Identifying Potentially Important Concepts and Relations in an Ontology *

Gang Wu, Juanzi Li, Ling Feng, and Kehong Wang

Department of Computer Science, Tsinghua University, Beijing 100084, P.R.China

Abstract. More and more ontologies have been published and used widely on the web. In order to make good use of an ontology, especially a new and complex ontology, we need methods to help understand it first. Identifying potentially important concepts and relations in an ontology is an intuitive but challenging method. In this paper, we first define four features for potentially important concepts and relation from the ontological structural point of view. Then a simple yet effective Concept-And-Relation-Ranking (CARRank) algorithm is proposed to simultaneously rank the importance of concepts and relations. Different from the traditional ranking methods, the importance of concepts and the weights of relations reinforce one another in CARRank in an iterative manner. Such an iterative process is proved to be convergent both in principle and by experiments. Our experimental results show that CARRank has a similar convergent speed as the PageRank-like algorithms, but a more reasonable ranking result.

1 Introduction

Ontology provides Artificial Intelligence and Web communities the remarkable capability of specifying shared conceptualization explicitly and formally. A diversity of ontologies have been widely used as the bases of semantic representation in many applications such as knowledge bases, multi-agents and the Semantic Web. As the amount, scale, and complexity of ontologies are increasing rapidly, it requires more efforts for ontologists and domain experts to understand them. Hence, *Ontology Understanding*, the process of getting familiar with an ontology [4], has to seek helps from computer intelligence.

The state-of-the-art ontology engineering projects, like IsaViz, Ontoviz, and Jambalaya, use information visualization techniques to represent ontologies. They have the ability to help humans understand and navigate in complex information spaces [9]. However, for a complex ontology, graphically presenting all concepts and relations indistinctively makes above tools generate unreadable visualization results. Users who are unfamiliar with the ontology will probably get lost in such a maze.

* This work is supported by the National Natural Science Foundation of China under Grant No.90604025 and the Major State Basic Research Development Program of China (973 Program) under Grant No.2003CB317007 and No.2007CB310803.

To resolve the problem, some researchers have proposed approaches by drawing users' attention to those potentially *important* (or alternatively *interesting*) concepts within one ontology. They calculate the importance of concepts either by tracking the user's browsing activities [7], or according to the concept hierarchy [20]. These solutions are straightforward. While more detailed information about ontology structure, like the correlation between concepts and relations, is not explored. In some other studies, traditional link analysis ranking algorithms on Web pages and objects are employed to rank the importance of concepts [3], and even the importance of relations [8, 17]. These solutions need the help of additional statistic information or time-consuming machine learning schemes.

In this paper, we propose a simple yet effective algorithm, named Concept And Relation Ranking (CARRank), for identifying potentially important concepts and relations in an ontology. By efficiently ranking the importance of concepts and relations simultaneously, CARRank can find out which concepts and relations might be the ones the ontology creator would like to suggest to users for further consideration. In this way, CARRank can promote the usability for ontology understanding. Users can even outline an interested *sub-scope* of an ontology, of which important parts are taken out. Although CARRank is rather an automatic ranking algorithm than a specific visualization approach, it can be easily integrated into the existing ontology visualization tools to provide a novel perspective. Main contributions of this paper include:

1) To make good use of ontology structural information, we give a graph representation of ontology which makes it easy for applying link analysis ranking algorithms while preserves the semantics expressed by RDF-based ontology languages.

2) To determine the potentially important concepts and relations in an ontology, we introduce an importance ranking model. The model tries to imitate the creation process of an ontology from the ontological structural point of view by defining four representative features.

3) To calculate the importance of concepts and relations, we propose an efficient algorithm according to the model, named CARRank. The difference between CARRank and existing PageRank-like algorithms is two-fold. Firstly, with this algorithm, the importance of vertices (i.e. concepts) and the weights of edges (i.e. relations) reinforce one another in an iterative process. Such a dynamic computation on edges weights as well as vertices importance has never been studied previously. Secondly, the directions of *walk* for the algorithms are opposed, which makes CARRank more suitable for supporting ontology understanding. CARRank is proved to be convergent, and thus is universal for simultaneously ranking vertices importance and edges weights in arbitrary directed labeled graph.

4) Experiments are conducted to demonstrate the effectiveness and efficiency of the approach to support understanding of ontologies.

The remainder of the paper is organized as follows. We review the closely related work in Section 2, and present our CARRank model in Section 3. We then bring forward the CARRank algorithm in Section 4. Experimental results are shown in Section 5. The final section is about the conclusion and discussion.

2 Related Work

Cognitive Support for Ontology Understanding. The DIaMOND project [7] and the holistic "imaging" ontology [20] are two most related studies. DIaMOND [7] is a plug-in for Protégé[1] to help users find concepts of interest within an ontology. By tracking user's navigation activities on an ontology, it continuously calculates the degree of interest for each concept. The navigation overhead can thus be reduced by drawing user's attention to the highlighted concepts of high interest degrees. The degree calculation of this method is user-specific. In [20], authors exploited degrees of interest of concepts as a filter for labeling important concepts in a large scale ontology. Its degree calculation is holistically based on concept hierarchy without considering non-subsumption relations between concepts. Our work differs from these approaches. First, we think that the importance measurement of a concept should take into account the contributions from all the other concepts in the ontology through relations including both subsumption and non-subsumption ones. Second, relations between concepts are also helpful for ontology understanding.

Ontology Ranking in the Semantic Web. OntoSelect [6], OntoKhoj [18], and AKTiveRank [1] are three approaches that were developed to select (or rank) one or more ontologies that satisfy certain criteria [19], with an ontology document as the ranking granularity. The first two approaches relied on the *popularity*, which assumed that ontologies referenced by many ontologies are more popular, while the third one considered several structural evaluation metrics, including Density (DEM), Betweenness (BEM), Semantic Similarity (SSM), and Class Match measure (CMM). Although AKTiveRank does not intend to rank the importance of concepts or relations in an ontology, the above complex networks analysis metrics it employs are useful for reference in this work. According to the pre-existing statistic information on instances, Swoogle [8] could enable both document level and term level ranking, including the class-property relationship ranking.

Compared with this line of research, our study aims to finding out potentially important information in a given ontology, so the granularity of output is concept and relation, rather than a whole ontology. Besides, the method can evaluate the importance of general relations of concepts, as well as concepts themselves. Furthermore, no prior knowledge or user interaction is required, which may be more applicable in dealing with new ontologies. Table 1 lists some of the differences.

Table 1: Related Work in the Semantic Web

	Concept Rank	Relation Rank	Ranking Methods
CARRank	✓	✓	CARRank
DIaMOND	✓	-	Tracking users' navigation
[20]	✓	-	Concept hierarchy
OntoSelect	-	-	PageRank-like
OntoKhoj	-	-	PageRank-like
AKTiveRank	-	-	CMM+DEM+SSM+BEM
Swoogle	✓	✓	PageRank-like

[1] http://protege.stanford.edu/

Ranking Algorithms. In ranking Web pages, hyperlink is the only relation to be considered. PageRank [5] pointed out that a good *authority* page is the one pointed to by many good authorities. The evaluation is performed in a *random surfer* manner over all pages on the Web graph. Unlike PageRank, HITS [13] exploited a mutual reinforcing relationship between *hub* pages and authority pages within a subgraph of the Web. By extension of PageRank and HITS, Reverse PageRank [10] was investigated as a reasonable approach to *browse* the Web, which reverses the direction of all hyperlinks before applying PageRank. In this study, we browse an ontology in a similar manner to Reverse PageRank.

Apart from the hyperlink relation, there exist more edge types in an ontology, such as property-of, subclass-superclass, etc. The edge type is an important factor in determining the importance of vertices. This was addressed recently in a series of object-level link analysis ranking algorithms. In the field of database, ObjectRank [3] applied link analysis methods to rank the importance of database objects and tuples. Different weights are set according to link types either manually or by statistic information. PopRank [17] is a machine learning approach to automatically assign the weights and rank the importance of Web objects. These weight assignment approaches are not applicable for ontology understanding where absence of priori knowledge is fairly common. We attempt to resolve it by evaluating the weights simultaneously in the ranking process according to only the mutually reinforcing relationship between concepts and relations.

3 CARRank Model

3.1 Ontology Graph

Before any link analysis could be performed, an ontology should be represented as a graph. As an ontology defines the concepts and the relations between them in certain domain [16, 11], it is suggested to model a concept as a vertex and a relation as a directed edge linking two concepts. We call such constructed graph the *ontology graph*.

Definition 1. *Given an ontology $\mathcal{O}$, the* ***ontology graph*** $\mathcal{G} = (\mathcal{V}, \mathcal{E}, l_{\mathcal{V}}, l_{\mathcal{E}})$ *of $\mathcal{O}$ is a directed labeled graph. $\mathcal{V}$ is a set of nodes representing all concepts in $\mathcal{O}$. $\mathcal{E}$ is a set of directed edges representing all relations in $\mathcal{O}$. $l_{\mathcal{V}}$ and $l_{\mathcal{E}}$ are labeling functions on $\mathcal{V}$ and $\mathcal{E}$ respectively.*

Definition 1 is a representation of an ontology at the syntactic level. Its semantic capabilities will be presented in section 3.4.

The ontology graph illustrated in Figure 1 is our running example. It describes concepts and relations in an open software project domain, especially the relationships between developers and projects.

3.2 Mapping RDF-based Ontology to Ontology Graph

In practice, the most important ontology languages in the Semantic Web are RDF Schema (RDFS) and OWL. In these languages, an ontology is expressed

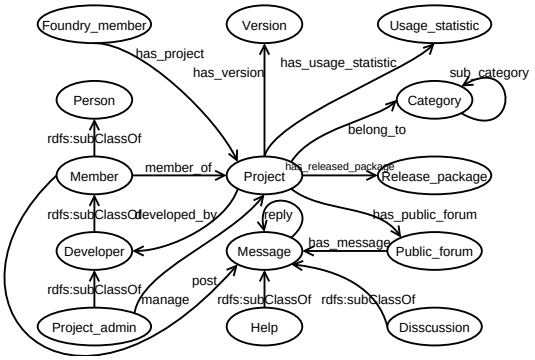

Fig. 1: The running example

as a set of triples. A triple $(s, p, o) \in (U \cup B) \times U \times (U \cup B \cup L)$ is called an RDF triple where U, B, and L are infinite sets of URI references, blank nodes, and literals respectively. Here, s is called the *subject*, p the *predicate*, and o the *object* of the triple. A set of such RDF triples is defined as an *RDF graph* [21], and represented as a directed labeled graph as shown in Definition 2. We will use the RDF graph to refer to both a set of RDF triples and its directed labeled graph representation throughout the rest of this paper.

Definition 2. *Let T be a set of RDF triples. The* ***directed labeled graph representation*** *of T is $G = (V, E, l_V, l_E)$, where*

$$V = \{v_x | x \in \mathit{subject}(T) \cup \mathit{object}(T)\}$$
$$E = \{e_{s,p,o} | (s, p, o) \in T\}$$
$$l_V(v_x) = \begin{cases} (x, d_x) & \text{if } x \text{ is literal } (d_x \text{ is datatype identifier}) \\ x & \text{else} \end{cases}$$
$$\mathit{from}(e_{s,p,o}) = v_s, \mathit{to}(e_{s,p,o}) = v_o, \text{ and } l_E(e_{s,p,o}) = p$$

V is the set of vertices in G. E is the set of directed edges. l_V and l_E are labeling functions on V and E. subject(T) and object(T) are used to achieve all the subjects and the objects in T. Function from() and to() return the starting and ending vertex of an edge.

However, for the same ontology, an RDF graph and an ontology graph are unequal. Suppose an ontology consists of a relation "manage" linking from "Project_Admin" to "Project". The ontology graph is shown in Figure 2. To express the same semantics, an RDF graph needs two triples (manage, rdfs:domain, Project_Admin) and (manage, rdfs:range, Project) as shown in Figure 3.

The difference lies in that, for an ontology, a relation does not exist as a directed edge but a vertex in an RDF graph. A relation is associated with a concept by the semantics of rdfs:domain or rdfs:range (the concept is named *domain* or *range* accordingly). Such indirect relationships will hinder the importance propagation during the ranking, because there is no path between the domain

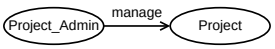

Fig. 2: An ontology graph representation

Fig. 3: An RDF graph representation

and the range. Hence, we propose a map function ω to map an RDF graph to an ontology graph in Definition 3.

Definition 3. *Let $G = (V, E, l_V, l_E)$ be the RDF graph of an ontology $\mathcal{O}$. We define a map $\omega : G \to \mathcal{G}$ as follows: $\omega(G) = (\mathcal{V}, \mathcal{E}, l_\mathcal{V}, l_\mathcal{E})$ where,*

$$
\begin{aligned}
\mathcal{V} &= V, \quad l_\mathcal{V} = l_V, \\
\mathcal{E} &= \{e_{s,p,o} | e_{s,p,o} \in E \wedge l_E(e_{s,p,o}) \neq \textit{rdfs:domain} \wedge l_E(e_{s,p,o}) \neq \textit{rdfs:range}\} \\
&\quad \cup E_{DR} \cup E_D \cup E_R, \\
E_{DR} &= \{e_{s,p,o} | \exists e_{p,\textit{rdfs:domain},s} \in E \wedge \exists e_{p,\textit{rdfs:range},o} \in E\}, \\
E_D &= \{e_{s,p,\textit{keg:Sink}} | \exists e_{p,\textit{rdfs:domain},s} \in E \wedge \nexists e_{p,\textit{rdfs:range},o} \in E\}, \\
E_R &= \{e_{\textit{keg:Source},p,o} | \exists e_{p,\textit{rdfs:range},o} \in E \wedge \nexists e_{p,\textit{rdfs:domain},s} \in E\}, \\
\forall e_{s,p,o} \in \mathcal{E},\ & \textit{from}(e_{s,p,o}) = v_s,\ \textit{to}(e_{s,p,o}) = v_o, \textit{and } l_\mathcal{E}(e_{s,p,o}) = p
\end{aligned}
$$

Here, keg:Source and keg:Sink are defined to be the virtual domain and range of those relations having no domain or range defined explicitly.

Each edge in the output ontology graph is an RDF triple. Therefore the same relation can be distinguished between different domain concepts and range concepts. The map removes those edges taking rdfs:domain or rdfs:range as their labels, while adds new labeled edges to directly link the domains to the ranges according to the rules in Definition 3. In this way, $\omega(G)$ presents an ontology graph that preserves the semantics of G and makes it easy for ranking. Thus, the RDF graph in Figure 3 can be mapped to the ontology graph in Figure 2. In fact, our running example shown in Figure 1 is mapped from a real ontology[2].

3.3 Model Description

The creation of an ontology is a composition process where the creator operates with a set of concepts and relations. Hence, the ontology could be considered as the image of the creator's own understanding of the knowledge, just like a literary work to its author. This phenomenon of human consciousness can be best explained with William James' famous *stream of consciousness* theory [12]. He observed that human consciousness has a composite structure including *substantive parts* (thought or idea) and *transitive parts* (fringe or penumbra), and keeps moving from thought to thought. Transitive parts play an important role in controlling the orderly advance of consciousness from one thought to another. By analogizing concepts and relations to substantive parts and transitive parts, the creation of an ontology could be described as drifting on the stream of the creator's consciousness of the domain knowledge from one concept to another

[2] http://keg.cs.tsinghua.edu.cn/project/software.owl

via a particular relation. The initially created concept has a certain possibility of being one of the creator's emphasis (suggestions to users). For the concepts to be suggested, the creator would always like to create more relations to describe its relationships with other concepts. Consequently, ontology users will implicitly follow the creator's stream of consciousness for understanding the ontology.

We characterize four features for potentially important concepts and relations which drive the drift on the stream of consciousness. It turns out to be our model for Concepts And Relations Ranking (the CARRank model):

1. A concept is more important if there are more relations starting from the concept.

2. A concept is more important if there is a relation starting from the concept to a more important concept.

3. A concept is more important if it has a higher relation weight to any other concept.

4. A relation weight is higher if it starts from a more important concept.

There are three meanings here. First, it explains what is *important* (or alternatively *interesting*). In this paper, term *importance* is used as a metric for measuring the extent that the ontology creator suggests a concept or relation to users. Second, a concept is regarded as a source that owns a set of relations related to other concepts. We refer to this character as the *hub* like that in HITS [13]. Finally, concepts and relations exhibit a mutually reinforcing relationship.

In our running example, concepts "Project", "Project_admin" and "Developer" are more attractive because they either have abundant relations to other concepts (e.g. "Project"), or locate deeply in the subsumption hierarchy (e.g. "Project_admin"), or have a relation to other attractive concept (e.g. "Developer"). Accordingly, relation "manage" between "Project" and "Project_admin" becomes more meaningful. These observations coincide with the creator's comment that declares to emphasize the relationship between developers and projects. Our inquiry to the creator about the design process is answered as follows: First defined the concept "Project" with some decorative literals such as "Version" and "Usage_statistic". Next, provided another concept "Developer" to complement the description of "Project" through a relation "developed_by" from "Project" to "Developer". Then, a hierarchy was built about "Developer" from "Person" to "Project_admin". The process continued until all information was included.

3.4 Semantic Abilities

By using ω mapping, any RDF-based ontology, like RDF Schema, DAML+OIL, and OWL (including three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full), can be ranked with the CARRank model. In the section of experiments, we will further analyze the ranking results of CARRank for the same ontology in three languages with different expressive powers.

Furthermore, CARRank even has the ability to support axioms expressed as rules, e.g. SWRL [22] rules, because there exists RDF-compatible model-theoretic semantics [15] of SWRL by which we can interpret SWRL rules in the

framework of RDF graphs. In a broad sense, any inference scheme for ontology is supported by CARRank, if it is resolvable on the level of RDF graphs.

Moreover, since a relation is represented as a vertex in an RDF graph, and then kept in the ontology graph after ω mapping, the hierarchies and properties of relations will also impact the global importance of these relations. That means if there is a deeper hierarchy or more properties for a specific relation, the importance of that relation is higher. Here, whereas we only concern about the comparison locally among relations starting from the same concepts rather than globally among all relations, because the importance may be quite different when associated with different concepts.

Finally, since ontology understanding is affected by many factors, here the importance only means some *potential* to be important in our context.

4 CARRank Algorithm

Definition 4. *Suppose an ontology graph $\mathcal{G}$ has $|\mathcal{V}| = n \geq 1$ concepts $v_1, ..., v_n \in \mathcal{V}$. The **adjacency matrix representation** of $\mathcal{G}$, $\mathbf{A} = (a_{i,j})$, is a $n \times n$ matrix where $1 \leq i, j, k \leq n$ and*

$$a_{i,j} = \begin{cases} 1 & \text{if } \exists e_{i,k,j} \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

*Let $\mathrm{w}(v_i, v_j)$ be a relation weight function, and $w_{i,j} = \mathrm{w}(v_i, v_j)$ be the weight of all relations from v_i to v_j. The **relation weight matrix representation** of $\mathcal{G}$, $\mathbf{W} = (w_{i,j})$, is a $n \times n$ matrix where $1 \leq i, j, k \leq n$, and*

$$\begin{cases} 0 < w_{i,j} \leq 1 & \text{if } \exists e_{i,k,j} \in \mathcal{E}, \\ w_{i,j} = 0 & \text{otherwise.} \end{cases} \tag{2}$$

Definition 5. *For any concept $v_i \in \mathcal{V}$, **the forward concepts** of v_i are defined as $F_{v_i} = \{v_j | v_j \in \mathcal{V} \wedge \exists e_{i,k,j} \in \mathcal{E}\}$, and **the backward concepts** of v_i are defined as $B_{v_i} = \{v_j | v_j \in \mathcal{V} \wedge \exists e_{j,k,i} \in \mathcal{E}\}$.*

Definition 6. *Suppose an ontology graph $\mathcal{G}$ has $|\mathcal{V}| = n \geq 1$ concepts $v_1, ..., v_n$. Let $\mathrm{r}(v_i)$ be an importance function on $\mathcal{V}$, and $r_i = \mathrm{r}(v_i)$ be the importance value of v_i where $0 \leq r_i \leq 1$, $\sum r_i = 1$, and $\mathbf{W} = (w_{i,j})$ be the relation weight matrix. We call $\mathbf{R} = (r_1, ..., r_n)$ the ontology graph $\mathcal{G}$'s **concept importance vector** , and $\mathbf{L}_i = (r_1 w_{i,1}, \cdots, r_n w_{i,n})$ the concept v_i's **relation importance vector** .*

It is possible that there exists more than one relation from concept v_i to concept v_j. Therefore, $r_j w_{i,j}$ is the total importance value of all the relations from concept v_i to concept v_j. Suppose there are $m > 0$ such relations, $e_{i,k_1,j}, ..., e_{i,k_m,j}$. We define the importance of individual relation $e_{i,k_l,j}$ to be $\frac{r_j w_{i,j}}{m}$ for any $1 \leq l \leq m$.

Since a concept, like a hub according to the first two features of our model, sinks the importance of other concepts, the computation for the importance is

totally the reverse of the process in PageRank. In fact, CARRank traces the stream of consciousness reversely similar to the idea of Reverse PageRank [10]. The difference is that it updates the weight of relations during the iteration according to the last two features of the model. Given an ontology graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, l_{\mathcal{V}}, l_{\mathcal{E}})$, after k ($k = 0, 1, 2, ...$) iterations, the importance of a concept $s \in \mathcal{V}$ and the weight of relation(s) from s to another concept $t \in \mathcal{V}$ are written as $\mathrm{r}_{k+1}(s)$ and $\mathrm{w}_{k+1}(s, t)$ respectively. They are recursively evaluated in Equations 3 and 4.

$$\mathrm{w}_{k+1}(s,t) = \frac{\mathrm{r}_k(s)}{\sum_{t_i \in B_t} \mathrm{r}_k(t_i)} \tag{3}$$

$$\mathrm{r}_{k+1}(s) = \frac{1-\alpha}{|\mathcal{V}|} + \alpha \sum_{t_i \in F_s} \mathrm{r}_k(t_i)\mathrm{w}_{k+1}(s,t_i) \tag{4}$$

Like PageRank-like algorithms, we use a damping factor $0 < \alpha < 1$ as the probability at which CARRank will get bored of reversely tracing the stream of consciousness and begin looking for another concept on the ontology graph.

Equations 3 and 4 reflect the features of our potentially important concepts and relations model. Equation 3 formalizes the last feature, which computes the weight of relation(s) starting from concept s to concept t at the $(k+1)$th iteration. The weight is in proportional to the importance of s and in the inverse ratio of the sum of all importance of t's backward concepts at the kth iteration. Therefore, an important concept will increase the weight of those relations starting from itself. Equation 4 formalizes the first three features, which compute the importance of concept s at the $(k+1)$th iteration. The importance consists of two parts. One is contributed by all the importance of s's forward concepts and the weight of relations from s to the forward concepts with probability α. The other is contributed by some independent jump probabilities (here is $\frac{1}{|\mathcal{V}|}$) when CARRank leaves the current stream of consciousness with probability $1 - \alpha$.

For any initial distribution of concept importance vector $\mathbf{R_0} = (r_1^0, r_2^0, ..., r_n^0)$, we have proved[3] that the iterative sequence $\{\mathbf{R}_k \mid k = 0, 1, 2, ...\}$ will converge to $\mathbf{R}^*$ which is the solution of this non-linear equations, i.e. the final result of concept importance vector. Correspondingly, $\mathbf{W}^*$ is the final result of the relation weight matrix. In numerical analysis, it is reasonable to take $\mathbf{R}_{k+1}$ as the approximation of $\mathbf{R}^*$ and stop the iterative process, if the difference between two successive iterations $\| \mathbf{R}_{k+1} - \mathbf{R}_k \|$ is small enough. Thus ranking the importance of the concepts is performed by sorting the entries in $\mathbf{R}^*$. With a slight effort, ranking the importance of the relations related to certain concept is performed by sorting the entries in the relation importance vector which is computed with $\mathbf{W}^*$ and $\mathbf{R}^*$.

[3] For the details of the proof, see our technical report [24]. The proof indicates that CARRank is a flexible algorithm for evaluating the importance of vertices and edges simultaneously in any kind of directed graph.

Let $\mathbf{A}$ be the adjacency matrix representation of an ontology graph[4], and $\mathbf{S}$ be the initial concept importance vector. In terms of Equation 3, 4 and the above descriptions, we present the CARRank algorithm as follows.

CARRank($\mathbf{A}, \mathbf{S}$)

1 $\mathbf{R}_0 \leftarrow \mathbf{S}, \mathbf{W}_0 \leftarrow 0, k \leftarrow 0$
2 **repeat**
3 $\quad \Sigma \leftarrow \mathbf{A}\mathbf{R}_k$
4 $\quad$ **for** $i \leftarrow 1, 2, ..., n$
5 $\quad\quad$ **do for** $j \leftarrow 1, 2, ..., n$
6 $\quad\quad\quad$ **do if** $\sigma^k_{i,j} \neq 0$
7 $\quad\quad\quad\quad$ **then** $w^{k+1}_{i,j} \leftarrow \frac{r^k_i}{\sigma^k_{i,j}}$
8 $\quad \mathbf{R}_{k+1} \leftarrow \mathbf{W}_{k+1}\mathbf{R}_k$
9 $\quad d \leftarrow \|\mathbf{R}_k\|_1 - \|\mathbf{R}_k\|_1$
10 $\quad \mathbf{R}_{k+1} \leftarrow \mathbf{R}_{k+1} + d\mathbf{E}$
11 $\quad \delta \leftarrow \|\mathbf{R}_{k+1} - \mathbf{R}_k\|_1$
12 $\quad k \leftarrow k + 1$
13 **until** $\delta < \varepsilon$
14 **return** $(\mathbf{W}_k, \mathbf{R}_k)$

The algorithm consists of two parts, the update of the relation weight matrix (line 3 to 7) and the update of the concept importance vector (line 8 to 10). $\sigma^k_{i,j}$ is the sum of ranks of concepts which are i's backward concepts at step k. Damping factor α in Equation 4 is represented in vector as $\mathbf{E}$ where $\|\mathbf{E}\|_1 = \alpha$. Ignoring the differences in concepts, $\mathbf{E}$ is usually a uniform distribution. Threshold $0 < \varepsilon < 1$ controls the termination of the iteration. The algorithm returns $\mathbf{R}_k$ and $\mathbf{W}_k$ as the limits of the concept importance vector and the relation weight matrix.

5 Experiments

We study the feasibility of CARRank from three aspects: ranking qualities, semantic abilities, and efficiencies.

5.1 Experimental Settings

Evaluation Metrics. The metric for measuring the efficiency of ranking algorithms is the number of iterations k that minimizes the difference between two successive iterations $\| \mathbf{R}_{k+1} - \mathbf{R}_k \|$ to a given threshold ε. A smaller k indicates a faster convergence.

In order to measure the quality of concepts ranking results, we employ a variant first 20 precision metric [14]. The improved first 20 precision, $\widetilde{P@20} = \frac{n_{1\sim3}\times 20 + n_{4\sim10}\times 17 + n_{11\sim20}\times 10}{279}$, assigns different weights for the first 3, the next 7, and the last 10 results to increase the value for ranking effectiveness.

Similarly, we define $PR = \frac{\sum_{c\in C_{1\sim20}} \frac{m_c}{5}}{|C_{1\sim20}|}$ to measure the quality of relation ranking results, where $C_{1\sim20}$ is the relevant concepts in the first 20 most important concepts, and m_c is the count of relevant relations in the first 5 most important relations starting from concept c.

A higher value of $\widetilde{P@20}$ or PR means a better quality of ranking the importance of concepts or relations.

[4] $\mathbf{A}$ is obtained by parsing an ontology file into an RDF graph, and mapping it to an ontology graph, and finally constructed according to Definition 4.

Ranking Methods. Most of the related work in Section 2 are not specific for ontology understanding as shown in Table 1. Appropriate modifications are made in order to make them comparable. **1)** We choose the standard PageRank(**PR**) algorithm [5] on behalf of those PageRank-like algorithms. **2)** We extract the importance based labeling method from [20] which represents the methods that only consider concept hierarchy(**CH**). **3)** AKTiveRank [1] algorithm is modified by only considering the aggregation of density and betweenness measures (**DEM+BEM**) for each concept as the importance. CMM and SSM are irrelevant to the task of ontology understanding.

Experimental Environments. The experiments were carried out on a Windows 2003 Server with two Dual-Core Intel Xeon processors (2.8 GHz) and 3GB memory. For some ranking methods, let damping factor $\alpha = 0.85$, and threshold $\varepsilon = 1 \times 10^{-6}$ by default.

5.2 Ranking Qualities

Table 2: Four ontologies

	Concept#	Property#	URL
OWL	17	24	http://www.w3.org/2002/07/owl.rdf
Software Project	14	84	http://keg.cs.tsinghua.edu.cn/persons/tj/ontology/software.owl
Copyright Ontology	98	46	http://rhizomik.net/ontologies/2006/01/copyrightonto.owl
Travel Ontology	84	211	http://learn.tsinghua.edu.cn:8080/2003214945/travelontology.owl

To evaluate our proposed approach, we tried to collect representative ontologies and their accurate answers (a list of ranked concepts and relations) as possible as we could. In this experiment, four representative ontologies from the SchemaWeb[5] dataset are selected as shown in Table 2. "OWL" is a well-known meta ontology. "Software Project" is a full version of our running example which has a small number of concepts and relations, while, "Copyright Ontology" and "Travel Ontology" are more complex.

We take the ontology creators' feedback to the ranking task as the reference answers. We sent emails to the four contact creators, and got three ranks (for Software Project, Copyright Ontology, and Travel Ontology) and one suggestion (the creator of OWL recommended [23] as his answer) back in their replies. In our inquiry email, the following ranking instruction is described:

> For each ontology file, list top 20 (or as many as you like) important concepts (with URI) of your ontology in your mind. And for each top concept, please give top 5 (or as many as you like) important relations (with URI) for that concept.

With these reference answers, we compare CARRank with the four other ranking methods mentioned above and a user study. The user study was conducted on 5 volunteers whose research interests include the Semantic Web. We provided each volunteer the four ontologies that they never knew about before, in their original file formats, e.g. RDF or OWL. And then, for each ontology, volunteers were required to independently give the top 20 important concepts and the top

[5] http://www.schemaweb.info/

5 important relations for each top concept as their own ranking results. In this way, given one of the four ontologies, for each volunteer, we can computed a $\widetilde{P@20}$ value and a PR values according to his/her ranking results. The arithmetic means on five $\widetilde{P@20}$ values and five PR values are used to represent the corresponding metrics of the user study.

Table 3 and 4 present the comparisons on concepts and relations ranking for a full version of our running example. Here, we choose one of the five ranking results collected in the user study which has the highest $\widetilde{P@20}$ value.

Items listed in italic bold font are relevant ranking results. In Table 3, there are 5 relevant items in the first 10 ranking results for PageRank, 7 for DEM+BEM, 7 for CARRank, and 6 for the user study. Obviously, CARRank and DEM+BEM both have better ranking qualities than the user study. It means that they can somewhat support the ontology understanding. It also shows that PageRank is not a proper method in ranking the importance of concepts with less relevant results than the user study. Both CARRank and DEM+BEM rank concept "Project" the first place. The major difference of their results is that DEM+BEM considers "Person" and "Project_admin", while CARRank considers "Help" and "Release package". However, "Person" is relatively not important in this ontology because it is a base class of "Developer" and "Member" in the class hierarchy and rarely instantiated. PageRank fails in ranking "Project" the first place, which greatly lower its ranking qualities.

Table 3: The importance of concepts – Software ontology

Rank	Reference Answer	PageRank	DEM+BEM	CARRank	User Study
1	Project	***Message***	***Project***	***Project***	***Project***
2	Member	has_usage_statistics	***Usage_statistics***	***Usage_statistics***	***Category***
3	Developer	statistics_bugs	***Developer***	Statistic_record	***Message***
4	Category	statistic_record_support	Statistic_record	***Developer***	Discussion
5	Public_forum	***Member***	***Member***	***Category***	Help
6	LastestNew	***Project***	***Message***	Release_package	Person
7	Message	***Developer***	***Public_forums***	***Member***	***Member***
8	Version	***Category***	Person	***Message***	***Developer***
9	homepage	super_category	***Category***	Help	Project_admin
10	Usage_statistics	page_views	Project_admin	***Public_forums***	***Public_forums***

As the other four ranking methods do not directly support to rank the importance of relations, Table 4 only gives the comparisons of CARRank and the user study. It lists the first 5 relations (if available) starting from each concept of the first 5 concepts in the reference answers[6]. Apparently, CARRank can better reflect the importance of relations except for the concept "Project", since its ranking results are closer to the reference answers most of the time. For concept "Project", several owl:DatatypeProperty type relations, e.g. "title", "summary", "activity_ranking", and "project_homepage", are given in the reference answers. Such relations usually link to those simple data type values which have no outgoing edges hence very low importance as concepts. Therefore, according to Equation 3, owl:DatatypeProperty type relations are assigned low importance. We believe that it is beyond the scope of link analysis ranking algorithms.

[6] In fact, every concept listed in Table 4 has more than five relations except "Public_Forum". However, the creator could not provide us more relations than the reference answers.

Table 4: The importance of relations – Software ontology

Top 5 Concepts		Ranking results Reference Answer	CARRank	User Study
Project	1	title	has_usage_statistics	*project_homepage*
	2	summary	developed_by	*title*
	3	activity_ranking	belong_to_category	*activity_ranking*
	4	project_homepage	translations	has_public_forum
	5	project_of_statistic	intended_audience	has_usage_statisitics
Member	1	login_name	post_message	person_name
	2	publicly_displayed_name	*site_member_since*	
	3	email_address	*login_name*	
	4	user_id	*email_address*	
	5	site_member_since	*publicly_displayed_name*	
Developer	1	skills	member_of_project	person_name
	2	project_role	*project_role*	
	3		*skills*	
	4		user_id	
	5			
Category	1	hasProject	*hasProject*	*super_category*
	2	category_name	*sub_category*	*sub_category*
	3	super_category	*super_category*	*category_name*
	4	sub_category	*category_name*	*hasProject*
	5			
Public_Forum	1	hasMessage	*hasMessage*	*hasMessage*
	2	belong_to_project		
	3	project_of_forum		
	4			
	5			

We further examine the quality of ranking results with $\widetilde{P@20}$ and PR. The comparisons are illustrated in Figure 4 and Table 5. CARRank has some affirmative ability for helping ontology understanding, because it obtained a better result than the user study did. Though the precision of CARRank for "Software" is only about 4 percentage higher than that of users' decision, the degree of the support will be amplified along with the increase of the ontology's scale and complexity as shown in Figure 4. We find users can hardly decide the top important concepts for "Copyright Ontology" for its complexity. Obviously, CARRank is helpful in this case. Another interesting observation is that our algorithm is also effective to those meta ontologies like "OWL".

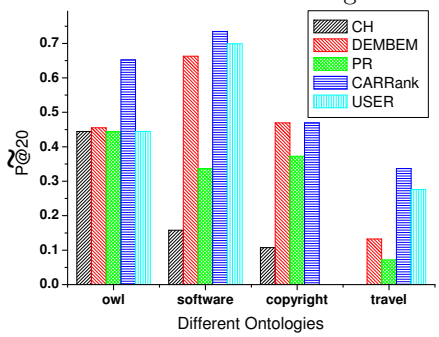

Fig. 4: The Comparison of Ranking Concepts

Table 5: The Comparison of Ranking Relations

	CARRank	User
copyright	0.06	0
software	0.586	0.562

5.3 Comparison of Semantic Abilities

To exhibit the semantic abilities of CARRank, we generate three variations of FOAF ontology[7], i.e. OWL-Full, OWL-DL, and OWL-Lite, with a tool named

[7] http://xmlns.com/foaf/spec/

foaf_cleaner [2]. Then, CARRank is applied on the three versions of FOAF and the original FOAF. Results are shown in Table 5.

Fig. 5: Top 10 Concepts for FOAF

	Original	OWL-Full	OWL-DL	OWL-Lite
Person	1	1	1	1
Document	2	2	2	2
Organization	3	3	3	5
Project	4	4	4	4
Agent	5	5	5	3
OnlineEcommerceAccount	6	6	6	7
OnlineChatAccount	7	7	7	8
OnlineGamingAccount	8	8	8	9
OnlineAccount	9	9	9	10
PersonalProfileDocument	10	10	10	11
Image	11	11	11	6
Group	12	12	12	12
Pearson Correlation Coefficients		1.0	1.0	0.867

Table 6: Top 10 Concepts for CYC

Rank	Concepts
1	RNAPolymerase
2	ExtensionOf-C-Regular
3	ClosedUnderGeneralizations-Classical
4	NetworkPortNumber
5	SimpleWord
6	GLFGraph
7	BrigadeOrRegimentSized
8	BrigadeOrRegimentSized
9	ExtensionOf-K-Normal
10	GLFAnalysisDiagramGraph

There are totally 12 concepts involved. The values in the first two columns are the concepts and their ranks produced by applying CARRank on the original FOAF ontology. The values in the last three columns are the ranks for the three versions. We use the Pearson Correlation Coefficient to measure the similarity of ranking results between one OWL version and the original version. The ranking results for the OWL-Full and OWL-DL are the same as that for the original one, though owl:imports of the OWL and RDFS ontologies are removed from the original, and owl:InverseFunctionalProperty on owl:DatatypeProperty is removed from OWL-Full. The only affection happens to the ranking results of OWL-Lite when owl:disjointWith is removed from OWL-DL. However, the similarity is still over 85%. This indicates that CARRank can capture most of the semantics even when the language expressive power changes.

Another challenge for semantic abilities of CARRank is to rank large scale ontologies, e.g. CYC[8] (23.7MB). Large scale ontologies are always developed collaboratively by many creators for a long time. Because of the limitations of individual creator and the limitation of the time, a global design intention may be unstable or even inconsistent. The interesting ranking results of CYC are listed in Table 6. There are 30432 classes and properties defined with 254371 RDF triples. It seems that CARRank ranks higher some abstract concepts for their complicated class hierarchy constructed with rdfs:subClassOf. Although it is hard to determine the quality of ranking results for such large scale ontology, we still suggest to use CARRank to periodically rank the concepts during its composition in order to discover early the deviation of design intention.

5.4 Efficiencies

Convergence Comparison. Figure 6 presents the comparisons among PageRank, Reverse PageRank, and CARRank. Rankings are performed on "Relationship"[9] ontology which has 169 vertices and 252 directed labeled edges in its

[8] http://www.cyc.com/2004/06/04/cyc
[9] http://purl.org/vocab/relationship/

ontology graph. Obviously, CARRank and Reverse PageRank have conformable convergent speed because both consider the hub score instead of authority score. The only difference is that the additional time spent on updating the relation weight matrix makes CARRank a little slower than Reverse PageRank.

On the other hand, the convergent speed of both CARRank and Reverse PageRank are quite different from that of PageRank. The reason is that PageRank considers authority score instead of hub score. Therefore, the convergent speed may be various with respect to the topological structure of the ontology graph. In Figure 6 the convergent speed of PageRank is much faster. However, take "UNSPSC"[10] ontology on SchemaWeb for another example. There are 19600 vertices and 29386 directed labeled edges. As shown in Figure 7, CARRank and Reverse PageRank express the same convergent speed and converge to the threshold early than PageRank. In any case, the convergent speed is acceptable for CARRank.

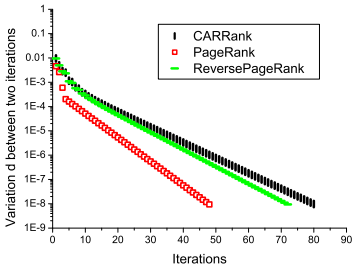

Fig. 6: Convergence ("Relationship")

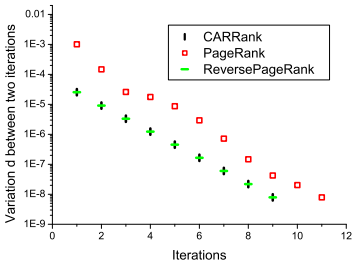

Fig. 7: Convergence ("UNSPSC")

6 Conclusion and Discussion

CARRank is a simple yet effective algorithm for identifying potentially important concepts and relations in an ontology. The experimental results show the feasibility of CARRank from the ranking qualities and the semantic abilities.

Although ontology understanding means much more than our proposed solution. we expect CARRank to be a preliminary step towards identifying potentially important concepts and relations user-independently. In addition, we also agree that being user-independent may not meet all the needs of application. Fortunately, CARRank can be personalize by letting user provide a sub-graph of the ontology which mainly contains the concepts and relations concerned about. It would be interesting to explore the ranking based on users' tasks and needs in the future work.

Acknowledgments

We would like to thank all the ontology creators who contributed their ranking results, and all the reviewers for their constructive comments and suggestions.

[10] http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml

References

[1] H. Alani, C. Brewster, and N. Shadbolt. Ranking ontologies with aktiverank. In *ISWC*, Athens, GA, USA, November 2006.

[2] R. Alford. Using FOAF and OWL, July 2005. http://www.mindswap.org/2005/foaf_cleaner/.

[3] A. Balmin, V. Hristidis, and Y. Papakonstantinou. Objectrank: Authority-based keyword search in databases. In *VLDB*, pages 564–575, 2004.

[4] E. P. Bontas and M. Mochol. Towards a cost estimation model for ontology engineering. In *Berliner XML Tage*, pages 153–160, 2005.

[5] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30(1-7):107–117, 1998.

[6] P. Buitelaar, T. Eigner, and T. Declerck. Ontoselect: A dynamic ontology library with support for ontology selection. In *The Demo Session at the ISWC*, 2004.

[7] T. d'Entremont and M.-A. Storey. Using a degree-of-interest model for adaptive visualizations in protégé. In *9th International Protégé Conference*, 2006.

[8] L. Ding, R. Pan, T. Finin, A. Joshi, Y. Peng, and P. Kolari. Finding and Ranking Knowledge on the Semantic Web. In *ISWC*, pages 156–170, November 2005.

[9] N. A. Ernst, M.-A. Storey, and P. Allen. Cognitive support for ontology modeling. *Int. J. Hum.-Comput. Stud.*, 62(5):553–577, 2005.

[10] D. Fogaras. Where to start browsing the web? In *IICS*, pages 65–79, 2003.

[11] T. R. Gruber. What is an ontology?, Dec 2001.

[12] W. James. *The principles of psychology.* Harvard, 1890.

[13] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[14] H. V. Leighton and J. Srivastava. First 20 precision among world wide web search services (search engines). *Journal of the American Society for Information Science*, 50(10):870–881, 1999.

[15] J. Mei and H. Boley. Interpreting swrl rules in rdf graphs. *Electr. Notes Theor. Comput. Sci.*, 151(2):53–69, 2006.

[16] R. Neches, R. Fikes, T. Finin, T. Gruber, R. Patil, T. Senator, and W. R. Swartout. Enabling technology for knowledge sharing. *AI Mag.*, 12(3):36–56, 1991.

[17] Z. Nie, Y. Zhang, J.-R. Wen, and W.-Y. Ma. Object-level ranking: bringing order to web objects. In *WWW*, pages 567–574, 2005.

[18] C. Patel, K. Supekar, Y. Lee, and E. K. Park. Ontokhoj: a semantic web portal for ontology searching, ranking and classification. In *WIDM*, pages 58–61, 2003.

[19] M. Sabou, V. Lopez, and E. Motta. Ontology selection for the real semantic web: How to cover the queens birthday dinner? In *Managing Knowledge in a World of Networks*, LNCS, pages 96–111, 2006.

[20] K. Tu, M. Xiong, L. Zhang, H. Zhu, J. Zhang, and Y. Yu. Towards imaging large-scale ontologies for quick understanding and analysis. In *ISWC*, 2005.

[21] W3C. Resource Description Framework (RDF): Concepts and Abstract Syntax, 2004. http://www.w3.org/TR/rdf-concepts/.

[22] W3C. SWRL: A Semantic Web Rule Language Combining OWL and RuleML, 2004. http://www.w3.org/Submission/SWRL/.

[23] T. D. Wang, B. Parsia, and J. Hendler. A survey of the web ontology landscape. In *ISWC*, 2006.

[24] G. Wu. Understanding an ontology by ranking its concepts and relations. Technical report, Tsinghua University, Jan 2008. http://166.111.68.66/persons/gangwu/publications/kegtr-carrank.pdf.