EEWDCO: The Efficient way of Enhancing Web Document Clustering using Ontologies

Raja Varma Pamba
LBS College of Engineering
Kasaragod , Kerala, India

Elizabeth Sherly
IIITMK, Technopark
Trivandrum, Kerala, India

ABSTRACT
The challenging aspects of immensely huge information in WWW poses a huge threat of retrieving correct information at the correct instant of time. When world is growing in a fast pace so is the exponential growth of information available in millions forms [4]. So retrieving information matching in synonym and in the right context is unmanageable in the current scenario. The need of the hour is to establish a system that could intelligently decipher the context and inherently extract the correct implicit meaning as what is expected by the user searching for information. Main contribution of this paper is in its novel attempt to incorporate the notion of HITS algorithm with the utilities of Ontologies for the effective web document clustering, which could in tremendous way enhance the present information retrieval engulfed in the hurdles of accessing, extracting, interpreting and finding relevant information as is expected by the user themselves.

1. INTRODUCTION
Today's Web has grown to an immense volume with millions of pages adding up daily making it massive and unstructured. Current scenario depicts a situation where Web could present the information or knowledge by matching the keywords or terms with its database and retrieve search results accordingly. For example a user might have queried to find the best possible tour package meeting his requirements stated. Presently the web has no such functionality where in it could infer from the given data what exactly user meant and expects as outcome, instead it matches all terms and finds all possible matches without any filtering of concepts ,meaning and just delivering the results. This scenario demands for a system which could ultimately retrieve links and web pages meeting the equivalent meanings by reading and understanding the minds of human in machine processible manner. This transition from a syntactic web to a semantic web adds quality and value addition to the present scenario by attaching meaning to the search query.

The whole paper is structured as follows: section 1 deals with Introduction. Section 2 deals with knowing Ontologies and understanding the utilities of Ontologies. Section 3 discusses about HITS algorithm. Section 4 proposes the algorithm for the new system which incorporates both Ontologies and HITS algorithm for web document clustering. Section 5 deals with conclusion and future work.

2. ONTOLOGIES FOR WEB DOCUMENTS
Semantic web could retrieve information nor cluster web pages meaningfully only if it could intelligently decipher what exactly is the concepts and what is the relationship between the content in search and the knowledge to be represented, exchanged and retrieved or classified[1][2]. For this to happen the framework of web document clustering should exist on top of a system that could interweave among all available links and web pages with relevant concepts and relate them in a meaningful manner. Ontology more over act as knowledge representations languages that create explicitly domain conceptualizations, consisting of concepts, their definitions and the relationship between them.[3]. In its most prevalent use an ontology refers to an engineering artifact, describing a formal, shared conceptualization of a particular domain of interest [1].Ontology help computers interact without the knowledge of individual system technologies, architecture and application domain. Ontology can be built on concepts newly or existing ontologies can be extended to fine tune with research work. For example ontologies for medical, travel, tourism are existing ontologies which if relevant to the field of research need only to reused or extended with new value additions. In simple words, Ontologies built for an application act as a dictionary wherein the application can refer for correct meaning before using or applying any rules for retrieval. In this proposed system we are trying to incorporate this basic utility service provided by an ontology for web documents. By this the proposed systems will try to deliver links pertaining to the concepts and meanings defined in Ontology for web documents, which matches in semantics with the query being searched.

Fig 1: Ontologies extracting relevant web links.

3. HITS (HYPERLINK-INDUCED TOPIC SEARCH) ALGORITHM
As per **HITS** algorithm in its normal functioning identifies good authorities and hubs for a topic by assigning two numbers to a page: an authority and a hub weight. Authority for the query finds those pages that are ideal for our query and hubs are those that contain useful links towards the authoritative pages. They act more like advertising the correct authoritative pages to link with. These weights are defined recursively. A higher authority weight occurs if the page is pointed to by pages with high hub weights. A higher hub weight occurs if the page points to many pages with high authority weights.In order to get a set rich in both hubs and authorities for a query *Q*,our work first collected the top 200 documents that contain the highest number of occurrences of the search phrase *Q*. These, as pointed out before may not be of tremendous practical relevance, but one has to start somewhere. Kleinberg HITS algorithm points out that the pages from this set called root (R_Q) are essentially very heterogeneous and in general contain only a few links to each other. So the web sub graph determined by these nodes is almost totally disconnected; in particular, we cannot enforce Page Rank techniques on R_Q.As a results of implementing HITS algorithm in our work, found huge list of top ranked web pages and documents being retrieved out of which many were irrelevant. Authorities for the query *Q* are not extremely likely to be in the root set R_Q. However, they are likely to be pointed out by at least one page in *R*Q. So it makes sense to

extend the subgraph R_Q by including all edges coming from or pointing to nodes from *R*Q. We denote by S_Q the resulting subgraph and call it the seed of our search. Notice that S_Q we have constructed is a reasonably small graph. It is also likely to contain a lot of authoritative sources for *Q*. The question that remains is how to recognize and rate them? To filter the best link to the authoritative pages matching our context defined in query it was necessary to incorporate ontologies to structure the queries to fit to the expected meaning. These two type of Web pages are extracted by iteration that consists of following two operations:

$$x_p = \sum_{q,q\to p} y_q$$

$$y_p = \sum_{q,p\to q} x_q$$

For a page p,the weight of x_p is updated to be the sum of y_p over all pages q that link to where the notation q→p indicates that q links to page p. A good hub increases the authority weight of the pages it points. A good authority increases the hub weight of the pages that point to it. The idea is then to apply the two operations above alternatively until equilibrium values for the hub and authority weights are reached. For example will see how to find Adjacency matrix A,u and v being calculated for a particular problem as given below in Fig [2] and Fig[3]:

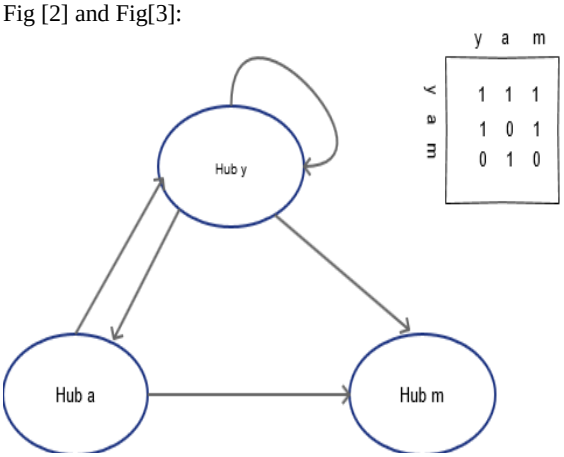

Fig 2: Example for Finding A from the subgraph

Let *A* be the adjacency matrix of the graph S_Q and denote the authority weight vector by *v* and the hub weight vector by *u*, where

$$v = \begin{bmatrix} x1 \\ x2 \\ \vdots \\ xn \end{bmatrix} \text{ and } u = \begin{bmatrix} y1 \\ y2 \\ \vdots \\ yn \end{bmatrix}$$

The two update operations described above in the figure translate to:

$$v = A^t.u$$
$$u = A.v$$

A= 1 1 1 / 1 0 1 / 0 1 0 $\quad$ $\bar{A}$ = 1 1 0 / 1 0 1 / 1 1 0

u(y) = 1 1 1 1 1
u(a) = 1 1 4/5 .75 ..... .732
u(m) = 1 1 1 1 1
v(y) = 1 1 1 1.....1
v(a) = 1 2/3 .71 .73 732
v(m) = 1 1/3 .29 .27 .268

Fig 3: Example for calculating u and v from A and A^t

We could consider that the initial weights of the nodes are as follows as per HITS algorithm:

$$u_0 = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \text{ and } v_0 = A^t \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

then, after *i* steps we get the system converges to:

$$v_i = (A^t.A).v_{i-1}$$

$$u_i = (A.A^t).u_{i-1}$$

4. PROPOSED SYSTEM- INCORPORATING HITS ALGORITHM WITH ONTOLOGY

The proposed system plans to implement a new system considering the limitations of HITS algorithm in retrieving huge amount of irrelevant hubs which has no semantic correlation with the query[2].To this search query if we could incorporate an ontology for web documents defined with concepts and relationship suiting to our retrieval to extract appropriate meaning ,then the issue of irrrelvalent hubs can be curtailed to a large extent. As Kleinberg argues [5], HITS algorithm extracts multiple densely linked collections of hubs and authorities on multiple eigenvectors as shown in Fig 4.

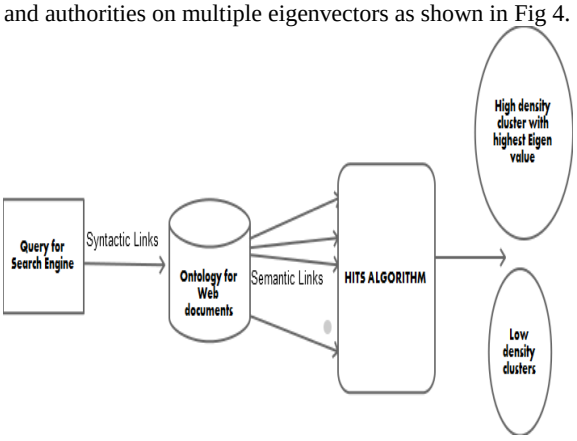

Fig 4: Proposed System of EEWDCO

The algorithm for the proposed system works as follows:

Step1: Build an ontology for web documents in protégé tool.
Step2: Collect al top list of ranked pages r for the query Q from a text based search engine Google. These r pages are referred to as root set R_Q.

Step 3: Expand R_Q to include any page pointed to by pages in R_Q and at most d pages pointing to pages in R_Q. we can call this set as seed, S_Q with size n. At this stage, this seed S_Q is processed with ontologies for web documents and is filtered to extract seeds which contains only valuable links matching in meaning to the query being searched. This we call Optimized seeds O_{SQ}

Step 4: Let $G[O_{SQ}]$ denote the subgraph induced on the pages in S_Q. Both transverse links and intrinsic links are seen in $G[O_{SQ}]$.All those links with same domain name ie intrinsic links are deleted and only those links with different domain names ie transverse links are retained for the next step.

Step 5: Make the n by n adjacency matrix A and its transpose matrix A^t.

Step 6: If v_1, ..., v_i, ... is the sequence of authority weights we have computed, then V_1, ..., Vi ... converges to the unique probabilistic vector corresponding to the dominant eigenvalue of the matrix A^tA. Return them as authorities. We denoted in here by V_k the vector *v*k normalized so that the sum of its entries is 1.

Step 7: Likewise, if u_1, ... ,u_i, ... are the hub weights that we have iteratively computed, then U_1, ... ,U_i, ... converges to the unique probabilistic vector corresponding to the dominant eigenvalue of the Matrix AA^t. We use the same notation, that $U_i = (1 / c)u_i$, where c is the scalar equal to the sum of the entries of the vector u_i.

Step 8: So the authority weight vector is the probabilistic eigenvector corresponding to the largest eigenvalue of A^tA, while the hub weights of the nodes are given by the probabilistic eigenvector of the largest eigenvalue of AA^t.

Step 9: The eigenvalue that we obtained represent the density of links in a cluster. The largest Eigen value obtained from the iterative algorithm can be called Primary eigenvalue that represent the highest measure of similarity between links and pages in a cluster. Remaining values forms the other sector of bipartite cluster which has lesser similarity.

5. CONCLUSION AND FUTURE WORK

In this article the proposed work stress upon the incorporation of HITS with the domain ontology for effective web document clustering. For this, once the domain ontology construction is done, the domain expert is required to validate and correct the generated ontology. Ontology editors and other toolkits are also necessary for the whole process of automated ontological construction, maintenance. And annotation. Additionally, the SOM algorithm can be adopted for concept clustering and defining taxonomic relationships. The attributes and operations of concepts can be extracted based on ontology construction. HITS algorithm helps in retrieving the best authorities and hubs respectively. The first prototype of the system being built is in its preliminary stage. The early results are encouraging in terms of the quality and robustness of our current implementation, however, there is clearly a much more work needed to make this system easy to use for our target user base. As a future work, the query system can be made more efficient. This involves in improvising the mapping techniques that could be used in order to determine which word of the ontology a generic word should be mapped to. This is a topic that can be planned for future, since it is used to define the small sets of terms that define a web document.

6. REFERENCES

[1] Gruber, T.R. A Translation approach to portable ontologies. Knowledge Acquisition, 1993, 5(2), 199-220.

[2] J. Kleinberg. Hubs, Authorities, and Communities. ACM Computing Surveys, 31(4es, Article No.5), 1999.

[3] Smith, Michael K; Welty, Chris & McGuinness, Deborah L. OWL Web Ontology Language Guide, W3C, 2008.pp.7-15

[4] D. Butler. Souped-up search engines. Nature, 405:112–115,May, 2000.

[5] J. Kleinberg. Authoritative sources in a hyperlinked environment,

[6] 1997. Research Report RJ 10076 (91892), IBM.