Search Engine Optimization through Spanning Forest Generation Algorithm

SANTOSH KUMAR GANTA, SATYA PAVAN KUMAR SOMAYAJULA
Department of CSE, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam, (A.P.), India
Asst Prof., Department of CSE, Avanthi College of Engg & Tech, Tamaram, Visakhapatnam, (A.P.), India
Email: santosh.ganta@gmail.com. & balasriram1982@gmail.com

Abstract: **Search engine technology has had to scale dramatically to keep up with the growth of the web. With the tremendous growth of information available to end users through the Web, search engines come to play ever a more critical role. Determining the user intent of Web searches is a difficult problem due to the sparse data available concerning the searcher. We qualitatively analyze samples of queries from seven transaction logs from three different Web search engines containing more than five million queries. The following are our research objectives: Isolate characteristics of informational, navigational, and transactional for Web searching queries by identifying characteristics of each query type that will lead to real world classification. Validate the taxonomy by automatically classifying a large set of queries from a Web search engine. This paper we deal with now is semantic web search engines is the layered architecture and we use this with relation based page rank algorithm.**

KEYWORDS: Web, search engines, ontology, web crawling, page rank algorithm.

1. INTRODUCTION

The search engine has no infrastructure or matching techniques to give correct or a related information for the query raised. Now the semantic web solves this problem. Each page contains meta data with notes, meanings, list of words, definitions, vocabulary for the annotations etc. annotations are based on the classes of concepts and relations among them. For an eg. a query is entered as hotels-hill station, Ooty. The result of the search engine shows several hotels in and around Ooty. The final results are of nothing to do with the selected city. Only two out of seven results satisfy user needs. The list is so ranked that the end user is not furnished with the information that satisfies his or her intention. Of the ten or twelve pages displayed only the first two may be of importance and the other retrieved information must be discarded ones. The user may go through the other pages only if he is interested. But the query he or she has raised would not get the proper result In this paper, we will prove that relations among concepts embedded into semantic annotations can be effectively exploited to define a ranking strategy for Semantic Web search engines. This sort of ranking behaves at an inner level (that is, it exploits more precise information that can be made available within a Web page) and can be used in conjunction with other established ranking strategies to further improve the accuracy of query results. With respect to other ranking strategies for the Semantic Web, our approach only relies on the knowledge of the user query, the Web pages to be ranked, and the underlying ontology. Thus, it allows us to effectively manage the search space and to reduce the complexity associated with the ranking task. We provide an overview of existing strategies for Semantic Web search, the basic idea behind the proposed approach is presented by resorting to practical examples, formal methodology for deriving the general rule is illustrated and concerning the implementation is provided. An analysis of the algorithm complexity is given, and Experimental results are discussed.

2. RELATED WORK AND MOTIVATION

Nevertheless, because of their general-purpose approach, it is always less uncommon that obtained result sets provide a burden of useless pages. It is not uncommon that even the most renowned search engines return result sets including many pages that are definitely useless for the user this is mainly due to the fact that the very basic relevance criterions underlying their information retrieval strategies rely on the presence of query keywords within the returned pages. The idea of exploiting ontology-based annotations for information retrieval is not new [7], [8], [11].Nevertheless, these first works did not focus on semantic relations, which are considered (and expected) to play a key role in the Semantic Web [11].

2.1 DISADVANTAGE

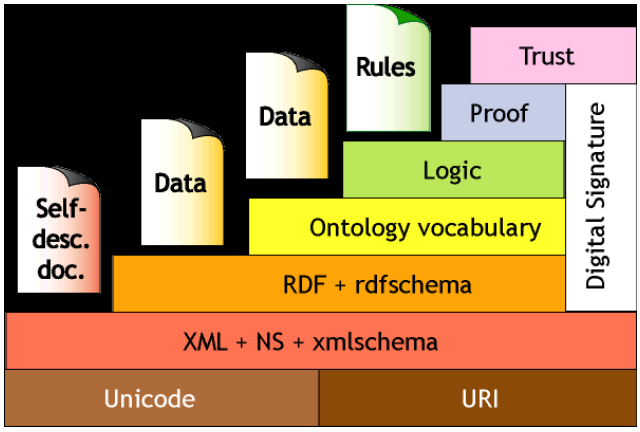
(i) Text based searching example (Google, yahoo, msn, Wikipedia).

(ii) Without semantic relationship to give exact result.
(iii) Query only focus single search engine.
(iv) Most existing search engines however, provide poor support to accessing the web results.
(v) No analysis of stopping keywords from the user
Query.
(vi) It will not give relevant or exact result.

The idea here is to make use of relations in semantic web page annotation to get an ordered result, where the pages best fit to the user query are displayed first. These first works do not focus on semantic relations, which is the key role in the semantic web [10]. Only recently an ordered way to achieve relation based ranking for semantic contents is found [2]. First attempts to enhance semantic web search engine with ranking capability is reported then the hidden relations are added. Similarity is computed between the relation instances and actual multiplicities of relation instances. The number of relations will largely exceed the number of concepts [1]. Ontology based lexical relation such as synonyms, antonyms and homonyms between key words have been used to expand query result.

3. PROTOTYPE OF A RELATION-BASED SEARCH ENGINE

To evaluate the feasibility of the proposed approach, we first constructed a controlled Semantic Web environment. To do this, we selected the well-known travel.owl ontology written in the OWL language, and we modified it by adding new relations in order to make it more suitable for demonstrating system functionality. We then created a knowledge base by either downloading or automatically generating a set of web pages in the field of tourism, and we embedded into them RDF semantic annotations based on the ontology above. Finally, we designed the remaining modules of the architecture, including a Webpage database, a crawler application, a knowledge database, an OWL parser (OwlDotNetApi), a query interface, and the true search engine module embedding the proposed ranking logic (Fig. 1). The crawler application collects annotated Web pages from the Semantic Web (in this case, represented by the controlled environment and its Web page collection) including RDF metadata and originating OWL ontology. RDF metadata are interpreted by the OWL parser and stored in the knowledge database. A graphics user interface allows for the definition of a query, which is passed on to the relation-based search logic. The ordered result set generated by this latter module is finally presented to the user. The details of the system workflow will be provided in the following sections, starting with the query definition process, since it was through the analysis of its dynamics that we came to the identification of our ranking strategy. The aim here to construct a controlled semantic web environment. We select Owl ontology [20] written in owl language and modify it by adding new relations. Then a knowledge base is created either by downloading or automatically generating a set of web pages in the field of tourism and we embedded in to them RDF semantic annotation based on the ontology.



3.1 WEB CRAWLER

Web crawlers are a central part of search engines, and details on their algorithms and architecture are kept as business secrets. When crawler designs are published, there is often an important lack of detail that prevents others from reproducing the work. There are also emerging concerns about search engine spamming, which prevent

major search engines from publishing their ranking algorithms.

3.2 SEMANTIC WEB ARCHITECTURE:

The crawler application collects annotated web pages from semantic web.OWL parser interpret the meta data and stores it in knowledge database. The GUI allows for the definition of a query and pass it to relation based search logic. Finally the ordered result generated module is presented to the end user.

3.3 QUERY DEFINITION AND PROCESS:

In search engine like Google [6] a query is specified by giving a set of key words linked with logic operators and document type, language etc. But semantic search engines are capable of exploiting concepts hidden behind each keyword with natural language interpretation techniques to further refine the result set.

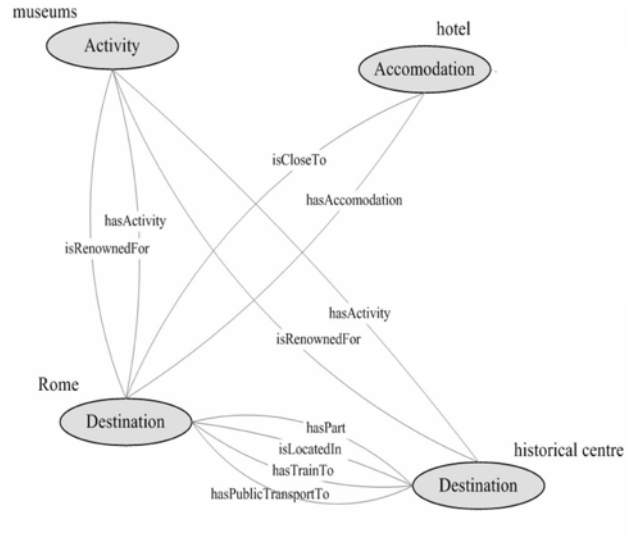
3.4 INTRODUCTION TO RELATION BASED RANKING:

Suppose the user specifies the keyword Ooty as a first keyword and then selects from the pull down menu one of the concepts such as destination or city, then he adds the second word 'hotel'. The traditional search engines, returns both pages without considering the information provided by the semantic mark. But the semantic search engine takes in to account keyword concept associations and would return a page only if both keywords are present in the same page and are related to associated concepts. It goes beyond pure "keyword isolation" Search.

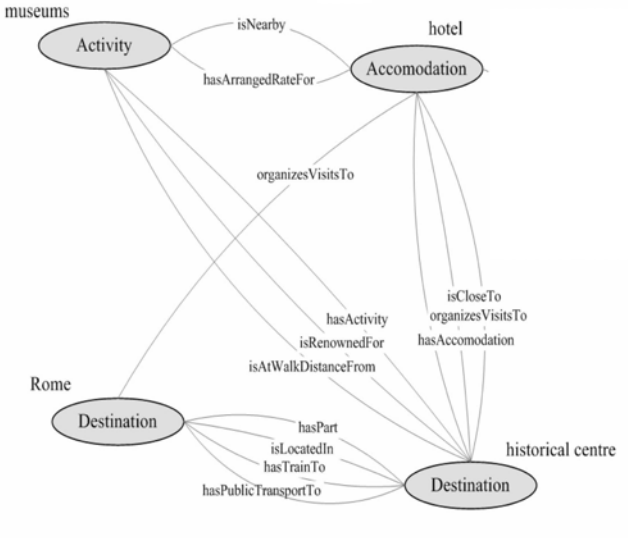
3.5 GRAPH BASED NOTATION AND METHODOLOGY:

In the ontology and annotation graphs, concepts and relations are translated in to graph nodes and edges. Two examples of annotated graphs built upon as many annotated web pages are shown in fig.2 Nodes/concepts linked only if there is at least one relation between those concepts in the ontology. If there is no relation with other concepts, the edge gets removed. In such a way, several relevance classes are defined, each characterized by a certain number of connected Concepts. Within each class pages are ordered, depending up on the probability measure.{A;B;C;D}. In this way, the size of the database can be reduced. The aim of this paper is to demonstrate that, given an ontology graph G and a query sub graph GQ, 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 Web pages). The proposed ranking strategy assumes that given a query, for each page p, it is possible to build a page sub graph Gqp using a methodology that is similar to the one used for G and GQ 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.



(a)



(b)

The methodology we propose in this paper starts from a age sub graph computed over an annotated page generates all the possible combinations of the edges belonging to the sub graph itself not including cycles. Since there could exist pages in which there are concepts that do not show any relations with other concepts but that could still be of interest to the user, the methodology progressively reduces the number of edges in the page sub graph and computes the probability that each of the resulting subgraphs obtained by a combination of the remaining edges is the one that matches the user's intention. Edge removal could lead to having concepts without any relation with other concepts. Thus, several relevance classes are defined, each characterized by a certain number of connected concepts. Within each class, pages are ordered depending on the probability measure above and presented to the user.

4. RELATION BASED RANKING FORMAL MODEL

In graph based formulation, OWL classes mapped in to vertices and OWL relation properties in to edges. Ontology graph is called G. According to graph theory the undirected graph G is defined as G(C, R), where C=set of concepts |c|=n is the total number of concepts available R=(Rij|i=1…n,j=1|….n,j>i) R is the set of edges in the graph and Rij={r1 ij,r2 ij,…,rm ij,m<n} is the set of edges between the concepts i and j. An example of ontology graph is fig.4. Given a particular query continuing specific Set of key words related to a

subset of ontology concepts it is possible to construct a query sub graph GQ. Given an ontology graph G and a query sub graph GQ, it is possible to define a ranking strategy.

4.1 RELEVANCE AND SEMANTIC RELATIONS:

Here we consider how to apply the methodology for the computation of a page relevance score. Let us imagine here are three keywords k1 k2 and k3 and associated concepts c1, c2 and c3 we now start from k1 c1 after k1 we insert k2 and c2 then finally k3 and c3. Suppose all the above facts are all exist in the pages P1 and p2. Now we want to rank these pages in order to present the user first the pages best fits his or her query. The semantic annotations and the page sub graphs of these pages are illustrated in figure. In figure 2c in the first page both c1 and c2 are linked with c1 through single relation. In the second page there exist two relations between c3 and c1. But there is no link between c2 and c1. Here it is difficult to compute the best fit, so probability calculation between c2 and c1 and c1 and c3 is calculated.

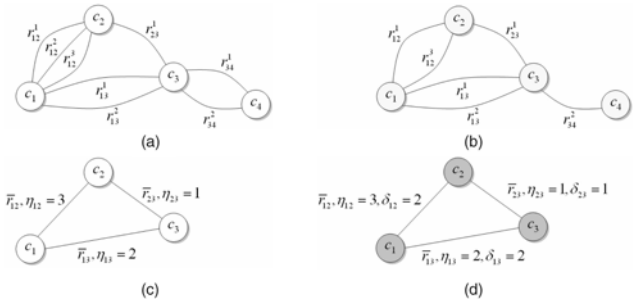


Fig. 5. (a) An ontology graph. (b) Query subgraph. (c) An example of an annotated page. (d) Page subgraph built upon the given ontology/query.

5. IMPLEMENTATION OF RANKING ALGORITHM

5.1 Overall Procedure

We now assemble the various steps illustrated in the previous sections to present the overall ranking methodology (whose workflow is depicted in Fig.). The user starts defining query keywords and concepts. Two strategies are followed for the calculation of relevance score. The search engine logic accesses the Web page database, constructs the initial result set including all those pages that contain queried keywords and concepts, and computes the query subgraph. Then, for each page in the result set, the page subgraph is computed. Starting from each subgraph, all page spanning forests (both constrained and unconstrained) are generated and used to compute the page score based on (6). Web pages are associated to relevance classes, and the final (ordered) result set is constructed.

5.2 Spanning Forest Generation Algorithm

According to (6), calculating the relevance score for a single page requires considering all the page forests and, for each forest, computing the constrained page relevance score. This requires finding an efficient way for both enumerating all the page forests for a given query and computing the page probability. The efficient algorithm for the identification of a minimum or maximum panning tree makes use of the so called disjoint set data structure. The algorithm that makes use of this set is called an *union find* algorithm. Essentially it incrementally builds sets of related objects. The implementation in C is contained in union_find.h andunion_find.c. A detailed analysis of its complexity is provided in Section 6. It is worth observing that the incremental approach adopted in this algorithm shows an additional benefit with respect to the decremental one. In fact, it becomes possible to impose an upper bound to the growth of page forests in terms of the number of edges. Since a larger number of edges means a higher accuracy in the estimation of page relevance accompanied by a larger computational cost, the possibility of introducing a threshold to the widest page forest to be considered could allow us to achieve a trade-off between ranking precision and complexity.

```
Label the edges in G_{Q,P} with an index ranging from 1 to R_{Q,P}
Define variables e and a to index graph edges
Set η_e = η_ij        //number of relations linking concepts
                      //i and j in the ontology graph (edge e∈R_{Q,P})
Set δ_e = δ_ij        //number of relations linking concepts
                      //i and j in the page subgraph (edge e∈R_{Q,P})
Set τ_e = η_ij / δ_ij //relation probability for edge e
Mark all the edges in G_{Q,P} as not visited
Allocate weight vector W of size |C_{Q,P}|-1
                      //W[l] stores the accumulated constrained
                      //probabilities for page forests of length l
Allocate vector Σ of size |C_{Q,P}|-1
                      //Σ[l] stores the number of page forests
                      //for a given length l
Initialize W and Σ to zero

for e=1, e≤|R_{Q,P}|, e=e+1
  mark edge e as visited
  visit(e,e,1,τ_e)
  W[1]=W[1]+τ_e
  Σ[1]=Σ[1]+1

function visit(o,e,l,s)
  a=e+1
  while a≤|R_{Q,P}| and l≤|C_{Q,P}|-1
    if a is not visited and a is safe
       //(does not introduce cycles, checked through DFS)
       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
    else
       a=a+1
```

6. EXPERIMENTAL RESULTS

In this section, the applicability of our technique into real scenarios will be analyzed by conducting two types of evaluations aimed at measuring the performance in terms of both time complexity and accuracy. The time complexity will be compared with that of [10], since our technique could be easily seen as an extension of it.

Nevertheless, since the methodology in [10] is not targeted at ranking the result set, the accuracy of results will be compared with that of a traditional search engine like Google.

6.1 Time Complexity

The computation of fair results concerning time complexity requires a sufficiently large repository with a significant number of annotated pages. Because of the difficulty of integrating the proposed technique within today's search engines like Google, in which a native semantic layer is actually missing, we chose to estimate the computation time over a synthetic Semantic Web environment. The positive effect of this choice is twofold. On one hand, it is possible to

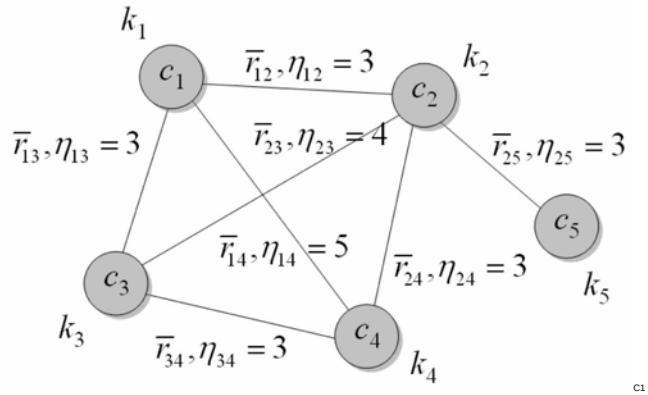


Fig. 6. Ontology used for measuring time complexity.

Work on as many pages as needed, thus effectively simulating the next-generation Semantic Web repositories. On the other hand, by statistically annotating Web pages, we do not incur in the risk of biasing the result. In order to compare our measures with those of [10], we worked with the same ontology (travel.owl), and we selected the same query (in the query, illustrated in Fig. 6, specific keywords and concepts defined in [10] have been replaced with numeric indexes). We automatically generated a Web page database with one million pages, each page containing all the keywords specified in the query. For each page, we constructed a semantic annotation based on the concepts defined in the selected ontology, randomly associating to each keyword one of the concepts in the ontology. We adjusted the statistical parameters so as to obtain a set of approximately 100,000 pages (precisely, 96,843 pages) including at least one of the keywords associated exactly to the concept specified in the query. Finally, we added semantic relations between concepts by uniformly distributing them across pages. In this way, each pair of concepts was linked by a variable subset of the relations associated to that pair in the ontology (each page containing approximately 10 relations). The distribution of concepts and relations in the Web page database for each concept cijci € CQ, reports the number of pages ontaining exactly the association keyword/concept (k, c) efined in the query. Moreover, for each couple of concepts ci; cjjci; cj € CQ, it reports the number of pages in which oth the concepts are associated to the keywords specified in the user's query

6.2 ACCURACY

The accuracy of the result set generated is evaluated by running the query "hotel-hill station", "Ooty". The web page written by the traditional search engine shows that out of scope pages are ranked as very relevant and potentially interesting pages are positioned at the end of the result. But now we manually annotate each page using concepts accommodation, destination, accommodation rating and activity in the travel.owl ontology. It is observed that the ranking is significantly improved. Out of the six entries at least four are to the satisfaction of the user query

7. CONCLUSION AND FUTUREWORK

The next-generation Web architecture represented by the Semantic Web will provide adequate instruments for improving search strategies and enhance the probability of
seeing the user query satisfied without requiring tiresome manual refinement. They mainly use page relevance criteria based on information that has to be derived from the whole knowledge base, making their application often unfeasible in huge semantic environments. In this work, we propose a novel ranking strategy that is capable of providing a relevance score for a Web page into an annotated result set by simply considering the user query, the page annotation, and the underlying ontology. Page relevance is measured through a probability-aware approach that relies on several graph-based representations of the involved entities. By neglecting the contribution of the remaining annotated resources, a reduction in the cost of the query answering phase could be expected. Despite the promising results in terms of both time complexity and accuracy, further efforts will be requested to foster scalability into future Semantic Web repositories based on multiple ontologies, characterized by billions of pages, and possibly altered through next generation "semantic" spam techniques

ACKNOWLEDGMENTS

I like to thank all my friends for their valuable suggestions for this paper.

REFERENCES

[1] B. Aleman-Meza, C. Halaschek, I. Arpinar, and A. Sheth, "A Context-Aware Semantic Association Ranking," Proc. First Int'l Workshop Semantic Web and Databases (SWDB '03), pp. 33-50, 2003.
[2] K. Anyanwu, A. Maduko, and A. Sheth, "SemRank: Ranking Complex Relation Search Results on the Semantic Web," Proc. 14th Int'l Conf. World Wide Web (WWW '05), pp. 117-127, 2005.
[3] R. Baeza-Yates, L. Caldero´n-Benavides, and C. Gonza´lez-Caro, "The Intention behind Web Queries," Proc. 13th Int'l Conf. String Processing and Information Retrieval (SPIRE '06), pp. 98-109, 2006.
[4] T. Berners-Lee and M. Fischetti, Weaving the Web. Harper Audio, 1999.
[5] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific Am., 2001.
[6] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual
[7] Web Search Engine," Proc. Seventh Int'l Conf. World Wide Web (WWW '98), pp. 107-117, 1998.
[8] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv, "XSEarch: A Semantic Search Engine for XML," Proc. 29th Int'l Conf. Very Large Data Bases, pp. 45-56, 2003. [8] L. Ding, T. Finin, A. Joshi, R. Pan, R.S. Cost, Y. Peng, P. Reddivari, V. Doshi, and J. Sachs, "Swoogle: A Search and Metadata Engine for the Semantic Web," Proc. 13th ACM Int'l Conf. Information and Knowledge Management (CIKM '04), pp. 652-659, 2004.
[9] L. Ding, T. Finin, A. Joshi, Y. Peng, R. Pan, and P. Reddivari, "Search on the Semantic Web," Computer, vol. 38, no. 10, pp. 62-69, Oct. 2005.

[10] Y. Li, Y. Wang, and X. Huang, "A Relation-Based Search Engine in Semantic Web," IEEE Trans. knowledge and Data Eng., vol. 19, no. 2, pp. 273-282, Feb. 2007.
[11] R. Guha, R. McCool, and E. Miller, "Semantic Search," Proc. 12th Int'l Conf. World Wide Web (WWW 03), pp. 700-709, 2003.

Authors Biography

Mr. G.Santosh Kumar received the B.Tech degree from the Department of Information Technology, GMRIT, JNTUniversity, Hyderabad in 2006 and He is currently pursuing M.Tech in the Department Of Computer Science and Engineering, Avanthi Institute of Engineering and Technology, Vishakhapatnam, JNTU university His research interests include Knowledge and Data Engineering, Web Technologies Information security

Mr. Satya P Kumar Somayajula is working as an Asst.Professor, in CSE Department, Avanthi Institue of Engg & Tech, Tamaram, Visakhapatnam,A.P., India. He has received his M.Sc(Physics) from Andhra University, Visakhapatnam and M.Tech (CST) from Gandhi Institute of Technology And Management University (GITAM University), Visakhapatnam, A.P., INDIA. He published 7 papers in reputed International journals & 5 National journals. His research interests include Image Processing, Networks security, Web security, Information security, Data Mining and Software Engineering.