THE UNDERSTANDING BETWEEN AGENT CRAWLERS BASED ON DOMAIN ONTOLOGY

*Yajun Du, Yingyu Wang, Shaoming Chen**

Abstract: Ontology is widely used in the computer domain to structure concepts that represent a view of world nowadays, which could formally specify semantic relationship among the terms. In this paper, we present coordination between agent crawlers based on ontology in Topic Specific Search Engines, and we try to measure understanding among them, relying on Formal Concept Analysis (FCA) instead of comparing the terms only. In literature, most papers on concept similarity in FCA are based on two different concepts in the same concept lattice, and whereas there is very little research related to different concept lattices or even different agents. We propose a novel method on concept similarity for computing the Concept-Concept similarity, the Concept-Ontology similarity and the Ontology-Ontology similarity, and at last we can deduce understanding among agent crawlers. Finally, we can guide the crawlers effectively in our Search Engine.

Key words: *Agent crawlers, ontology, formal concept analysis, concept similarity, degree of understanding*

Received: December 31, 2008
Revised and accepted: June 11, 2012

1. Introduction

In this era exploding with information, more and more users expect to retrieve available information from Search Engine. At this time, Topic Specific Search Agents (TSSA) have become the most important tools as an efficient and effective retrieval approach. In topic specific search field, most researchers focus on the autonomy of agent [1–4], i.e. they studied a single agent and how to crawl on the web for efficiently retrieving more relevant web pages. However, we want to discuss coordination among agents by their communication when they crawl on the web through TSSA.

*Yajun Du, Yingyu Wang, Shaoming Chen
School of Mathematics & Computer Science, Xihua University, Chengdu 610039, Sichuan, China, E-mail: duyajun@mail.xhu.edu.cn

©ICS AS CR 2012

1.1 Coordination and communication

In paper [5], the authors divide TSSA into two classes: F-Agent and C-Agent. The former makes task plans, manages the C-Agents as well as deals with communication among TSSA, whereas the latter crawls on the web for retrieving relevant pages. Based on the contract net protocol [6], a negotiation protocol is proposed to control the cooperation and competition among TSSA. Fig. 1 shows the communication model of TSSA.

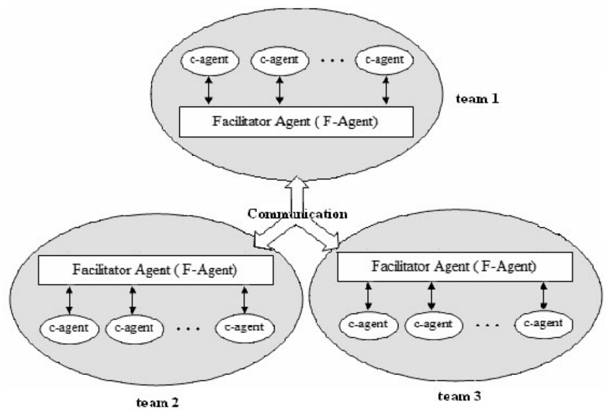

Fig. 1 *The communication model of TSSA.*

Agents in TSSA do not act alone but they coordinate to accomplish their subtasks. C-Agent will request cooperation from other C-Agents in two conditions: First, when the candidate links exceed the limitative amount, then C-Agent may appeal to the C-Agents in the same team for sharing in partial candidate links; second, cooperation occurs between different teams. When C-Agent crawls on the web, it will encounter some links which are not its target but they are other C-Agents' task, so the cooperation among C-Agents in different teams is needed. Fig. 2 shows the both instances:

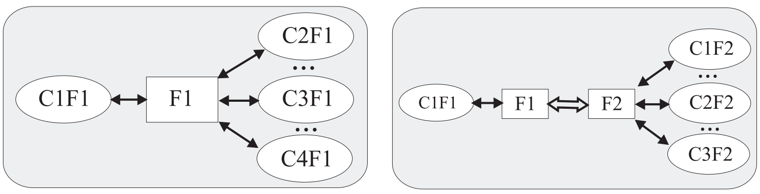

Fig. 2 *The cooperation among agents in one team and in different teams.*

Therefore, communication and negotiation is essential to allow the agents to adjust their local schedules in order to achieve global objectives.

1.2 Ontology and formal concept analysis

Ontology is widely used in the computer domain to structure concepts that represent a view of world nowadays, which could formally specify semantic relationship among the terms. Also, definitions of ontology are given by many researchers. Ontology is a "formal, explicit specification of a shared conceptualization", where a "conceptualization" is an abstract model of some phenomenon of the world which identifies the relevant concepts (or entities) and relationships among the concepts of that phenomenon. "Shared" means that an ontology captures consensual knowledge, whereas "formal" refers to the fact that an ontology should be machine-understandable. An ontology contains a set of interrelated concepts, each associated with a formal definition providing an unambiguous meaning of the concept in the given domain [7]. It is represented as: $O = (C, A^c, R, A^R, H, I, X)$, where C is a concept set, A^c is an attribute set of concepts, R is a relationships set, A^R is an attribute set of relationships, H is a hierarchies set, I is an instances set, X is an axiom set.

Formal Concept Analysis (FCA) proposed by Wille in [8] provides a theoretical framework for the design and discovery concept hierarchies from relational information system. FCA has been used in information retrieval and knowledge discovery etc. It is especially suitable for exploration for symbolic knowledge (concepts) contained in a formal context, such as corpus, database, or ontology [9]. In this perspective, a concept is not an abstraction but, on the basis of the observation of the reality, it is a clustering of objects and related common attributes [10].

1.3 Research objectives

According to the method introduced above, all cooperating agents must understand each other before further communication, no matter if the cooperation happens in one team or in different teams. Based on this point of view, we put forward coordination among agent crawlers.

In paper [11], the authors narrow the concept to represent only an object or thing that has a name in a natural language, but they try to measure similarity between concepts in different agents. Some of these methods are used in our research and give us some understanding, although the procedures use the word "comparison" instead of "concept comparison".

Because the natural language is ambiguous and its understanding needs a high level of intelligence, it cannot achieve complete understanding of semantics. This paper proposes applying FCA to TSSA and enhancing match technology to the level of concept. We construct ontology for each agent crawler relying on concept lattice, and try to measure understanding among them by computing concept similarity based on ontology. When an agent crawler goes into a certain domain, it can be compared with the agents in the existing domain at first. If the understanding that the agent has about the vast majority of existing agents is less than a certain threshold (this threshold is defined by experiments), it is illustrated that this agent does not belong to that domain and we should exclude it.

This paper is organized as follows. Section 2 gives an overview of related works. Section 3 introduces FCA in detail and constructs the concept lattice of the agent crawler ontology. Then we analyze the understanding between two agent crawlers in detail in Section 4. Section 5 concludes our work and suggestions are given.

2. Related Works

Ontology and FCA both aim at modelling "concepts", therefore many researches concentrate on how FCA can be used to support ontology engineering and some methods to build Oontology relying on FCA have been proposed so far [12–14]. Note that, in general, from a theoretical point of view, ontology concepts are identified with FCA concepts [15]. However, in many applications, the canonical match is between ontology concepts and FCA attributes, which is the approach followed in this paper. Therefore, FCA attributes can be seen as concepts, in the sense that for building concepts, other concepts that play the role of attributes are needed [13].

Currently, FCA techniques appear interesting in supporting difficult activities that are becoming fundamental in the development of the semantic web [7, 10, 15, 20]. Assessing concept similarity is one of such activities which is growing in importance within ontology engineering and, in particular, ontology merging and ontology alignment [10, 17, 21]. With the fast development of the semantic web, it is likely that the number of ontologies will greatly increase during the next few years, which leads to the arising demand for rapid and accurate assessing concept similarity. So, assessing the similarity between concepts has attracted much attention of the researchers [10, 17, 19].

There are many ways to calculate the concept similarity based on FCA [10] [16–18]. The prerequisite of the method presented in [10] is the existence of a predefined domain ontology containing similarity degrees for any pair of concept descriptors (attributes) in the domain. Such similarity degrees are established by a panel of experts in the given domain, according to a consensus system. The method [16] is based on the attribute in FCA. The measurement of concept similarity is defined by semantic similarity and semantic distance. [17] proposes measuring the concept descriptor similarity by following the information content approach that calculates attribute similarity by using the noun frequencies that are defined in a lexical database. A new similarity computational model based on concept lattice is introduced in [18]; this model allows to compute concept similarity according to the meet-irreducible elements. There are very few researches on measuring the similarity between two concepts in different domain ontologies, or even in two different domain ontologies.

3. Agent Crawler Ontology

3.1 Formal concept analysis

Definition 3.1 [8] A formal context is an ordered triple $K = (G, M, I)$, where G, M are finite nonempty sets and $I \subseteq G \times M$ is a binary relation. The elements in G are interpreted to be objects, and elements in M are said to be attributes. If

$(g, m) \in I$, the object g is said to have the attribute m. The incidence relation of a formal context can be naturally represented by an incidence table.

Here we describe G as the set of URLs in agent crawler, and describe M as the set of terms in web pages which G links to. Each couple (g, m) denotes the fact that the object $g \in G$ is related to the attribution $m \in M$.

Example 3.1 Give $K = (G, M, I), G = \{1, 2, 3, 4, 5\}$ is the URLs set in one agent crawler. $M = \{$ *Internet, Technology, Network, Webpage, Information, Spider*$\}$ is the set of terms in web pages which G links to. $K =$ *(G, M, I)* is a formal context, and its incidence relation described in Tab. I below, where *I, t, n, w, i, s* stand for *Internet, Technology, Network, Webpage, Information, Spider*, respectively.

	Internet	***Technology***	***Network***	***Webpage***	***Information***	***Spider***
1				√	√	√
2	√	√	√	√		√
3	√	√	√	√		
4		√	√		√	√
5		√		√	√	√

Tab. I *Formal context.*

Definition 3.2 [8] Given $K =$ *(G, M, I)* a formal context. For a set of web pages $X \subseteq G$, a set of terms Y defined on M, the operators $\uparrow$ and $\downarrow$ are defined as follows:

$$\uparrow: \quad P(G) \to P(M), X^{\uparrow} = \{m \in M; \forall g \in X, (g, m) \in I\}$$

$$\downarrow: \quad P(M) \to P(G), Y^{\downarrow} = \{g \in G; \forall m \in Y, (g, m) \in I\}$$

Definition 3.3 [8] A formal concept of a context $K =$ *(G, M, I)* is a pair $(A, B) \in P(G) \times P(M)$ such that $A^{\uparrow} = B$ and $B^{\downarrow} = A$. The set A is called the formal concept's extent and the set B its intent.

The subset *L(G, M, I)* of $P(G) \times P(M)$ formed by all the formal concepts of the context is a complete lattice with the order relation: $(A, B) \leq (C, D)$ if and only if A$\subseteq C$ (or equivalently $B \supseteq D$).

This relation shows the hierarchy between the concepts of the context. The lattice *(L(G, M, I),*$\leq$) is said to be the *concept lattice* of the context $K =$ *(G, M, I)* with LUB and GLB are given as follows:

$$\bigvee_{i=1}^{n} (A_i, B_i) = \left(\left(\bigcup_{i=1}^{n} A_i \right)^{\uparrow\downarrow}, \bigcap_{i=1}^{n} B_i \right)$$

$$\bigwedge_{i=1}^{n} (A_i, B_i) = \left(\bigcap_{i=1}^{n} A_i, \left(\bigcap_{i=1}^{n} B_i \right)^{\uparrow\downarrow} \right)$$

Example 3.2 Given a formal context $K = (G, M, I)$, whose incidence relation is shown in Tab. I, we can build a concept lattice shown in Fig. 3.

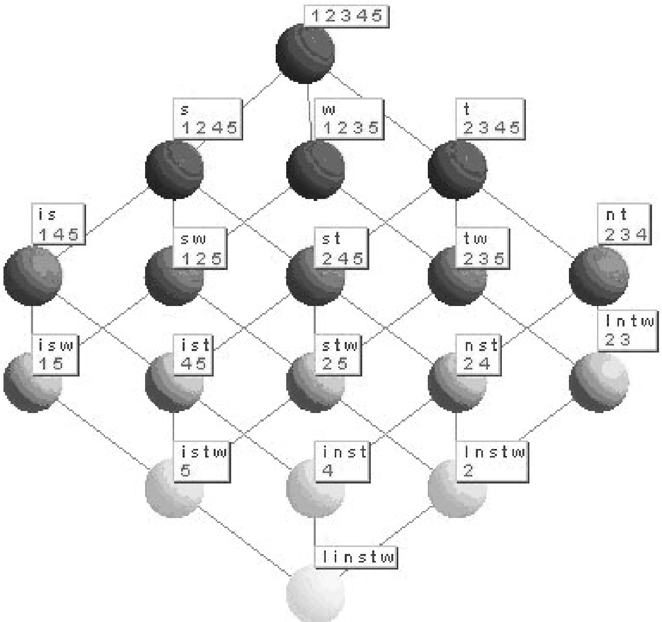

Fig. 3 *Concept lattice corresponding to formal context (Tab. I).*

3.2 Agent crawler ontology

Here we give a simple example to construct the concept lattice. The concept lattice is actually a reflection of the relationships between the concepts of the semantic web; the lattice construction process is in fact the concept clustering process. Through such a lattice, which may be represented in a graphical form as a network, one can be able to find some hidden - direct or indirect – relationship between some concepts. Such a concept lattice can portray an ontology, we work with similarly an ontology of an agent crawler.

Definition 3.4 (*Agent crawler ontology*). An *agent crawler ontology O* is specified by a set of concepts C, and a set of semantic relations, such as ISA, part-of, relatedness, etc. We use the concept lattice in FCA indicating the partial order between a given set of concept C.

In this paper, the approach generally adopted in many applications for combining FCA and ontologies has been followed [14]: Given an ontology and a context $K = (G, M, I)$, the concept lattice can be constructed by integrating ontology concepts and FCA attributes.

4. Measuring the Understanding between Two Agent Crawlers

In Section 3, we constructed the agent crawler ontology relying on FCA. We use O_A and O_B as the ontologies of agent crawlers A (the talker or sender) and B (the listener), c_A and c_B as the concepts (nodes of their own ontology), respectively, in the rest of the paper. This section finds the concept $c_B \in O_B$ most similar to $c_A \in O_A$; and then adding the similarities over all $c_B \in O_B$, we measure the degree of understanding of A with respect to O_B. From this we can predict how well A knows the concepts in O_B.

4.1 Concept-concept similarity

In the real world, hyperlinks between two web pages are very important parts of their semantic relations. Here let *ext* be the extension similarity between two concepts (X_1, W_1) and (X_2, W_2) of two agent crawlers:

$$ext = \frac{|X_1 \rightarrow_L X_2| + |X_1 \leftarrow_L X_2|}{2 * \max(|X_1|, |X_2|)},$$

where $|X_1 \rightarrow_L X_2|$ is the number of hyperlinks in which URLs in (X_1, W_1) link to URLs in (X_2, W_2) and $|X_1 \leftarrow_L X_2|$ is the number of hyperlinks in which URLs in (X_2, W_2) link to URLs in (X_1, W_1), respectively. We will add this extension to the following formula and make the similarity values more accurate.

We use the ***WordNet*** [22], which is a lexical database for the English nouns, as our ontology computing the similarity between two terms. Let n, m be the cardinalities of the sets W_1, W_2, respectively, i.e. $n = |W_1|$, $m = |W_2|$, and suppose that $n < m$. The set $F(W_1, W_2)$ of the candidate sets of pairs is defined by all possible sets of n pairs of attributes defined as follows [10]:

$$F(W_1, W_2) = \{\{\langle a_1, b_1\rangle \ldots \langle a_n, b_n\rangle\} | \, a_h \in W_1, b_h \in W_2, \forall h = 1, \ldots, n, \text{ and } a_h \neq a_k, b_h \neq b_l, \forall k, l \neq h\}.$$

In ***WordNet,*** nouns are organized essentially according to the *ISA, part-of*, and for each noun, a set of synonyms is given. Consider each pair $\langle a_h, b_h\rangle$ of each candidate set, the similarity degree of each pair $cs\langle a_h, b_h\rangle$ is defined as follows:

$$\mathrm{cs}\langle a_h, b_h\rangle = \begin{cases} 1 & a_h = b_h \text{ or } a_h \text{ and } b_h \text{ have synonymous relation in WordNet} \\ \alpha & a_h \text{ and } b_h \text{ have ISA relation in WordNet} \\ \beta & a_h \text{ and } b_h \text{ have PartOf relation in WordNet} \\ 0 & \text{Others} \end{cases}$$

where α,β $(0 \leq \alpha, \beta \leq 1)$ that can be given by the user.

Consider two concepts (X_1, W_1) and (X_2, W_2) of the different agents, then the concept-concept similarity between (X_1, W_1) and (X_2, W_2) is defined as follows:

$$sim\left((X_1, W_1), (X_2, W_2)\right) = w * ext + (1 - w) * \left[\frac{1}{m} \max_{f \in F(W_1, W_2)} \left(\sum_{(a,b) \in f} cs(a, b)\right)\right], \quad (1)$$

where w is a weight such that $0 \leq w \leq 1$, that can be established by the user to enrich the flexibility of the method. $\max\limits_{f \in F(W_1,W_2)} \left(\sum\limits_{(a,b)\in f} cs(a,b) \right)$ is the greatest sum of similarity within all possible candidate sets of pairs. A candidate set of pairs is a subset of $W_1 \times W_2$ such that there are no two pairs in the set sharing an element. Note that $sim\left(\left(X_1, W_1\right), \left(X_2, W_2\right)\right) = sim\left(\left(X_2, W_2\right), \left(X_1, W_1\right)\right)$, that is *Sim* has the symmetry of the similarity and *sim* always represent a value between 0 and 1.

4.2 Concept-ontology similarity

With the concept-concept similarity, we can calculate the similarity between a concept c_i in O_B and the first hierarchical concept (the direct subsequence concept of the LUB), selecting the max as a candidate concept. Next, we calculate the sub-concept of the candidate and for the same reason, we can find one (or more) road R_i in O_B which describe the concept c_i. We use the modified Dijkstra algorithm to calculate the concept-ontology similarity $sim(c_i, O_B)$; the main idea of which is defined as follows:

```
Input: C_i, O_B
Output: Sim(C_i, O_B)
If sim(C_i, MinFactor) < Threthold
      Exit;
Elseif sim(C_i, MaxFactor) < Threthold
      Exit;
Else {
      pushStack(Road, MinFactor);
      pushStack(Road, MaxFactor);
      }
While(!emptyStack(Road))
{
      for(int i=0;i<ve.length;i++){
            max = max <ve[i]?ve[i]:max;
      }
      flag:array[1..100]of boolean;
      fillchar(flag,sizeof(flag),false);
      flag[1]:=true;
      for x:=2 to n do
      {
            for i:=2 to n do
            if (a[1,i]>max)and(flag[i]=false) then
            max:=a[1,i];
            maxn:=i;
      }
      flag[maxn]:=true;
      for j:=1 to n do
          if (j<>maxn) and (a[1,maxn]+a[maxn,j]>a[1,j]) and(flag[j]=false) then
```

```
            a[1,j]:=a[1,maxn]+a[maxn,j];
        pushStack(Road, a[1,j]);
        }
}
While(!emptyStack(Road))
{       t=popStack(Road);
        Sim(C_i, O_B)+=sim(C_i, t);
}
return Sim(C_i, O_B);
```

4.3 Ontology-ontology similarity

We define the similarity between O_A and O_B as follows:

$$sim(O_A, O_B) = \frac{\sum\limits_{c_i \in O_A} \left[sim(c_i, O_B) * (1 + c)^{l_1} \right]}{n}, \tag{2}$$

where n is the number of concepts in O_A and c is a weight describing the depth of the concept lattice which has relevance with concepts similarity, here we set $c = 0.01$ [23]. l_1 is the level with the concept (X_1, W_1) located in the concept lattice.

4.4 Degree of understanding between two agent crawlers

This degree is a measure of the (imperfect) grasp of a concept by an agent A. The idea is that the more relations the concept has in O_A, the larger the degree of knowledge is. The degree of knowledge $dk_A(c)$ of A about a concept c is a number between 0 and 1.

The value *sim* calculated between two concepts (such as c_A and c_B) in different agents (O_A and O_B) above can be thought of as the degree of understanding that agent B has about concept c_A. Each c_A that forces B to answer $sim = 0$ indicates that B has no idea (no concept) about this c_A. We can add all these *sv*'s for every concept $c_A \in O_A$ and find the degree of understanding that agent B has about O_A. It is not difficult to make the conclusion that the degree of understanding of B about $O_A = sim(O_A, O_B)$.

5. Experiment Model

Fig. 4 shows the system architecture. Firstly, the agent searches for web pages according to the keywords and URLs, and then we use Galicia [26] to build the concept lattice (the building method has been introduced in Chapter 3). After that we can calculate the similarity based on the ontology, and finally we can decide whether the agent crawls are associated with the web pages and keywords, otherwise the agent no longer crawls the URLs.

1. Spider: search the web pages from the Internet. Fig. 5 shows the user interface for parameter settings: start URL, start threads (Agent number),

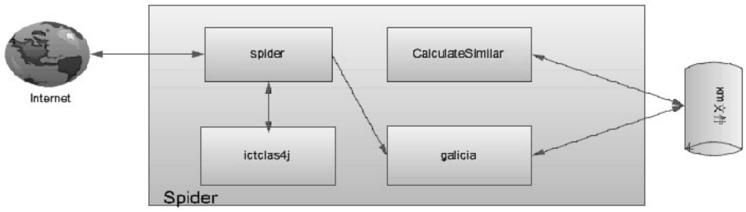

Fig. 4 *System architecture.*

keywords, domain filtering, maximum search depth (using the breadth-first search method), the largest number of URLs, and most important of all, the local store path of the WordNet;

2. Ictclas4j: load the data dictionary, build the word path and select the optimal path;

3. Galicia [26]: (Fig. 7 shows a concept lattice using the Galicia)

   (a) Bivariate table file: establish the bivariate table which maps objects according to the relationships of the URLs and the keywords, then generate the file with the postfix *. bin.xml.

   (b) The concept lattice files: build a concept lattice object according to the given *.bin.xml file, and automatically start a thread to produce the file with the same name *.lat.xml.

   (c) Display of concept lattices graphically: parse the *.lat.xml given by the user into the concept lattice objects and transmit the objects to the display module.

4. Calculate Similarity: Similarity computation is the main idea of this paper; in Section 4, we implement the method for computing the concept-concept similarity, the concept-ontology similarity and the ontology-ontology similarity.

At this point, we would like to mention that we use Java API for WordNet Searching [25]. As its name implies, the Java API for WordNet Searching (JAWS) is an API that provides Java applications with the ability to retrieve data from the WordNet database. It is a simple and fast API that is compatible with both the 2.1 and 3.0 versions of the WordNet database files and can be used with Java 1.4 and later.

In our experiment, we use some of these APIs as follows:

(1) getHypernyms: return the direct hypernyms of this synset.

(2) getInstanceHypernyms: return the instance hypernyms of this synset.

(3) getHyponyms: return the direct hyponyms of this synset.

(4) getInstanceHyponyms: return the instance hyponyms of a synset, where instance hyponyms represent specific (usually real-world) instances of something.

(5) getPartHolonyms: return the holonyms (whole that includes this part) of this type.

(6) getPartMeronyms: return the meronyms (inherited parts) of this type.

Fig. 5 *User interface.*

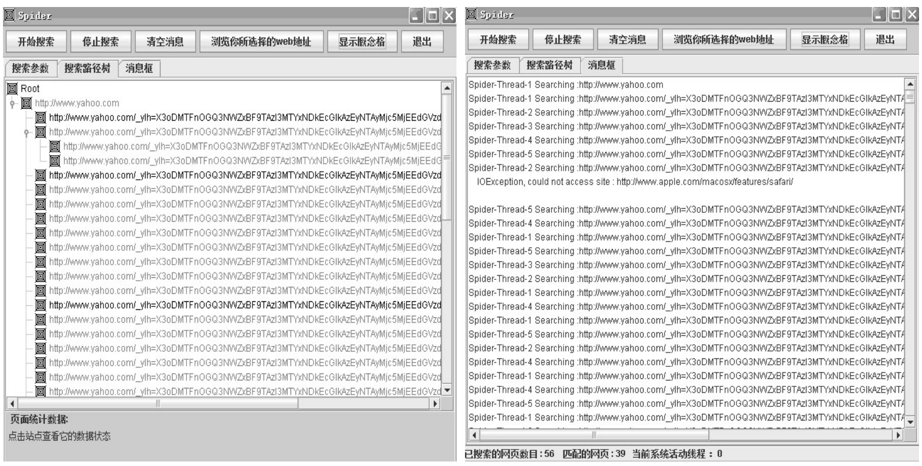

Fig. 6 *Crawling.*

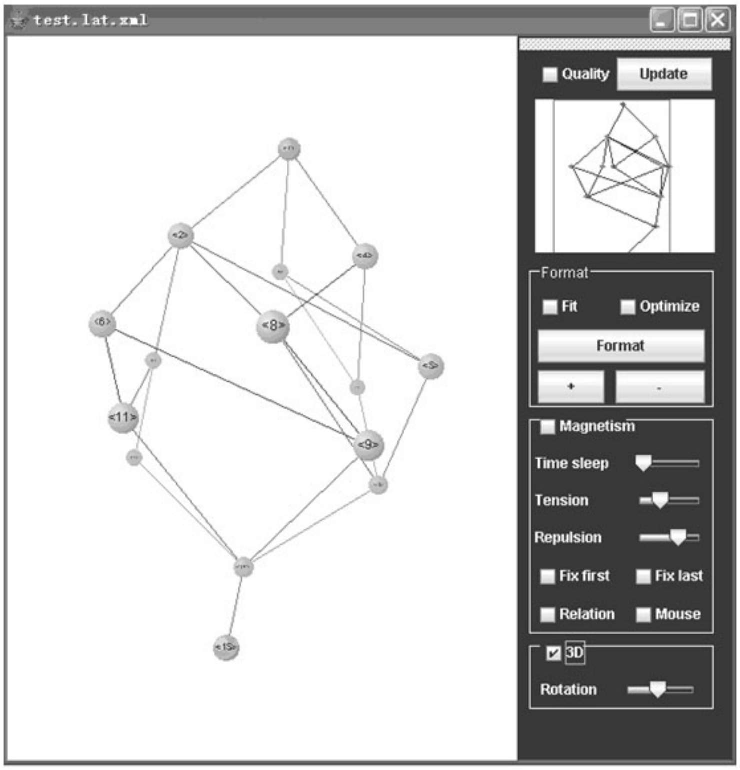

Fig. 7 *Formal concept lattice.*

Fig. 8 shows one result of our method and analogously the spiders crawling in the Internet and pair of the agent crawlers given a similarity value – in this case, we say these agent crawlers understand each other.

Spider-Thread-1	Spider-Thread-2	0.29158875
Spider-Thread-1	Spider-Thread-3	0.39900064
Spider-Thread-1	Spider-Thread-4	0.39848804
Spider-Thread-1	Spider-Thread-5	0.3083337

Fig. 8 *Similarity.*

6. Conclusions and Future Works

This paper proposes a novel method of measuring the understanding between two agent crawlers in the related domain. When an agent crawler goes into a certain domain, at first, it can be compared with the agents in the existing domain. If the understanding that the agent has about the vast majority existing agents is less than a certain threshold (this threshold is defined by experiments), it is illustrated that this agent does not belong to that domain and we should exclude it.

For future work, we put forward the following recommendations:

(1) We only discussed the coordination between two agent crawlers; then we will measure the understanding among multi-agent crawlers, which is a more complex task, considering the competition among them.

(2) Now, our research still remains in the field of theory and some parameters, such as $cs\langle a_h, b_h\rangle$ and c in Section 4 and even the threshold introduced above, must be deliberated. So carrying out experiments and deduction repeatedly is the primary work in the future.

Acknowledgements

Work herein reported was supported by the National Natural Science Foundation of China under Grant No. 60872089 and 61271413. This work is an extension of reference [27].

References

[1] Liu H. Y., Janssen J., Milios E.: Using HMM to learn user browsing patterns for focused Web crawling, Data & Knowledge Engineering, **59**, 2006, pp. 270-291.

[2] Ching-Chi Hsu, FanWu: Topic-specific crawling on the Web with the measurements of the relevancy context graph, Information Systems, **31**, 2006, pp. 232-246.

[3] Chakrabarti S., van den Berg M., Dom B.: Focused crawling: A new approach to topic-specific Web resource discovery, Proceedings of the 8th International WWW Conference, Toronto, Canada, 1999.

[4] Diligenti M., Coetzee F., Lawrence S., Giles C., Gori M.: Focused crawling using context graphs, Proceedings of the 26th International Conference on Very Large Databases (VLDB 2000), Cairo, Egypt, 2000.

[5] Xiang D., Du Y. J., Yi L. Z., Li K.: Coordination and Communication among Topic-specific Search Agents, Proceedings of the 3rd International Conference on Natural Computing, Haikou, 2007, pp. 703-707.

[6] Smith R. G.: The contract net protocol: high level communication and control in a distributed problem solver, IEEE Transactions on Computers **C-29**, 12, 1980, pp. 1104-1113.

[7] Ding Y., Fensel D., Klein M., Omelayenko B.: The Semantic Web: Yet another Hip, Data and Knowledge Engineering, 2002.

[8] Wille R.: An approach based Restructuring lattice theory: hierarchies of concepts, In Ordered Sets, I. Rival, Ed. Reidel, 1982, pp. 445-470.

[9] de Souza K. X. S., Davis J.: Aligning ontologies and evaluating concept similarities, On the Move to Meaningful Internet Systems 2004: Coopls, DOA, and ODBASE, Springer, Berlin, 2004, pp. 1012-1029.

[10] Formica A.: Ontology-based Concept Similarity in Formal Concept Analysis, Information Sciences, 2006, pp. 2624-2641.

[11] Guzman-Arenas A., Olivares-Ceja J. M.: Measuring the understanding between two agents through concept similarity, Expert Systems with Applications, **30**, 2006, pp. 577-591.

[12] Haav H. M.: A Semi-automatic Method to Ontology Design by Using FCA, Proc. of the CLA 2004 Intl. Workshop on Concept Lattices and their Applications Ostrava, Czech Republic, **9**, 2004, pp. 13-24.

[13] Cimiano P., Stumme G., Hotho A., Tane J.: Conceptual Knowledge Processing with Formal Concept Analysis and Ontologies, Proc. of the The Second Intl. Conf. on Formal Concept Analysis (ICFCA 04), Springer, 2004, pp. 189-207.

[14] Obitko M., Snasel V., Smid J.: Ontology Design with Formal Concept Analysis, Proc. of the CLA 2004 Intl. Workshop on Concept Lattices and their Applications Ostrava, Czech Republic, **9**, 2004, pp. 111-119.

[15] Bain M..: Inductive construction of ontologies from Formal Concept Analysis, Australian Conference on Artificial Intelligence, 2003, pp. 88-99.

[16] Lin Z. C., Zhu G. J.: A concept similarity algorithm based on FCA, Computer Technology and Development, **9**, 2008, pp. 112-114 (in Chinese with English abstract).

[17] Formica A.: Concept similarity in Formal Concept Analysis: An information content approach, Knowledge-Based Systems, **21**, 2008, pp. 80-87.

[18] Cao Z. W., Qian J., Zhang W. M., Deng S.: A FCA-based Approach for Concept Similarity Computation, Fuzzy Systems and Mathematics, **22**, 2008, pp. 155-162.

[19] Belohlavek R.: Similarity relations in concept lattices, Journal of Logic and Computation, **10**, 2000, pp. 823-845.

[20] Berners-Lee T. et al.: The Semantic Web, Scientific American, 2001.

[21] Klein M.: Combining and relating ontologies: an analysis of problems and solution. In: A. Gomez-Perez et al. (Eds.), WS on Ontologies and Information Sharing, IJCAI'01, Seattle, 2001.

[22] WordNet 2.1: A lexical database for the English language http://www.cogsci.princeton.edu/cgi-bin/webwn, 2005.

[23] Huang G., Zhou Z. R.: Research on domain ontology-based concept semantic similarity computation, Computer Engineering and Design, **5**, 2007, pp. 2460-2463 (in Chinese with English abstract).

[24] Zhi H. L., Zhi D. J., Liu Z. T.: Concept Similarity Based on Concept Lattice, Computer Science, **35**, 9, 2008 (in Chinese with English abstract).

[25] http://lyle.smu.edu/∼tspell/jaws/index.html

[26] http://www.iro.umontreal.ca/∼galicia/

[27] Wang Y. Y., Du Y. J., Chen S. M.: The understanding between two Agent Crawlers based on domain Ontology. In: CINC, 2009.