Deep2s: Improving Aspect Extraction in Opinion Mining With Deep Semantic Representation

XUELIAN LI1,2, BI WANG1,2, LIXIN LI1,2, ZHIQIANG GAO1,2, QIAN LIU3, HANCHENG XU4, AND LANTING FANG5,6

1Key Laboratory of Computer Network and Information Integration, Ministry of Education, Southeast University, Nanjing 210096, China
2School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
3School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing 210000, China
4International Studies College, National University of Defense Technology, Nanjing 210000, China
5School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China
6Purple Mountain Laboratories, Nanjing 210000, China

Corresponding author: Zhiqiang Gao (zqgao@seu.edu.cn)

This work was supported by the National Science Foundation of China under Grant 61702279, Grant 61602260, and Grant 61502095.

ABSTRACT Syntactical rule based approaches for aspect extraction, which are free from expensive manual annotation, are promising in practice. These approaches extract aspects mainly through the dependency relations in the surface sentence structures. However, deep and rich semantic information hidden in sentences which can help improve aspect extraction, is difficult for them to capture. In order to address the problem, this paper first proposes to employ Logic Programming to explore the feasibility of deep semantic representation, then proposes Deep2S, a hybrid rule-based method to improve the performance of aspect extraction. Deep2S integrates deep semantic representation such as Abstract Meaning Representation (AMR) with syntactic structure. It can take advantage of the syntactical rules to obtain dependency relations in the surface structure as well as the semantic rules to capture deep semantic information. Our experiments are conducted on eight popular review datasets using two evaluation metrics. Experimental results demonstrate the usefulness of deep semantic representation and the ability of Deep2S to improve the performance of aspect extraction in opinion mining.

INDEX TERMS Opinion mining, aspect extraction, abstract meaning representation, deep semantic representation, logic programming.

I. INTRODUCTION Aspect extraction, also named opinion target extraction, is one of the most important tasks of opinion mining or sentiment analysis [1]–[3]. It focuses on the identification of product aspects or attributes from online reviews. For example, we can extract “battery” as aspect terms from the user’s reviews in Example 1 and Example 2.

Example 1: “The camera has a good battery.”
Example 2: “The battery of the camera is good.”

In the task of aspect extraction, syntactical rule based approaches such as double propagation (DP) [4] have been proven to be promising in practice. This is because they are usually domain independent and free from expensive manual annotation. Syntactical based approaches mainly utilize the surface syntactical information of dependency trees to extract aspects. For example, from Figure 1 (b) and (c), we can obtain the dependency relations (i.e., “nsubj-dobj”) and “nmod”) between “battery” and “camera” in Example 1 and 2. But these approaches cannot capture deep and rich semantic knowledge hidden in user’s review sentences, such as semantic relation “domain” which means “battery” is an aspect or attribute of “camera”, as shown in Figure 1 (a). If we know “camera” is a product, we can extract “battery” as its aspect. It is intuitive that semantic information can help improve the performance of aspect extraction.

To explore the usefulness of semantic knowledge in aspect extraction, deep semantic representation (e.g., Abstract Meaning Representation, or AMR [5]) which abstracts away from syntactic idiosyncrasies, is adopted to capture deep and rich semantic information from user’s reviews. Due to the diversity and flexibility of natural language, the same or similar reviews are often described in different sentence structures
or expressed in casual manner [6]. It is obvious that two different sentences with the same meaning often have the unique AMR graph or the same AMR subgraph. For example, Figure 1 (a) shows the common semantic structure of AMR graph of Example 1 and 2. In AMR graph, some words which do not contribute to the meaning of a sentence, such as function words ("to", "the"), are omitted from the sentence. We can see that the depth between "battery" and "camera" in AMR graph of Figure 1 (a) is one while in dependency tree of Figure 1 (b) is two. This means semantic representation has deep semantics. Furthermore, the semantic relation "domain" between "camera" and "battery" can be captured by AMR subgraph, which means semantic representation has rich semantics.

To improve the performance of aspect extraction, a hybrid rule-based approach with the capacity to capture surface structure and deep semantics called Deep2S is proposed. Deep2S is based on Answer Set Programming (ASP) framework – a variant of Logic Programming. Since the non-monotonicity and scalability of ASP, this paper uses ASP rules to represent syntactical rules and semantic rules. Specifically, DP and rule selection-DP (RSDP) [8] are employed for syntactic rule pattern to acquire surface structure information. And AMR is used as semantic rule pattern to capture deep and rich semantic information.

The major contributions of this paper could be summarized as follows:

1) We first propose to employ ASP to implement semantic structure based aspect extraction by adopting AMR graph to represent the semantic relations between aspects and opinion words. Then, we propose Deep2S, a hybrid rule-based approach to improve the performance of aspect extraction by combining syntactic and semantic rules.

2) Deep semantic representation such as Abstract Meaning Representation, is explored to capture deep and rich semantic information in customer reviews. This paper hypothesizes that the semantic relations between aspects and opinion words or other entities can be captured by AMR subgraphs.

3) In our experiments, we evaluate our proposed method on eight popular aspect extraction evaluation datasets. The experimental results have shown that deep semantic representation is useful, although its performance is not as good as syntactic structure of dependency grammar. Moreover, our method Deep2S which integrates shallow structure and deep semantic information, can improve aspect extraction. Our code is publicly available at https://github.com/njirene/Deep2S.

In the remainder of this paper, section 2 mentions related work. Section 3 introduces Deep2S model. Subsequently, section 4 presents experiments. Finally we conclude with a summary and some future research directions in section 5.

II. RELATED WORK
In this section, we first introduce applications of syntactic structures, and then semantic structures in NLP tasks.

A. SYNTACTIC STRUCTURE BASED APPROACHES FOR NLP TASKS
Syntactic structure has been found to be useful in NLP tasks such as sentence classification [9], named entity recognition [10], [11], relation extraction [12], [13], machine translation [14] and aspect extraction [4], [15]–[17].

With respect to aspect extraction in opinion mining, there are two major approaches, one is rule-based approach and the other is deep learning approach. The former is mainly based on syntactic or dependency structure of each sentence to design some rules for extracting aspects [4], [8], [16], [18]–[22], while the latter is mainly based on syntactic or dependency structure of each sentence to yield embeddings or learned features [17], [23]–[28].

1) RULE-BASED APPROACH
Rule-based approaches which are usually unsupervised and free from expensive manual annotation, are widely adopted in aspect extraction.

Qiu et al. [4] proposed a double propagation approach to perform opinion lexicon expansion and target extraction tasks given a set of seed opinion words. This approach employed
a dependency grammar to describe the relations of opinion targets and opinion words.

Liu et al. [8] proposed an automated rule set selection approach (RSDP) to automatically select a subset of high quality rules from a set of syntactical extraction rules based on dependency relations.

Liu et al. [16] utilized DP approach as baseline and improved the syntactic rule-based method by similarity-based and association-based recommendations.

Wu et al. [22] proposed a hybrid unsupervised method to perform aspect term and opinion target extraction by combining chunk-level rules based on dependency relations with machine learning methods.

As we all know, there are a lot of works focusing on syntactic rules which are proven to be helpful to improve the performance of natural language processing applications. However, little research has been done on whether semantic rules can contribute to applications such as aspect extraction in opinion mining. In this paper, we will validate whether semantic rules are useful for aspect extraction.

2) DEEP LEARNING APPROACH

Some studies have shown that deep learning approaches can achieve better performance.

Yin et al. [23] utilized the embeddings of words and dependency paths as features in CRF to extract aspects.

Wang et al. [17] encoded high-level features such as the information of dependency paths into RNN to extract aspect terms and the opinions.

Ye et al. [24] proposed dependency-tree based convolutional stacked neural networks to capture the syntactic features of sentences for aspect extraction.

Li et al. [11] exploited multi-order dependency-based word embeddings which indicated that the dependency-based context can capture most of the stable and valuable syntactic features and partial semantic information.

Luo et al. [29] proposed a bidirectional dependency tree network to extract dependency structure features from the given sentences. This method constructed a bidirectional propagation mechanism on the dependency tree to enhance the tree-structured representation.

However, deep learning methods require a great number of data to train models, and are usually uninterpretable. In this paper, we focus on the rule-based approach. Our work is most related to the DP [4] and RSDP [8], both of which are based on syntactical rules. There are two differences between our work and the two works. The first one is that their work only extracts aspects modified by opinion adjectives, while our work can also extract aspects modified by opinion verbs, adverbs, as well as aspects without opinion words. The second difference is that we consider them as our bases and propose to incorporate semantic rules with syntactic rules to improve the performance of aspect extraction.

B. SEMANTIC STRUCTURE BASED APPROACHES FOR NLP TASKS

Semantic structure based dependency parsing aims to produce graph-structured semantic dependency representations of sentences instead of tree structured syntactic dependency representations.

Abstract Meaning Representation as a form of deep semantic representation, is a semantic representation where the meaning of a sentence is encoded as a rooted, directed, acyclic graph. AMR mainly focuses on content words, some surface words such as function words (“to”, “the”) which do not contribute to the meaning of a sentence, will be omitted from a sentence. Recently, AMR has been utilized as some NLP tasks, such as entity linking [30], event extraction [31], [32], text summarization [33]–[36], machine translation [37].

Pan et al. [30] adopted AMR to represent semantic information about entity mentions for entity linking.

Li et al. [31] improved event detection by adding AMR features on the basis of the commonly used features (e.g., lexical, syntactic, and entity information). In this task, AMR was adopted to capture the deeper semantic contexts of the trigger words and represent the relations between the event trigger word and entities.

Rao et al. [32] used a deep semantic representation based on AMR to extract events in biomedical text and hypothesized that an event is an AMR subgraph. Based on the assumption, event extraction task is considered as a graph identification problem. And a supervised neural network-based model is trained to identify the event subgraph.

Dohare et al. [34] generated the story AMR graph using co-reference resolution and Meta Nodes for automatic summarization.

Song et al. [37] utilized a graph recurrent network (GRN) to encode AMR graphs without breaking the original graph structure, experimental results on an English-to-German dataset show that the structural semantic information from AMR can improve neural machine translation.

Compared to the conventional dependency parsing, AMR captures deeper and more semantic meaning [36]. To our knowledge, no existing work has exploited semantic structure of AMR for enhancing aspect extraction. To fill in this gap, this paper employs ASP rule based method to explore the usefulness of semantic structure in aspect extraction. That is because the existing studies all adopted neural network-based methods, no studies use rule-based method to explore the usefulness of AMR in NLP tasks. Thus, we employ deep semantic representation based on AMR to capture deep semantic information.

III. Deep2S MODEL

In this section, we first introduce problem definition where important concepts about Deep2S model are formulated. Then, we describe the entire framework of Deep2S. Finally, we give a detailed introduction of the four steps in the
framework, namely, ASP fact generation, semantic rule generation, syntactical rules used in our model and ASP answer set computing.

A. PROBLEM FORMULATION

In this subsection we formulate the problem to be investigated. To illustrate AMR can capture deep and rich semantic information, we start with some basic definitions.

**Definition 1 (Identity Tree):** For any sentence \( S \), \( S \) is sentence space, for \( \forall S \in S, S \equiv < w, r >, w \) is a sequence of words, \( r = [r_{ij}, i = w_i, j = w_j] \). There exists mapping \( f: S \rightarrow T \) and its inverse mapping \( f^{-1}: T \rightarrow S \), \( T \) is tree space. If satisfying \( T = f(S) \) and \( S = f^{-1}(T) \), where tree \( T \in T \), and \( T = < V, E, > \), \( V \) is a set of nodes in \( T \), \( E \) is the set of edges in \( T \), then \( f \) is the identity mapping of the sentence \( S \), and \( T \) is the identity tree of the sentence \( S \). That is, specify that each word \( w_i \) in \( S \) has one-to-one correspondence with a node \( v \) in the tree \( T \), \( v \in V \), and the relation between words, \( r_{ij} \) has one-to-one correspondence with the edge \( e_{ij} \) in the \( T \), \( e_{ij} \in E \), and vice versa.

![FIGURE 2. Dependency trees and AMR graphs of "The camera has a good battery." and "I recommend unreservedly the camera to any potential buyer"
](image)

For example, two syntactic trees in Figure 2 (a) and (b) can be taken as identity tree, since they can be mapped one-to-one with sentences respectively, and vice versa. Consider tree as a special case of graph, but AMR graph is not taken as an identity tree, because AMR graphs cannot be mapped one-to-one with sentences, as is shown in Figure 2 (c) and (d).

**Definition 2 (Deep Semantics):** For any sentence \( S \), if exists identity mapping \( f \) making \( T = f(S) \) and any mapping \( f' \) making \( T' = f'(S) \), where \( T' = < V', E', > \), then tree \( T' \) is considered to have deep semantics over \( T \), if one of the following conditions is met:

1. if \( \text{depth}(T') < \text{depth}(T) \), namely, the depth of the tree \( T' \) is less than the depth of its identity tree \( T \);
2. there exists words \( w_1, w_2 \), which are mapped as nodes \( A, B \) in \( T \) and \( A', B' \) in \( T' \); if \( \text{distance}(A) < \text{distance}(A') \) or \( \text{distance}(B) < \text{distance}(B') \), where \( A \rightarrow B \) represents the path from node \( A \) to node \( B \) in identity tree \( T \). Namely, node \( A \) in tree \( T' \) is taken as the starting node, and its path to the designated node \( B \) is less than that of the identity tree \( T \).

For example, “The camera has a good battery.”, the depth of AMR graph (in Figure 2 (c)) is less than that of syntactic tree (in Figure 2 (a)). This means that semantic representation has deep semantics.

**Definition 3 (Rich Semantics):** For any sentence \( S \), if \( \exists w_i, r_{ij} \in S, w_i \in C, r_{ij} \in C, w_i, r_{ij} \) satisfy the mapping \( g: C \rightarrow Z \), where \( C \) is ontology name space, \( Z \) is ontology space, meanwhile, if exists identity mapping \( f \) making \( T = f(S) \) and any mapping \( f' \) making \( T' = f'(S) \), where \( T' = < V', E' > \), then \( T' \) is considered to have rich semantics over \( T \), if one of the following conditions is met:

1. if enabling the word \( w_i \) (e.g., buyer) to abstract into an ontology \( z \) (e.g., person in Figure 2 (d)) \( [38], z = g(w_i), z \in V' \), then this node \( z \) is considered to have rich semantics;
2. if enabling the implicit semantic relation \( r_{ij} \) in \( S \) to abstract into ontology, \( z = g(r_{ij}), z \in E' \), then the edge \( z \) is considered to have rich semantics.

For example, “The camera has a good battery.”, the semantic relation “domain” between “camera” and “battery” is considered as ontology, which can be captured by semantic parser, not by syntactic parser. This means that semantic representation has rich semantics.

**Definition 4 (AMR Graph):** An AMR graph \( G = (V, E) \) is a rooted, directed, acyclic graph, where \( V \) is the set of nodes, or equivalently the vocabulary of entity, and \( E \) is the set of edges. Each edge in \( E \) connects one entity \( v_i \in V \) with another entity \( v_j \in V \) via a semantic relation type \( e_{ij} \in E \). \( N \) is a set of semantic relations. Each edge in \( E \) is denoted as a triple \( T = (v_i, e_{ij}, v_j) \).

**Definition 5 (AMR Subgraph):** We hypothesize that semantic relation between aspects and opinion words or other entities is an AMR subgraph. Let \( M \) be a set of semantic relations conforming to this condition, which is a subset of \( N \). Let \( G_s = (V_s, E_s) \) represent AMR subgraph, where \( V_s \) is a subset of \( V, E_s \) is a subset of \( E \). \( G_s \) is a subgraph of \( G \).

B. ASP BASED FRAMEWORK FOR ASPECT EXTRACTION

Answer Set Programming originates from non-monotonic logic and logic programming. It is a logic programming paradigm based on the answer set semantics [7], [39], which offers an elegant declarative semantics to the negation as failure operator in Prolog. An ASP program consists of rules of the form:

\[ l_0 : -l_1, \ldots, -l_m, \quad \text{not} \quad l_{m+1}, \ldots, \quad \text{not} \quad l_n. \]

where each \( l_i \) for \( i \in [0 \ldots n] \) is a literal of some signature, i.e., expressions of the form \( p(t) \) or \( \neg p(t) \) where \( p \) is a predicate and \( t \) is a term, and not is called negation as failure or default negation. For instance, \( \text{amr}(A, \text{mod}, O) \) is
an atom with the predicate $amr$, and three terms, one constant $mod$ and two variables $A$ and $O$, respectively. A rule without body is called a fact.

Our proposed model Deep2S is based on ASP framework for extracting aspects. The entire ASP based framework consists of the following steps:

1. Parse review texts, develop algorithms to extract related information and represent them as ASP facts;
2. Construct semantic rules and represent them by ASP rules;
3. Use syntactical rules as base and represent them by ASP rules;
4. Compute the answer set of the logic program resulted from the first and the third steps to perform shallow aspect extraction, and then from the first and the second steps to perform deep aspect extraction using an ASP solver like clingo.\(^1\) By integrating both syntactical-based and semantic-based rules, Deep2S finally extracts aspect terms from the answer set.

### C. ASP FACT GENERATION

In this paper, we hypothesize that the semantic relations between aspects and opinion words or other entities can be captured by AMR subgraphs. In order to obtain AMR subgraph, an algorithm called AMR-SE (short for AMR Subgraph Extraction) given in Algorithm 1 is developed.

AMR-SE includes two main steps:

**Step 1 (Lines 1-3):** This step is to obtain AMR graph. Line 1 initializes AMR subgraph $G_s$ as empty. AMR-SE starts with reading each review sentence $S$ from the training data $D$. For each sentence $S$, AMR parser is used to automatically generate meaning representation, namely, AMR graph $G = (V, E)$ (lines 2-3). The time complexity of parsing is $O(n)$, $n$ is the number of the training data $D$.

**Step 2 (Lines 4-18):** This step is to identify AMR subgraphs from the AMR graph. AMR subgraph may contain the candidate aspects. After obtaining AMR graph $G = (V, E)$ of a sentence (line 4), we identify AMR subgraphs conforming to semantic relation set $\mathcal{M}$. For each $e_{ij}$ in $E$ (line 5), we do the following steps. If the edge $e_{ij}$ belongs to semantic relation set $\mathcal{M}$, the triple $(v_i, e_{ij}, v_j)$ will be added into $G_s$ (lines 6-7). Then we search the edges connecting with the nodes $v_i$ and $v_j$, respectively. If the edge $e_{ih}$ connecting with the node $v_i$ belongs to $\mathcal{M}$, we insert the triple $(v_i, e_{ih}, v_h)$ into $G_s$ (lines 8-10). If the edge $e_{jk}$ connecting with the node $v_j$ belongs to $\mathcal{M}$, we insert the triple $(v_j, e_{jk}, v_k)$ into $G_s$ (lines 11-13). After adding all nodes and edges related to semantic relation set $\mathcal{M}$ into $G_s$, we can obtain AMR subgraphs of the candidate aspects (line 18). The time complexity of identifying AMR subgraphs is $O(m)$, $m$ is the number of AMR subgraphs. Therefore, the total time complexity of Algorithm 1 is $O(mn)$.

After identifying AMR subgraphs from AMR graph, we can represent them as ASP facts. For example, in

**Algorithm 1 AMR-SE($D, \mathcal{M}$)**

- **Input:** Training data $D$, semantic relation set $\mathcal{M}$
- **Output:** AMR subgraph $G_s$

```plaintext
1: $G_s = ()$; //Initialize $G_s$ as empty
2: for each sentence $S$ in $D$ do
3:   parse $S$ into AMR graph $G = (V, E)$;
4: for $G = (V, E)$ of $S$ do
5:   for each $e_{ij}$ in $E$ do
6:     if $e_{ij} \in \mathcal{M}$ then
7:       insert $(v_i, e_{ij}, v_j)$ into $G_s$;
8:     if $e_{ih}$ in $\mathcal{M}$, when $e_{ih}$ connecting with $v_i$ then
9:       insert $(v_i, e_{ih}, v_h)$ into $G_s$;
10:      end if
11:     if $e_{jk}$ in $\mathcal{M}$, when $e_{jk}$ connecting with $v_j$ then
12:       insert $(v_j, e_{jk}, v_k)$ into $G_s$;
13:      end if
14:     end if
15: end for
16: end for
17: end for
18: Output $G_s$ as AMR subgraph.
```

Figure 1 (a), AMR subgraphs like “(battery, mod, good)” and “(battery, domain, camera)” can be represented as ASP facts like $f_1 - f_2$:

- $f_1$ $amr$ (“battery”, “mod”, “good”).
- $f_2$ $amr$ (“battery”, “domain”, “camera”).

### D. SEMANTIC RULE GENERATION

This subsection explains in detail how to generate semantic rules. In order to obtain semantic rules, we first construct the seed rules manually, then propose an algorithm to extend them and obtain the final rule set, and finally detail some examples of the final semantic rules. In the following, we start with several related key concepts used in semantic rules.

**TABLE 1. Description of semantic relations.**

| Relation | Description |
|----------|-------------|
| arg0     | who is the recommender |
| arg1     | what is the thing recommended |
| arg2     | whom is being recommended to |
| manner   | recommend manner |
| mod      | the relationship between a word and its modifier |
| duration | the length of time that sth lasts or continues |
| domain   | general semantic relation, domain |
| degree   | represent intensifiers, comparatives and superlatives |
| compared-to | represent comparatives and superlatives |
| pos      | represents the number of possession |
| and, or  | represent conjunction along with opx |

The first concept is semantic relation. Since AMR can capture the notation of “who is doing what to whom”, AMR graph is adopted to describe semantic relation of a sentence. It uses approximately 100 relations. And a dozen of them are used in this paper, whose meaning is shown in Table 1. These semantic relations are illustrated by an example of
TABLE 2. Four types of seed semantic rules based on the propagation mechanism in DP.

| RuleID | Input & Output | Seed Semantic Rules |
|--------|----------------|---------------------|
| R1     | O \rightarrow A | aspect (A) \sim amr(A, NR, O), opinionword(O), pos(A, NN), pos(O, JJ), not generalWord(A). |
| R2     | A \rightarrow O   | opinionword(O) \sim amr(O, Osem, H), amr(H, Asen, A), aspect(A), pos(O, JJ). |
| R3     | A_i \rightarrow A_j | aspect (A_j) \sim amr(A_i, NR, A), aspect(A_i), pos(A_i, NN), not generalWord(A_i). |
| R4     | O_i \rightarrow O_j | opinionword(O_j) \sim amr(O_i, Semf, H), amr(O_i, Sems, H), Semf \sim Sems, opinionword(O_j), pos(O_j, JJ). |

1) SEED RULE CONSTRUCTION
In light of syntactical rules in DP [4], this paper constructs manually four types of seed semantic rules based on the development datasets. The propagation mechanism in DP is still utilized in semantic rules in the process of aspect extraction. Table 2 displays the four types of seed semantic rules.

R1: extracting the aspects using the given seed opinion words.
R2: extracting new opinion words using the extracted aspects in R1.
R3: extracting new aspects using the extracted aspects in R1.
R4: extracting new opinion words using the extracted opinion words in R2 and the given opinion words.

In the table, column 1 is the rule ID, column 2 is the input and output of rules, and column 3 is the four types of seed rules. In these rules, “NR”, “Asen”, “Osem”, “Sems” and “Semf” belong to semantic relations M, “NN” contains noun (nn) and noun phrase (np), “JJ” contains adjective (jj) and adjective phrase (jp). “A” stands for aspect. “O” stands for opinion word. “H” means any word. Take the first rule as an example,

aspect (A) \sim amr(A, NR, O), opinionword(O), pos(A, NN), pos(O, JJ), not generalWord(A).

2) FINAL RULE SELECTION
After constructing the seed semantic rules R, the next step is to extend R and obtain the final extraction rules. In order to do this, an algorithm named DeepSem is proposed, as shown in Algorithm 2. It consists of three main steps:

Step 1 (Lines 1-4): This step is to obtain the extended rules \text{ER}. Line 1 initializes the extended rules \text{ER} and the final rule set \text{FRS} as empty. For each rule r_i in the seed semantic...
rules \( \mathcal{R} \), we expand it by replacing relations \( \text{REL} \) (including “NR”, “Asem”, “Osem”, “Sems” and “Semf”) in \( r_j \) with semantic relations \( M \), replacing \( NN \) with noun (nn) and noun phrase (np) and replacing JJ with adjective (jj) and adjective phrase (jp) or discarding JJ, and finally obtain the extended rules \( \mathcal{E}_R \) (lines 2-4).

Step 2 (Lines 5-10): This step is to identify the rules which can extract the candidate aspects from the extended rules \( \mathcal{E}_R \). For each rule \( r_j \) in \( \mathcal{E}_R \), perform aspect extraction using an ASP solver based on ASP facts \( \mathcal{F} \) and rule \( r_j \) (lines 5-6). If rule \( r_j \) cannot extract any aspect, delete \( r_j \) from \( \mathcal{E}_R \) (lines 7-9). Thus the remaining rules can extract one or more candidate aspects.

Step 3 (Lines 11-15): This step is to rank the rules based on their precision. For each rule \( r_k \) in extended rules \( \mathcal{E}_R \), compute its precision, follow the descending order to rank these rules and insert \( r_k \) at the end of \( \mathcal{F}_R \mathcal{S} \) (lines 12-14). Then we can obtain the final rule subset.

In the following, we detail the semantic rules to extract the aspects in Figure 2 (c) and (d).

\textbf{Rule1:} Rule1 means “if a word \( A \), whose POS is noun (nn), has the semantic relation \( \text{mod} \) with an opinion word \( O \), and has the semantic relation \( \text{domain} \) with \( H \), then \( A \) is an aspect”, which can be formulated by the following ASP rule:

\[
\text{aspect}(A) :- \text{amr}(A, \text{mod}, O),
\text{amr}(A, \text{domain}, H),
\text{opinionword}(O), \text{pos}(A, \text{nn}),
\text{not generalWord}(A).
\]

where \text{not} is used to exclude the general words such as “person”, “thing” and so on. \( \text{amr}(A, \text{mod}, O) \) represents there is a semantic relation \( \text{mod} \) between \( A \) and \( O \). \( \text{amr}(A, \text{domain}, H) \) represents that there is a semantic relation \( \text{domain} \) between \( A \) and \( H \). If we know \( H \) is a product, \( A \) may be an aspect of \( H \) based on \( \text{amr}(A, \text{domain}, H) \). For example, from the AMR graph in Figure 2 (c), we can extract the aspect “battery”, as there is a relation \( \text{mod} \) between “good” and “battery” and a relation \( \text{domain} \) between “battery” and “camera”, “good” is an opinion word.

\textbf{Rule2:} Rule1 means “if a word \( A \) whose POS is noun (nn) is not a general word, a word \( H \) is a trigger word, it has the semantic relation \( \text{arg1} \) with its direct object \( A \) and has the semantic relation \( \text{manner} \) with an opinion word \( O \), then \( A \) is an aspect”. This rule can be represented as an ASP rule:

\[
\text{aspect}(A) :- \text{amr}(A, \text{arg1}, A),
\text{amr}(H, \text{manner}, O),
\text{opinionword}(O),
\text{pos}(A, \text{nn}), \text{triggerWord}(H),
\text{not generalWord}(A).
\]

For example, from the AMR graph in Figure 2 (d), we can extract the aspect “camera”, as there is a relation \( \text{arg1} \) between “recommend” and “camera” as well as a relation \( \text{manner} \) between “recommend” and “unreservedly”, “recommend” is a trigger word, and “unreservedly” is an opinion word.

\section{Syntactical Rules}

In this work, syntactical dependency relations between opinion words and opinion targets are adopted to extract opinion targets. Syntactical rules from DP \cite{4} and RSDP \cite{8} are used as the base in our work. Detailed information about these rules can be found in \cite{4} and \cite{8}.

In the following, we detail the syntactical rule for extracting the aspect in Figure 2 (a). For example, one syntactical rule from DP \cite{4} is that if there is an adjectival modifier \( \text{amod} \) between \( A \) and \( O \), where \( A \) is a noun and \( O \) is an opinion word, then \( A \) is identified as an aspect. This can be represented as an ASP rule \textbf{Rule21:}

\[
\text{aspect}(A) :- \text{depends}(A, \text{amod}, O),
\text{pos}(A, \text{nn}),
\text{opinionword}(O).
\]

where \text{depends}(A, \text{amod}, O) represents there is a dependency relation \( \text{amod} \) between \( A \) and \( O \), \text{pos}(A, nn) represents \( A \) is a singular noun, \text{opinionword}(O) represents \( O \) is an opinion word, \( \text{aspect}(A) \) represents \( A \) is an opinion target. For example, from “The camera has a good battery.” in Figure 2 (a), we can extract the aspect “battery” as there is a relation \( \text{amod} \) between “good” and “battery”, “good” is an opinion word, and “battery” is a noun.

\section{ASP Answer Set Computing}

After obtaining ASP facts and ASP rules, we compute the answer set of the logic program using an ASP solver. Then the aspect terms are extracted from the answer set.

The following is an example to illustrate how our logic framework works for aspect extraction. For the sentence “The camera has a good battery.”, the POS facts of its words and the syntactic facts between words are automatically acquired by Stanford Parser,\footnote{http://nlp.stanford.edu:8080/parser/} and the semantic facts between words are automatically obtained by JAMR\footnote{https://github.com/jflanigan/jamr} \cite{41}. These facts are represented by program \( P_1 \). Then, let \( P_2 \) be the program of the ASP rules. Based on rules \textbf{Rule1}, \textbf{Rule2} and facts \( f_1 - f_{14} \), we run an ASP solver like clingo to compute the answer sets of \( P = P_1 \cup P_2 \). Finally, the solver outputs a unique answer set including all atoms in \( P_1 \) and the aspect will be extracted from the sentence, namely, \text{aspect} (“battery”).

\[
\begin{align*}
\text{f}_1 & \quad \text{amr} ("\text{battery"}, \text{mod}, "\text{good"}). \\
\text{f}_2 & \quad \text{amr} ("\text{battery"}, \text{domain}, "\text{camera"}). \\
\text{f}_3 & \quad \text{depends} ("\text{camera"}, \text{det}, "\text{The"}). \\
\text{f}_4 & \quad \text{depends} ("\text{has"}, \text{nsubj}, "\text{camera"}). \\
\text{f}_5 & \quad \text{depends} ("\text{has"}, \text{dobj}, "\text{battery"}). \\
\text{f}_6 & \quad \text{depends} ("\text{battery"}, \text{det}, "\text{a"}). \\
\text{f}_7 & \quad \text{depends} ("\text{battery"}, \text{amod}, "\text{good"}). \\
\text{f}_8 & \quad \text{pos} ("\text{The"}, \text{dt}). \\
\text{f}_9 & \quad \text{pos} ("\text{camera"}, \text{nn}). \\
\text{f}_{10} & \quad \text{pos} ("\text{has"}, \text{vbz}). \\
\text{f}_{11} & \quad \text{pos} ("\text{a"}, \text{dt}). \\
\text{f}_{12} & \quad \text{pos} ("\text{good"}, \text{jj}). \\
\text{f}_{13} & \quad \text{pos} ("\text{battery"}, \text{nn}). \\
\text{f}_{14} & \quad \text{opinionword} ("\text{good"}).
\end{align*}
\]
IV. EXPERIMENTS

We now evaluate the proposed method, first introduce the experiment datasets, evaluation metrics and comparative methods, then detail experimental results.

A. DATASETS

Three publicly available aspect-annotated corpora are used. The first one is from [40] which contains five review datasets from four domains: digital cameras (D1, D2), cell phone (D3), MP3 player (D4), and DVD player (D5). The second one is from SemEval-2014 and SemEval-2015 which contains three review datasets from two domains: SemEval-2014 Restaurants (D6), SemEval-2014 Laptops (D7), and SemEval-2015 Restaurants (D8). The third one is from [8], including reviews of three products: computer (D9), wireless router (D10), and speaker (D11), which are used as the development datasets to construct the seed semantic rules manually. The seed opinion words of the first and second corpus are provided by [40] and the original annotated datasets, respectively. Table 3 shows the detailed information of each review dataset.

| Data  | Product                   | # of Sentences | # of Aspects |
|-------|---------------------------|----------------|--------------|
| D1    | Digital camera            | 597            | 237          |
| D2    | Digital camera            | 346            | 174          |
| D3    | Cell phone                | 546            | 302          |
| D4    | MP3 player                | 1716           | 674          |
| D5    | DVD player                | 740            | 296          |
| D6    | Restaurant-14(Train)      | 3044           | 3699         |
| D7    | Restaurant-14(Test)       | 800            | 1134         |
| D8    | Restaurant-15(Train)      | 3048           | 2373         |
| D9    | Restaurant-15(Test)       | 800            | 654          |
| D10   | Restaurant-15(Test)       | 1315           | 1279         |
| D11   | Restaurant-15(Test)       | 685            | 597          |
| D12   | Computer(Development)     | 531            | 354          |
| D13   | Speaker(Development)      | 879            | 307          |
| D14   | Speaker(Development)      | 689            | 440          |

For the first corpus, due to the small size of each dataset, we have followed the same way of cross domain test as adopted by [8]. Namely, for D1 to D5, leave-one-out cross validation is utilized. For example, the annotated data from D1 to D4 is used for selecting rules, then the selected rules are used to test D5. According to [8]’s statement, the rules selected through this way can be proven to be domain independent. For the second corpus, the train-test split is the same as the original dataset, as shown in Table 3.

B. EVALUATION METRICS

We have adopted precision, recall, and F1-score as our evaluation metrics. For the comparison, we choose the same two ways used by [8] to compute the extracting results: (1) based on multiple occurrences of each aspect term, and (2) based on distinct occurrence of each aspect term.

In customer reviews, users often review an opinion target many times, for example, the target “camera” is frequently mentioned in camera dataset. For (1), if any occurrence of “camera” is extracted, it means that all occurrences of “camera” are identified. If no one occurrence of “camera” is identified, it means that all the occurrences have been lost. For (2), if any occurrence of “camera” is identified, it is regarded as one extraction. If no one is identified, it is regarded as one loss. Both of the above evaluation metrics have their own importance. In (1), the most frequent opinion targets are considered with respect to their frequency and it means that identifying (or losing) the most frequent opinion targets is identifying (or losing) the most important opinion targets. And in (2), all the opinion targets have the same importance.

As evaluation measures for aspect extraction, we employ true positive (TP), false positive (FP) and false negative (FN) values to calculate precision, recall, and F1-score. To calculate these values, let \( A \) be a set of extracted aspects and let \( T \) be the set of manually annotated aspects. Hence, TP is \( |A \cap T| \), FP is \( |A \setminus T| \), FN is \( |T \setminus A| \).

For (1), we follow the same precision, recall and F1-score metrics as adopted by [8].

\[
\text{Precision} = \frac{\sum_{i=1}^{|A|} f_i \times E(a_i, A)}{\sum_{i=1}^{|A|} f_i}, \\
\text{Recall} = \frac{\sum_{i=1}^{|T|} f_i \times E(a_i, T)}{\sum_{i=1}^{|T|} f_i}, \\
F_1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},
\]

where \( f_i \) is the term frequency of \( a_i \), \( E(a_i, A) \) (or \( E(a_i, T) \)) equals to 1 if \( a_i \) is an element of \( A \) (or \( T \)), otherwise \( E(a_i, A) \) (or \( E(a_i, T) \)) equals to 0.

For (2), the standard evaluation metrics are adopted as follows, F1-score is the same as in (1):

\[
\text{Precision} = \frac{TP}{TP + FP}, \\
\text{Recall} = \frac{TP}{TP + FN}, \\
F_1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\]

C. COMPARATIVE METHODS

To validate the performance of our proposed method Deep2S on aspect extraction, we compare Deep2S against the syntactical rule based baselines (e.g., DP, DP+, RSDP, RSDP+) as well as the state-of-the-art and most recent baselines (e.g., TF-RBM, CNN-LP, R-DNN). We compare with DP, DP+, RSDP and RSDP+ because we utilize DP, DP+ rules and RSDP, RSDP+ rules as the part of our method.

DP and DP+ refer to the original double propagation algorithm in [4], and utilize 8 aspect extraction syntactical patterns. The difference between them is that the former uses 8 dependency relations (i.e., mod, subj, s, obj, obj2, sobj, obj2, sobj),

4https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
5http://alt.qcri.org/semeval2014/task4/index.php?id=data-and-tools
6http://alt.qcri.org/semeval2015/task12/index.php?id=data-and-tools
RSDP and RSDP+ perform aspect extraction by selecting the optimal rule subset from all the rules of DP and DP+ using the greedy algorithm, respectively. The two baselines are proposed in [8].

DeepSem utilizes the final semantic rules to extract aspects.

Deep2S-DP combines DeepSem and all the rules of DP. Deep2S-DP+ combines DeepSem and all the rules of DP+.

Deep2S-RSDP combines DeepSem and all the rules of RSDP.

Deep2S-RSDP+ combines DeepSem and all the rules of RSDP+.

TF-RBM is a sequential pattern rules-based aspect extraction approach proposed in [6].

CNN-LP is a supervised approach which combines deep convolutional neural network and linguistic patterns to improve the aspect extraction performance, which is proposed in [42].

R-DNN is a hybrid unsupervised approach which combines chunk-level linguistic rules and deep neural network for aspect extraction, which is proposed in [22].

D. EXPERIMENTAL RESULTS

Table 4 shows the comparison of precision of our proposed method Deep2S (e.g., Deep2S-DP, Deep2S-DP+, Deep2S-RSDP, and Deep2S-RSDP+) compared with the syntactical rule based baselines (e.g., DP, DP+, RSDP and RSDP+), the semantic rule based baseline (e.g., DeepSem), the sequential pattern rules-based baseline (e.g., TF-RBM), as well as the baseline combining rules and neural network (e.g., CNN-LP). Similarly, Table 5 shows the comparison of their recall and Table 6 shows the comparison of their $F_1$-scores. For each method, column 1 labeled “mul” refers to the first evaluation metric, namely, based on multiple occurrences of each aspect term, and column 2 labeled “dis” refers to the second evaluation metric, namely, based on distinct occurrence of each aspect term. All these methods utilized the same resources, as we have followed, thus, comparison of these methods is achievable.

In our experiments, DP, DP+, RSDP and RSDP+ are the bases, our approaches (e.g., Deep2S-DP, Deep2S-DP+, Deep2S-RSDP, and Deep2S-RSDP+) combine DeepSem with them. From the three tables, we observe that our approaches perform more consistently for all evaluation metrics of both multiple and distinct evaluations and produce better results than the bases. In Table 4, for both D1 to D5, and D6 to D8, Deep2S-RSDP gets the best average precisions. In Table 5, for both D1 to D5, and D6 to D8, Deep2S-DP+ gets the best average recalls. In Table 6, for both D1 to D5, and D6 to D8, Deep2S-RSDP+ gets the best average $F_1$-scores. It is obvious that our approaches outperform the bases significantly, which proves the semantic rules can mine deep and rich linguistic information hidden in user’s review sentences. This semantic information can help improve the performance of aspect extraction. For example, in Figure 1 (a), the semantic relation “domain” which means “battery” is an aspect or attribute of “camera” can be captured to help extracting aspect.
From Table 4, Table 5, and Table 6, we can see that the average results of Deep2S-DP and Deep2S-DP+ are higher than those of DP and DP+. And the average results of Deep2S-RSDP and Deep2S-RSDP+ are higher than those of RSDP and RSDP+. Compared with the corresponding four bases on two evaluation metrics, for D1 to D5, the F1-scores of Deep2S-DP, Deep2S-DP+, Deep2S-RSDP, and Deep2S-RSDP+ improve by 4.4% and 4.2%, 1.5% and 5.1%, 5% and 6.3%, 2.8% and 4.6%; for D6 to D8, improve by 1.2% and 1.6%, 0.2% and 0.9%, 2.5% and 3.5%, 1.4% and 1.7%, respectively. This means that our methods adding semantic rules are markedly better than the bases which only use syntactical rules in shallow structure.

In order to validate where the performance gain is coming from, we conduct the experiment of DeepSem. As we all know, the improved performance may be affected by the two factors — one is the use of semantic information, and another is the additional number of rules. From the three tables, we observe that for D1 to D5, the average precision results of DeepSem are higher than those of DP and DP+, while the average recall results of DeepSem are higher than DP and lower than DP+. This is expected because DP+ has more rules than DeepSem. The average F1-scores of DeepSem are higher than both of them. For D6 to D8, the average precision results of DeepSem are higher than DP and DP+ while its recall results are lower than both of them, and the average F1-scores of DeepSem are slightly less than them. This shows that DeepSem may capture the deep and rich semantic knowledge of those review texts such as grammatically incorrect sentences or excessively long sentences. However, there is some loss in recalls, this is expected because DeepSem adds less syntactical rules. These results are in accord with [8].

Another recent approach combining linguistic rules with neural network (i.e., R-DNN) is employed to compare with our best method (Deep2S-RSDP+), as shown in Figure 3. We can observe that Deep2S-RSDP+ outperforms R-DNN significantly in both D6 and D7. Specifically, Deep2S-RSDP+ performs better performance than R-DNN in terms of precision and F1-scores but drops a little in recall. Compared with R-DNN, the recall of Deep2S-RSDP+ decreases 7.2% and 2.5% but precision improves by 12.9% and 23.2% and the F1-score by 2.4% and 10.0%, respectively. We believe one of the key reasons is that our method uses semantic rules to capture deep and rich information such as “domain” which R-DNN may be not.
on almost every dataset compared with Deep2S-RSDP, although the precision of Deep2S-RSDP+ decreases slightly on some datasets. The average F1-score of Deep2S-RSDP+ is improved compared with Deep2S-RSDP. This shows that Deep2S-RSDP+ is able to take good advantage of the additional syntactical rules (in RSDP+), maintaining a high recall with a little loss in precision. We also observe that the overall recalls of Deep2S-DP and Deep2S-DP+ are higher than Deep2S-RSDP and Deep2S-RSDP+, this is expected because Deep2S-RSDP and Deep2S-RSDP+ have less syntactical rules.

In summary, our experiment results show that both semantic rule based method and syntactic rule based method provide valuable information for aspect extraction. Our best method Deep2S-RSDP+ on two evaluation metrics outperforms the state-of-the-art syntactical rule based method RSDP+, and the sequential pattern rules-based method (e.g., TF-RBM), as well as the methods combining rules and neural network. This result indicates that the semantic rules can be utilized in conjunction with the syntactic rules to obtain the best performance. This is consistent with previous study [43], where semantic structures were combined with syntactic structures through embeddings for DDI extraction. Since the explainability and scalability of rules, our proposed method can combine syntactic structures with semantic structures by rules. We can conclude that the proposed approach can capture shallow structure and deep semantic information, and can improve the performance of aspect extraction.

E. ANALYSIS

Semantic structure-based methods are hypothesized to have less rules, since the sentences with the same meaning but different syntactic structures such as Example 1 and Example 2 have the same semantic representation of AMR graph, as shown in Figure 1.

To validate this hypothesis, we compare semantic structure-based method (e.g., DeepSem) with syntactic structure-based methods (e.g., DP, DP+, RSDP, RSDP+). Note that DP is the general rule while DP+, RSDP, and RSDP+ are the refined rules of DP. The datasets used in our experiment have more or less the same rule number, so we take the dataset D1 as an example. As shown in Figure 4, the rule number of DP is 8, fewer than that of DP+ (521), RSDP (36), RSDP+ (270), respectively. We can see that the refined rules (e.g., DP+, RSDP, RSDP+) require more rules than the general rule (e.g., DP). DP+ has more rules, it achieves higher recall and lower precision than RSDP, RSDP+, as shown in Figure 5.

Note that RSDP, RSDP+ and DeepSem use the same greedy algorithm to select the final rule set. We can observe that DeepSem has less rules than RSDP, RSDP+ and it achieves higher recall than both of them. This may be because DeepSem can capture deep semantic relation (e.g., “domain”) between concepts (“battery” and “camera”) in the different sentences such as Example 1 and Example 2 which have the same semantic structure.

DeepSem achieves lower precision than RSDP and RSDP+; but higher precision than DP and DP+. The results using semantic structure alone are not promising as we expected. One reason might be that the semantic structure discards some surface linguistic knowledge such as some surface words and dependency relations which may be important for extracting aspects. And another reason might be due to the limitations of semantic structure parser such as JAMR which cannot deeply capture noun-noun or noun-adjective relations. And these noun-noun or noun-adjective relations are useful for aspect extraction.

V. CONCLUSION AND FUTURE WORK

In this paper, we first propose to employ ASP to explore the usefulness of semantic structure of AMR in aspect extraction. Although its performance is not so good as syntactic structure of dependency grammar, the experimental results demonstrated that ASP could be used effectively and concisely for the study of semantic structure. Then we propose a hybrid rule-based approach with the capacity to capture
structure and deep semantic for improving the performance of aspect extraction in opinion mining. The proposed methods are capable of employing syntactical rule-based approach to capture surface structure information, and exploiting semantic rule-based approach to capture deep semantic information. Compared with the baselines, our proposed method produces better results on two evaluation metrics. For example, our method outperforms the best syntactical rule based method by 2.8% and 4.6% for D1 to D5, 1.4% and 1.7% for D6 to D8, respectively. We also compared them with the sequential pattern rules-based baseline and the state-of-the-art statistical based baselines. The experimental results demonstrated their superior performance. Our findings suggest that semantic structure is a powerful semantic formalism which holds potential for aspect extraction task.

In our future work, we plan to employ semantic rules to extract implicit aspects and validate the usefulness of deep semantic information using the datasets in other languages like Chinese. We also plan to explore dependency based embeddings and semantic based embedding to extract aspects. Moreover, we will explore the different compositional strategies to combine semantic and syntactic structures as well as part of speech.

REFERENCES

[1] B. Liu, Sentiment Analysis and Opinion Mining. San Rafael, CA, USA: Morgan & Claypool, 2012.
[2] H. Peng, L. Xu, L. Bing, W. Lu, and F. Huang, “Knowing what, how and why: A near complete solution for aspect-based sentiment analysis,” in Proc. 34th AAAI Conf. Artif. Intell., 2020, pp. 1–9.
[3] R. Wang, D. Zhou, M. Jiang, J. Si, and Y. Yang, “A survey on opinion mining: From stance to product aspect,” IEEE Access, vol. 7, pp. 41101–41124, 2019.
[4] G. Qiu, B. Liu, J. Bu, and C. Chen. “Opinion word expansion and target extraction through dialogue propagation,” Comput. Linguistics, vol. 37, no. 1, pp. 9–27, Mar. 2011.
[5] L. Banerescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. “Abstract meaning representation for sembanking,” in Proc. 7th Linguistic Annotation Workshop Interoperability Discourse, 2013, pp. 178–186.
[6] T. A. Rana and Y.-N. Cheah, “A two-fold rule-based model for aspect extraction,” Expert Syst. Appl., vol. 89, pp. 273–285, Dec. 2017.
[7] M. Gelfond and V. Lifschitz. “The stable model semantics for logic programming,” in Proc. ICLP/SLP, vol. 88, 1988, pp. 1070–1080.
[8] Q. Liu, Z. Gao, B. Liu, and Y. Zhang. “Automated rule selection for aspect extraction in opinion mining,” in Proc. 24th Int. Joint Conf. Artif. Intell., 2015, pp. 1291–1297.
[9] A. Komninos and S. Manandhar, “Dependency based embeddings for sentence classification tasks,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Student Lang. Technol., 2016, pp. 1490–1500.
[10] S. MacAvaney and A. Zeldes. “A deeper look into dependency-based word embeddings,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Student Res. Workshop, 2018, pp. 40–45.
[11] C. Li, J. Li, Y. Song, and Z. Lin, “Training and evaluating improved dependency-based word embeddings,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 5836–5842.
[12] Y. Zhang, P. Qi, and C. D. Manning. “Graph convolution over pruned dependency trees improves relation extraction,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 2205–2215.
[13] P. Gupta, S. Rajaram, H. Schütze, and T. Runkler, “Neural relation extraction within and across sentence boundaries,” in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 6513–6520.
[14] H. Yu, W. Xu, S. Lin, and Q. Liu, “Machine translation evaluation metric based on dependency parsing model,” ACM Trans. Asian Low-Resource Lang. Inf. Process., vol. 18, no. 4, pp. 44–59, 2019.
[15] Y. Wu, Q. Zhang, X. Huang, and L. Wu, “Phrase dependency parsing for opinion mining,” in Proc. EMNLP, 2009, pp. 1533–1541.
[16] Q. Liu, B. Liu, Y. Zhang, D. S. Kim, and Z. Gao, “Improving opinion aspect extraction using semantic similarity and aspect associations,” in Proc. 13th AAAI Conf. Intell., 2016, pp. 2986–2992.
[17] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, “Recursive neural condition random fields for aspect-based sentiment analysis,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2016, pp. 616–626.
[18] L. Zang, B. Liu, S. H. Lim, and E. O’Brien-Strain, “Extracting and ranking product features in opinion documents,” in Proc. COLING, 2010, pp. 1462–1470.
[19] Y. Zhao, W. Che, H. Guo, B. Qin, Z. Su, and T. Liu, “Sentence compression for target-polarity word collocation extraction,” in Proc. COLING, 2014, pp. 1360–1369.
[20] Q. Liu, Z. Gao, B. Liu, and Y. Zhang, “A logic programming approach to aspect extraction in opinion mining,” in Proc. WI-IAT, 2013, pp. 276–283.
[21] Q. Liu, Z. Gao, B. Liu, and Y. Zhang, “Automated rule selection for opinion target extraction,” Knowl.-Based Syst., vol. 104, pp. 74–88, Jul. 2016.
[22] C. Wu, F. Wu, S. Wu, Z. Yuan, and Y. Huang, “A hybrid unsupervised method for aspect term and opinion target extraction,” Knowl.-Based Syst., vol. 148, pp. 66–73, May 2018.
[23] Y. Yin, F. Wei, L. Dong, K. Xu, M. Zhang, and M. Zhou, “Unsupervised word and dependency path embeddings for aspect term extraction,” in Proc. JCAI, 2016, pp. 2979–2985.
[24] H. Ye, Z. Yan, Z. Luo, and W. Chao, “Dependency-tree based convolutional neural networks for aspect term extraction,” in Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining, 2017, pp. 350–362.
[25] W. Wang, S. J. Pan, and D. Dahlmeier, “Memory networks for fine-grained opinion mining,” Artif. Intell., vol. 265, pp. 1–17, Dec. 2018.
[26] H. Xu, B. Liu, L. Shu, and S. Y. Philip, “Double embeddings and CNN-based sequence labeling for aspect extraction,” in Proc. 56th Annul. Meet. Assoc. Comput. Linguistics, 2018, pp. 592–598.
[27] X. Li, L. Bing, P. Li, and W. Lam, “A unified model for opinion target extraction and target sentiment prediction,” in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 6714–6721.
[28] S. Khaled-E-Fatima, A. Javed, A. Banjar, A. Itaza, H. Dawood, H. Dawood, and A. Alamri, “A multi-layer dual attention deep learning model with refined word embeddings for aspect-based sentiment analysis,” IEEE Access, vol. 7, pp. 114795–114807, 2019.
[29] H. Luo, T. Li, B. Liu, B. Wang, and H. Unger, “Improving aspect term extraction with bidirectional dependency tree representation,” IEEE/ACM Trans. Audio, Speech, Lang. Process., vol. 27, no. 7, pp. 1201–1212, Jul. 2019.
[30] X. Pan, T. Cassidy, U. Hermjakob, H. Ji, and K. Knight, “Unsupervised entity linking with abstract meaning representation,” in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Hum. Lang. Technol., 2015, pp. 1130–1139.
[31] X. Li, T. H. Nguyen, K. Cao, and R. Grishman, “Improving event detection with abstract meaning representation,” in Proc. 1st Workshop Comput. Linguistics News Storylines, 2016, pp. 1–15.
[32] S. Rao, D. Marcu, K. Knight, and H. Daumé, III, “Biomedical event extraction using abstract meaning representation,” in Proc. BioNLP, 2017, pp. 126–135.
[33] S. Dohare, H. Karnick, and V. Gupta, “Text summarization using abstract meaning representation,” 2017, arXiv:1706.01678. [Online]. Available: http://arxiv.org/abs/1706.01678
[34] S. Dohare, V. Gupta, and H. Karnick, “Unsupervised semantic abstractive summarization,” in Proc. 3rd Workshop Comput. Linguistics Student Res. Workshop, 2018, pp. 74–83.
[35] H. Hardy and A. Vlachos, “Guided neural language generation for abstractive summarization using abstract meaning representation,” in Proc. Conf. Empirical Methods Natural Lang. Process., 2018, pp. 768–773.
[36] F. Liu, J. Flanigan, S. Thomson, N. Sadeh, and N. A. Smith, “Toward abstractive summarization using semantic representations,” 2018, arXiv:1805.10399. [Online]. Available: http://arxiv.org/abs/1805.10399
[37] L. Gao, D. Gildner, Y. Zhang, Z. Wang, and J. Su, “Semantic neural machine translation using AMR,” Trans. Assoc. Comput. Linguistics, vol. 7, pp. 19–31, Mar. 2019.
[38] J. F. Allen, O. Bahkshee, W. de Beumont, L. Galescu, and C. M. Teng, “Effective broad-coverage deep parsing,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 4776–4783.
[39] P. Bonatti, F. Calimeri, N. Leone, and F. Ricca, “Answer set programming,” in A 25-Year Perspective on Logic Programming, Berlin, Germany: Springer-Verlag, 2010, pp. 159–182.
XU'ELIAN LI received the B.S. degree in information and computing science and the M.A. degree in applied linguistics from Nanjing Tech University, Nanjing, China, in 2010 and 2013, respectively. She is currently pursuing the Ph.D. degree in computer science and technology with Southeast University, Nanjing. Her search interests include opinion mining, information extraction, natural language processing, artificial intelligence, and semantics.

BI WANG is currently pursuing the Ph.D. degree in computer science and technology with Southeast University, Nanjing, China. His current research interests include reinforcement learning, natural language processing, and artificial intelligence.

LIXIN LI received the B.S. degree from Soochow University. He is currently pursuing the M.S. degree in computer science and technology with Southwest University. His current research interests include sentiment analysis, natural language processing, and artificial intelligence.

ZHIQIANG GAO was born in Hebei, in 1966. He received the B.A. and M.D. degrees from Southeast University, and the Ph.D. degree from the Department of Mechanical Engineering, Tsinghua University, in 1995. After graduation, he was worked for National CAD Application and Training Network–Nanjing Center, as well as the Department of Mechanical Engineering, Southeast University. As a Postdoctoral Researcher, he went to Japan to do researches on digital city and artificial intelligence with the Department of Social Informatics, Kyoto University, in 2000. He has been an Associate Professor with the Department of Computer Science and Engineering, Southeast University, since October 2002. In December 2007, he joined the Institute of Web Science (IWS) to study Ontology Learning and Natural Language Processing (NLP), as well as large scale reasoning. He is interested in almost everything, such as material science and manufacturing, numerical simulation of heat transfer and fluid flow, CAD, CG, visualization, VR, game theory, agent oriented programming, semantic Web, knowledge capture, web mining, military simulation, as well as philosophy, traveling, basketball, and Taijiquan. He was concentrated in human-like and conversational agent, multi-agent systems, agent oriented programming and machine learning. In 2003, a project was launched about interactive multi-agent system used for shooter training simulation, in which agents are hosted by Augmented Reality. He has to construct 3D buildings and terrains, design and debug scenarios, and recognize behavior of shooters in the real world. Although, this project is more complicated than designing ordinary CG games, it is much more interesting and challenging. He is also involved in 973 projects for semantic grid. Recently, he is responsible for two NSFC (National Science Foundation of China) projects, which are related to ontology learning and inductive logic programming. Speech recognition, text understanding, and image recognition are faced with the same puzzles, and natural language processing is one of the best test bed for various innovative AI ideas. Machine Learning may play a key role in these fields in the future. Additionally, semantic annotation of 3D environments for multi-agents to plan and act rationally is an important branch of Semantic Web. He firmly believes that a new era for search engine based on deep NLP, Machine Learning, and large scale reasoning is coming.

QIAN LIU was born in 1986. She received the Ph.D. degree from Southeast University. She entered the Nanjing University of Posts and Telecommunications, in 2016. Her research focuses on artificial intelligence and natural language processing, with applications in opinion mining and relation extraction. Her ongoing research focuses on complex non-monotonic information extraction tasks.

HANCHENG XU received the M.A. degree with Xi’an International Studies University and the Ph.D. degree with Shanghai International Studies University. He completed his Postdoctoral ship with Heilongjiang University. He is currently a Professor with the International Studies College, National University of Defense Technology. His research interests are computational linguistics and traditional Russian language studies and teaching.

LANTING FANG received the Ph.D. degree from Southeast University, Nanjing, China, in 2018. She is currently a Lecturer with the School of Cyber Science and Engineering, Southeast University. Her current research interests include NLP, information extraction, social media, and sentiment analysis.