Case and Relation (CARE) based Page Rank Algorithm for Semantic Web Search Engines

Ms.N.Preethi[1] , and Dr.T.Devi[2],

[1]Assistant Professor in Department of Computer Applications, Dr. N.G.P. Institute of Technology, Coimbatore. Tamil Nadu, India

[2]Reader and Head, Department of computer Applications, Bharathiar University, Coimbatore. Tamil Nadu, India .

Abstract

Web information retrieval deals with a technique of finding relevant web pages for any given query from a collection of documents. Search engines have become the most helpful tool for obtaining useful information from the Internet. The next-generation Web architecture, represented by the Semantic Web, provides the layered architecture possibly allowing data to be reused across application. The proposed architecture use a hybrid methodology named Case and Relation (CARE) based Page Rank algorithm which uses past problem solving experience maintained in the case base to form a best matching relations and then use them for generating graphs and spanning forests to assign a relevant score to the pages.

Keywords: Semantic Web, Page Rank Algorithm, CARE Page Rank Algorithm

1. Introduction

The World Wide Web is a dynamic architecture, which support many people to exchange their information. Search engines have become the most helpful tool for obtaining useful information from the internet. However, the search results returned by even the most popular search engines are not satisfactory. It is not uncommon that search engines return a lot of Web page links that have nothing to do with the user's need. It surprises users because they do input the right keywords and search engines do return pages involving these keywords, and, yet, the majority of the results are useless. In order to enhance the existing search performance semantic web architecture is proposed by W3C.

With a purpose to divulge where the problem stays, the most prominent search engine google has been tendered with a key in having the following keywords of word order "airports in Tamilnadu". From the upshots we find only two hyperlink in a straight match to the parent folio of 'airports in Tamilnadu' produced by goggle even at the very first resultant page. We find a link showing the list of airports throughout India which certainly includes Tamilnadu, furthermore there is an another link showing an article published in The Hindu newspaper on the subject matter correlated with the fed-stuff. Since the page contains the keyword 'airport' the search engine yielded the resulted page showing this unrelated link amidst the related one. When this page is subjected for a meticulous analysis in regard with why the keyword 'airport' appears there. It is just because the link showing the article dated june, 2011 has been made mention of the keyword 'airport' while on their reference to the MBA course in aviation and airport management. We input the keywords airport and Tamilnadu in the search engine with an intention to track down the list of airports located in Tamilnadu. Unanimously with unconcealed thoughts we think there exist some relation between the keywords we

submitted and desire for the assurance of relation in the resultant pages of the search engine. The submission is vivid; airports in Tamilnadu. Nevertheless the correlation between the keywords are expunged instantly after submission to the search engine because under the system architecture of the current web. We certainly not arrive at the possibilities to record the relations between the entities. Thus we are not shown up by the search engines with the data we need.

Relation lost – this has been a means for the entire problem! In some way or other everything is found everywhere in chains. For e.g. With regard to airports and Tamilnadu one of the relations between them is 'located in'. Together the relations from the semantics of 'airport' in this context; airport located in Tamilnadu. Factually speaking the semantics of an entity is always found in connection between machines; the bond of affinity that exist between the entities in prior to the comprehension of the semantics of each other by the machines.

The next generation Web [2], [5], Semantic Web, offers a solution to this problem in the system architecture level. In fact, in the Semantic Web, each page possesses semantic metadata that record additional details concerning the Web page itself. Annotations are based on classes of concepts and relations among them. The "vocabulary" for the annotation is usually expressed by means of an ontology that provides a common understanding of terms within a given domain.

2. Research Background

The Semantic Web is a collaborative movement led by the World Wide Web Consortium (W3C) [1] that promotes common formats for data on the World Wide Web. By encouraging the inclusion o f semantic content in web pages, the Semantic Web aims at converting the current web of unstructured documents into a "web of data". It builds on the W3C's Resource Description Framework (RDF).[4] According to the W3C, The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries.[4] The term was coined by Tim Berners-Lee,[3] the inventor of the World Wide Web and director of the W3C, which oversees the development of proposed Semantic Web standards. He defines the Semantic Web as a web of data that can be processed directly and indirectly by machines.

As with the WWW, the growth of the Semantic Web will be driven by applications that use it. Semantic search is an application of the Semantic Web to search. Search is both one of the most popular applications on the Web and an application with significant room for improvement. We believe that the addition of explicit semantics can improve search. Semantic Search attempts to augment and improve traditional search results (based on Information Retrieval technology) by using data from the Semantic Web. Traditional Information Retrieval (IR) technology is based almost purely on the occurrence of words in documents. Search engines like Google [4], augment this in the context of the Web with information about the hyperlink structure of the Web. The availability of large amounts of structured, machine understandable information about a wide range of objects on the Semantic Web offers some opportunities for improving on traditional search. Before getting into the details of how the Semantic Web can contribute to search, we need to distinguish between two very different kinds of searches.

Navigational Searches:

- In this class of searches, the user provides the search engine a phrase or combination of words which he/she expects to find in the documents. There is no straightforward, reasonable interpretation of these words as denoting a concept. In such cases, the user is using the search engine as a navigation tool to navigate to a particular intended document.

Research Searches:

- In many other cases, the user provides the search engine with a phrase which is intended to denote an object about which the user is trying to gather/research information. There is no particular document which the user knows about that he/she is trying to get to. Rather, the user is trying to locate a number of documents which together will give him/her the information he/she is trying to find.

Semantic search attempts to improve the results of research searches in 2 ways.

- Traditional search results take the form of a list of documents/Web pages. We augment this list of documents with relevant data pulled out from Semantic Web. The Semantic Web based results are independent of and augment the results obtained via traditional IR techniques.
- The search phrase in Research Searches typically denotes one (or occasionally two) real-world concepts. We believe that it might be useful for the text retrieval part of the search engine to have an understanding of these concepts denoted by the search phrase. Understanding the denotation can help understand the context of the search, the activity the user is trying to perform, drive expectations on the categories of documents (pertaining to the object) likely to exist, etc.

Case-based reasoning (CBR), broadly construed, is the process of solving new problems based on the solutions of similar past problems. An auto mechanic who fixes an engine by recalling another car that exhibited similar symptoms is using case-based reasoning.

It has been argued that case-based reasoning is not only a powerful method for computer reasoning, but also a pervasive behavior in everyday human problem solving; or, more radically, that all reasoning is based on past cases personally experienced. This view is related to prototype theory, which is most deeply explored in cognitive science.

Case-based reasoning has been formalized for purposes of computer reasoning as a four-step process[7]:

Retrieve: Given a target problem, retrieve from memory cases relevant to solving it. A case consists of a problem, its solution, and, typically, annotations about how the solution was derived. For example, suppose Fred wants to prepare blueberry pancakes. Being a novice cook, the most relevant experience he can recall is one in which he successfully made plain pancakes. The procedure he followed for making the plain pancakes, together with justifications for decisions made along the way, constitutes Fred's retrieved case.

Reuse: Map the solution from the previous case to the target problem. This may involve adapting the solution as needed to fit the new situation. In the pancake example, Fred must adapt his retrieved solution to include the addition of blueberries.

Revise: Having mapped the previous solution to the target situation, test the new solution in the real world (or a simulation) and, if necessary, revise. Suppose Fred adapted his pancake solution by adding blueberries to the batter. After mixing, he discovers that the batter has turned blue – an undesired effect. This suggests the following revision: delay the addition of blueberries until after the batter has been ladled into the pan.

Retain: After the solution has been successfully adapted to the target problem, store the resulting experience as a new case in memory. Fred, accordingly, records his new-found procedure for making blueberry pancakes, thereby enriching his set of stored experiences, and better preparing him for future pancake-making demands.

3. Related Works

To make use of relations in the semantic web authors measure the distance between the systematic description of both query and retrieval resources. First, explode initial set of relation by adding hidden relation taken from the query. Similarly compute ratio between the relation instance linking concepts specified in the user query and actual relation instance in the semantic knowledge. Today's search engine is targeted to the web rather than the semantic web. A similar approach has been integrated into AI methodologies to address the problem of query answering. Query logs are also used to construct a user profile to be later used to improve the accuracy of web search.

Semrank is one of the existing methods for ranking , which gives the basic idea of ranking and also provides the maximum information in the result to achieve the goal, K. Anyanwu et al. defines two measures named "uniqueness" and "discrepancy"(Anyanwu,K., Maduko, and A., Sheth, 2005). An additional added value of SemRank is that of the computation of the ranking, which exploits a so-called "modulative relevance model" that is capable of taking into account the particular context/purpose in/for which a query has been submitted (conventional or discovery search).

A totally different solution is represented by OntoLook (Li,Y., Wang ,Y., Huang ,X., 2007). The basic idea is that if a graph-based representation of a Web page annotation can be provided, where concepts and relations (together with their multiplicities) are modelled as vertices and weighted edges, respectively, it becomes possible to define a series of cuts removing less relevant concepts from the graph. This allows for the generation of a so-called candidate relation-keyword set (CRKS) to be submitted to the annotated database, which can significantly reduce the presence of uninteresting pages in the result set. It is worth observing that the strategy behind OntoLook only allows us to empirically identify relations among concepts that should be less relevant with respect to the user query.

In fact, a ranking strategy like the PageRank , used by Google is only one of the ranking algorithms used to organize results to be displayed to the user (Junghoo,C., Garcia-Molina,H., Page,L.1998) ,(Page,.L, Brin,S., Motwani,R., Winograd,T.,1998) and (Brin.S, & Page,L,1998). Many other statistical and text-matching techniques are used together with PageRank. Of course, PageRank can be used in conjunction with (Li,Y., Wang ,Y., Huang ,X., 2007) to exploit relevance feedback and post process the result set. But the use of the remaining techniques is not feasible since they cannot be reasonably applied into a concept-relation-based framework where ontology is predominant on pure text.

Finally, the proposed technique is not intended to replace the ranking strategies of actual

search engines. In fact, it relies on relevance information that is totally different from that exploited, for example, in algorithms like SemRank, Pagerank, and others. Rather, it should be understood as a pre-processing step to produce a semantic- aware ordered result set to be later (or simultaneously) treated with existing (popular) techniques in order to come to an increased hit ratio in user query processing.

4. CARE Algorithm

In this paper we propose a new methodology which uses Textual Case Based Reasoning and Relation-based Page Ranking. Let us assume that a user enters the query "apple" and "banana". We find that both of them are sub-concepts of fruits. Textual Case Based Reasoning used to find it by using the previous knowledge of solving similar problem.

The semantic knowledge is encapsulated in a taxonomy, $T_i = U< h_{\_}, h^{-} >$, where $< h_{\_}, h^{-} >$ is a hypernym-hyponym relationship pair. A hypernym, h^{-}, is a term whose semantic range includes the semantic range of another word called hyponym, $h_{\_}$. In our example fruit is the hypernym of apple, whilst apple is the hyponym in this relationship. T is recursively extracted from the Web.

Hyper-hyponym relationships are ideal for taxonomy creation because they capture the is-a relation that is typically used when building ontologies. The basic Hearst extraction patterns summarize the most common expressions in English for $< h_{\_}, h^{-} >$ relationship discovery. Expressions like "X such as Y" (also "X including Y", and "X especially Y") are used to extract the relationship "Y is a hyponym of X". For example, given the term "food" a search for "food such as" in the text "food such as grapes and cereal" will discover hyponyms "grapes" and "cereal". In reality the set of candidate hyponyms needs filtered so that irrelevant relationships are removed. Therefore taxonomy generation can be viewed as a 2-staged search-prune process which when repeated on newly discovered terms generates the taxonomy in a top-down manner.

TSI presented in the previous section, calls for a bottom-up taxonomy discovery approach, because the BOC representation is based on finding h^{-} from $h_{\_}$ in V (and not $h_{\_}$ from h^{-}). Therefore we need to start with leaf nodes corresponding to terms in BOW (Ci) and progressively extract higher-level concepts from the Web. Hearst's patterns can still be used albeit in an inverse manner. For example to extract h^{-} for term "fish" we can use the pattern "X such as fish", where X is our h^{-}. We have also had to refine these inverse patterns in order to remove false positives that are common due to problems with compound nouns and other similar grammatical structures.

Knowledge extracted from the Web may contain relationships that are contextually ir-relevant to the TCBR system. Verification patterns with a conjunction is commonly used for this purpose: "h^{-}such as $h_{\_}$ and *"; checks if an extracted hypernym (h^{-}) is also a common parent to a known hyponym ($h_{\_}$) and other candidate hyponyms(). This can be done by using Pruning as disambiguation method. The resulting relationship is used in the relation-based technology. By using this result the number of relations generated is reduced and the best relation can be choosing by the user. Then from this result we can use the Graph based notation and generate Spanning forest and by using its probability value to rank the pages.

A graph based representation can be designed based on ontology for a domain. In ontology graph G, where OWL classes are mapped into graph vertices and OWL relation properties are mapped into graph edges. Thus, the existing

relations between pair of concepts in the domain are depicted by means of connected vertices in the graph.

According to graph theory, the undirected graph G can be defined as G(C,R) where C={c1,c2……..cn} is the set of concepts that can be identified in the ontology, |C| = n is the total number of concepts available.

$R=\{R_{ij} \mid i=1,2,\ldots..n,\ j=1,2,\ldots n,\ j>i\}$ is the set of edges in the graph.

Example of ontology graph is illustrated in Figure 2(a).Since queries are specified by the user by providing a collections of keywords and associated concepts, a single query can be formally expressed as $Q=\{(k_t,c_t)\}$. Given a particular query containing a specific set of keywords related to a subset of ontology concepts, it is possible to construct a query sub graph G_Q.

The aim of this paper is to demonstrate that, with the ontology graph G and a query sub graph G_Q, it is possible to define a ranking strategy capable of assigning each page including queried concepts a relevance score based on the semantic relations available among concepts within the page itself (thus neglecting the contribution of the remaining WebPages). The proposed ranking strategy assumes that given a query Q, for each page p, it is possible to build a page sub graph $G_{Q,p}$ using a methodology that is similar to the one used for G and G_Q and exploiting the information available in page annotation A. By expressing page annotation A as a graph, we have A=(AC,AR), where AC and AR are the sets of annotated concepts and relations, respectively.

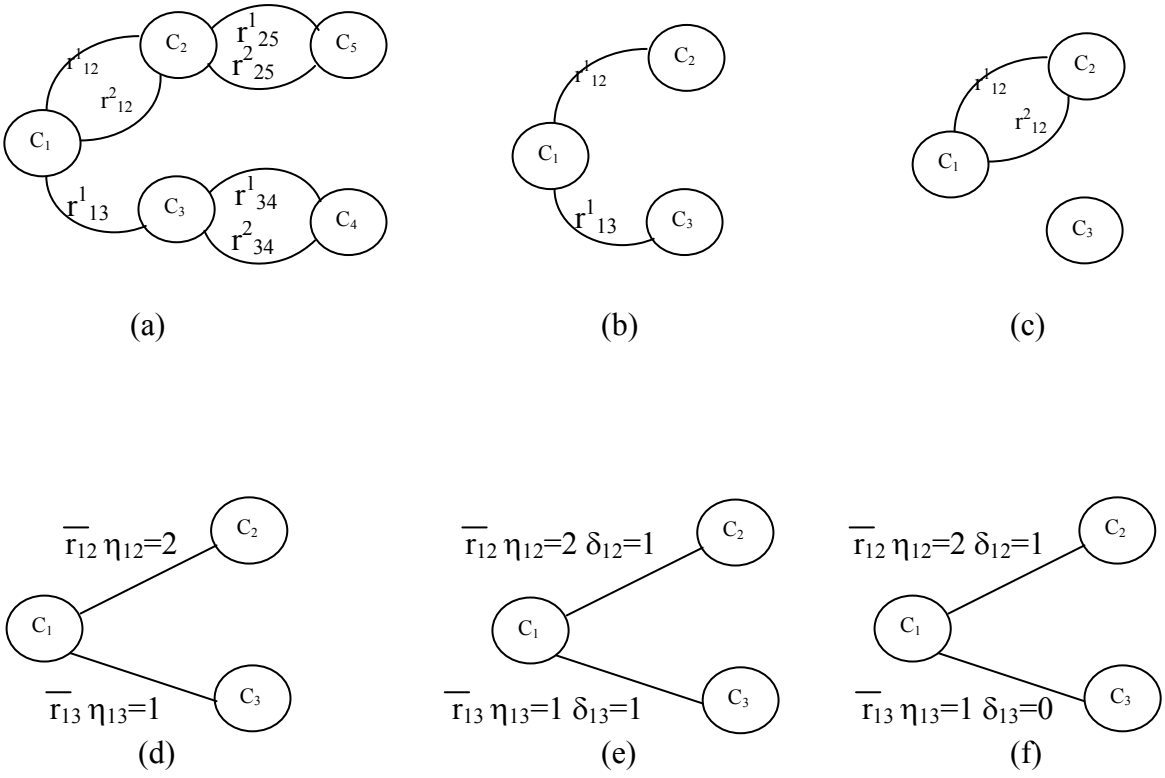

Fig .1 (a) Is an ontology graph. (b) Query subgraph obtained for a given query specifying concepts C_1,C_2 and C_3. (c) and (d) A first example of page annotation p1 and the related page subgraph. (e) and (f) A second example of page annotation p2 and the related page subgraph.

From the above premise, it is to be considered as the computation of a page relevance score. An analysis (now from a formal point of view) of the steps followed by a user during the process of query definition becomes relevant. In this case, the user specifies a query composed by conceptt c1, c2, and c3 over a novel ontology. Based on the considerations above, a measure of page relevance can be computed by estimating, for each concept, the probability of having a relation between that concept and another concept and that such relation is exactly the one in the user's

However, it can be demonstrated that this probability can be expressed also in different terms, capable of taking into account situations in which a particular concept can be related to more than one concept. Specifically, the probability that each concept is related to other concepts is given by the probability of having c1 linked to c2 and c2 linked to c3 or c1 linked to c2 and c1 linked to c3 or c2 linked to c3 and c1 linked to c3. The situations above can be modeled by using graph theory. In fact, having each concept related to at least another concept in the query is equivalent to considering all the possible spanning forests (a collection of spanning trees, one for each connected component in the graph) for page subgraph $G_{Q,p}$ given the query Q. In Figures. 2(e), 2(f) and 2(g), all the possible spanning.forests of the page subgraph in Fig. 2(d) are shown. We call $SF^f_{Q,p}$ the f^{th} page spanning forest computed over $G_{Q,p}$. We define $P(SF^f_{Q,p})$ as the probability that $SF^f_{Q,p}$ is the spanning forest of interest to the user. By simplifying the notation and replacing $r_{ij,}$ p with $r^p_{ij,}$ the probability for page p can be computed as

$$P(Q,p)=P(((r^p_{12}\cap r^p_{23})\cap SF^1_{Q,p})\cup((r^p_{12}\cap r^p_{13})\cap SF^2_{Q,p})\cup((r^p_{23}\cap r^p_{13})\cap SF^3_{Q,p})) \quad (1)$$

Since the events are not correlated, it is also

$$P(Q,p) = P(r^p_{12}\cap r^p_{23}).P(SF^1_{Q,p}) + P(r^p_{12}\cap r^p_{13}).P(SF^2_{Q,p}) + P(r^p_{23}\cap r^p_{13}).P(SF^3_{Q,p}).$$

$$= P(r^p_{12}).P(r^p_{23}).P(SF^1_{Q,p}) + P(r^p_{12}).P(r_{13}{}^p).P(SF^2_{Q,p}) + P(r^p_{23}).P(r^p_{13}).P(SF^3_{Q,p}). \quad (2)$$

where $P(r_{ij}, p)$ can be replaced with

$$\mathbf{T_{ij}=\delta_{ij}/\eta_{ij}.}$$

Since the probability for a single page spanning forest to be the one of interest to the user is the same with respect to the remaining ones, if we define $\sigma_{Q,p}$ as the number of spanning forests for $G_{Q,p}$, we have

$$P(Q,p) = \frac{P(r^p_{12}).P(r^p_{23}) + P(r^p_{12}).P(r^p_{13}) + P(r^p_{23}).P(r^p_{13})}{\sigma_{Q,p}} \quad (3)$$

and according to the definition of relation probability, it is

$$P(Q,p) = \frac{[T_{12}.T_{23}+T_{12}.T_{13}+T_{23}.T_{13}]}{\sigma_{Q,p}} \quad (4)$$

Based on the number of constrained page spanning forests that can be generated from the page subgraph for a given number of edges, the probability of that page can be calculated as the sum of the probabilities computed for each constrained page spanning forest of a given length divided by the total number of constrained page spanning forests of that length that can be originated by the page subgraph.

Algorithm : Case and Relation (CARE) based Page Rank Algorithm

Input : User Search Query (Q)
Output : Set of Web pages satisfy User Query
Procedure : CARE Algorithm

CARE (Q)

Begin

ho ← set of hyponym (user query)
he ← set of hypernym (stored in knowledge base)
hse ← set of hyper-hyponym relation
hspe ← set of pruned hyper-hyponym relation
G(C,R) ← Ontology graph
where C – Concepts (nodes) in the Ontology graph and
R – Edges (Relations) between the nodes in the Ontology graph
$G_q (C_q , R_q)$ *← Query subgraph*
where C_q *– Concepts (nodes) in the Query subgraph and*
R_q *– Edges (Relations) between the nodes in the Query subgraph*
$G_a (C_a , R_a)$ *← Annotated graph*
where C_a *– Concepts (nodes) in the annotated page graph and*
R_a *– Edges (Relations) between the nodes in the annotated page graph.*
$G_p(C_p,R_p)$ *←Page subgraph*
where C_p *– Concepts (nodes) in the Page subgraph and*
R_p *– Edges (Relations) between the nodes in the Page subgraph.*
Foreach Page subgraph do
Begin
Label the edged in G_p *with an index ranging from 1 to* R_p
Define variable e and a to index graph edges
Set $\eta_e = \eta_{ij}$
Set $\delta_e = \delta_{ij}$
Set $\tau_e = \delta_{ij} / \eta_{ij}$
Mark all the edges in G_p *as not visited*
Allocate weight vector W of size $|C_p|$*-1*
Allocate vector Σ *of size* $|C_p|$*-1*
Initialize W and Σ *to zero*
for e=1, $e \leq |R_p|$*, e=e+1*
Begin
mark edge e as visited
visit (e,e,1, τ_e*)*
$W[1] = W[1] + \tau_e$
$\Sigma[1] = \Sigma[1] + 1$
End
End
End

Visit (o,e,l,s)
Begin
    a = e + 1
    while a≤|R_p| and l≤|C_p|-1
    Begin
        If a is not visited and a is safe then
        Begin
            mark edge a as visited
            visit (o,a,l+1,s×τ)
            W[l+1] = W[l+1] + s
            Σ[l+1] = Σ[l+1] + 1
            set edge a as not visited
        End
        Else
            a = a+1;
    End
End

5. Conclusion

The next-generation web architecture represented by the semantic web will provide adequate instruments for improving search strategies thereby enhancing the probability of seeing the user query satisfied without requiring tiresome manual refinement. However, actual methods for ranking the returned result set will have to be adjusted to fully exploit additional contents characterized by semantic annotations including ontology-based concepts and relations. Several ranking algorithms for the Semantic web exploiting relation-based metadata have been proposed. In this work, a novel ranking strategy has been proposed that is capable of providing a relevant score for a web page into an annotated result set by simply considering the user query, the page annotation, and the underlying ontology. Experimental analysis shows optimistic results in terms of both time complexity and accuracy.

References

[1] "XML and Semantic Web W3C Standards Timeline", 2012-02-04

[2] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web", Scientific Am., vol. 284, no. 5, pp. 34-43, 2001.

[3] Berners-Lee, Tim; James Hendler and Ora Lassila (May 17, 2001), "The Semantic Web", *Scientific American Magazine*, Retrieved March 26, 2008.

[4] "W3C Semantic Web Activity", World Wide Web Consortium (W3C), November 7, 2011, Retrieved November 26, 2011.

[5] A. Gomez-Perez and O. Corcho, "Ontology Languages for the Semantic Web", IEEE Intelligent Systems, vol. 17, no. 1, pp. 54-60, Jan.-Feb. 2002.

[6] The Google.com search engine, http://www.google.com/, 2004.

[7] Agnar Aamodt and Enric Plaza, "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches", *Artificial Intelligence Communications,* 7 (1994): 1, 39-52.