Conceptualization of Entity Relationship Based on Knowledge Graph

Yang Yu, Youlang Ji, Jun Zhu, Hongying Zhao, and Jingjing Gu

***Abstract*—As numbers of high-quality, large volume knowledge graphs appear, information extraction work has been enriched with more semantic knowledge. However, the entity relation extraction based on the knowledge graph is still at a very intuitive early stage, and the key issue it faces is the relation recognition and classification. In order to break the shackles of ontology and describe the relationship between the entities with fine-grained types, we propose a two-step bottom-up abstraction approach for relation conceptualization based on conceptual taxonomy that is automatically constructed. Given an entity relation, we figure out a group of Top-K concept pairs to abstract the relation, according to the typicality, diversity and coverage features. Our experimental evaluation shows that our method performing significantly high precision and quality for detecting fine-grained relationships.**

***Index Terms*—Knowledge graph, entity relationship, conceptualization, clustering.**

I. Introduction

There are a large number of unstructured or semi-structured texts on the Internet. A mainstream view is to translate these non-structured semi-structured texts into structured semantic information. This work is called information extraction including Entity Extraction, Relation Extraction [1] and Event Extraction. With the emergence of some high-quality and bulky Knowledge Graphs such as DBpedia, Freebase, YAGO, Probase, etc., there is a new research direction in information extraction. These knowledge maps contain automatically constructed data such as Entity, Concept, Semantic Relation. As a background knowledge, these data can be fully utilized in the process of information extraction: the structured information extracted in the past will be related to the knowledge map, and the relation of the knowledge map itself will build a large amount of extracted information into a larger structure , Thus promoting new information extraction.

Entity relationship extraction is extracted from the text, including two kinds of data extraction: (1) the relationship model: A set of entities that have a relationship usually appears in the text when combined with some of the more frequently occurring contexts, which is called relational patterns. Such as "Arg1 locates in Arg2", "Arg1 acts as Arg3 for Arg2" and etc. Placeholder Arg # in relational mode represents an entity in the relational schema, and the entity's category needs to be qualified. So a complete relational schema should look like "<Person> write a song <Song>". Some of the more advanced relational patterns include equivalent statements and thus represent a type of relational pattern, such as "<Person> write a [adj] song <Song>". (2) Relationship instance. A relationship instance refers to a set of entities that have a relationship that corresponds to a relationship pattern. As you can see, relational patterns and relational instances are mutually reinforcing. Relational patterns can be used to extract relational instances, and relational instances can also be used to discover relational patterns from text. Usually this paper said the relationship between the entity extraction including the relationship model and relationship instance.

Manuscript received July 7, 2018; revised September 1, 2018.

Yang Yu, Youlang Ji, Jun Zhu, Hongying Zhao are with the Jiangsu Electric Power Company, GaoYou County Electric Power Supply, Company Gaoyou 225600, China (e-mail: 642826764@qq.com).

Jingjing Gu is with the Jiangsu Electric Power Company, Jurong County Electric Power Supply Company, Jurong 212400, China.

II. Related Work

Early relationships were extracted by manually defined relationship categories. Rosario and Hearst (2001) classify the relationships in the pharmaceutical field into 13 categories [2]. Stephens *et al.* (2001) extracted the genetic relationships into 17 specific classes [3]; Nastase and Szpakowicz (2003) The relational structure consists of five classes in the first layer and 30 classes in the second layer for the extraction of nouns modifiers [4]. A great deal of work (Kim and Baldwin, 2005 [5]; Nakov and Hearst, 2008 [6]; Nastase *et al.*, 2006 [7]; Turney and Littman, 2005 [8]) are based on a specific area or common sense relationship classification.

The knowledge base has a rich taxonomy of named entities. Most of the knowledge-based relational extraction systems use unsupervised or semi-supervised methods. The well-known knowledge base-based relationship extraction includes: TextRunner / ReVerb (Banko 2007 [9]; Fader 2011 [10]), NELL (Carlson 2010 [11]; Mohamed11 [12] Dynamic vocabulary extraction (Hoffmann 2010 [13]), LDA clustering (Yao 2011 [14]), PATTY (Nakashole 2012 [15]; Nakashole 2013 [16]).

Knowledge-based information extraction systems use semi-supervised or bootstrapping which need reliable seed instances including entity instances and relational instances. But some small amounts of human-tagged data often contain some untrusted of the data, and these seeds will affect some of the columns after the iterative extraction process. To overcome this problem, scientists at Carnegie Mellon University's Department of Machine Learning proposed the Coupling Semi-Supervised Method [17] and in 2010 [11] introduced the NELL (Never Ending Language Learning) system.

doi: 10.18178/ijmlc.2018.8.5.730

III. CONCEPTUALIZATION OF ENTITY RELATIONSHIP

A. Definition

Concept-Entity: In the concept classification system, concepts are abstract representations of entities, and entities are concrete examples of concepts. The relationship between an entity and a concept is represented by the isA edge. Such as isA (apple, company), isA (apple, fruit). The subgraph consisting of isA edges in knowledge map is a directed acyclic graph. The biggest existing concept classification system is Probase released by Microsoft in 2012, whose the number of concepts covered is the largest (about 2.6M) in all the knowledge maps at present, and its isA relationship is based on probability.

Entity relationship: Entity relationship is actually an abstract representation of many entity pairs. This paper argues that a set of concept pairs abstracted from many entities can be used to represent a relationship.

B. Problem Description

Given a knowledge map G , where the node is the entity or concept e , the edge is $x(e_i, e_j)$, which means that there is a side of the relationship r between the entities e_i and e_j . For a binary relation r, there is a set of entity pairs $E(r)=\{(e_i,e_j)|(e_i,e_j)\in E(G)\wedge r(e_i,e_j)\}$ in the knowledge graph G. We refer to the conceptual pair set $CP(r)$ for the entity after conceptualization in $E(r)$. Obviously, this paper hopes $CP(r)$ as small as enough to describe a relationship r. Each concept in $CP(r)$ is sufficiently typical to be able to abstract a subset of $E(r)$, while all concepts have sufficiently high coverage of the population, that is, diversity should be sufficiently rich.

Here we need to explain the meaning of typicality and coverage. Typicality refers to the semantics of this concept that can not be too broad nor too rare when the article uses a concept to abstractly describe an entity. It is easy to see that the typicality and coverage are contradictory. This article needs a compromise solution to weigh the advantages and disadvantages.

This article can be roughly divided into three objectives:

1) If the goal is to generate one concept pair from n entity pairs, the generated concepts will be categorized according to relationships that will depend on the isA relationship. For example, the relation r_1 is equivalent to the concept pair (c_{11}, c_{12}) and the relationship r_2 is equivalent to the concept pair (c_{21}, c_{22}). If the concept satisfy $c_{11} \preccurlyeq c_{21}$ and $c_{12} \preccurlyeq c_{22}$, $(c_{11}, c_{12}) \preccurlyeq (c_{21}, c_{22})$, that is, r_1 is a sub-relationship of r_2. What this goal requires is a common compromise where the concept is as representative as possible with the highest guaranteed coverage.

2) If the goal is to generate m concept pairs by n pairs of entities. Then the generated concepts have implications for the existence of child relationships and thus naturally establish a relational classification system, and new relationships can also be found from m concept pairs. This goal requires a compromise between compromises because there is bias and noise for the dataset of the knowledge map, and there may be bias and interference terms for m concepts that meet the high coverage. The offset is result from data imbalance or incompleteness, and noise is a somewhat invalid concept pair, as explained in more detail in Section 2.1.3. Noise can be filtered by setting a threshold, which is inherently difficult to control due to insufficient signal strength. In order to make the confidence of the relational classification system higher, the generated m concepts should be as accurate as possible, so the goal of the compromise should be the opposite of the goal of a program, that is, the concept of the relationship generated by the concept should give priority to the high typical, While the coverage of the second.

3) If the goal is to establish a complete relationship classification system, then the results of the above two goals can be combined to establish a relationship classification system, while the relationship can be automatically completed.

The third goal, the second goal is the most critical, from the second goal can be deduced the first, and then derive a third goal.

C. Algorithm Design

1) Concept pair sorting

Given a relationship r and knowledge graph G, we need to generate all pairs of concepts, and then select the candidate pairs of concepts, and finally rank the candidate concepts according to their typicality.

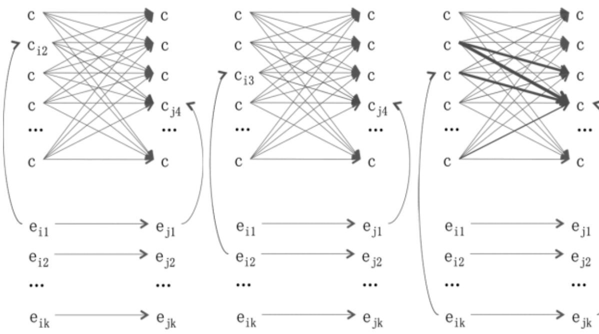

Fig. 1. Cumulative weight iteration process.

First, Cartesian product is calculated for all concepts C in the knowledge graph G, and then generate all the concept pairs $C\times C$. The edge set $E(G)$ is retrieved to obtain the set $E(r)$ with relation r:

$$C\times C=\begin{pmatrix}(c_1,c_1) & (c_1,c_2) & \cdots & (c_1,c_k)\\ (c_2,c_1) & (c_2,c_2) & \cdots & (c_2,c_k)\\ \vdots & \vdots & \ddots & \vdots\\ (c_k,c_1) & (c_k,c_2) & \cdots & (c_k,c_k)\end{pmatrix} E(r)=\begin{pmatrix}(e_{i1},e_{j1})\\ (e_{i2},e_{j2})\\ \\ (e_{ik},e_{jk})\end{pmatrix} \tag{1}$$

Next, we will calculate the typicality of each candidate concept pair, and then sort the candidate concept pairs according to the typicality. $C\times C$ in fact constitutes a bipartite graph which each point represents a concept, as shown in Fig. 1. The edge (c_i, c_j) represents a concept pair, and the edge weight represents the typicality of a conceptual pair. At initialization, the weights of the edges are all zero. Then we traverse each pair (e_i, e_j) in $E(r)$ and add weights to different edges (c_i, c_j) by calculating a typical function $f((c_i,c_j),(e_i,e_j))$. So all The edge will have a cumulative weight. Figuratively,

each pair of entities has "voting rights" for (e_i, e_j), "votes" for (c_i, c_j) for the concept of "preference," and when all entities vote for the end, each concept pair get a "total score of votes."

The cumulative weight is calculated as:

$$w(c_i, c_j) = \sum_{(e_i,e_j)\in E(r)} f\left((c_i, c_j),(e_i, e_j)\right) \quad (2)$$

It needs to be discussed that there are many ways to choose a typical function *f*.

Typical functions defined from the perspective of average accumulation:

$$f\left((c_i, c_j),(e_i, e_j)\right) = \begin{cases} 1 & isA(e_i, c_i) \wedge isA(e_j, c_j) \\ 0 & otherwise \end{cases} \quad (3)$$

This definition assumes that each pair of pairs of concepts generated by the entity is equal. This is the simplest way to define it. Its typicality depends entirely on the summation of $E(r)$, so the effect of its cumulative result depends on the size of $E(r)$. If $E(r)$ is too small, its accumulation the more powerful edges are often the more "generic" concept than the classic concept.

Typical functions defined from the point of frequency accumulation:

$$f\left((c_i, c_j),(e_i, e_j)\right) = n(e_i, c_i)\cdot n(e_j, c_j) \quad (4)$$

Defining typical functions from the perspective of probability accumulation:

$$f\left((c_i, c_j),(e_i, e_j)\right) = p(c_i|e_i)\cdot p(c_j|e_j) \quad (5)$$

where

$$p(c_i|e_i) = \frac{n(e_i, c_i)}{\sum_{isA(e_i,c_j)} n(e_i, c_j)} \quad (6)$$

Based on the above considerations, the algorithm in this paper uses PCW as a definition of a canonical function. The final PCW edge weight function is:

$$w(c_i, c_j) = \frac{1}{Z}\sum_{(e_i,e_j)\in E(r)} f\left((c_i, c_j),(e_i, e_j)\right) \quad (7)$$

where Z is the normalization constant,

$$Z = \sum_{(c_i,c_j)\in C\times C}\sum_{(e_i,e_j)\in E(r)} f\left((c_i, c_j),(e_i, e_j)\right) \quad (8)$$

In fact, in the process of actual algorithm compilation, we take into account the high spatial complexity of $C\times C$, but the time complexity of traversing $E(r)$ is too high. So we need to sample and use the pruning strategy and greedy algorithm.

2) Clustering compression

After the first concept of sorting algorithms, the number of candidate pairs of concepts has been greatly reduced. This article hopes to get a diverse set of concepts for the collection to describe a relationship, which can be based on the diversity of the concept of this article to amend the sort.

Due to the concept has a certain degree of typicality, this article does not consider the weight problem in the calculation of shared entities. In addition, we assume that a concept pair that is more pan in the knowledge graph and also more generic in a subset $E(r)$. Therefore, we define an entity pair set $EP_r(c_i, c_j)$ corresponding to a concept pair (c_i, c_j) under relation *r*:

$$EP_r(c_i, c_j) = \left\{(e_i, e_j)\middle| isA(e_i, c_i)\wedge isA(e_j, c_j)\wedge(e_i, e_j)\in E(r)\right\} \quad (9)$$

In this paper, we define that the similarity between two pairs of concept pairs *(c_i, c_j)* and *(c_k, c_l)* is the Jaccard distance of the corresponding entity to the set under the relation *r*:

$$J_r\left((c_i, c_j),(c_k, c_l)\right) = \frac{\left|EP_r(c_i, c_j)\cap EP_r(c_k, c_l)\right|}{\left|EP_r(c_i, c_j)\cup EP_r(c_k, c_l)\right|} \quad (10)$$

This article may wish to creatively consider $C(r)$ as an undirected graph, each node is a conceptual pair, the weight of the node is the PCW value, each edge represents the similarity between concept pairs, the edge weight is Jaccard distance.

Obviously, this article should choose a clustering method and has the following characteristics:

- Undifferentiated sparse graph with edge weights
- The number of clusters is unknown in advance
- Able to adapt to a variety of shapes
- Strong noise robustness

Therefore, this paper chooses Markov Clustering Algorithm (Dongen 2000 [18], [19]). This algorithm mainly uses the random walk thought on the probability graph to iteratively multiply the matrix to converge. Finally, Degree segmentation cluster. Algorithm is as follows:

Algorithm 1: Markov Clustering Algorithm

Input: undirected graph G, power parameter e, expansion coefficient *r*

1. Generate probabilistic adjacency matrix M from G.
2. Add a self-loop for each node in M (optional)
3. Normalize matrix M
4. Expand the matrix with e power:
   $M:=(M)^e$
5. According to the expansion coefficient r matrix expansion by column (Inflation):
   For each column i of M
   $M(:,i):= (M(:,i))^r$
   $M(:,i):= M(:,i)/Sum(M(:,i))$
6. Repeat steps 4 and 5 until the matrix converges
7. According to connectivity segmentation matrix to get the cluster

Through MCL clustering, concept pairs in C (r) are divided into multiple clusters. Each cluster represents a kind of concept pair, and all concept pairs in the cluster have potential common semantics. This potential common semantic abstraction is abstraction of a group of entities, that is, the entity relationship.

IV. Experiments

A. Data Processing

Because the algorithm designed in this paper is based on binary relations, this paper chooses ObjectProperty in

DBpedia as a data set of entity relations. Each object attribute has a large number of entity pairs in DBpedia. The concept of classification system chose the current number of concepts Probase. This paper takes the intersection of DBpedia's entities and Probase's entities.

B. The Accuracy of Domain and Range

The entity relationship of the experimental comes from the object properties of DBpedia. Each object property has a corresponding domain and a range of entity types for defining object properties. Some domains or scopes of object properties are global.

This article randomly selected 30 object properties for domain and range analysis. Each group of entities can be compressed into a set of concept pairs after two-step algorithm, and each pair of concepts in each group corresponds to a cluster in the second-step clustering algorithm. This paper examines the partial ordering of (domain, range) for each pair of concepts (c_i, c_j) for each cluster in a relation r:

$$v_r(c_i, c_j) = \begin{cases} 1 & c_i \prec= \text{domain}(r) \wedge c_j \prec= \text{range}(r) \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

This paper calculates the accuracy of all concept pairs in a cluster:

$$p(r) = \frac{1}{|MCP(r)|} \sum_{C \in MCP(r)} \frac{1}{|C|} \sum_{(c_i, c_j) \in C} v_r(c_i, c_j) \quad (12)$$

In addition, this article also calculated the accuracy of each cluster center (m_i, m_j):

$$p^m(r) = \frac{1}{|MCP(r)|} \sum_{C \in MCP(r)} \frac{1}{|C|} v_r(m_i, m_j) \quad (13)$$

The experimental results are shown as following. Top-K indicates the first K cluster centers (the number of clusters with some relations is less than K) of the final output of the algorithm, and x / y means that x of y cluster centers is correct.

Relation	Domain	Range	Precision	Top-3	Top-1
notableIdea	Person	#	100.00%	1/1	1/1
influencedBy	Person	Person	100.00%	2/2	1/1
deathPlace	Person	Place	100.00%	1/1	1/1
knownFor	Person	#	100.00%	1/1	1/1
leader	#	Person	100.00%	1/1	1/1
president	#	Person	100.00%	2/2	1/1
influenced	Person	Person	97.53%	2/2	1/1
location	#	Place	97.30%	1/1	1/1
birthPlace	Person	Place	96.25%	1/1	1/1
nationality	Person	Country	88.30%	2/3	1/1
artist	MusicalWork	Agent	88.24%	2/3	1/1
country	#	Country	87.91%	1/1	1/1
writer	Work	Person	87.80%	3/3	1/1
product	Organisation	#	78.95%	1/1	1/1
director	Film	Person	78.21%	1/1	1/1
family	Species	Species	76.47%	2/3	1/1
genre	#	Genre	74.12%	2/3	1/1
author	Work	Person	71.88%	2/2	1/1
album	#	Album	68.42%	2/2	1/1
Average	#	#	**89.02%**	**92.98%**	**100%**

Fig. 2. Compares DBpedia domain-range experiments.

From Fig. 2, the experimental results show that the accuracy of the entity relationships found by the proposed algorithm reaches 89.02% in all generated concept pairs, 92% in Top-3 cases and 100% in Top-1 cases. In addition, we can see from the experimental results that there are still some enlarged or incorrect concept pairs in a clustered cluster, which also confirms why we need to select the appropriate cluster center concept pair.

C. Clustering Effect

The essence of the second-step clustering algorithm is to discover different sub-relationships under the same relationship. Therefore, this paper designs experiments to evaluate whether clustering can discover a set of concepts from different pairs of entities.

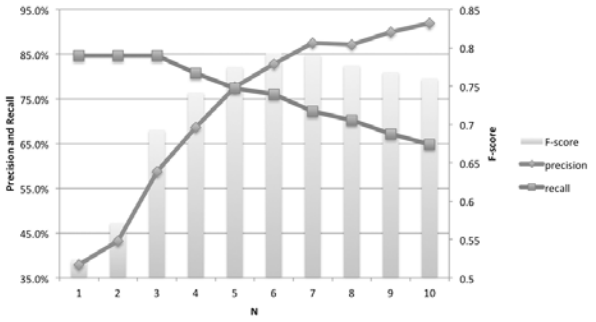

Fig. 3. Experiment with changing cluster size N.

As shown in Fig. 3, this article mentions that larger clusters are more reliable and of higher quality because of the greater number of their signals. Therefore, according to different size as a criterion for selecting clusters, this paper proposes the concept of combination relation *rc*. For the set, the paper compares the concept of cluster size larger than N in each combination relation *rc* with the concept pairs of all relations *ra*, if the similarity high, then return the combination *rc* contains a relationship *ra*.

D. Quality Assessment

The final output of the algorithm is the central concept pair of a group of Top-K clusters. In order to evaluate whether the concept given in the end is of high quality, and evaluate the accuracy and the recall rate of the algorithm, we randomly selected 30 groups to do manual scoring and used Mean Mean Precision (MAP) Average quality and random quality.

Each relationship provides 10 sets of concept pairs. The first few generated by the algorithm for at most K pairs of concepts, and the rest of the concept pair is generated according to the relationship between pairs of random concepts. Each relationship was awarded 10 scores for each relationship (3: good, 2: medium, 1: bad, 0: not relevant). Fig. 4 is the relationship between the score of the writer.

Relation	Concept	Mean
writer		
Algorithm generation	(song, artist)	3
	(film, director)	2.2
	(book, writer)	3
Random generation	(descriptive title, mccarthy-blacklisted u.s. writer)	0.7
	(song, entertainer)	1.4
	(scent, music icon)	0.3
	(song, guest)	0.5
	(day-to-day issue, film personality)	0.2
	(song, lady musician)	1.4
	(neo-realist film, director)	1.4

Fig. 4. 'Writer' relationship scoring.

The following Fig. 5 is the scoring of all relations:

Relation	MAP	Random MAP	Quality	Random Quality
notableIdea	100%	11%	2.70	0.77
influencedBy	100%	63%	2.85	1.35
influenced	100%	38%	3.00	1.11
country	100%	0%	2.70	0.34
birthPlace	100%	11%	2.90	0.60
deathPlace	100%	0%	2.70	0.25
party	100%	43%	2.70	1.34
director	100%	44%	3.00	1.04
writer	100%	0%	2.73	0.84
title	100%	0%	2.23	0.64
nationality	67%	0%	2.70	0.69
knownFor	100%	22%	2.60	0.96
officialLanguage	100%	0%	3.00	0.47
author	50%	25%	1.50	1.13
ethnicGroup	100%	0%	2.30	0.54
family	100%	29%	2.33	0.98
place	100%	11%	2.40	0.90
developer	100%	13%	2.70	0.75
genre	67%	14%	2.37	0.77
spokenIn	100%	11%	2.70	0.80
foundedBy	100%	11%	2.80	0.82
location	100%	0%	2.70	0.79
city	67%	0%	1.83	0.66
leader	100%	22%	2.60	0.76
team	100%	22%	2.90	0.84
president	100%	13%	2.65	0.71
product	100%	11%	2.90	0.83
position	100%	0%	2.00	0.50
artist	33%	14%	1.47	0.93
album	50%	0%	1.90	0.59
Average	**91%**	**14%**	**2.53**	**0.79**

Fig. 5. Relationship scoring accuracy and quality.

MAP means the average accuracy. The accuracy is equivalent to the relevance, that is, the concept given by the algorithm is irrelevant to the other if the score is good or moderate. Random MAP represents the average accuracy of randomly generated concept pairs. Experiments show that the accuracy of the algorithm generated by the concept pair is as high as 91%, far exceeding the 14% accuracy rate of the randomly generated concept. Quality represents the average score of concept pairs given by each relational algorithm, and Random quality represents the average score of randomly generated concept pairs. Experiments show that the quality of concept pairs generated by the algorithm is much higher than the quality of random generation.

V. Conclusion

In the information extraction, the entity relationship extraction depends on the construction of the relationship classification system. So the identification of the entity relationship and the description of the features are very important.

In this paper, concept classification system is applied to propose a two-step abstract bottom-up relationship conceptualization method. According to the concept of typicality, diversity, coverage and other characteristics as entity relationship, a set of Top-K concept pairs are given. The algorithm proposed in this paper is characterized by: 1) Use a richer conceptual classification system of entities to discover finer granular relationships; 2) Consider a pair of entities or pairs of concepts as an object, preserving the potential for a pair of entities or concepts Entity relationships. A more typical pair of concepts is chosen through the overlay optimization of a large number of pairs of entities. The advantage of the algorithm is that it is not limited to a pair of coarse-grained entity types, but can generate entity relations with finer granularity by bottom-up abstraction of entity pairs. This helps to build a semantic-based relational classification system, and also helps to discover new relationships between finer types of entities.

Experiments show that the proposed algorithm can find finer granular entity relationships of an entity type, and can also separate relations with higher accuracy from a composite entity relationship. Meanwhile, the concept generated by this algorithm for describing relationships is more accurate than that of High quality.

References

[1] L. Taesung *et al.*, "Attribute extraction and scoring: A probabilistic approach," in *Proc. International Conference on Data Engineering (ICDE)*, 2013.

[2] R. Barbara and M. Hearst, "Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy," in *Proc. the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-01)*, 2001.

[3] J. S. Matthew, "Detecting gene relations from medline abstracts," *Pacific Symposium on Biocomputing*, vol. 6, 2001.

[4] N. Vivi and S. Szpakowicz, "Exploring noun-modifier semantic relations," in *Proc. 5th International Workshop on Computational Semantics (IWCS-5)*, 2003.

[5] K. Su Nam and T. Baldwin, "Automatic interpretation of noun compounds using WordNet similarity," *Natural Language Processing*, Springer Berlin Heidelberg, pp. 945-956, 2005.

[6] N. Preslav and M. A. Hearst, "Solving relational similarity problems using the web as a corpus," *ACL*., 2008.

[7] N. Vivi *et al.*, "Learning noun-modifier semantic relations with corpus-based and WordNet-based features," in *Proc. the National Conference on Artificial Intelligence*, vol. 21, no. 1, 2006.

[8] D. T. Peter and L. Michael, "Corpus-based learning of analogies and semantic relations," *Machine Learning*, vol. 60, no. 1-3, pp. 251-278, 2005.

[9] Y. Alexander *et al.*, "Textrunner: Open information extraction on the web," in *Proc. Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. Association for Computational Linguistics*, 2007.

[10] F. Anthony, S. Soderland, and O. Etzioni, "Identifying relations for open information extraction," in *Proc. the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, 2011.

[11] C. Andrew *et al.*, "Toward an architecture for never-ending language learning," *AAAI*, vol. 5, 2010.

[12] P. M. Thahir, R. Estevam, J. Hruschka, and M. Tom, "Discovering relations between noun categories," in *Proc. the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, 2011.

[13] H. Raphael *et al.*, "Knowledge-based weak supervision for information extraction of overlapping relations," in *Proc. the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.

[14] Y. Limin *et al.*, "Structured relation discovery using generative models," in *Proc. the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics*, 2011.

[15] N. Ndapandula, G. Weikum, and F. Suchanek. "Patty: A taxonomy of relational patterns with semantic types," in *Proc. the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Association for Computational Linguistics*, 2012.

[16] N. Ndapandula, T. Tylenda, and G. Weikum, "Fine-grained semantic typing of emerging entities," *ACL*, vol. 1, 2013.

[17] C. Andrew *et al.*, "Coupling semi-supervised learning of categories and relations," in *Proc. the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing. Association for Computational Linguistics*, 2009.

[18] V. Dongen and S. Marinus, "Graph clustering by flow simulation," 2001.

[19] V. D. Stijn, "A cluster algorithm for graphs," Report-Information systems10, pp. 1-40, 2000.

Yang Yu was born in 1986 at Jiangsu, China. He is a master and engineer. He has been engaged in power marketing customer service management, service quality evaluation analysis and power demand side management for a long time.

Youlang Ji was born in 1975 at Jiangsu, China. He is a master and senior engineer. He has been engaged in power marketing management for a long time. He is good at big data analysis and deep mining, familiar with power customer service management and research.

Jun Zhu was born in 1988 at Jiangsu, China. She is a master and engineer. She has been engaged in power marketing customer service management and research for a long time. She is now focusing on power knowledge base research and construction.

Hongying Zhao was born in 1989 at Henan, China. She is a master and engineer. She has been engaged in power marketing customer service management and research for a long time. She is now specializing in omni-channel power customer big data hotspot analysis and research.

Jingjing Gu was born in 1989 at Jiangsu, China. He is a master. He has engaged in first-line power marketing major customer service, research direction is the quality control of power marketing business.