A Survey on Domain Specific Search Engine

Shweta Redkar, Norman Dias & J. A. Laxminarayana

Goa College of Engineering, Farmagudi ,Ponda-Goa.
E-mail : shwetaredkar@gmail.com, diasnorman1@gmail.com & jal@gec.ac.in

***Abstract* - Domain-specific search engines are becoming increasingly popular because they offer increased accuracy and extra features not possible with general, Web-wide search engines. The paper focuses on improving domain specific search engine which is becoming more popular as compared to Web-Wide Search Engines as they are difficult and time consuming to maintain. At the same time, they are unable to provide sufficient relevant documents to represent the target text. We discuss research in reinforcement learning, hierarchical clustering and information extraction that enables efficient spidering, populates topic hierarchies, and identifies informative text segments. Topic wise hierarchy of text segment is made using hierarchical clustering. The proposed approach of generating topic hierarchies for text patterns, which provide a basis for the in-depth analysis of text patterns on a larger scale, can benefit many information systems.**

***Index Terms— Domain Specific Search Engine, Hierarchical clustering, Agglomerative clustering, Topic Hierarchy, Reinforcement Learning, Text Mining and Information Extraction*.**

I. INTRODUCTION

The Web has become a very rich source of information for almost any field, ranging from music to histories, from sports to movies, from science to culture, and many more. However, it has become increasingly difficult to search for desired information on the Web. Users are facing the problem of information overload [1], in which a search on a general-purpose search engine such as Google (www. google.com) results in thousands of hits.

Because a user cannot specify a search domain (e.g. medicine, music), a search query may bring up Web pages both within and outside the desired domain. For example, a user searching for "cancer" may get Web pages related to the disease as well as those related to the Zodiac sign. As a result, the user has to browse through the list of results to identify relevant Web pages, a task which requires significant mental effort. Directory services such as Yahoo! (www.yahoo.com) provide users with a hierarchy of classified topics. While the precision is high, recall rate suffers as each page included in the results has to be manually evaluated. When we know that we want information of a certain type, or on a certain topic, a domain specific search engine can be a powerful tool [2].

Every search engine must begin with a collection of documents to index. When aiming to populate a domain-specific search engine, a web-crawling spider need not explore the Web indiscriminantly, but should explore in a directed fashion to find domain-relevant documents efficiently [2]. We frame the spidering task in a reinforcement learning framework [9], allowing us to mathematically define "optimal behaviour".

Creation of topic hierarchy with hierarchical clustering considers only those web pages as that contains the user's input query topic, not all web pages. This eliminates the problem of finding positive examples and enables us to make domain-specific search engines at low cost.Extracting topic-relevant pieces of information from the documents of a domain-specific search engine allows the user to search over these features in a way that general search engines cannot. Information extraction, the process of automatically finding specific textual substrings in a document.

The paper discusses the methods to automate many aspects of creating and maintaining domain specific search engines by using machine learning techniques. These techniques allow search engines to be created quickly with minimal effort, and are suited for re-use across many domains. The paper also investigates a machine learning methods for efficient topic-directed spidering that is building a browsable topic hierarchy,

and extracting topic-relevant substrings. At the same time the paper also addresses the problem of generating topic hierarchies for diverse text segments, and presents a practical approach that deals with the problem using the Web as an additional knowledge source.

II. SPIDERING AS REINFORCEMENT LEARNING

In machine learning, the term "reinforcement learning" refers to a framework for learning optimal decision making from rewards or punishment [9]. It differs from supervised learning in that the learner is never told the correct action for a particular state, but is simply told how good or bad the selected action was, expressed in the form of a scalar "reward."

A task is defined by a set of states, $s \in S$, a set of actions, $a \in A$, a state-action transition function, $T : S \times A \rightarrow S$, and a reward function, $R : S \times A \rightarrow \Re$. At each time step, the learner (also called the agent) selects an action, and then as a result is given a reward and its new state. The goal of reinforcement learning is to learn a policy, a mapping from states to actions, $\pi : S \rightarrow A$, that maximizes the sum of its reward over time.

As an aid to understanding how reinforcement learning relates to spidering, consider the common reinforcement learning task of a mouse exploring a maze to find several pieces of cheese. The agent's actions are moving among the grid squares of the maze. The agent receives a reward for finding each piece of cheese. The state is the position of the mouse and the locations of the cheese pieces remaining to be consumed (since the cheese can only be consumed and provide reward once). Note that the agent only receives immediate reward for finding a maze square containing cheese, but that in order to act optimally it must choose actions considering future rewards as well[2].

In the spidering task, the on-topic documents are immediate rewards, like the pieces of cheese. An action is following a particular hyperlink. The state is the set of on-topic documents remaining to be consumed, and the set of hyperlinks that have been discovered. The key feature of topic-specific spidering that makes reinforcement learning the proper framework is that the environment presents situations with delayed reward. The problem now is how to practically apply reinforcement learning to spidering. The state-space is enormous and does not allow the spider to generalize to hyperlinks that it has not already seen [3].

Reinforcement Learning is highly effective framework for the spidering problem. In both the data sets, the reinforcement learning outperforms the traditional spidering with the breadth first search by factor of three or more [8] .

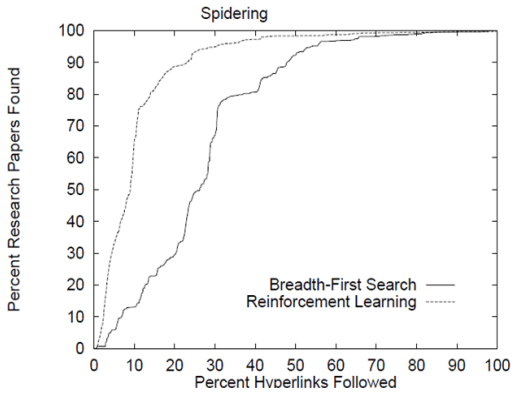

Fig 1. Performance of reinforcement learning Spidering versus traditional Breadth First Search

III. PROPOSED ARCHITECTURE

The proposed work is to organize text patterns in a form of topic hierarchies discovered by mining the search-result pages from the Web. User submits the query as a input to the domain search engine. The engine extracts topic from the query and do the topic mining on the input query in the form of text segments.

A. Text Mining

Text databases are rapidly growing due to the increasing amount of information available in electronic form, such as electronic publications, various kinds of electronic documents, e-mail, and the World Wide Web. Nowadays most of the information in government, industry, business, and other institutions are stored electronically, in the form of text databases [4].

Data stored in most text databases are semi structured data in that they are neither completely unstructured nor completely structured. For example, a document may contain a few structured fields, such as title, authors, publication date, and category, and so on, but also contain some largely unstructured text components, such as abstract and contents. There have been a great deal of studies on the modeling and implementation of semi structured data in recent database research. Moreover, information retrieval techniques, such as text indexing methods, have been developed to handle unstructured documents [4].

Traditional information retrieval techniques become inadequate for the increasingly vast amounts of text data. Typically, only a small fraction of the many available documents will be relevant to a given individual user. Without knowing what could be in the documents, it is difficult to formulate effective queries for analyzing and extracting useful information from the data. Users need tools to compare different documents, rank the importance and relevance of the documents, or find patterns and trends across multiple documents.

Thus, text mining has become an increasingly popular and essential theme in data mining [4].

B. Hierarchical Clustering

A hierarchical clustering method works by grouping data objects into a tree of clusters. Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or top-down (splitting) fashion. The quality of a pure hierarchical clustering method suffers from its inability to perform adjustment once a merge or split decision has been executed. That is, if a particular merge or split decision later turns out to have been a poor choice, the method cannot backtrack and correct it [4].

The broad and shallow multi-way-tree representation, instead of the narrow and deep binary-tree one, is believed more suitable for humans to browse, interpret, and do deeper analysis. The proposed model use the Hierarchical Agglomerative Clustering algorithm (HAC) which is nothing but a bottom-up strategy that starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied to produce a natural and comprehensive multi-way-tree hierarchy [4].

This hierarchical clustering technique is combined with min-max partitioning to generate a natural and comprehensive multi-way cluster hierarchy tree. The proposed approach produces a natural and comprehensive hierarchical tree where each category contains an appropriate number of sub-categories and so on. This broad and shallow multi-way-tree representation, instead of the narrow and deep binary-tree representation, is easier and more suitable for humans to browse, interpret, and do deeper analysis.

To generate a multi-way-tree hierarchy from a binary-tree representation, a top-down approach is used to decompose the hierarchy into several sub-hierarchies first, and to then recursively apply the same decomposing procedure to each sub-hierarchy. The key idea is to determine a suitable level at which to cut the binary-tree hierarchy and create the most appropriate sub-hierarchies; that is, these sub-hierarchies are with the best quality and number preference over those produced by cutting at the other levels. Through recursively decomposing the sub-hierarchies, a new multi-way-tree hierarchy can be constructed.

IV. TOPIC HIERARCHY

The proposed method is to build a domain-specific web search engine, that consider only those web pages that contain the user's input query keywords; not all web pages [24]. The problem is to find that the topic from the text segment that provide enough generalization to handle all future user topics then generate a natural and comprehensive cluster hierarchy form topics generated from inputs text segments where each category also contains an appropriate number of sub-categories and so on.

This broad and shallow multi-way-tree representation, instead of the narrow and deep binary-tree one, is believed easier and more suitable for humans to browse, interpret, and do deeper analysis. For this purpose, the proposed model uses a Hierarchical Agglomerative Clustering to generate topic clusters. The algorithm consists of two phases: Hierarchical Agglomerative Clustering -based clustering to construct a binary-tree cluster hierarchy and min-max partitioning to generate a natural and comprehensive multi-way-tree hierarchy structure from the binary-tree one.

A. Agglomerative Clustering

In the Hierarchical Agglomerative Clustering process, at each iteration step, the two most-similar clusters are merged to form a new one, and the whole process halts when there exists only one unmerged cluster. Hierarchical Agglomerative Clustering builds a binary-tree cluster hierarchy in a bottom-up fashion [5]. Let v1, v2, ……, vn be the input object vectors, and let C1, C2, . . ., Cn be the corresponding singleton clusters.

Let Cn+i be the new cluster created at the i^{th} step. The output binary-tree hierarchy can be expressed as a list, C1, . . . , Cn, Cn+1, . . . , C2n-1, with two functions, left(Cn+i) and right(Cn+i),

$1 \leq i < n$, indicating the left and right children of the internal cluster node Cn+I, respectively. Hierarchical Agglomerative Clustering algorithm is a specific function used to measure the similarity between any pair of clusters Ci and Cj. There are four well-known inter-cluster similarity , Sim functions:

(SL) the single-linkage function, defined as the largest similarity between two objects in both clusters:

$$Sim_{SL}(Ci, Cj) = \max Sim(va, vb)$$

$va \in C1, vb \in Cj$

(CL) the complete-linkage function, defined as the smallest similarity between two objects in both clusters:

$Sim_{CL}(Ci, Cj) = \min Sim(va, vb)$

$va \in C1, vb \in Cj$

(AL) the average-linkage function, defined as the average of all similarities among the objects in both clusters:

$Sim_{AL}(Ci, Cj) = Sim_{A}(Ci, Cj)$

(CE) the centroid function, defined as the similarity between the centroids of the two clusters:

$Sim_{CE}(Ci, Cj) = Sim(ci, cj)$

where ci and cj are the centroids of Ci and Cj , respectively, and, for a cluster Cl, the k^{th} feature weight of its centroid, Cl, is defined as:

$c_{l,k} = \Sigma\, v_{i,k} / \mid C_l \mid$

$vi \in C_l$

Usually, the clusters produced by the single-linkage method are isolated but not cohesive, and there may be some undesirably elongated clusters.

Min-Max Partitioning

To generate a multi-way-tree hierarchy from a binary tree, a top-down partitioning approach to first decompose the hierarchy into several sub-hierarchies, and then recursively apply the same decomposing procedure to each sub-hierarchy. Let the level between {Cn+i-1, Cn+i} be n−I [24].

Requirement of "natural" clusters is that they must be cohesive and isolated from the other clusters. The criterion for determining a proper cut level is to heuristically satisfy this requirement.

Let the inter-similarity between two clusters Ci and Cj be defined as the average of all pair wise similarities among the objects in Ci and Cj , i.e., $sim_A(Ci, Cj)$, and let the intra similarity within a cluster Ci be defined as the average of all pair wise similarities within Ci, i.e., $sim_A(Ci, Cj)$.

The partitioning approach *[24]* ,finds a particular level that minimizes the inter-similarities among the clusters produced at the level and maximizes the intra-similarities of all those clusters; that is why the approach is named as min-max partitioning [6]. Let C be a set of clusters; our quality measurement of C based on its cohesion and isolation is defined as:

$$Q(C) = \frac{1}{|C|} \sum_{C_I \in C} \frac{Sim_A\left(C_i, \bar{C}_i\right)}{Sim_A(C_i, C_i)}$$

Where

$$\overline{C_i} = \bigcup_{k \neq i} C_k$$

is the compliment of Ci. Note that the smaller the Q(C) value is, the better the quality of the given set of clusters, C, is.

Finally, to partition the given binary-tree hierarchy, the best cut level is chosen as the level l with the minimum Q(LC(l))/N(LC(l)) value. To name a cluster is a rather intellectual and challenging work. It is not easy to determine a name for a cluster. There exist different methods to accomplish this task. In proposed model, the most-frequent co-occurred feature terms from the composed instances or the text segment is the name of the cluster.

V. INFORMATION EXTRACTION

Information extraction is concerned with identifying phrases of interest in textual data. For many applications, extracting items such as names, places, events, dates, and prices is a powerful way to summarize the information relevant to a user's needs.[7] In the case of a search engine over research papers, the automatic extraction of informative text segments can be used to (1) allow searches over specific fields, (2) provide useful effective presentation of search results (*e.g.* showing title in bold), and (3) match references to papers. In the case of a domain-specific search engine, the automatic identification of important information can increase the accuracy and efficiency of a directed search.

VI. RELATED WORK

Search engines usually use spiders (also referred to as Web robots, crawlers, worms, or wanderers) to retrieve pages from the Web by recursively following URL links in pages using standard HTTP protocols. The following methods are commonly used to locate Web pages relevant to a particular domain: The spiders can be restricted to stay in particular Web domains, because many Web domains have specialized contents [12,14]. Some spiders are restricted to collect only pages at most a fixed number of links away from the starting URLs starting domains [12,13]. More sophisticated spiders analyze Web pages and hyperlinks to decide what documents should be downloaded [3]. These methods have different levels of performance in efficiency and effectiveness, but in most cases the resulting collection is still noisy and needs further processing. Filtering programs are needed to filter irrelevant and low-quality pages from the collection to be used in the vertical search engine. For search engines that employ filtering, the techniques used include: Domain experts manually

determine the relevance of each Web page (e.g., Yahoo). In the simplest automatic way, the relevance of a Web page can be determined by the occurrences o particular keywords (e.g., computer) [11]. TF*IDF (term frequency * inverse document frequency) is calculated based on domain-expert created lexicon. Web pages are compared with a set of relevant documents, and those with a similarity score above a certain threshold are considered relevant [15].Text classification techniques, such as Naive Bayesian classifier, also have been applied to Web page filtering [3].

Text classification is the study of classifying textual documents into predefined categories. The topic has been extensively studied at SIGIR conferences and evaluated on standard testbeds. There are a few major approaches. For example, the Naïve Bayesian method has been widely used [17,3]. It uses the joint probabilities of words and categories to estimate the probabilities of categories given a document. Documents with a probability above a certain threshold are considered relevant. The k-nearest neighbor method is another widely used approach in text classification. For a given document, the k neighbours that are most similar to a given document are first identified [19,20].

Feedforward/backpropagation neural network was usually used [21]. Term frequencies or TF*IDF of the terms are used as the input to the network. Based on learning examples, the network will be trained to predict the category of a document. Another new technique used in text classification is called support vector machine (SVM), a statistical method that tries to find a hyperplane that best separates two classes [18]. Joachims first applied SVM in text classification problem [22]. It has been shown that SVM achieved the best performance on the Reuters-21578 data set [23].

It is not easy to build these search engines. There are two major challenges to building vertical search engines ,locating relevant documents from the Web and Filtering irrelevant documents from a collection. This study tries to address these issues and propose new approaches to the problems. Aimed at combining different Web content and structure analysis techniques to build spider programs for vertical search engines, developed and compared three versions of Web spiders, namely, Breadth-First Search (BFS) Spider, PageRank Spider, and Hopfield Net Spider.[10]

VII. CONCLUSION

Vertical search engine is a fast emerging technology, giving serious competitions to generic search engines. Unsupervised learning can automatically create a topic hierarchy and generate keywords.. Although clustering text segments is in essence considered very difficult, with huge amounts of on-line documents indexed by search engines, most of text segments can get adequate topic relevant contextual information. Also, a clustering algorithm for generating a natural multi-way-tree cluster hierarchy is developed. In this paper , we surveyed various techniques to develop domain specific search engines.The approach was also proven useful in various Web information applications.

VIII. REFERENCES

[1] C. M. Bowman, P. B. Danzig , U.Manber, and F. Schwartz Scalable Internet Resource Discovery: Research Problems and Approaches, Communications of the ACM, 37(8),1994.

[2] A. McCallum , K. Nigam , J. Rennie, , K.Seymorey Building Domain-Specific Search Engines with Machine Learning Techniques.

[3] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, "A machine learning approach to building domain-specific search engines," IJCAI-99, p. 662–667, 1999.

[4] Jiawei Han and Micheline Kamber, "Data Mining: Concepts and Techniques, Second Edition," Issue Date: 2006.

[5] B. Mirkin, "Mathematical Classification and Clustering," Kluwer, 1996.

[6] C. Ding, X. He, H. Zha, M. Gu, and H. Simon, "A min-max" cut algorithm for graph partitioning and data clustering," In Proceedings of ICDM'01, pages 107–114, 2001.

[7] J. Rennie , A. McCallum Using Reinforcement Learning to Spider the Web efficiently.

[8] J. Rennie, Andrew Kachites McCallum –Using Reinforcement Learning to Spider the Web effifiecntly, Proceedings of the 16th International Conference on Machine Learning (ICML) , 1999.

[9] Kaelbling L. P.: Littman, M. L. and Moore , A. W. 1996.Reinforcement Learning : A survey. Journal of Artificial Intelligence Research 237-285.

[10] Michael Chau, "Spidering and Filtering Web Pages for Vertical Search Engines" 2002.

[11] Cho, J., Garcia-Molina, H., and Page, L. "Efficient Crawling through URL Ordering," in Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, Apr 1998.

[12] Manber, U., Smith, M., and Gopal, B. "WebGlimpse: Combining Browsing and Searching," in Proceedings of the USENIX 1997

Annual Technical Conference, Anaheim, California, Jan 1997.

[13] Sumner, R. G., Jr., Yang, K., and Dempsey, B. J. "An Interactive WWW Search Engine for User-defined Collections," in Proceedings of the 3rd ACM Conference on Digital Libraries, Pittsburgh, Pennsylvania, USA, Jun 1998, pp. 307-308.

[14] Witten, I. H., Bainbridge, D., and Boddie, S. J. "Greenstone: Open-source DL Software," Communications of the ACM, 44(5),pp. 47.

[15] Baujard, O., Baujard, V., Aurel, S., Boyer, C., and Appel, R. D. "Trends in Medical Information Retrieval on the Internet,"Computers in Biology and Medicine, 28, 1998, pp. 589-601.

[16] Chakrabarti, S., Dom, B., and Indyk, P. "Enhanced Hypertext Categorization Using Hyperlink," in Proceedings of ACM SIGMOD International Conference on Management of Data, Seattle, Washington, USA, Jun 1998.

[17] Koller, D. and Sahami, M. "Hierarchically Classifying Documents Using Very Few Words," in Proceedings of the 14th International Conference on Machine Learning (ICML'97), 1997, pp. 170-178.

[18] Vapnik, V. Statistical Learning Theory, Wiley, Chichester, GB, 1998.

[19] Masand, B., Linoff, G., and Waltz, D. "Classifying News Stories Using Memory Based Reasoning," in Proceeedings of the

15th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'92), pp. 59-64.

[20] Iwayama, M. and Tokunaga, T. "Cluster-based Text Categorization: A Comparison of Category Search Strategies," in Proceedings of the 18th Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'95), pp. 273-281.

[21] Wiener, E., Pedersen, J. O., and Weigend, A. S. "A Neural Network Approach to Topic Spotting," in Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR'95), 1995.

[22] Joachims, T. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," in Proceedings of the European Conference on Machine Learning, Berlin, 1998, pp. 137-142.

[23] Yang, Y. and Liu, X. "A Re-examination of Text Categorization Methods, in Proceedings of the 22nd Annual International ACM Conference on Research and Development in Information Retrieval (SIGIR'99), 1999, pp. 42-49.

[24] S. Kakkar, N. Kumar "Creating Topic Hierarchy with Clustering in Domain Specific Search Engines" International Journal of Engineering Science and Advanced Technology , ISSN: 2250–3676 Volume-2, Issue-2, 358 – 364.

◈◈◈