Dependency Graph based Chinese Semantic Parsing

Yu Ding[1], Yanqiu Shao[2], Wanxiang Che[1], Ting Liu[1]

[1]Harbin Institute of Technology, Harbin, China
{dingyu008,wanxiang}@gmail.com;tliu72@vip.126.com
[2]Beijing Language & Culture University, Beijing, China
yqshao163@163.com

Abstract. Semantic Dependency Parsing (SDP) is a deep semantic analysis task. A well-formed dependency scheme is the foundation of SDP. In this paper, we refine the HIT dependency scheme using stronger linguistic theories, yielding a dependency scheme with more clear hierarchy. To cover Chinese semantics more comprehensively, we make a break away from the constraints of dependency trees, and extend to graphs. Moreover, we utilize SVM to parse semantic dependency graphs on the basis of parsing of dependency trees.

Keywords: semantic analysis, semantic dependency graph, auto parsing of dependency graph.

1 Introduction

Semantic analysis is the ultimate goal of natural language processing on sentence level. For sentences as “张三(Tom)吃(eat)了(already)苹果(apple)” and “苹果(apple) 被(been) 张三(Tom) 吃(eat) 了(already),” both of which are semantically identical though with different forms of expression, Their semantic form is “吃(eat)(张三(Tom)，苹果(apple))” .This semantic information is helpful to word-sense disambiguation, information retrieval, machine translation and so on.

SDP integrates dependency structure and semantic information in the sentence, based on dependency grammar [1], which is a deep semantic analysis task. SDP consists of two steps, the first is to construct a dependency structure according to dependency grammar, i.e. to find out all pairs of words with direct semantic relations in a sentence, and then assign semantic labels between each word pairs.

The corpora with semantic-oriented dependencies already exist. [2,3] have annotated a corpus in the scale of one million words manually. They adjusted the semantic relations defined by HowNet, combining similar labels, eliminating those rarely used and revising those with semantic blurs and differences.

HIT semantic dependency is established by Research Center for Social Computing and Information Retrieval in Harbin Institute of Technology in 2011. It is also based on the semantic framework of HowNet, with the combination of LuChuan and Yuan Yulin semantic representation systems. [4] annotated 10 thousand of sentences in Penn Chinese Treebank. [5] organized the international public assessment in SemEval-2012 using this corpus. Many research institutions in China participated in this share task. However, flaws of HIT semantic dependency exist 1) there are too many fine-grained labels, many of which are rarely mentioned or used; 2) HIT corpus is annotated on news corpus, Whether it is able to cover complex linguistic phenomena is in question; and 3) there are much overlapping between labels. Thus it needs further improvement.

SDP is studied based on dependency trees. But semantic structure in Chinese often cannot be completely expressed by trees. For instance, "我们(we) 选(select) 他(he) 当(as) 班长(monitor)." The head of "他（he）" is "选(select)" referring to a patient relation, but there is also a relation between "当(as)" and "他(he)." Another example is "我(I)头(head)痛(ache)的(de)厉害(serious)，还(still)流(flow)鼻涕(snot)" According to the tree, the head of "我(I)" is "头(head)", referring to a possessive relation; yet there is also a semantic relation between "流(flow)鼻涕(snot)" and "我(I)." Thus limitation of dependency trees is obvious.

Due to the mentioned flaws, in this paper we introduce a semantic dependency scheme with stronger theory foundations on the basis of HIT's. To cover Chinese semantics in a more all-round way, we propose semantic dependency graphs on the extension of the dependency tree structure. This paper parses dependency graphs on the basis of a parser for dependency trees and a SVM classifier.

2 The Dependency Graph Scheme

2.1 Structure of Semantic Dependency Graphs

According to real corpus on large scale and considering that Chinese is a kind of parataxis language with little syntactic restrictions, we extend the traditional structure of dependency trees to dependency graphs.

Definition: Semantic Dependency Graphs are directed acyclic graphs. Nodes refer to words and sides refer to semantic relations between words with one semantic label. There is one and only one node without any head as the center of whole graph.

The graphs overcome the limitation of dependency trees, allowing the existence of more than one head on certain nodes, as well as crossing of arcs. As in Figure 1 and Table 1, node “她(she)” has semantic relations with both “脸色(face)” and “病(disease),” which means that there are two heads for “她(she)”(notice that semantic parsing here is different from topics in pragmatics)，meanwhile arcs (病(disease), 她(she)) and (难看(terrible-looking), 现在(now)) cross.

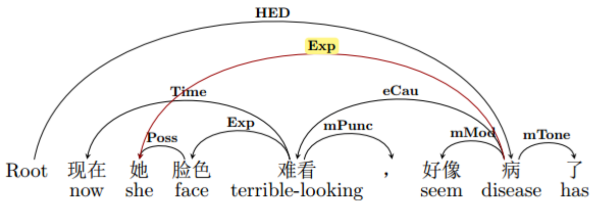

Fig. 1. An example of dependency graph

Table 1. Table form of dependency graph

Word Index	Word	Head Index	Head Word	Semantic	Descriptions
1	现在	4	难看	Time	Time
2	她	3	脸色	Poss	Possessor
2	她	7	病	Exp	Experiencer
3	脸色	4	难看	Exp	Experiencer
4	难看	7	病	eCau	Event Cause
5	，	4	难看	mPunc	Punctuation
6	好像	7	病	mMod	Modal Mark
7	病	0	HED	HED	--
8	了	7	病	mTone	Tone Mark

There are basically four rules in the theory of traditional dependency grammar (DG)[1]. Dependency graphs break the rules of traditional DG on “No element depends directly on more than one others” and “no crossings of arcs are allowed”. But its core idea still inherits from DG, for example the relations are transitive, irreflexive, and anti-symmetric. Therefore dependency graph is an extension of the DG.

2.2 Semantic Relation Scheme on the Basis of Chinese Parataxis Network

There are several problems in HIT semantic dependency scheme. There are too many semantic labels, and some of which only appear a few times. Much overlapping also appeared. HIT corpus is annotated on news corpus, but news sentences cover only limited means of Chinese expression. Thus it needs further improvement.

On the basis of Chinese parataxis network of semantic relations defined by Lu Chuan[6], this paper borrows relation labels, the classification of semantic units and the idea of semantic combination, and also integrates the characteristics of DG, to construct a semantic relation scheme of more clarity.

2.3.1 Semantic units and semantic combination

The semantic units can be divided, from high to low, into **event chain**, **event**, **argument**, **concept** and **mark**. It's worth pointing out that concept equals a simple concept in human thoughts basically, or a notional word in syntax and mark means the information which is attached to the entity information being conveyed by speakers, such as the tone and mood of the speaker. These sematic units correspond to complex sentence, minor sentence, chunk, notional word and function word. The semantic of one sentence can be expressed by the event chain which comprises of events represented by each minor sentence. The semantic of minor sentences can be expressed by the central and side arguments, while the semantic of central argument is expressed by predication concept and side argument by other referential or defining concepts. Concepts are related by marks.

The semantics of sentences comprises of semantic units, and combination ways between these units including semantic relations and semantic attachments. The semantic attachment refers to the marks of semantic units. Semantic relation includes symmetric and asymmetric relation. Symmetric relation includes coordination, selection and equivalence relations. Meanwhile, asymmetric relation includes:

Cooperative relation happens between the central concept and side concept. For example, in "工人(worker)修理(repair)管道(pipeline)", "管道(pipeline)"is the patient of "修理(repair)" and they form cooperative relation. Semantic roles usually refer to cooperative relations, and this paper defines 31 semantic roles, see appendix. **Additional relation** refers to the modification between additional concept and the central concept within the side argument, includes all kinds of roles, e.g. "地下(underground)

的(de)管道(pipeline)" (管道(pipeline), 地下(underground): Loc). **Connectional relation** means the bridging relation between two events that are of neither symmetric nor nested relation. For example, "如果(If) 天气(weather)好(good), 我(I) 会(will) 去(go) 颐和园(the Summer Palace)." the former event is the hypothesis of the latter event. There are 15 relations with respect to event in the new semantic scheme.

This paper is an improvement of the existing HIT semantic dependency scheme. It provides re-organization of HIT semantic dependency system on above theoretical basis. All semantic relations are shown in appendix.

2.3.2 Important rules

Firstly, if two words are semantic associated in a sentence, then the dependency structure must reflect this association, either through direct or transitive arc. But for the sake of simplification, if two associated words have already got indirect dependency arcs and the relation can be inferred, then there's no need for direct arc. Secondly, Chinese will not generate modifying circles, so does semantic dependency graphs.

2.3.3 Special situations

(1) Reverse relations

When the modifier in a phrase is a predication concept, it is marked as reverse relation. For example, in phrase "出现(emerge)的(de)彗星(comet)" the head of 出现(emerge) is 彗星(comet), and "彗星(comet)的(de) 出现(emerge)." the head of 彗星(comet) is 出现(emerge). Though they are different in syntactic structure and syntactic hierarchy, semantic relations on both arcs are the same, apart from the opposite direction of arcs. To distinguish them, "r" is added to sematic roles. This is consistent with HIT semantic dependency scheme.

(2) Nested events

If two events have a nested relation, namely that one event is degraded as a constituent of another and then they are "nested". For example, in the sentence "爷爷(grandpa)看见(see)他(he)的(de)<u>小(little)孙女(granddaughter)在(is)玩(play)计算机(computer)</u>.", the underlined part is degraded as a content of the action "看见(see)". The nested relation is labeled as "d-role".

3 Semantic Dependency Graph Analysis

This part will provide detailed analysis of dependency graphs.

An analysis of corpus data reveals that the occurrence of crossed arcs can be divided into following categories.1) many omissions exist in recount structures; 2) the omission of central predicates can easily lead to crossed arcs; 3) in some sentences, the important part is pre-posed and later referred to by pronouns, namely the occurrence of equivalence relation is often accompanied by crossed arcs; and 4) flexible “ba”-sentences. Following are corresponding examples.

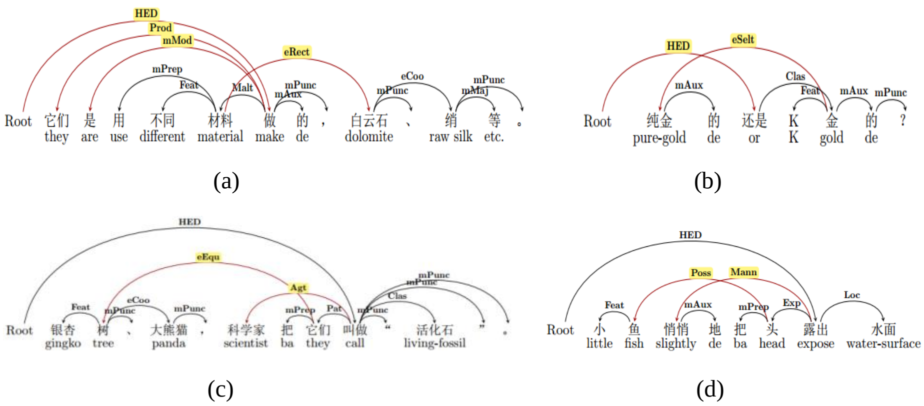

Fig. 2. Examples of crossed dependency arcs

The occurrence of nodes with multiple heads can be divided into 1) pivotal sentences caused by the causatives; 2) the reference of pronouns often cause multiple heads; 3) in a sentence comprising of several clauses, when the latter clause omits its subject and its subject is different from the former clause, there is a need for labeling the relation between predicate in the latter clause and its corresponding subject, As shown in figure 3 c), the subject of “流(flow)” is “我(I)” while the subject of “痛(pain)” is “头(head)”. It is easy to treat “头(head)” as the subject of “流(flow)” mistakenly with the induction of dependency arcs if omit the arc of “流(flow)” and “我(I)”. 4) An interlock takes place as the event degraded and the central word in the interlocking structure does not have direct semantic association with the modified part. Figure 4 gives examples under each situation.

In traditional SDP, the dependency arcs beyond the trees are omitted, such as (“采(pick)”,“自己(self)”: Agt) ，(“他(he)”,“汤姆(Tom)”: eEqu), (“流(flow)”,“我(I)”:

Agt) and (“谈论(discuss)”,“话题(topic)”: rCont). Clearly, the omission of these arcs is a loss of the semantic in Chinese.

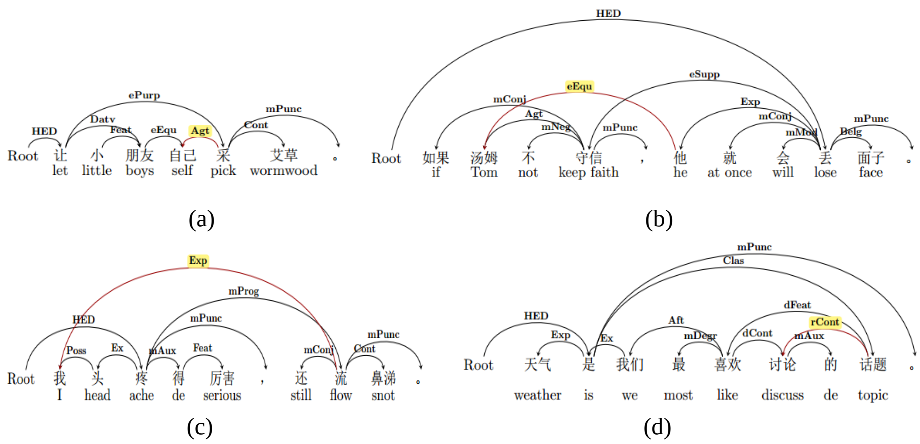

Fig. 3. Examples of nodes with multiple heads

It is known from the above analysis that, these non-projective dependency trees and dependency graphs are common language phenomena in Chinese and have a considerable quantity.

4 Automatic parsing of semantic dependency graph

Semantic dependency graph is an extension of dependency tree, which supplement relations discarded by trees. So based on the parsing of dependency trees, we use SVM to find arcs didn’t appear in the trees and then assign semantic labels to each arcs.

Table 2. Statistics of all types of sentences

	Number	**Sent Percentage**	**Arc Percentage**
all sentences	10501	--	--
with crossed arcs	374	3.56%	0.33%
with multiple arcs	1025	9.77%	1.02%
with both	264	2.51%	--

We annotated 10,501 sentences, which mainly come from primary textbooks and Chinese-English bilingual corpus. The average length is 18 words. The statistics of sentences with crossed arcs and nodes with multiple heads is shown in table 2.

4.1 Parsing of Dependency Trees

We adopt transition-based dependency parsing algorithm to process non-projective dependency trees. The transition actions we used come from [7]. The main idea of this algorithm is that using a queue to keep the tokens popped out of stack in order to be compared with following unprocessed tokens in the buffer, thus this algorithm can successfully process non-projective dependency trees. The decoding decision is consulted by gold-standard trees during training and beam-search [8] during decoding. Learning algorithm adopts averaged-perceptron [9, 10].

4.2 Parsing Dependency Graphs based on SVM

There are two steps to construct dependency graphs. Firstly, set up human-written rules based on the analysis of section 3 to get the candidate arcs, and then use SVM to select the arcs really needed to be incorporated. Finally, carry out another SVM on selected arcs and determine the semantic relations.

SVM classification relies on the design of features [11, 12]. We use some features proposed by [13] as the basis, such as the unigram bigram trigram features. Apart from those, we also add feature about the frequency of dependency arcs within the training corpus and about the two words' nearest common ancestor, including morphology, distances to ancestor and postags on the path to common ancestor respectively.

Analysis of real corpus reveals that dependency arcs existing beyond the dependency trees have numerable semantic relations. Therefore, the process of assigning semantic labels can be treated as a multi-classification problem. The features used here are the same as arcs identification used.

4.3 Results and Analysis of Dependency Tree Parsing

Table 3. Corpus statistics of our experiments

	Total	**Project Trees**	**Graphs**	**Non-Project Trees**
Train	8082	7206	790	86
Test	2015	1800	195	20
Dev	404	360	40	4

Table 3 shows the corpus statistics of our experimental data. The result of parsing of dependency tree is that UAS is 85.38% and LAS is 69.37%.

It is relatively low for LAS. In annotating stage, to further clarify the annotation, the hierarchy of the labels is clearly set, we can first pin down the main category of a sentence and find out the specific label later on, but all labels are equally treated by the parser. All together 98 labels are processed, thus the training is insufficient.

Table 4. Results of postags

Postag	UAS	LAS
NN	86%	58%
VV	78%	60%
VE	71%	61%
NT	84%	65%
VC	73%	66%
VA	81%	66%

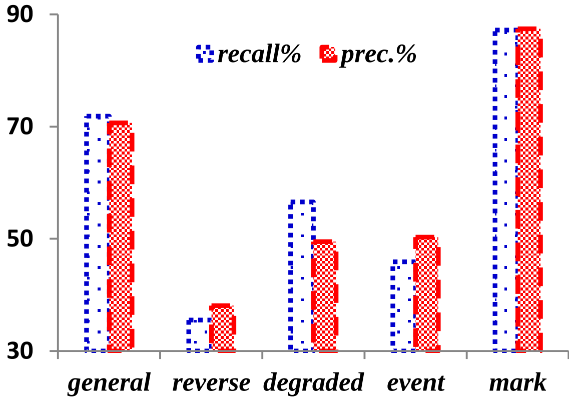

Fig. 4. Prec., Recall of semantic category

Figure 4 demonstrates reverse relations, nested relations and event relations have a relatively low rate of precision and recall. Semantic roles correspond to reverse relations and nested relations. In the training set, reverse relations and nested relations rarely exist. Thus the training on these two kinds of relations is not sufficient enough.

Semantic relations between events are similar to syntactic relations between them and no further features could help, and the arc length is longer on average. As a result, the precision of event relations is also relatively low. Whereas semantic marks is relatively higher, since more adverbs and conjunctions make up semantic marks and all the words included in each mark can be enumerated.

Table 4 shows that verbs have low rate of UAS and LAS. The following are reasons. Firstly, the existence of compressed sentences makes labeling of verbs more difficult. Two verbs form many relations, for example, eSucc, ePurp, dMann and so on. Secondly, Pivotal phrases in pivotal sentences usually express different meanings, and lead to multiple semantic relations between two actions, as eResu and eCau etc. Thirdly, relations between two clauses are represented by two kernel words of the two events. Mostly, the kernel words are verbs. However, according to our previous analysis, event relations have a relatively low rate of precision.

4.4 Results and Analysis of Appending Additional Arcs

The experimental data is the test set in part 4.2. There are 195 Sentences with 210 extra arcs. Using rules to select candidate arcs, the recall is 98.86%. However, the recall in test set is only 76.42%. False arcs and labels created by auto-parsing fail the rules. To control the scale of candidate arcs, we cannot over generalize these rules. So the recall here is a compromise. The average amount of candidate arcs for each sentence is 6.8.

Table 5. The evaluation of positive instances

Feature Set	P%	R%	F%
basic	60.00	52.67	56.15
+ Word_pair_freq	62.29	51.90	56.62
+ Nearest_common_ancestor	68.57	50.00	57.83

The evaluation result in positive instances is shown in Table 5.When add two features subsequently, the result rises 1.68%. The feature of word pair frequency helps on finding heads of arcs. The common ancestor features help to distinguish whether two words are in the same semantic chunk and to find distance between two words. This is helpful to eEqu, since if two words are close to their common ancestor, even if they are distant from one another they can still express relatively close semantic information.

After analyzing the erroneous sentences, it is clearly illustrated that the majority of errors occur with eEqu mainly in the sentences with multiple nouns and pronouns, still eEqus between closer nouns or pronouns are recognized better than far distance. Errors produced by the tree parser is cascaded here, features extracted for SVM is partly wrong. This is another important reason affecting the result of classification. Both of these problems stand in the way of the construction of semantic dependency graphs.

Categorizing the previously classified dependency arcs to different semantic labels, we have a precision of 71.96%.

5 Conclusion

At the beginning, we analyzed the flaws of HIT semantic dependency scheme, to refine it, We utilized strong linguistic theories to propose a new scheme with clear semantic hierarchy and more standard labels. Moreover, it extends, on the basis of DG, the structure to dependency graphs, which are more suitable for Chinese semantics. We annotated 10 thousand sentences with our dependency scheme. In the later part, we will share the corpus with other researchers. Lastly, we introduce an auto parsing system of semantic dependency graphs. The serial process of the system led to cascading errors, causing parsing results of low accuracy. How to enhance the parsing of dependency graphs is our main work in the future.

Acknowledgement

We thank the anonymous reviewers for their constructive comments, and appreciatively acknowledge the support of the National Natural Science Foundation of China (NSFC) via Grant 61170144, 61133012 and 61370164.

6 References

1. Jane J. Robinson. Dependency structures and transformation rules. Language, 46(2):259–285, (1970).
2. Mingqin Li, Juanzi Li, Zhendong Dong, Zuoying Wang, and Dajin Lu. Building a large chinese corpus annotated with semantic dependency. In Proceedings of the second SIGHAN workshop on Chinese language processing. (2003).
3. Mingqin Li, Juanzi Li, Zuoying Wang, Dajin Lu. A Statistical Model for Parsing Semantic Dependency Relations in a Chinese Sentence. Chinese Journal of Computers. 12(27): 1680-1687, (2004).
4. Lijie Wang. Chinese Semantic Dependency Parsing[D]. Harbin: Harbin Institute of technology. (2010).
5. Wanxiang Che, Meishan Zhang, Yanqiu Shao, and Ting Liu. 2012. Semeval-2012 task 5: Chinese semantic dependency parsing. In Proceedings of SemEval 2012 pages 378–384, Montr ´eal, Canada, 7-8 June. Association for Computational Linguistics, (2012).
6. Chuan Lu. The Parataxis of Chinese Grammar. Commercial printing house, (2001).
7. Jinho D, Choi. and Andrew McCallum. Transition-based dependency parsing with selectional branching. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria. (2013).
8. Yue Zhang and Stephen Clark. Syntactic processing using the generalized perceptron and beam search. Computational Linguistics. 37(1):105–151, (2011).
9. Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In Proceedings of the EMNLP conference, pages 1–8, Philadelphia, PA. (2002).
10. Michael Collins and Brian Roark. Incremental parsing with the perceptron algorithm. In Proceedings of ACL, pages 111–118, Barcelona, Spain, July. (2004.)

11. Corinna Cortes, Vladimir Vapnik. Support-vector networks[J]. Machine learning, 20(3):273–297,1995.
12. Vladimir N. Vladimir Vapnik. The Nature of Statistical Learning Theory. Springer. 1995.
13. Kyung-Mi Park and Young-Sook Hwang and Hae-Chang Rim. Two-Phase Semantic Role Labeling based on Support Vector Machines. Proceedings of CoNLL, 2004.

7 Appendix

Table of whole semantic dependency relations

Semantic roles	
Subject roles	Agent(Agt), Experiencer(Exp), Affection(Aft), Possessor(Poss)
Object roles	Patient(Pat),Content(Cont),Product(Prod),Origin(Orig),Dative(Datv)
Copular roles	Belongings(Belg), Classification(Clas), According(Accd)
Cause roles	Reason(Reas), Intention(Int), Consequence(Cons)
Condition roles	Manner(Mann), Tool(Tool), material(Matl)
Space-time roles	Time(Time), Location(Loc), Direction(Dir),Process(Proc), Scope(Sco)
Measurement roles	Quantity(Quan),Quantity-phrase(Qp),Frequency(Freq),Sequence(Seq)
Other roles	Feature(Feat), Host(Host), Comparison(Comp) , Name-modifier(Nmod), Time-modifier(Tmod)
Reverse relations	
r + semantic roles	
Nested relations	
d + semantic roles	
Event relations	
Symmetric relations	Coordination(eCoo), Selection(eSelt), Equivalent(eEquv)

Consecutive relations	Precedent(ePrec), Successor(eSucc), Progression(eProg), adversative(eAdvt), Cause(eCau), Result(eRes), Inference(eInf), Condition(eCond), Supposition(eSupp), Concession(eConc), Method(eMetd), Purpose(ePurp), Abandonment(eAban), Preference(ePref), Summary(eSum), Recount(eRect)
Semantic marks	
Relation marks	Conjection(mConj), Auxiliary(mAux), Preposition(mPrep)
Attachment marks	Tone(mTone), Time(mTime), Range(mRang), Degree(mDegr), Frequency(mFreq), Direction(mDir), Parenthesis(mPars), Negation(mNeg), Modal(mMod)
Other marks	Punctuation(mPunc), Majority(mMaj), Vain(mVain), Separation(mSepa)