Mining the Web for Association Similarity between Concepts

Li Chen
Institute of Command Automation, PLA University of Science and Technology, Nanjing, China
Email: ivan.chen@163.com

Zi-lin Song, Ying Zhang
Institute of Command Automation, PLA University of Science and Technology, Nanjing, China

Ning Zhou
China Electronic Systems Engineering Company, Beijing, China

***Abstract*—Measures of similarity between two terms or concepts have been widely used in the domain of Natural Language Processing, Semantic Web, and so on. There are mainly two kinds of methods for measuring similarity. One is based on prior manually built taxonomy or Ontology; the other which is usually referred to as the statistical approaches is based on the corpus. However, the ontology-based method has problem of coverage and the corpus-based method has the problem of sparse data. In order to overcome these problems, a huge data source World Wide Web was used to calculate similarity between concepts. The concept similarity was measured using the association rule mining in the snippets returned from Web search engines. The most influential algorithm for association rule mining is Apriori. In order to improve the efficiency of Apriori algorithm and use it to measure the concept similarity, there are three main improvements in Apriori algorithm. The experimental result shows that the algorithm can improve the precise of measuring concept similarity.**

***Index Terms*—concept similarity; snippets; association similarity; improved Apriori algorithm**

I. INTRODUCTION

Measures of similarity between two terms or concepts have been widely used in the domain of Natural Language Processing, such as word-sense disambiguation [1], language modeling [2], synonym extraction [3], and automatic keyphrase extraction [4]. Concept similarity has also been used in Semantic Web related applications such as automatic annotation of Web pages [5], community mining [6], keyword extraction [7] and so on.

The methods for measuring similarity can be divided into two main categories: one is based on prior hand-crafted knowledge (Ontology) or some kinds of classification system (Taxonomy), for example, WordNet [8] or BRICO [9] which are widely used; the other is the use of corpus statistics.

Several approaches of the first category have been proposed in the use of WordNet for similarity measurement. These approaches can be grouped into edge-based measures which are a natural and direct way of evaluating semantic similarity in taxonomy. Edge-based measures consider the length of the paths that links the words, as well as the positions of words in the taxonomic structure [10, 11], Information Content measures which find the difference of the contextual information between words as a function of their occurrence probability with respect to a corpus [12, 13]. Hybrid methods combine synsets with word neighborhoods and other features [14, 15].These approaches, although enough accurate, face a coverage problem since similarity can only be measured between concepts which appear in the ontology or taxonomy.

For the second category, most approaches share a common assumption: similar concepts have similar distributional behavior in a corpus. In this category, we can find co-occurrence based measures [16] and context-based measures [17, 18]. Typically corpus-based methods suffer from the sparse data problem: they perform poorly when the words are relatively rare, due to the scarcity of data.

Nowadays, with the rapid development of Internet technology, regarding the World Wide Web as a large and real-time data source has become an active research topic. Web search engines provide an efficient interface to analyze vastly numerous documents in the web. Page counts and snippets are two useful information sources provided by most Web search engines.

Matsuo et al., [19] proposed the use of Web hits for extracting communities on the Web. They measured the association between two personal names using the overlap (Simpson) coefficient, which is calculated based on the number of Web hits for each individual name and their conjunction (i.e., AND query of the two names).

Chen et al., [20] proposed a double-checking model using text snippets returned by a Web search engine to compute semantic similarity between words. For two words P and Q, they collect snippets for each word from a Web search engine. Then they count the occurrences of word P in the snippets for word Q and the occurrences of

doi:10.4304/jsw.7.5.1006-1013

word Q in the snippets for word P. These values are combined nonlinearly to compute the similarity between P and Q. This method depends heavily on the search engine's ranking algorithm. Although two words P and Q might be very similar, there is no reason to believe that one can find Q in the snippets for P, or vice versa.

Iosif et al, [21] proposed a web-based metrics for similarity computation between words. The metrics used a web search engine in order to exploit the retrieved information for the words of interest and downloaded a number of the top ranked documents for application. The proposed metrics work automatically, without consulting any human annotated knowledge resource.

Sahami [22] measured similarity between two queries using snippets returned for those queries by a search engine. For each query, they collected snippets from a search engine and represented each snippet as a TF-IDF-weighted term vector. Similarity between two queries was then defined as the inner product between the corresponding vectors. They did not compare their similarity measure with ontology-based similarity measures.

In this paper we do not focus on the first category (This is also our work under way). We try to use World Wide Web to calculate similarity. The concept similarity was measured using the association rule mining in the snippets returned from Web search engines. The most influential algorithm for association rule mining is Apriori. In order to improve the efficiency of Apriori algorithm and use it to measure the concept similarity, there are three main improvements in Apriori algorithm.

The rest of this paper is organized as follows: In section 2 we give the definition of the concept similarity and the knowledge of association rules in data mining. Then we give the definition of association similarity and propose an approach to measure association similarity. The example of our approach is given in section 3. In section 4, we present experiments for evaluating our proposed approach. The last section presents the conclusions and future work.

II. Measuring Concept Similarity Using the Association Rule Mining

A. Concept Similarity

No matter what kind of methods to measure the concept similarity, the connotation of concept similarity is always the same. In order to formalize the similarity concept, we first give a simple definition of similarity, as follows:

Definition: Concept Similarity. When there are some important common characteristics between the two concepts, the concepts are similar in some respects. The degree of similarity is denoted as $sim(CP_a, CP_b)$, where CP_a represents the concept a, and CP_b represents the concept b.

A similarity function is a real-valued function $sim: S \to [0,1]$, on a set S measuring the degree of similarity between two concepts of S. Though there may be split opinions about the properties of sim, it is generally agreed that sim ought to be reflexive and symmetric, i.e.

$\forall CP_a, CP_b \in S$ it holds:

$$\begin{aligned} &sim(CP_a, CP_a) = 1 \\ &sim(CP_a, CP_b) = sim(CP_b, CP_a) \end{aligned} \quad (1)$$

The range of similarity is between 0 and 1. If two concepts are exactly equal to each other, the similarity between them is 1. If there are no common characteristics between two concepts, the similarity is 0.

B. Association Rule Mining

Web search engines provide an efficient interface to analyze vastly numerous documents in the web. Page counts and snippets are two useful information sources provided by most Web search engines.

Page count of a concept is the number of pages that contain the concepts. Page count for the concepts CP_a AND CP_b can be considered as a global measure of co-occurrence of concepts CP_a and CP_b on the Web.

Most popular co-occurrence measures are $Jaccard$, $Dice$ and $Overlap$. The $Jaccard$ coefficient between concepts CP_a and CP_b, $Jaccard(CP_a, CP_b)$, is defined as:

$$Jaccard(CP_a, CP_b) = \frac{N(CP_a \cap CP_b)}{N(CP_a) + N(CP_b) - N(CP_a \cap CP_b)} \quad (2)$$

Where the $N(CP_a)$ denotes the number of pages that contain the concept CP_a and the $N(CP_b)$ denotes the number of pages that contain the concept CP_b. The $N(CP_a \cap CP_b)$ denotes the number of pages that contain both the concept CP_a and CP_b.

The $Dice$ coefficient between concepts CP_a and CP_b, is defined as:

$$Dice(CP_a, CP_b) = \frac{2N(CP_a \cap CP_b)}{N(CP_a) + N(CP_b)} \quad (3)$$

The $Overlap$ coefficient between concepts CP_a and CP_b, is defined as:

$$Overlap(CP_a, CP_b) = \frac{N(CP_a \cap CP_b)}{min(N(CP_a), N(CP_b))} \quad (4)$$

Despite its simplicity, using page counts alone as a measure of co-occurrence of two concepts presents several drawbacks. First, page count analyses ignore the position of a word in a page. Therefore, even though two words appear in a page, they might not be related. Secondly, page counts of a polysemous word (a word with multiple senses) might contain a combination of all its senses. For an example, page count for apple contains page counts for apple as a fruit and apple as a company. Moreover, given the scale and noise in the Web, some words might occur arbitrarily, i.e. by random chance, on some pages. For those reasons, page counts alone are unreliable when measuring semantic similarity.

Definition: Snippets. The snippets are a brief window of text extracted by a search engine around the query term in a document. The snippets can provide useful information regarding the local context of the query term. Processing snippets is also efficient as it obviates the trouble of downloading web pages, which might be time consuming depending on the size of the pages. This paper proposes a novel method to measure concept similarity based on snippets using the technology of the association rule mining.

Association rule [23] is an important knowledge in data mining. Association rule mining is an effective method to find interesting association or correlation relationship among a large set of data items. Let $I = \{I_1, I_2, ..., I_m\}$ be a set of all items (in this paper a concept is an item). Any subset of the set $I = \{I_1, I_2, ..., I_m\}$ is called item set or concept set which is denoted as X, that is $X \subseteq I$. An item set that contains k items is a $k-itemset$.

Let $T = \{T_1, T_2, ..., T_n\}$, the task-related data, be a set of database transactions where each transaction T_i contains a certain number of items (these items are included in the set $I = \{I_1, I_2, ..., I_m\}$) such that $T_i \subseteq I$. Each transaction is associated with an identifier, called TID. In this paper, a context in the snippets is a transaction.

Definition: Support Count. Let X be the concept set. The number of contexts which contain X in the snippets $T = \{T_1, T_2, ..., T_n\}$ is called support count about X, denoted as $\sigma(X)$. Mathematically, the support count about X can be denoted as:

$$\sigma(X) = \left|\{T_i \mid X \subseteq T_i, T_i \in T\}\right| \quad (5)$$

Where the symbol $|\bullet|$ represents the number of elements in the set.

Definition: Support. The support about X can be denoted as $support(\mathrm{X})$ and represented as:

$$support(X) = \frac{\sigma(X)}{N} \times 100\% \quad (6)$$

where $N = |T|$ which represents the number of all contexts in the snippets.

Definition: Association Rule. If X and Y are concept sets, such that $X \subset I$, $Y \subset I$ and $X \bigcap Y = \varnothing$, an association rule is an implication of the form $X \Rightarrow Y$. The strength of association rules can be measured by support (abbreviated as sup) and confidence (abbreviated as $conf$). Support and confidence can be described as follows:

$$support(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{N} \times 100\% \quad (7)$$

Or abbreviated as $sup(X \Rightarrow Y)$

$$confidence(X \Rightarrow Y) = \frac{\sigma(X \cup Y)}{\sigma(X)} \times 100\% \quad (8)$$

Or abbreviated as $conf(X \Rightarrow Y)$

Definition: Association Rule Mining. The association rule mining is to find all rules that the support is greater than $minsup$ and the confidence is greater than $minconf$, where $minsup$ and $minconf$ is the corresponding support and confidence threshold.

A common strategy for association rule mining is a two-step process:

(1) Find all frequent item sets: By definition, each of these item sets will occur at least as frequently as a pre-determined $minsup$.

(2) Generation strong association rules from the frequent item sets: By definition, these rules must satisfy support threshold ($minsup$) and confidence threshold ($minconf$).

The most influential algorithm for association rule mining is Apriori [24]. Apriori employs an iterative approach known as a level-wise search, where $k-itemsets$ are used to explore $(k+1)-itemsets$. First, the frequent $1-itemsets$ is found. This set is denoted L_1. L_1 is used to find L_2, the frequent $2-itemsets$, which is used to find L_3, and so on, until no more frequent $k-itemsets$ can be found. The finding of each L_k requires one full scan of database. In order to use the Apriori property, all nonempty subsets of a frequent

itemset must also be frequent. This property is based on the following observation. By definition, if an itemset I dosen't satisfy the minimum support threshold, min_sup, then I is not frequent, that is, $P(I) < min\_sup$. If an item A is added to the itemset I, then the resulting itemset (i.e. $I \cup A$) cannot occur more frequently than I. Therefore, $I \cup A$ is not frequent either, that is $P(I \cup A) < min\_sup$.

C. Mining Association Similarity Using Improved Apriori Algorithm

In order to improve the efficiency of Apriori algorithm and use it to measure the association similarity, there are three main improvements in Apriori algorithm:

1. Since the measurement of similarity processes between two concepts, our algorithm only needs to find $2-itemsets$, such that we only need to scan the snippetss twice. This can greatly decrease the complexity of the algorithm and reduce the cost of scanning the database.

2. In Apriori algorithm, the choice of support threshold is very important. If the support threshold is too large, the association rules will be filtered too much and the usability of concept sets will depress. If the support threshold is too small, a great deal of concept sets will be generated and increase the processing time. Because we only scan the snippets twice here, we could choose a smaller support threshold in order to measure the similarity more precisely. In this paper, we set the support threshold as:

$$minsup' = min\{support(I_a), support(I_b), minsup\} \quad (9)$$

Where the $minsup$ represents the initial support threshold, $support(I_a)$ and $support(I_b)$ is the support of testing concepts.

3. The Apriori algorithm is only used to find the association rules; we need another method to measure the similarity between concepts.

Definition: The Vector of Testing Concept. Let $FC = \{fc_1, fc_2, \cdots, fc_n\}$ be the set of feature concepts, $C = \{I_a, I_b\}$ be the set of testing concept (the measurement of similarity cannot but process between two concepts), the vector of testing concept can be denoted as:

$$V_{I_a} = \{conf(fc_1 \Rightarrow I_a), \cdots, conf(fc_n \Rightarrow I_a)\} \quad (10)$$

$$V_{I_b} = \{conf(fc_1 \Rightarrow I_b), \cdots, conf(fc_n \Rightarrow I_b)\} \quad (11)$$

Definition: Association Similarity. Let V_{I_a} and V_{I_b} be the vector of testing concept I_a and I_b, the association similarity between I_a and I_b can be denoted as:

$$sim(I_a, I_b) = \frac{\sum_{i=1}^{n} V_{I_a i} \times V_{I_b i}}{\sqrt{(\sum_{i=1}^{n} V_{I_a i}^2)(\sum_{i=1}^{n} V_{I_b i}^2)}} \quad (12)$$

Then we give the algorithm of measuring association similarity as follows:

Input: initial support threshold ($minsup$); testing concepts I_a, I_b.

Output: the association similarity between testing concepts which is denoted as $sim(I_a, I_b)$.

Step 1: Get the text snippets returned from Web search engines about testing concepts I_a, I_b. Generate candidate $1-concept\ sets$ and update the support threshold according to the equation (9);

Step 2: According to the new support threshold, generate frequent $1-concept\ sets$;

Step 3: Remove the testing concept and generate the processed frequent $1-concept\ sets$;

Step 4: Connect the testing concepts to the processed frequent $1-concept\ sets$ and generate the candidate $2-concept\ sets$;

Step 5: Generate the vector of testing concept according to the equation (10) and (11);

Step 6: According to the equation (12), calculate $sim(I_a, I_b)$.

III. THE EXAMPLE OF ALGORITHM

In order to interpret the algorithm more clearly, we give a concrete example. We need to measure the similarity between the concept I_1 and the concept I_2. Snippets returned by search engines contain 7 texts (note that the results returned from search engines in reality will much larger). The returned snippets contain 14 concepts $\{I_1, I_2, ..., I_{14}\}$. Each snippet contains one or more concepts. The support count threshold $min\sigma = 3$ (for simplicity, we use support count threshold here, corresponding support threshold $minsup = 0.42\%$).

(1) The returned snippets;

TABLE 1
The returned snippets

TID	concepts
01	$I_1, I_4, I_5, I_6, I_9, I_{13}$
02	$I_1, I_2, I_5, I_6, I_7, I_8$
03	$I_2, I_3, I_7, I_8, I_{11}, I_{14}$
04	$I_1, I_2, I_3, I_5, I_9, I_{14}$
05	$I_1, I_4, I_5, I_6, I_9, I_{12}$
06	$I_1, I_4, I_5, I_9, I_{10}, I_{11}$
07	$I_2, I_3, I_7, I_8, I_{10}, I_{13}$

(2) Generate candidate $1-concept\ sets$;

TABLE 2
Candidate $1-concept\ sets$

concept sets	support count
I_1	5
I_2	4
I_3	3
I_4	3
I_5	5
I_6	3
I_7	3
I_8	3
I_9	4
I_{10}	2
I_{11}	2
I_{12}	1
I_{13}	2
I_{14}	2

according to the equation (3), calculate that $min\sigma' = 3$.

(3) Generate frequent $1-concept\ sets$;

TABLE 3
Frequent $1-concept\ sets$

concept sets	support count
I_1	5
I_2	4
I_3	3
I_4	3
I_5	5
I_6	3
I_7	3
I_8	3
I_9	4

(4) Remove the testing concept I_1, I_2;

TABLE 4
Processed frequent $1-concept\ sets$

concept sets	support count
I_3	3
I_4	3
I_5	5
I_6	3
I_7	3
I_8	3
I_9	4

(5) Connect and generate the candidate $2-concept\ sets$;

TABLE 5
Candidate $2-concept\ sets$

concept sets	support count
(I_1, I_3)	1
(I_1, I_4)	3
(I_1, I_5)	5
(I_1, I_6)	3
(I_1, I_7)	1
(I_1, I_8)	1
(I_1, I_9)	4
(I_2, I_3)	3
(I_2, I_4)	0
(I_2, I_5)	2
(I_2, I_6)	1
(I_2, I_7)	3
(I_2, I_8)	3
(I_2, I_9)	1

(6) Generate the vector of testing concept:

$$confidence(I_k \Rightarrow I_1) = \frac{\sigma(I_k \cup I_1)}{\sigma(I_k)} \times 100\%$$

$$confidence(I_k \Rightarrow I_2) = \frac{\sigma(I_k \cup I_2)}{\sigma(I_k)} \times 100\%$$

$V_{I_1} = \{0.33, 1, 1, 1, 0.33, 0.33, 1\}$

$V_{I_2} = \{1, 0, 0.4, 0.33, 1, 1, 0.25\}$

(7) Finally calculate the association similarity between I_1 and I_2:

$$sim(I_1, I_2) = \frac{\sum_{i=1}^{7} V_{I_1 i} \times V_{I_2 i}}{\sqrt{(\sum_{i=1}^{7} V_{I_1 i}^2)(\sum_{i=1}^{n} V_{I_2 i}^2)}} = 0.52 \cdot$$

IV. Experiment and Result Analyses

In this section, we will compare our approach with other methods we have mentioned in section 1. First we will compare with Sahami and Iosif's methods which are also methods to measure the concept similarity based on World Wide Web. Then we will compare other methods, including that are based on the ontology and based on the World Wide Web.

Although there is no standard way to evaluate computational measures of concept similarity, one reasonable way to judge would seem to be agreement with human similarity ratings. This can be assessed by using a computational similarity measure to rate the similarity of a set of word pairs, and looking at how well its ratings correlate with human ratings of the same pairs.

An experiment by Miller and Charles [25] provided appropriate human subject data for the task. In their study, 38 undergraduate subjects were given 30 word pairs that were chosen to cover high, intermediate and low levels of similarity, and asked to rate similarity of meaning for each pair on a scale from 0 (no similarity) to 4 (perfect synonymy). The average rating for each pair thus represents a good estimate of how similar the two concepts are, according to human judgments. Miller-Charles ratings can be considered as a reliable benchmark for evaluating similarity measures. Note that most researchers have used only 28 word pairs of the Miller-Charles for evolutions because of the omission of two word pairs in earlier versions of WordNet.

The result of measurement of association similarity is shown in the table 6. We also give the figure 1 in order to show the result clearly. In figure 1, the similarity of Miller-Charles is also transformed to the range of [0, 1].

Through the figure we can see that the method of Sahami is coarse and not so precise. The method of Iosif is better than Sahami. But in some word pairs such as "coast-forest" and "crane-implement", the method of Iosif is also inaccurate compared to Miller-Charles. Our approach is more precise in general although in some word pairs the result is not so accurate. Especially the measurement of word pair "asylum-madhouse" is inaccurate because in the snippets the related contexts

TABLE 6
Similarity of word pairs

Word Pair	M&C	Sahami	Iosif	Ours
noon-string	0.08	0.08	0.16	0.01
rooster-voyage	0.08	0.2	0	0.03
glass-magician	0.11	0.14	0.18	0.21
chord-smile	0.13	0.09	0.4	0.26
coast-forest	0.42	0.25	0.76	0.41
lad-wizard	0.42	0.15	0.37	0.23
monk-slave	0.55	0.1	0.19	0.31
forest-graveyard	0.84	0	0.11	0.03
coast-hill	0.87	0.29	0.18	0.39
food-rooster	0.89	0.8	0.35	0.5
monk-oracle	1.1	0.05	0.47	0.18
car-journey	1.16	0.19	0.52	0.43
brother-lad	1.66	0.24	0.58	0.65
crane-implement	1.68	0.15	0.1	0.58
brother-monk	2.82	0.27	0.63	0.37
implement-tool	2.95	0.42	0.8	0.54
bird-crane	2.97	0.22	0.59	0.83
bird-cock	3.05	0.06	0.44	0.91
food-fruit	3.08	0.18	0.79	0.58
furnace-stove	3.11	0.31	1	0.94
midday-noon	3.42	0.29	0.74	0.63
magician-wizard	3.5	0.23	0.59	0.68
asylum-madhouse	3.61	0.21	0.51	0.34
coast-shore	3.7	0.38	0.5	0.75
boy-lad	3.76	0.47	0.67	0.71
gem-jewel	3.84	0.21	0.53	0.66
journey-voyage	3.84	0.52	0.75	0.91
automobile-car	3.92	1	0.76	0.9

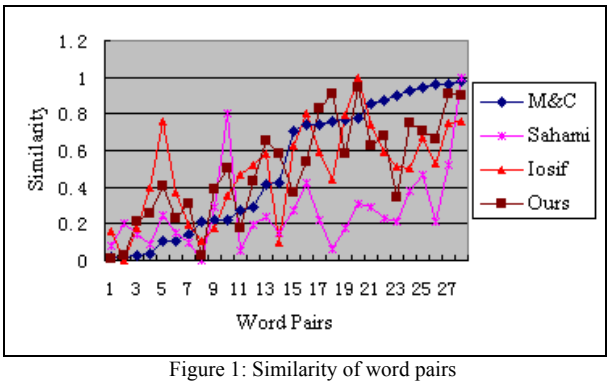

Figure 1: Similarity of word pairs

TABLE 7
Correlation coefficient of all methods

Methods	Correlation
M&C	1
Li	0.822
Jiang	0.848
X-Similarity	0.75
Iosif	0.71
Sahami	0.579
Ours	0.787

about asylum and madhouse are rare. This proves that the number of related contexts can influence the result of similarity greatly. We may improve the result by increasing the number of context in the snippets or combining our approach with ontology.

In table 7, we compare the methods we have mentioned in section 1. As many published work did, we give the correlation coefficient between the human judgments (M&C) and other methods including our approach. The method performs well when the correlation coefficient is near to 1. From the table we can see that the ontology-based methods such as Li, Jiang, and X-Similarity perform better than other methods. The main reason for this performance gap is that ontology-based measures are based on resources made by domain experts where concepts are structured around a logical semantic tree. However, if the testing concepts are not appeared in the ontology, the ontology-based methods will do nothing but fail of calculating. This is the coverage problem of those methods and the reason why most researchers use only 28 word pairs of Miller-Charles for experiments but not the 30 word pairs. On the contrary, the Web-based methods don't have this problem exactly. Taking Web-based methods into consideration, our approach performs better than the others and almost nearly to the ontology-based methods. Therefore, in principle our proposed method could be used to measure similarity between concepts, especially those which are not listed in WordNet or other manually compiled ontology.

V. Conclusions and Future Work

In this paper we first give the definition of the concept similarity and analyze common methods for measuring concept similarity. Then we propose an approach to

measure association similarity using improved Apriori algorithm which is usually used to mining association rules in large database. According to our approach, we can measure the similarity between concepts fast and precisely. In the end, we present experiment for evaluating our proposed approach and compare with other methods which is also used to measure the similarity between concepts.

Our plans for future work include the improvement of our algorithm for more precise similarity calculation and apply our research to Natural Language Processing, Semantic Web and so on.

REFERENCE

[1] P. Resnik. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Nature Language [J]. Journal of Artificial Intelligence Research. 1999, 11: 95-130.

[2] M. A. Alvarez, S. Lim. A Graph Modeling of Semantic Similarity between Words [C]. Proceedings of IEEE International Conference on Semantic Computing. Irvine, California, 2007: 355-362.

[3] D. Lin. Automatic retreival and clustering of similar words [C]. Proceedings of the 17th COLING. 1998: 768-774.

[4] P. D. Turney. Learning Algorithms for Keyphrase Extraction [J]. Information Retrieval. 2000, 2: 303-336.

[5] P. Cimano, S. Handschuh, S. Staab. Towards the self-annotating web [C]. Proceedings of the 13th international conference on World Wide Web (WWW'04). 2004.

[6] P. Mika. Ontologies are us: A unified model of social networks and semantics [C]. Proceedings of International Semantic Web Conference (ISWC'05). Galway, Ireland, 2005: 522-536.

[7] J. Mori, Y. Matsuo, M. Ishizuka. Extracting keyphrases to represent relations in social networks from web [C]. Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI'07). 2007: 2820-2825.

[8] G. A. Miller. WordNet: A Lexical Database for English [J]. Communications of the ACM. 1995, 38(11): 39-41.

[9] K. Haase. Interlingual BRICO [J]. IBM Systems Journal. 2000, 39: 589-596.

[10] Y. H. Li, Z. A. Bandar, D. Mclean. An Approach for Measuring Semantic Similarity between Words using Multiple Information Sources [J]. IEEE Transactions on Knowledge and Data Engineering. 2003, 15(4): 871-882.

[11] R. Rada, H. Mili, E. Bicknell, et al. Development and Application of a Metric on Semantic Nets [J]. IEEE Transactions on Systems, Man, and Cybernetics. 1989, 19(1): 17-30.

[12] P. Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy [C]. Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI'95). 1995: 448-453.

[13] J. J. Jiang, D. W. Conrath. Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy [C]. Proceedings of the 10th International Conference on Research on Computational Linguistics (ROCLING X'97). Taiwan, China, 1997: 8-22.

[14] G. M. E. Petrakis, G. Varelas, A. Hliaoutakis, et al. X-Similarity: Computing Semantic Similarity between Concepts from Different Ontologies [J]. Journal of Digital Information Management. 2006, 4(4): 233-237.

[15] A. Formica. Concept similarity by evaluating information contents and feature vectors a combined approach [J]. Communications of the ACM. 2009, 52(3): 145-149.

[16] K. W. Church, P. Hanks. Word association norms, mutual information, and lexicography [J]. Computational Linguistics. 1990, 16(1): 22-29.

[17] D. Carmel, E. Farchi, Y. Petruschka, et al. Automatic Query Wefinement using Lexical Affinities with Maximal Information Gain [C]. Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'02). Tampere, Finland, 2002: 283-290.

[18] R. Albertoni, M. De Martino. Asymmetric and Context-Dependent Semantic Similarity among Ontology Instances [C]. Journal on Data Semantics X , LNCS 4900. 2008: 1-30.

[19] Y. Matsuo, J. Mori, M. Hamasaki, et al. Polyphonet: An advanced social network extraction system [C]. Proceedings of the 15th International Conference on World Wide Web (WWW'06). 2006.

[20] H. Chen, M. Lin, Y. Wei. Novel association measures using web search with double checking [C]. Proceedings of the COLING/ACL'06. 2006: 1009-1016.

[21] E. Iosif, A. Potamianos. Unsupervised Semantic Similarity Computation using Web Search Engines [C]. Proceedings of 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI'07). 2007: 381-387.

[22] M. Sahami, T. Heilman. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets [C]. Proceedings of the 15th International Conference on World Wide Web (WWW'06). Edinburgh, Scotland, 2006.

[23] R. Agrawal, T. Imielinski, A. Swami. Mining Association Rules between Sets of Items in Large Databases [C]. Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD'93). Washington, DC, USA, 1993: 207-216.

[24] R. Agrawal, R. Srikant. Fast Algorithms for Mining Association Rules in Large Databases [C]. Proceedings of 20th International Conference on Very Large Data Bases (VLDB'94). Santiago, Chile, 1994: 487-499.

[25] G. A. Miller, W. G. Charles. Contextual Correlates of Semantic Similarity [J]. Language and Cognitive Processes. 1991, 6(1): 1-28.

Li Chen received the Bachelor of Engineering and Master of Engineering degrees from Institute of Meteorology, PLA University of Science and Technology in 2005 and 2008, respectively. He began his PhD study in Institute of Command Automation, PLA University of Science and Technology since 2008. His research interests include semantic Web, data mining, and Web services.

Zi-lin Song received the Bachelor of Engineering and Master of Engineering degrees from Tsinghua University in 1968 and 1981, respectively. At present, He is the professor of Institute of Command Automation, PLA University of Science and Technology. His research interests include data mining, artificial intelligence and semantic Web.

Ying Zhang received Master of Engineering degrees from Institute of Meteorology, PLA University of Science and Technology in 2008. She began her PhD study in Institute of

Command Automation, PLA University of Science and Technology since 2008. Her research interests include requirement engineering, knowledge-based systems and service-oriented computing.

Ning Zhou received the Bachelor of Engineering and Master of Engineering degrees from Institute of Meteorology, PLA University of Science and Technology in 2003 and 2007, respectively. He received the PhD degree from Institute of Command Automation, PLA University of Science and Technology in 2009. His research interests include software engineering, Web services and data mining.