International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:8, 2007

Effective Keyword and Similarity Thresholds for the Discovery of Themes from the User Web Access Patterns

Haider A Ramadhan, and Khalil Shihab

Abstract**—**Clustering techniques have been used by many intelligent software agents to group similar access patterns of the Web users into high level themes which express users intentions and interests. However, such techniques have been mostly focusing on one salient feature of the Web document visited by the user, namely the extracted keywords. The major aim of these techniques is to come up with an optimal threshold for the number of keywords needed to produce more focused themes. In this paper we focus on both keyword and similarity thresholds to generate themes with concentrated themes, and hence build a more sound model of the user behavior. The purpose of this paper is two fold: use distance based clustering methods to recognize overall themes from the Proxy log file, and suggest an efficient cut off levels for the keyword and similarity thresholds which tend to produce more optimal clusters with better focus and efficient size**.**

Keywords**—**Data mining, knowledge discovery, clustering, data analysis, Web log analysis, theme based searching.

I. Introduction

SEVERAL prototype systems were developed in this area, which include WebWatcher [1] DiffAgent [2], Alexa [3] and Letizia [4]. However, techniques followed by these systems, tough novel, are considered primitive and fail to construct comprehensive models of the user profiles. For example, WebWatcher analyzes hyperlinks in the pages visited by the users and then recommends those links which the system guesses are promising in matching the goal of the session. Letizia attempts to infer user intentions by tracking his/her browsing behavior. Links found on the pages visited by the user are automatically explored by the system and are presented to the user on demand. Hence, the main goal here is to perform some degree of automatic Web exploration by anticipating future page accesses. Obviously, a more solid approach is needed to build the user model which can spell out various access patterns of the user. The impetus for the work reported in this paper came from our need for a complete user profile which would allow us to design a fully automatic Web navigation system and a theme based search engine.

The aim of the former system is to recognize a set of pages which are of high interest to the user and then automatically retrieve such pages whenever a change or update is discovered in them. Some work has already been reported in this area which captures the pages or the user interests explicitly by asking the users to provide the URLs [6]. Next the system fetches these pages and constructs a template for each page. The system periodically fetches the pages in the background, constructs the templates, matches them with the initial templates stored in the database, and notifies the users when a change in the templates is discovered. Although being a genuine improvement, explicitly capturing the user intentions may not be an efficient way to implement such important tools, an implicit way to achieve the same is needed. The aim of the theme based searching is to analyze user access patterns, cluster them into groups representing themes or topics, and have them fed into a theme based search engine which would focus on retrieving pages highly relevant to the themes and would avoid pages which are not relevant to the user topics. The purpose of this paper is to find out the optimal keyword and similarity thresholds needed to come up with more focused themes through using clustering techniques.

II. Log File Processing

For discovering users access patterns, two approaches have been suggested. The first approach [15] attempts to capture the browsing movement, forward and backward, between Web pages in a directed graph called *Traversal Oath Graph*. In this approach, a set of *maximal forward references* which represent different browsing sessions are first extracted from the directed graphs. By using *association rules*, the frequently traversed paths can be discovered. These paths represent most common traversal patterns of the user. In the second approach, user access logs are examined to discover clusters of similar pages which represent categories of common access patterns. The task of discovering user access patterns and clustering them into themes is a three-phase process. The input to the process is the user access log saved on the Web proxy server. The log file contains records for each user accessing the Web. Each record in the file represents a page request by user client machine [6,7]. In summary, the purpose of phase one is to clean up the log file and get it converted into a vector form. Next, Generalization is used to consolidate all related URLs into their main home page URL. Frequency of visits and the

Haider Ramadhan and Khalil Shihab are with Sultan Qaboos University, PO Box 36 Muscat 123, Oman (phone: 968-24415407; fax: 00968-24413415; e-mail: haiderr@squ.edu.om).

updated total time spent are also counted and added to the vector.

In phase two, the TFIDF (Term Frequency/Inverse Document Frequency [8]) algorithm is used to extract keywords from the documents. Since TFIDF normally computes the weights for the words as well, some extra pre-processing was performed to strip the weights from the words. These keywords are taken from the title tag, keyword tags, header tags, meta tags, and emphasized words. According to the threshold used in the experiment, a certain number of keywords are extracted and added to the initial vector produced in the previous phase. Total time spent and the visit frequency are the two measures we use to prioritize the words in the vector. The last phase of the discovery process is to produce the topics of interests from the term vectors. A distance based clustering technique is used to form the topics. The output is a small number of topic vectors representing themes. Each vector contains a predefined number of keywords adjusted in the order according to the time spent and the number of visits.

III. Clustering Process

Many intelligent software agents have used clustering techniques in order to retrieve, filter, and categorize documents available on the World Wide Web. Traditional clustering algorithms either use a priori knowledge of document structures to define a distance or similarity among these documents or use probabilistic techniques, e.g. Bayesian classification. These clustering techniques use a selected set of words (features) appearing in different documents as the dimensions. Each such feature vector, representing a document, can be viewed as a point in this multi-dimensional space [9]. New clustering algorithms that can effectively cluster documents, even in the presence of a very high dimensional feature space, have recently been reported. These clustering techniques, which are based on generalizations of graph partitioning, do not require pre-specified ad hoc distance functions, and are capable of automatically discovering document similarities or associations.

TABLE I
Total and Size of Clusters

	ST = 1				ST = 2				ST = 3				ST = 4				ST = 5			
KT	5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30
Total clusters	53	39	26	19	75	61	53	44	112	83	61	30	164	151	109	79	189	158	119	62
Average	34.2				58.3				71.5				125.7				132			

Clustering in a multi-dimensional space using traditional distance or probability-based methods has several drawbacks [10]. First, it is not trivial to define a distance measure in this space. Some words are more frequent in a document than others. Taking only the frequency of the keyword occurrence is not enough as some documents are larger than others. Furthermore, some words may occur more frequently across documents. Second, the number of all the words in all the documents can be very large. Distance-based schemes [11], such as k-means analysis, hierarchical clustering and nearest neighbor clustering, generally require the calculation of the mean of document clusters. Similarly, probabilistic methods such as Bayesian classification used in AutoClass [12] do not perform well when the size of the feature space is much larger than the size of the sample set. However, in the research reported here we do not have a variable length of keywords among documents. The keyword threshold is set fixed for every experiment. In addition, the number of all words in documents is not considered as a criteria for feature selection in our experiments. Hence, it was felt that the distance based clustering would fit our need and would not need to deal with the drawbacks mentioned above. We use a version of the Nearest Neighbor Algorithm [13] with an ad hoc distance similarity metrics. A total of 218 web pages were retrieved and grouped into four broad learning categories: news, business, finance, and economics. These pages correspond to the clustered vectors. The retrieved pages were downloaded, labeled, and archived. The labeling allowed us to easily calculate an entropy (discussed shortly). Subsequent references to any page were directed to the archive. This ensured a stable data sample since some pages are fairly dynamic in content. A total of five experiments were conducted. Documents were clustered using the Nearest Neighbor Algorithm (NNA) referenced earlier.

IV. The Evaluation

Only two methods of feature selection were used, namely Keyword Threshold (KT) and Similarity Threshold (ST). The KT refers to the number of words extracted from upper portions of the pages and ranged from 5, 10, 20 and 30 words. The ST ranged from 1 to 5, and was used as a measure to compare the similarity among generated clusters and to consolidate them when a given ST is satisfied. For example, with KT is set to 5 and ST to 3, only 5 keywords are used from each page and those clusters having at least 3 keywords in common are consolidated to form a single cluster.

Our objective is to find the correlation between KT and ST, and their influence on the maximum and mean sizes of the clusters produced. We also aimed at finding out total number of clusters produced across various values of KT and ST. Traditionally, it has been reported that smaller ST values tend to produce few but large clusters with less focus as far as topics are concerned, while large ST values tend to generate large number of clusters which are smaller in size and better in focus The entropy based analysis [9] was used to assess how focused the clusters are in relation to the four broad classes of the categories mentioned above. When a cluster, for example, contains documents from one category only, the entropy value is 0 for the cluster, and when a cluster contains documents from several categories the entropy value of the cluster becomes higher. Hence, lower entropy values tend to suggest more focused clusters in their topics and vice versa. The total entropy is the average entropies of the clusters. This is attributed to the fact that a smaller ST value is expected to make clusters get consolidated (combined) at a higher rate since having few words in common among clusters is more typical than having large number of words in common among clusters. As a result, it would be safe to *hypothesize* that high entropy values would be associated with lower ST values, while large ST values would be related to lower entropy values. We compare the results of the five experiments by comparing their entropies across various feature selection criteria mentioned above (i.e. ST and KT values). Table I shows the relationship between various KT and ST values. The number of clusters tends to increase when the threshold values are near the end of the test range. To show the percentage of overall increase in the number of clusters that is associated with the increase in the ST values, Table I shows the weighted increase in the number of clusters across all KT values for some ST value.

TABLE II
MAXIMUM SIZE OF CLUSTERS

	ST = 1				ST = 2				ST = 3				ST = 4				ST = 5			
KT	5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30	5	10	20	30
Max Size	65	92	123	205	52	85	111	173	39	68	106	148	19	41	87	130	5	31	79	118

It is noticed that the weighted increase in the number of clusters is steady when ST values increase from 1 to 2. With ST=1, the average number of clusters is 34.2 for all KT values. When ST becomes 2, the average number of clusters produced is 58.3, an increase of 24.1 (70%) over ST=1. When ST becomes 3, average number of clusters produced is 71.5, an increase of 13.2 (23%) over ST=2. With ST=4, the average increase is noticed to be 54.2 (76%) over ST=3. Finally with ST=5, the average increase is 6.3 (5%) over ST=4. Three main observations can be stated here. First, the number of clusters produced tends to increase across all KT values as the ST values increases. Second, this increase is not at the same pace for different KT values. It is noticed that for any ST value, number of clusters tend to be high for smaller KT values and tend to decrease as the KT values increase. Hence, this clearly shows the inverse relationship between ST and KT values. Third, largest average increase in the number of clusters was noticed to be for ST=4 (76%). With ST=5, the average increase drastically dropped to only 5%. This may imply that the similarity threshold value of 4 is the cut off value we seek which tends to produce optimal or semi optimal number of clusters that maintain good focus. Table II provides some insights into the maximum size of the clusters produced. Few observations can be made here. First, maximum size of the clusters across all KT values tends to decrease as ST values increase. For example, the decrease in the maximum cluster size from ST=1 to ST=5 for KT=5 is from 65 to 5, hence a reduction of 92%. The reduction, as shown in Table II, for KT=10 is 67%, for KT=20 is 36%, and for KT=30 is 42%. It may be stated that the level of reduction becomes more steady when KT=20, since the next reduction at KT=30 (6%) is not as big as the reduction from KT=10 to KT=20 which is 31%. Although cautiously, it could be argued that the keyword thresholds of 10 and 20 along with the similarity threshold of 4 are the cut off values that tend to be recommended by the above results. Very few studies reported on the recommended combination of both threshold values. However, it has been found that KT values of 5 and 20 tend to work best using other distance based clustering methods such as Autoclass and, HAC [12], and non-distance based methods such as Principal Component Clustering (PCA) [14]. As a consequence, it is safe to state that the Nearest Neighboring Algorithm used in this study did not deviate from the path reported by others. Fig. 1 deals with using the entropy-based analysis to get some insights into the focus of the clusters produced in relation to various threshold values and the four categories mentioned earlier.

As stated before, when a cluster, contains documents from one category only, the entropy value is 0 for the cluster, and when a cluster contains documents from several categories the entropy value of the cluster becomes higher. Hence, lower entropy values tend to suggest more focused clusters in their topics and vice versa. The total entropy used in the figure is the average entropies of all the clusters. Few observations can be noted from the figure. First, lower ST and higher KT values tend to generate clusters with higher entropies, hence implying that such clusters are very general in their topics. With high ST values, clusters tend to be small in their size mostly contain keywords from documents which come from a certain class. This observation seems to prevail even when KT values are 20 and 30. However, higher KT values still tend to generate clusters with high entropy values, implying that they contain keywords from documents belonging to more than one class, hence are of less focus in their topics. The figure also shows that the gap between entropy values is more evident when KT values change from 10 to 20, and that the gap between 20 and 30 is not as the former one.

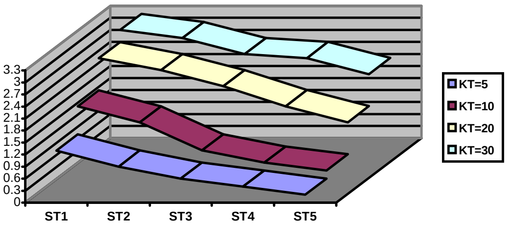

Fig. 1 Entropy comparison

We can cautiously state that the threshold values of ST=4 and KT=5 or 10 represent the best combination of the thresholds which produce clusters with lower entropy values and hence with better focus. The study used four closely related classes, namely news, business, finance, and economy. It is possible that classes with less relevance could produce different results since un-related classes tend to have fewer keywords common among their documents. Therefore, smaller clusters with better focused keywords should be expected. It is hoped that such claim could also be formally verified. Similar results have also been reported but with two main differences [14]. First, the reported results considered the focus of the clusters across a range of KT values with no relationship with ST values. Second, the algorithm used in that experiment for clustering was a non-distance based one. In that experiment, it was found that the method, i.e. the PCA algorithm referenced earlier in this paper, worked best with KT values 5 and 20. In our case, KT values 5 and 10 seemed to produce best results. Of course, the quality of the clusters can be better judged by looking at the distribution of class labels among clusters. We hope this task would be completed in the near future.

V. Conclusion

Several advanced future Web based applications would rely on having clear models of the users in relation to their access behavior and patterns. These applications may include automatic navigation and theme based searching. In this paper, we attempted to find out the optimal keyword and similarity thresholds needed to come up with more focused themes from the user log files. Nearest Neighbor Algorithm as a representative of the distance based methods was used to cluster various user access patterns into themes or topics. We also used entropy based analysis to find out the focus of these clusters. The results clearly verified the claim that smaller ST values tend to produce few but large clusters with less focus as far as topics are concerned, while large ST values tend to generate large number of clusters which are smaller in size and better in focus. In short, it was found that the threshold values of ST=4 and KT=5 or 10 represent the best combination of features which produce clusters with lower entropy values and hence with better focus.

References

[1] http://citeseer.nj.nec.com/armstrong97webwatcher.html

[2] Jones D. H. IndustryNet: A model for Commerce on the Web, IEEE Expert, Oct., pp 54-59, 1995.

[3] Willmot D. Alexa, *PC Magazine Online,* January, 1999.

[4] http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia.html

[5] Balabanovic M. and Shoham Fab Y. Content-based collaborative recommendation, *Communications of the ACM,* 40(3): 66-72, 1997.

[6] Tan B. Web information monitoring for competitive intelligence, Cybernetics and Systems, 33, 3, 225-235, 2000.

[7] Srivastava J., Cooley R., Deshpande M., and P.-N. Tan. *Web usage mining: Discovery and applications of usage patterns from web data*. SIGKDD Explorations, 1(2), 2000.

[8] Salton G. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley, Reading, Mass., USA, 1999.

[9] Karypis G. Multilevel hypergraph partitioning: Application in VLSI domain, Proceedings of ACM/IEEE Design Automation Conference, 1997.

[10] Chang C. Customizable multi-engine search tool with clustering. Proceedings of 6th International Web Conference, 1997.

[11] Jain A. Algorithms for Clustering Data. Prentice Hall, 1998.

[12] Titterington D. Statistical Analysis of Finite Mixture Distributions. John Wiley & Sons, 1985.

[13] Lu S. and Fu K. A sentence-to-sentence clustering procedure for pattern analysis. IEEE Transactions on Systems, Man, and Cybernetics, 8, 381-389, 1987.

[14] Moore J. Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering, TR 9405380, Department of Computer Science, University of Minnesota, 2001.

[15] Cheung D. Discovering User Access Patterns on the Web, Knowledge Based Systems, 10, 463-470, 1998.