DOI: 10.3969/j. issn. 1673-5188. 2017. 01. 008
http://www.cnki.net/kcms/detail/34.1294.TN.20170206.1519.002.html, published online February 6, 2017

A Method for Constructing Open-Domain Chinese Entity Hypernym Hierarchical Structure

CAI Hongbo[1], CHEN Hong[2], and LIU Shen[1]
(1. Harbin Institute of Technology, Harbin 150001, China;
2. ZTE Corporation, Nanjing 210012, China)

Abstract

Entity relation is an essential component of some famous knowledge bases, such as Freebase, Yago and Knowledge Graph, while the hyponymy plays an important role in entity relations that show the relationship between the more general terms (hypernyms) and the more specific instances of the terms (hyponyms). In this paper, we present a comprehensive scheme of open-domain Chinese entity hypernym hierarchical construction. Some of the most important unsupervised and heuristic approaches for building hierarchical structure are covered in sufficient detail along with reasonable analyses. We experimentally evaluate the proposed methods and compare them with other baselines. The result shows high precision of our method and the proposed scheme will be further improved with larger scale corpora.

Keywords

Entity hypernym; hierarchization; Apriori Algorithm; suffix hypernym; open-domain

1 Introduction

Up till now, there exists a vast amount of free text on the Web, including newswire, blogs, product reviews, emails, governmental documents, and so on. How could a computer help the human to understand all of the data? A popular idea is turning unstructured text into structured one that could represent information concisely. Therefore, in recent years, some famous knowledge bases were constructed, directly showing the structural relations between entities.

Among hyponymy relations, we can classify and cluster entities by constructing the whole knowledge structure that includes the same level relations and affiliations as well. The relation between "植物 (plant)" and "动物 (animal)" is the same level, but the one between "植物 (plant)" and "生物 (living thing)" is affiliation, called hyponymy academically. If we can reorganize and supplement such relations between entities, we will get lots of information from the entity relation knowledge graphs, such as the position of a certain entity in the entity classification system. We can also know that "动物 (animal)" and "植物 (plant)" both belong to "生物 (living thing)". It is of great significance in search recommendation of search engines. For example, in a search engine, if a user put in "姚明 (YAO Ming)" and he/she will get the recommended information of "林书豪 (Jeremy Lin)" and "易建联 (YI Jianlian)" according to the entity knowledge graph, which is one of the functions of Knowledge Graph of Google. Baidu, a search engine company in China, is doing the similar project, which has begun loading online. Entity is the basic unit in natural language processing. Entity relation extraction is a traditional problem of natural language processing, which also makes some benefits for many other natural language processing tasks and information retrieval. Constructing an accurate and comprehensive entity relation graph is a great academic significance and has practical value for artificial intelligence.

There are some other issues relevant to entity relation graphs such as knowledge maps [1] that connect knowledge with locations. However, our entity relation graphs are based on hyponymy.

Entity relations are also relevant to entity relation graphs, and many scholars have proposed related solutions. Qian [2] exploited constituent dependencies to produce the dynamic syntactic parse tree and combined the entity semantic information to improve the relation extraction performance. Fader [3] proposed Reverb system based on [4]. The Reverb system firstly recognizes the word that describes the relation, and then makes the noun phrases in context of the word be the relation arguments to constitute relation triple. Although entity relation extraction and entity relation graphs have a lot in common, the entity relation graph mainly constructs the structure of all the entities so that many other entities, instead of the certain entity itself, are involved when building the relation between every two entities. Che [5] proposed entity relation extraction based on similarity computation.

The relevant research also contains the discovery of new knowledge and academic hotspot research, which has significance for the discovery of new research points. Chen [6] started to use the method of constructing knowledge graphs to investigate the development direction of academic research as well as some promising research areas recent years.

All the previous research work concentrated on entity rela-

This work was supported by ZTE Industry-Academia-Research Cooperation Funds.

tion construction of restricted domain. There is still less research for entity relation construction of open-domain. We propose a method for constructing entity hypernyms hierarchical structure for open-domain entity type diversification, hierarchizing the hypernyms of open-domain entities.

We firstly mine the hierarchical relation between entity hypernyms by using the association between frequent itemsets. We then use the suffix information of entity hypernyms to hierarchize and complete the hierarchical structure. We propose three hierarchical methods to hierarchize entity hyponymy in different ways.

2 Hyponymy Hierarchization

All the entities we used are from Sogou Cell Dictionary[1] and Baike[2]. We obtain a large number of entities and hypernyms according to the method of hypernym discovery based on the Internet [7]. There is no hierarchy between the hypernyms. For example, "花 (flower)" and "植物 (plant)" are the hypernyms of "百金花 (centaury)", and actually "植物 (plant)" is also a hypernym of "花 (flower)". "猫科动物 (Felidae)", "哺乳动物 (mammal)" and "动物 (animal)" are the hypernyms of "美洲豹 (catamount)", however, "动物 (animal)" is a hypernym of "哺乳动物 (mammal)" while "哺乳动物 (mammal)" is also a hypernym of "猫科动物 (Felidae)". Before hierarchizing, all hypernyms are at the same level.

There are a large number of hyponymy relations between hypernyms. We need to obtain the hyponymy between hypernyms by data mining based on the entities and hypernyms.

2.1 Hyponymy Hierarchization Based on Apriori Algorithm

2.1.1 Problem Analysis

There are a large number of hyponymy relations between hypernyms [8]. By observing the data, we find that if B is a hypernym of A, most entities belong to A also belong to B. But only a small number of entities belong to B and also belong to A. For example, "植物 (plant)" is a hypernym of "单子叶植物 (monocotyledon)". Then most entities that have the hypernym "单子叶植物 (monocotyledon)" also have the hypernym "植物 (plant)" while part of entities that belong to "植物 (plant)" belong to "单子叶植物 (monocotyledon)". We could discover the hyponymy between A and B by calculating the association between A and B.

2.1.2 Frequent Itemset Association Rules Mining

The association between hypernyms is pretty similar to the frequent itemset in data mining, which we can obtain by Apriori Algorithm. Apriori Algorithm is a traditional algorithm in data mining. It aims at identifying the frequent individual items and can be used to judge whether there is hyponymy between two hypernyms. In Apriori Algorithm, there are two important parameters: confidence and support. They play important roles in our experiment.

After simple analyzing, we can know that the probability for the low support frequent itemsets contain hypernyms is relatively low as well as the low confidence frequent itemsets. Instead, the probability could be high if the confidence and support are also high.

The input for Apriori Algorithm is the entities and their hypernyms. Each entity can have several hypernyms. The output is the confidence and support for each hypernym relation between hypernyms.

Table 1 shows the sample input: each line is a hypernym relation, and mainly 2 parts.

Table 2 shows the sample output: each line is a hypernym relation, and mainly 4 parts.

The confidence is the threshold for estimating the accuracy of the hypernym relation, and its value ranges from 0 to 1. The support is the threshold for the statistical support of the hypernym relation and its value is an integer.

The confidence and support between each two hypernyms need to be calculated as follows.

$$confidence(A,B)=\frac{count(A,B)}{count(A)}, \quad (1)$$

$$support(A,B)=count(A,B), \quad (2)$$

where A and B are both hypernyms. Other variables and functions are shown in **Table 3**.

If hypernyms A and B always co-occur, this may indicate that A is one of the hypernyms of B and meanwhile B is also

▼**Table 1. Sample input**

Entity	Hypernym
百金花 (centaury)	植物 (plant)
百金花 (centaury)	花 (flower)
百金花 (centaury)	中药 (traditional Chinese medicine)
日本角鲨 (Squalus japonicus)	生物 (living thing)
日本角鲨 (Squalus japonicus)	动物 (animal)
黄色白茧蜂 (Phanerotoma flava Ashmead)	昆虫 (insect)

▼**Table 2. Sample output**

Hypernym A	The hypernym of A	Confidence	Support
川菜 (Sichuan Cuisine)	饮食 (diet)	1	38
石竹亚纲 (Caryophyllidae)	植物 (plant)	0.968994	64
种子植物门 (Spermatophyta)	植物 (plant)	0.977477	1501
石竹目 (Caryophyllales)	种子植物门 (Spermatophyta)	1	63
鸟类 (bird)	动物 (animal)	1	58
冬青属 (ilex L.)	双子叶植物纲 (Dicotyledoneae)	0.907029	21

[1] http://pinyin.sogou.com/dict/
[2] Baidu Baike (http://baike.baidu.com/) and Hudong Baike (http://www.baike.com/)

▼Table 3. Definitions of variables and functions

Variables and functions	Definition
Confidence (A, B)	The probability that B is one hypernym of A for pre-estimation accuracy.
Support (A, B)	The statistical support that B is one hypernym of A.
Count (A, B)	The number of times A and B co-occur in the same entity hypernym set.
Count (A)	The number of times A occurs in all the entity hypernym set.

one of the hypernyms of A. We consider A and B have strong association that they should be synonyms. In this situation, there are no hypernym relations between A and B.

2.1.3 Algorithm Improvement and Optimization

We use two thresholds, confidence and support, to determine whether a hypernym relation can stand only when the confidence and support reach the specific thresholds. After some experiments, we find that if the support of a hypernym relation is just a little lower than the threshold while its confidence is much higher the threshold, it should also probably be correct. For this case, we improve the algorithm and use new evaluation methods.

1) Linear optimization

For a hypernym relation that needs to be judged, we set support as x, confidence as y and the hypernym exponent as H. We use four more parameters, C_1, C_2, S_1 and S_2 to determine whether B is one of the hypernyms of A. The meaning of each parameter is shown as follows:

C_1: forward direction confidence, the number of times for A and B co-occurring divided by the numbers of times for A occurring. It is a pre-estimation whether B is one of the hypernyms of A.

C_2: backward direction confidence, the number of times for A and B co-occurring divided by the numbers of times for B occurring. It is a pre-estimation whether A is one of the hypernyms of B.

S_1: minimum support. We consider the hypernym relation disconfirmed if x is lower than $S1$.

S_2: basic support. We consider the hypernym relation confirmed if x is higher than S_2 and y is higher than C_1.

H: the hypernym exponent for B to A, which can be estimated as follows:

$$H = y - \max\left(1 - \frac{(x - S_1)(1 - C_1)}{S_2 - S_1}, C_1\right), \quad (3)$$

of which the image description is shown in **Fig. 1**.

For each hypernym relation, we firstly calculate its support x and confidence y, then check whether $x>S_2$ (above the blue line) or $S_1 \leqq x \leqq S_2$ (above the red line). If a hypernym relation satisfies the condition, H will be positive, otherwise negative.

Actually, the blue line shows a traditional evaluation method while the red line shows the newly designed in this paper.

In summary, we calculate H according to its formula and determine a hypernym relation should be reserved or not by the positive or negative result.

2) Logarithmic optimization

For each hypernym relation, we get the same x, y and H as for the linear optimization. We use three parameters, C_1, C_2, and S, to determine whether B is one of the hypernyms of A. The meaning of each parameter is shown as follows:

C_1 and C_2: the same as those for linear optimization.

S: the support threshold.

H: the hypernym exponent for B to A, which can be estimated as:

$$H = y \log(x) - C_1 \log(S). \quad (4)$$

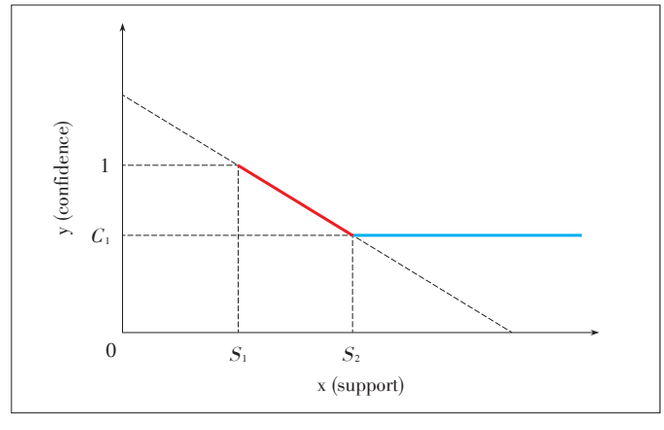
Similar to linear optimization, we calculate H according to its formula and determine a hypernym relation should be reserved or not by the positive or negative result.

2.1.4 Tongyici Cilin

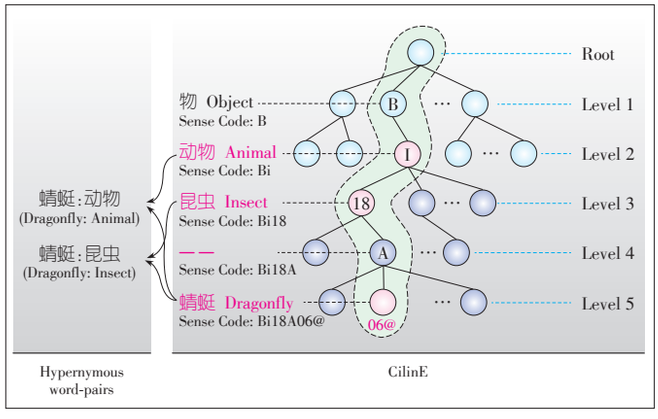
HIT-SCIR (Harbin Institute of Technology - Research Center for Social Computing and Information Retrieval) Tongyici Cilin (Extended) [9] (Cilin) is a Chinese Semantic Dictionary built by the Research Center for Social Computing and Information Retrieval in Harbin Institute of Technology. It includes 77,343 words, constructed into a 5-level hyponymy structure.

Cilin is built artificially and contains lots of commonsense hypernyms, which is complementary to the hypernym relations dug out automatically, is suitable to solve the problem that there are some hiatuses in the topmost hypernym chain [10].

There are 5 levels (not including the root) for the data in Cilin (**Fig. 2**), of which the first level has 12 categories such as "人(human)", "物 (object)", "时间与时空 (time and space)", "抽象事物 (abstract thing)", "特征 (property)", "动作 (action)", "心理活动 (mental activity)", "活动 (activity)", "现象与状态 (phenomena and state)", "关联 (relevance)", "助词 (auxiliary word)" and "敬语 (honorific)". Because we mainly process the words about entities, we keep the previous four categories from function words.



▲Figure 1. The determination for hypernym relation.



▲**Figure 2. Cilin hierarchical structure.**

We need to extract all the hypernym relations and find lots of polysemy that means one word belongs to different categories. After observing the data, we find that lots of polysemy is not accurate, such as "林肯 (Lincoln)" in "人 (human)" while also in "汽车 (automobile)", we just discover the "林肯 (Lincoln)" in "人 (human)" in data. So we pick up the polysemy for special processing in next filtration. Furthermore, there are some mistakes in some names of categories in Cilin, and we try to fix these mistakes in our experiments.

We keep the words relevant to the entities that have good quality for relation filtration, after extracting the hypernym relations in Cilin. For example, the "辈分 (generation)" in "人 (human)" and the "性能 (performance)" in "抽象事物 (abstract thing)" will be filtered.

When we are extracting the hypernyms of "哈尔滨工业大学 (Harbin Institute of Technology)", we may get "大学 (university)" and "高校 (college and university)", which are synonyms. We use the synonyms in Cilin and combine them.

2.2 Hyponymy Hierarchization Based on Suffix Hypernym

2.2.1 Problem Analysis

In hyponymy hierarchization experiment, we firstly use Apriori Algorithm to discover the hypernym relations between hypernyms by digging out the association of frequent itemset. However, Apriori Algorithm cannot discover the hypernym relations for those words occur just a few times. For this situation, we hierarchize the hypernyms based on suffix hypernym in case to obtain new hypernym relations.

We find that some hypernyms are the suffixes of their hyponyms. For example, "医院 (hospital)" is the suffix of words such as "哈工大校医院 (HIT University Hospital)", then "医院 (hospital)" is a hypernym of "哈工大校医院 (HIT University Hospital). Therefore we utilize the suffix information to discover and complete the entity hierarchical construction.

2.2.2 Suffix Hypernym

We define suffix hypernym that if word A is the suffix of word B, A is most likely to be a hypernym of B and A is the suffix hypernym. By observing the data, "运动员 (athlete)" is usually the suffix of other words such as "篮球运动员 (basketball athlete)" and "足球运动员 (football athlete)" and it is one of their hypernyms as well. Thus, "运动员 (athlete)" is a suffix hypernym.

The method, unlike Apriori Algorithm, is designed by the characteristic of Chinese that the suffix is usually the head word. The suffix hypernym occurs a lot so that we design the following steps to discover the hierarchical relations between hypernyms:

Step 1: count the frequency of each word that be the suffix of others among all the hypernyms;

Step 2: choose the words that have a high statistical frequency more than the threshold as suffix hypernym;

Step 3: do the suffix matching among hypernyms in order to obtain new hypernym relations.

2.3 Hyponymy Hierarchization Based on Classification

2.3.1 Problem Analysis

Most entities can be classified into "人 (human)", "物 (object)", "时间 (time)", "空间 (space)" and "抽象事物 (abstract thing)", which are the 5 hypernyms all from the top of Cilin. The roots of some hypernyms are not in the 5 top hypernyms, so we propose the hyponymy hierarchizing algorithm based on classification in order to put the hypernyms with no roots in the top 5 hypernyms into the 5 basic hypernyms. According to the data analysis, considerable entities do not reach "人 (human)", "物 (object)", "时间 (time)", "空间 (space)" and "抽象事物 (abstract thing)" at all. There is still a lot to do to enrich the hypernyms in the whole entity relation graph, especially the hypernym relations near the root.

With a hypernym non-polysemy assumption, we consider a hypernym belongs to "人 (human)", "物 (object)", "时间 (time)", "空间 (space)" and "抽象事物 (abstract thing)" but has no polysemy. The assumption is important and we need to analyze its correctness.

An entity may have several hypernyms, for example, "苹果 (apple)" belongs to "水果 (fruit)", "电影 (movie)" as well as "手机 (mobile phone)". So the entity "苹果 (apple)" may be polysemic. However, a hypernym, no matter it is "水果 (fruit)", "电影 (movie)" or "手机 (mobile phone)", will not be polysemic, because the hypernym itself stands for a category. Thus, the hypernym non-polysemy assumption is valid.

A hypernym in hierarchy has several fathers and children and they should also belong to one of "人 (human)", "物(object)", "时间 (time)", "空间 (space)" and "抽象事物 (abstract thing)". We actually put the hypernyms into small sets, in which the words are all belong to the same top hypernym among the 5 ones. The model is called cheat-in-exam model. First, we assume that there is an exam for the students in a class. All the students finish the exam by themselves without

copying others and get the class average score. For the second time, we set the students in groups and each group includes 4 students. Everyone in the same group is allowed to copy each other and we get another class average score. Usually, the second class average score is higher than the first one. This is the cheat-in-exam model.

2.3.2 Good Hypernym

When the confidence of the hypernym of an entity is more than 0.985, we call this hypernym a good hypernym.

We need the good hypernyms for hyperynm hierarchizing based on classification because most of the input data of hypernyms are noise.

After the hypernym non-polysemy assumption, we use the cheat-in-exam model to optimize our algorithm and also consider the calculated classification of the hypernym in the same set.

For similarity calculation, we use Backward Maximum Matching Algorithm according to the characteristics of Chinese. We use the words with their roots in the top 5 hypernyms as priori knowledge to guide other hypernyms to hierarchize.

Our algorithm use the idea of K nearest neighbors, that is, if the root of a hypernym is not in the top 5 hypernyms, we find the closed hypernyms with their roots in the top 5 hypernyms to determine its top hypernym.

We find the longest suffix of the hypernym with unknown top hypernym among the hypernyms with certain top hypernym in the top 5 and also find out the longest suffixes of its parents and children to determine the classification of the set of their own. This is the a cheat-in-exam model.

3 Experiments

3.1 Experiment Data

The data of entities and hypernyms we used are shown in **Table 4**.

3.2 Experiment of Hyponymy Hierarchization Based on Apriori Algorithm

Our statistics shows that there are 136,039 hypernyms in all 700 thousand words. We have 30,453 good hypernyms, 22.4% of all the hypernyms. Most good hypernyms occur with more than 0.985 confidence. Thus, the hypernyms with confidence lower than 0.985 probably are noise instead of hypernyms.

We adjust the confidence and support and find that the accuracy of result will be improved with the increase of confidence but the number of hypernym relations decreased. It is the same for the adjustment of support.

▼Table 4. The data of entities and hypernyms

The number of entities	The number of hypernyms	The average number of hypernyms for each entity
745,620	9,010,192	12.1

Using Apriori Algorithm to discover the association of frequent itemset, we obtained 8327 hypernym relations between hypernyms. **Table 5** shows the parameters setup for this experiment.

There are some indirect edges in the 8327 hypernym relations. For example, "被子植物 (angiosperm)" belongs to "植物(plant)" and "生物 (living thing)" while "植物 (plant)" belongs to "生物 (living thing)", so we can get the hypernym relation that "被子植物 (angiosperm)" belongs to "生物 (living thing)" because "植物 (plant)" belongs to "生物 (living thing)". We filter such redundant relations such as relation A→C while A→B and B→C exist.

There are 5422 relations reserved after filtering indirect edges of the original 8327 relations. We randomly picked up 200 relations of 5422 relations for manual evaluation and the precision is 97.0%.

The experiment results (**Table 6**) are basically consistent with expectations, while the result of linear optimization is the best with high precision and many hypernym relations.

3.3 Experiment of Hyponymy Hierarchization Based on suffix hypernym

We obtained 8747 hypernym relations by hyponymy hierarchization based on suffix hypernym. We did two more steps for these relations: filtering the indirect edges and duplicating relations.

There are 7503 reserved after pre-processing of the original 8747 hypernym relations. We randomly picked up 300 relations of 7503 relations for manual evaluation and the precision is 96.7% with 290 correct relations.

As shown in **Table 7**, we obtained a large number of hypernym relations between hypernyms by discovering the suffix hypernyms and using them to hierarchize the hypernyms. However, these relations are usually limited in some domains and have limited forms. We chose the two-character-suffix for high performance instead of one-character-hypernyms such as "人

▼Table 5. Parameters setup

Parameter	Definition	Value
C_1	Forward direction confidence	0.9
C_2	Backward direction confidence	0.8
S_1	Minimum support	5.0
S_2	Basic support	10.0

▼Table 6. Experiment results of hyponymy hierarchization based on Apriori Algorithm

	Precision	The relation number (no indirect edge)	Percentage of number increase
No optimization	97.0% (±1%)	3396	-
Linear optimization	97.0% (±1%)	5422	60.0%
Logarithmic optimization	84.5% (±1%)	4073	19.9%

(human)" and "物 (thing)", although "人 (human)" and "物 (thing)" are also hypernyms of many entities.

3.4 Experiment of Hyponymy Hierarchization Based on Classification

Most Chinese words have two characters or more. A Chinese word and its last character probably have different meanings such as "亚洲地理 (Asian geography)" and "理 (idea)". According to the data, the precision reaches about 80% even if just one character is matched during the process of backward maximum matching of most hypernyms. A higher precision will be reached when backward maximum matching is used for two or more characters. As for matching more than two characters, the result shows low performance that only 6.26% can be classified for good hypernyms and 3.30% for all hypernyms, even when the cheat-in-exam model is used.

We use the words with two or more characters backward maximum matched as closed words and 44.33% of good hypernyms are able to be classified.

After the experiments above, we classified 5119 hypernyms out of 11,547 unclassified good hypernyms. We randomly picked up 200 relations for manual evaluation and 187 of them are correct with the precision of 93.5%.

In summary, we obtained 5119 hypernym relations with the precision of 93.5% from good hypernyms, and 19,970 hypernym relations from all hypernyms.

The experiment results (**Table 8**) show that the classification reorganizes the entire entity relation graph, especially the part closed to the root. Although the hyponymy hierarchization based on classification discovers lots of hypernym relations, which are all closed to the root, the information provided is relatively limited.

Hyponymy hierarchization based on Apriori Algorithm hierarchizes hypernyms by using the associations between hypernyms; hyponymy hierarchization based on suffix hypernym hierarchizes hypernyms by using the suffix information; and hyponymy hierarchization based on classification hierarchizes hypernyms also by using the suffix information to complete the hypernym hierarchical structure. These three methods hierarchize the entity hypernyms in three different ways and the experiment results show that they all have high precision and obtain good results.

▼Table 7. Experiment results of hyponymy hierarchization based on suffix hypernym

Precision	The relation number	The relation number after filtering indirect edges and duplicating relations
96.7%	8747	7503

▼Table 8. Experiment results of hyponymy hierarchization based on classification

Precision	The relation number	The number of good hypernyms
93.5%	19,970	5119

4 Conclusions

We obtained a large number of entities and their hypernyms by extracting the data from the Internet, and then constructed the hierarchical structure of hypernyms based on Apriori Algorithm, suffix hypernym and classification and completed the hypernym hierarchical structure. We achieved good experiment results with high precision.

Acknowledgement

We thank the anonymous reviewers for their helpful comments.

References

[1] Q. Tang, F. Gao, and J. Wang, "Knowledge map concepts analysis and research," *Information Studies: Theory and Application*, vol. 34, no.1, pp. 121–125, 2011.
[2] L. Qian, G. Zhou, F. Kong, Q. Zhu, and P. Qian, "Exploiting constituent dependencies for tree kernel-based semantic relation extraction," in *Proc. 22nd International Conference on Computational Linguistics*, Manchester, UK, Aug. 2008, pp. 697–704.
[3] A. Fader, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proc. Conference on Empirical Methods in Natural Language Processing*, Edinburgh, UK, Jul. 2011, pp. 1535–1545.
[4] M. Banko, M. J. Cafarella, S. Soderland, M. Broadhead, amd O. Etzioni, "Open information extraction from the web," in *20th International Joint Conference on Artificial Intelligence*, Hyderabad, India, Jan. 2007, pp. 2670–2676.
[5] W. Che, T. Liu, and S. Li, "Automatic entity relation extraction," in *The First National Conference on Information Retrieval and Content Security (NCIRCS' 2004)*, Shanghai, China, 2004, 1–6.
[6] C. Chen, Y. Chen, J. Hou, and Y. Liang, "CiteSpace II: detecting and visualizing emerging trends and transient patterns in scientific literature," *Journal of the China Society for Scientific and Technical Information* vol. 28, no. 3, pp. 401–421, 2009.
[7] R. Fu, B. Qin, and T. Liu, "Exploiting multiple sources for open-domain hypernym discovery," in *Proc. 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Seattle, USA, 2013, pp. 1224–1234.
[8] R. Fu, "Open-domain named entity recognition and hierarchical category acquisition," Ph.D. dissertation, Harbin Institute of Technology, Harbin, China, 2014.
[9] W. Che, Z. Li, and T. Liu, "LTP: a Chinese language technology platform," in *Proc. 23rd International Conference on Computational Linguistics, Demonstrations Volume*, Beijing, China, Aug. 2010, pp. 13–16.
[10] C. Friedman, G. Hripcsak, W. DuMouchel, et al., "Natural language processing in an operational clinical information system," *Natural Language Engineering*, vol. 1, no. 1, pp. 83–108, 1995.

Manuscript received: 2015-11-05

Biographies

CAI Hongbo (hbcai@ir.hit.edu.cn) received his master's degree from School of Computer Science and Technology, Harbin Institute of Technology, China. He is an engineer at Alibaba Corporation. His main research interest is entity relation graphs.

CHEN Hong (chen.hong3@zte.com.cn) received her B.S. degree from the Department of Information Engineering, Nangjing University of Posts and Telecommunications, China in 2007. She is a senior research engineer at ZTE Corporation. Her research interests incloud social network analysis and intelligent question answering. She holds five patents .

LIU Shen (sliu@ir.hit.edu.cn) received his master's degree from School of Computer Science and Technology, Harbin Institute of Technology, China. He is an engineer at Miaozhen System. His main research interest is entity relation extraction.