ISCIT 2004

International Symposium on Communications
and Information Technologies 2004 (ISCIT 2004)
Sapporo, Japan, October 26- 29, 2004

Similarity Score for Information Filtering Thresholds

Jun Lai and Ben Soh
Department of Computer Sciences and Computer Engineering
Latrobe University, Melbourne Australia
Tel: +61-3-9479 1280, Fax: +61-3-9479 3060
E-mail: jun@cs.latrobe.edu.au ben@cs.latrobe.edu.au

Abstract

The rapid growth of the on-line information has led to the development of many techniques for information filtering. The tremendous growth in the amount of information available and the number of visitors to web sites in the recent years poses some key challenges for information filtering and retrieval. Web visitors not only expect high quality and relevant information, but also wish that the information be presented in an as efficient way as possible. The traditional filtering methods, however, only consider the relevant values of document. These conventional methods fail to consider the efficiency of documents retrieval. In this paper, we propose a new algorithm to calculate an index called document similarity score based on elements of the document. Using the index, document profile will be derived. Any documents with the similarity score above a given threshold will be clustered. Using these pre-clustered documents, information filtering and retrieval can be made more efficient. Experimental results clearly show our proposed method tremendously improves the efficiency of information filtering and retrieval.

Keywords: elements, clustering, information filtering, information retrieval, web crawlers, search engine, World Wide Web

1. Introduction

The amount of information in the world is increasing far more quickly than our ability to process it. All of us have known the feeling of being overwhelmed by the number of new books, journal articles, and conference proceedings coming out each year. There is a need for new technologies that can help us sift through all the available information in a more efficient way.

Generally, there are three existing information filtering methods:

- Content-based filtering (also known as cognitive filtering): Here, the system searches for items similar to those the user prefers, based on a comparison of content using text-learning methods. Only the content and properties of a document contribute to the filtering, and each user operates independently. However, this approach has difficulty capturing different types of content and has problem of over-specialization. When the system recommends items scoring highly against user's preferences, the user is restricted to seeing items similar to those already rated.
- Collaborative filtering (also known as social filtering): Here, documents are recommended for a user based on the likes of other users with similar tastes. User profiles are used to compare with each other. Groups of similar profiles are identified and users belonging to one group will be presented the same set of documents. The major drawback of this method is that if the number of users is small, a user whose taste is unusual would not get high quality recommendation.
- Rule-based filtering: It uses demographic or other kind of purposely collected data of users to build user profiles and then define a set of rules to tailor the content delivery based on the facts specified in the user profiles. However, the creation and maintenance of rules are generally manual. As the system gets complicated, there will be difficulties managing it without conflict of logics.

All the above conventional filtering systems consider only the relevance and importance to the users in different ways without due consideration for efficiency. However, as the system gets complicated, the efficiency is also crucial. Surveys have shown that about 85% of Internet users using search engines and search service to find specific information are not

satisfied with the performance of the current generation of search engines. The dissatisfaction arises from the slow retrieval speed, communication delays and poor quality of retrieved results [1].

In this paper, we propose a new efficient method called Elements and Document Profile Based Information Clustering, which can cluster more than two documents, using elements of the documents. The method involves two stages: (i) Restructuring Operation, and (ii) Document Clustering. An algorithm is applied to compute a correlation score of documents. The documents with the similarity score above a given threshold will be clustered together. A definition of document profile is derived. Then, all the documents are clustered, based on the document profile. Our proposed method computes the scores independently of the number of documents.

2. Restructuring Operation

As mentioned in Section 1, the current conventional information filtering methods have mainly been focused on clustering two documents [2]. There has not been much effort on clustering more than two documents. To this end, we propose in this paper a new algorithm so that information filtering can be carried out in a more efficient way.

Currently, there are a number of metadata standards proposed for web pages. Among them are two well-publicized, solid efforts: the Dublin Core Metadata standard and the Warwick frame-word. The Dublin Core is a 15-Element Metadata Element Set proposed to facilitate fast and accurate information retrieval on the Internet [3]. In the paper, we propose a restructuring operation by using keywords appearing in the elements of the documents. Those keywords will be matching the search query when web visitors search on the Internet. Keywords are given different weights ranging from 0 to 1 (inclusive), depending on which element this keyword is in. Using sensitivity analysis, the weights are calibrated by the system designer, based on the importance and relevance of the keyword concerned. For example, the keyword appearing in the element Title should have a higher weight than the one appearing in the element Content.

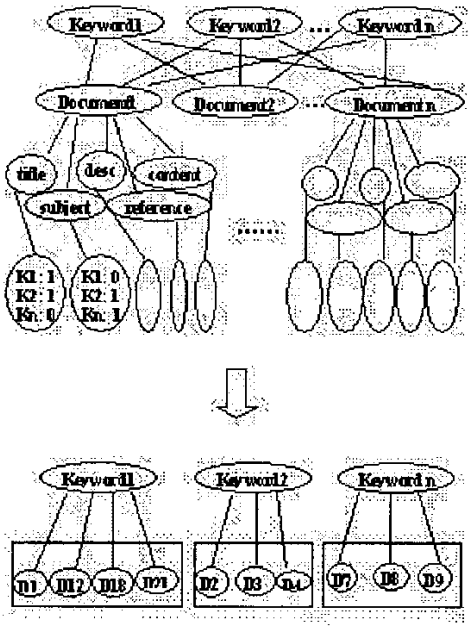

Fig.1 Restructuring operation and clustering of documents

Fig. 1 shows the idea of restructuring operation of documents. On the top level of this tree structure are keywords in relation to some documents, which are on the second level. The third level contains elements of the document. The numbers on the bottom level represent the number of times that keyword appears in that element. The documents are clustered based on the word in the elements of the document after the restructuring operation. (The proposed method will be further discussed in the next section.) In this way, when web visitors enter a keyword to search information, the pre-clustered documents will be utilized in an efficient way. For example, the clusters shown in Figure 1 are: (i) documents 1, 15, 18 and 22, (ii) documents 2, 3 and 4, (iii) documents 7, 8 and 10.

3. Document Clustering

We have proposed a restructuring operation to cluster documents in the previous section. In this section, we will discuss the second stage of our proposed method: the algorithm and technique of documents clustering. The subsections are organized as follows: section 3.1 presents the approach of calculation of similarity score. Then the document profiles will be derived in section 3.2. Finally, the k-mean clustering algorithm [4] will be applied.

3.1 Computation of similarity score

To compute the similarity score of documents, we initially select some keywords appearing in those documents. Each keyword is assigned a weight, ranging form 0 to 1, as mentioned previously. For example, the weight in the element Title should be higher than that in the element Description, while the weight in the element Description should be higher than that in the element content. The number of times that word appearing in the document also affects the value of relevance among documents.

Table 1 shows an example of the number of times a keyword appearing in the elements (namely Title, Subject, Description, Content, and Reference) of a particular document, which is indicated by document ID (d). The similarity score is computed as the sum of all product of keyword weight and the number of times that keyword appears in the document, using the following formula:

$$SS_{(d,K)} = \sum_{j=1}^{n} (W_{Kj} * Count_{Kj}) \qquad (1)$$

where:

- SS is the similarity score of a document based on the keyword K.
- K j is the keyword appearing in the element j of the document $(1 \leq j \leq n)$.
- W $_{Kj}$ is the pre-defined weight of the keyword K $_j$ appearing in the element j, determined by a system administrator via sensitivity analysis.
- Count $_{Kj}$ is the number of times that the keyword K appearing in the element j of the document.

For instance (in Table 1), for the document ID 21, the keyword "filtering" appears in the element Title once, Subject twice, in Description twice, Content 10 times and Reference 3 times. The similarity score of document ID 21 based on the keyword "filtering" is computed as follows:

$SS(21, \textit{filtering}) = (0.99*1+0.8*2+0.7*2+0.5*10+0.4*9) = 12.59$

Table 1. Number of times that keywords appear in the elements of a particular documents

D	21	45	567	789
Title	1	1	0	0
Subject	2	1	0	1
Description	2	7	3	7
Content	10	4	1	4
Reference	9	8	1	7

Table 2. Similarity score of documents

D	21	45	567	789
Title	1	1	0	0
Subject	2	1	0	1
Description	2	7	3	7
Content	10	4	1	4
Reference	9	8	1	7
SS	12.59	11.89	3	10.5

Using formula (1). The similarity score (SS) for all the documents in Table 1 can be calculated and the results are shown in Table 2.

Furthermore, the similarity score of a document based on various keywords can also be computed. Table 3 shows the SS of document ID 21 in terms of keywords "filtering", "clustering", "data" and "item":

3.2 Deriving document profile (DP)

From the calculation of similarity score, the document profile can be derived as follows:

$$DP_{(d)} = \{ (K, SS_{(d,K)} \mid K \in C, SS_{(d,K)} \geq SS_{threshold} \} \qquad (2)$$

where:

- D denotes a document ID.
- K denotes a keyword.
- C is a set of all keywords to which the document can be related.
- $SS_{(d,K)}$ is the similarity score for document d for the keyword K.
- $SS_{threshold}$ is the minimum similarity score acceptable for a document to relate to that keyword.

Table 3. Similarity score of document ID 21 in terms of various keywords

D	21	21	21	21
Keyword	filtering	clustering	Data	item
Title	1	1	0	0
Subject	2	1	0	1
Description	2	7	1	6
Content	10	5	3	4
Reference	9	6	1	5
SS	12.59	11.59	2.6	9

If $SS_{(d,K)}$ is less than $SS_{threshold}$, then the keyword K is not applicable to the profile of the document d, represented by 0 in the document profile Table. From formula (2), each document can have a profile based on the keywords and similarity score calculated by formula (1). For example, if a given threshold is 9, the profile of document 21 is:

$DP_{(21)}$ = {(filtering, 12.59), (clustering, 11.59), (item, 9)}.

3.3 Clustering Algorithm

There are various clustering algorithms available. In this paper, we use the same concept as in the k-mean algorithm [4] to split a set of documents into some desired clusters.

The basic idea of K-mean is to find a single partition of the data, which has k number of clusters such that objects in the same cluster are close to each other in some sense, and those in different clusters are distant. In our case, numbers of the clusters depend on the value of $SS_{threshold}$.

Using the document profile proposed in Section 3.2, we can measure the correlation of similarity score among the documents. Based on this correlation, the documents will be clustered. Table 4 shows the document profile for the example documents shown in Table 1.

In our example, the objects of clustering are the documents and the keyword appearing in the elements of the documents (i.e. the document profile). Therefore, the documents in the same cluster will be considered as relevant to that keyword, that is the documents in the same cluster contain relevant, inter-related information, based on one or more keywords.

For example, if the given threshold is 9, then the document 21 is not relevant to the keyword "data", while document 45 is not relevant to the keyword "item". The final pass of the algorithm produces the clusters of (21, 45) for the keyword "filtering", (21, 567) for the keyword "item", (45, 567) for the keyword "data".

Table 4: Document profile

D	21	45	567	789
Filtering	12.59	11.89	0	10.5
clustering	11.59	0	0	18.1
data	0	10.3	23	0
item	9	0	12.3	12.4

4. Experimentation

4.1 System Design

An experimental system (including a web crawler) is designed to evaluate the performance of the proposed method. The web crawler is implemented with a COM object that starts from a URL specified by the user and follows the outgoing links to search for the given keywords. In this paper, for a given keyword, the similarity score of each URL located will be computed using our proposed algorithm. If the similarity score (SS) is greater than the given threshold ($SS_{threshold}$), then the keywords, similarity score and URL will be stored in the document profile (DP) using XML. The URLs associated with keywords are then clustered and indexed as in the second stage of our proposed method. The web crawlers will keep searching keyword one by one until the number of web pages crawled reaches a user-specified limit. The architecture is shown in Fig. 2

Hence, the DP contains the index of all documents, which have undergone restructuring and clustering operations using our proposed method, as shown in Fig. 1 On the other hand, the web crawler in our experimentation mimics a search on documents without restructuring and clustering.

4.2 Experimental Results

Tests are conducted on a Pentium 4 computer with 2.0 GHz processor, 248 MB of RAM, Windows XP operating system.

The quantitative results (using the keywords "Information", "Technology", "Sports", "Special", and "Health"; and $SS_{threshold} = 0.9$) are summarized in Table 5, showing the comparisons between the times of retrieving a document structured and clustered using our proposed method and the times of retrieving the same document without using our proposed method. The results clearly show our proposed method improves the efficiency of information filtering and retrieval in milliseconds based on a size of 100k databases. Therefore, the efficiency of retrieval of keywords in a large sized database will be tremendously improved.

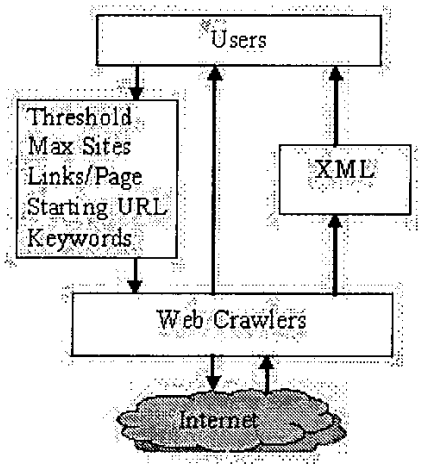

Fig. 2 The architecture of elements based filtering system

Table 5. Experimental results

Keywords	Information	Technology	Sports	Special	H
SS *threshold*	0.9				
Starting URL	http://www.ninemsn.com.au				
Max sites	500				
Max Links/Page	20				
Retrieval of Clustered keywords (sec)	.00102	.00143	.00124	.00101	.00155
Retrieval of Non-Clustered keywords (sec)	.00129	.00162	.00160	.00112	.00195

The diagram is shown in Fig. 3

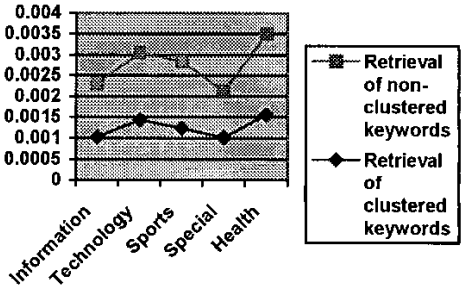

Fig. 3 Diagram of experimental results

Fig. 4. The clustered and not-clustered XML

Fig. 4 shows the clustered and non-clustered documents. The clustered documents are structured with the similarity score calculated, while the non-clustered documents are scattered without similarity score calculated.

5. Conclusion

In this paper, we propose elements and document profile based information clustering method. This new method computes the correlation similarity score among documents. The document with similarity score above a given threshold will be clustered. Then we derive the document profile based on the similarity score. Therefore, the document will be clustered based on different keywords. Although some clustering methods

have been proposed, there has not been much focus on clustering more than two documents. Our approach computes similarity score and derives document profile offline. The documents are clustered and this makes information retrieval more efficient. The experimental results have shown significant differences between the document retrieval times using our method and the retrieval times without using our method. For future research, we would like to investigate how this method can be optimized.

6. References

[1] Kobayashi, M and Takeda, K, "Information Retrieval on the Web", *ESSIR 2000, LNCS 1980*, Springer-Verlag 2000, pp. 242-285.

[2] Meng, X and Chen, Z, "Personalized Web Search With Clusters", *International Conference on Internet Computing, IC'03*, 2003, pp. 46-52.

[3] Dublin Core -- http://dublincore.org/documents/dces/

[4] J, A Hartigan. "Clustering Algorithms. WILEY Publication, 1975.

[5] J. Mostafa, S. Mukhopadhyay, M. Palakal, W. Lam, "A Multilevel Approach to Intelligent Information Filtering: Model, System, and Evaluation", *ACM Transactions on Information Systems*, Vol. 15, No. 4, October 1997, pp. 368–399.

[6] S. Vrettos, A. Stafylopatis., "A Fuzzy Rule-Based Agent for Web Retrieval-Filtering", *Web Intelligence: Research and Development : First Asia-Pacific Conf.*, Springer-Verlag, October, 2001, pp. 448-453.

[7] F. Yang, Y. Zhu, B. Shi, "A New Algorithm for Performing Ratings-Based Collaborative Filtering", *Web Technologies and Applications: 5th Asia-Pacific Web Conf.*, Springer-Verlag, 2003, pp. 239 – 250.

[8] Robertson. S. and Walker. S., "Threshold setting in adaptive filtering", *Journal of Documentation,* 2000, 56, 3:312-331.

[9] P. Resnick et al., "GroupLens: An Open Architecture for Collaborative Filtering of Newnews:, *Proc. ACM 1994 Conf. Computer Supported Cooperative Work*, ACM Press, 1994, pp. 175-186.