Exploiting Multiple Sources for Open-domain Hypernym Discovery

Ruiji Fu, Bing Qin, Ting Liu*
Research Center for Social Computing and Information Retrieval
School of Computer Science and Technology
Harbin Institute of Technology, China
{rjfu, bqin, tliu}@ir.hit.edu.cn

Abstract

Hypernym discovery aims to extract such noun pairs that one noun is a hypernym of the other. Most previous methods are based on lexical patterns but perform badly on open-domain data. Other work extracts hypernym relations from encyclopedias but has limited coverage. This paper proposes a simple yet effective distant supervision framework for Chinese open-domain hypernym discovery. Given an entity name, we try to discover its hypernyms by leveraging knowledge from multiple sources, i.e., search engine results, encyclopedias, and morphology of the entity name. First, we extract candidate hypernyms from the above sources. Then, we apply a statistical ranking model to select correct hypernyms. A set of novel features is proposed for the ranking model. We also present a heuristic strategy to build a large-scale noisy training data for the model without human annotation. Experimental results demonstrate that our approach outperforms the state-of-the-art methods on a manually labeled test dataset.

1 Introduction

Hypernym discovery is a task to extract such noun pairs that one noun is a hypernym of the other (Snow et al., 2005). A noun H is a hypernym of another noun E if E is an instance or subclass of H. In other word, H is a semantic class of E. For instance, "actor" is a hypernym of "Mel Gibson"; "dog" is a hypernym of "Caucasian sheepdog"; "medicine" is a hypernym of "Aspirin". Hypernym discovery is an important subtask of semantic relation extraction and has many applications in ontology construction (Suchanek et al., 2008), machine reading (Etzioni et al., 2006), question answering (McNamee et al., 2008), and so on.

Some manually constructed thesauri such as WordNet can also provide some semantic relations such as hypernyms. However, these thesauri are limited in its scope and domain, and manual construction is knowledge-intensive and time-consuming. Therefore, many researchers try to automatically extract semantic relations or to construct taxonomies.

Most previous methods on automatic hypernym discovery are based on lexical patterns and suffer from the problem that such patterns can only cover a small part of complex linguistic circumstances (Hearst, 1992; Turney et al., 2003; Zhang et al., 2011). Other work tries to extract hypernym relations from large-scale encyclopedias like Wikipedia and achieves high precision (Suchanek et al., 2008; Hoffart et al., 2012). However, the coverage is limited since there exist many infrequent and new entities that are missing in encyclopedias (Lin et al., 2012). We made similar observation that more than a half of entities in our data set have no entries in the encyclopedias.

This paper proposes a simple yet effective distant supervision framework for Chinese open-domain hypernym discovery. Given an entity name, our goal is to discover its hypernyms by leveraging knowledge from multiple sources. Considering the case where a person wants to know the meaning of an unknown entity, he/she may search it in a search engine and then finds out the answer after going through the search results. Furthermore, if he/she finds an entry about the entity in an authentic web site, such as Wikipedia, the information will help him/her under-

*Email correspondence.

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1224–1234,
Seattle, Washington, USA, 18-21 October 2013.

stand the entity. Also, the morphology of the entity name can provide supplementary information. In this paper, we imitate the process. The evidences from the above sources are integrated in our hypernym discovery model.

Our approach is composed of two major steps: hypernym candidate extraction and ranking. In the first step, we collect hypernym candidates from multiple sources. Given an entity name, we search it in a search engine and extract high-frequency nouns as its main candidate hypernyms from the search results. We also collect the category tags for the entity from two Chinese encyclopedias and the head word of the entity as the candidates.

In the second step, we identify correct hypernyms from the candidates. We view this task as a ranking problem and propose a set of effective features to build a statistical ranking model. For the parameter learning of the model, we also present a heuristic strategy to build a large-scale noisy training data without human annotation.

Our contributions are as follows:

- We are the first to discover hypernym for Chinese open-domain entities by exploiting multiple sources. The evidences from different sources can authenticate and complement each other to improve both precision and recall.
- We manually annotate a dataset containing 1,879 Chinese entities and their hypernyms, which will be made publicly available. To the best of our knowledge, this is the first dataset for Chinese hypernyms.
- We propose a set of novel and effective features for hypernym ranking. Experimental results show that our method achieves the best performance.

Furthermore, our approach can be easily ported from Chinese to English and other languages, except that a few language dependent features need to be changed.

The remainder of the paper is organized as follows: Section 2 discusses the related work. Section 3 introduces our method in detail. Section 4 describes the experimental setup. Section 5 shows the experimental results. Conclusion and future work are presented in Section 6.

2 Related Work

Previous methods for hypernym discovery can be summarized into two major categories, i.e., pattern-based methods and encyclopedia-based methods.

Pattern-based methods make use of manually or automatically constructed patterns to mine hypernym relations from text corpora. The pioneer work by Hearst (1992) finds that linking two noun phrases (NPs) via certain lexical constructions often implies hypernym relations. For example, NP_1 is a hypernym of NP_2 in the lexical pattern "such NP_1 as NP_2". Similarly, succeeding researchers follow her work and use handcrafted patterns to extract hypernym pairs from corpora (Caraballo, 1999; Scott and Dominic, 2003; Ciaramita and Johnson, 2003; Turney et al., 2003; Pasca, 2004; Etzioni et al., 2005; Ritter et al., 2009; Zhang et al., 2011).

Evans (2004) considers the web data as a large corpus and uses search engines to identify hypernyms based on lexical patterns. Given an arbitrary document, he takes each capitalized word sequence as an entity and aims to find its potential hypernyms through pattern-based web searching. Suppose `X` is a capitalized word sequence. Some pattern queries like "such as `X`" are threw into the search engine. Then, in the retrieved documents, the nouns that immediately precede the pattern are recognized as the hypernyms of `X`. This work is most related to ours. However, the patterns used in his work are too strict to cover many low-frequency entities, and our experiments show the weakness of the method.

Snow et al. (2005) for the first time propose to automatically extract large numbers of lexico-syntactic patterns and then detect hypernym relations from a large newswire corpus. First, they use some known hypernym-hyponym pairs from WordNet as seeds and collect many patterns from a syntactically parsed corpus in a bootstrapping way. Then, they consider all noun pairs in the same sentence as potential hypernym-hyponym pairs and use a statistical classifier to recognize the correct ones. All patterns corresponding to the noun pairs in the corpus are fed into the classifier as features. Their method relies on accurate syntactic parsers and it is difficult to guarantee the quality of the automatically extracted patterns. Our experiments show that their method is inferior to ours.

Encyclopedia-based methods extract hypernym relations from encyclopedias like Wikipedia (Suchanek et al., 2008; Hoffart et al., 2012). The user-labeled information in encyclopedias, such as category tags in Wikipedia, is often used to derive hypernym relations.

In the construction of the famous ontology YAGO, Suchanek et al. (2008) consider the title of each Wikipedia page as an entity and the corresponding category tags as its potential hypernyms. They apply a shallow semantic parser and some rules to distinguish the correct hypernyms. Heuristically, they find that if the head of the category tag is a plural word, the tag is most likely to be a correct hypernym. However, this method cannot be used in Chinese because of the lack of plurality information.

The method of Suchanek et al. (2008) cannot handle the case when the entity is absent in Wikipedia. To solve this problem, Lin et al. (2012) connect the absent entities with the entities present in Wikipedia sharing common contexts. They utilize the Freebase semantic types to label the present entities and then propagate the types to the absent entities. The Freebase contains most of entities in Wikipedia and assigns them semantic types defined in advance. But there are no such resources in Chinese.

Compared with previous work, our approach tries to identify hypernyms from multiple sources. The evidences from different sources can authenticate and complement each other to improve both precision and recall. Our experimental results show the effectiveness of our method.

3 Method

Our method is composed of two steps. First, we collect candidate hypernyms from multiple sources for a given entity. Then, a statistical model is built for hypernym ranking based on a set of effective features. Besides, we also present a heuristic strategy to build a large-scale training data.

3.1 Candidate Hypernym Collection from Multiple Sources

In this work, we collect potential hypernyms from four sources, i.e., search engine results, two encyclopedias, and morphology of the entity name.

We count the co-occurrence frequency between the target entities and other words in the returned snippets and titles, and select top N nouns (or noun phrases) as the main candidates. As the experiments show, this method can find at least one hypernym for 86.91% entities when N equals 10 (see Section 5.1). This roughly explains why people often can infer semantic meaning of unknown entities after going through several search results.

Furthermore, the user-generated encyclopedia category tags are important clues if the entity exists in a encyclopedia. Thus we add these tags into the candidates. In this work, we consider two Chinese encyclopedias, Baidubaike and Hudongbaike[1], as hypernym sources.

In addition, the head words of entities are also their hypernyms sometimes. For example, the head word of "皇帝企鹅 (Emperor Penguin)" indicates that it's a kind of "企鹅 (penguins)". Thus we put head words into the hypernym candidates. In Chinese, head words are often laid after their modifiers. Therefore, we try to segment a given entity. If it can be segmented and the last word is a noun, we take the last word as the head word. In our data set, the head words of 41.35% entities are real hypernyms (see Section 5.1).

We combine all of these hypernym candidates together as the input of the second stage. The final coverage rate reaches 93.24%.

3.2 Hypernym Ranking

After getting the candidate hypernyms, we then adopt a ranking model to determine the correct hypernym. In this section, we propose several effective features for the model. The model needs training data for learning how to rank the data in addition to parameter setting. Considering that manually annotating a large-scale hypernym dataset is costly and time-consuming, we present a heuristic strategy to collect training data. We compare three hypernym ranking models on this data set, including Support Vector Machine (SVM) with a linear kernel, SVM with a radial basis function (RBF) kernel and Logistic Regression (LR).

[1]Baidubaike (`http://baike.baidu.com`) and Hudongbaike (`http://www.baike.com`) are two largest Chinese encyclopedias containing more than 6.26 million and 7.87 million entries respectively, while Chinese Wikipedia contains about 0.72 million entries until September, 2013.

Feature	Comment	Value Range
Prior	the prior probability of a candidate being a potential hypernym	[0, 1]
Is_Tag	whether a candidate is a category tag in the encyclopedia page of the entity if it exists	0 or 1
Is_Head	whether a candidate is the head word of the entity	0 or 1
In_Titles	some binary features based on the frequency of occurrence of a candidate in the document titles in the search results	0 or 1
Synonyms	the ratio of the synonyms of the candidate in the candidate list of the entity	[0, 1]
Radicals	the ratio of the radicals of characters in a candidate matched with the last character of the entity	[0, 1]
Source_Num	the number of sources where the candidate is extracted	1, 2, 3, or 4
Lexicon	the hypernym candidate itself and its head word	0 or 1

Table 1: The features for ranking

3.2.1 Features for Ranking

The features for hypernym ranking are shown in Table 1. We illustrate them in detail in the following.

Hypernym Prior: Intuitively, different words have different probabilities as hypernyms of some other words. Some are more probable as hypernyms, such as *animal*, *plant* and *fruit*. Some other words such as *sun*, *nature* and *alias*, are not usually used as hypernyms. Thus we use a prior probability to express this phenomenon. The assumption is that if the more frequent that a noun appears as category tags, the more likely it is a hypernym. We extract category tags from 2.4 million pages in Baidubaike, and compute the prior probabilities $prior(w)$ for a word w being a potential hypernym using Equation 1. $count_{CT}(w)$ denotes the times a word appeared as a category tag in the encyclopedia pages.

$$prior(w) = \frac{count_{CT}(w)}{\sum_{w'} count_{CT}(w')} \quad (1)$$

In Titles: When we enter a query into a search engine, the engine returns a search result list, which contains document titles and their snippet text. The distributions of hypernyms and non-hypernyms in titles are compared with that in snippets respectively in our training data. We discover that the average frequency of occurrence of hypernyms in titles is 15.60 while this number of non-hypernyms is only 5.18, while the difference in snippets is very small (Table 2). Thus the frequency of candidates in titles can be used as features. In this work the frequency is divided into three cases: greater than 15.60, less than 5.18, and between 5.18 and 15.60. Three binary features are used to represent these cases.

	Avg. Frequency in titles	Avg. Frequency in snippets
Hypernym	15.60	33.69
Non-Hypernym	5.18	30.61

Table 2: Distributions of candidate hypernyms in titles and snippets

Synonyms: If there exist synonyms of a candidate hypernym in the candidate list, the candidate is probably correct answer. For example, when "药品 (medicine)" and "药物 (medicine)" both appear in the candidate list of an entity, the entity is probably a kind of medicine. We get synonyms of a candidate from a Chinese semantic thesaurus – Tongyi Cilin (Extended) (CilinE for short)[2] and compute the score as a feature using Equation 2.

$$ratio_{syn}(h, l_e) = \frac{count_{syn}(h, l_e)}{len(l_e)} \quad (2)$$

Given a hypernym candidate h of an entity e and the list of all candidates l_e, we compute the ratio of the synonyms of h in l_e. $count_{syn}(h, l_e)$ denotes the count of the synonyms of h in l_e. $len(l_e)$ is the total count of candidates.

[2]CilinE contains synonym and hypernym relations among 77 thousand words, which is manually organized as a hierarchy of five levels.

Radicals: Chinese characters are a form of ideogram. By far, the bulk of Chinese characters were created by linking together a character with a related meaning and another character to indicate its pronunciation. The character with a related meaning is called radical. Sometimes, it is a important clue to indicate the semantic class of the whole character. For example, the radical "虫" means insects, so it hints "蜻蜓 (dragonfly)" is a kind of insects. Similarly "疒" hints "淋巴癌 (lymphoma)" is a kind of diseases. Thus we use radicals as a feature the value of which is computed by using Equation 3.

$$radical(e,h) = \frac{count_{RM}(e,h)}{len(h)} \quad (3)$$

Here $radical(e,h)$ denotes the ratio of characters radical-matched with the last character of the entity e in the hypernym h. $count_{RM}(e,h)$ denotes the count of the radical-matched characters in h. $len(h)$ denotes the total count of the characters in h.

3.2.2 Training Data Collection

Now training data is imperative to learn the weights of the features in Section 3.2.1. Hence, we propose a heuristic strategy to collect training data from encyclopedias.

Firstly, we extract a number of open-domain entities from encyclopedias randomly. Then their hypernym candidates are collected by using the method proposed in Section 3.1. We select positive training instances following two principles:

- Principle 1: Among the four sources used for candidate collection, the more sources from which the hypernym candidate is extracted, the more likely it is a correct one.
- Principle 2: The higher the prior of the candidate being a hypernym is, the more likely it is a correct one.

We select the best candidates following Principle 1 and then select the best one in them as a positive instance following Principle 2. And we select a candidate as a negative training instance when it is from only one source and its prior is the lowest. If there are synonyms of training instances in the candidates list, the synonyms are also extended into the training set.

Domain	# of entities	
	Dev.	Test
Biology	72	351
Health Care	61	291
Food	75	303
Movie	51	204
Industry	56	224
Others	35	136
Total	350	1529

Table 3: The evaluation data

In this way, we collect training data automatically, which are used to learn the feature weights of the ranking models.

4 Experimental Setup

In this work, we use Baidu[3] search engine, the most popular search engine for Chinese, and get the top 100 search results for each entity. The Chinese segmentation, POS tagging and dependency parsing is provided by an open-source Chinese language processing platform LTP[4] (Che et al., 2010).

4.1 Experimental Data

In our experiments, we prepare open-domain entities from dictionaries in wide domains, which are published by a Chinese input method editor software Sogou Pinyin[5]. The domains include biology, health care, food, movie, industry, and so on. We sample 1,879 entities from these domain dictionaries and randomly split them into 1/5 for development and 4/5 for test (Table 3). We find that only 865 (46.04%) entities exist in Baidubaike or Hudongbaike. Then we extract candidate hypernyms for the entities and ask two annotators to judge each hypernym relation pair true or false manually. A pair `(E, H)` is annotated as true if the annotators judge "`E` is a (or a kind of) `H`" is true. Finally, we get 12.53 candidate hypernyms for each entity on average in which about 2.09 hypernyms are correct. 4,330 hypernym relation pairs are judged by both the annotators. We measure the agreement of the judges using the Kappa coefficient (Siegel and Castellan Jr, 1988). The

[3] http://www.baidu.com
[4] http://ir.hit.edu.cn/demo/ltp/
[5] http://pinyin.sogou.com/dict/

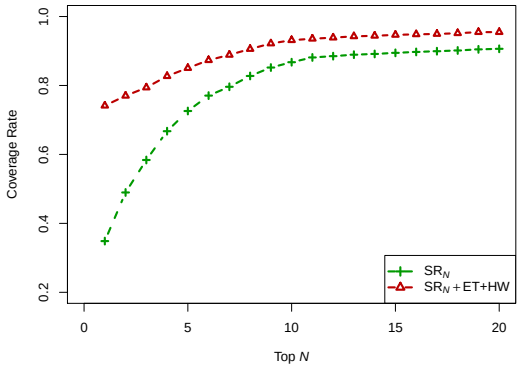

Figure 1: Effect of candidate hypernym coverage rate while varying N

Kappa value is 0.79.

Our training data, containing 11,481 positive instances and 18,378 negative ones, is extracted from Baidubaike and Hudongbaike using the heuristic strategy proposed in Section 3.2.2.

4.2 Experimental Metrics

The evaluation metrics for our task include:

Coverage Rate: We evaluate coverage rate of the candidate hypernyms. Coverage rate is the number of entities for which at least one correct hypernym is found divided by the total number of all entities.

Precision@1: Our method returns a ranked list of hypernyms for each entity. We evaluate precision of top-1 hypernyms (the most probable ones) in the ranked lists, which is the number of correct top-1 hypernyms divided by the number of all entities.

R-precision: It is equivalent to Precision@R where R is the total number of candidates labeled as true hypernyms of an entity.

Precision, Recall, and F-score: Besides, we can convert our ranking models to classification models by setting thresholds. Varying the thresholds, we can get different precisions, recalls, and F-scores.

5 Results and Analysis

5.1 The Coverage of Candidate Hypernyms

In this section, we evaluate the coverage rate of the candidate hypernyms. We check the candidate hypernyms of the whole 1,879 entities in the development and test sets and see how many entities we can collect at least one correct hypernym for.

Source	Coverage Rate	Avg. #
SR_{10}	0.8691	9.44*
ET	0.3938	3.07
HW	0.4135	0.87†
SR_{10} + ET	0.8909	12.02
SR_{10} + HW	0.9117	9.75
ET + HW	0.7073	3.92
SR_{10} + ET + HW	0.9324	12.53

Table 4: Coverage evaluation of the candidate hypernym extraction

There are four different sources to collect candidates as described in Section 3.1, which can be divided into three kinds: search results (SR for short), encyclopedia tags (ET) and head words (HW). For SR, we select top N frequent nouns (SR_N) in the search results of an entity as its hypernym candidates. The effect of coverage rate while varying N is shown in Figure 1. As we can see from the figure, the coverage rate is improved significantly by increasing N until N reaches 10. After that, the improvement becomes slight. When the candidates from all sources are merged, the coverage rate is further improved.

Thus we set N as 10 in the remaining experiments. The detail evaluation is shown in Table 4. We can see that top 10 frequent nouns in the search results contain at least one correct hypernym for 86.91% entities in our data set. This coincides with the intuition that people usually can infer the semantic classes of unknown entities by searching them in web search engines.

The coverage rate of ET merely reaches 39.38%. We find the reason is that more than half of the entities have no encyclopedia pages. The average number of candidate hypernyms from ET is 3.07. Note that the number is calculated among all the entities. We also calculate the average number only for the present entities in encyclopedias. The number reaches 6.68. The reason is that for many present entities, the category tags include not only hypernyms

*For some of entities are rare, there may be less than 10 nouns in the search results. So the average count of candidates is less than 10.

†Not all of the entities can be segmented. We cannot get the head words of the ones that cannot be segmented.

Method	Present Entities		Absent Entities		All Entities	
	P@1	R-Prec	P@1	R-Prec	P@1	R-Prec
$M_{Pattern}$	0.5542	0.4937	0.4306	0.3638	0.5229	0.4608
M_{Snow}	0.3199	0.2592	0.2827	0.2610	0.3092	0.2597
M_{Prior}	0.7339	0.5483	0.3940	0.3531	0.5494	0.4423
$M_{SVM-linear}$	0.8569	0.6899	0.6157	0.5837	0.7260	0.6322
$M_{SVM-rbf}$	0.8484	0.6940	0.6241	0.5901	0.7266	0.6376
M_{LR}	**0.8612**	**0.7052**	**0.6807**	**0.6258**	**0.7632**	**0.6621**

Table 5: Precision@1 and R-Precision results on the test set. Here the present entities mean the entities existing in the encyclopedias. The absent entities mean the ones not existing in the encyclopedias.

but also related words. For example, "布拉德利中心 (Bradley Center)" in Baidubaike have 5 tags, i.e., "NBA", "体育 (sports)", "体育运动 (sports)", "篮球 (basketball)", and "场馆 (arena)". Among them, only "场馆 (arena)" is a proper hypernym whereas the others are some related words indicating merely thematic vicinity. Comparing the results of SR_{10} and SR_{10} + ET, we can see that collecting candidates from ET can improve coverage, although many incorrect candidates are added in at the same time.

The HW source provides 0.87 candidates on average with 41.35% coverage rate. That is to say, for these entities, people can infer the semantic classes when they see the surface lexicon.

At last, we combine the candidates from all of the three sources as the input of the ranking methods. The coverage rate reaches 93.24%.

We also compare with the manually constructed semantic thesaurus CilinE mentioned in Section 3.2.1. Only 29 entities exist in CilinE (coverage rate is only 1.54%). That is why we try to automatically extract hypernym relations.

5.2 Evaluation of the Ranking

5.2.1 Overall Performance Comparison

In this section, we compare our proposed methods with other methods. Table 5 lists the performance measured by precision at rank 1 and R-precision of some key methods. The precision-recall curves of all the methods are shown in Figure 2. Table 7 lists the maximum F-scores.

$M_{Pattern}$ refers to the pattern-based method of Hearst (1992). We craft Chinese Hearst-style patterns (Table 6), in which E represents an entity and H represents one of its hypernyms. Following Evans (2004), we combine each pattern and each entity and submit them into the Baidu search engine. For example, for an entity E, we search "E 是一个 (E is a)", "E 等 (E and other)", and so on. We select top 100 search results of each query and get 1,285,209 results in all for the entities in the test set. Then we use the patterns to extract hypernyms from the search results. The result shows that 508 correct hypernyms are extracted for 568 entities (1,529 entities in total). Only a small part of the entities can be extracted hypernyms for. This is mainly because only a few hypernym relations are expressed in these fixed patterns in the web, and many ones are expressed in more flexible manners. The hypernyms are ranked based on the count of evidences where the hypernyms are extracted.

Pattern	Translation
E 是(一个/一种) H	E is a (a kind of) H
E (、)等 H	E(,) and other H
H (，)叫(做) E	H(,) called E
H (，)(像)如 E	H(,) such as E
H (，)特别是 E	H(,) especially E

Table 6: Chinese Hearst-style lexical patterns

M_{Snow} is the method originally proposed by Snow et al. (2005) for English but we adapt it for Chinese. We consider the top 100 search results for each known hypernym-hyponym pairs as a corpus to extract lexico-syntactic patterns. Then, an LR classifier is built based on this patterns to recognize hypernym relations. This method considers all nouns co-occurred with the focused entity in the same sentences as candidate hypernyms. So the number of candidates is huge, which causes inefficiency. In

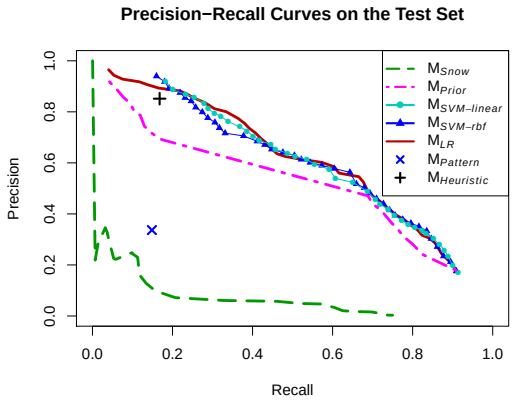

Figure 2: Precision-Recall curves on the test set

our corpus, there are 652,181 candidates for 1,529 entities (426.54 for each entity on average), most of which are not hypernyms. One possible reason is that this method relies on an accurate syntactic parser and it is difficult to guarantee the quality of the automatically extracted patterns. Even worse, the low quality of the language in the search results may make this problem more serious.

M_{Prior} refers to the ranking method based on only the prior of a candidate being a hypernym. As Table 5 shows, it outperforms M_{Snow} and achieves comparable results with $M_{Pattern}$ on Precision@1 and R-Precision.

Based on the features proposed in Section 3.2.1, we train several statistical models based on SVM and LR on the training data. $M_{SVM-linear}$ and $M_{SVM-rbf}$ refer to the SVM models based on linear kernels and RBF kernels respectively. M_{LR} refers to the LR model. The probabilities[6] output by the models are used to rank the candidate hypernyms. All of the parameters which need to be set in the models are selected on the development set. Table 5 shows the best models based on each algorithm. These supervised models outperform the previous methods. M_{LR} achieves the best performance.

The precision-recall plot of the methods on the test set is presented in Figure 2. $M_{Heuristic}$ refers to the heuristic approach, proposed in Section 3.2.2, to collect training data. Because this method cannot provide ranking information, it is not listed in Table 5. For fair comparison of R-precision and recall, we add the extra correct hypernyms from $M_{Pattern}$ and M_{Snow} to the test data set. The models based on SVM and LR still perform better than the other methods. $M_{Pattern}$ and M_{Snow} suffer from low recall and precision. $M_{Heuristic}$ get a high precision but a low recall, because it can only deal with a part of entities appearing in encyclopedias. The precision of $M_{Heuristic}$ reflects the quality of our training data. We summarize the maximum F-score of different methods in Table 7.

[6]The output of an SVM is the distance from the decision hyper-plane. Sigmoid functions can be used to convert this uncalibrated distance into a calibrated posterior probability (Platt, 1999).

Method	Max. F-score
$M_{Pattern}$	0.2061
M_{Snow}	0.1514
$M_{Heuristic}$	0.2803
M_{Prior}	0.5591
$M_{SVM-linear}$	0.5868
$M_{SVM-rbf}$	**0.6014**
M_{LR}	0.5998

Table 7: Summary of maximum F-score on the test set

Feature	P@1	R-Prec	Max. F-score
All	**0.7632**	**0.6621**	**0.5998**
– Prior	0.7534	0.6546	0.5837
– Is_Tag	0.6965	0.6039	0.5605
– Is_Head	0.7018	0.6036	0.5694
– In_Titles	0.7436	0.6513	0.5868
– Synonyms	0.7495	0.6493	0.5831
– Radicals	0.7593	0.6584	0.5890
– Source_Num	0.7364	0.6556	0.5984
– Lexicon	0.7377	0.6422	0.5851
– Source_Info	0.6128	0.5221	0.5459

Table 8: Performance of LR models with different features on the test set

5.2.2 Feature Effect

Table 8 shows the impact of each feature on the performance of LR models. When we remove any one of the features, the performance is degraded more or less. The most effective features are Is_Tag and Is_Head. The last line in Table 8 shows the performance when we remove all features about the source information, i.e., Is_Tag, Is_Head, and

Entity	Top-1 Hypernym	Entity	Top-1 Hypernym
头孢哌酮钠(cefoperazone sodium)	药品(drug)	圆舵鲣(bullet tuna)	鱼类(fish)
佛手卷(finger citron rolls)	小吃(snack)	锆英石(zirconite)	矿石(ore)
复仇者联盟(The Avengers)	电影(movie)	费利克斯托(Felixstowe)	港口(port)
套叠鞭(mastigium)	基准(datum)	基节臼(coxal cavity)	植物(plant)
乙醇胺磷酸转移酶 (Ethanolamine phosphotransferase)	生物 (organism)	彗发 (coma)	知识 (knowledge)

Table 10: Examples of entity-hypernym pairs extracted by M_{LR}

Domain	P@1	R-Prec	Max. F-score
Biology	0.8165	0.7203	0.6424
Health Care	0.7354	0.5962	0.6061
Food	0.7450	0.6634	0.6938
Movie	0.9310	0.8069	0.7031
Industry	0.6286	0.5841	0.4624
Others	0.6324	0.4936	0.4318

Table 9: Performance of M_{LR} in various domains

Source_Num. The performance is degraded sharply. This indicates the importance of the source information for hypernym ranking.

5.2.3 The Performance in Each Domain

In this section, we evaluate the performance of M_{LR} method in various domains. We can see from Table 9 that the performance in movie domain is best while the performance in industry domain is worst. That is because the information about movies is abundant on the web. Furthermore, most of movies have encyclopedia pages. It is easy to get the hypernyms. In contrast, the entities in industry domain are more uncommon. On the whole, our method is robust for different domains. In Table 10, some instances in various domains are presented.

5.3 Error Analysis

The uncovered entities[7] and the false positives[8] are analyzed after the experiments. Some error examples are shown in Table 10 (in red font).

[7]Uncovered entities are entities which we do not collect any correct hypernyms for in the first step.

[8]False positives are hypernyms ranked at the first places, but actually are not correct hypernyms.

Uncovered entities: About 34% of the errors are caused by uncovered entities. It is found that many of the uncovered entities are rare entities. Nearly 36% of them are very rare and have only less than 100 search results in all. When we can't get enough information of an unknown entity from the search engine, it's difficult to know its semantic meaning, such as "套叠鞭 (mastigium)", "基节臼 (coxal cavity)", "彗发 (coma)". The identification of their hypernyms requires more human-crafted knowledge. The ranking models we used are unable to select them, as the true synonyms are often below rank 10.

False positives: The remained 66% errors are false positives. They are mainly owing to the fact that some other related words in the candidate lists are more likely hypernyms. For example, "生物 (organism)" is wrongly recognized as the most probable hypernym of "乙醇胺磷酸转移酶 (Ethanolamine phosphotransferase)", because the entity often co-occurs with word "生物 (organism)" and the latter is often used as a hypernym of some other entities. The correct hypernyms actually are "酶 (enzyme)", "化学物质 (chemical substance)", and so on.

6 Conclusion

This paper proposes a novel method for finding hypernyms of Chinese open-domain entities from multiple sources. We collect candidate hypernyms with wide coverage from search results, encyclopedia category tags and the head word of the entity. Then, we propose a set of features to build statistical models to rank the candidate hypernyms on the training data collected automatically. In our experiments, we show that our method outperforms the state-of-the-art methods and achieves the best preci-

sion of 76.32% on a manually labeled test dataset. All of the features which we propose are effective, especially the features of source information. Moreover, our method works well in various domains, especially in the movie and biology domains. We also conduct detailed analysis to give more insights on the error distribution. Except some language dependent features, our approach can be easily transfered from Chinese to other languages. For future work, we would like to explore knowledge from more sources to enhance our model, such as semantic thesauri and infoboxes in encyclopedias.

Acknowledgments

This work was supported by National Natural Science Foundation of China (NSFC) via grant 61133012, 61073126 and the National 863 Leading Technology Research Project via grant 2012AA011102. Special thanks to Zhenghua Li, Wanxiang Che, Wei Song, Yanyan Zhao, Yuhang Guo and the anonymous reviewers for insightful comments and suggestions. Thanks are also due to our annotators Ni Han and Zhenghua Li.

References

Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 120–126, College Park, Maryland, USA, June.

Wanxiang Che, Zhenghua Li, and Ting Liu. 2010. Ltp: A chinese language technology platform. In *Coling 2010: Demonstrations*, pages 13–16, Beijing, China, August.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 168–175.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

Oren Etzioni, Michele Banko, and Michael J Cafarella. 2006. Machine reading. In *AAAI*, volume 6, pages 1517–1519.

Richard Evans. 2004. A framework for named entity recognition in the open domain. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*, 260:267–274.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545.

Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 2012. Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, pages 1–63.

Thomas Lin, Mausam, and Oren Etzioni. 2012. No noun phrase left behind: Detecting and typing unlinkable entities. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 893–903, Jeju Island, Korea, July.

Paul McNamee, Rion Snow, Patrick Schone, and James Mayfield. 2008. Learning named entity hyponyms for question answering. In *Proceedings of the Third International Joint Conference on Natural Language Processing*, pages 799–804.

Marius Pasca. 2004. Acquisition of categorized named entities for web search. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 137–145.

John Platt. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Alan Ritter, Stephen Soderland, and Oren Etzioni. 2009. What is this, anyway: Automatic hypernym discovery. In *Proceedings of the 2009 AAAI Spring Symposium on Learning by Reading and Learning to Read*, pages 88–93.

Cederberg Scott and Widdows Dominic. 2003. Using lsa and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 111–118.

Sidney Siegel and N John Castellan Jr. 1988. Nonparametric statistics for the behavioral sciences. McGraw-Hill, New York.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304.

Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. Yago: A large ontology from wikipedia and wordnet. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(3):203–217.

Peter Turney, Michael L Littman, Jeffrey Bigham, and Victor Shnayder. 2003. Combining independent modules to solve multiple-choice synonym and analogy problems. In *Proceedings of the International Conference RANLP-2003*, pages 482–489.

Fan Zhang, Shuming Shi, Jing Liu, Shuqi Sun, and Chin-Yew Lin. 2011. Nonlinear evidence fusion and propagation for hyponymy relation mining. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1159–1168, Portland, Oregon, USA, June.