ILP-based Conceptual Analysis for Chinese NPs

Paul D. Ji
Center for Language and Philology
Oxford University
Paul_dji@yahoo.com.uk

Stephen Pulman
Computing Laboratory
Oxford University
sgp@clg.ox.ac.uk

ABSTRACT

In this paper, we explore a conceptual resource for Chinese nominal phrases, which allows multi-dependency and distinction between dependency and the corresponding exact relation. We also provide an ILP-based method to learn mapping rules from training data, and use the rules to analyze new nominal phrases.

1 Introduction

Nominal phrases have long been a concern in linguistic research and language processing (e.g., Copestake and Briscoe, 2005; Giegerich, 2004). Generally, nominal phrases can be classified into two categories according to whether they contain attributive clauses or not. We focus on nominal phrases without attributive clauses.

Closely related with nominal phrases, nominal compounds or base NPs have also attracted a great attention in language processing. Generally, nominal compounds refer to nominal phrases consisting of a series of nouns, while base NPs refer to non-recursive nominal phrases. However, such compounds or base NPs usually co-occur with other non-nominal words in running texts, and it is impossible to separate them during analysis. Furthermore, there exist syntactic makers for attributive clauses, e.g., 'which' or 'who' in English and ' (of)' in Chinese, nominal phrases without attributive clauses tend to be a better linguistic category for theoretical and practical investigation.

To analyze NPs, we need first to determine what kinds of information are to be recognized. In this work, we focus on conceptual relatedness between words. For example, in *linguistics and law books*, *linguistic* and *law* are both conceptually related with *books*, although *linguistic*s doesn't have a superficial syntactic relation with *books*. Then, we need to fulfill two sub-tasks. One is about representation, i.e., what schemes are to be used. The other is about analysis, i.e., how to derive the formal representation.

Regarding representation scheme, one possible strategy would be using syntactic structures, as are usually used in analysis for sentences. However, syntactic components for NPs, unlike those for sentences (e.g., V, VP, A, AP, and S, etc.), are difficult to differentiate, and rules governing nominal phrases are especially difficult to determine. As an example, consider (bank loan interest), which is a nominal compound consisting of three serial nouns. For such a NP, if a rule with binary combination is used, it would produce two structures for the unambiguous NP. If a rule with triple combination is used, as in Chinese Treebank (Xue et al., 2005), it would be difficult to disclose the lexical relation between (bank) and (loan).

Another possible representation strategy would be using dependency structures (Mel'cuk, 1988). Under this strategy, a NP could be represented as a dependency tree, which captures various lexical *control* or *dependency* in the phrase. However, traditional framework only focuses on syntactic dependency, while conceptual relatedness may exist without syntactic relations. For example, for *(economic development and law construction in Shanghai)*, in traditional dependency analysis, (Shanghai) would depend on the conjunction word (and), since conjunction words are usually regarded as *heads* in coordinate structures. Although the relatedness may go downward from the head, it would be difficult to derive the relatedness between (Shanghai) and (economic) or (law), since the two words are even not heads of the conjuncts (eco-

Coling 2008: Companion volume – Posters and Demonstrations, pages 47–50
Manchester, August 2008

nomic development) and (law development).

As to analysis of NPs, there have been a lot of work on statistical techniques for lexical dependency parsing of sentences (Collins and Roark, 2004; McDonald et al., 2005), and these techniques potentially can be used for analysis of NPs if appropriate resources for NPs are available. However, these techniques are all meant to building a dependency tree, while the conceptual relatedness in NPs may form a graph, with multi-dependency allowed. Additionally, these methods generally suffer from the difficulty of local estimation from limited contexts and the structural information is difficult to be exploited (Califf and Mooney, 2003).

Recently, relational learning methods in general and inductive logic programming (ILP) in particular have attracted a great of attention due to their capability of going beyond finite feature vectors and exploiting unbounded structural information from data (Califf and Mooney, 2003; Page et al., 2003; Srinivasan et al., 2003).

In this work, we try to extend syntactic dependency to conceptual dependency to capture the embedded lexical relatedness, and use ILP to analyze nominal phrases, making use of the structural information provided by the resources based on conceptual dependency.

2 Conceptual dependency

In comparison with syntactic dependency, conceptual dependency may allow a word to be dependent on multiple words at the same time. For example, in (Economic development and law construction in Shanghai), (Shanghai) conceptually relates with both (economic) and (law), while in (activity of blood donation for university student volunteers), (university student) relates on both (volunteer) and (blood donation).

In addition, syntactic dependency doesn't exactly specify what kind of relatedness held between words, although the words denoting the relatedness may occur within NPs. For example, 1) is an ambiguous compound with two possible interpretations listed in 2).

1) (student discussion)
2) i) discussion by students
   ii) discussion about students

However, the dependency trees corresponding with the two interpretations remain the same: (student) depends on (discussion) in both cases. In fact, their difference lies in the exact semantic relations held between the words: (student) is *agent* and *patient* of (discussion) in 2i) and 2ii) respectively. This suggests that only syntactic dependency is not enough to reflect conceptual difference.

Notice that in (2), the relations between the two words (student) and (discussion) are denoted by two proper nouns, *agent* and *patient*, which may never co-occur with them in running texts. However, in some cases, some word co-occurring with two conceptually related words do denote the relatedness exactly. Consider (car in read color), where (color) relates with both (car) and (red). In the conceptual view, (color) can be seen as a feature of (car), and (red) can be seen as a kind of value for the feature, as was also adopted in dealing with adjectives in WordNet (Fellbaum, 1988). In this setting, (red) directly depends on (car), and (color) represents the relation between them.

In building the resource for Chinese NPs, the conceptual relatedness is based on semantic reference, while the dependency is based on syntactic or potential syntactic relations. The feature words we adopt are mostly listed in a medium class, coded as *Dn*, in a Chinese thesaurus, Tongyici Cilin (henceafter Cilin, Mei et al., 1982). Function words (e.g., (from)), Part words (e.g., (leg)) and Number words (e.g., (count)) are also regarded as feature words.

3 ILP-based Analysis

Fig. 1 gives the overall structure of the analysis procedure.

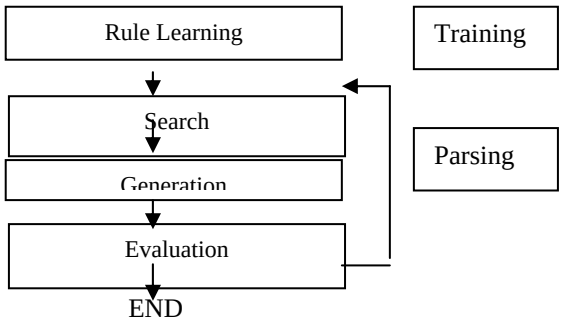

Fig. 1 Overall structure of analysis procedure

The analysis consists of two phases, training and parsing. During the training phase, rules are learned for mapping from conceptual dependency graphs to word strings based on training examples. During the parsing phase, there are three steps. Search is to find candidate dependency graphs, Generation is to generate word strings from candidate dependency graphs using

the learned rules, and Evaluation is to compare the generated word strings with the original NPs.

3.1 Training: learning rules

For each training sample, we have a nominal phrase and its corresponding conceptual dependency graph. To learn the rules mapping from dependency graphs to word strings, we need to tag the words with their sense labels, which denote the synsets in the thesaurus (Mei et al., 1982). For the sense tagging, we used the same method as in (Yarowsky, 1992) and used the minor categories in the thesaurus as the synsets.

Generally, a rule consists of two parts, Gr and Sr. Gr is a dependency sub-graph and Sr is a sense label string. Intuitively, conceptual configuration in Gr is represented by the label string of Sr.

To capture more structural information, we need to find the maximal sub-graph in the training data, whose corresponding labels form a continuous substring in the training data. But the problem is NP hard, and we thus use heuristics to find am optimally maximal sub-graphs. However, the search has a bias to larger sub-graphs, and to avoid the bias, we set the coverage of a sub-graph as the penalty. Here, the coverage of the sub-graph refers to the percentage of the nodes in the sub-graphs among all the nodes in the training data. The overall algorithm is:

i) to find the most common edge in the training data, whose corresponding label strings are continuous;
ii) to add another edge to the sub-graph, if the label strings corresponding with the new sub-graph are still continuous until the coverage of the sub-graph doesn't increase.

After finding such a sub-graph, we merge all the nodes into one, and merge the sense label strings into one, and repeat the process until all the nodes in the training data are covered The result of the learning is a set of rules, and each rule specifies a sub-graph and a label string.

For example, w got a rule which includes the sub-graph in Fig. 2 and sense label string in 3).

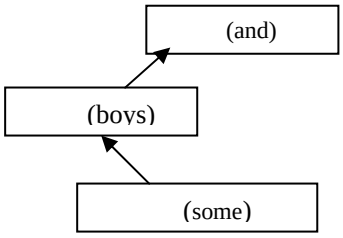

Fig. 2. Sub-graph in a rule.

3) SL( )SL( )SL( )

For rule generalization, we don't try to compress the rule set, and simply use the sense hierarchy in the thesaurus, including the minor, medium and major classes.

3.2 Parsing

After training phase, we get a set of learned rules. During parsing, the task is to find a conceptual dependency graph for a new input data, which would generate the NP using the learned rules.

The optimal parsing can be implemented in a greedy manner. First, one dependency with two words is selected. Then, another word is added if the resulted conceptual dependency graph generates a word string which *best* matches the input nominal phrase. This process can be repeated until the graph includes all the words in the data.

To compare the generated word string with the original input, we use *edit distance* between them, which is based on the times of *operations* (including *adjacent move*, *deletion*, *insertion*) needed to convert one word string to another.

4 Experiments and Evaluation

There are 10,000 nominal phrases annotated in the resource, and they were selected from 1,221 articles form the corpora of China daily, 1992. Table 1 gives the statistics of the resource.

num	'de' structure	Nominal compound	Dependency with feature	Multi-dependency
10K	4,234	5,766	1,235	976

Table 1. Statistics of NP resource

Here, 'de' structure refer to the phrase with word ' ' (of). Nominal compounds refer to the nominal phrases with no occurrence of ' '(of). Dependency with features refers to those tagged with features, which also occur in the same NPs. Multi-dependency refers to the number of mono-dependencies occurring in the multi-dependency.

We randomly selected 10% of the training data as closed test data, and the other 90% or less as training data. To evaluate the performance of the dependency analysis, we used *F*-scores as evaluation measure as usual. Fig. 4 shows the results for overall dependency, multi-dependency and dependency with features. The results are averaged over 10 random runs.

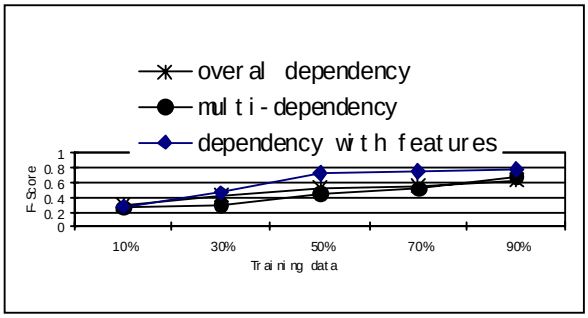

Fig. 3 Performance with varying training data

Fig. 3 demonstrates that with more training data, the performance generally improved. The performance for dependency with features seemed better than that for overall dependency or multi-dependency. To check the reason, we found that we treated the Amount words in Number-Amount structures as features, and these words are generally easier to be identified, since they tend to be unambiguous. Once they were recognized as Amount words, the relevant dependency would be correctly identified.

For an open test, we selected another 1,000 nominal phrases from the same corpus, but from different time period (1994). Such phrases were annotated with the same standard as those training data. Fig. 4 shows the results with varying training data.

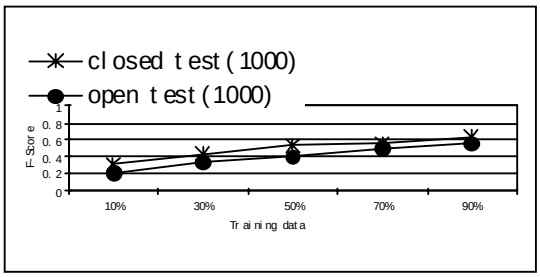

Fig. 4 Comparison: Closed test and open test

Fig. 4 shows that the open test performance is generally worse than that of the closed test. Notice that although the test data was selected from the same resource, but with a different period, which may account for the different performance.

5 Conclusion

In this paper, we described a resource for lexical conceptual dependency of Chinese nominal phrases. Compared with other ones, it allows multi-dependency and distinguishes dependency and relation, which exactly denotes what kinds of dependency held. We also provided an ILP-based analysis method, in which some rules mapping from conceptual dependency to word strings are learned from the training data, and then the rules are used to find the conceptual dependency graph for a new data. Compared with other search strategies, this method makes use of the structural information and allows construction of a dependency graph, not just a dependency tree.

References

Califf, M.R. and Mooney, R.J. 2003. Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction, JMLR: 4:177-210.

Collins M. and Roark, B. 2004. Incremental parsing with the perceptron algorithm. In *Proc. of the 42rd Annual Meeting of the ACL*.

Copestake, A. and Briscoe, T. 2005. Noun compounds revisited. In John I. Tait, editor, *Charting a New Course: Natural Language Processing and Information Retrieval*. Springer, Berlin, 2005.

Fellbaum, editor. 1998. WordNet: An Electronic Lexical Database. The MIT Press.

Giegerich, H.J. Compound or phrase? English noun-plus-noun constructions and the stress criterion. *English Language and Linguistics*, 8(1):1-24, 2004.

McDonald, R., Pereira, F., Ribarov, K. and Hajič, J. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proc. of HLT/EMNLP*.

Mei, J., Zhu, Y., Gao, Y., and Yin, H. 1982. Tongyici Cilin. Shanghai Dictionary Press.

Mel'cuk, I., 1988. Dependency Syntax: Theory and Practice. Albany. State Univ. of New York Press.

Page, D. Srinivasan A. 2003. ILP: A Short Look Back and a Longer Look Forward. Journal of Machine Learning Research 4: 415-430

Srinivasan, A. Ross D. K., Michael B. 2003. An Empirical Study of the Use of Relevance Information in Inductive Logic Programming. Journal of Machine Learning Research 4: 369-383.

Xue, N.W., Xia, F., Chiou, F.D.and Palmer, M. The Penn Chinese TreeBank: Phrase Structure Annotation of a Large Corpus. Natural Language Engineering, 11(2): 207-238.

Yarowsky, D. 1982. Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. In *Proceedings, COLING-92*. pp. 454-460, 1992.