ICIC Express Letters ICIC International ©2016 ISSN 1881-803X
Volume **10**, Number **5**, May **2016** pp. 1101–1108

AN ONTOLOGY EVOLVING-DRIVEN PERSONALIZED SEMANTIC SEARCH SYSTEM

Yi Liu, Yu Wang and Deli Yang

Department of Management and Economics
Dalian University of Technology
No. 2, Linggong Road, Ganjingzi District, Dalian 116023, P. R. China
yiliu@mail.dlut.edu.cn; { ywang; somdyang }@dlut.edu.cn

Received October 2015; accepted January 2016

Abstract. *Semantic search based on ontology is a relatively new promising search technology in Web search. The quality of search result is crucial for the success of a search system and depends mainly on the domain ontology in semantic search. However, the domain ontologies usually fail to be updated in time and even in some cases still stay in the initial state that does not contain enough semantics to provide the powerful support for semantic search as the Web evolves continuously. As a result, the quality of search result was often poor. This paper proposes an ontology evolving-driven semantic search system that combines the automatic ontology evolution with semantic search. The aim of this modification is to efficiently enhance the performance of the semantic search by periodically automatic ontology evolution, which can enrich the domain ontology by adding more semantics into it. Furthermore, an algorithm of personalized ranking based on ontology is presented in this paper in order to ensure achieving the high-quality ranking results. The performance of the proposed system is evaluated with three experiments, and the experimental results show that the improved semantic search system has a much better performance than the one without ontology evolution.*

Keywords: Semantic search, Ontology evolution, Personalized ranking, Domain ontology, Semantic analysis

1. Introduction.

Efficient search is one such major envisioned application of this next generation Web popularly known as Semantic Web that brings structure to the meaningful content of Web pages [1]. The ontology is used as a standard knowledge representation for the Semantic Web and applicable to automatic reasoning, knowledge representation and reuse [2], which typically consists of a number of classes, concepts, relations, instances, and axioms [3]. The ever growing amount of ontology-based semantic mark-up in the Web provides an opportunity to start working in the direction of a new generation of open intelligent applications [4].

To the best of our knowledge, none of existing ontology-based semantic search systems deals with the automatic evolving of domain ontology [5] from search results. Most of them focus on constructing ontologies or introducing ready-made domain ontologies from other domains. If there is a need of updating the ontology, another one tries to achieve the objective by manually evolving ontologies [6-10] or integrating the existing ontology with the other ontologies from the same domain [11,12]. The efficiency and timeliness of manually evolving ontologies are often unsatisfactory. In addition, during the manually evolving relatively large scale ontologies, artificial mistakes in semantic relations occur anytime. Integrating ontologies will inevitably bring about the heterogeneity of ontologies, which will lead to the inconsistency of semantics serving as the cause of poor-quality search results even wrong in certain cases. In fact, the goal of these systems mentioned above is to improve or correct ontology from the combination of other ontologies but not to evolve a domain ontology by using new knowledge of the domain.

In this paper, we propose an ontology evolution-driven semantic search system named DLOSSS by introducing automatically evolving domain ontologies that is effectively combined with semantic search. The main idea behind the modification is to provide the stronger semantic support for semantic search in order to obtain more relevant search results. As the quality of the result ranking is centered in a search system, we developed a personalized ranking algorithm to ensure the high-quality ranking results.

The remaining part of this paper is structured as follows. Section 2 explains in detail the procedures of the main modules in DLOSSS. Section 3 discusses the experimental results taken and, finally, Section 4 closes this article with concluding remarks.

2. The Key Technologies.

The DLOSSS includes several key technologies, such as the user interest model, semantic analysis process, personalized ranking, and automatic ontology evolution.

2.1. The user interest model.

The user interest model is used to determine individual users' intentions, which poses an important challenge to personalization and information filtering. The model captures the semantics of the interest of a user in the domain ontology. We call this model the user contextual interest model.

The model is defined as

$$\boldsymbol{U} = (\boldsymbol{P}_d, \boldsymbol{H}_y, \boldsymbol{L}_k, \boldsymbol{C}_s, \boldsymbol{R}_s, \boldsymbol{F}_n) \tag{1}$$

where $\boldsymbol{U}$ is the context of the user session; $\boldsymbol{P}_d$ is the personal data of the user, including profiles, major, preferences, and so on. Long-term actions are stored in a history log and are collected and ranked according to user context, and any changes are automatically updated in the context; $\boldsymbol{H}_y$ is the historical context indicated in the user log; $\boldsymbol{L}_k$ refers to links to the context; $\boldsymbol{C}_s$ and $\boldsymbol{R}_s$ denote the joint sets of concepts and semantic relations, respectively, in the domain ontology; and the function $\boldsymbol{F}_n$ represents $\boldsymbol{C}_s \times \boldsymbol{C}_s \rightarrow \boldsymbol{R}_n$, which is a special semantic relation of concepts.

2.2. The semantic analysis process.

The essence of the semantic analysis is that each word of the text is assigned an autonomous agent capable of negotiating with other similar agents about the meaning of each word in the sentence and its general meaning on the basis of domain ontology.

During negotiations the word agents can speculate about the possible word meanings and their semantic relations, find and resolve meaning conflicts, detect implicit information on the basis of domain knowledge, take into account the context of the word usage within one sentence and inter-phrasal context thus connecting the words of various sentences into one semantic network composed of descriptors that contain formal monosemantic description of the initial text meaning and simple to compare similarity of information they contain on the base of the ontology, as described in [13].

The algorithm of comparing semantic descriptors was developed for semantic vector matrix. This algorithm is based on finding in one of the descriptors the sub-network which is close to the network of the other descriptor as much as possible. Similarity degree of two sub-networks is defined as similarity degree of their respective pairs of nodes. Similarity degree of two nodes depends on relative position of corresponding concepts in the ontology and on the values of attributes connected with nodes under comparison.

In DLOSSS, we use the classical Vector Space Model (VSM) to construct a relational matrix of concepts. In this matrix, the rows represent the classes as m, and the columns represent the KUTs. Each entry is a submatrix of classes with the same meaning as the original classes.

Singular Value Decomposition (SVD) is a well-known method for decomposing matrices. In DLOSSS, we apply SVD to the semantic matrix $\boldsymbol{S}_{m\times n}$ to decompose it into three matrices:

$$\boldsymbol{S}_{m\times n} = \boldsymbol{U}_{m\times d}\boldsymbol{\Sigma}_{d\times d}\boldsymbol{V}_{d\times n} \tag{2}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are two orthogonal matrices; d is the rank of matrix $\boldsymbol{S}$; and $\boldsymbol{\Sigma}$ is a diagonal matrix of size $d \times d$, with diagonal entries that are the singular values of matrix $\boldsymbol{S}$ and are stored in decreasing order according to descriptors comparison algorithm. SVD supplies the best lower rank approximation of matrix $\boldsymbol{S}$. We can then reduce the matrix $\boldsymbol{\Sigma}$ into the k largest rank matrix $\boldsymbol{\Sigma}_{k\times k}$ by keeping the k largest values.

We then derive $\boldsymbol{U}'$ from $\boldsymbol{U}$ and $\boldsymbol{V}'$ from $\boldsymbol{V}$ to obtain $\boldsymbol{S}' = \boldsymbol{U}' \times \boldsymbol{\Sigma}' \times \boldsymbol{V}'$, the rank-$k$ approximation of the original matrix $\boldsymbol{S}$. $\boldsymbol{U}'$ is constructed by the first k columns of $\boldsymbol{U}$ according to the k highest-order singular values. The semantic attribute matrix $\boldsymbol{S}'$, where each item is represented by k latent variables, is smaller than the original n. As a result, this matrix is less sparse than $\boldsymbol{S}$ and thus significantly improves similarity computations and lowers the related cost of computation.

We propose a personal ranking algorithm to further improve the quality of the result ranking.

We use the classical Vector Space Model (VSM) to represent the query:

$$\boldsymbol{q} = \left(q_{1,t} : w_1^U, q_{2,t} : w_2^U, q_{3,t} : w_3^U, \ldots, q_{n,t} : w_n^U\right) \tag{3}$$

$$\boldsymbol{q}_{n,t} = (q_{n,1}, q_{n,2}, q_{n,3}, \ldots, q_{n,t}) \tag{4}$$

where $\boldsymbol{q}$ represents the vector of all keywords in the query and the corresponding homonyms, and $\boldsymbol{q}_{n,t}$ represents the vector of the nth keyword and the corresponding synonyms. The w_i^u is the personalization weight of keyword $\boldsymbol{q}_{i,t}$ in the vector $\boldsymbol{q}$ according to the group level:

$$w_i = \frac{freq(q_{i,t})}{\sum_{j=1}^{n} freq(q_{j,t})} \tag{5}$$

where *freq*$(q_{i,t})$ represents the frequency of the keyword $\boldsymbol{q}_{i,t}$ and its relevant synonyms in the group level of the user.

In order to compute the similarity of a KUT to query $\boldsymbol{q}$, the query is represented by a vector:

$$\boldsymbol{S} = \left(q_{1,t} : w_1^S, q_{2,t} : w_2^S, q_{3,t} : w_3^S, \ldots, q_{n,t} : w_n^S\right) \tag{6}$$

$$\boldsymbol{q}_{n,t} = \left(q_{n,1} : w_{n,1}^S, q_{n,2} : w_{n,2}^S, \ldots, q_{n,t} : w_{n,t}^S\right) \tag{7}$$

where $\boldsymbol{S}$ represents the vector of the similarities of all keywords to a KUT, $\boldsymbol{w}_n^S$ denotes the weight of the nth keyword in KUT m, $\boldsymbol{q}_{n,t}$ represents the vector of the nth keyword and the corresponding synonymies, and $w_{n,t}^S$ denotes the weight of the synonym t of the nth keyword in KUT m.

The weight $w_{n,i}^S$ of each synonym $\boldsymbol{q}_{n,i}$ in the vector $\boldsymbol{q}_{n,t}$ uses the traditional tf/idf measure [14]:

$$w_{n,i}^S = freq\left(\boldsymbol{q}_{n,i}\right) \times \log\frac{n}{N} \tag{8}$$

where *freq*$(\boldsymbol{q}_{n,i})$ represents the frequency of the $\boldsymbol{q}_{n,i}$ in KUT m, n is the number of the nth keyword and the corresponding synonymies in KUT m, and N is the number of all keywords and the corresponding synonymies in KUT m, including the homonymies. On this basis, we get the weight of the keyword l in KUT m:

$$w_l^S = \sum\nolimits_{i=1}^{t} w_{l,i}^S \quad (l = 1, \ldots, n) \tag{9}$$

Finally, we use the standard cosine similarity computing the similarity of KUT m to query $\boldsymbol{q}$ as

$$sim\left(\boldsymbol{S}_{q,K}, \boldsymbol{q}\right) = \cos(\theta) = \frac{\boldsymbol{S}_{q,K} \cdot \boldsymbol{q}}{|\boldsymbol{S}_{q,K}| \times |\boldsymbol{q}|} \tag{10}$$

2.3. **The ontology evolution process.** The ontology evolving module is composed of a Corpus Analyzer, an MAS, a Coordinator Agent and Protégé [15].

The goal of the *Corpus Analyzer* is to identify relevant candidate terms as well as relevant lexical relations. The **multi-agent system** (MAS) has, as input, the triples returned by the *Corpus Analyzer*. The MAS consists of (*i*) *term* agents that represent the terminological component of the ontology and (*ii*) *concept* agents that represent the conceptual part of the ontology. The creation of agents and their communications are managed by the *Coordinator* Agent. The agentified ontology generated by MAS is finally transformed into domain ontology via *Protégé*.

Term Agent Behaviors

A *term* agent has a status (*term* or *candidate term*) indicating if it is in the ontology or not yet. A *term* agent is connected by a lexical relation to other *term* agents. It must also be connected to one *concept* agent. Each relation between *term* agents is tagged by the confidence of the triples $< T_i$, Rel, $T_j >$. A *term* agent has two objectives.

(1) In order to **denote a *concept* agent**, a *term* agent asks for the creation of a *concept* agent to the *coordinator*. This creation is done if, in the current MAS, a *concept* agent having the same label does not exist. The *coordinator* transmits thereafter the identifier of this new *concept* agent to the *term* agent. Then, the *term* agent sends to the *concept* agent a request for establishing a denotation relation. This request is always accepted by the *concept* agent. The confidence of the denotation link is equal to the greatest value of the lexical relations of the *term* agent.
(2) A *term* agent **processes its outgoing lexical relations**. A lexical relation has a confidence and a status (*not treated, treated or refused*). A *term* agent processes its relations from the most relevant to the less relevant. To do this, a *term* agent sends a request to its *concept* agent in order to transform the lexical relation. A *concept* agent processes the request, and then notifies the *term* agent by a message of acceptance or refusal. The *term* agent updates the status of the processed relation. If the relation is refused, the *concept* agent sends a "refuse" message and the status of the lexical relation will be refused. A *term* agent can request again, later to process the refused relation if its confidence increases. When a *term* agent asks for the management of a synonymy relation, its *concept* agent sends a denotation request to the target *term* agent of this relation. If the confidence of the request is greater than the current denotation link of the target *term* agent, the latter accepts the request, changes its denotation link and notifies the *concept* agent by message of acceptation. The target *term* agent refuses the request in the contrary case. The initial *term* agent is thereafter notified.

Concept Agent Behaviors

A *concept* agent has a status (*concept* or *candidate concept*) indicating if the concept is inherent to ontology or not yet. A *concept* agent is connected by conceptual relations to other *concept* agents and connected by denotation links to other *term* agents. Every relation can have the status (*not treated, treated or refused*). A *concept* agent has two objectives.

(1) A *concept* agent receives requests coming from *term* agents for **processing lexical relations** ❶ (in Figure 1). In order to process a request coming from a *term* agent, the *concept* agent gets the *concept* agent denoted by the *term* agent that is target of this lexical relation. Then it sends a request for establishing a conceptual relation with this *concept* agent ❷. When a *concept* agent receives a request to create a conceptual relation, it can accept or refuse the relation by sending a notification ❸ (it will refuse if it has a stronger conceptual relation). When a *concept* agent receives a notification, it updates the status of the concerned relation and its links with the other *concept* agents ❹. A *concept* agent can propose later a "refused" conceptual relation if its

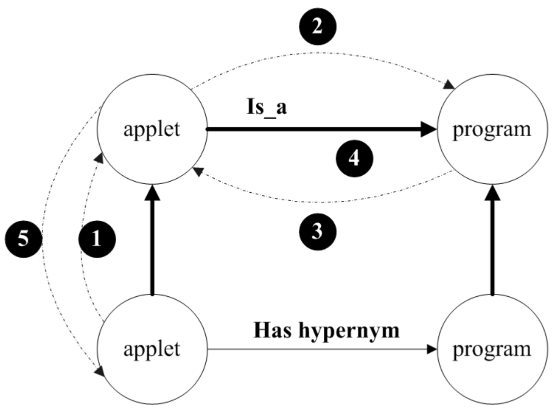
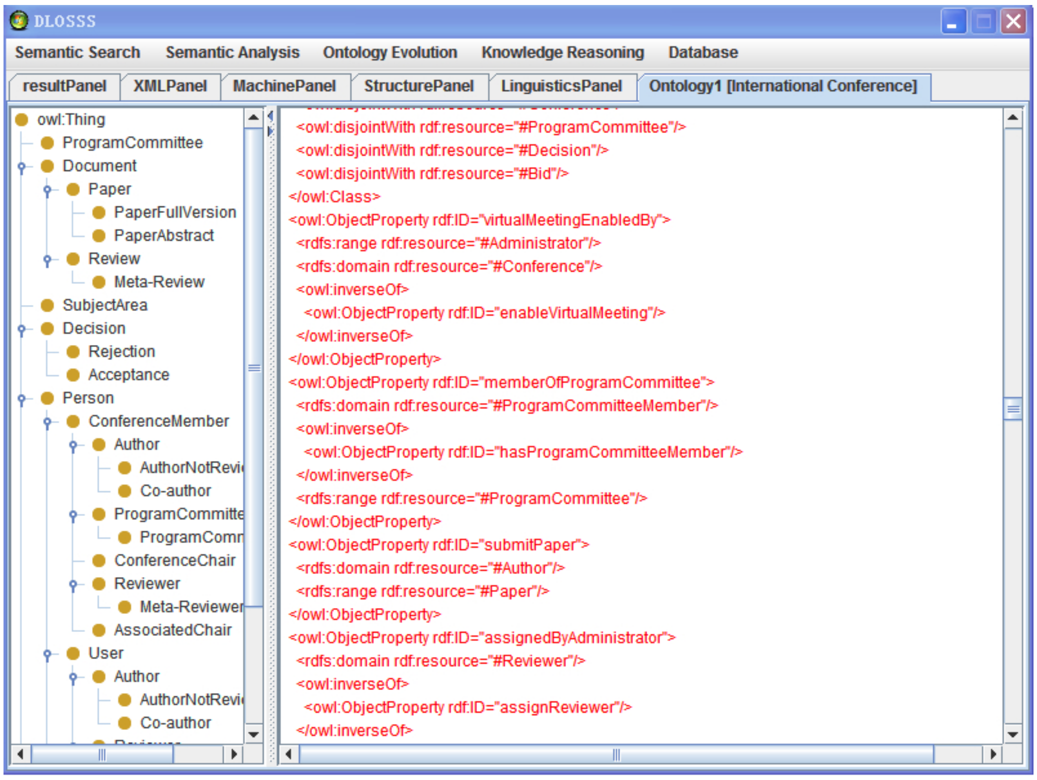


FIGURE 1. The demonstrating to establish a certain relation



```
owl:Thing
  ProgramCommittee
  Document
    Paper
      PaperFullVersion
      PaperAbstract
    Review
      Meta-Review
  SubjectArea
  Decision
    Rejection
    Acceptance
  Person
    ConferenceMember
      Author
        AuthorNotRevi
        Co-author
      ProgramCommitte
        ProgramComr
      ConferenceChair
      Reviewer
        Meta-Reviewer
      AssociatedChair
    User
      Author
        AuthorNotRevi
        Co-author
```

```
 <owl:disjointWith rdf:resource="#ProgramCommittee"/>
 <owl:disjointWith rdf:resource="#Decision"/>
 <owl:disjointWith rdf:resource="#Bid"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="virtualMeetingEnabledBy">
 <rdfs:range rdf:resource="#Administrator"/>
 <rdfs:domain rdf:resource="#Conference"/>
 <owl:inverseOf>
  <owl:ObjectProperty rdf:ID="enableVirtualMeeting"/>
 </owl:inverseOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="memberOfProgramCommittee">
 <rdfs:domain rdf:resource="#ProgramCommitteeMember"/>
 <owl:inverseOf>
  <owl:ObjectProperty rdf:ID="hasProgramCommitteeMember"/>
 </owl:inverseOf>
 <rdfs:range rdf:resource="#ProgramCommittee"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="submitPaper">
 <rdfs:domain rdf:resource="#Author"/>
 <rdfs:range rdf:resource="#Paper"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="assignedByAdministrator">
 <rdfs:domain rdf:resource="#Reviewer"/>
 <owl:inverseOf>
  <owl:ObjectProperty rdf:ID="assignReviewer"/>
 </owl:inverseOf>
```

FIGURE 2. The performance of enhanced semantics in domain ontology after evolving

confidence evolves. The initial term agent of the lexical relation is notified thereafter by the *concept* agent that established the conceptual relation ❺.

(2) A *concept* agent must have a preferred label. This label is the label of the *term* agent that is connected to it and has the greatest confidence value. This label can evolve if new *term* agents denote the *concept* agent or if the confidence of the denotation links has evolved.

3. Experimental Results and Discussion.

This section evaluates the efficiency of the improved semantic search system by comparing it with the one without ontology evolution and tests the performance of result ranking as well as the precision and recall rate.

We took a ready-made ontology "International Conference" as the domain ontology in experiment, and allowed the DLOSSS to perform the first ontology evolution, as shown in Figure 2.

Initially this ontology was composed of 382 concepts, and 14 concepts resided in inappropriate places that arise mainly from the inappropriate relations to other concepts. After evolving, there is a total of 469 concepts inherent to the ontology including 8 inappropriate concepts. The number of concepts in the ontology was 23% higher than that in the initial ontology. It is interesting to notice that 7 out of 14 inappropriate concepts were corrected during evolving ontology due to the added relations in corpus. The processing time of semantic analysis alone is 31s, and that of semantic analysis and ontology evolution together is 37s. In other words, the response time of the DLOSSS is 6s more than that of the system without evolution. In contrast to the significantly enhanced semantics, the increased cost of runtime is relatively smaller and accepted thanks to the parallel run of the two processes.

To further assess the performance of the DLOSSS, we selected two ready-made ontologies, "International Conference" and "The Internet of Things", and one manual-made ontology, "Wine", as the domain ontologies in tests. Table 1 summarizes the results of the three tests in terms of all concepts in ontology ($\mathbf{C_{all}}$), right concepts in ontology ($\mathbf{C_{right}}$), all retrieved web-pages ($\mathbf{P_{all}}$), the related web-pages ($\mathbf{P_{related}}$), the number of appropriate ranking ($\mathbf{R_{appropriate}}$), and the corrected number of inappropriate ranking in last time ($\mathbf{R_{corrected}}$).

Table 1. The performance in main aspects after domain ontologies evolving

Domain	Before Evolving					After Evolving					
Ontology	C_{all}	C_{right}	P_{all}	$P_{related}$	$R_{appropriate}$	C_{all}	C_{right}	P_{all}	$P_{related}$	$R_{corrected}$	$R_{appropriate}$
International conference	382	368	1236	1175	613	523	517	1429	1392	106/562	871
The Internet of things	1138	1115	2015	1927	869	1662	1643	2363	2315	232/1058	1245
Wine	103	103	632	609	249	389	385	870	853	58/360	477

First of all, the system performed the semantic search process and the semantic analysis process without evolution (i.e., identical to the operations made by the system without evolution). For the sake of the reality and the equity the DLOSSS was then implemented to make 5 times of evolving of domain ontologies in which the queries autonomously expressed by experimental participants with different backgrounds differ from the query in first time (the time without evolution). Finally, the DLOSSS fulfilled all functions using the same query as first time. The experimental results of the first time and the last time are shown in Table 1. The results make it clear that the DLOSSS has a better performance than the system without ontology evolution on all tested cases. In particular, the quality of result ranking after evolving shows a much better performance than the system without evolution. To assess the precision and recall rate, Figure 3 is made. The number of related pages is used here instead of recall rate because of failing to directly calculate the recall rate in Web search.

The line graph in Figure 3 indicates that there is continuous increment of precision on the whole during the ontology evolution. This proves that the periodically automated ontology evolution will enhance the performance of the precision rate further. We can notice that the precision rates have noticeable improvements after the first evolving because the initial ontologies were not evolved in long time before this experiment that seriously affected the precision rate owing to the poor semantics in ontologies. On the other hand, the histogram (bar chart) in Figure 3 shows that the number of relevant pages retrieved apparently increased after each evolution. It is beyond question that the semantic enhancements make a great contribution to the recall rate.

Summarizing, the experimental results confirm that the semantic enhancement can successfully improve the quality of the result ranking and the precision and recall rate.

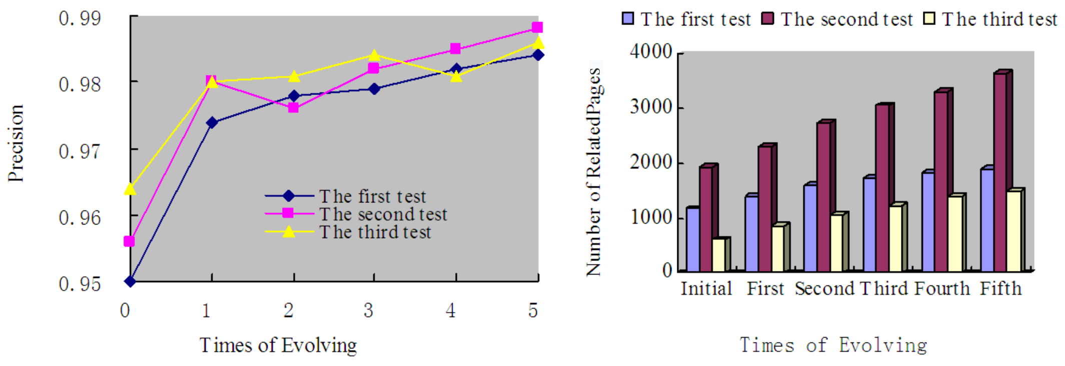


FIGURE 3. The precision rate and recall rate

It also proves our claim that the proposed DLOSSS system outperforms the one without ontology evolution in terms of the result ranking and the precision and recall rate with the help of ontology evolution.

4. Conclusions and Future Work.

This paper focuses on the problem that inadequate semantic can lead to the poor search results in semantic Web search. The study makes the following contributions: 1) the personalized search results by extracting personal data of the user (preferences, profiles, etc.); 2) a novel result ranking algorithm to ensure the high-quality ranking results; 3) the automated ontology evolution to enrich the semantics in domain ontology for better search results. The experimental results confirm the superior performance of the proposed DLOSSS system when compared to the one without ontology evolution in search results.

REFERENCES

[1] T. B. Lee, J. Hendler and O. Lassila, *The Semantic Web*, Scientific American, pp.34-43, 2001.

[2] H. B. Sun, W. H. Fan and W. M. Shen, Ontology fusion in high-level-architecture-based collaborative engineering environments, *IEEE Trans. Syst., Man, Cybern., Syst.*, vol.43, no.1, pp.2-13, 2013.

[3] M. Nagy and M. Vargas-Vera, Multiagent ontology mapping framework for the semantic web, *IEEE Trans. Syst., Man, Cybern., Syst.*, vol.41, no.4, pp.693-704, 2011.

[4] V. Jindal, S. Bawa and S. Batra, A review of ranking approaches for semantic search on web, *Inf. Processing and Management*, vol.50, no.2, pp.416-425, 2014.

[5] L. Li and T. Li, An empirical study of ontology-based multi-document summarization in disaster management, *IEEE Trans. Syst., Man, Cybern., Syst.*, vol.44, no.2, pp.162-171, 2014.

[6] T. P. Sakthi, P. Revathy and C. R. Robin, Design and development of an ontology based personal web search engine, *Proc. of the 2nd Int. Conf. on Communication, Computing and Security*, vol.1, pp.299-306, 2012.

[7] V. S. K. Nagireddi and S. Mishra, An ontology based cloud service generic search engine, *Proc. of the 8th ICCSE*, pp.335-340, 2013.

[8] O. Nasraoui and L. Zhuhadar, Improving recall and precision of a personalized semantic search engine for E-learning, *Proc. of the 4th International Conference on the Digital-Society*, pp.216-221, 2010.

[9] M. Park, K. W. Lee and H. S. Lee, Ontology-based construction knowledge retrieval system, *Ksce Journal of Civil Engineering*, vol.17, no.7, pp.1654-1663, 2013.

[10] W. X. Li and H. Yuan, The research of semantic search engine based on ontology, *Techniques of Automation and Applications*, vol.32, no.6, pp.16-18, 2013.

[11] L. R. Eduardo, G. M. Ignacio and V. G. Rafael, Financial news semantic search engine, *Expert Systems with Applications*, vol.38, no.12, pp.15565-15572, 2011.

[12] X. Chen, H. J. Chen and X. Bi, BioTCM-SE: A semantic search engine for the information retrieval of modern biology and traditional Chinese medicine, *Computational and Mathematical Methods in Medicine*, pp.1-13, 2014.

[13] M. Kanteev, I. Minakov and G. Rzevski, Multi-agent meta-search engine based on domain ontology, *Proc. of the 2nd Int. Workshop on Autonomous Intell. Sys. – Agents and Data Mining*, vol.4476, pp.269-274, 2007.

[14] M. Y. Liu, S. F. Liu and X. Y. Wang, Knowledge-domain semantic searching and recommendation based on improved ant colony algorithm, *Journal of Bionic Engineering*, vol.10, no.4, pp.532-540, 2013.

[15] Z. Sellami and V. Camps, Evaluation of a multi-agent system for the evolving of domain ontologies from texts, *Proc. of the 10th Int. Conf. on Practical Applications of Agents and Multi-Agent Syst.*, vol.155, pp.169-179, 2012.