

온톨로지 트리 기반 멀티에이전트 세만틱 유사도매칭 알고리즘

A Multi-Agent Improved Semantic Similarity Matching Algorithm Based on Ontology Tree

Qian Gao, 조 영 임*
(Qian Gao¹ and Young Im Cho²)

¹School of information, Shandong Polytechnic University

²College of Information Technology, University of Suwon

Abstract: Semantic-based information retrieval techniques understand the meanings of the concepts that users specify in their queries, but the traditional semantic matching methods based on the ontology tree have three weaknesses which may lead to many false matches, causing the falling precision. In order to improve the matching precision and the recall of the information retrieval, this paper proposes a multi-agent improved semantic similarity matching algorithm based on the ontology tree, which can avoid the considerable computation redundancies and mismatching during the entire matching process. The results of the experiments performed on our algorithm show improvements in precision and recall compared with the information retrieval techniques based on the traditional semantic similarity matching methods.

Keywords: semantic web, similarity matching algorithm, ontology, multi-agent

I. INTRODUCTION

1. Semantic Web

The traditional keyword-based information retrieval technique performs keyword searching in documents by matching the keywords that users specify in their queries. The systems with the technique also maintain a word index to accomplish searching [1,2], such as Google. One crucial problem with these systems is that they do not have the ability to understand the meanings of the keywords (i.e. semantics).

Tim Berners-Lee (2001) [3] proposed a solution to this problem, which extends the current web by giving information a well-defined meaning, better enabling computers and people to work in cooperation. The key to reach this Semantic Web is the use of metadata to annotate documents; thus, software agents would be able to search, locate, discover, or link documents better than keyword-based information retrieval engines of today.

The information retrieval based on the semantic web is to make use of semantic information of the computer procedure's automatic analysis information source to search and find the information source with specific knowledge unit. Tim Berners-Lee proposed seven layer structures of the semantic web [3], as shown in Fig. 1.

The first layer (URI/IRI) is the basis of the whole semantic web, using URI (Uniform Resource Identifier) and IRI (Internationalized Resource Identifiers) to identify the resources of the web.

The second layer (XML) is used to indicate the content and structure of the data. The XML standard provides the necessary

means to declare and use simple data structures, which are stored in XML documents and are machine-readable.

The third layer (RDF) is used to describe the resources on the Web and their type. A Resource Description Framework is a standard for representing metadata information that can be identified with a Universal Resource Identifier (URI).

The fourth layer (Ontology) is used to describe languages and to describe various resources and the relation between them.

The fifth layer (Logic) is used to make logical inference based on the last four layers.

The sixth layer (Proof) provides authentication mechanism according to logical rules and then judges the credibility of the given proof in combination with the application mechanism of the Trust layer.

The seventh layer (Trust) is used to build confidential relation between users.

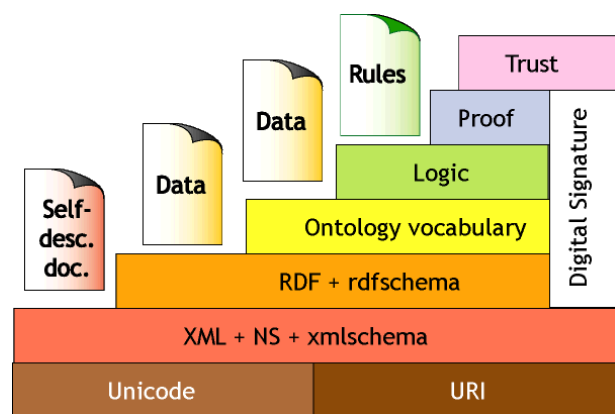


그림 1. 세만틱웹 프레임워크의 계층적 구조.
Fig. 1. Hierarchical structure of Semantic Web framework.

* 책임저자(Corresponding Author)

논문접수: 2012. 8. 27., 수정: 2012. 9. 14., 채택확정: 2012. 9. 25.

Qian Gao: Shandong Polytechnic University(gq@spu.edu.cn)

조영임: 수원대학교 컴퓨터학과(ycho@suwon.ac.kr)

Among the 7 layers, the 2nd, 3rd, 4th and 5th layers of the semantic web system structure are the key to realize the semantic retrieval based on semantic web.

2. Ontology

Ontology was developed in Artificial Intelligence to facilitate knowledge sharing and reuse. As the knowledge representation of the semantic web, the ontology is called the semantic web once applied to the World Wide Web.

There are a number of related studies on ontology development and ontology-based retrieval in literature. In this paper we briefly overview some of them. Lihua Wu et al. (2009) [4] proposed a personalized intelligent web retrieval system based on the knowledge-base concept and latent semantic indexing model, which shields the information irrelevant to users' needs and helps users find needed information on the Internet automatically in accordance with their personal interests and professional requirements. Wei-Dong Fang et al. (2005) [5] proposed a framework to improve precision by searching RDF data instead of text words. This framework extends the traditional VSM (Vector Space Model) and defines weighting scheme to assign different weights to concepts while encountering different relationships between them. SSRM (Semantic Similarity Retrieval Model) [6] finds similar concepts and assigns them initial weights using tf. idf. Then higher weights are assigned to the concepts located in the same semantic neighborhood. K.Saruladha et al. (2010) [7] proposed a semantic similarity measure based on information content metric and information content metric which provides a solution to the sparse data problem prevalent in corpus independently and takes into consideration hyponymy and meronymy (a type of hierarchy that deals with part-whole relationships) relations.

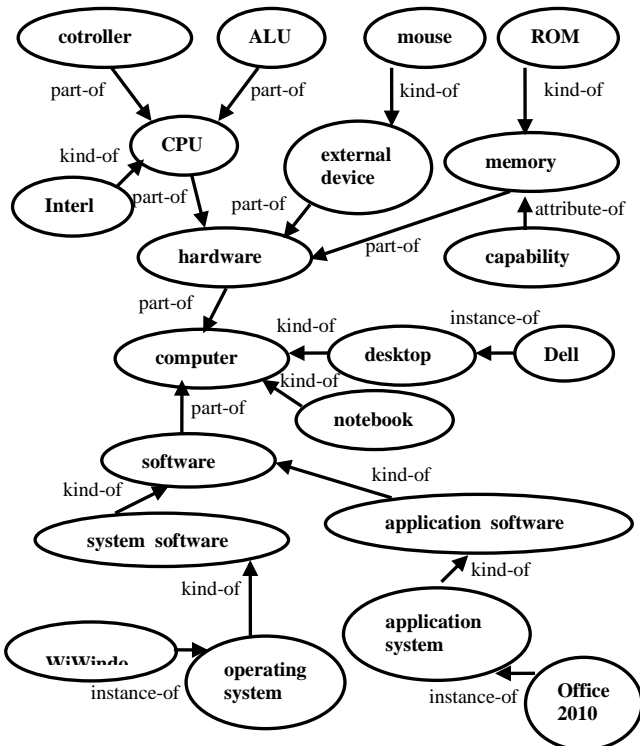


그림 2. 컴퓨터 온톨로지 트리.
Fig. 2. Computer Ontology tree.

Different fields need to build different domain ontologies. Computers exchange information between different fields through the understanding of the ontology. Each document on the semantic web is an ontology, while a big document can be divided into several small ontologies. For example, Fig. 2. shows the partial ontology tree of computer science.

In order to make use of each related ontology to provide users with answers needed by means of inference, our paper adopts Ontology Matching as the essential step to set up mapping relationship between ontologies through comparing the concepts between each other. Through the ontology matching, different ontologies can understand each other by means of the mapping relationship between concepts, hence the realization of the inference based on multiple ontologies.

3. Resource Description Framework

A Resource Description Framework is a standard for representing metadata information that can be identified by using a URI (Universal Resource Identifier). To describe metadata, RDF statements are expressed in triples: subject (represented by a URI or a blank node), predicate or property (represented by a URI) and object (represented by a URI, a blank node or a literal). This triple can be effectively modeled as directed graphs [8].

As shown in Fig. 3, the subject and the object of the triple are modeled as nodes and the predicate as a directed link that describes the relationship between the nodes. The direction of the link always points towards the object [8]. A URI is a more general form of URL (Uniform Resource Locator), which allows information about a resource to be recorded without the use of a specific network address. A blank node is used when a subject or an object node is unknown. It is also used when the relationship between a subject and an object node is n-ary (as is the case with RDF containers). A literal is basically a string with an optional language tag. It is used to represent values like names, dates, and numbers. A typed literal is a string combined with a data type, whereas the data type is always a URI [8].

4. Main idea of Improved Semantic Similarity Matching Algorithm

The traditional semantic matching methods based on the ontology tree adopt the down-up matching sequence, i.e., during the matching process, preferentially match the leaf node of the tree of the ontology, and then match the upper node, and so on. This method has three weaknesses: first, the calculation of the similarity of all the leaf nodes without any screening rules may lead to considerable computation redundancies during the matching process; second, the smaller concept always belongs to a larger one, so that, under the circumstance of the mismatch of larger concepts, even if the small concepts are similar, they still cannot be identified as a correct match; third, the diversity of

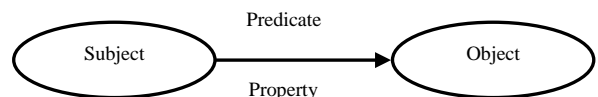


그림 3. RDF 트리플의 다이어렉트 그래프.
Fig. 3. RDF triple as directed graph.

partition granularity may lead to the structural diversity of ontology tree, thus it is likely to result in the mismatch during the process of ontology matching [9].

The weaknesses mentioned above lead to many false matches, causing the falling precision. Focusing on this problem, this paper first uses Resource Description Framework to represent the ontology tree so as to describe the relationship between each term worked out from the certain ontology, and then proposes a new Up-Down Semantic Similarity Matching algorithm based on the Ontology Tree by using three kinds of similarity matching strategy to realize the similarity matching between users' query ontology constructed by Knowledge Processing Agent and application ontology which is the description of the Metadata in form of Ontology.

II. MATERIALS AND THE PROPOSED METHOD

1. The Framework of the Multi-Agent Information Retrieval System Based on Ontology Tree

Based on the principle of relying on the semantic web technology to increase the semantics for information retrieval, we construct a multi-agent information retrieval System based on the semantic web of ontology, shown as Fig. 4.

IA (Interface Agent) is the interface for users and systems to communicate with each other, through which, users send queries to the system in form of $Q(q_1, q_2, \dots, q_m)$, and the system sends the retrieval results back to users.

KCSA (Knowledge Collecting and Storing Agent) is used to construct the domain ontology base as well as to make use of domain ontology base to construct the application ontology base of the information resources (metadata information) to realize the representation and the storage of the knowledge. The metadata information has two components: Source Metadata and Content Metadata. The Source Metadata contains metadata information (i.e. title, URI, statistics, category) about documents. They help in

identifying relevant documents in order to avoid querying all available documents. Content Metadata contains metadata of contents of the documents in the form of RDF triples.

PQRA (Personalized Query Refinement Agent) is responsible for transforming the query of the user to expanded query ontology according to the domain ontology base. In this agent, we adopt three kinds of strategies to realize the personalized query refinement. Firstly, we use knowledge-based query expansion strategy to realize the domain the query belongs to, so that we can expand the query with hypernym (kind-of relationship), hyponym (part-of relationship) and allomorph (instance-of relationship) according to the ontology tree. Secondly, we use user-device-based query expansion strategy to expand the query to realize the personalized query refinement by considering the personal interest of all the intelligent devices for the same user. Thirdly, we use weighted query expansion strategy to calculate the score of the candidate term obtained at the stage of the user-device-based query expansion to determine the final extension term according to use frequency of the device. After the three kinds of strategy, the expanded query will be in form of $Q(O_1, O_2, \dots, O_n)$; here $O_i(i = 1, 2, \dots, n)$ are the concepts in the ontology tree.

SMA (Semantic Matching Agent) adopts the Semantic Matching algorithm (see section 2.3) to realize the similarity matching between the expanded query ontology and application ontology. Finally, send the matching concept of application ontology in form of $Q(AO_1, AO_2, \dots, AO_n)$ to the information retrieval agent.

IRA (Information Retrieval Agent) adopts the method given by Gao & Cho (2012) [10] to acquire the retrieval result.

PQRA and SMA are crucial to realize the more efficient and more precise personalized information retrieval. My another paper has illustrated the detail of PQRA, and this paper mainly concentrates on how to realize the Semantic Similarity Matching between two ontology, hence in the following section. Section 2.2 will introduce the Ontology Similarity Calculation Method used by SMA in details.

2. Ontology Similarity Calculation Method

In order to compare two ontologies and calculate the similarity between them, we first calculate the degree of correlation to judge the relevance of the ontology, and then calculate the similarity of the concept from the name, the semantics and the attribute to form the overall conceptual degree of similarity based on the main idea of the similarity matching method of Maedche [11].

2.1 Concept Relevance Calculation

Relevance refers to the degree of correlation between concepts. Similarity means lexical coincidence in some characteristics. Relevance indicates some similarities between concepts without coincidence of the characteristics. Hence, similarity is just a special aspect of relevance. For example, "baby" is closely relevant to "breast milk", but they are not similar; "airplane" is similar to "train" functionally, but they are not closely relevant. Similar entity is usually regarded as relevant because of their similarity such as "fixed-line telephone" and "mobile phone". Completely irrelevant entity can not be similar. We can use a number between 0 and 1 to measure the degree of the relevance with 0 representing non-relevant and 1 representing the most relevant.

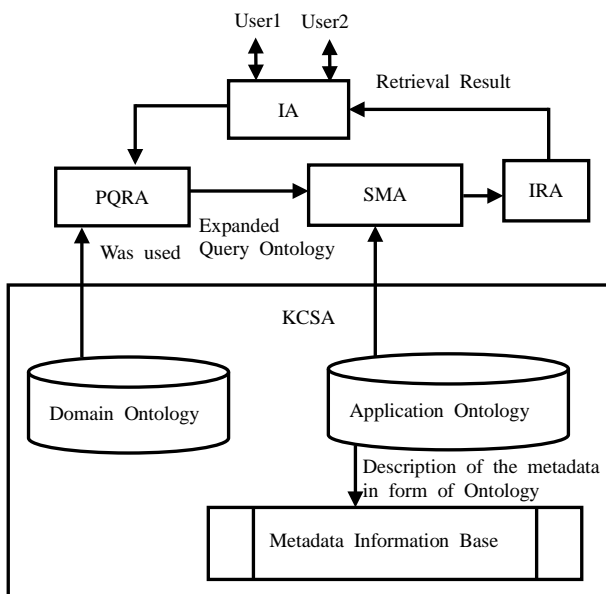


그림 4. 온톨로지 기반 멀티에이전트 정보검색시스템의 프레임워크.

Fig. 4. The framework of the multi-agent information retrieval system based on Ontology tree.

This paper adopts Hirst-St-Onge's semantic relevance method [12,13] to calculate the degree of relevance. The main idea is: if two concepts in the synset of WordNet have a short connected path, they have high semantic relevance; while there is no such path, $Rel_{HS}(C_i, C_j) = 0$. The function $Rel_{HS}(C_i, C_j)$ is defined as Equation (1):

$$Rel_{HS}(C_i, C_j) = m - len(C_i, C_j) - n \times turn(C_i, C_j) \quad (1)$$

Here, C_1 and C_2 are conceptual collection of the ontologies O_1 and O_2 ; $C_i \in C_1, C_j \in C_2$; m and n are two constant parameters; $turns(C_i, C_j)$ indicate the number of the path diversions in the synset and $len(C_i, C_j)$ is the length of the path.

2.2 Similarity Calculation of Concept Semantics

The semantic similarity of concept refers to the calculation of the similarity through the linguistic features of the concept. Semantic relevance is proportional with semantic similarity. This paper adopts the method based on the WordNet to calculate the degree of similarity of the concept. The function $SS(C_i, C_j)$ is defined as Equation (2):

$$SS(C_i, C_j) = -\log \frac{(len(C_i, C_j) + 1)}{2 \times depth} \quad (2)$$

Here, $len(C_i, C_j)$ refers to the length of the shortest path between the synset; $depth$ is the height of the classifier tree. When two words have the same meaning, $len(C_i, C_j) = 0$, $depth = 1$, $SS(C_i, C_j) = 1$.

2.3 Similarity Calculation of Concept Name and Concept Attribute

The similarity calculation of concept name refers to considering only the linguistic similarity of two concepts, not the similarity of the semantics of the concept.

The similarity calculation of the concept attribute refers to the consideration of the similarity of the concept attribute from the name and the data type. If the attributes of the concepts or the domain and the range of the concept attribute are similar, the concepts are similar.

The concept name, the name of the concept attribute and the data type of the concept attribute are all text types (string), so that we can use string similarity calculation method—"edit distance", to calculate the similarity of the concept name and concept attribute. Edit distance [14], also called Levenshtein distance, can be used to calculate the similarity between two strings. Given two character strings s_1 and s_2 , the Levenshtein distance between them is the minimum number of edit operations required to transform s_1 into s_2 . Most commonly, the Levenshtein operations allowed for this purpose are: (i) insert a character into a string; (ii) delete a character from a string and (iii) replace a character of a string by another character.

The function $SN(C_i, C_j)$ used to calculate the similarity of concept name is defined as Equation (3):

$$SN(C_i, C_j) = \frac{1}{e^{\frac{ed(C_i, C_j)}{C_i.len + C_j.len - ed(C_i, C_j)}}} \quad (3)$$

Assume N_1 and N_2 are the names of the Attribute of C_1 and C_2 respectively, and $N_i \in N_1, N_j \in N_2$, then the function $SAN(N_i, N_j)$

used to calculate the similarity of concept attribute name is defined as Equation (4):

$$SAN(N_i, N_j) = \frac{1}{e^{\frac{ed(N_i, N_j)}{N_i.len + N_j.len - ed(N_i, N_j)}}} \quad (4)$$

Assume D_1 and D_2 are the data type of the Attribute of C_1 and C_2 respectively, and $D_i \in D_1, D_j \in D_2$, then the function $SAD(D_i, D_j)$ used to calculate the similarity of concept attribute data type is defined as Equation (5):

$$SAD(D_i, D_j) = \frac{1}{e^{\frac{ed(D_i, D_j)}{D_i.len + D_j.len - ed(D_i, D_j)}}} \quad (5)$$

Here the "ed(.)" refers to the Levenshtein distance of the strings and "Len" refers to the length of the strings.

3. The Improved Semantic Similarity Matching Algorithm (ISSMA) Based on the Ontology Tree

Step1: Analysis of ontology.

The analysis of ontology aims to parse the ontology files into tree structure according to the hierarchical structure of the concept, each node of the tree being a class, or attributes of the ontology. The Classes Node of the ontology tree has the same ancestor node—"Class", and the Attribute Node of the ontology tree has the same ancestor node—"Property". During the process of analysis, we should remove the public prefix and suffix of the Class Name or Attribute Name.

Step2: Up-Down Matching Process according to the Ontology Tree.

Step2.1: Initialization of Matching Set.

Firstly, find the node sets N_s belonging to query ontology and N_a belonging to application ontology. For all the $En_s \in N_s$, $En_a \in N_a$, the immediate predecessor of En_s is the classified root node of the query ontology ($Parent(En_s)=Class_s$), and the immediate predecessor of En_a is the classified root node of the application ontology ($Parent(En_a)=Classst$).

Step2.2: Match the elements from N_s and N_a to accomplish the first level of the matching task of the ontology tree.

① Calculate the Relevance of Concept

Adopt the Equation (1) to calculate the relevance of two concepts. If the relevance of C_i and C_j satisfies that $Rel_{HS}(C_i, C_j) \geq T_1$, then C_i and C_j are a pair of relevant matching. Here T_1 is a preset threshold.

② Calculate the Similarity of Concept

If C_i and C_j are a pair of relevant matching, then use the Equation (6) to calculate the similarity of the two concepts.

$$S(C_i, C_j) = \alpha \times SS + \beta \times SN + \delta \times SAN + \gamma \times SAD \quad (6)$$

Here

$$\alpha = \frac{SS}{SS + SN + SAN + SAD} \quad \beta = \frac{SN}{SS + SN + SAN + SAD}$$

$$\delta = \frac{SAN}{SS + SN + SAN + SAD} \quad \gamma = \frac{SAD}{SS + SN + SAN + SAD}$$

If the similarity of C_i and C_j satisfies that $S(C_i, C_j) \geq T_2$ and $S(C_i, C_j) = \text{Max}(S(C_i, C_1), S(C_i, C_2), S(C_i, C_3), \dots, S(C_i, C_m))$ then C_i and C_j are a pair of similar matching. Here m is the number of the elements of N_a , $j \leq m$, and T_2 is a preset threshold.

Step2.3: Select child nodes of matching pair.

Assume that C_i and C_j have been decided as a similar matching pair, then narrow the matching scope to the subtree of C_i and C_j to get the new sets of N_s and N_a , where $Parent(Ens) = C_i$, $Parent(Ena) = C_j$.

Execute step2.2 again to select the new matching pairs from N_s and N_a . Execute the step2.3 repeatedly, until no matching pairs can be selected from the N_a and N_s .

Step2.4: End.

Step3: Match again to find all the matching pairs.

In the former I have mentioned that the diversity of partition granularity may lead to the structural diversity of ontology tree, thus it is likely to result in the missed matching during the process of ontology matching. Therefore, it is not enough to only match the ontology according to the ontology tree strictly. We should make full use of the existing matching pairs that have been decided as legal matching pairs to find the concept not matched and match them again. The process is as follows:

If the concept N_s belonging to query ontology and N_a belonging to application ontology satisfies $S(N_s, N_a) \geq T_3$ (T_3 is a preset threshold) and N_s and N_a have not been matched, then specify the screening range as the following two subtrees, the root node of one subtree is the ancestor node (N_s') of N_s . The root node of another subtree is the ancestor node (N_a') of N_a , and the N_s' N_a' must be a legal matching pair.

During this process, all the root nodes of the class node—Classs and Classt, and all the root nodes of the attribute node—Propertyt and Property are regarded as legal matching pairs (Virtual).

Step4: End.

4. The Overall Algorithm of the Multi-Agent Information Retrieval System Based on Ontology Tree
The overall algorithm is as Fig. 5.

III. SIMULATION RESULTS

With CPU-- Inter Core2, 2 G Memory, Java as programming language and Eclipse integrated development environment, this study analyzes and operates the ontology files by means of open source framework “Jena”.

In order to evaluate the performance of our proposed Semantic Similarity Matching Algorithm based on ontology tree, this study adopted a subset of the Reuter’s RCV1 corpus [15] used in TREC-11 [16]. The corpus consists of about 800,000 new stories from year 1996-1997, but we only use the items between 1997-6-20 to 1997-7-20. Due to the wide range of topics in the corpus, this study selected about 25 business related topics for our simulation which correspond to a collection of about 8000 business new documents ranging from economic espionage to commodity trading. A business domain ontology tree is constructed containing 100 business related concepts, used in the experiment as the domain ontology. A prototype system is also implemented for semantic information retrieval framework discussed above. In our experiment, a set of random 80 queries are executed with our prototype system.

We compare the results of the multi-agent information retrieval system (OTMAIRS) based on ontology tree with those of the SST (Simple Semantic based Technique) [17] which uses traditional

```

1. IA accepts the query input by users in form of IA-Q( $q_1, q_2, \dots, q_m$ )
2. QR refines the query in form of QR-Q( $O_1, O_2, \dots, O_n$ )
3. SMA adopts ISSMA to realize the similarity matching between two ontologies
   Step1. Analyze ontology
   Step2. Up-Down Matching Process according to the Ontology Tree
   For  $Ens \in N_s$  do  $Parent(Ens)=Classs$ ;
   For  $Ena \in N_a$  do  $Parent(Ena)=Classt$ ;
   SmilaMat( $C_i, C_j$ )
   { If  $(Rel_{HS}(C_i, C_j) \geq T_1)$ , then
   { If  $((S(C_i, C_j) \geq T_2)$  and  $(S(C_i, C_j) = \text{Max}(S(C_i, C_1), S(C_i, C_2), S(C_i, C_3), \dots, S(C_i, C_m))))$  then
     {  $C_i$  and  $C_j$  are a pair of similar matching;
       Add ( $C_i, C_j$ ) to the matching set S; }
   }
   }
   Do
   { S1=S;
     For (subtree of  $C_i$ ) and (subtree of  $C_j$ ) do
       { If  $Parent(Ens) = C_i$  then  $Ens \in N_s$ ;
         If  $Parent(Ena) = C_j$  then  $Ena \in N_a$ ;
           SmilaMat ( $Ens, Ena$ ); }
          $C_i = Ens; C_j = Ena;$ 
       } While ( $S1 \triangleleft S$ )
   }
Step3. Match again to find all the matching pairs.
For ( $S(N_s, N_a) \geq T_3$ ) and  $(N_s, N_a) \cap S = \Phi$ )
{ If (root node “RN1” of subtree1 = ancestor node ( $N_s$ )) and (root node “RN2” of subtree2 = ancestor node ( $N_a$ )) then
  Add (RN1, RN2) to the matching set S;
}
4. IRA using the matching concept SMA-Q( $AO_1, AO_2, \dots, AO_n$ ) to retrieve the information from the web
5. IRA returns the retrieval result back to the IA
    
```

그림 5. 온톨로지 기반 멀티에이전트 정보검색시스템의 전체 알고리즘.

Fig. 5. The overall algorithm of the multi-agent information retrieval system based on Ontology Tree.

VSM (Vector Space Model), also called term frequency (tf) inverse document frequency (idf) to carry out searching, and evaluate the performance of the two methods from the precision and recall. The Precision (P) is the fraction of retrieved documents relevant which is defined as Equation (7). Recall (R) is the fraction of relevant documents retrieved which is defined as Equation (8), and Equation (9) is used to calculate the precision improvement of our method compared with the simple semantic based technique.

$$Precision = \frac{\#(\text{relevant items retrieved})}{\#(\text{retrieved items})} = P(\text{relevant}|\text{retrieved}) \tag{7}$$

$$Recall = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})} = P(\text{retrieved}|\text{relevant}) \tag{8}$$

$$IMP = \frac{OTMAIRS.Precision - SST.Precision}{SST.Precision} \times 100\% \tag{9}$$

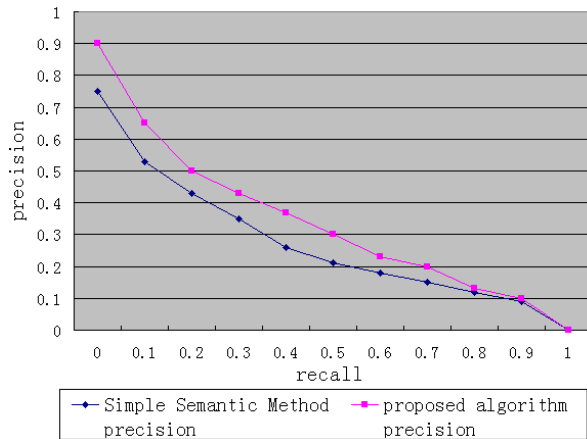


그림 6. 5000개 문서에 대한 50개 쿼리의 평균 11-포인트 정 확률/재현율 그래프

Fig. 6. Averaged 11-point precision/recall graph across 50 queries for 5000 documents.

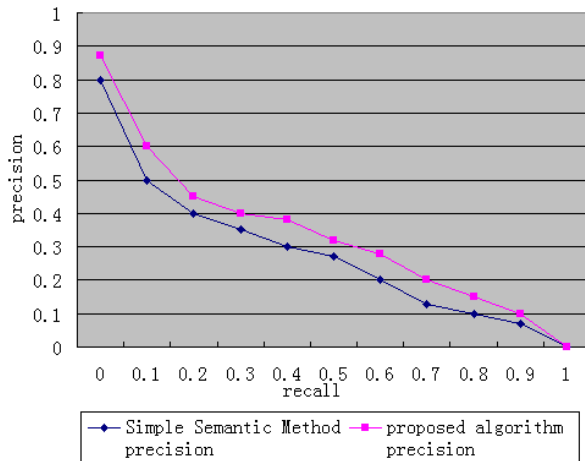


그림 7. 8000개 문서에 대한 80개 쿼리의 평균 11-포인트 정 확률/재현율 그래프

Fig. 7. Averaged 11-point precision/recall graph across 80 queries or 8000 documents.

Shown as Fig. 6 (5000 documents, 50 random queries) and Fig. 7 (8000 documents, 80 random queries) using eleven-point interpolated average precision, we can see that for the same recall, the average IMP of our method is about 24.75 % with 50 random queries for 5000 documents, and about 28.74 % with 80 random queries for 8000 documents, i.e., our method is more effective for large amount of information retrieval.

IV. CONCLUSION

The traditional ontology matching methods may lead to considerable computation redundancies, as well as misjudge the small concept to be a similar matching under the condition when the larger one is not matched. This paper proposes a new multi-agent up-down semantic similarity matching algorithm based on the ontology tree. First, we build a Multi-Agent Information Retrieval System Based on Ontology Tree composed of Interface Agent, Query Refinement Agent, Semantic Matching Agent and Knowledge Collecting and Storing Agent; Second we make use of

the domain ontology base to construct the application ontology base of the information resources (metadata information) to realize the representation and the storage of the knowledge, and use RDF triple, the subject, predicate and object of RDF triple to enable the Metadata to concentrate on the combination of concept and their relationship similarity at the same time; third, in order to reduce the computation redundancies to the minimum, this paper comprehensively considers the relevance of the concept and the similarity of the concept semantic, concept name and concept attribute which are used as measurement to decide whether or not ontology is matched. Simulation shows that the new algorithm can improve the precision and flexibility of the similarity matching compared with the traditional semantic similarity matching algorithm.

REFERENCES

- [1] X. Li, F. Bian, H. Zhang, C. Diot, R. Govindan, and G. Iannaccone, "MIND: a distributed multi-dimensional indexing system for network monitoring," *IEEE Infocom-06 Barcelona*, Apr. 2006.
- [2] A. Ntoulas, G. Chao, and J. Cho, "The infocious web search engine: improving web searching through linguistic analysis," *International World Wide Web Conference Committee (IW3C2) ACM, Chiba Japan*, May 2005.
- [3] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, May 2001.
- [4] L. Wu, J. P. Feng, and Y. Luo, "A personalized intelligent web retrieval system based on the knowledge-base concept and latent semantic indexing model," *2009 Seventh ACIS International Conference on Software Engineering Research, Management and Applications*, pp. 45-50, Dec. 2009.
- [5] W.-D. Fang, L. Zhang, Y.-X. Wang, and S.-B. Dong, "Towards a semantic search engine based on Ontologies," *IEEE Proceedings of the Fourth International Conference on Machine Learning and Cybernetics Guangzhou China*, pp. 1913-1918, Aug. 2005.
- [6] G. Varelas, E. Voutsakis, and P. Raftopoulou, "Semantic similarity methods in WordNet and their application to information retrieval on the web," *7th ACM International Workshop on Web Information and Data Management*, pp. 10-16, Nov. 2005.
- [7] K. Saruladha, Dr. G. Aghila, and S. Raj, "A new semantic similarity metric for solving sparse data problem in ontology based information retrieval system," *IJCSI International Journal of Computer Science Issues*, vol. 7, no. 11, pp. 40-48, May 2010.
- [8] J. Mustafa, S. Khan, and K. Latif, "Ontology based semantic information retrieval," *2008 4th International IEEE Conference Intelligent Systems*, vol. 3, pp. 2214-2219, Sep. 2008.
- [9] Chuan li, "Research on an efficient ontology matching algorithm," Chongqing University, 2011.
- [10] Q. Gao and Y. I. Cho, "A multi-agent information retrieval system based on Ontology," *12th International Conference on Intelligent Autonomous System*, Jun. 2012.
- [11] A. Maedche and S. Staab, "Measuring similarity between Ontologies," *Proc of European Conference on Knowledge Acquisition and Management, London: Springer-Verlag*, pp. 251-263, 2002.
- [12] A. Budanitsky and H. G. Evaluating, "WordNet-based measures

oflexical semantic relatedness,” *Computational Linguistics*, vol. 32, no. 1, pp. 13-47, 2006.

- [13] P. Pantel and L. Dekang, “Discovering word senses from text,” *Proc of ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York:ACM Press, pp. 613-619, 2002.
- [14] P. Bouquet, J. Euzenat, E. Franconi, L. Serafini, G. Stamou, and S. Tesseractis, “Specification of a common framework for characterizing alignment,” *Knowledge Web Deliverable*, 2004.
- [15] Reuters corpus volume 1. <http://about.reuters.com/researchandstandards/corpus/>
- [16] TREC 2002 Collections. <http://trec.nist.gov/data/>
- [17] D. Bonino, F. Corno, L. Farinetti, and A. Bosca, “Ontology driven semantic search,” *WSEAS Transaction on Information Science and Application*, vol. 1, pp. 1597-1605, Dec. 2004.



Qian Gao

2001년 Shandong University of Science and Technology 학사. 2008년 Shandong University 석사. 2012년~현재 수원대학교 컴퓨터학과 박사과정. 관심영역은 신경회로망, 인공지능, 정보검색, 유비쿼터스



조 영 임

1994년 고려대학교 컴퓨터학과 박사. 1996년 삼성전자 선임연구원. 2000년~현재 수원대학교 컴퓨터학과 교수. 관심영역은 지능시스템, 유비쿼터스, 인공지능, 정보검색.