http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (4): 825-831, 2010
ISSN 1812-5638

A Semantic-Aware Algorithm to Rank Concepts in an OWL Ontology Graph

Yixi Chen, Keting Yin and Xiaohu Yang
College of Computer Science and Technology, Zhejiang University, Hangzhou, 310027, China

Abstract: In order to help person understand complex and large-scale ontologies, we presented a concept ranking algorithm for OWL (Web Ontology Language) ontologies. This algorithm is based on the ontological structure and semantic meanings of concepts/relations. Different from the traditional ranking methods, this algorithm applies semantic meanings (patterns) on concepts and relations. The importance of concepts reinforces one another in an iterative manner; only the semantically correct paths can flood relevant components of importance vector from a concept to its neighbors in ontology graphs. The experimental results show that such a semantically ranking method provides more reasonable ranking result than PageRank-like algorithms.

Key words: Ontologies, ranking algorithm, semantically correct, concept importance, ontology graph

INTRODUCTION

As the semantic web is growing rapidly and going to be popular, lots of large scale and complex ontologies are established and published to describe domain knowledge on the web. It requires much effort for domain experts and researchers to understand those ontologies. Moreover, the ranked concepts of ontologies can assist query result sorting and provide higher query preciseness to users. Although many tools, like IsaViz, Ontoviz and OWLPrefuse, are developed to help ontology understanding with visualization techniques, users still would get lost in a complex graphically presenting. Those tools always display concepts as nodes in directed/undirected graphs, but without any kind of weighting or sorting. This makes the visualized results unreadable when they are huge.

To resolve the problem, some researchers have proposed approaches to rank concepts in ontology. There are mainly two kinds of ranking techniques in semantic web: ontology internal structure oriented and semantic web link structure oriented. The first kind of techniques includes AKTiveRank (Alani *et al.*, 2006), OntoSearch (Jiang and Tan, 2006) and SemSearch (Uren *et al.*, 2008), which is based on internal structure. The second kind of technique includes Ontokhoj (Patel *et al.*, 2003) and ReConRank (Hogan *et al.*, 2006), which is based on the semantic web link structure (Rajapaksha and Kodagoda, 2008). Rajapaksha and Kodagoda (2008) proposed an approach considering both internal structure and the link structure in Semantic Web. Sankar *et al.* (2010) proposed an ontologies ranking method based on OWL language constructs. Comparing to those researches, present study aims to rank the concepts in a given ontology, rather than the whole ontologies in Semantic Web. Many ranking techniques are inspired from PageRank (Brin and Page, 1998), like ontokhoj (Patel *et al.*, 2003) and swoogle (Ding *et al.*, 2004). Hyperlink or link is the only relation to be considered for PageRank-like algorithms. They can not well use implicated meanings on Semantic Web and would not be semantic-aware ranking approaches.

Graves *et al.* (2008) proposed a method to rank nodes in an RDF graph. This approach is based on the idea The more central the node is, easier it is to reach the rest of the graph. They thought a node is more important while it is more central. So their approach maps RDF to undirected graph, change the problem to computing the All-Pairs Shortest Path problem. Zhang *et al.* (2006) described an approach to rank vocabularies by considering both the structure and textual content. The relevance of each vocabulary is got by evaluate the textual score. Wu *et al.* (2008) proposed an algorithm called CARRank to identifying potentially important concepts and relations in an ontology. CARRank maps ontology to directed graph and uses an iterative process to calculate the importance of concepts and relations. All of above approaches map ontology to graph and assign weights to its edges. The semantic meanings of edges are lost in the resulting graph.

Hirst and St-Onge (1997) associated a direction in Upwards (U), Downwards (D) and Horizontal (H) for each relation type and gave three rules to define a semantically correct path in terms of the three directions. It enumerated

Corresponding Author: Yixi Chen, No. 38, Zheda Road, Hangzhou, Zhejiang, 310027, China
Tel: +86-13738196926 Fax: +86-0571 28111775

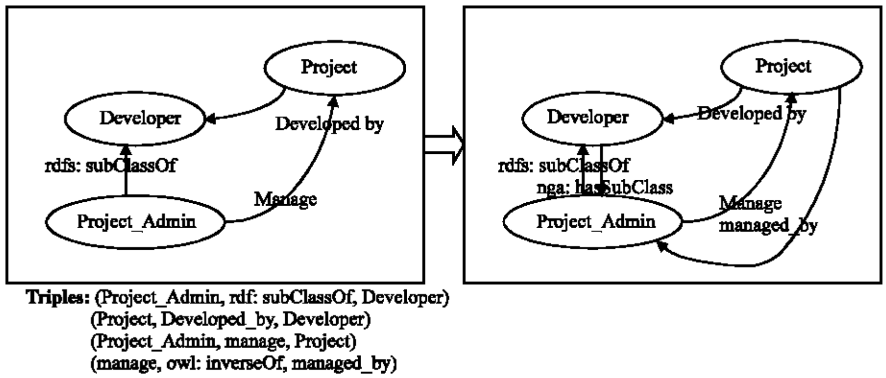

Fig. 1: The inferred ontology graph

only 8 patterns of semantically-correct paths which match their three rules: {U, UD, UH, UHD, D, DH, HD, H}. Present study applies the semantically-correct patterns to concepts ranking and extends it to nine semantically correct patterns in OWL graph.

To the best of our knowledge, no previous ranking algorithm considers whether the connectivity between two nodes is semantically correct or not. In this study, we propose a structural algorithm that can be used for identifying potentially important concepts in OWL Ontology. Concept hierarchy and object properties are used to construct ontology graph. Only the semantically correct links in the graph are used in our ranking algorithm. The ranking algorithm is based on the following idea. First, it infers the OWL Ontology and converts it to directed labeled graph. The edges of a graph are classified into four types (patterns): Upwards (U), Downwards (D), Horizontal (H) and Non-Hierarchy (N). Then the graph is used in an iterative fix-point computation whose results tell us the importance of nodes in that graph. To compute the importance, we treat nine kinds of patterns to be semantically correct and define the importance of a node as a nine-dimensional vector. A component of one node's vector can propagate to its neighbor through edge e if and only if the result pattern is still semantically correct by combining the component's pattern and the type (pattern) of edge e. We call our algorithm the semantic-aware importance flooding ranking algorithm (use SIFRank in short).

DATA MODEL

Ontology graph: As Wu *et al.* (2008), we model a concept as a vertex (node) and a relation as a directed edge linking two concepts in OWL ontology. The RDF-based ontology also can be mapped to ontology graph with the process defined by Wu *et al.* (2008). The constructed graph called ontology graph is defined as below:

Definition 1: Given an ontology O, the ontology graph $G = (V, E, l_V, l_E)$ of O is a directed labeled graph. V is a set of nodes representing all concepts in O. E is a set of directed edges representing all relations in O. l_V and l_E are labeling functions on V and E, respectively. $e_{s,t}$ is the edge connecting ontology nodes s and t, where edge e belongs to edges set E, node s and t belong to nodes set V, we denote by $e \in E$, $s,t \in V$.

In this study, we only focus the edges mapped from hierarchy and object property. The data type property is not considered in our algorithm as it always does not express the relationships among concepts. However, some relations in OWL ontology are implicit and indirect. As the example shown in Fig. 1, the ontology defines two triples: (Project_Admin, manage, Project) and (manage, owl:inverseOf, managed_by). Owl:inverseOf is a build-in OWL property, which is often used to define relations in both directions (McGuinness and Harmelen *et al.*, 2004). Accordingly, there would be an implicit edge to represent the meaning of (Project, managed_by, Project_Admin) in the ontology graph.

Moreover, OWL uses rdfs:subClassOf to statement the case that one class is a subclass of another. But no property is defined as the inverse property of rdfs:subClassOf, while we find the semantic of such relation is very important.

To full use the semantic information in ontology, an inference step is done before establishing the ontology graph. This inference step is mainly used to mine the implicit semantic meanings described above. For example, in order to express the semantically inverse meaning of rdfs:subClassOf, property nga:hasSubClass is defined as triple (nga:hasSubClass, owl:inverseOf, rdfs:subClassOf). Thus, the right part of Fig. 1 shows additional relations we can get by the inference. We can apply more inference rules on the ontology depending on its features.

The semantic of relations: In this study, only the concept hierarchy and object properties are interested. Both the hierarchy and object properties are mapped to edges in ontology graph. Those edges are classified into four kinds of sets (patterns) depending on their meanings: Upwards (U), Downwards (D), Horizontal (H) and Non-Hierarchy (N). An upward direction corresponds to generalization. For instance, in Fig. 1 Developer is a semantically more general concept than Project_Admin. The property rdfs:subClassOf between them is an upward link. Similarly, a downward link corresponds to specialization. Horizontal links mean same to or similar to. For a standard OWL ontology graph generated, upward links set includes property rdfs:subClassOf; downward links set includes property nga:hasSubClass; horizontal links set includes property owl:equivalentClass; and other object properties are belonged to non-hierarchy links set.

Of cause, such classification is depending on the semantic meanings implicated by the links. If a user defined object property is describing One class is same as another, then it belongs to horizontal links set. In our algorithm, one edge should belong to one and only one kind of above four sets.

In the ontology graph, only a few paths can be considered as semantically corrects and these paths obey to a given set of rules. Extending the research of (Hirst and St-Onge, 1997), nine pattern of paths are thought to be semantically correct in ontology graph, they are {U, UD, UH, UHD, D, DH, HD, H, N}. In Fig. 1, the path from Project_Admin to Developer through rdfs:subClassOf is in pattern U. The path Project_Admin-manage-Project-Developed_by-Develope is in pattern N, as all the links in this path are object properties without hierarchy meanings. The path Project-Developed_by-Developer-nga:hasSubClass-Project_Admin is in pattern ND. So, it is not a semantically correct path and the importance of 'Project_Admin' can not flood to Developer through this path.

According to the nine patterns, the importance of a concept node is expressed as a nine-dimensional vector. Each component of the vector shows the corresponding importance of the node for one semantically correct pattern. Then we have the following definition for the importance of a concept node.

Definition 2: The importance of concept node v in ontology graph $G = (V, E, l_V, l_E)$ is vector r(v), where $r(v) = (r_U(v), r_{UD}(v), \ldots r_H(v), r_N(v))$ and $v \in V$. Edge $e_{s,t}$ indicates a connection from node s to t, where $s,t \in V$.

RANKING ALGORITHM

Before we go into details of our ranking algorithm, let us briefly walk through the main process of the algorithm. Consider OWL ontology O, a sequence of steps that allows us to determine the importance of concepts can be expressed as the following script:

- G = InfOnt2Graph(O)
- InitialImp = Initialize(G)
- Importance = SIF(G, initialImp, W_s)
- Result = Order(importance, W_s)

As a first step, we infer ontology O and translate it into graph G. Then the second step assigns initial importance value to each node and classifies the edges into four semantic edge sets (U, D, H and N). Without loss of generality, each dimension of importance can be assigned with the default value 1. In the third step, operator SIF is applied to calculate the importance of the nodes. Operation SIF uses an iterative semantic-aware importance flooding ranking (SIFRank) algorithm which is based on a fix-point computation. The algorithm terminates after a fix-point has been reached, i.e., the importance of all Nodes in graph G stabilize; or the computation reaches some maximal number of iterations. W_s is a 1×9 vector expressing the weights of the nine kinds of semantically correct patterns, the summation of the components of W_s is 1. W_s is always a fixed value which is pre-defined before the running of SIFRank. As a last operation in the script, operator Order gives a list of nodes (concepts) ordered by the importance. The above sections have introduced the first and second steps. In the next context, we will explain the third and fourth steps.

The semantically correct joins: The internal data model that we use for concepts ranking is based on directed labeled graphs. Every edge in the graph is belonged to one of the four semantic sets (U, D, H, N). Each node has an importance vector with nine dimensions (U, UD, UH, UHD, D, DH, HD, H, N). If two nodes v_1 and v_2 are linked by edge $e_{v1,v2}$, an importance component of v_1 could flood into v_2 through $e_{v1,v2}$ only and if only the join of this component's and $e_{v1,v2}$'s patterns is still semantically correct. The join operation is defined as below.

Definition 3: Assume $p_1p_2\ldots p_n$ is a pattern of one importance component, where $p_1,p_2,\ldots,p_n \in \{U,D,H,N\}$ and $1 = n = 9$. p_e is the pattern of edge e. The 'join' result of them is:

- $p_1p_2\ldots p_n$, if $p_n = p_e$
- $p_1p_2\ldots p_n\, p_e$, if $p_n \neq p_e$ and $p_1p_2\ldots p_n\, p_e$ is a semantically correct pattern
- θ, if $p_n \neq p_e$ and $p_1p_2\ldots p_n\, p_e$ is not a semantically correct pattern

Table 1: The semantically correct path construct matrix

	U	D	H	N
U	U	UD	UH	Ø
UD	Ø	UD	Ø	Ø
UH	Ø	UHD	UH	Ø
UHD	Ø	UHD	Ø	Ø
D	Ø	D	DH	Ø
DH	Ø	Ø	DH	Ø
H	Ø	HD	H	Ø
HD	Ø	HD	Ø	Ø
N	Ø	Ø	Ø	N

U: Upwards; D: Downwards; H: Horizontal; N: Non-Hierarchy. Ø means the patterns joining is not semantically correct

Table 1 shows the results of 'join'. It lists the four kinds of edges in the first line and the nine patterns of importance components in the first column. In the Table 1, Ø means the patterns joining is not semantically correct. Otherwise, the filled value is the new semantic pattern. For example, the cell (UD, D) with value UD in the table means that the UD component of importance can flood to its neighbors through an edge with kind of D and the new kind of semantic pattern after join is UD.

In reality, Table 1 gives the rules to transfer one node's component patters to corresponding patterns when does the importance flooding. From it, we can conclude four transfer matrixes corresponding to the four kinds of edges to express those transfer rules.

Definition 4: The transfer matrix TM(type($e_{s,t}$)) of edge $e_{s,t}$ defines the importance flooding rules based on the semantically correct pattern for edge type($e_{s,t}$), where type($e_{s,t}$)$\in${U,D,H,N} returns the semantic pattern of $e_{s,t}$ and $e_{s,t}$ is the edge connecting ontology nodes s and t. Assume r(s) is an importance vector of node s, the result of r(s)•TM(type($e_{s,t}$)) would only includes the semantically correct components after the patterns joining of r (s) and $e_{s,t}$.

From definition 4, there are four transfer matrixes according to the four kinds of edges. In Fig, 2, M1 is a node's importance vector; M2 is the transfer matrix for horizontal (H) edges. The product of M1 and M2 only leave the patterns U, UH, D, DH and H. However, the result does not express the components of importance vector with the same order of M1. To overcome this, a relocation matrix is required to adjust the result of M1•M2.

Definition 5: The relocation matrix RM(type($e_{s,t}$)) of edge $e_{s,t}$ defines the rules to make product vector result of r•TM(type($e_{s,t}$))•RM(type($e_{s,t}$)) has the same semantic patterns order with r, where type($e_{s,t}$) returns the semantic pattern of $e_{s,t}$ and the semantic pattern belongs to the set {U,D,H,N}.

M3 in Fig. 2 is a sample of relocation matrix for horizontal (H) edges.

$$\underbrace{\overset{M_1}{\begin{vmatrix} U \\ UD \\ UH \\ UHD \\ D \\ DH \\ H \\ HD \\ N \end{vmatrix}}^{T} \bullet \overset{M_2}{\begin{vmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{vmatrix}}}_{|U\ \ 0\ \ UH\ \ 0\ \ D\ \ DH\ \ H\ \ 0\ \ 0|} \bullet \overset{M_2}{\begin{vmatrix} 0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0 \end{vmatrix}}$$

Fig. 2: Sample with a horizontal edge

The iterative algorithm: The previous section provided the definition of transfer matrix and relocation matrix. The iterative ranking algorithm will be introduced now. Given an ontology graph G = (V, E, l_V, l_E), after k (k = 0, 1, 2, ...) iterations, the importance of a concept s $\in$ V is written as $r^{k+1}(s)$. It is recursively evaluated in Eq. 1 and 2:

$$r'^{k+1}(t) = \sum_{\text{foreach } e_{s,t}} (r^k(s) \cdot TM(type(e_{s,t})) \cdot RM(type(e_{s,t}))) \qquad (1)$$

$$r^{k+1}(t) = normalize(r'^{k+1}(t)) \qquad (2)$$

Equation 1 computes the importance of concept t at the (k+1)th iteration. The raw importance is reinforced from its adjacent concepts, which have the edges ending with concept t. Equation 2 means that, for each iteration, the value calculated out by our flooding algorithm (r'^{k+1}) should be normalized to get the new importance. For instance, divided by the maximal value of $r'^{k+1}(i)$•W_s for that iteration. Thus, over a number of iterations, the initial importance of any node propagates through the graph based on the semantically correct paths. The importance vector of the ontology graph after k-1 iterations is R^k=($r^k(s_1)$•W_s, $r^k(s_2)$•W_s, ..., $r^k(s_n)$•W_s).

Let $\Delta(R^{k+1}, R^k) = \|R^{k+1}-R^k\|$ to be the difference between two successive iterations. The Eq. 1 and 2 computation are performed iteratively until the Euclidean length of the residual vector$\Delta(R^{k+1}, R^k)$ becomes less than ε for some k>0. If the computation does not converge, we terminate it after some maximal number of iterations.

Let O be the inferred ontology graph, W_s be the important weight vector and I be the initial importance vector of nodes ($r^k(s_1)$•W_s, r (s_2^k)•W_s, ..., r (s_n^k)•W_s). In terms of Eq. 1, 2 and above description, we present the semantic-aware importance flooding algorithm as follows:

1: SIFRank(O, W_s, I)
2: $R^0 \leftarrow I \bullet W_s$, $k \leftarrow 0$

3: Repeat
4: Foreach node t in O
5: Foreach edge $e_{v,t}$ in O
6: $r'^{k+1}(t) \leftarrow r'^{k+1}(t) + r^k(t) \bullet TM(type(e_{v,t})) \bullet RM(type(e_{v,t}))$
7: Normalize $\leftarrow$ max $(r'^{k+1} \bullet W_s)$
8: Foreach node t in O
9: $r^{k+1}(t) \leftarrow r'^{k+1}(t)$/normalize
10: $R^{k+1} \leftarrow (r^{k+1}(S_1), r^{k+1}(S_2), \ldots, r^{k+1}(S_n))$
11: $\Delta \leftarrow \|R^{k+1} - R^k\|$
12: $k \leftarrow k+1$
13: Until $\Delta < \varepsilon$ or k>MAX_ITERATION_NUM
14: Return R^k

The algorithm computes the raw importance vector for each node with Eq. 1 (step 4-6) and select the number (step 7) to normalize all the importance vectors (step 8-9). After that, the importance vector R^{k+1} for the whole graph is constructed (step 10) and used to calculate the distance Δ between two successive iteration (step 11). Threshold $0<\varepsilon<1$ controls the termination of the iteration with the pre-defined maximum iteration number (step 13). Finally, the algorithm returns R^k as the concept importance vector for OWL Ontology graph O.

EVALUATION AND DISCUSSION

Evaluation: In our implementation, we parsed and inferred ontology by Jena, a semantic web develop tool. The inferred result was mapped to a directed graph and assigned one of the semantic meanings, {U, D, H, N}, to every edges. The nodes in the directed graph are assigned to initial importance values. Then SIFRank algorithm calculating is based on the directed graph and weights (W_s) of nine kind of semantic patterns. Figure 3 shows the steps to rank the ontology concepts.

To evaluate the quality of the algorithm for ontology concepts ranking tasks, we conducted a user study with help of twelve volunteers from one university. We tried to collect representative ontologies and their accurate answers (a list of ranked concepts) and then compared results among accurate answers, user provided answers, PageRank algorithm results and our algorithm results. Users should give out their answers in restrict time. In our experience, gave users 15 min if the total number of concepts and relations less than 100; gave 30 min if the total number is 100~200; gave 45 min if the total number is 200~500 and gave 1 h if the total number larger than 500. The final concepts order from user study is calculated out by averaging the twelve orders from users. Noteworthy is that almost no two users could give the same concept importance order for an ontology. Therefore, we could hardly expect any automatic procedure to produce excellent results.

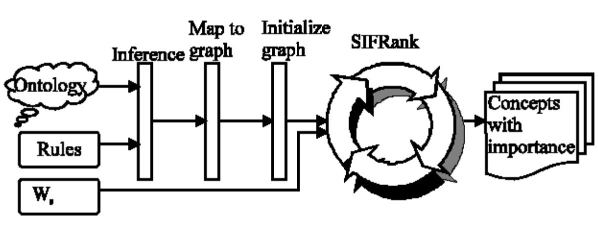

Fig. 3: The main work flow of SIFRank algorithm

Similar to Wu *et al.* (2008), we employed a variant first 20 precision metric (Leighton and Srivastava, 1999) to measure the quality of concepts ranking results. The improved first 20 precision,

$$P_{20} = (n_{1|-3} \times 20 + n_{4|-10} \times 17 + n_{11|-20} \times 10)/279$$

assigns different weights for the first 3, the next 7 and the last 10 results to increase the value for ranking effectiveness (Wu *et al.*, 2008; Leighton and Srivastava, 1999).

In our experience, we used $\varepsilon = 10^{-4}$. We thought the more pure a semantic correct pattern is, the higher weight it should have in W_s. So the pattern U would have larger weight than UH and UH has larger weight than UHD. One exception is the pattern N, it may be constructed by various object properties, so we gave the lowest weight to N. Then we use {0.14, 0.1, 0.1, 0.06, 0.14, 0.1, 0.1, 0.14, 0.12} for W_s {U, UD, UH, UHD, D, DH, HD, H, N}.

We have tested the SIFRank algorithm against several different datasets that belongs to different domains, which are listed as below.

- Software Project Ontology (http://keg.cs.tsinghua.educn/project/software.owl): It describes concepts and relations in an open software project domain, especially the relationships between developers and projects. It contains 14 concepts and 84 property relations
- Wine Ontology (http://www.w3.org/TR/2003/PR-owl-guide-20031209/wine): It contains information about different types of wines, characteristics, regions, among others. It contains 74 concepts and 13 property relations
- Copyright Ontology (http://rhizomik.net/ontologies/2006/01/copyrightonto.owl): It represents information about copyright. It contains 98 concepts and almost 46 property relations
- Clinical Ontology (http://acgt.googlecode.com/svn-history/r213/trunk/document.owl): It contains information about prevented clinical research and contains more than 200 concepts and 200 property relations

Table 2: The importance of concepts b software project ontology adapted from Wu *et al.* (2008)

No.	Ref. Answer	PageRank	CARRank	User Study	SIFRank
1	Project	***Message***	***Project***	***Project***	***Category***
2	Member	has_usage_statistics	***Usage_statistics***	***Category***	***Project***
3	Developer	statistics_bugs	Statistic_record	***Message***	***Usage_statistics***
4	Category	Statistic_record_support	***Developer***	Discussion	***Developer***
5	Public_forums	***Member***	***Category***	Help	Release_package
6	LastestNew	***Project***	Release_package	Person	Statistic_record
7	Message	***Developer***	***Member***	***Member***	***Version***
8	Version	***Category***	***Message***	***Developer***	***Message***
9	homepage	supper_category	Help	Project_admin	***Member***
10	Usage_statistics	Page_views	***Public_forums***	***Public_forums***	***Public_forums***
Total		5	7	7	8

Italic bold font are relevant ranking results

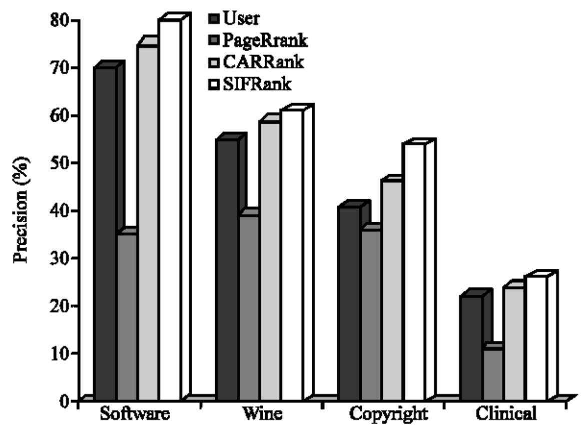

Fig. 4: The accuracy comparison of ranking algorithm

Table 2 adapted from Wu *et al.* (2008) presents the comparisons on concepts ranking for software project ontology, which uses a benchmark. Items listed in italic bold font are relevant ranking results. There are 5 relevant items in the first 10 ranking results for PageRank, 7 for CARRank, 8 for SIFRank and 6 for the user study. The average accuracy in percent achieved by the 6 users, PageRank and our algorithm are summarized in Fig. 4. Obviously, SIFRank have better ranking qualities than the user study.

DISCUSSION

To the best of our knowledge, it is the first time that the semantic correctness measure is used in ontology concepts ranking. Our approach focuses on OWL ontology and relies on two hypotheses. Firstly, SIFRank uses only concept hierarchy and object properties in OWL ontology, only a few patterns can be considered as semantically corrects and these patterns obey to a given set of rules. Secondly, the edge in directed graph has type in set of {U, D, H, N}, which depends on its semantic meaning. But all edges are considered to have the same weight/cost. W_s, the weights of the nine semantically correct patterns, are empirical data, which depends on the features of the input ontology. In future, we would do more experiences to evaluate how to get the best W_s values.

Please note nearly no two users could give the same concept importance order for a large ontology and there is no standard to order the concepts by their importance. The importance of concepts may be different under special scenarios, so there is no benchmark to precisely measure the ranking results. Therefore, we could hardly expect any automatic procedure to produce excellent results. In our research, we found when the ontology became larger, it required more and more time for user understanding the ontology.

Table 2 and Fig. 4 show the comparisons on concepts ranking. Obviously, SIFRank and CARRank have better ranking qualities than the user study. SIFRank is better than CARRank in our experience, so SIFRank algorithm could give a suggested order of concepts to help user understand a new and large scale ontology. It also shows that PageRank approach gives less relevant results than the user study and it is not a proper method in concepts ranking.

CONCLUSIONS

In this study we present a concept ranking algorithm for OWL ontologies based on the ontological structure and semantic meanings of relations. We define semantically correct join patterns to be used in the semantic-aware importance flooding algorithm. This algorithm considers both ontology structure and semantic meanings. The experimental results show the feasibility of the algorithm from the ranking qualities.

Present method classifies the edges of ontology graph into four kinds of semantic sets and treats them having the same weight. In future work, we will do experiences on more kinds of ontologies. We will investigate how to calculate the weights of edges by the semantic meanings and also ranking the properties in ontology. We would like to improve the ranking by incorporating user feedback into the ranking algorithm to increase precision.

REFERENCES

Alani, H., C. Brewster and N. Shadbolt, 2006. Ranking ontologies with AKTiveRank. Proceeding of the 5th International Semantic Web Conference (ISWC), Aug. 29-Jan. 6, Georgia, USA., pp: 1-15.

Brin, S. and L. Page, 1998. The anatomy of a large-scale hypertextual web search engine. Comput. Networks ISDN Syst., 30: 107-117.

Ding, L., T. Finin, A. Joshi, R. Pan and R.S. Cost *et al*., 2004. Swoogle: A search and metadata engine for the semantic web. Proceedings of the 13th ACM International Conference on Information and Knowledge Management, Nov. 8-13, Washington DC. USA., pp: 652-659.

Graves, A., S. Adali and J. Hendler, 2008. A method to rank nodes in an RDF graph. The 7th International Semantic Web Conference, Oct. 26-30, Karlsruhe, Germany, pp: 82-87.

Hirst, G. and D. St-Onge, 1997. Lexical Chains as Representation of Context for the Detection and Correction Malapropisms. In: WordNet: An Electronic Lexical Database, Fellbaum, C. (Ed.). The MIT Press, Cambridge, MA, pp: 305-332.

Hogan, A., A. Harth and S. Decker, 2006. ReConRank: A scalable ranking method for semantic web data with context. Proceedings of the 2nd Workshop on Scalable Semantic Web Knowledge Base Systems, Nov. 5, Athens, Georgia, pp: 258-271.

Jiang, X. and A.H. Tan, 2006. OntoSearch: A full-text search engine for the semantic web. The 21 National Conference on Artificial Intelligence, July 16-20, Boston, pp: 1325-1330.

Leighton, H.V. and J. Srivastava, 1999. First 20 precision among world wide web search services (search engines). J. Am. Soc. Inform. Sci., 50: 870-881.

McGuinness, D.L. and F. van Harmelen, 2004. OWL web ontology language overview. W3C Recommendation. http://www.w3.org/TR/2004/REC-owl-features-20040210.

Patel, C., K. Supekar, Y. Lee and E. Park, 2003. Ontokhoj: A semantic web portal for ontology searching, ranking and classification. Proceedings of the 5th ACM International Workshop on Web Information and Data Management, Nov. 7-8, New Orleans, Louisiana, USA., pp: 58-61.

Rajapaksha, S.K. and N. Kodagoda, 2008. Internal structure and semantic web link structure based ontology ranking. Proceedings of the 4th International Conference on Information and Automation for Sustainability, Dec. 11-14, Colombo, pp: 86-90.

Sankar, V.R., A. Damodaram and P.R. Krishna, 2010. Ranking ontologies based on OWL language constructs. Inform. Technol. J., 9: 553-560.

Uren, V., Y.G. Lei and E. Motta, 2008. SemSearch: Refining semantic search. Proceedings of the 5th Annual European Semantic Web Conference, June 1-5, Tenerife, Spain, pp: 874-878.

Wu, G., J.Z. Li and L. Feng, 2008. Identifying potentially important concepts and relations in an ontology. Proceedings of the 7th International Semantic Web Conference, Oct. 26-30, Karlsruhe, Germany, pp: 33-49.

Zhang, X., H.D. Li and Y.Z. Qu, 2006. Finding important vocabulary within Ontology. The 1st Asian Semantic Web Conference, Sept. 3-7, Beijing, China, pp: 106-112.