Expert Systems with Applications 39 (2012) 4111–4121

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Searching and ranking method of relevant resources by user intention on the Semantic Web

Myungjin Lee [a], Wooju Kim [a,*], Sangun Park [b,1]

[a] Dept. of Information and Industrial Engineering, Yonsei University, 134 Shinchon-dong, Seodaemoon-ku, Seoul 120-749, Republic of Korea
[b] Division of Business Administration, Kyonggi University, 94-6 Yiui-dong, Yeongtong-gu, Kyonggi 443-760, Republic of Korea

ARTICLE INFO

Keywords:
Semantic search
Ontology retrieval
Semantic Web
Semantic associations
Spreading activation

ABSTRACT

As the information on the Internet dramatically increases, more and more limitations in information searching are revealed, because web pages are designed for human use by mixing content with presentation. In order to overcome these limitations, the Semantic Web, based on ontology, was introduced by W3C to bring about significant advancement in web searching. To accomplish this, the Semantic Web must provide search methods based on the different relationships between resources.

In this paper, we propose a semantic association search methodology that consists of the evaluation of resources and relationships between resources, as well as the identification of relevant information based on ontology, a semantic network of resources and properties. The proposed semantic search method is based on an extended spreading activation technique. In order to evaluate the importance of a query result, we propose weighting methods for measuring properties and resources based on their specificity and generality. From this work, users can search semantically associated resources for their query, confident that the information is valuable and important. The experimental results show that our method is valid and efficient for searching and ranking semantic search results.

1. Introduction

Information retrieval (Singhal, 2001) is the science of searching for relevant documents, information within documents, or metadata about documents. Most information retrieval systems compute a numeric score based on how well each object matches with the user's query and rank the objects according to their scores. After that, the top ranking objects are displayed for the users. The first generation of automated information retrieval systems was introduced in the 1950s and 1960s. These systems highlighted the need for research into information retrieval technology, after which Tim Berners-Lee suggested a hypertext project, called the World Wide Web (Lee & Cailliau, 1990), in 1989. From the onset of the World Wide Web, continued progress in network technologies and data storage techniques has enabled the digitalization of huge numbers of documents. Consequently, search engines became very common and may be the best instantiation of information retrieval models in the huge hyperlink network. However, the existing web makes it more and more difficult for users to find relevant information, as the available information continues to dramatically increase. One of the problems is that web pages are designed by mixing content with presentation. Hyper Text Markup Language (HTML) (World Wide Web Consortium, 1992) focuses on describing the structures of web pages. HTML provides users with the means to display online documents with headings, text, tables, lists, and photos and to retrieve online information via hypertext links. Another problem is that a web search engine is typically based on keyword search techniques. Generally web search engines gather information from different sources via a web crawler by collecting, parsing, and storing keyword data from the texts of web pages to facilitate fast and accurate information retrieval. When users enter query keywords into a search engine, the web search engine examines its index of resources and provides a list of best-matching web pages according to its page ranking criteria based only on the query keywords. In order for users to locate information accurately, web page semantics should be separate from the syntax of HTML, and the web search engines need to be able to search for information based on semantics, rather than only on keywords. This need for semantics has lead to the creation of the Semantic Web.

The Semantic Web is an evolutionary progression of the World Wide Web in which the semantics of information and services are defined, making it possible for the web to satisfy user requests for web content (Lee, Hendler, & Lassila, 2001). This new system is

* Corresponding author. Tel.: +82 2 2123 7754; fax: +82 2 364 7807.
E-mail addresses: xml@yonsei.ac.kr (M. Lee), wkim@yonsei.ac.kr (W. Kim), supark@kgu.ac.kr (S. Park).
[1] Tel.: +82 31 249 9459.

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.09.127

based on the idea of providing information with explicit and formal machine-accessible descriptions of meaning. In order to make and exchange the semantics of information, the ontology that defines a common information-sharing vocabulary is generally used. An ontology, which is a formal explicit description of concepts or classes in a domain of discourse (Gruber, 1993), can be used to annotate data using metadata and interrelations. An ontology in this context consists of resources on the World Wide Web and their relationships – the network structure. The current web is a network structure that consists of web pages, with only one relationship denoting the hyperlink. However, an ontology produces a more complex network structure because it includes descriptions of concepts and represents various kinds of user-defined relationships between concepts; we call this a semantic network (Sowa, 1992). The applications in the Semantic Web can obtain an increased accuracy when processing information, providing the potential to improve the way in which search engines perform. Therefore, a different search method from that of the traditional keyword search is needed to identify relevant information in a user query. One of the core differences between the semantic search and the keyword search is the utilization of interrelationships among data, which is a resource in the Semantic Web.

The search method proposed in this paper allows for the identification of all concepts which are related to a user query even if the concepts do not explicitly include any query string. This ability is based on the spreading activation method (Crestani, 1997). Traditional search methods determine whether at least one of the query keywords appears within the documents and, if so, provide the documents to the user as a search result. For example, if a user inputs a query with the keywords "Metaweb Technology," then the search engine locates documents that include the query keywords and provides the documents to the user for review. However, our method based on spreading activation provides semantically related concepts (e.g., persons, companies, etc.) to the query "Metaweb Technology" as the search result. Therefore, the properties which relate resources in the Semantic Web are very important in semantic searching because they show why and how each resource is related to the query. Web pages in the current web are connected by only hyperlink relations, but resources in the Semantic Web are connected by one or more properties. This means, however, that different properties can imply different importances for the connected resources. The traditional keyword search generally shows just a ranked list of the keyword similarities between the user query and the identified documents. However, the search results of a semantic search can be sorted by the weights of the properties and resources, as each has its own individual importance. In order to rank the search results, the weights of properties and resources are assigned based on *specificity* and *generality*. A very interesting outcome has resulted from these weighting methods. The questions of where to start and how to span and explore the semantic network are the main issues in our research. Moreover, the visual presentation of the search results is also an important issue.

In this paper, we propose a semantic search method based on the spreading activation method and used to locate relevant results which are most semantically related to a user query. The approach is to retrieve all concepts that are related to a given keyword even if the keyword does not appear within the document. Moreover, we discuss the assignment of the weights of properties and resources in order to support semantic searching, to provide users with properly ranked search results. In other words, the processes of assigning weights and spreading to other resources on the semantic network are examined. Finally we evaluate the proposed search method over real-world data to compare our approach with another semantic search method and to test the effects of weighting.

The paper is organized as follows. Section 2 discusses related works. In Section 3 we define the data model and describe the weighting method for properties and resources. In Section 4 we propose a semantic search method. Section 5 discusses our semantic search system, and we present experimental results in Section 6. Finally we conclude the paper and our future work in Section 7.

2. Related works

Recently, a number of semantic search approaches have been published, and their application areas are diverse. However, they are based on a common set of ideas, presented and connected by Mangold (2007). Mangold presented a categorization scheme that is used to classify different approaches for semantic searches along several dimensions. In particular, he introduced categories for the following criteria: architecture, coupling, transparency, user context, query modification, ontology structure, and ontology technology. He selected ten different semantic document retrieval systems, i.e., Simple HTML Ontology Extensions (SHOE) (Heflin & Hendler, 2000), Inquirus2 (Glover, Lawrence, Gordon, Birmingham, & Giles, 2001), TAP (Guha, McCool, & Miller, 2003), etc. He compared the systems by means of the classification criteria and discussed issues that are open to further research and application development. According to his research, our system can be classified as a tight coupling between web pages and the ontology, meaning that the metadata of documents refer explicitly to concepts of a specific ontology. Therefore, our approach is classified as a graph-based approach that perceives both ontological concepts and documents as the nodes of a graph.

The MultimediaN E-Culture project (Schreiber et al., 2008), one of the semantic search systems, demonstrates how the novel Semantic Web and presentation technologies can be deployed to provide better indexing and search support within large virtual collections of cultural heritage resources. To search semantic paths, this system checks all RDF literals in the repository for matches to the given keyword and traverses the RDF graph until a resource of interest is found. Finally, the results are clustered based on the paths from the matching literals to their result. This research has some similarity with our approach, but it lacks the ability to assign weights to properties and resources and the ability to identify the search-ending threshold. These limitations are the most important issues with this system, because they are two of the determinants for expanding the semantic network. Moreover, information travels in one direction only in this limited system: always from the query object in the triple to the corresponding subject.

Some ranking techniques for the Semantic Web have been proposed. SemRank (Anyanwu, Maduko, & Sheth, 2005), which ranks results based on their predictabilities, is one of them. SemRank is based on a relevance model which is a rich blend of semantic and heuristic-including information-theoretic techniques that support the novel idea of modulative searches, in which users may vary their search modes to effect changes in results ordering depending on their need. To rank results of semantic associations, a model is constructed for measuring the information content of a semantic association by considering the occurrence of an edge as an event and RDF properties as its outcomes. In other words, SemRank proposes a measure of a property's uniqueness relative to those of other properties in the description base. However, to apply the spreading activation method in a semantic search, available resources must also be considered. Accordingly, we expand this property measurement and propose a new measure that takes into consideration the uniqueness of the resources in the semantic network.

In associative retrieval, relationships among information items are often represented as a network, where information items are represented by nodes, and associations are shown as the links

connecting the nodes. The spreading activation model (Cohen & Kjeldsen, 1987) is a method for searching and processing associative networks or semantic networks. The pure spreading activation model (Crestani, 1997) is made up of a conceptually simple processing technique for a network data structure. The search process of spreading activation is initiated by labeling a set of source nodes with weights or activation values and then iteratively propagating or spreading those activation values to other connected nodes. Mostly, these weights are real values that decay as activation propagates through the network. The activation method may originate in alternate paths, identified by distinct markers and terminate when the two alternate paths intersect at the same node. The results of the spreading activation process are the activation levels of nodes and the semantic paths reached at termination. However, the pure spreading activation model has some serious drawbacks. Therefore, some heuristics or inference rules are proposed to enhance the pure model. Distance constraint, fan-out constraint, path constraint, and activation constraint are commonly used in the spreading activation model. Basically, our search method is based on the extended spreading activation model, using some constraints for the termination condition.

RSS (Ning, Jin, & Wu, 2008), a framework enabling ranked searching on the Semantic Web, is a semantic search system based on the spreading activation method. The heterogeneity of relationships is fully exploited to determine the importance of resources; for example, supporting a semantic search and providing users with properly ordered search results. This system manually assigns edge weights on the schema level and applies them to the instance level. To globally rank the importance of resources in the data instance graph, the system uses a random surfing model that performs a Markovian walk following an edge whose transition probability is the same as the uniform probability applied in standard PageRank. Then an extended spreading activation algorithm, proposed to retrieve the resources most semantically related to the query, is applied. However, it is inappropriate for users to manually assign edge weights because an ontology generally has many kinds of properties. Therefore, node weights are typically not properly reflected in spreading activation because they are used to determine the initial activation value. In order to solve these problems, we propose methods for measuring the weights of properties and nodes and for effectively applying spreading activation.

3. Calculating edge and node weights

In this section we first define the data model of the knowledge base of the Semantic Web. After that, we discuss the methods for assigning the weights of properties and resources and for normalizing spreading activation.

3.1. Definition of knowledge base

Ontology is a mechanism for representing formal and shared domain knowledge, consisting of a hierarchy of concepts, role relations among concepts and instances attached to concepts. The ontology that is used in the Semantic Web provides an understanding of the domain of a data set, and all information is semantically related and linked, so that the user can search for relationships between information. Therefore, semantic search results consist of semantically associated concepts.

The goal of our semantic search method is to rank the search results after applying spreading activation. In order to apply the spreading activation method and to rank the search results, we need to extend the definition of ontology to include the weights of properties and instances. An extended knowledge base for spreading activation can be represented, as follows:

$$KB = (I, C, P),$$
$$C = \{c_i | c_i \in \text{owl:Class}\},$$
$$P = \{\langle p_i, f(p_i)\rangle | p_i \in \text{rdf:Property}, f(p_i) \in \Re^+\},$$
$$I = \{\langle i_i, f(i_i)\rangle | i_i \in c_j, f(i_i) \in \Re^+\},$$

where C denotes the set of concepts, P denotes the set of properties, and I denotes the instance set of all concepts. Specially, let $c_i \in$ *owl:Class* denotes a concept, $p_i \in$ *rdf:Property* denote a property, and $i_i \in c_j$ denote an instance of concept c_j. $f(x)$ denotes a function for measuring the property and instance weights. The weight function $f(x)$ consists of two steps. The first step is to calculate the weight based on *specificity* or *generality*, and the second step is normalization to a real number. Section 3.2 will further discuss the weighting function. This weighting function is assigned to instances and properties that connect instances. In other words, only instances and properties have weight values. Because the target of a semantic search through spreading activation is limited to ABox statements associated with class instances, the search results of our search method are a set of instances related to a user query.

OWL has two types of properties: object properties that show relationships between instances of two classes and datatype properties that show relationships between class instances, RDF literals and XML Schema datatypes. We assume that datatype properties are treated with instance values in our case. Our method expands semantic searching to include object properties and instances over the semantic network because datatype properties are not relationships between instances. We use datatype properties for comparing the similarity between a concept and a query string and to provide additional information about concepts.

3.2. Edge and node weights based on specificity

In information theory (Shannon, 1948), self-information is a measure of the information content associated with the outcome of a random variable. By definition, the amount of self-information contained in a probabilistic event depends only on the probability of that event: the smaller is its probability, the larger is the amount of self-information associated with the event actually occurring. Further, by definition, the measure of self-information has the following property: if an event C is composed of two mutually independent events A and B, then the amount of information at the proclamation that C has happened equals the sum of the amounts of information at the proclamations of event A and event B. Self-information is measured by the negative logarithm of the probability of the occurrence of the event. Taking into account these properties, the self-information $S(\chi = x_i)$ associated with outcome i with probability pr_i is:

$$S(\chi = x_i) = \log\left(\frac{1}{\log pr_i}\right) = -\log pr_i.$$

If χ is a discrete random variable or an event that has possible outcome values x_1, x_2,…,x_n occurring with probabilities pr_1, pr_2,…,pr_n, the self-information gained or the uncertainty removed by knowing that χ has the outcome x_i is given by the above formula. In other words, χ denotes an event that has a possible outcome value, x_i, occurring with probability pr_i.

Based on this, SemRank (Anyanwu et al., 2005) developed a model for measuring the self-information of a semantic association by considering the occurrence of edge as an event and the RDF properties as its outcomes. The notion for a property was first defined and then extended to a sequence of properties along a path. In order to measure the self-information of edges, they defined two types of measure; *specificity* and *θ-specificity*. The *specificity* of a property is a measure of its uniqueness relative to all other properties in the semantic network of ABox. Therefore, the *θ-specificity* of

a property is a measure of its uniqueness relative to all other properties in the ABox whose domain and range belong to the same semantic network. To determine the total self-information gain of the property, the values of self-information due to both *specificity* and *θ-specificity* are combined.

As noted above, SemRank proposed only a measure of a property's uniqueness relative to those of the other properties. However, to apply spreading activation over the semantic network, we need to consider the weights of nodes. Accordingly, we expand this measure of properties and propose a new measure that considers the uniqueness of resources for spreading the network. For any valid instance $i \in I$, the probability that $\chi = i$ is given by:

$$\Pr(\chi = i) = \frac{|(i,*,*) + (*,*,i) - (i,*,i)|}{|(*,*,*)|}.$$

$\Pr(\chi = i)$ denotes the probability of triples that include instance i in the ABox statements and is the *specificity* of instance i. The *specificity* of a resource is a measure of its uniqueness relative to all other resources. The information content of the occurrence of an instance i in the semantic network due to its *specificity* is:

$$S_S(i) = S(\chi = i) = -\log \Pr(\chi = i),$$

where $S_s(i)$, the *specificity* of instance i, denotes the self-information of an instance i based on the probability of its occurrence relative to all other instances I.

Likewise, it is possible to develop a similar measure which exploits the semantics of RDF and RDFS. In the RDF Schema, there are two properties, domain and range, for describing how properties and classes are intended to be used together in RDF data. The domain property is used to indicate that a particular property applies to a designated class, and the range property is used to indicate that the values of a particular property are instances of a designated class or to indicate that the value of a property is given by a typed literal. If one property has the domain or range assertion of it in the ontology, a subject or object of triples using that property is an instance of class that is designated by the domain or range assertion even the type of instance is not declared. Thus, given the RDF resources are typed, any two instances i_1 and i_2 have a finite number of valid properties that connect them. Using this information, we defined the *θ-specificity* of an instance. In other words, the *specificity* of an instance is a measure relative to other all instances, and the *θ-specificity* of an instance is a measure relative to other valid instances connected by any property whose domain or range is classes of a given instance. In order to define the *θ-specificity* of an instance, given that the properties are typed, any property p has a finite number of valid instances that may be connected by property p. Therefore, we can estimate the self-information of an instance i linked by p with respect to only the valid instances.

θ_i represents the interpretation of all of the valid instances that may be connected by p_i. Thus, the probability that $\chi \in \theta_i$ is given by:

$$\Pr(\chi \in \theta_i) = \frac{|\theta_i|}{|(*,*,*)|} = \frac{|(*,p_i,*)|}{|(*,*,*)|},$$

where p_i denotes the property, with its domain or range defined as a class of instance i. For a given valid instance i, if $\chi \in \theta_i$, the probability that $\chi = i$ is given by:

$$\Pr(\chi = i|\chi \in \theta_i) = \frac{\Pr(\chi = i)}{\Pr(\chi \in \theta_i)} = \frac{|(i,*,*) + (*,*,i) - (i,*,i)|}{|(*,p_i,*)|}.$$

This probability indicates the ratio of the number of connections where the instance i is used to the number of connections where the valid instances are used. The *θ-specificity* of a resource is a measure of its uniqueness relative to all other instances whose class belongs to the same semantic network. The self-information of the occurrence of a valid instance i in the semantic network due to its *θ-specificity* can then be defined as:

$$S_{\theta-S}(i) = S(\chi = i|\chi \in \theta_i) = -\log \Pr(\chi = i|\chi \in \theta_i).$$

In order to determine the total self-information of instances, we combine the self-information of *specificity* and *θ-specificity*:

$$S_i(i) = S_S(i) + S_{\theta-S}(i).$$

Nodes that denote instances on the semantic network have their own weight value based on uniqueness. Nodes that are a sparsely occupied are more valuable than nodes that are densely occupied. Weights of nodes are used for propagating activation to the semantic network and for calculating the rank values of search results and initial activation values of starting nodes that are similar to the query string.

3.3. Normalization of weight values

We discussed how to assign weights to edges and nodes based on uniqueness. However, values of *specificity* and *θ-specificity* that are computed by the above formula have large values. In general, this is because the frequency of a property is a small number of triples in the semantic network, but the ontology is vast, even if it includes a specific domain. Because we use a negative logarithm function to measure the weight value, if the probability of an edge or node comes close to 0, then its weight value approximates infinity. Thus, these weight values are not fit for use in the spreading activation method. We define a function to normalize the weight values, converting them from considerable numbers to real numbers in the range [0.0...1.0].

In order to aggregate a sequence of numbers into the range of [0.0...1.0], we use a logistic function of the most common sigmoid curve (Bob, Hay, & Jannedy, 2003). This function models the S-shaped curve of growth of some set P, where P is a population. The initial stage of growth is approximately exponential. As saturation begins, the growth slows, and, at maturity, the growth stops. A simple logistic function (Gershenfeld, 1998) may be defined by the formula:

$$P(t) = \frac{1}{1 + e^{-t}},$$

where the variable P is a population, and the variable t is time (Weisstein). At $t = 0$, the logistic function $P(0) = 0.5$. As t increases, $P(t)$ increases rapidly at first but then more slowly as it approaches its upper bound of 1. The lower bound of the function is 0. If we now let t range over the real numbers from $-\infty$ to $+\infty$, then we obtain the S-curve in a range from 0 to 1.

However, the weight values of our measures, based on a negative logarithm function, range over the real number from 0 to $+\infty$; thus, the simple logistic function is not appropriate for normalizing the weights of edges and nodes. Therefore, we need to adjust the logistic function for application to our formula. The changed logistic function for normalization is defined by the formula (Smith, 1993):

$$P(t) = \frac{1}{1 + e^{-y\cdot(t-x)}},$$

where x denotes the mean of the distribution and y denotes the scale parameter. In other words, the curve of the logistic function shifts as the values of x and y control the slope. At $t = x$, the logistic function $P(t) = 0.5$.

At first, we use an average to shift the center point of the logistic function. Namely, the curve of the logistic function shifts to the right as the weight value approaches the average. Therefore, to adjust the slope of the logistic curve, we use the standard deviation

and the standard normal distribution. The standard deviation of a statistical population is the square root of its variance and a widely used measure of the variability or dispersion (Dodge, 2006). It shows how much variation there is from the average. It may be thought of as the average difference of the scores from the mean of distribution, how far they are away from the mean. A low standard deviation indicates that the data points tend to be very close to the mean, whereas high standard deviation indicates that the data are spread out over a large range of values. Conversely, having a standard normal random variable, Z, we can always construct another normal random variable with specific mean, μ, and variance, σ^2. A standard normal random variable, Z, is set at 1.285, where the standard normal distribution accounts for about 90% of the sample population. We assume that weights which are not included in this 90% can be treated as outliers and can be ignored by the normal distribution. The slope of the logistic curve is adjusted until the normal random variable X is 0.9 in the logistic function.

Finally, we calculate the logistic function that shifts and adjusts the slope of the curve to the normalized weight values. We need two types of normalization functions because the ranges of the weight values of the edges and those of the nodes are different. The normalized function for the weight of the edges is defined by the formula:

$$\mathrm{NI}^p(p) = \frac{1}{1 + e^{-s_p \cdot (\mathrm{S}_p(p) - \mu_p)}},$$

where $\mathrm{S}_p(p)$ denotes the values of self-information of property p, μ_p denotes an average value of the edge weights $\mathrm{S}_p(p)$, and s_p denotes a slope value of the logistic function for the normalized edge weights.

The normalized function for the weight of the nodes is defined by the formula:

$$\mathrm{NI}^i(i) = \frac{1}{1 + e^{-s_i \cdot \left(\mathrm{S}_i(i) - \mu_i\right)}},$$

where $\mathrm{S}_i(i)$ denotes the values of self-information of instance i, μ_i denotes an average value of the node weights $\mathrm{S}_i(i)$, and s_i denotes a slope value of the logistic function for the normalized node weight.

Until now, we discussed the measurement of the weights of edges and nodes based on uniqueness. This weight method is helpful in finding information that is sparsely connected. We will show the experimental results of this measure in Section 6.

3.4. Edge and node weights based on generality

In this section, we discuss another method for measuring the weights of edges and nodes based on *generality*. These two ideas, *specificity* and *generality*, are polar opposites. Unlike with *specificity*, edges and nodes that have a lot of connections have higher weight values than do those that have only a few connections in the weight method based on *generality*.

In order to measure the weight based on *generality*, we defined the opposing function of self-information that defines a negative logarithmic function of probability in the range [0.0...1.0]. It is defined by the following formula:

$$\mathrm{G}(\chi = x_i) = -\log(-pr_i + 1).$$

With this weight method, $\mathrm{G}(\chi = x_i)$ is 0 where the probability pr_i is 0. As the probability pr_i approaches 1, its value approximates infinity. This is a problem in our case because the probability is often 1. For example, if there is only one property defined between two classes, then its probability by θ becomes 1. In this case, we assign *bigM* as its weight value since $\mathrm{G}(\chi = x_i)$ is $+\infty$. We set the maximum values of the edges or node weights to *bigM*.

The process for calculating weight value based on *generality* is equal to the process based on *specificity*; only the base function is different. *Generality* of each property and instance is defined by the following formula:

$$\mathrm{G}_p(p) = \alpha \cdot \left(-\log\left(-\frac{|(*, p, *)|}{|(*, *, *)|} + 1\right)\right) + (1 - \alpha) \cdot \left(-\log\left(-\frac{|(*, p, *)|}{|(i_i, *, i_j)|} + 1\right)\right),$$

where $i_i = \{instances\ of\ c | c \in p.domain\}$, $i_j = \{instances\ of\ c | c \in p.range\}$.

$$\mathrm{G}_i(i) = \alpha \cdot \left(-\log\left(-\frac{|(i, *, *) + (*, *, i) - (i, *, i)|}{|(*, *, *)|} + 1\right)\right) + (1 - \alpha) \cdot \left(-\log\left(-\frac{|(i, *, *) + (*, *, i) - (i, *, i)|}{|(*, p_i, *)|} + 1\right)\right),$$

where $p_i = \{p_j | p_j.domain \in C(i)\ or\ p_j.range \in C(i)\}$.

In order to measure the weights of edges and nodes based on *generality*, we define normalized functions for edge weight, $NG_p(p)$, and for node weight, $NG_i(i)$. This weight method is helpful for identifying information that has many connections.

4. Semantic searching based on spreading activation

In this section, we present our semantic search process which finds information related to the user query. Section 4.1 describes the semantic search process based on extended spreading activation. In Section 4.2, we discuss the constrained spreading activation model.

4.1. Semantic search algorithm

In order to search relative information from a query in the semantic network, we use the semantic search process, which extends the spreading activation algorithm (Collins & Loftus, 1975; Crestani, 1997; Preece, 1981). The spreading activation algorithm works by navigating the semantic network. Given an initial set of concepts from the user query, the spreading activation method obtains a set of closely related concepts by navigating through the linked concepts based on relationships. The extended spreading activation method is one of the main parts of the proposed semantic search system.

The pure spreading activation model is quite simple. The spreading activation process starts by placing a specified initial activation value on the starting nodes that is similar to that of the user query. The processing technique is defined by a sequence of iterations in which each iteration is followed by another iteration until halted by the user or upon triggering some termination condition. Generally, an iteration consists of a spreading phase and a termination check phase. In the spreading phase, the spreading action affects the other closest nodes to the starting node. The activation weight of a node is computed as a function of the weighted sum of the inputs to that node from the directly connected nodes. In the termination check phase, the procedure terminates when either there are no more nodes or when the constraint condition is satisfied. After the spreading process is complete, a set of nodes are obtained and ranked according to their activation values. There are many ways of spreading the activation over a network (Preece, 1981).

In the first step of spreading activation, the initial set of starting nodes is located by the query keyword from the ontology. The initial set includes the starting nodes of the spreading process and the initial activation value of each node. In other words, the initial set is composed of pairs of starting nodes and their initial activation

values. We now define the initial set of starting nodes in a semantic relation search

$$\text{Initial Set} \quad IS = \{(i_1, w_1), (i_2, w_2), \cdots, (i_n, w_n)\},$$

where i_i denotes the instance that is similar to the user query, and w_i denotes an initial activation value of a starting node. In fact, it is possible to set different weights to the initial activation values of starting nodes depending on the node's similarity with the user query. However, we assign the same weight, 1.0, to all starting nodes. It is one of our future goals to apply the relevance-like term frequency (Jones, 1972) to the initial activation value. If the user input "BLU," Back Light Unit, as a keyword in a semantic search, the proposed system searches all of the candidates from the ontology. The initial set includes all instances that have the word "BLU" included in their datatype properties, and its initial activation value is set at 1.0. The activation starts with the initial set.

After identifying the initial set, it spreads to all the other instances connected to the initial nodes. When spreading reaches another node, its output value must be determined. The output function of instance j is formulated, as follows:

$$A_j(t) = \tanh(IP_j(t)),$$
$$IP_j(t) = \underset{P_{ij} \in edges \ \ between \ \ i \ \ and \ \ j}{\text{MAX}} (NI^p(p_{ij}) \cdot A_i(t)) + A_j(t-1),$$

where t denotes a point of time, $A_j(t)$ denote an activation value of node j at time t, $IP_j(t)$ denotes an input value of node j from node i at time t, and $NI^p(p_{ij})$ denotes the normalized edge weight of property p that connects node i to node j. The activation value of every node at time 0, except the nodes in the initial set, is 0. The output of a node is given by the function *tanh*() that denotes a hyperbolic tangent (Spanier & Oldham, 1987). It has an S-shaped sigmoid function in the range from -1 to 1. In our case, the input value to the node is greater than 0; thus, the activation value of the node has a value between 0 and 1. The input value of a node is defined as the sum of the maximum value that multiplies the edge weight and the activation value of the previous node and the activation value of the node at time $(t-1)$. In the pure spreading activation model, a summation $\sum$ is used instead of the maximum. However, using the maximum value provided the best results in our initial test.

The input function of the node above considers only the edge weight from the previous node to the current node. The purpose of this paper is to determine more unique or more general weighting methods based on spreading activation. Using these new methods, users can get better results considering the weights of edges and nodes than when only considering the weights of edges. We will show this in Section 6. In order to apply node weight, we extend the input function, as below:

$$IP_j(t) = \underset{P_{ij} \in edges \ \ between \ \ i \ \ and \ \ j}{\text{MAX}} (NI^p(p_{ij}) \cdot NI^i(j) \cdot A_i(t)) + A_j(t-1),$$

where $NI^i(j)$ denotes the weight of node j.

Pulse after pulse, the activation spreads over the network, reaching distant nodes. The procedure terminates when either there are no more nodes or when the termination condition is achieved. After the spreading process is complete, a set of nodes and their activation values have been determined. The search results are ranked in order of activation value and provided to the user. Fig. 1 presents a pseudo code of our spreading activation algorithm.

However, this algorithm has some problems. One of the problems of spreading activation is that the propagation might reach the entire network. In order to solve this problem, the activation is spread according to constraint rules.

4.2. Constraints of spreading activation

As mentioned above, the spreading activation procedure terminates until there are no more nodes to fire; thus, the propagation might reach the entire network. In order to protect against spreading the entire network, Cohen and Kjeldsen (1987), as well as Crestani (Electronic, 1997) proposed some constraints. At first, a distance constraint corresponds to the simple heuristic rule that the strength of the relationship between two nodes decreases with their semantic distance. The spread of activation should cease when it reaches nodes that are more than a given distance from the initial set of nodes. The fan-out constraint is implemented to avoid an excessively wide spreading, which could derive from nodes with a very broad semantic meaning. Path constraint can be modeled using the weights on links or, if the links are labeled, diverting the activation flow to a particular path, while blocking it from following other less meaningful paths. Activation constraint is possible to control the spreading of the activation on the network using the threshold function at a single node level. This can be achieved by changing the threshold value in relation to the total level of activation over the entire network at any single pulse. Lastly, class constraint is used when the activation must not propagate through nodes of a given class type.

```
Set activationValue<node, value>
List paths
while(paths is not empty)
    path = remove(0) of paths
    preNode = getLastNode of path
    preActivationValue = getActivationValue of preNode
    Find liked nodes Nodes from preNode
    for each(node ∈ Nodes)
        if(visitedNode(node))
            exit
        max = Max(edgeWeight(from preNode to node) * preActivationValue)
        inputValue = max + getActivationValue of node
        activationValue = tanh(inputValue);
        Update activationValue of node
        Add node to path
        Add path to paths
```

Fig. 1. A pseudocode of the spreading activation algorithm.

Among these constraints, we use activation constraint, distance constraint, and class constraint. In order to apply activation constraint, we use the threshold of input function $IP_j(t)$, which the activation would not propagate from the node. We set the threshold to 0.1, at which the only input function considered is the edge weight. This is allowable because we assume that the values which are not included in the normalized 90% of the population are treated as outliers. The threshold is set to 0.01, at which the input function considers both edge weight and node weight. In order words, our search algorithm propagates to other nodes only when the input value is larger than the threshold value.

The distance constraint limits spreading according to the path distance. In order to apply the distance constraint, we re-defined the activation function, as follows:

$$A_j(t) = \tanh((1 - decayFactor \cdot depth) \cdot IP_j(t)),$$

where $decayFactor \cdot depth \leqslant 1$,

decayFactor denotes the decay factor, which decreases the activation value during every level of propagation, and *depth* denotes the propagation distance from the initial instance. The reason for using the decay factor is that the strength of association between instances decreases with increasing propagation distance. We assign 0.3 as the decay factor, meaning that the maximum propagation distance is restricted to three depths in the spreading process. As Crestani indicates, setting the maximum value to three steps is usually sufficient (Crestani, 1997).

Lastly, the purpose of the class constraint is to identify instances that are included in a specific user-appointed class. If a user sets a specific class in a query, then the spreading activation process tests and terminates when activation intersects with nodes of the given class type. A user can simply locate instances of the class of interest through this constraint.

5. Implementation of a semantic search

The aims of our research are to propose ontology-based semantic search methods in the Semantic Web. Fig. 2 shows the components of our association-based search system.

Web data is translated into a triple structure and stored on semantic metadata through the semantic annotator. Users can find information related to their query string using the semantic search system.

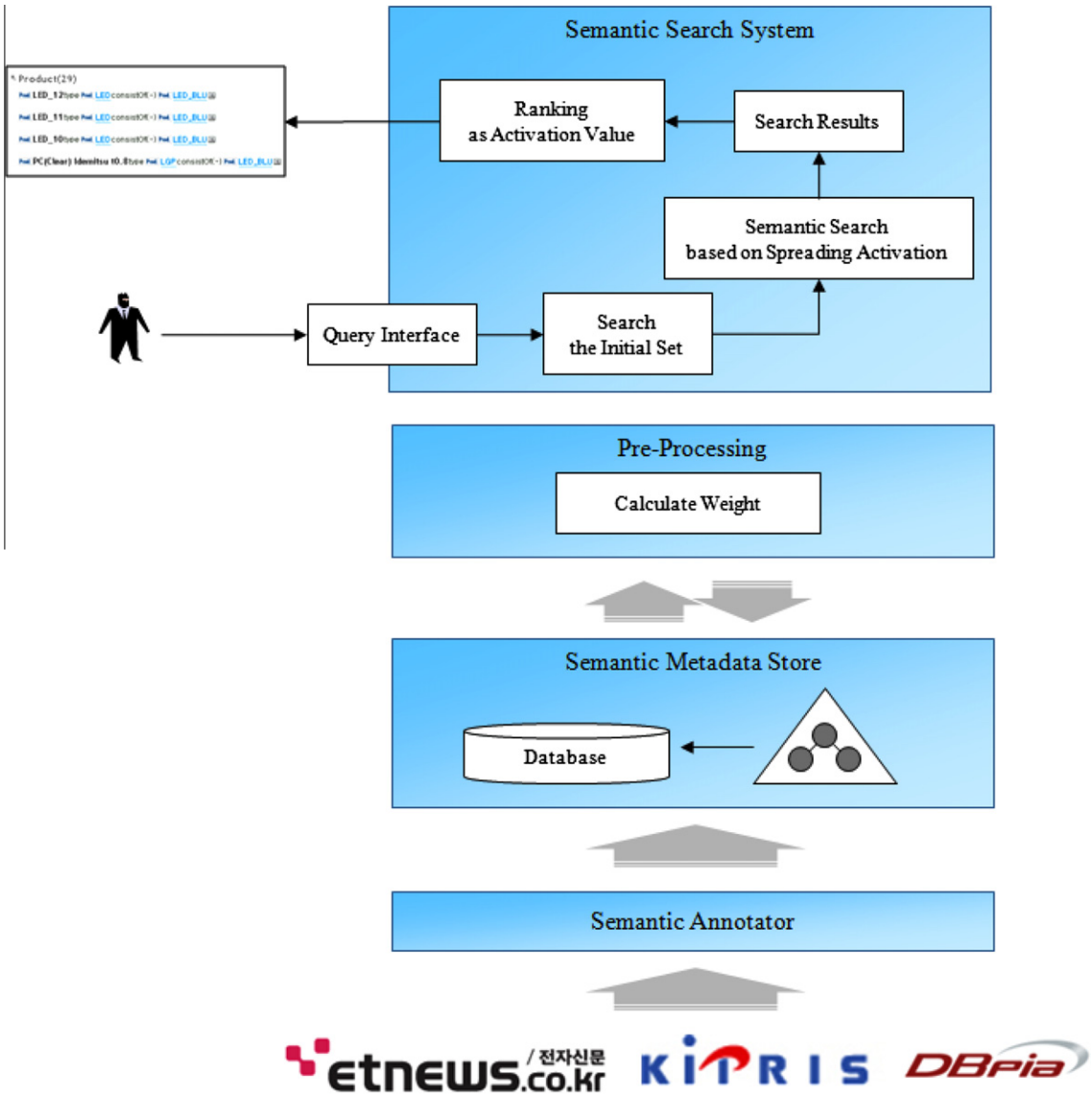
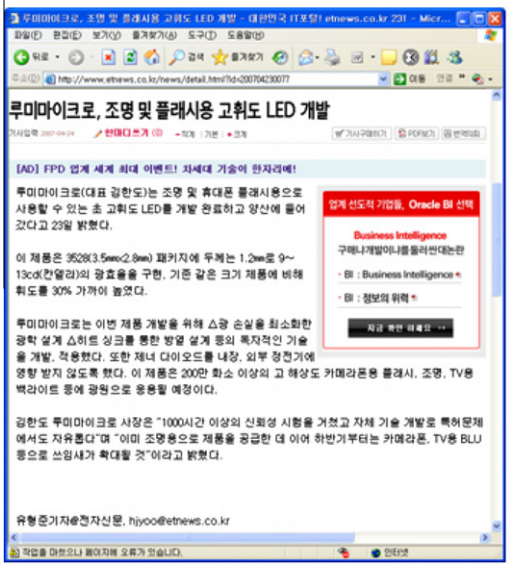


Fig. 2. Architecture of a semantic search system.




```
<News rdf:ID="NW006">
    <dc_source rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
        http://www.etnews.co.kr/news/detail.html?id=200704230077
    </dc_source>
    <dc_title>
        <Title rdf:ID="Title_22">
            <koreanTitle rdf:datatype="&xsd;string">
                루미마이크로, 조명 및 플래시용 고휘도 LED 개발
            </koreanTitle>
        </Title>
    </dc_title>
    <dc_creator>
        <Reporter rdf:ID="Reporter_uhyeongjun">
            <name rdf:datatype="&xsd;string">유형준</name>
        </Reporter>
    </dc_creator>
    <dc_date rdf:datatype="&xsd;dateTime">
        2007-04-24T00:00:00
    </dc_date>
    <articleCode rdf:datatype="&xsd;string">
        NW006
    </articleCode>
</News>
```

Fig. 3. Semantic annotation from a news web site.

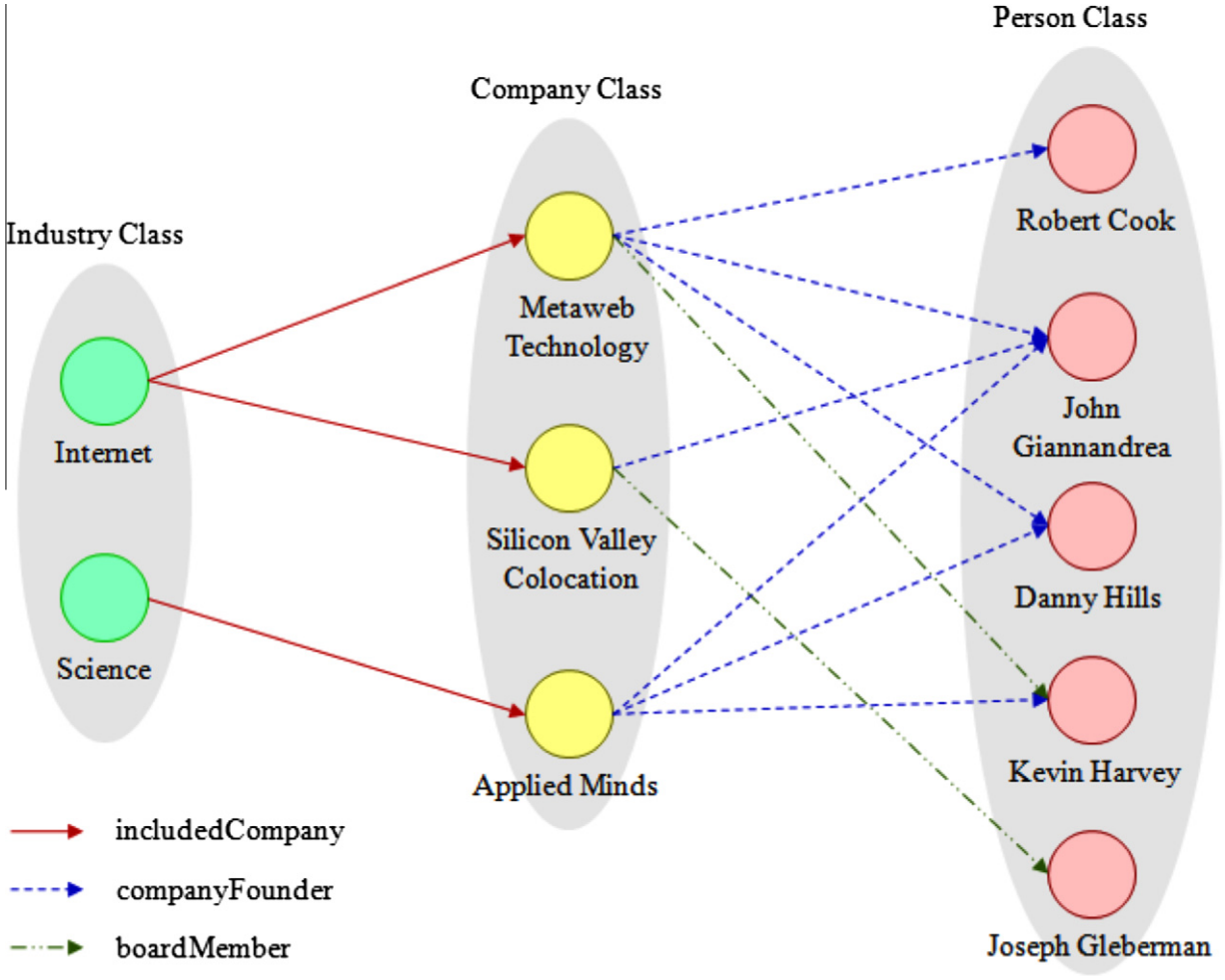


Fig. 4. Simple experimental ontology.

Web data is changed to ontology-based data in RDF and OWL through the semantic annotator, and its ontology is stored in the ontology database (Park, Lee, & Hong, 2007). In order to reduce search time, the weights of properties and instances for spreading activation are calculated during pre-processing before the semantic search, because it is a time-consuming process. If a user requests a semantic search with a query keyword, the search agent identifies an initial set for starting the spreading activation. After the spreading activation is complete, the search results are ranked according to their activation values. Consequently, the user is provided with instances and resources related to the query.

In order to search the semantic information, we constructed an ontology about information of the target domain. In order to build an ontology on the domain of electronics, we gathered information that was published in one year from different Korean data sources, such as an electronics news site (Electronic Times Internet), a patent service site (Korea Intellectual Property Rights Information Service), and a research paper service site (Smart Academic Knowledge Base DBpia). For building an ontology, we used semi-automatic semantic annotation. At first, we extracted concepts, like the title, author, keyword, URL, and published date, from each news source. We then made triples of relationships between concepts. Ontology that we obtained from different sites was included in seven categories – News, Person, Article, Product, Patent, and Technology, but the data sources are different. This means that the information from each source was separated from that of the other sources; it was impossible to connect information between sources, while independent searching was possible at each source. The users who work in the electronics industry, especially the companies which produce electronic parts, are very sensitive to new technology, trends, and parts producers. They read articles related to electronics and search for information about patents and technology from each site because the data sources are all different. This problem can be solved through an ontological search system with a semantic network because this information is semantically related over several fields. For example, a patent is related to a trend or technology, and the patent is owned by a company. Therefore, we transformed and combined the information of the sources into ontology so that they are connected to each other. This is one of the important improvements of the semantic search system.

Ontology provides an understanding of the domain of the data set. The ontology on the Semantic Web involves publishing in languages specifically designed for Web Ontology Language (OWL) based on the Resource Description Framework (RDF). The RDF data model is based upon statements in the form of subject-predicate-object expressions, called triples in RDF terminology. All information is semantically related and ontologically linked, so that the user can search for the relationships between information. Therefore, search results can represent semantic association paths between resources.

Fig. 3 shows a semantic annotation result from a news web site. We gathered title, author, date, source URL, and other information from each web site and built an OWL ontology, as shown at the right side of Fig. 3.

6. Semantic search experiment results

In this section, we experimentally evaluate our search algorithm. In order to demonstrate our semantic search system, our experiments consist of verification and evaluation. The experiments for verification are to ascertain the correctness of the search process, and the experiments for evaluation are to compare and judge our search results with other semantic association search methodologies.

In order to compare the effects and efficiencies of the weighting methods and the spreading activation constraint, we built a simple ontology that is illustrated in Fig. 4.

Before presenting the experimental result, we describe each weighting method of edges and nodes. We use two kinds of weighting methods, *specificity* and *generality*. The fewer edges or nodes that are used, the higher will be the *specificity* value assigned by the weighting method. On the contrary, the weighting method based on *generality* will assign a higher value to an edge that is used more frequently. Table 1 illustrates the weight values according to *specificity* and *generality*.

From this table, the instances of Cook and Gleberman have the largest weight value in the case of *specificity* because these are the least used. Therefore, the boardMember relation property is the largest. On the contrary, the Metaweb Technology instance has more connections than does the Internet instance, but the Internet instance has the largest weight value in the case of *generality* because it is a more general concept from the θ of view.

Table 3 shows the search results on the same conditions as in Table 2 but with an input function of spreading activation that considers both edge and node weights. The threshold is automatically set to 0.01, giving consideration to both edge and node weights. The node weight of Silicon Valley Colocation is larger than that of the node weight of Metaweb Technology, thus it is ranked first. Therefore, Joseph Gleberman is ranked second, though it is father from the starting node than is Metaweb Technology, because we did not consider a distance constraint. As is seen from the search results, we can get effective results when the spreading activation simultaneously considers edge and node weights.

Table 1
Weight values of edges and nodes.

Edge/node	Weight value	
	Uniqueness	Generalness
includedCompany	0.367	0.494
companyFounder	0.215	0.839
boardMember	0.863	0.165
Internet	0.223	0.933
Science	0.664	0.375
Metaweb Technology	0.134	0.861
Silicon Valley Colocation	0.390	0.455
Applied Minds	0.223	0.681
Robert Cook	0.901	0.160
John Giannandrea	0.300	0.572
Danny Hills	0.570	0.322
Kevin Harvey	0.570	0.322
Joseph Gleberman	0.901	0.160

Table 2
Search results using edge weight based on uniqueness and the activation constraint.

Ranking	Results	Activation value
1	Silicon Valley Colocation	0.351
	Metaweb Technology	0.351
2	Joseph Gleberman	0.294
	Kevin Harvey	0.294

Table 3
Search results using edge and node weights based on uniqueness and the activation constraint.

Ranking	Results	Activation value
1	Silicon Valley Colocation	0.142
2	Joseph Gleberman	0.110
3	Metaweb Technology	0.049
4	Kevin Harvey	0.024

Table 4 gives the top five search results when the spreading activation uses a distance constraint instead of the activation constraint in Table 3. In this example, the activation process considers every node as possible because of the small semantic network; thus, the original search results include every node. Silicon Valley Colocation is ranked first, and Metaweb Technology is ranked second because of the application of the distance constraint.

Lastly, Table 5 gives the top five search results where the spreading activation is applied with a distance constraint with edge and node weights based on *generality*. Table 5 shows the search results ordered by the number of connections. In other words, popular concepts can be identified if edge and node weights are applied to the spreading activation method based on *generality*.

We discussed the effects of weighting methods and spreading activation. The purpose of experiments for evaluation is to compare and judge the quality of our semantic association search system with other methodologies. In order to evaluate our works, we chose two comparable methodologies, SemRank (Anyanwu et al., 2005) and RSS (Ning et al., 2008). In order to evaluate these systems, we conducted a survey according to the evaluation method in RSS. First, we chose five test queries which are all on the ontology for experiments, and we got the top ten search results of each question from four methodologies: SemRank, RSS, and our systems based on *specificity* and *generality*. Then we invited ten experts who hold Ph.D.'s or Master's degrees related to the Semantic Web, ontology engineering, or electronics. Among them, five are from our laboratory, and the others are outside members. Each expert graded each search result using scores from 0 to 1. The scoring of relevance is regulated as follows: 0 for irrelevant result, 0.3 for a slightly relevant one, 0.6 for a fairly relevant one, and 1 for a highly relevant one. We averaged their scores and evaluated the effectiveness of each methodology using scored precision. The scored precision for the result list υ_i is defined as follows:

$$sp(\upsilon_i) = \frac{\sum_{s=1}^{10}\sum_{j=1}^{k} score(\upsilon_i, s, j)}{10 \times k},$$

where $score(\upsilon_i, s, j)$ denotes the score that the sth expert marked for the jth entry of the list υ_i in which only the top k ($k = 10$) entries are selected. For each result list υ_i, we calculated $sp(\upsilon_i)$ which denotes the average score that ten experts marked for the top k entries of υ_i. The evaluation results that we obtained are shown in Fig. 5.

Fig. 5 shows evaluation results of each semantic association search system. In the majority of cases, our approach based on generality produces better results than others. Therefore, two approaches to find general concepts, RSS and generality based search, show better evaluation results than others to find specific concepts. It mean that people generally focus on finding well-know and popular information that have many connections when they search something. For SemRank, RSS, and our systems, the average scored precisions over all five test queries are 0.61, 0.65, 0.60, and 0.67, respectively. Thus, the results show that our search model based on generality outperforms the SemRank and RSS model.

From the experimental results for evaluation, we can conclude that our semantic association search system is effective and can find more favorable relevant resources from the query than other methodologies. In this chapter, we showed experiment results for verification and evaluation. Consequently, from our semantic association search system, users can provide semantically relevant information with the query according to their *interestingness*: *specificity* and *generality*.

Table 4
Top five search results using edge and node weights based on uniqueness and the distance constraint.

Ranking	Results	Activation value
1	Silicon Valley Colocation	0.100
2	Metaweb Technology	0.034
3	Joseph Gleberman	0.031
4	Kevin Harvey	0.007
5	Robert Cook	0.003

Table 5
Top five search results using edge and node weights based on generality and the distance constraint.

Ranking	Results	Activation value
1	Metaweb Technology	0.289
2	Silicon Valley Colocation	0.156
3	John Giannandrea	0.055
4	Danny Hills	0.031
5	Robert Cook	0.016

7. Conclusions

The Semantic Web expresses knowledge in terms of concepts, properties, and instances, so that Semantic Web knowledge can be represented as nodes and relationships between nodes. Accordingly, the search method on the Semantic Web has to support the utilization of interrelationships among data, which are noted as resources.

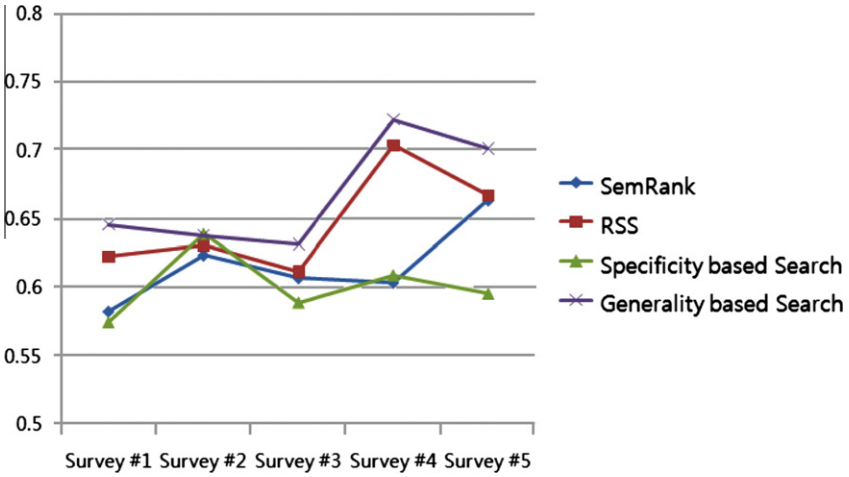


Fig. 5. Comparison of the scored precision.

In this paper, we proposed a semantic search methodology that consists of the evaluation of the amount of information and the spreading of the information retrieval in an ontological semantic network of resources and properties. To achieve the goals, we proposed a method for measuring the self-information of a semantic association that consists of resources and properties based on a measure of its uniqueness relative to other resources and properties in the semantic network. We also proposed another weighting method for identifying popular concepts based on generality. According to the experimental results, the effectivenesses of these systems were shown. In addition, we found that semantic paths are more valuable and important in a semantic network using the semantic search process, which extends the spreading activation algorithm. Spreading activation can propose a set of concepts which seem to be strongly connected to a given concept even though no explicit relationships between the concepts exist in the knowledge base. The search results can be greatly expanded with concepts which are most semantically related to the query through our semantic search method. From this work, we provide search results that are connected and ordered relations between search keyword and other resources as a link in a semantic network. The experimental results show that our method is feasible and leads to favorable semantic search results. We believe that the proposed semantic search is useful in all application domains and will make it possible to link concepts with semantically meaningful ontology instances that are present in the knowledge base. We believe that our research has shown a practical implementation of a semantic search on the Semantic Web.

For our future works, we are planning to refine our metrics for measuring the semantic relation in the semantic graph and to apply our methodology to social networks. In addition, we will try to develop methods for initial activation values considering node weight and relevance with query keywords. Finally, we will build a semantic search web site to evaluate our method using feedback.

Acknowledgements

This research was financially supported by the Ministry of Knowledge Economy (MKE) and the Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Strategic Technology and also supported in part by the Yonsei University Research Fund of 2011.

References

Anyanwu, K., Maduko, A., & Sheth, A. (2005). SemRank: Ranking complex relationship search results on the semantic web. In *Proceedings of the 14th international conference on World Wide Web* (pp. 117–127).

Bob, R., Hay, J., & Jannedy, S. (2003). *Probabilistic linguistics*. The MIT Press.

Cohen, P. R., & Kjeldsen, R. (1987). Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management, 23*(4), 255–268.

Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review, 82*(6), 407–428.

Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review, 11*(6), 453–482.

Dodge, Y. (2006). *The oxford dictionary of statistical terms*. Oxford University Press.

Electronic Times Internet. URL: <http://www.etnews.co.kr/>.

Gershenfeld, N. (1998). *The nature of mathematical modeling*. Cambridge University Press.

Glover, E. J., Lawrence, S., Gordon, M. D., Birmingham, W. P., & Giles, C. L. (2001). Web search–your way. *Communications of the ACM, 44*(12), 97–102.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition, 5*(2), 199–220.

Guha, R., McCool, R., & Miller, E. (2003). Semantic search. In *Proceedings of the 12th international conference on World Wide Web* (pp. 700–709).

Heflin, J., & Hendler, J. (2000). Searching the Web with SHOE. In *AAAI workshop on artificial intelligence for web search* (pp. 35–40).

Jones, K. S. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation, 28*, 11–21.

Korea Intellectual Property Rights Information Service (KIPRIS). URL: <http://eng.kipris.or.kr/>.

Lee, T. B., & Cailliau, R. (1990). *WorldWideWeb: Proposal for a HyperText Project*. URL: <http://www.w3.org/Proposal.html>.

Lee, T. B., Hendler, J., & Lassila, O. 2001. The Semantic Web. Scientific American Magazine.

Mangold, C. (2007). A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies, 2*(1), 23–34.

Ning, X., Jin, H., & Wu, H. (2008). RSS: A framework enabling ranked search on the semantic web. *Information Processing & Management, 44*(2), 893–909.

Park, J., Lee, M., & Hong, J. S. (2007). A study on development of RDF triple storage system for retrieval of metadata in the semantic web. *Society for e-Business Studies, 12*(2), 291–304.

Preece, S. E. (1981). *A spreading activation network model for information retrieval*. PhD dissertation, University of Illinois, Urbana-Champaign.

Schreiber, G., Amin, A., Aroyo, L., Assem, M., Boer, V., Hardman, L., et al. (2008). Semantic annotation and search of cultural-heritage collections: The MultimediaN E-Culture demonstrator. *Web Semantics: Science Services and Agents on the World Wide Web, 6*(4), 243–249.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*, 379–423.

Singhal, A. (2001). Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 24*(4), 35–42.

Smart Academic Knowledge Base DBpia. URL: <http://www.dbpia.co.kr/>.

Smith, M. (1993). *Neural networks for statistical modeling*. International Thomson Computer Press.

Sowa, J. F. (1992). *Semantic networks. In encyclopedia of artificial intelligence*. John Wiley & Sons.

Spanier, J., & Oldham, K. B. (1987). *The hyperbolic tangent and cotangent functions. In an atlas of functions*. Springer, pp. 279–284.

Weisstein, E. Logistic Equation. URL: <http://mathworld.wolfram.com/LogisticEquation.html>.

World Wide Web Consortium (1992). Tags used in HTML. URL: <http://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>.