAE | 76.33 | 57.61 | 53.89 | 53.32 | 76.33 | 76.33 | 76.33 | 75.07 | 20.00 | 77.76 | 73.99 | 2.22 | 85.46 | 74.53 | 4.00 | 81.43 |
| MemNet | 76.45 | 71.93 | 56.34 | 57.97 | 76.45 | 76.45 | 76.45 | 76.90 | 62.50 | 76.39 | 70.23 | 11.11 | 87.67 | 73.41 | 18.87 | 81.64 |
| RAM | 76.21 | 51.23 | 52.71 | 51.62 | 76.21 | 76.21 | 76.21 | 79.00 | 0.00 | 74.68 | 68.50 | 0.00 | 89.65 | 73.37 | 0.00 | 81.48 |
| CABASC | 76.21 | 61.73 | 56.28 | 57.30 | 76.21 | 76.21 | 76.21 | 76.47 | 31.25 | 77.47 | 71.39 | 11.11 | 86.34 | 73.84 | 16.39 | 81.67 |

**TABLE 24.** Experimental results of Restaurants16.

| | Accuracy | Macro | | | Micro | | | Precision | | | Recall | | | F1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 | Precision | Recall | F1 | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. |
| ContextAvg | 80.56 | 49.61 | 52.56 | 51.04 | 80.56 | 80.56 | 80.56 | 61.82 | 0.00 | 87.01 | 66.67 | 0.00 | 91.00 | 64.15 | 0.00 | 88.96 |
| AEContextAvg | 80.79 | 49.87 | 52.99 | 51.37 | 80.79 | 80.79 | 80.79 | 62.33 | 0.00 | 87.26 | 68.14 | 0.00 | 90.83 | 65.11 | 0.00 | 89.01 |
| LSTM | 83.12 | 76.89 | 59.24 | 58.23 | 83.12 | 83.12 | 83.12 | 64.23 | 75.00 | 91.43 | 81.86 | 6.82 | 89.03 | 71.98 | 12.50 | 90.22 |
| GRU | 83.47 | 69.29 | 60.53 | 61.34 | 83.47 | 83.47 | 83.47 | 67.81 | 50.00 | 90.07 | 77.45 | 13.64 | 90.51 | 72.31 | 21.43 | 90.29 |
| BiGRU | 83.47 | 77.36 | 59.66 | 61.39 | 83.47 | 83.47 | 83.47 | 68.18 | 75.00 | 88.91 | 73.53 | 13.64 | 91.82 | 70.75 | 23.08 | 90.34 |
| BiLSTM | 82.54 | 52.03 | 53.70 | 52.81 | 82.54 | 82.54 | 82.54 | 69.70 | 0.00 | 86.38 | 67.65 | 0.00 | 93.45 | 68.66 | 0.00 | 89.78 |
| TD-LSTM | 84.17 | 52.67 | 57.08 | 54.70 | 84.17 | 84.17 | 84.17 | 67.22 | 0.00 | 90.78 | 79.41 | 0.00 | 91.82 | 72.81 | 0.00 | 91.29 |
| TC-LSTM | 82.07 | 55.80 | 54.73 | 54.06 | 82.07 | 82.07 | 82.07 | 66.82 | 12.50 | 88.07 | 70.10 | 2.27 | 91.82 | 68.42 | 3.85 | 89.90 |
| AT-LSTM | 82.77 | 51.85 | 55.44 | 53.56 | 82.77 | 82.77 | 82.77 | 67.11 | 0.00 | 88.43 | 75.00 | 0.00 | 91.33 | 70.83 | 0.00 | 89.86 |
| AT-GRU | 83.82 | 52.68 | 56.04 | 54.30 | 83.82 | 83.82 | 83.82 | 69.06 | 0.00 | 88.99 | 75.49 | 0.00 | 92.64 | 72.13 | 0.00 | 90.78 |
| AT-BiGRU | 83.47 | 77.57 | 57.55 | 58.06 | 83.47 | 83.47 | 83.47 | 69.44 | 75.00 | 88.26 | 73.53 | 6.82 | 92.31 | 71.43 | 12.50 | 90.24 |
| AT-BiLSTM | 82.89 | 85.01 | 58.75 | 56.88 | 82.89 | 82.89 | 82.89 | 63.20 | 100.00 | 91.84 | 83.33 | 4.55 | 88.38 | 71.88 | 8.70 | 90.08 |
| ATAE-GRU | 82.31 | 51.16 | 55.11 | 53.01 | 82.31 | 82.31 | 82.31 | 64.41 | 0.00 | 89.09 | 74.51 | 0.00 | 90.83 | 69.09 | 0.00 | 89.95 |
| ATAE-LSTM | 82.19 | 51.70 | 53.10 | 52.33 | 82.19 | 82.19 | 82.19 | 69.07 | 0.00 | 86.02 | 65.69 | 0.00 | 93.62 | 67.34 | 0.00 | 89.66 |
| ATAE-BiGRU | 82.54 | 84.85 | 57.17 | 56.33 | 82.54 | 82.54 | 82.54 | 65.42 | 100.00 | 89.14 | 76.96 | 4.55 | 90.02 | 70.72 | 8.70 | 89.58 |
| ATAE-BiLSTM | 83.35 | 78.88 | 58.85 | 59.36 | 83.35 | 83.35 | 83.35 | 67.24 | 80.00 | 89.39 | 76.47 | 9.09 | 91.00 | 71.56 | 16.33 | 90.19 |
| IAN | 82.19 | 73.57 | 57.66 | 56.30 | 82.19 | 82.19 | 82.19 | 64.17 | 66.67 | 89.87 | 79.90 | 4.55 | 88.54 | 71.18 | 8.51 | 89.20 |
| LCRS | 81.61 | 68.60 | 57.16 | 59.36 | 81.61 | 81.61 | 81.61 | 70.31 | 50.00 | 85.50 | 66.18 | 13.64 | 91.65 | 68.18 | 21.43 | 88.47 |
| CNN | 81.84 | 73.19 | 55.21 | 55.14 | 81.84 | 81.84 | 81.84 | 65.44 | 66.67 | 87.48 | 69.61 | 4.55 | 91.49 | 67.46 | 8.51 | 89.44 |
| GCAE | 79.98 | 49.95 | 50.00 | 49.74 | 79.98 | 79.98 | 79.98 | 66.47 | 0.00 | 83.38 | 56.37 | 0.00 | 93.62 | 61.01 | 0.00 | 88.20 |
| MemNet | 81.26 | 67.07 | 55.25 | 57.94 | 81.26 | 81.26 | 81.26 | 70.41 | 46.15 | 84.64 | 58.33 | 13.64 | 93.78 | 63.81 | 21.05 | 88.98 |
| RAM | 83.47 | 52.61 | 55.12 | 53.83 | 83.47 | 83.47 | 83.47 | 70.00 | 0.00 | 87.83 | 72.06 | 0.00 | 93.29 | 71.01 | 0.00 | 90.48 |
| CABASC | 83.12 | 52.33 | 54.63 | 53.44 | 83.12 | 83.12 | 83.12 | 69.57 | 0.00 | 87.42 | 70.59 | 0.00 | 93.29 | 70.07 | 0.00 | 90.26 |

pre-trained models to specific tasks (e.g., ASC). Learning the word embedding via a language model used for specific tasks has been commonly used, such as Word2Vec [68] and GloVe [69]. It has almost become the standard for NLP. Moreover, to obtain the context-sensitive representation of words, some work [106]–[108] has been proposed and obtained

**TABLE 25.** Experimental results of Twitter.

| | Accuracy | Macro | | | Micro | | | Precision | | | Recall | | | F1 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Precision | Recall | F1 | Precision | Recall | F1 | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. | Neg. | Neu. | Pos. |
| ContextAvg | 68.35 | 68.69 | 64.26 | 65.82 | 68.35 | 68.35 | 68.35 | 70.15 | 67.88 | 68.03 | 54.34 | 80.64 | 57.80 | 61.24 | 73.71 | 62.50 |
| AEContextAvg | 69.94 | 69.57 | 66.57 | 67.75 | 69.94 | 69.94 | 69.94 | 67.11 | 70.66 | 70.95 | 58.96 | 80.06 | 60.69 | 62.77 | 75.07 | 65.42 |
| LSTM | 69.22 | 69.64 | 65.13 | 66.52 | 69.22 | 69.22 | 69.22 | 66.46 | 69.12 | 73.33 | 63.01 | 81.50 | 50.87 | 64.69 | 74.80 | 60.07 |
| GRU | 68.79 | 67.37 | 68.11 | 67.71 | 68.79 | 68.79 | 68.79 | 64.32 | 73.80 | 64.00 | 68.79 | 70.81 | 64.74 | 66.48 | 72.27 | 64.37 |
| BiGRU | 67.20 | 67.63 | 62.14 | 63.68 | 67.20 | 67.20 | 67.20 | 67.11 | 66.74 | 69.03 | 58.96 | 82.37 | 45.09 | 62.77 | 73.74 | 54.55 |
| BiLSTM | 68.21 | 67.75 | 64.84 | 65.98 | 68.21 | 68.21 | 68.21 | 69.18 | 69.13 | 64.94 | 58.38 | 78.32 | 57.80 | 63.32 | 73.44 | 61.16 |
| TD-LSTM | 71.82 | 72.21 | 68.11 | 68.67 | 71.82 | 71.82 | 71.82 | 65.15 | 73.59 | 77.88 | 74.57 | 82.95 | 46.82 | 69.54 | 77.99 | 58.48 |
| TC-LSTM | 72.69 | 72.76 | 69.65 | 70.90 | 72.69 | 72.69 | 72.69 | 74.00 | 72.56 | 71.71 | 64.16 | 81.79 | 63.01 | 68.73 | 76.90 | 67.08 |
| AT-LSTM | 70.95 | 69.94 | 69.17 | 69.52 | 70.95 | 70.95 | 70.95 | 69.01 | 73.54 | 67.28 | 68.21 | 76.30 | 63.01 | 68.60 | 74.89 | 65.07 |
| AT-GRU | 70.66 | 71.21 | 66.47 | 67.97 | 70.66 | 70.66 | 70.66 | 70.00 | 70.07 | 73.55 | 64.74 | 83.24 | 51.45 | 67.27 | 76.09 | 60.54 |
| AT-BiGRU | 71.97 | 75.33 | 67.73 | 69.62 | 71.97 | 71.97 | 71.97 | 89.00 | 69.93 | 67.05 | 51.45 | 84.68 | 67.05 | 65.20 | 76.60 | 67.05 |
| AT-BiLSTM | 69.80 | 68.93 | 67.73 | 68.14 | 69.80 | 69.80 | 69.80 | 70.55 | 72.65 | 63.59 | 59.54 | 76.01 | 67.63 | 64.58 | 74.29 | 65.55 |
| ATAE-GRU | 69.94 | 70.11 | 65.51 | 67.11 | 69.94 | 69.94 | 69.94 | 68.97 | 69.73 | 71.64 | 57.80 | 83.24 | 55.49 | 62.89 | 75.89 | 62.54 |
| ATAE-LSTM | 68.64 | 68.86 | 65.22 | 66.60 | 68.64 | 68.64 | 68.64 | 69.54 | 68.25 | 68.79 | 60.69 | 78.90 | 56.07 | 64.81 | 73.19 | 61.78 |
| ATAE-BiGRU | 70.23 | 71.31 | 66.28 | 68.07 | 70.23 | 70.23 | 70.23 | 72.99 | 68.60 | 72.34 | 57.80 | 82.08 | 58.96 | 64.52 | 74.74 | 64.97 |
| ATAE-BiLSTM | 70.95 | 72.77 | 66.38 | 68.38 | 70.95 | 70.95 | 70.95 | 80.34 | 69.10 | 68.87 | 54.34 | 84.68 | 60.12 | 64.83 | 76.10 | 64.20 |
| IAN | 71.82 | 73.00 | 67.15 | 69.11 | 71.82 | 71.82 | 71.82 | 76.52 | 70.21 | 72.26 | 58.38 | 85.84 | 57.23 | 66.23 | 77.24 | 63.87 |
| LCRS | 68.06 | 67.63 | 64.93 | 65.96 | 68.06 | 68.06 | 68.06 | 70.00 | 69.25 | 63.64 | 56.65 | 77.46 | 60.69 | 62.62 | 73.12 | 62.13 |
| CNN | 67.77 | 66.41 | 64.26 | 65.02 | 67.77 | 67.77 | 67.77 | 66.67 | 70.94 | 61.63 | 53.18 | 78.32 | 61.27 | 59.16 | 74.45 | 61.45 |
| GCAE | 72.11 | 72.12 | 70.04 | 70.85 | 72.11 | 72.11 | 72.11 | 75.69 | 72.65 | 68.00 | 63.01 | 78.32 | 68.79 | 68.77 | 75.38 | 68.39 |
| MemNet | 69.65 | 69.09 | 66.76 | 67.68 | 69.65 | 69.65 | 69.65 | 71.17 | 70.57 | 65.52 | 67.05 | 78.32 | 54.91 | 69.05 | 74.25 | 59.75 |
| RAM | 70.09 | 71.32 | 64.93 | 66.48 | 70.09 | 70.09 | 70.09 | 70.62 | 68.84 | 74.51 | 65.32 | 85.55 | 43.93 | 67.87 | 76.29 | 55.27 |
| CABASC | 68.64 | 69.74 | 64.64 | 66.44 | 68.64 | 68.64 | 68.64 | 75.00 | 67.07 | 67.14 | 58.96 | 80.64 | 54.34 | 66.02 | 73.23 | 60.06 |

great success. Peters *et al.* [109] trained an LSTM-based language model on a large number of texts. Recently, Delvin *et al.* [110] developed a BERT model, which was based on the multi-layered transformer mechanism. It predicted the loss function of the masked words in the sentence and the next sentence in the pre-trained model. These models first obtained a context-sensitive representation of the input text, and then applied this representation to specific tasks. The results showed that this method has been significantly improved in grammatical analysis, reading comprehension, text classification, and other tasks [110]. Existing work [85], [86], [94] integrated BERT into ASC and obtained significant improvements, which showed the effectiveness of pre-training. Thus, how to quickly find a suitable pre-training model and automatically select the best application method for ASC is an interesting and promising research topic.

### B. DEEP MULTI-TASK LEARNING FOR ASC

Deep multi-task learning plays a significant role in NLP tasks that lack sufficient training data. Deep multi-task learning builds sharing networks and specific network structures at the output layer for different tasks. It enhances the model to learn the knowledge and information shared between various tasks. Among the reviewed studies, several studies [36], [111] applied multi-task learning to ASC in a deep neural framework and achieved some improvements over single task learning. The advantages of adopting deep neural network based multi-task learning can be summarized as follows: 1) learning multiple tasks can avoid overfitting by generating the shared hidden representations; 2) auxiliary task provides interpretable output for explaining the classification; 3) multi-task can alleviate the sparsity problem for an implicit data augmentation provided. It is interesting to extract the aspect and predict the sentiment jointly. Except for applying auxiliary tasks, the deep multi-task learning for cross-domain ASC where each specific task aims at predicting classification for each domain also can be introduced.

### C. EXPLAINABLE ASC WITH DEEP LEARNING

A common disadvantage of deep learning is that it is highly non-interpretable. As such, making explainable ASC with deep learning seems to be an important task. It is natural to assume that big and complex neural network models are just fitting the data with any true understanding. This is precisely why this direction is both exciting and also crucial. The advantages of applying explainable deep learning for ASC are two-fold. First, explainable predictions allow the user to understand the reasons behind the classifications of the network (i.e. why is the sentiment polarity of the review/comment positive or negative?). Second, the practitioner can understand the model more via its explain-ability. It is worth of extracting the opinion words in the sentence with respect to the given aspect to provide an explanation for the prediction of deep learning models.

In addition, attention-based neural models play an important role in interpretability for ASC since the attentive weights provide insights into the model and give explainable results to the practitioners and users. Given that models are already able to highlight what contributes to the decision, we believe that designing better attentional mechanisms is a promising direction.

### D. COMMONSENSE KNOWLEDGE FOR ASC

How to integrate commonsense knowledge into deep learning models has become a significant research topic in the field of NLP, such as question answer [112], [113], machine reading comprehension [114], [115]. Common sense is the objective facts that the majority of human understand and accept, such as ''the Earth is round'', ''water is liquid '', ''the sun rises from the east'' and so on. Common sense plays an important role in machines to make a deeper understanding of natural language. However, obtaining common sense is a huge challenge, and it will affect the process of artificial intelligence once there is a breakthrough. Ma *et al.* [27] incorporated commonsense knowledge of sentiment-related concepts into standard LSTM model for ASC. However, there is no in-depth study on how to apply commonsense knowledge in the ASC, there is some work to be paid attention to. For example, graph neural network (GNN) [116], [117] has obtained great success in graph embedding recently. Thus, it would be worth to model the common sense and knowledge through GNN to take the relationships between the entities and relations into account.

### E. LOW-RESOURCE METHODS FOR ASC

The problem of poor labeled data resources (e.g., ASC datasets) in NLP is referred to as the low-resource NLP problem. Apart from enhancing data ability by integrating domain knowledge (such as dictionaries and rules), the following strategies are also useful: 1) adding more manual annotation data via active learning methods, unsupervised and semi-supervised methods to utilize unlabeled data; 2) adopting multi-task learning methods for learning information from other tasks, other domains and other languages; 3) introducing transfer learning approaches to take advantage of other models. Deep transfer learning obtains the high-level abstract representation that disentangles the difference between different domains. Existing work [9], [36] showed the effectiveness of deep learning in capturing the similarities and differences across different domains and generating better classification on cross-domain platforms. Therefore, the low-resource approach for ASC is a significant area to be explored. For instance, it would be interesting to find out what each layer of neural model learned from the different domains and which layers to transfer. In addition, for different transfer tasks, how to determine which tasks and the order of tasks to transfer is a promising direction.

### F. DEEPER NEURAL NETWORKS FOR ASC

From previous studies [47], [55], we found that most existing deep learning models for ASC consisted of three to four layers. Going deeper has shown to outperform shallow neural network models in many tasks [118], [119]. However, deeper neural networks for ASC is largely unclear. If going deeper provides good performance, how to train the deep architecture? If not, what is the reason behind it? Thus, deeper neural network for ASC is an under-explored area where more work is expected.

## VIII. SUMMARY

Both deep learning and ASC are ongoing hot research topics in the past decade. A large number of new techniques and emerging models are proposed for deep learning-based ASC each year. In this article, we provide an extensive review of the most notable work up to date on deep learning-based ASC. In particular, we propose a classification scheme for organizing and clustering existing publications and highlight a bunch of influential research prototypes. Then, we discuss and provide an in-depth analysis about the advantages and disadvantages of applying deep learning techniques for ASC tasks. We also collect almost all the benchmark datasets for researchers to study and implement several classical state-of-the-art methods for ASC. In addition, we evaluate the effectiveness of these methods on five public standard datasets with widely used evaluation measures. Finally, we detail some of the most challenging open problems and promising future research directions.

Deep learning has achieved good success in the field of ASC, which will enable multiple application domains (e.g., products, economics, biomedicine, healthcare and policies [6], [10], [120]–[124]) to benefit from the knowledge learned from ASC. Deep learning will be key for stepping forward in the development of ASC. However, most of neural models still learn explicit emotions (e.g., opinion words) in sentences. For the implicit emotional expressions such as irony, deep reasoning, common sense, etc., the recent neural networks cannot learn them well, which are not difficult for humans. Furthermore, these models are heavily dependent on the size of data. Thus, there is still a long way to go in this field with deep learning. We hope this survey can provide readers with a comprehensive understanding of the key aspects of deep learning-based ASC to clarify the most notable advancements and shed some light on future studies.

## ACKNOWLEDGMENT

The authors greatly appreciate anonymous reviewers and the associate editor for their valuable and high quality comments that greatly helped to improve the quality of this article. The computation is performed in ECNU Public Platform for Innovation (001). All the work is done when the first author is a visiting Ph.D. student at the Information Retrieval and Knowledge Management Research Lab, York University, Canada. His visit is supported by an NSERC discovery grant.

## REFERENCES

[1] B. Liu, ''Sentiment analysis and opinion mining,'' *Synthesis Lectures Hum. Lang. Technol.*, vol. 5, no. 1, pp. 1–167, 2012.

[2] B. Pang, L. Lee, and S. Vaithyanathan, ''Thumbs up?: Sentiment classification using machine learning techniques,'' in *Proc. Conf. Empirical Methods Natural Lang. (EMNLP)*, Philadelphia, PA, USA, Jul. 2002, pp. 79–86.

[3] 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 Eval. (SemEval)*, 2014, pp. 27–35.

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

[5] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit, "Semeval-2016 task 5: Aspect based sentiment analysis," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 19–30.

[6] B. Pang and L. Lee, "Opinion mining and sentiment analysis," *Found. Trends Inf. Retr.*, vol. 2, nos. 1–2, pp. 1–135, 2008.

[7] M. Tsytsarau and T. Palpanas, "Survey on mining subjective data on the web," *Data Mining Knowl. Discovery*, vol. 24, no. 3, pp. 478–514, 2012.

[8] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10760–10773, 2009.

[9] Z. Li, Y. Wei, Y. Zhang, X. Zhang, X. Li, and Q. Yang, "Exploiting coarse-to-fine task transfer for aspect-level sentiment classification," in *Proc. AAAI*, 2019, pp. 1–8.

[10] K. Schouten and F. Frasincar, "Survey on aspect-level sentiment analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 3, pp. 813–830, Mar. 2016.

[11] S. Wang, Z. Chen, and B. Liu, "Mining aspect-specific opinion using a holistic lifelong topic model," in *Proc. 25th Int. Conf. World Wide Web, Int. World Wide Web Conf. Steering Committee*, 2016, pp. 167–176.

[12] Z. Chen, A. Mukherjee, and B. Liu, "Aspect extraction with automated prior knowledge learning," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2014, pp. 347–358.

[13] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai, "Topic sentiment mixture: Modeling facets and opinions in weblogs," in *Proc. 16th Int. Conf. World Wide Web*, 2007, pp. 171–180.

[14] S. Poria, E. Cambria, and A. Gelbukh, "Aspect extraction for opinion mining with a deep convolutional neural network," *Knowl.-Based Syst.*, vol. 108, pp. 42–49, Sep. 2016.

[15] O. Irsoy and C. Cardie, "Opinion mining with deep recurrent neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 720–728.

[16] Y. Lu and C. Zhai, "Opinion integration through semi-supervised topic modeling," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 121–130.

[17] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, "Target-dependent Twitter sentiment Classification," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Portland, OR, USA, Jun. 2011, pp. 151–160.

[18] 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 Eval. (SemEval)*, 2014, pp. 437–442.

[19] 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.

[20] S. Moghaddam and M. Ester, "The FLDA model for aspect-based opinion mining: Addressing the cold start problem," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 909–918.

[21] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Comput. Linguistics*, vol. 37, no. 1, pp. 9–27, 2011.

[22] D.-T. Vo and Y. Zhang, "Target-dependent twitter sentiment classification with rich automatic features," in *Proc. IJCAI*, 2015, pp. 1347–1353.

[23] 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 Annu. Meeting Assoc. Comput. Linguistics*, vol. 2. Jun. 2014, pp. 49–54.

[24] 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.

[25] M. Zhang, Y. Zhang, and D.-T. Vo, "Gated neural networks for targeted sentiment analysis," in *Proc. AAAI*, 2016, pp. 3087–3093.

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

[27] Y. Ma, H. Peng, and E. Cambria, "Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM," in *Proc. AAAI*, 2018, pp. 5876–5883.

[28] S. Ruder, P. Ghaffari, and J. G. Breslin, "A hierarchical model of reviews for aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 999–1005.

[29] Y. Wang, M. Huang, and L. Zhao, "Attention-based lstm for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 606–615.

[30] M. Yang, W. Tu, J. Wang, F. Xu, and X. Chen, "Attention based LSTM for target dependent sentiment classification," in *Proc. AAAI*, 2017, pp. 5013–5014.

[31] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "Effective attention modeling for aspect-level sentiment classification," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1121–1131.

[32] J. Wang, J. Li, S. Li, Y. Kang, M. Zhang, L. Si, and G. Zhou, "Aspect sentiment classification with both word-level and clause-level attention networks," in *Proc. IJCAI*, 2018, pp. 4439–4445.

[33] Y. Tay, A. T. Luu, and S. C. Hui, "Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis," in *Proc. AAAI*, 2018, pp. 5956–5963.

[34] J. Cheng, S. Zhao, J. Zhang, I. King, X. Zhang, and H. Wang, "Aspect-level sentiment classification with heat (hierarchical attention) network," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 97–106.

[35] J. Zeng, X. Ma, and K. Zhou, "Enhancing attention-based LSTM with position context for aspect-level sentiment classification," *IEEE Access*, vol. 7, pp. 20462–20471, 2019.

[36] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "Exploiting document knowledge for aspect-level sentiment classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Melbourne, VIC, Australia, vol. 2, Jul. 2018, pp. 579–585.

[37] D. Hazarika, S. Poria, P. Vij, G. Krishnamurthy, E. Cambria, and R. Zimmermann, "Modeling inter-aspect dependencies for aspect-based sentiment analysis," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 2, 2018, pp. 266–270.

[38] D. Ma, S. Li, X. Zhang, and H. Wang, "Interactive attention networks for aspect-level sentiment classification," in *Proc. IJCAI*, 2017, pp. 4068–4074.

[39] J. Liu and Y. Zhang, "Attention modeling for targeted sentiment," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 2, 2017, pp. 572–577.

[40] F. Fan, Y. Feng, and D. Zhao, "Multi-grained attention network for aspect-level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3433–3442.

[41] S. Gu, L. Zhang, Y. Hou, and Y. Song, "A position-aware bidirectional attention network for aspect-level sentiment analysis," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 774–784.

[42] B. Huang, Y. Ou, and K. M. Carley, "Aspect level sentiment classification with attention-over-attention neural networks," in *Proc. SBP-BRiMS*, 2018, pp. 197–206.

[43] S. Zheng and R. Xia, "Left-center-right separated neural network for aspect-based sentiment analysis with rotatory attention," Feb. 2018, *arXiv:1802.00892*. [Online]. Available: https://arxiv.org/abs/1802.00892

[44] W. Xue and T. Li, "Aspect based sentiment analysis with gated convolutional networks," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL)*. Melbourne, VIC, Australia, vol. 1, Jul. 2018, pp. 2514–2523.

[45] B. Huang and K. Carley, "Parameterized convolutional neural networks for aspect level sentiment classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1091–1096.

[46] C. Fan, Q. Gao, J. Du, L. Gui, R. Xu, and K.-F. Wong, "Convolution-based memory network for aspect-based sentiment analysis," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2018, pp. 1161–1164.

[47] X. Li, L. Bing, W. Lam, and B. Shi, "Transformation networks for target-oriented sentiment classification," in *Proc. ACL*, 2018, pp. 946–956.

[48] Y. Tay, L. A. Tuan, and S. C. Hui, "Dyadic memory networks for aspect-based sentiment analysis," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 107–116.

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

[50] J. Yang, R. Yang, C. Wang, and J. Xie, "Multi-entity aspect-based sentiment analysis with context, entity and aspect memory," in *Proc. 22nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[51] P. Zhu and T. Qian, "Enhanced aspect level sentiment classification with auxiliary memory," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 1077–1087.

[52] N. Majumder, S. Poria, A. Gelbukh, M. S. Akhtar, E. Cambria, and A. Ekbal, "IARM: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 3402–3411.

[53] S. Wang, S. Mazumder, B. Liu, M. Zhou, and Y. Chang, "Target-sensitive memory networks for aspect sentiment classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 957–967.

[54] S. Wang, G. Lv, S. Mazumder, G. Fei, and B. Liu, "Lifelong learning memory networks for aspect sentiment classification," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 861–870.

[55] Q. Liu, H. Zhang, Y. Zeng, Z. Huang, and Z. Wu, "Content attention model for aspect based sentiment analysis," in *Proc. World Wide Web Conf. World Wide Web, Int. World Wide Web Conf. Steering Committee*, 2018, pp. 1023–1032.

[56] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 129–136.

[57] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Lisbon, Portugal, Sep. 2015, pp. 1412–1421.

[58] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," Feb. 2017, *arXiv:1702.00887*. [Online]. Available: https://arxiv.org/abs/1702.00887

[59] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," Sep. 2014, *arXiv:1409.0473*. [Online]. Available: https://arxiv.org/abs/1409.0473

[60] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1378–1387.

[61] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2440–2448.

[62] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1693–1701.

[63] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," Nov. 2016, *arXiv:1611.01603*. [Online]. Available: https://arxiv.org/abs/1611.01603

[64] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[65] Y. Kim, "Convolutional neural networks for sentence classification," Aug. 2014, *arXiv:1408.5882*. [Online]. Available: https://arxiv.org/abs/1408.5882

[66] X. Li and W. Lam, "Deep multi-task learning for aspect term extraction with memory interaction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 2886–2892.

[67] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. ICLR*, 2013, pp. 1–12.

[68] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[69] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. 2014 Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[70] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[71] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Assoc. Comput. Linguistics, 2011, pp. 151–161.

[72] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1631–1642.

[73] R. Socher, E. H. Huang, J. Pennin, C. D. Manning, and A. Y. Ng, "Dynamic pooling and unfolding recursive autoencoders for paraphrase detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 801–809.

[74] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[75] A. Graves, "Generating sequences with recurrent neural networks," Aug. 2013, *arXiv:1308.0850*. [Online]. Available: https://arxiv.org/abs/1308.0850

[76] K. Greff, R. K. Srivastava, and J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[77] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," Dec. 2014, *arXiv:1412.3555*. [Online]. Available: https://arxiv.org/abs/1412.3555

[78] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2016, pp. 1480–1489.

[79] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12 pp. 2493–2537, Aug. 2011.

[80] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. COLING*, 2014, pp. 2335–2344.

[81] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *Handbook Brain Theory Neural Netw.*, vol. 3361, no. 10, p. 1995, 1995.

[82] M. S. Akhtar, P. Sawant, S. Sen, A. Ekbal, and P. Bhattacharyya, "Solving data sparsity for aspect based sentiment analysis using cross-linguality and multi-linguality," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol. (NAACL-HLT)*. New Orleans, LA, USA: Association for Computational Linguistics, vol. 1, Jun. 2018, pp. 572–582.

[83] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Recursive neural conditional random fields for aspect-based sentiment analysis," *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Austin, TX, USA, Nov. 2016, pp. 616–626.

[84] J. Wagner, P. Arora, S. Cortes, U. Barman, D. Bogdanova, J. Foster, and L. Tounsi, "DCU: Aspect-based polarity classification for SemEval task 4," in *Proc. 8th Int. Workshop Semantic Eval. (SemEval)*, 2014, pp. 223–229.

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

[86] 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. Linguistics*, 2019, pp. 1–12.

[87] J. Saias, "Sentiue: Target and aspect based sentiment analysis in SemEval-2015 Task 12," in *Proc. 9th Int. Workshop Semantic Eval. (SemEval@NAACL-HLT)*. Denver, CO, USA: The Association for Computer Linguistics, Jun. 2015, pp. 767–771.

[88] Z. Toh and J. Su, "Nlangp at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features," in *Proc. 10th Int. Workshop Semantic Eval. (SemEval)*, 2016, pp. 282–288.

[89] S. Ruder, P. Ghaffari, and J. G. Breslin, "Insight-1 at semeval-2016 task 5: Deep learning for multilingual aspect-based sentiment analysis," Sep. 2016, [Online]. Available: https://arxiv.org/abs/1609.02748

[90] S. Pateria, "Aspect based sentiment analysis using sentiment flow with local and non-local neighbor information," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 2635–2646.

[91] B. Wang, M. Liakata, A. Zubiaga, and R. Procter, "Tdparse: Multi-target-specific sentiment recognition on twitter," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 483–493.

[92] L. Deng and J. Wiebe, "Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 179–189.

[93] M. Saeidi, G. Bouchard, M. Liakata, and S. Riedel, "Sentihood: Targeted aspect based sentiment analysis dataset for urban neighbourhoods," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics, Tech. Papers*, 2016, pp. 1546–1556.

[94] C. Sun, L. Huang, and X. Qiu, "Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2019, pp. 1–6.

[95] M. Zhang, Y. Zhang, and D. T. Vo, "Neural networks for open domain targeted sentiment," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 612–621.

[96] D. Marcheggiani, O. Täckström, A. Esuli, and F. Sebastiani, "Hierarchical multi-label conditional random fields for aspect-oriented opinion mining," in *Proc. 36th Eur. Conf. IR Res. (ECIR)*, Amsterdam, The Netherlands, Apr. 2014, pp. 273–285.

[97] L. Fang, M. Huang, and X. Zhu, "Exploring weakly supervised latent sentiment explanations for aspect-level review analysis," in *Proc. 22nd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2013, pp. 1057–1066.

[98] I. Titov and R. McDonald, "A joint model of text and aspect ratings for sentiment summarization," *Proc. ACL, HLT*, 2008, pp. 308–316.

[99] B. Lu, M. Ott, C. Cardie, and B. K. Tsou, "Multi-aspect sentiment analysis with topic models," in *Proc. 11th IEEE Int. Conf. Data Mining Workshops*, Dec. 2011, pp. 81–88.

[100] M. Mitchell, J. Aguilar, T. Wilson, and B. Van Durme, "Open domain targeted sentiment," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1643–1654.

[101] L. Deng and J. Wiebe, "MPQA 3.0: An entity/event-level sentiment corpus," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 1323–1328.

[102] G. Ganu, N. Elhadad, and A. Marian, "Beyond the stars: Improving rating predictions using review text content," in *Proc. WebDB*, vol. 9, 2009, pp. 1–6.

[103] W. X. Zhao, J. Jiang, H. Yan, and X. Li, "Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Assoc. Comput. Linguistics, 2010, pp. 56–65.

[104] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, vol. 5, 2015, pp. 1–15.

[105] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhi, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[106] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. (2018). *Improving Language Understanding by Generative Pre-Training*. [Online]. Available: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[107] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 328–339.

[108] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, San Francisco, CA, USA, Tech. Rep. 1.8, 2019.

[109] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1, 2018, pp. 2227–2237.

[110] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Oct. 2018, *arXiv:1810.04805*. [Online]. Available: https://arxiv.org/abs/1810.04805

[111] M. Schmitt, S. Steinheber, K. Schreiber, and B. Roth, "Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1109–1114.

[112] R. West, E. Gabrilovich, K. Murphy, S. Sun, R. Gupta, and D. Lin, "Knowledge base completion via search-based question answering," in *Proc. 23rd Int. Conf. World Wide Web*, 2014, pp. 515–526.

[113] L. Dong, F. Wei, M. Zhou, and K. Xu, "Question answering over freebase with multi-column convolutional neural networks," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process.*, vol. 1, 2015, pp. 260–269.

[114] B. Yang and T. Mitchell, "Leveraging knowledge bases in LSTMs for improving machine reading," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics (ACL)* 2017, Vancouver, BC, Canada, vol. 1, 2017, pp. 1436–1446.

[115] T. Mihaylov and A. Frank, "Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, Melbourne, Victoria, Australia, vol. 1, 2018, pp. 821–832.

[116] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[117] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–14.

[118] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

[119] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[120] I. J. Arnold and E. B. Vrugt, "Fundamental uncertainty and stock market volatility," *Appl. Financial Econ.*, vol. 18, no. 17, pp. 1425–1440, 2008.

[121] R. E. Goldsmith and D. Horowitz, "Measuring motivations for online opinion seeking," *J. Interact. Advertising*, vol. 6, no. 2, pp. 2–14, 2006.

[122] Y. Chen and J. Xie, "Online consumer review: Word-of-mouth as a new element of marketing communication mix," *Manage. Sci.*, vol. 54, no. 3, pp. 477–491, 2008.

[123] Y. Liu, X. Huang, A. An, and X. Yu, "ARSA: A sentiment-aware model for predicting sales performance using blogs," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Amsterdam, The Netherlands, Jul. 2007, pp. 607–614.

[124] Y. Liu, X. Huang, A. An, and X. Yu, "Modeling and predicting the helpfulness of online reviews," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 443–452.

**JIE ZHOU** is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, East China Normal University, China. Since 2016, he has published several referred papers in international conferences, such as AAAI, ICTAI, and IJCNN. His research interests include sentiment analysis, aspect-level sentiment classification, retrieval model, and neural networks. He received a scholarship from the China Scholarship Council and received Top-3 in the KDD CUP Competition several times.

**JIMMY XIANGJI HUANG** received the Ph.D. degree in information science from the City, University of London. He was a Postdoctoral Fellow with the School of Computer Science, University of Waterloo, Canada. He joined York University as an Assistant Professor, in 2003, where he is currently a York Research Chair Professor and the Director of the Information Retrieval & Knowledge Management Research Laboratory (IRLab). He was awarded tenure and promoted to Full Professor, in 2006 and 2011, respectively. Since 2003, he has published more than 230 refereed papers in journals, such as the *ACM Transactions on Information Systems*, the *Journal of American Society for Information Science and Technology*, *Information Processing & Management*, the IEEE Transactions on Knowledge and Data Engineering, *Information Sciences*, *Information Retrieval*, *BMC Bioinformatics*, *BMC Genomics*, and *BMC Medical Genomics*, lead-edited six books and multiple book chapters, and the international conference proceedings, such as ACM SIGIR, ACM CIKM, KDD, ACL, COLING, IEEE ICDM, IJCAI, and AAAI. He is a Senior Member of the ACM Distinguished Scientist. He received the Deans Award for Outstanding Research, in 2006, the Early Researcher Award, formerly the Premiers Research Excellence Award, in 2007, the Petro Canada Young Innovators Award, in 2008, the SHARCNET Research Fellowship Award, in 2009, the Best Paper Award at the 32nd European Conference on Information Retrieval (ECIR 2010), U.K., and the LA&PS Award for Distinction in Research, Creativity, and Scholarship (established researcher), in 2015. He was the General Conference Chair of the 19th International ACM CIKM Conference, in 2010, and will be the General Conference Chair of the 43rd ACM SIGIR Conference, in 2020.

**QIN CHEN** received the Ph.D. degree from the Department of Computer Science and Technology, East China Normal University, China. She is currently a Postdoctoral Fellow with the School of Data Science, Fudan University, China. Since 2013, she has published many refereed papers in top-tier conferences and journals, such as AAAI, SIGIR, TKDE, and TIST. Her research interests include information retrieval, natural language processing, and deep learning applied to text data.

**QINMIN VIVIAN HU** received the Ph.D. degree in computer science from York University, Toronto, Canada. She was an Associate Professor with East China Normal University, Shanghai, China, and a Postdoctoral Fellow with the MRI Research Facility, Wayne State University, USA. She is currently an Assistant Professor with the Department of Computer Science, Ryerson University, Toronto. She has published more than 30 refereed papers in top-tier journals, such as the IEEE Transactions on Knowledge and Data Engineering and the *ACM Transactions on Intelligent Systems and Technology* and conferences, such as AAAI and ACM SIGIR. She received the National NSERC Postdoctoral Fellowship as one of the best Ph.D. fellows in Canada, in 2013.

**TINGTING WANG** is currently pursuing the master's degree with the Department of Computer Science and Technology, East China Normal University, China. Her research interests include sentiment analysis, aspect-level sentiment classification, retrieval model, and neural networks.

**LIANG HE** received the bachelor's and Ph.D. degrees from the Department of Computer Science and Technology, East China Normal University, China, where he is currently a Professor and the Director of the Department of Computer Science and Technology. He holds more than ten patents and has published two monographs and more than 70 refereed papers in national and international journals and conference proceedings. His current research interests include knowledge processing, user behavior analysis, and context-aware computing. He has been awarded the Star of the Talent in Shanghai. He is also a Council Member of the Shanghai Computer Society, a member of the Academic Committee, the Director of the Technical Committee of the Shanghai Engineering Research Center of Intelligent Service Robot, and a Technology Foresight Expert of the Shanghai Science and Technology in focused areas. He has been hosted a number of National Science and Technology Support and participated in the National 13th Five-Year Technology Support Programs, the Shanghai Science and Technology Long-Term Development Plan, and the Shanghai 13th Five-Year Science and Technology Plan. He received the Shanghai Science and Technology Progress Award for five times and received the First Prize, in 2013, and the Second Prize, in 2015.