Improving Aspect Extraction based on Rules through Deep Syntax-Semantics Communication

Anonymous ACL submission

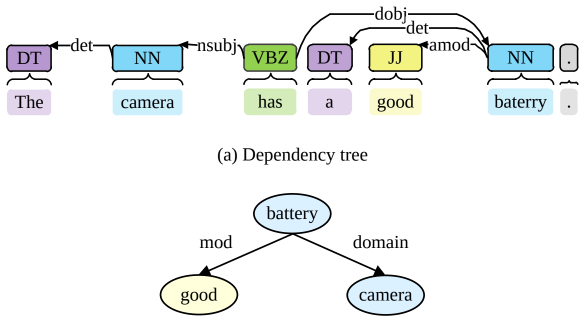
Abstract

Recent studies show integrating language resources which consist of lexical resources, syntactic resources and semantic resources can improve the performance of natural language processing (NLP) tasks. The existing methods mostly perform simple integration through concatenating these resources successively, seldom consider complementary relationship among them, such as the deep communication of syntactic and semantic relations between words. To enhance deep syntax-semantics communication, this paper takes aspect term extraction (ATE) task as an example and explores four integration strategies of language resources. These strategies, based on Answer Set Programming (ASP) rules, have interpretability. Experiments on eight ATE datasets show that our strategies achieve superior performance, demonstrating that they are highly effective in integrating language resources.

1 Introduction

Language resources have been widely used for various NLP tasks (Wei et al., 2020; Zhang and Qian, 2020; Junqi et al., 2021). These resources consist of lexical resources, syntactic resources, and semantic resources. Lexical resources include word form, part of speech (POS), sequence of word forms, sequence of POS, word embedding, language model, etc. Syntactic resources include dependency grammar (DEP), phrase structure grammar, etc. Semantic resources include predicate argument structure, semantic role, abstract meaning representation (AMR), sentence embedding, etc.

Many studies have shown that integrating language resources can bring improved performance for NLP tasks, such as machine translation (Song et al., 2019; Wei et al., 2020), information extraction (Rastegar-Mojarad et al., 2017; Wang et al., 2017b), aspect extraction (Yang and Huang, 2016; Veyseh et al., 2020). The early approaches have involved the rule-based (Qiu et al., 2011; Jiang et al., 2017) and conditional random field (CRF)-based methods (Yang and Huang, 2016) while the recent work has focused on neural network models (Wang et al., 2017b; Vashishth et al., 2019).



(a) Dependency tree

(b) AMR graph

Figure 1: Dependency tree and AMR graph of Example 1 "The camera has a good battery."

Though achieving impressive progress, these methods mostly integrate language resources through a straightforward way such as simple concatenation where lexical, syntactic and semantic information of a sentence are concatenated successively. But this way neglects complementary relationship among resources such as the internal integration of syntactic and semantic relations between words in a sentence. As we all know, language is a combination of form and meaning, complementary relationship among lexical, syntax and semantics is very important for NLP tasks. Take dependency tree and AMR graph of **Example 1** "The camera has a good battery." as an instance. Figure 1 (a) shows its POS and syntactic structure of dependency tree where the noun "battery" and the noun "camera" have dependency relation "nsubj-dobj", while (b) presents its semantic structure of AMR graph where "battery" and "camera" have the semantic relation "domain". If we want to extract the aspect "battery", it is more accurate to extract it when using the deep communication of POS, syntax and semantics.

To enhance deep syntax-semantics communication, this paper proposes four integration strategies which are Semantic Rule Concatenation based strategy (**Sem-C**), Syntactic Rule Concatenation based strategy (**SYN-C**), Semantic Graph Structure based strategy (**SEM-G**) and Syntax Tree Structure based strategy (**SYN-G**). SEM-C and SYN-C based on ASP rule concatenation can take full advantage of the scalability of rules while SEM-G and SYN-G based on language structure can maintain the structure of tree and graph. Based on the non-monotonicity and scalability of Answer Set Programming (ASP) (Gelfond, 1988), this paper employs ASP which offers detail-giving, natural-language explanations for its answers to perform the four strategies. In our paper, the lexical resource adopted is POS, the syntactic resource adopted is DEP, and the semantic resource adopted is AMR.

To demonstrate the effectiveness of our proposed strategies, we take aspect term extraction (ATE) task as an example and conduct experiments on eight publicly aspect-annotated datasets. Experimental results show that our strategies outperform the baselines by a large margin, offering an alternative for the integration of language resources. Further analysis indicates that our strategies have interpretability and are highly effective in performing deep syntax-semantics communication, verifying the importance of complementary relationship between syntax and semantics for NLP tasks. We will publish all source codes and datasets of this work on Github for further research explorations https://github.com/njirene/SynSem.

2 Related Work

Recent studies have shown that integrating language resources can bring improved performance for NLP tasks (Marcheggiani et al., 2018; Song et al., 2019; Rastegar-Mojarad et al., 2017; Veyseh et al., 2020). With regard to the integration of language resources, there has been considerable work combining lexical resources only (Hu and Liu, 2004; Ma et al., 2019), a range of work integrating lexical and syntactic resources (Qiu et al., 2011; Liu et al., 2015), some work integrating lexical and semantic resources (Li et al., 2012; Li and Chang, 2019; Dohare et al., 2017; Hardy and Vlachos, 2018), and fewer combining all of them (Yang and Huang, 2016; Wang et al., 2017b).

In these researches of integrating lexical, syntactic and semantic resources, they mostly use neural network and CRF which have been utilized as some NLP tasks, such as information extraction (Wang et al., 2017b; Rastegar-Mojarad et al., 2017), machine translation (Marcheggiani et al., 2018) and aspect extraction (Yang and Huang, 2016). For example, Wang et al. (2017b) propose to concatenate word embeddings, dependency embeddings, and AMR embeddings as features and use SVM and random forest for drug-drug interaction. Marcheggiani et al. (2018) propose to exploit semantics in neural machine translation with graph convolutional networks (GCN) where syntax and semantic are combined together in the same GCN layer. Yang and Huang (2016) propose a hybrid approach which incorporates domain lexicon with syntactic and semantic features to perform aspect extraction. The approach acquires domain lexicon using CRF and simply combines it with POS, dependency structure, semantic role based on word embedding.

However, these studies mostly perform shallow integration through concatenating language resources successively, and neglect the deep integration and complementary relationship among them. Moreover, these methods are not interpretable when integrating syntactic and semantic resources, and require a great deal of annotated data to train models.

This paper focuses on ASP rule-based approach. Different from the existing approaches, our approach can achieve complementary relationship between semantic and syntactic resources. The scalability of ASP makes our integration strategies flexible and general, i.e., it can be used to combine almost all language resources.

3 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 (Gelfond, 1988; Bonatti et al., 2010), 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, ..., l_m, not l_{m+1}, ..., not l_n.

where each l_i for $i \in [0...n]$ is a **literal** of some signature, i.e., expressions of the form $p(t)$ where

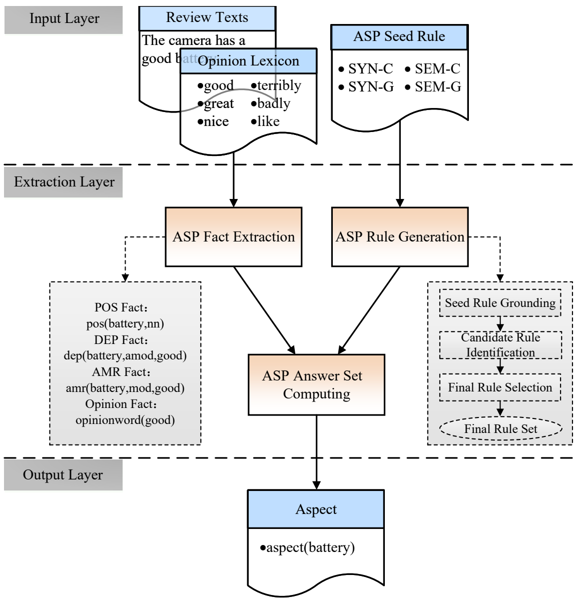


Figure 2: ASP based framework for aspect extraction.

p is a predicate and t is a term, and not is called $negation\ as\ failure$ or $default\ negation$. For instance, amr (A, mod, O) is an atom with the predicate amr, and three terms, one constant mod and two variables A and O. A rule without body is called a $fact$.

In this paper, ASP based framework for extracting aspects is proposed, which consists of input layer, extraction layer and output layer, as shown in Figure 2. The input layer includes review text, opinion word lexicon, and ASP seed rule. The extraction layer consists of three modules: ASP fact extraction, ASP rule generation and ASP answer set computing. The output layer outputs the results of ASP answer set computing module. The three modules in extraction layer are the most important steps, we introduce them as follows.

3.1 ASP Fact Extraction

To extract ASP facts from syntax tree and AMR graph automatically, we develop an algorithm which is given in Algorithm 1. First, for every sentence $s_i \in \mathcal{S}$, use Stanford Parser[1] to parse POS of words and dependency relations between words, represent them as ASP facts and insert them into $Fact_{pos}$, $Fact_{dep}$, respectively (lines 1-3). Second, use semantic parser JAMR[2] to obtain semantic relations between words in s_i, represent them as ASP facts and insert them into $Fact_{amr}$(line 4). Third, obtain the opinion word facts $Fact_{opn}$(lines 5-9).

[1]http://nlp.stanford.edu:8080/parser/
[2]https://github.com/jflanigan/jamr

Algorithm 1 FactExtraction($\mathcal{S}$, $\mathcal{O}$)

Require: Review text data $\mathcal{S} = \{s_1, ..., s_n\}$, opinion word lexicon $\mathcal{O}$

Ensure: ASP fact knowledge base $Fact$, including POS facts $Fact_{pos}$, syntactic facts $Fact_{dep}$, semantic facts $Fact_{amr}$ and opinion facts $Fact_{opn}$

1: $Fact = \{\}$
2: **for** every sentence $s_i \in \mathcal{S}$ **do**
3: $Fact_{pos}$, $Fact_{dep} \leftarrow$ SyntacticParsing(S)
4: $Fact_{amr} \leftarrow$ SemanticParsing(S)
5: **for** every word $w \in S$ **do**
6: **if** match(w, $\mathcal{O}$) **then**
7: represent w into opinion facts and insert them into $Fact_{opn}$
8: **end if**
9: **end for**
10: $Fact \leftarrow Fact_{pos} \cup Fact_{dep} \cup Fact_{amr} \cup Fact_{opn}$
11: **end for**

Finally, a set of ASP fact knowledge base $Fact$ will be extracted by repeatedly implementing Algorithm 1, which can be used for SEM-C and SYN-C (line 10). Figure 2 shows ASP facts of Example 1. For SEM-G and SYN-G, their facts can be obtained by integrating POS, syntactic, and semantic facts based on language structure, whose algorithm can be found in Appendix.

3.2 ASP Rule Generation

In the process of ASP rule generation, we first construct a few ASP seed rules of four strategies manually. Then, an algorithm (Algorithm 2) is proposed to refine the seed rules and generate the final rules, including the following three steps: generate a lot of instances based on ASP seed rules (lines 1-4); identify the candidate rule set from the grounding rule set (lines 5-10); select the high quality rule subset by greedy algorithm (lines 11-18). Importantly, line 11 ranks rules by precision first because high precision rules are more desirable. The recall can be improved by using more rules. Then we select rules by F_1 because we want the final rule set to produce overall good extraction result. If a rule extracts the same aspect with other rules, F_1 is unchanged, this step is to prevent the elimination of useful rules. Table 1 shows the examples of the final rules in which the number of amr, dep, and pos literals is up to three.

Algorithm 2 RuleGeneration($Fact$, $\mathcal{R}_{seed}$, $\mathcal{L}$)

Require: ASP facts $Fact$, ASP seed rules $\mathcal{R}_{seed}$, labeled aspects $\mathcal{L}$

Ensure: Final ASP rule set $\mathcal{R}$

```
R = {}, ER = {};
for Literals in R_seed do
    replace relations REL in Literal with relations d_ij in Fact using 4 strategies and obtain ER;
end for
for each rule r_i ∈ ER do
    solve {r_i} ∪ Fact;
    if {aspect(A) | {r_i} ∪ Fact |= aspect(A) } ∩ L = ∅ then
        delete r_i from ER;
    end if
end for
Rank r_j ∈ ER by presicion;
for each rule r_j ∈ ER in descending order do
    solve {r_j} ∪ R ∪ Fact, compute F_1 in R
    if F_1 increased or unchanged then
        insert r_j into R;
    end if
end for
Output R as the final ASP rule set.
```

3.3 ASP Answer Set Computing

An ASP program consists of ASP facts and ASP rules. Compute the answer set of the logic program using an ASP solver like clingo[3]. Then the aspect terms are extracted from the answer set.

4 Four Strategies for Enhancing Deep Syntax-Semantics Communication

Next we give a more detailed description of our strategies. Table 1 shows the integration methods in four strategies and the corresponding rules. In SEM-C and SYN-C, two or more language resources in a sentence are concatenated into the rule body by expanding; while in SEM-G and SYN-G, two or more resources in a sentence are incorporated into one resource for use.

4.1 SEM-C

SEM-C is a strategy of incorporating syntactic information into semantic rules through ASP rule concatenation, namely, based on semantic rules, DEP and POS in a sentence are concatenated into the rule body by using the literals of ASP rules. It includes three integration methods: **AMR-POS-C** (incorporating POS literal into AMR rule), **AMR-DEP-C** (incorporating DEP literal into AMR rule), **AMR-DEP-POS-C** (incorporating DEP and POS literals into AMR rule), as shown in Table 1. Taking AMR-DEP-POS-C for example,

aspect(A) :- *amr(A,mod,O)*, *dep(A,amod,O)*,
pos(A,nn), *opinionword(O)*,
not generalWord(A).

where *amr(A,mod,O)* means that aspect A and opinion word O in AMR graph have semantic relation *mod*. *dep(A,amod,O)* means that in syntax tree, nodes A and O have the dependency relation *amod*. *pos(A,nn)* represents the POS of aspect A is a noun. *opinionword(O)* means O is an opinion word. *not generalWord(A)* means not is used to exclude the general words such as "person", "thing" and so on, which are from the lexicon constructed by (Liu et al., 2013). SEM-C incorporates POS, syntactic and semantic information into one rule. From Example 1, we can extract aspect "battery" since the relations between "good" and "battery" satisfy the constrains of this rule.

4.2 SYN-C

SYN-C is a strategy of incorporating semantic information into syntactic rules through ASP rule concatenation, namely, based on syntactic rules, AMR and POS in a sentence are concatenated into the rule body by using the literals of ASP rules. It includes three integration methods: **DEP-POS-C** (incorporating POS into syntactic rule), **DEP-AMR-C** (incorporating AMR into syntactic rule), **DEP-AMR-POS-C** (incorporating AMR and POS into syntactic rule). Although the compositional ways of DEP-AMR-C, DEP-AMR-POS-C in SYN-C and AMR-DEP-C, AMR-DEP-POS-C in SEM-C are different, their integration rules are the same, as the literals of each rule are unordered, as shown in Table 1.

4.3 SEM-G

SEM-G is a strategy of incorporating syntax tree into semantic graph based on language structure. Based on AMR graph, it incorporates grammatical relation labels between the nodes in syntax tree and the corresponding POS of the nodes into semantic relation labels to refine the edge labels in AMR graph. And SEM-G includes three integration methods: **AMR-POS-G** (incorporating POS into AMR graph), **AMR-DEP-G** (incorporating

[3] https://potassco.org/

Table 1: Integration methods in four strategies and the corresponding rule instances of Example 1 "The camera has a good battery."

Strategies	Methods	Final Rule Instance
SEM-C	AMR-POS-C	*aspect(A) :- amr(A,mod,O), pos(A,nn), opinionword(O), not generalWord(A).*
	AMR-DEP-C	*aspect(A) :- amr(A,mod,O), dep(A,amod,O), opinionword(O), not generalWord(A).*
	AMR-DEP-POS-C	*aspect(A) :- amr(A,mod,O), dep(A,amod,O), pos(A,nn), opinionword(O), not generalWord(A).*
SYN-C	DEP-POS-C	*aspect(A) :- dep(A,amod,O), pos(A,nn), opinionword(O), not generalWord(A).*
	DEP-AMR-C	*aspect(A) :- dep(A,amod,O), amr(A,mod,O), opinionword(O), not generalWord(A).*
	DEP-AMR-POS-C	*aspect(A) :- dep(A,amod,O), amr(A,mod,O), pos(A,nn), opinionword(O), not generalWord(A).*
SEM-G	AMR-POS-G	*aspect(A) :- amr-pos(A,mod-nn-jj,O), opinionword(O), not generalWord(A).*
	AMR-DEP-G	*aspect(A) :- amr-dep(A,mod-amod,O), opinionword(O), not generalWord(A).*
	AMR-DEP-POS-G	*aspect(A) :- amr-dep-pos(A,mod-amod-nn-jj,O), opinionword(O), not generalWord(A).*
SYN-G	DEP-POS-G	*aspect(A) :- dep-pos(A,amod-nn-jj,O), opinionword(O), not generalWord(A).*
	DEP-AMR-G	*aspect(A) :- dep-amr(A,amod-mod,O), opinionword(O), not generalWord(A).*
	DEP-AMR-POS-G	*aspect(A) :- dep-amr-pos(A,amod-mod-nn-jj,O), opinionword(O), not generalWord(A).*

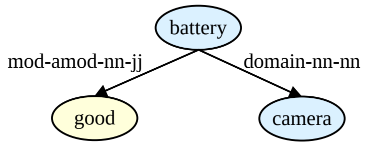


Figure 3: The integration way of AMR-DEP-POS-G

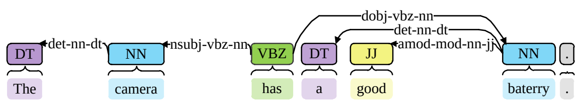


Figure 4: The integration way of DEP-AMR-POS-G

syntax tree into AMR graph), **AMR-DEP-POS-G** (incorporating syntax tree and POS into AMR graph). Taking AMR-DEP-POS-G as an example,

aspect(A) :- *amr-dep-pos(A,mod-amod-nn-jj,O),*
opinionword (O),
not generalWord(A).

where *amr-dep-pos(A,mod-amod-nn-jj,O)* means that in AMR graph, the edge which nodes A and O connect has a concrete relation *mod-amod-nn-jj*. *mod-amod-nn-jj* represents the semantic relation *mod*, grammatical relation *amod* between nodes A and O, and their POS are combined into the same edge, as shown in Figure 3.

4.4 SYN-G

SYN-C is a strategy of incorporating semantic graph into syntax tree based on language structure. It incorporates semantic relation labels between the nodes in AMR graph and the corresponding POS of the nodes into grammatical relation labels to refine the edge labels in syntax tree. The intuition of this strategy is to guarantee every word in a sentence occur in syntax tree. And SYN-G includes three integration methods: **DEP-POS-G** (incorporating POS into syntax tree), **DEP-AMR-G** (incorporating AMR graph into syntax tree), **DEP-AMR-POS-G** (incorporating AMR graph and POS into syntax tree). Taking DEP-AMR-POS-G as an example,

aspect(A) :- *dep-amr-pos(A,amod-mod-nn-jj,O),*
opinionword (O),
not generalWord(A).

where *dep-amr-pos(A,amod-mod-nn-jj,O)* means that in syntax tree, the edge which nodes A and O connect has a concrete relation *amod-mod-nn-jj*. *amod-mod-nn-jj* represents the grammatical relation *amod*, semantic relation *mod* between nodes A and O, and their POS are combined into the same edge of syntax tree, as shown in Figure 4.

5 Experiments

5.1 Datasets

Three publicly aspect-annotated corpora are adopted. The first which is from (Hu and Liu, 2004) contains five review datasets from four domains: digital cameras (D1, D2), cell phone (D3), MP3 player (D4), and DVD player (D5). The second which is from SemEval-2014 and SemEval-2015contains three review datasets from two domains: SemEval-2014 Restaurants (D6), SemEval-2014 Laptops (D7), and SemEval-2015 Restaurants (D8). The third which is from (Liu et al., 2015) contains three datasets: computer (D9), wireless router (D10), and speaker (D11), which are used as the development datasets to construct the seed rules manually. The seed opinion words are offered by (Hu and Liu, 2004) and the original annotated datasets, respectively. Table 2 shows the detailed information of each dataset.

For the first corpus, due to the small size of each dataset, we have followed the same way of cross domain test in (Liu et al., 2015). Namely, for D1 to D5, leave-one-out cross validation is utilized. For the second corpus, the train-test split is the same as the original dataset, as shown in Table 2.

Table 2: Detailed information of the datasets.

Data	Product	# Sentences	# 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
	Restaurant-14(Test)	800	1134
D7	laptop-14(Train)	3048	2373
	laptop-14(Test)	800	654
D8	Restaurant-15(Train)	1315	1279
	Restaurant-15(Test)	685	597
D9	Computer(Dev)	531	354
D10	Router(Dev)	879	307
D11	Speaker(Dev)	689	440

5.2 Evaluation Metrics

For the comparison, we choose the same two ways used by (Liu et al., 2016) to compute the extracting results: 1) mul, based on multiple occurrences of each aspect term, and 2) dis, based on distinct occurrence of each aspect term. And F_1-score are adopted to evaluate the performance.

5.3 Comparative Methods

To validate the performance of our proposed four strategies (e.g., SEM-C, SYN-C, SEM-G and SYN-G) on aspect extraction, we compare them against three kinds of baselines. The first is the rule based baselines, including syntactic rule based methods such as **DP** (Qiu et al., 2011), **RSG$^+$** (Liu et al., 2016) and **DEP** (without POS proposed in this paper), and semantic rule based method **AMR** (without POS proposed in this paper). The second is the combination method of neural network and linguistic rules, such as **CNN+LP** (Poria et al., 2016) and **RINANTE** (Dai and Song, 2019) which is based on the dependency relations and POS of words. The third is the recent deep learning-based methods, such as **CMLA** (Wang et al., 2017a), **GMTCMLA** (Yu et al., 2019) and **SpanMlt** (Zhao et al., 2020) which uses BiLSTM encoder and BERT encoder.

5.4 Experimental Results

The comparison results for all methods are shown in Table 3. We observe the average F_1 of our proposed DEP-AMR-POS-G method on mul and dis evaluation metrics are higher than all baselines in both five datasets and three datasets. Specifically, 1) compared with the syntactical rule based baselines, the total average F_1 of DEP-AMR-POS-G on mul and dis, improve by 11.2% and 17.8% (DP), 1.6% and 3.6% (RSG$^+$), 1.6% and 3.6% (DEP), respectively. This means that our method adding semantic rules is markedly better than the baselines which only use syntactic rules. 2) Compared with the semantic rule based method, DEP-AMR-POS-G are much better than AMR without considering POS and syntax, showing that POS and syntax are beneficial to identify aspect terms. 3) Compared with the combination methods of neural network and linguistic patterns such as CNN+LP and RINANTE, as well as the deep learning-based methods such as CMLA, GMTCMLA and SpanMlt, DEP-AMR-POS-G still performs well. We believe one of the key reasons is that our method can accomplish complementary relationship among resources and capture more information which baselines do not.

From Table 3, we can see that in both five and three dataset, the average F_1-scores of DEP-AMR-POS-G on mul are up to 89.2% and 87.9%, respectively, which are higher than the best methods DEP-AMR-POS-C in SYN-C and AMR-DEP-POS-G in SEM-G. While DEP-AMR-POS-G in D1-D5 drops only a little F_1-score in dis below DEP-AMR-POS-C. But they both perform better than AMR-DEP-POS-G. This indicates DEP-AMR-POS-G and DEP-AMR-POS-C, which do not omit linguistic information by integrating three language resources based on syntax, has better extraction performance. The results of AMR-DEP-POS-G incorporating syntactic structures and POS into semantic graph are not promising as we expected. One reason might be due to that the semantic structures discard some surface linguistic knowledge which are important for extracting aspects.

We further see that in each strategy, the performance of the methods combining three language resources is better than the methods combining only two resources. This indicates that the added semantic or syntactic resources can complement the original resources, which can improve the performance of aspect extraction.

Moreover, DEP-POS-C in SYN-C and DEP-POS-G in SYN-G are better than AMR-POS-C in SEM-C and AMR-POS-G in SEM-G. This indicates semantic resource is useful although its performance is not so good as syntactic resource. One reason might be due to that semantic relation "mod"

Table 3: F_1-score comparison of the baselines and our methods.

	Methods	Metrics	D1	D2	D3	D4	D5	avg	D6	D7	D8	avg
Baseline	DP	mul	80.0	80.8	82.8	78.1	74.0	79.1	85.2	73.0	83.6	80.6
		dis	70.0	67.9	67.9	62.6	62.4	66.1	73.7	68.5	75.6	72.6
	RSG+	mul	87.8	90.2	86.3	85.3	87.7	87.4	86.7	75.9	86.2	82.9
		dis	78.0	83.2	74.7	73.0	72.7	76.3	76.4	70.2	79.4	75.3
	DEP	mul	81.2	89.8	78.7	79.6	77.4	81.3	83.3	71.7	76.2	77.1
		dis	72.3	80.8	67.5	66.0	61.2	69.6	68.7	66.5	65.1	66.8
	AMR	mul	66.2	79.9	76.3	76.6	73.7	74.5	76.5	62.4	77.9	72.3
		dis	57.0	66.3	65.0	63.0	55.7	61.4	62.4	57.5	63.0	61.0
	CNN+LP	mul	88.0	84.0	87.0	89.0	90.0	87.6	-	-	-	-
		dis	-	-	-	-	-	-	-	-	-	-
	RINANTE	mul	-	-	-	-	-	-	-	-	-	-
		dis	-	-	-	-	-	-	86.5	80.2	69.9	78.9
	CMLA	mul	-	-	-	-	-	-	-	-	-	-
		dis	-	-	-	-	-	-	85.2	77.8	70.7	77.8
	GMTCMLA	mul	-	-	-	-	-	-	-	-	-	-
		dis	-	-	-	-	-	-	84.5	78.7	70.5	77.9
	SpanMlt	mul	-	-	-	-	-	-	-	-	-	-
		dis	-	-	-	-	-	-	85.2	77.9	71.1	78.1
SEM-C	AMR-POS-C	mul	84.7	89.2	84.0	82.3	85.6	85.2	84.4	68.7	80.2	77.8
		dis	79.0	81.1	74.5	73.7	78.2	77.3	72.3	65.1	68.2	68.5
	AMR-DEP-C	mul	85.1	90.5	89.0	80.2	76.6	84.3	84.8	77.2	82.0	81.3
		dis	76.9	80.8	77.9	66.2	64.0	73.1	72.4	70.4	70.0	70.9
	AMR-DEP-POS-C	mul	88.4	92.6	86.5	85.3	88.8	88.3	88.9	79.1	89.0	85.7
		dis	81.4	86.9	77.3	74.9	81.8	80.5	79.4	72.8	80.8	77.7
SYN-C	DEP-POS-C	mul	88.1	93.3	84.2	84.2	87.2	87.4	86.4	77.9	86.4	83.6
		dis	79.7	86.6	74.0	71.8	75.3	77.5	75.7	72.5	79.9	76.0
	DEP-AMR-C	mul	85.1	90.5	89.0	80.2	76.6	84.3	84.8	77.2	82.0	81.3
		dis	76.9	80.8	77.9	66.2	64.0	73.1	72.4	70.4	70.0	70.9
	DEP-AMR-POS-C	mul	88.4	92.6	86.5	85.3	88.8	88.3	88.9	79.1	89.0	85.7
		dis	81.4	86.9	77.3	74.9	81.8	**80.5**	79.4	72.8	80.8	77.7
SEM-G	AMR-POS-G	mul	80.8	85.8	82.4	81.4	84.5	83.0	84.5	68.7	82.1	78.4
		dis	75.2	76.1	74.1	71.9	72.5	73.9	72.0	65.6	68.4	68.7
	AMR-DEP-G	mul	76.4	82.2	78.1	76.7	67.5	76.2	80.5	72.1	77.1	76.6
		dis	68.4	66.7	68.2	63.4	55.1	64.4	66.5	66.7	65.3	66.2
	AMR-DEP-POS-G	mul	86.2	84.3	83.4	81.7	80.3	83.2	85.9	73.7	87.9	82.5
		dis	79.9	74.5	76.0	72.0	69.8	74.4	74.5	69.1	77.2	73.6
SYN-G	DEP-POS-G	mul	82.1	83.7	82.1	86.2	82.9	83.4	85.7	69.4	86.2	80.4
		dis	71.6	77.5	71.5	79.2	74.5	74.9	73.8	64.0	77.3	71.7
	DEP-AMR-G	mul	68.0	78.7	81.7	87.4	76.2	78.4	78.9	80.5	81.7	80.4
		dis	51.0	68.9	67.1	75.6	64.2	65.4	63.9	74.2	69.9	69.3
	DEP-AMR-POS-G	mul	88.5	91.6	89.8	90.6	85.2	**89.2**	90.6	82.8	90.4	**87.9**
		dis	82.6	80.8	77.8	77.7	75.7	78.9	83.7	80.5	83.4	**82.1**

in AMR, which covers more than a dozen relations in syntax trees such as "amod", "nmod", is very coarse-grained. But fine-grained dependency relations (e.g., "amod") between opinion words and aspects are important. Another reason might be due to the parser performance limitation of semantic structure. For example, JAMR whose average parsing performance is less than 70% (Flanigan et al., 2014). From Table 2, we can see our methods are suitable for small samples. For example, in D2 dataset, the number of its sentences is 346, our methods can obtain good performance, this indicates our methods don't require more data.

5.5 Efficiency Analysis

Since our proposed strategies use the same ASP solver and algorithm on the same datasets, we take D6 dataset as an example to compare their final rule number and running time. As shown in Table 4, DEP-AMR-POS-G which has better performance and higher precision, requires more rules and running time while AMR-DEP-POS-C whose performance is the second but recall is higher, requires less rules and running time. Therefore, we find that different integration strategies have their own characteristics and different application scenarios.

Table 4: Rule number and running time of the best methods in four strategies.

Methods	Num	P	R	F_1	Time
AMR-DEP-POS-C	990	81.7	77.9	79.4	259m
DEP-AMR-POS-C	990	81.7	77.9	79.4	259m
AMR-DEP-POS-G	1525	89.0	64.1	74.5	239m
DEP-AMR-POS-G	**2022**	91.6	77.1	**83.7**	491m

5.6 Error Analysis

We analyze the error types of aspects which are labeled but not extracted by taking D8 dataset as an example. Three types are summarized: the first is parsing error; the second is that there is no direct or indirect syntactic or semantic relations between aspects and opinion words; the third is the rule with too much or too less constraints.

Table 5: Distribution of the extracted results of three methods, "+" indicates the aspect extracted is correct, and "-" indicates that the aspect extracted is wrong.

	1	2	3	4	5	6	7	8
AMR-DEP-POS-C	–	–	–	–	+	+	+	+
AMR-DEP-POS-G	–	–	+	+	–	–	+	+
DEP-AMR-POS-G	–	+	–	+	–	+	–	+
Num of Aspects	98	25	0	27	20	48	3	302

As shown in Table 5, the extracting results of three methods are divided into eight cases. Case 2 represents the number of aspects which AMR-DEP-POS-C and AMR-DEP-POS-G don't extract but DEP-AMR-POS-G extracts is 25. The number of different aspects labeled in D8 dataset is 523, AMR-DEP-POS-C extracts 373 aspects, AMR-DEP-POS-G extracts 332 and DEP-AMR-POS-G extracts 402. 121 aspects labeled are not extracted from DEP-AMR-POS-G, which reduces 4 aspects in the first error, 17 aspects in the second error and 8 aspects in the third error compared with AMR-DEP-POS-C. This shows that DEP-AMR-POS-G can capture the deep interaction of language structure and further improve the performance of NLP tasks.

5.7 Case Study

To better understand in which conditions our best method DEP-AMR-POS-G helps, we examine the instances that cannot extracted by AMR-DEP-POS-C and AMR-DEP-POS-G, but correctly extracted by DEP-AMR-POS-G. Moreover, an interpretable model should be able to pinpoint exactly why a particular prediction was made, and provide the reason in a clear and natural way (Letham et al., 2012). To better understand the interpretability, we present the extracting process of our three best methods using the following sentence.

The veggie burger made from a nice blend of chickpeas and carrots.

Take extracting *chickpeas* as an example, the most related ASP facts are listed:

f_1 *pos(chickpeas, nn).*
f_2 *pos(carrots, nn).*
f_3 *pos(nice, jj).*
f_4 *dep(chickpeas, conj, carrots).*
f_5 *dep(blend, amod, nice).*
f_6 *amr(and, op, chickpeas).*
f_7 *amr(and, op, carrots).*
f_8 *amr-dep-pos(and, op-cc-nn, chickpeas).*
f_9 *amr-dep-pos(and, op-cc-nn, carrots).*
f_{10} *dep-amr-pos(blend, amod-mod-vb-jj, nice).*
f_{11} *dep-amr-pos(blend, prep-vb-nn, chickpeas).*
f_{12} *opinionword(nice).*

R1: AMR-DEP-POS-C rule

aspect(A_j) :- amr(and, op, A_j),amr(and, op, A_i), dep(A_j, conj, A_i), pos(A_j,nn), aspect(A_i), not generalWord(A_j).

R2: AMR-DEP-POS-G rule

aspect(A_j) :- amr-dep-pos(and, op-cc-nn,A_j), amr-dep-pos(and, op-cc-nn, A_i), aspect(A_i), not generalWord(A_j).

R3: DEP-AMR-POS-G rule

aspect(A) :- dep-amr-pos(H, prep-vb-nn, A), dep-amr-pos(H, amod-mod-vb-jj, O), opinionword (O), not generalWord(A).

If we don't know carrots is an aspect in advance, we cannot extract chickpeas as an aspect based on **R1** and **R2**, this error is caused by the second type of error. But based on **R3**, chickpeas can be extracted as an aspect. The extracting process is:

aspect(chickpeas) :-
dep-amr-pos(blend, prep-vb-nn, chickpeas),
dep-amr-pos(blend, amod-mod-vb-jj, nice),
opinionword(nice), not generalWord(chickpeas).

6 Conclusion

This paper proposes four integration strategies based on ASP rules to enhance deep syntax-semantics communication of language resources. Specifically, ASP rule concatenation and language structure are explored to capture complementary relationship of internal structure between syntax and semantics from different perspectives. Experiments on eight ATE datasets show that our strategies can obtain better results and have different application scenarios. In-depth analysis indicates our methods have interpretability and don't require a lot of data to train models.

In future, we plan to use graph neural network to explore our proposed integration strategies and we also plan to use our strategies to perform other NLP tasks and validate their generalization.

References

Piero Bonatti, Francesco Calimeri, Nicola Leone, and Francesco Ricca. 2010. Answer set programming. In *A 25-year perspective on logic programming*, pages 159–182.

Hongliang Dai and Yangqiu Song. 2019. Neural aspect and opinion term extraction with mined rules as weak supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5268–5277.

Shibhansh Dohare, Harish Karnick, and Vivek Gupta. 2017. Text summarization using abstract meaning representation. *arXiv preprint arXiv:1706.01678*.

Jeffrey Flanigan, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1436.

M Gelfond. 1988. The stable model semantics for logic programming. *Proc.international Conf. and Symp.on Logic Programming*, pages 1070–1080.

Hardy Hardy and Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using abstract meaning representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 768–773.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD '04*, pages 168–177.

Tengjiao Jiang, Changxuan Wan, Dexi Liu, Xiping Liu, and Guoqiong Liao. 2017. Extracting target-opinion pairs based on semantic analysis. *Chinese Journal of Computers*, (3):617–633.

Dai Junqi, Yan Hang, Sun Tianxiang, Liu Pengfei, and Qiu Xipeng. 2021. Does syntax matter? A strong baseline for aspect-based sentiment analysis with roberta. In *NAACL'21*.

Benjamin Letham, Cynthia Rudin, Tyler H McCormick, and David Madigan. 2012. Building interpretable classifiers with rules using bayesian analysis.

Gui-Ru Li and Chia-Hui Chang. 2019. Semantic role labeling for opinion target extraction from chinese social network. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 1042–1047.

Shoushan Li, Rongyang Wang, and Guodong Zhou. 2012. Opinion target extraction using a shallow semantic parsing framework. In *Twenty-sixth AAAI conference on artificial intelligence*.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2013. A logic programming approach to aspect extraction in opinion mining. In *WI-IAT '13*, pages 276–283.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015. Automated rule selection for aspect extraction in opinion mining. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2016. Automated rule selection for opinion target extraction. *Knowledge-Based Systems*, 104:74–88.

Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3538–3547.

Diego Marcheggiani, Joost Bastings, and Ivan Titov. 2018. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 486–492.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Majid Rastegar-Mojarad, Ravikumar Komandur Elayavilli, Yanshan Wang, Sijia Liu, Feichen Shen, and Hongfang Liu. 2017. Semantic information retrieval: exploring dependency and word embedding features in biomedical information retrieval. In *Proceedings of the BioCreative VI Challenge Evaluation Workshop*, pages 74–77.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using amr. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Shikhar Vashishth, Manik Bhandari, Prateek Yadav, Piyush Rai, Chiranjib Bhattacharyya, and Partha Talukdar. 2019. Incorporating syntactic and semantic information in word embeddings using graph convolutional networks. In *ACL*.

Amir Pouran Ben Veyseh, Nasim Nouri, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020. Introducing syntactic structures into target opinion word extraction with deep learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8947–8956.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017a. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Yanshan Wang, Sijia Liu, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Fei Liu, and Hongfang Liu. 2017b. Dependency and amr embeddings for drug-drug interaction extraction from biomedical literature. In *Proceedings of the 8th acm international conference on bioinformatics, computational biology, and health informatics*, pages 36–43. ACM.

Xiangpeng Wei, Heng Yu, Yue Hu, Rongxiang Weng, Luxi Xing, and Weihua Luo. 2020. Uncertainty-aware semantic augmentation for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2724–2735.

Feng Sen Yang and He Yan Huang. 2016. A hybrid method of domain lexicon construction for opinion targets extraction using syntax and semantics. *Journal of Computer ence and Technology*, 31(3):595–603.

Jianfei Yu, Jing Jiang, and Rui Xia. 2019. Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(1):168–177.

Mi Zhang and Tieyun Qian. 2020. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3540–3549.

He Zhao, Longtao Huang, Rong Zhang, Quan Lu, et al. 2020. Spanmlt: A span-based multi-task learning framework for pair-wise aspect and opinion terms extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3239–3248.