SEMANTIC WEB MINING BASED ON BIG DATA USING RDF FOR WEB SEARCHE'S

Ms A.Sivasankari[1], Mrs N.Sindhuja[2], Mrs S.Shanthi[3]

[1]Head of the department , Department of computer science, D.K.M College for women (Autonomous), Vellore.

[2]Research scholar, Department of Computer science, D.K.M College for women (Autonomous),Vellore.

[3]Assistant professor ,Department of computer science, D.K.M College for women (Autonomous),Vellore.

Abstract - The large amount of Semantic Web data and its fast growth pose a significant computational challenge in performing efficient and scalable reasoning. On a large scale, the resources of single machines are no longer sufficient and we are required to distribute the process to improve performance Constructing transfer inference forest and effective assertion triples, the storage is largely reduced and the reasoning process is simplified and accelerated. Resource description framework is a basic representation of ontologies used to describe the knowledge in the Semantic Web. A large volume of Semantic Web data, the fast growth of ontology bases has brought significant challenges in performing efficient and scalable reasoning. Centralized reasoning methods are not sufficient to process large ontologies. Distributed reasoning methods are thus required to improve the scalability and performance of inferences. We have implemented Web PIE (Web-scale Inference Engine) and we demonstrate its performance on a cluster of up to 64 nodes. We have evaluated our system using very large real-world datasets (Bio2RDF, LLD, LDSR) and the LUBM synthetic benchmark, scaling up to 100 billion triples. Results show that our implementation scales linearly and vastly outperforms current systems in terms of maximum data size and inference **speed.**

Key Words: Web Usage Mining, Adaptive hypermedia PEL, Ontology, Clustering, Privacy Preserving, Greed Algorithm

I. INTRODUCTION

A "web search engine" is a software system that is designed to search for information on the World Wide Web. The search results are generally presented in a line of results often referred to as Search Engine Results Pages (SERPs). The information may be a mix of web pages, images, and other types of files. Some search engines also mine data available in databases or open directories. Unlike web directories, which are maintained only by human editors, search engines also maintain real-time information by running an algorithm on a web crawler.

The web search engine has long become the most important portal for ordinary people looking for useful information on the web. However, users might experience failure when search engines return irrelevant results that do not meet their real intentions. Such irrelevance is Largely due to the enormous variety of user's contexts and backgrounds, as well as the ambiguity of texts. Personalized Web Search (PWS) is a general category of search techniques aiming at providing better search results, which are tailored for individual user needs. As the expense, user information has to be collected and analyzed to figure out the user intention behind the issued query.

The solutions to PWS can generally be categorized into two types, namely click-log-based methods and profile-based ones. The click-log based methods are straightforward— they simply impose bias to clicked pages in the user's query history. Although this strategy has been demonstrated to perform consistently and considerably well, it can only work on repeated queries

from the same user, which is a strong limitation confining its applicability. In contrast, profile-based methods improve the search experience with complicated user-interest models generated from user profiling techniques. Profile-based methods can be potentially effective for almost all sorts of queries, but are reported to be unstable under some circumstances.

1.1 PROBLEM STATEMENT

The two web based systems consisting of Generic system and Semantic Search Engine system with objective of Web Search Engine is to develop the Generic Web Search Engine frame work and semantic search engine. In the traditional way of teaching, practicing, and assessing, the teachers design or choose assignments for weekly exercise sheet according to the course, the exercise sheet may be distributed as a printed document or made available online. The students can work through the exercise sheet at home and present their solution at the blackboard. The teacher gives feedback and the tutor may take notes about student's performance. For large groups of students, manual correction is labor and time-intensive but the problems are especially grave for programming assignments, with the rise in online education, the CDL wishes to integrate their modules into several distances learning course to attract more learning providers.

Online courses are instructional content which are delivered through online. The Hybrid courses content are in the class room settings and Web facilities courses content are partially in the classroom settings.

II. Related work

Expressing meaning" is the most important feature of the Semantic Web. In order to accomplish this goal, several layers of representational structures are needed. They are presented in the figure below, among which the following layers are the basic ones:

- The XML layer, which represents the structure of data;
- The RDF layer, which represents the meaning of data;
- The Ontology layer, which represents the formal common agreement about meaning of data;
- The Logic layer, which enables intelligent reasoning with meaningful data.

2.1 Description logics as ontology languages for the semantic web:

The Semantic Web aims to build machine-understandable Web resources, whose information can then be shared and processed both by automated tools, such as search engines, and by human users. This sharing of information between different agents requires semantic mark-up. To make sure that different agents have a common understanding of these terms, one needs ontologies in which these terms are specified precisely, and which thus establish a shared terminology between the agents. The use of ontologies in this context requires a well-designed, well-defined, and Web-compatible ontology language with supporting reasoning tools. The syntax of this language should be both intuitive to human users and compatible with existing Web standards. Its semantics should be formally specified and its expressive power should be adequate.

Reasoning is an important consideration in the design of ontology. It can be employed in different development phases. During ontology development, it can be used to test whether concepts are consistent and to derive implied relations. In particular, one usually wants to compute the concept hierarchy. Interoperability and integration of different ontologies is also an important issue. Reasoning may also be used when the ontology is deployed, i.e., when a Web page is already annotated with its concepts.

Regarding an ontology language for the Semantic Web, there was a joint US/EU initiative for a W3C ontology standard, which was for historical reasons called DAML+OIL [1]. This language has a syntax based on RDF Schema [2], and it is based on common ontological primitives from Frame Languages (which support human understandability). Its semantics can be defined by a translation into an expressive version of Description Logic, known as DLSHIQ [3], and the developers have tried to find a good compromise between expressiveness and the complexity of reasoning. Although reasoning in SHIQ is decidable, it has a rather high worst-case complexity. Nevertheless, there is an optimized SHIQ reasoner (FaCT) [4] available, which behaves quite well in practice. Some of the features of SHIQ make the DL expressive enough to be used as an ontology language. Firstly, SHIQ provides number restrictions that are more expressive than other versions. Secondly, SHIQ allows the formulation of complex terminological axioms. Thirdly, SHIQ also allows for inverse roles, transitive roles, and sub roles. Fact is based on the tableaux reasoning algorithm, which computes the deductive closure of the axioms. Thus it is highly inefficient.

- A completion graph or a tableau that represents a model of the DL language.
- A set of tableau expansion rules to construct a complete and consistent completion graph.
- A set of blocking rules to detect infinite cyclic models and ensure termination. A set of clash conditions to detect logical contradictions

2.1. Profile Based Personalization

The main focus of profile-based PEL in the previous works was on improving the search utility. The basic idea is to tailor the search results by referring to, often implicitly, a user profile that reveals an individual information goal. The previous solutions to PEL can be reviewed on two aspects, namely representation of profiles, and the measure of the effectiveness of personalization.

Many profile representation are available in literature to facilitate different personalization strategies. Previous techniques utilize term lists or bag of words to represent their profile. The most recent works of profiles are built in hierarchical structure due to their stronger descriptive ability, better scalability and higher access efficiency. Mapping from one ontology to another one is expressing of the way how to translate statements from ontology to the other one. Often it means translation between concepts and relations. In the simplest case it is mapping from one concept of the first ontology to one concept of the second ontology.

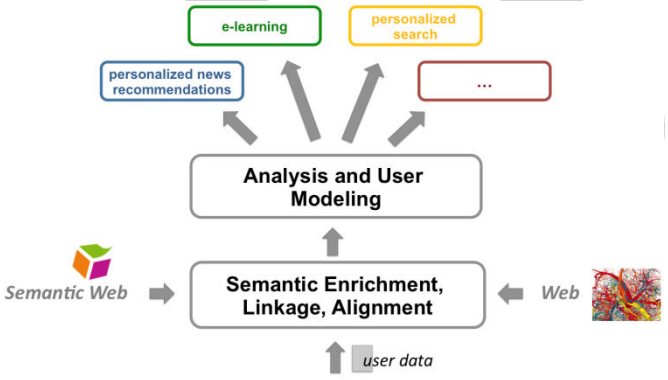

Fig 1: the Search Engine based Profile Search

The hierarchical representations are constructed with the existing weighted topic hierarchy/graph and so on. Another work is built automatically via term-frequency analysis on the user data. in our proposed UPS framework, our focus is not on the implementation of the user profiles. Our framework can adopt any hierarchical representation based on taxonomy of knowledge.

2.2 User Interest Profiling

Web Search Engine uses "concepts" to model the interests and preferences of a user. In mobile searches the location information is important so the concepts are further classified into two different types as content concepts and location concepts. The concepts are modeled as ontology's to capture the relationships between the concepts. The characteristics of the content concept sand location concepts are dissimilar. So, we propose two different techniques to build the content ontology and location ontology. This ontology's indicate a possible concept space from the user's queries which are maintained with Ontologies data for further preference adaptation. Ontologies are adopted to model the concept space in Web Search Engine since they not only represent concepts but also capture the relationships between the concepts.

2.3. Personalized Ranking Functions

Ranking SVM (RSVM) is employed to learn a personalized ranking function for rank adaptation of the results according to the user content and location preferences while receiving the user's preferences. From the search results of the document features, a set of content concepts and location concepts can be extracted for a given query. Since each document can be represented by a feature vector, it can be treated as a point in the feature space. RSVM aims at finding a linear ranking function which holds many document preference pairs as possible, when preference pairs are used as the input. An adaptive implementation, SVM light available at, is used in our experiments. The two main issues in the RSVM training process are discussed below:

1. How to extract the feature vectors for a document;
2. How to combine the content and location weight vectors into one integrated weight vector.

III. PREVIOUS IMPLEMENTATIONS

The Web Search Engine engine has become the most important portal for ordinary people who are looking for useful information on the web. However, when the search engine returns irrelevant results that do not meet their real intentions the users experience failures in this case. A major problem in Web Search Engine is that the interactions between the users and search engines are

limited by small form factors of the web devices. This result in submission of shorter more ambiguous queries by the web users compared to their Web Search Engine counterparts. In order to return highly relevant results to the users, the Web Search Engine engines must be able to profile the users interests and personalize the search according to the users profiles. To capture a users interests from personalization, analyze the users Ontologies data. Most of the previous work assumed that all concepts are of the same type.

A major problem in mobile search is that the interactions between the users and search engines are limited by the small form factors of the mobile devices. As a result, mobile users tend to submit shorter, hence, more ambiguous queries compared to their web search counterparts. In order to return highly relevant results to the users, mobile search engines must be able to profile the users' interests and personalize the search results according to the users' profiles. A practical approach to capturing a user's interests for personalization is to analyze the user's clickthrough data. Leung, et. al., developed a search engine personalization method based on users' concept preferences and showed that it is more effective than methods that are based on page preferences. However, most of the previous work assumed that all concepts are of the same type. Observing the need for different types of concepts.

It proposes a privacy-preserving personalized web search framework UPS, which can generalize profiles for each query according to user-specified privacy requirements. Relying on the definition of two conflicting metrics, namely personalization utility and privacy risk, for hierarchical user profile, we formulate the problem of privacy-preserving personalized search as 5-Risk Profile Generalization, In order to handle the queries that focus on location information, a number of location-based search systems designed for location queries have been proposed. Proposed a location-based search system for web documents. Location information was extracted from the web documents, which was converted into latitude-longitude pairs.

IV. SYSTEM IMPLEMETNATION

4.1 ONTOLOGY BASED RESOURCE DESCRIPTION FRAMEWORK (RDF)

Phase 1: Using WordNet Ontology for Document Features Reduction

Ontology plays a vital role in document clustering process by decreasing the large number of documents features from thousands to tens of features only. The features reduction process utilizes ontology characteristic which includes semantic relations between words such as synonyms and hierarchical relations between words. From hierarchy relations we can get word parent and use it for representing document features. For example the words corn, wheat, and rice can be represented by only one word which is plant. Also words such as beef and pork can be represented by words meat or food depending of the degree of hierarchy that will be used in the clustering process. Exploiting semantic relations between words will help in setting documents that contain words such as rice, wheat and corn at the same cluster. This paper utilizes semantic relations that are included in WordNet ontology as follow:user profile into account. This process adds the inherited properties to the properties of the local user profile. Then the process loads the data for the foreground and the background of the map according to the described selection in the user profile.

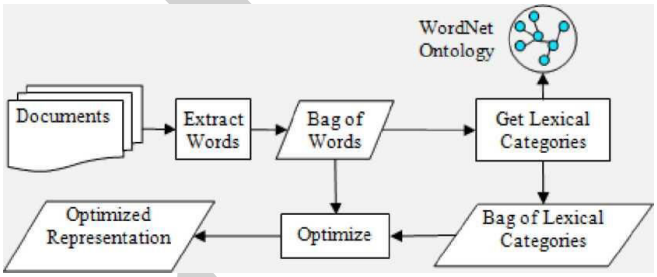

Fig : Steps for representing documents as bag of lexical categories

Given a set of documents, the first step in this phase is the Extract-Words process. The Extract-Words process removes stop words and extracts the remaining words, it generates two files; the vocabulary file that contains the list of all words; and the document-words file which stores associations between words and document (bag of words). Figure (2) displays the format of the resulting document-words file [18].

D //Number of documents

W //Number of in words in the vocabularv

NNZ // Number of nonzero counts in the bag-of-TVords

docIDwordlD count Document identifier . word identifier . and count

docIDwordlD count

docIDwordlD count

docIDwordlD count

The next process is "Get Lexical Categories"; it converts the bag of words into a bag of WordNet lexical categories. This process involves the following steps:

- Mapping each word to its WordNet lexical category. This step generates a WordID-CategoryID file. In case of the word doesn't have a corresponding WordNet lexical category, it is mapped to Uncategorized category.
- Generating a bag of lexical categories that replaces " docID wordID count " to "docID CategoryID count".
- The third process is "Optimize", the input to this process is the bag of words file or bag of lexical categories, and the output is an optimized representation.

In case of bag of words, each document will be represented by one line as follows:

"docID word1ID:count word2ID:count wordnID:count ".

In case of bag of lexical categories, each document will be represented also by one line as follows:

"docIDLexical1ID:countLexical2ID:countLexicalnID:count"

The optimized representation of bag of word reduces file size dramatically. For example, for "PubMed" bag of words dataset [18]: it reduced the bag of words file from 6.3 gigabytes to 928.475 megabytes**.**

Phase 2: Bisecting k-means Implementation over MapReduce Framework

To overcome the continuous increase in document size, we used MapReduce to run the bisecting k-means algorithm. To implement bisecting k-means over the MapReduce framework, first, traditional k-means algorithm is implemented to generate two clusters; we adapt the method which is presented in [19] as follow:

Initialize centers.

1. In Map function each document vector is assigned to the nearest center. The key of map function is the document Id and the value is document vector, the map function emits cluster index, and associated document vector.
2. In Reduce function new centers are calculated. The key will be the cluster index and the value is document vector. Using cluster index instead of Cluster centroid in reduce function as a key reduce the amount of data that will be aggregated by reduce workers nodes.
3. In the clear function of reducer, new centers are saved in a global file. It will be used for next iteration of k-means algorithm.
4. Finally, if convergence is achieved then stop, else go to step 2.

The main idea of implementing the bisecting k-means algorithm to run on MapReduce is controlling Hadoop Distributed File System file paths of dataset, cluster centers, and output clusters. Controlling HDFS paths help in implementing algorithms that use multiple MapReduce iterations such as Bisecting k-means algorithm.

4.5 ALGORITHM IMPLEMENTATION
ALGORITHMIC ANALYSIS:

1. Bisecting k-means over MapReduce Algorithm

Bisecting k-means(PI,k,Nd,Pcc,Po)

Output: Clusters' centers, set of clusters. BEGIN:

R= 0 // R is number of clusters obtained FirstTime = true //the first time of calling basic

k-means

while (R < K) BEGIN

B asic- K-means(PI,Nd,PCC,P0); if (FirstTime)

R = R+ 2;

FirstTime = false;

else

R = R+1;

PI = Path of Largest cluser returned by

Basic-K-means PCC= concatenate(PCC,"1"); P0 = concatenate(P0,"1");

END

END

V. RESOURCE DESCRIPTION FRAMEWORK (RDF)

The Resource Description Framework (RDF) is a family of World Wide Web Consortium (W3C) specifications originally designed as a metadata data model. It has come to be used as a general method for conceptual description or modeling of information that is implemented in web resources, using a variety of syntax notations and data serialization formats. It is also used in knowledge management applications.

The Resource Description Framework (RDF) is a general framework for how to describe any Internet resource such as a Web site and its content. An RDF description (such descriptions are often referred to as metadata, or "data about data") can include the authors of

the resource, date of creation or updating, the organization of the pages on a site (the sitemap), information that describes content in terms of audience or content rating, key words for search engine data collection, subject categories, and so forth.

The Resource Description Framework will make it possible for everyone to share Web site and other descriptions more easily and for software developers to build products that can use the metadata to provide better search engines and directories, to act as intelligent agents, and to give Web users more control of what they're viewing. The RDF is an application of another technology, the Extensible Markup Language (XML), and is being developed under the auspices of the World Wide Consortium

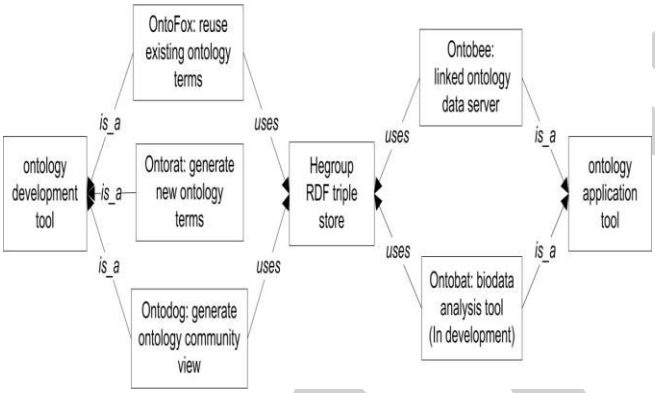

Fig : RDF Framework

RDF Works

An Internet resource is defined as any resource with a Uniform Resource Identifier (URI). This includes the Uniform Resource Locators (URL) that identifies entire Web sites as well as specific Web pages. As with today's HTML META tags, the RDF description statements, encased as part of an Extensible Markup Language (XML) section, could be included within a Web page (that is, a Hypertext Markup Language - HTML - file) or could be in separate files. RDF is now a formal W3C Recommendation, meaning that it is ready for general use. Currently, a second W3C recommendation, still at the Proposal stage, proposes a system in which the descriptions related to a particular purpose (for example, all descriptions related to security and privacy) would constitute a class of such like descriptions (using class here much as it is used in object-oriented programming data modeling and programming).

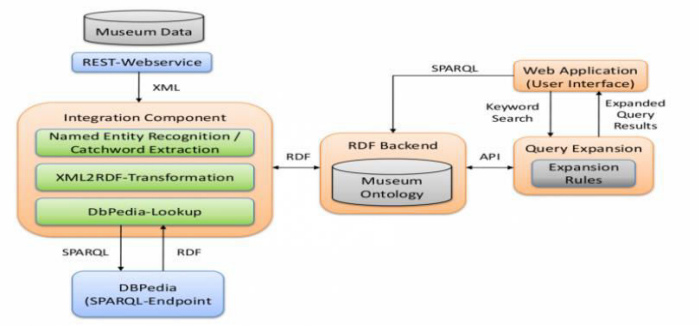

Fig : Corporate Semantic Web

The development of World Wide Web and its usage grows, it will continue to generate ever more content, structure, and usage data and the value of Web mining will keep increasing. Research needs to be done in developing the right set of Web metrics, and their measurement procedures, extracting process models from usage data, understanding how different parts of the process model impact various Web metrics of interest, how the process models change in response to various changes that are made changing stimuli to the user, developing Web mining techniques to improve various other aspects of Web services, techniques to recognize known frauds and intrusion detection.

RDF Crawler is a stand-alone application, which is given URIs and builds an RDF database from it (or increments an existing database).Ontology servers and other tools dealing with Meta information sometimes need to retrieve facts describing resources on the Web. The current standard of making statements about Web resources is RDF (Resource Description Framework), and there are a few more standards which build on top of the RDF, e.g. RDFS and OIL. Therefore we may need a utility to download RDF information from all over the Internet. This utility will be henceforth called RDF Crawler. It is a tool which downloads interconnected fragments of RDF from the Internet and builds a knowledge base from this data. At every phase of RDF crawling we maintain a list of URIs to be retrieved as well as URI filtering conditions (e.g. depth, URI syntax), which we observe as we iteratively download resources containing RDF. To enable embedding in other tools, RDF Crawler provides a high-level programmable interface (Java API). RDF Crawler utility is just a wrapper around this API either a console application, or a windows application or a servlet.

$$p(C|F_1,\ldots,F_n)$$

over a dependent class variable C with a small number of outcomes or classes, conditional on several feature variables F_1 through F_n. The problem is that if the number of features n is large or if a feature can take on a

large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes' theorem, this can be written

$$p(C|F_1,\ldots,F_n) = \frac{p(C)\ p(F_1,\ldots,F_n|C)}{p(F_1,\ldots,F_n)}.$$

In plain English, using Bayesian Probability terminology, the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}.$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on C and the values of the features F_i are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model

$$p(C, F_1, \ldots, F_n)$$

Which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$\begin{aligned} p(C, F_1, \ldots, F_n) &= p(C)\ p(F_1, \ldots, F_n|C) \\ &= p(C)\ p(F_1|C)\ p(F_2, \ldots, F_n|C, F_1) \\ &= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ p(F_3, \ldots, F_n|C, F_1, F_2) \\ &= p(C)\ p(F_1|C)\ p(F_2|C, F_1)\ \ldots p(F_n|C, F_1, F_2, F_3, \ldots, F_{n-1}) \end{aligned}$$

Now the "naive" conditional independence assumptions come into play: assume that each feature F_i is conditionally independent of every other feature F_j for $j \neq i$, given the category C. This means that

$$p(F_i|C, F_j) = p(F_i|C)$$
$$p(F_i|C, F_j, F_k) = p(F_i|C)$$
$$p(F_i|C, F_j, F_k, F_l) = p(F_i|C)$$

and so on, for $i \neq j, k, l$. Thus, the joint model can be expressed as This means that under the above independence assumptions, the conditional distribution over the class variable is:

$$\begin{aligned} p(C|F_1,\ldots,F_n) &\propto p(C, F_1, \ldots, F_n) \\ &\propto p(C)\ p(F_1|C)\ p(F_2|C)\ p(F_3|C)\ \cdots \\ &\propto p(C) \prod_{i=1}^{n} p(F_i|C)\,. \end{aligned}$$

$$p(C|F_1,\ldots,F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

Where the evidence $Z = p(F_1, \ldots, F_n)$ is a scaling factor dependent only on $F_1, \ldots, F_n$, that is, a constant if the values of the feature variables are known.

As there is no publically available dataset for personalized search evaluation purpose, we exploited a search log of a commercial web search engine, namely Exhaled, and we extracted the search history of 10 users collected along three months. Descriptions of search sessions, queries and document collection are given below. Cluster configuration is how we setup the Hadoop nodes, for example the number of Hadoop nodes, the processor frequency of node machines, and the number of reducers. These configurations are factors that outside the system's internal logic, but could have effect on its performance.

EVALUATION RESULT:

The first page provides more informative comparison. I found that Google and at least one other search engine returns 7% of results of queries in the first page. Google refers 7.9% queries to its own content on the first page of results without agreement from either rival search engine. Meanwhile, Bing and at least one other engine refer to Microsoft content in 3.2% of the queries. Bing references Microsoft content without agreement from either Google or Blekko in 13.2% of the queries:

Fig 3: Search Results list Google Content on the First page

Percentage of Google Organic Results with Google Content Not Ranked Similarly by Rival Search Engines

	Google Content Not Mentioned in Corresponding Top 1, 3, 5 or First Page of Results		
	Bing	Blekko	Bing & Blekko
Top 1	78.6%	57.1%	50.0%
N = 14	11	8	7
Top 3	37.5%	58.3%	29.2%
N = 24	9	14	7
Top 5	38.7%	64.5%	35.5%
N = 31	12	20	11
First Page	51.1%	68.9%	48.9%
N = 45	23	31	22

Table: Percentage of Google Search results with Google Content based Search

When Google ranks its own content highly, at least one rival engine typically agrees with this ranking. For example, when Google places its own content in its Top 3 results, at least one rival agrees with this ranking in over 70% of queries. Bing especially agrees with Google's rankings of Google content within its Top 3 and 5 results, failing to include Google content that Google ranks similarly in only a little more than a third of queries.

A Closer Look at Google vs Bing

On E&L's own terms, Bing results are more biased than Google results; rivals are more likely to agree with Google's algorithmic assessment (than with Bing's) that its own content is relevant to user queries. Bing refers to Microsoft content other engines do not rank at all more often than Google refers its own content without any agreement from rivals. Figures 4 and 5 shows the same data presented above in order to facilitate direct comparisons between Google and Bing.

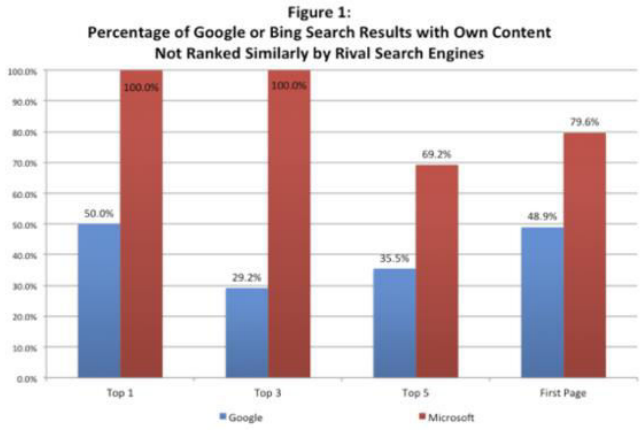

Fig 4: Percentage of Google or Bing Search Result

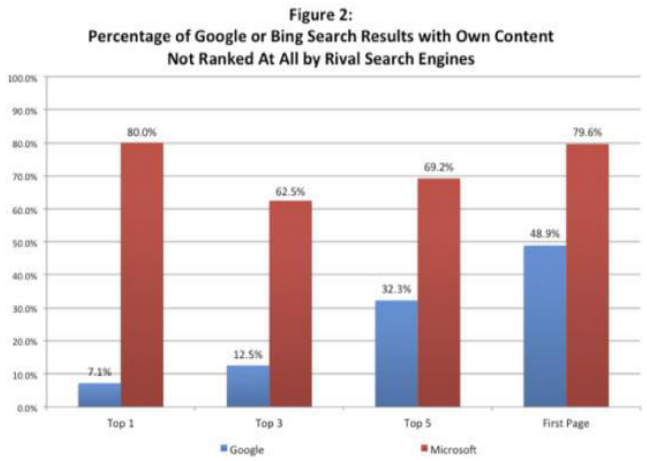

Fig 5: Percentage of Google or Bing Search Result

The Bing search results for these 32 queries are more frequently "biased" in favor of its own content than are Google's. The bias is greatest for the Top 1 and Top 3 search results.This study finds that Bing exhibits far more "bias" than E&L identify in their earlier analysis. For example, in E&L's study, Bing does not refer to Microsoft content at all in its Top 1 or Top 3 results; moreover, Bing refers to Microsoft content within its entire first page 11 times, while Google and Yahoo refer to Microsoft content 8 and 9 times, respectively. Most likely, the significant increase in Bing's "bias" differential is largely a function of Bing's introduction of localized and personalized search results and represents serious competitive efforts on Bing's behalf.

CONCLUSION

I presented firstly basic Semantic Web and Web Usage Mining notions. Then, we discussed about the application of techniques coming from the new emerging area of Semantic Web Mining in the domain of e-Learning systems and analyzed the significant role of Ontologies. We expounded and argued about our proposed approach for producing recommendations to users in a given e-Learning corpus. Finally, we concluded with the description of the recommendation engine's operation and presented an algorithm for making effective recommendations.

As shown in the paper, the proposed personalization scenario tries to integrate the Semantic Web vision by using Ontologies with Using Mining techniques in order to better service the needs and the requirements of learners. We strongly believe that the combination of domain's ontology and frequent item sets, which include all the information about users' navigational attitude, enhances the whole process and produces better recommendations. The system first finds an initial recommendation set and then uses the frequent item sets to enrich it, taking into consideration other users' navigational activity.

References:

[1] Z. Dou, R. Song, and J.-R. Wen, *"A Large-Scale Evaluation and Analysis of Personalized Search Strategies,"* Proc. Int'l Conf. World Wide Web (WWW), pp. 581-590, 2007.

[2] J. Teevan, S.T. Dumais, and E. Horvitz, *"Personalizing Search via Automated Analysis of Interests and Activities,"* Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR), pp. 449-456, 2005.

[3] M. Spertta and S. Gach, *"Personalizing Search Based on User Search Histories,"* Proc. IEEE/WIC/ACM Int'l Conf. Web Intelligence (WI), 2005.

[4] B. Tan, X. Shen, and C. Zhai, *"Mining Long-Term Search History to Improve Search Accuracy,"* Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD), 2006.

[5] K. Sugiyama, K. Hatano, and M. Yoshikawa, *"Adaptive Web Search Based on User Profile Constructed without any Effort from Users,"* Proc. 13th Int'l Conf. World Wide Web (WWW), 2004.

[6] X. Shen, B. Tan, and C. Zhai, *"Implicit User Modeling for Personalized Search,"* Proc. 14th ACM Int'l Conf. Information and Knowledge Management (CIKM), 2005.

[7] X. Shen, B. Tan, and C. Zhai, *"Context-Sensitive Information Retrieval Using Implicit Feedback,"* Proc. 28th Ann. Int'l ACM SIGIR Conf. Research and Development Information Retrieval (SIGIR), 2005.

[8] F. Qiu and J. Cho, *"Automatic Identification of User Interest for Personalized Search,"* Proc. 15th Int'l Conf. World Wide Web (WWW), pp. 727-736, 2006.

[9] J. Pitkow, H. Schu¨ tze, T. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, and T. Breuel, *"Personalized Search,"* Comm. ACM, vol. 45, no. 9, pp. 50-55, 2002.

[10] Y. Xu, K. Wang, B. Zhang, and Z. Chen, *"Privacy-Enhancing Personalized Web Search,"* Proc. 16th Int'l Conf. World Wide Web (WWW), pp. 591-600, 2007.

[11] A. Krause and E. Horvitz, *"A Utility-Theoretic Approach to Privacy in Online Services,"* J. Artificial Intelligence Research, vol. 39, pp. 633-662, 2010.

[12] J.S. Breese, D. Heckerman, and C.M. Kadie, *"Empirical Analysis of Predictive Algorithms for Collaborative Filtering,"* Proc. 14th Conf. Uncertainty in Artificial Intelligence (UAI), pp. 43-52, 1998.