Extracting Hyponymy Relation between Chinese Terms*

Yongwei Hu and Zhifang Sui

Institute of Computational Linguistics (ICL), Peking University, Beijing, China
yongweihu@pku.edu.cn, szf@pku.edu.cn

Abstract. This paper studies the problem of the automatic acquisition of the hyponymy (is-a) relation in sentences and develops a new method for it. In this paper, we treat the task of identifying hyponymy relation as two separate problems and solve them based on the following three techniques: term type's commonality, sequential patterns, property nouns and domain verbs.

Keywords: hyponymy, relation extraction, pattern-based, sequential pattern, property noun, domain verb.

1 Introduction

Detecting terms and hyponymy relation among them in text data has many applications. The previous work on identifying hyponymy relation commonly used pattern-based methods and had several problems. We can classify terms that have hyponymy relation into two types: hyponym and hypernym. This paper develops a novel approach for acquiring hyponymy relation by modeling commonality of hyponyms and that of hypernyms separately. This method is different from traditional approaches in that terms having hyponymy relation don't need to occur syntactically near one another. Intuitively, this method could extract more relation instances from corpora. In order to solve the second problem, we introduce the Sequential Patterns (SP), which is another pattern representation method and is well-known in Data Mining field.

The rest of this paper is structured as follows. Section 2 discusses related works. Section 3 defines the problem. Section 4 records our preliminary experiments we ran. Finally, section 5 makes a conclusion of our work.

2 Related Work

Research on recognizing relations can be classified into three categories. The first category uses statistical techniques, such as (Miller et al., 2000), (Zhao and Grisman, 2005), and (Zhou et al., 2006). Statistical approaches perform well on large corpora, but for their good performance a large number of features have to be explored and

* This paper is supported by 863 High Technology Project of China (No.2006AA01Z144), NSFC Project 60503071 and Beijing Natural Science Foundation 4052019.

H. Li et al. (Eds.): AIRS 2008, LNCS 4993, pp. 567–572, 2008.

many training examples must be labeled, which is expensive and time-consuming for some domain.

The second category makes use of hand-crafted or automatically extracted rules. This type of approaches is pioneered by Hearst (1992). Manually selecting as seed instances a list of term pairs for which the target relation is known to hold, Hearst sketched an algorithm to learn patterns that indicate the relation of interest, and then use these pat-terns to extract more instances. These methods extract patterns from sentences containing both terms of seed instances, which limit the number of relation instances we can get because that not all relation instances would occur syntactically near one another.

Another related work is about Sequential Patterns (SP). SPs have been used in many fields to solve quite different problems, such as, (Sun et al. 2007), (Jindal and Liu, 2006). The work in (Sun et al. 2007) focuses on the problem of detecting erroneous/correct sentences.

3 Proposed Technique

This section first defines the problem in a formal way and then presents our solution.

3.1 Problem Statement

Let T be a set of terms in a domain D. Given a corpus, we could treat all terms in it as T. We say term t1 in T is a hyponym of term t2 if people accept sentences constructed from the frame *A/An t1 is a (kind of) t2*. Here, t2 is said to be a hypernym of t1. Let T_{hypo} be the set of all hyponyms in T and T_{hyper} the set of all hypernyms in T. A hyponymy relation, r, is in the form of <t1, t2>, where term t1 is a hyponym of term t2. Let R_T be a set of relations among terms in T and $T_{hypo} \times T_{hyper}$ represent the set of all term pairs composed of terms in T_{hypo} and T_{hyper}, and, obviously, $R_T \subseteq T_{hypo} \times T_{hyper.}$

We treat the task of identifying hyponymy relation as two separate problems. The first problem is defined as follows. Note that terms are already labeled in corpora and given to us as input.

Problem 1(Term Type Recognition). Suppose T is the set of terms in corpora D; recognize the set of hyponyms T_{hypo} and the set of hypernyms T_{hyper} in D. Problem 1 is solved in next subsection. After identifying terms' type, the next problem at hand is that of determining whether a term pair has the hyponymy relation.

Problem 2(Relation Identification). Given two sets T_{hypo} and T_{hyper}, identify legal term pairs. A term pair (t1, t2) is legal if it satisfies the following constraints: $t1 \in T_{hypo}$, $t2 \in T_{hyper}$ and t1 is a hyponym of t2.

3.2 Term Type Recognition

To solve this problem, we first present the following assumption.

Hypothesis 1. *If two terms in T hold the same term type (either hyponym or hypernym), their occurrences in text data tend to have similar context.*

For many domains, this assumption is intuitively true. Based on the assumption, for a given corpus T, ideally, we could recognize all terms that are hyponyms and all those that are hypernyms, and get the two sets, namely, T_{hypo} and T_{hyper}. The strategy we adopt for this recognition problem is similar in spirit to the pattern-based techniques used in earlier relation extraction works. The difference lies in that patterns here are composed of distant words in sentences and that we want to extract patterns indicating term types (i.e. hyponym and hypernym) rather than hyponymy relation.

In order to extract patterns from sentences, we introduce the idea of Sequential Patterns (SP) from Dining Mining. The definitions of sequence and sequential pattern and the algorithms for extracting such patterns are introduced in (Sun et al. 2007).

3.3 Relation Identification

Terms in a specified domain are usually associated with meaningful phrases which could be used to show their semantic features and are usually domain-specific. For example, the noun phrase容量(*volume*) describes a property of the term 随机存储器(*RAM*) in sentence "随机存储器的容量是大多数任务的关键参数(*RAM volume is a critical parameter for the majority of tasks*)." In sentence, "这种驱动使用SCSI子系统存取USB存储器(*This driver uses the SCSI subsystem to access to the USB storage device*)", the verb phrase 存取(*access*) indicate the action we can take on the term USB存储器(*USB storage device*), a property of the term USB存储器(*USB storage device*).

In terms of Part of Speech (POS), we classify phrases that could show terms' properties into two categories: property noun and domain verb. Phrase 容量(*volume*) is one example of property noun. As other examples, phrase 速度(*speed*) describing term 处理器(*CPU*), phrase 大小(*size*) describing term 笔记本电脑(*notebook*). Phrase 存取(*access*) is one example of domain verb and 关闭(*turn off*) associated with 计算机(*computer*) is another example.

Note that two terms having hyponymy relation are often described by similar property nouns and domain verbs. Take relation r = <笔记本电脑*(notebook),* 计算机*(computer)*> as an example. Term 计算机(*computer*) can be described with property noun大小(*size*), so can term笔记本电脑*(notebook)*, and they both can be described with domain verb关闭(*turn off*). Therefore, if we found property nouns and domain verbs connected with every term in term set T, it would be easy to solve the second problem, by just selecting all those term pairs described by similar property nouns and domain verbs.

Property nouns and domain verbs in a specific domain D1 could be specified manually. In this paper, we get all the verbs and nouns relatively specific to corpus T1 in D1 and use them as the domain verbs and property nouns. We treat all extracted phrases as property nouns and domain verbs in T1. This is because property nouns and domain verbs are domain-specific and corpus T2 is used to filter out all those phrases. After dividing terms into hyponym and hypernym and extracting phrases which show properties of terms, we construct for each term a feature vector which

consists of all the phrases we extracted. If a term includes a phrase, the corresponding feature is set at 1. Term pair <t1, t2> having hyponymy relation must satisfy some constraints. For example, term t1 and t2 cannot be the same; the similarity between t1 and t2 must be bigger than a threshold *min_sim*. We sort the identified relation instances according to the similarities of their terms at last.

4 Experiments

The following subsections describe the experiments we ran in computer domain and the experimental results.

4.1 Experimental Setup

In order to evaluate our algorithm, we first collected sentences from the book 计算机科学技术百科全书(Encyclopedia of Computer Science and Technology), which are mostly technical essays in computer domain, and tagged all terms in these sentences. Among the collected sentences, 3623 sentences contain terms and 740 terms are labeled. There are about 1282 hyponymy relation instances. In order to extract property nouns and domain verbs in target domain, we collected 1000 sentences from the Chinese broadcast news training data for ACE 2004, which are mainly daily news and definitely a different domain.

4.2 Experimental Results

Term Type Recognition. The experiment needs some relation instances as seeds to bootstrap. The seeds we selected are: <笔记本计算机(notebook), 计算机(computer)> , <磁带存储器(tape), 存储器(storage)>, <键盘(keyboard), 输入设备(input device)>, <环网(ring network), 局域网 (LAN)>. We adopted the frequent sequence mining algorithm in (Pei et al., 2001) for learning patterns. In order to ensure that our discovered pattern p is not too general, this mining algorithm needs us to specify an argument, min_sup, denoting the minimum number of terms whose context contains the pattern p. In our experiment, min_sup is empirically set to 5 for hyponym and 7 for hypernym. At last, we get two sets T_{hypo} and T_{hyper}. T_{hypo} contains 452 terms and T_{hyper} contains 523 terms. Note that the number of terms in T_{hypo} T_{hyper} is larger that 740, the total number of terms in the corpus. This is because some terms are actually both hyponym and hypernym. In addition, there are also terms that are not contained in any set, such as term临界区. This is mainly due to the data sparseness problem in the corpus and few sentences contain these terms. The performance of the step is showed in Table 1.

Table 1. Result of Term Type Recognition

Type	P	R	F
hyponym	70.82	92.14	80.08
hypernym	62.34	85.78	72.21

Table 2. Performance of Relation Identification Effected by *k*

k	P	R	F
300	87.67	20.51	33.25
400	83.75	26.13	39.83
500	77.60	30.27	43.55
600	74.83	35.02	47.72
800	76.38	41.42	51.01
1000	64.10	50.00	56.18
1200	57.00	53.35	55.12

Table 3. Performance of Relation Identification Effected by *min_sim*

min_sim	#instances	P	R	F
0.9	121	88.43	8.35	15.25
0.8	543	57.83	24.49	34.41
0.7	1028	61.67	49.45	54.89
0.5	8231	8.65	55.54	14.97
0.3	21384	3.85	64.20	7.26

Property Nouns and Domain Verbs. This step is relatively simple. For the parameter, freq, we empirically set at 10. Some examples of the extracted property nouns: 类型(type), 价格(price), 性能(performance), 体积(size), 速度(speed), 复杂性(complexity), 效率(efficiency). Some examples of the discovered domain verbs: 计算(calculate), 运算(operate), 加(add), 转换(transform), 命中(hit), 执行(execute), 检索(search), 存储(store), 储存(store), 保存(save), 存放(put), 输入(input), 输出(output), 传送(send), 传输(transfer),共享(share), ,分布(distribute),通信(communicate). Due to space limitation, we do not show all the phrases we extracted.

Relation Identification. The experimental results are presented in Table 2, Table 3. We calculated the precision, recall, and F-score. There are two different ways to affect the number of relation instances our algorithm extract, by setting parameter k, the amount of relations our algorithm outputs, or setting another parameter min_sim, which determines when two terms should be identified as a hyponymy relation. Table 2 reports the performance of the first method. And the performance of the second method is presented in Table 3. As can be seen from Table 3, the highest precision is achieved when min_sim is set at 0.9 and with large threshold, the performance deterioration is significant. At the same time, this proves our assumption that terms having hyponymy relation are usually described by similar property nouns and domain verbs. As shown in Table 2, our technique got the best performance, e.g. 56.18%, when we set k at 1000. When k is relatively small, we can achieve high precision. This is because we sorted all the extracted instances according to their terms' similarities and then the k-top instances have the largest similarities.

Comparing with Other Methods In this paper, we compare our technique with (Hearst, 1992). As discussed in Section 2, Hearst (1992) pioneered the pattern-based relation extraction method, and proposed a relation extraction framework which is used by nearly all pattern-based like methods. The best result achieved by this approach is: precision: 42.24% recall: 39.78%, f-measure: 40.97%. It is obvious that our method outperforms Hearst(1992) in terms of precision, recall and f-measure. After

comparing the relation instances they found, we realize that many instances got by our method don't necessarily contain terms that occur in the same sentence. That is to say, even though two terms appear far enough in the corpus, our technique could still determine whether they have the hyponymy relation. As stated in Section 2, in all earlier pattern-based like methods we know of, terms having the target relation must occur syntactically near one another. Therefore, these methods could not find term instances far away in the corpus as well.

5 Conclusions

This paper proposed a new method to identify hyponymy relation. Empirical evaluating in Computer domain demonstrated the effectiveness of our techniques. This method is actually based on two assumptions. One is that the same term type has similar context. The other is that two terms having the hyponymy relation will be described by similar property nouns and domain verbs in the corpus. Our method could find relation instances on a global level, which is its improvement over other pattern-based methods.

References

1. Sun, G., Cong, G., Liu, X., Lin, C.-Y., Zhou, M.: Mining sequential patterns and tree patterns to detect erroneous sentences. In: AAAI (2007)
2. Hearst, A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: Proc. Of COLING 1992, pp. 23–28 (1992)
3. Miller, S., Fox, H., Ramshaw, L., Weischedel, R.: A novel use of statistical parsing to extract information from text. In: Proc of 6th Applied Natural Language Processing Conference, Seattle, USA, 29 April- 4 May (2000)
4. Zhao, S.B., Grisman, R.: Extracting relations with integrated information using kernel methods. In: Proc. Of ACL 2005, pp. 419–426 (2005)
5. GuoDong., Z., Jian, S., Min, Z.: Modeling Commonality among Related Classes in Relation Extraction. In: Proc. Of ACL 2006, pp. 121–128 (2006)