Using Web structure and summarisation techniques for Web content mining

Lihui Chen *, Wai Lian Chue

School of Electrical and Electronic Engineering, Division of Information Engineering, Nanyang Technological University, South Spine, Block S1, Nanyang Avenue, 639798 Republic of Singapore

Received 9 July 2003; accepted 19 August 2004
Available online 7 October 2004

Abstract

The dynamic nature and size of the Internet can result in difficulty finding relevant information. Most users typically express their information need via short queries to search engines and they often have to physically sift through the search results based on relevance ranking set by the search engines, making the process of relevance judgement time-consuming. In this paper, we describe a novel representation technique which makes use of the Web structure together with summarisation techniques to better represent knowledge in actual Web Documents. We named the proposed technique as Semantic Virtual Document (SVD). We will discuss how the proposed SVD can be used together with a suitable clustering algorithm to achieve an automatic content-based categorization of similar Web Documents. The auto-categorization facility as well as a “Tree-like” Graphical User Interface (GUI) for post-retrieval document browsing enhances the relevance judgement process for Internet users. Furthermore, we will introduce how our cluster-biased automatic query expansion technique can be used to overcome the ambiguity of short queries typically given by users. We will outline our experimental design to evaluate the effectiveness of the proposed SVD for representation and present a prototype called iSEARCH (Intelligent SEArch And Review of Cluster Hierarchy) for Web content mining. Our results confirm, quantify and extend previous research using Web structure and summarisation techniques, introducing novel techniques for knowledge representation to enhance Web content mining.

© 2004 Elsevier Ltd. All rights reserved.

Keywords: Knowledge representation of Web Documents; Web structure; Summarisation; Web content mining; Content-based automatic Web Document clustering

* Corresponding author. Tel.: +65 6790 4484; fax: +65 6793 3318.
E-mail addresses: elhchen@ntu.edu.sg (L. Chen), wailian@pmail.ntu.edu.sg (W.L. Chue).

0306-4573/S - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2004.08.003
1. Introduction

The rapid growth of the Internet has led to the development of Internet2. Web surfers view information retrieved from the Internet as rich and relevant. Due to the enormous amount of information on the Internet, users typically use search engines to assist them discover relevant information. The results of Graphic, Visualisation and Usability (GVU) Centre's October 1998 tenth WWW user survey showed that about 85% of people use search engines to locate information (GVU, 1998). However, the dynamic nature and size of the Internet can result in searches that are incomplete, outdated or large number of documents returned. In addition, users of search engines typically have little or no training on how to best utilise them and they also refrain from using the advanced search features that many search engines now offer. Researchers have developed many different techniques to address this challenging problem of locating relevant Web information effectively and efficiently. Examples of such techniques include meta-searching, post-retrieval analysis and enhanced visualisation of search results (Chen, Fan, Chau, & Zeng, 2001; Hearst & Pedersen, 1996; Zamir & Etzioni, 1999).

1.1. Objectives

The main objective of this research is to investigate how the Web structure together with summarisation techniques can be used for Web content mining to address the challenging problem of locating relevant Web information effectively and efficiently with the help of search engine technologies. In other words, the following novel techniques will be exploited:

- A method to better represent knowledge in actual Web Documents.
- Content-based automatic clustering of Web Documents.
- Intuitive GUI for visualising and browsing the clustering results.
- Term selection in pseudo-relevance feedback to overcome the ambiguity of short queries.

1.2. Related work

The motivation for the research detailed in this paper is twofold. It emanates both from a need to plug a research gap in content-based knowledge representation of actual Web Documents and a need to provide users with a better means to assist them discover relevant information from the Internet more effectively and efficiently.

Web mining is the use of data mining techniques to automatically discover and extract information from Web Documents and services (Etzioni, 1996). It can be classified into three categories: Web content mining, Web structure mining and Web usage mining (Kosala & Blokeel, 2000). Web content mining refers to the discovery of useful information from Web contents. It encompasses resource discovery from the Web (Chakrabarti, van der Berg, & Dom, 1999; Cho, Garcia-Molina, & Page, 1998), document categorisation and clustering (Kohonen et al., 2000; Zamir & Etzioni, 1999), and information extraction from Web pages (Chang, Hsu, & Lui, 2003; Tolle & Chen, 2000). Many search engines are available on the Internet, each having its own characteristics and employing different algorithms to index, rank and present Web Documents. Users typically use search engines to assist them discover relevant information or to achieve certain level of Web content mining. However, current search engines have the following major limitations:

- Users are presented with either too few or too many search results based on relevance ranking and have to physically sift through them one by one (Tombros & Sanderson, 1998).
• Users typically use short keywords as the query (Spink & Xu, 2000) that may not fully describe their interest as they may have only a vague idea of what information is needed. Another recent survey conducted by NEC Research Institute shows that about 70% of Web users typically use only a single keyword or search term (Butler, 2000).
• The search results have low precision, which is due to the irrelevance of many search results. This results in a difficulty finding the relevant information (Kosala & Blokeel, 2000).
• The search results have low recall, which is due to the inability to index all the information available on the Web such as dynamically generated Web Documents. This results in a difficulty finding the unindexed information that is relevant (Kosala & Blokeel, 2000).

To overcome the first limitation listed above, Web content mining techniques have been applied to generally improve the searching experience (Chen et al., 2001; Zamir & Etzioni, 1999). On the other hand, we believe that the various search engines are also actively involved in researching new techniques (Google Search Engine, n.d.; Olsen, 2002) to overcome the last limitation listed above.

The multi-stage process of search, starting with a general query and then getting more specific, has been investigated and is well documented in non-Web search (Marchionini, 1995). However, little work has been done to incorporate such idea into Web search and the most relevant work we found in literature is reported in Chang and Hsu (1999) and Crimmins and Smeaton (1999). Furthermore, categorisation and clustering techniques have also been investigated as a post-retrieval document browsing technique, where search results are classified into categories such that the user can browse and navigate through the set of retrieved documents more easily. NorthernLight Search Engine (online) is an example of a search engine that categorises retrieved Web pages into predefined search categories called “Custom Search Folders”. Another approach is to categorise Web pages on the fly without resorting to predefined categories. For instance, SONIA (Sahami, Yusufali, & Baldonado, 1998) is a meta-search engine that clusters search results, extracting keywords to describe each cluster and allow the user to expand search within a cluster. Scatter/Gather (Cutting, Karger, Pedersen, & Tukey, 1992; Cutting, Karger, & Pedersen, 1993; Hearst & Pedersen, 1996) is another example of a system that allows users to iteratively refine their search by clustering documents interactively and browsing the results.

Most of the Web Documents available on the Internet are defined through Hyper Text Markup Language (HTML) that allows an author to organise the presentation of a document content by means of special tags and interpreted by Web browsers. Web Documents can contain both multimedia information and connections to other documents through hyperlinks. A hyperlink is often created based on the principle that links are connections among documents that are similar. Hyperlinks are increasingly being used to improve the ability to organise, search and analyse the Web (Brin & Page, 1998; Yang, Slattery, & Ghani, 2002). Previous research has shown that extended anchortext instead of document full-text (Glover, Tsioutsouliklis, Lawrence, Pennock, & Flake, 2002) and query-biased summarisation technique (White, Jose, & Ruthven, 2003) are more effective in representing Web Documents. Moreover, results have shown that query expansion using document summaries can be considerably more effective than using full-document expansion (Lam-Adesina & Jones, 2001).

In this paper, we will describe how the proposed Semantic Virtual Document (SVD) can be applied to better represent knowledge in actual Web Documents. We will also discuss and present content-based automatic clustering of Web Documents using the Hierarchical Agglomerative Clustering (HAC) algorithm and a “Tree-like” GUI interface for post-retrieval document browsing to enhance the relevance judgement process. Furthermore, we will also introduce how our cluster-biased automatic query expansion technique can be used to overcome the ambiguity of short queries typically given by users. We will also outline our experimental design to evaluate the effectiveness of the proposed SVD via a prototype system 

iSEARCH.
1.3. Organisation

The remainder of this paper is organised as follows: Section 2 discusses the proposed SVD for knowledge representation. In Section 3, we present the well-known HAC algorithm and the proposed SVDs to automatically reorganise the results returned by search engines. Section 4 presents our “Tree-like” visual interface for browsing the document clustering results while cluster-biased automatic query expansion will be presented in Section 5. In Section 6, we will outline the design and results of our experiments to validate the proposed methods via iSEARCH. Finally, Section 7 provides conclusions and future work.

2. Semantic Virtual Documents

We will discuss and present our techniques for knowledge representation of actual Web Documents using the proposed SVD, which contain context-dependent summaries that are highly descriptive of the actual Web Documents contents. Each SVD not only makes use of extended anchortext instead of document full-text (Glover et al., 2002) and query-biased summarisation technique (Tombros & Sanderson, 1998; White et al., 2003) but also incorporates our novel anchortext-biased summarisation technique in order to represent each Web Document semantically.

A SVD is defined as a collection of Semantic Summaries created from links pointing to an actual Web Document and Query-biased Summaries for the actual Web Document (Top 2 highest scoring sentences will be selected based on the query-biased sentence scoring) as shown in Fig. 1. If a Web Document has no in-bound links, the SVD will consist of only the Query-biased Summaries for the actual Web Document (Top 5 highest scoring sentences will be selected based on the query-biased sentence scoring).

2.1. Hyperlinks and anchortext

Most content on the Internet is formatted for human consumption. But some contents without explicitly labeling, a computer has no idea what they are (Frauenfelder, 2001). Hyperlinks (or citations) are being actively used to improve Web search engine ranking (Brin & Page, 1998), improve Web crawlers (Cho et al., 1998), discover Web communities (Flake, Lawrence, & Giles, 2000), organise search results into hubs and authorities (Kleinberg, 1999), make predictions about similarity between research papers (Lawrence, Giles, & Bollacker, 1999) and even to classify target Web pages (Yang et al., 2002). The basic assumption
made by link analysis is that a link is often created because of a subjective connection between the original document and the cited or linked document (Glover et al., 2002).

On the Internet, virtually every Web page contains several hyperlinks to other pages, often in parts of the World Wide Web developed by somebody else and thus turns the static Web pages into active hubs of related information (Gallagher, 2002). A hyperlink has two components: The URL pointing to the destination page and associated anchortext (in the source page) describing the link as shown in Fig. 2.

Since a page creator physically determines the anchortext associated with each link, the anchortext may better summarise the contents of the destination page. This allows for connecting words to destination pages and is the basic mechanism used by search engine Google with their PageRank algorithm (Brin & Page, 1998).

2.2. Summarisation

Although the anchortext may better summarise the contents of a destination page, the anchortext alone may not be sufficient to fully reflect the user’s information need due to a lack of context information. Therefore, to better summarise the contents of a given destination page, we adopt the sentence extraction model, which is capable of producing acceptable summaries that are domain independent (White et al., 2003) to find a subset of the document that is indicative of its contents, typically by scoring words and then sentences according to specific rules.

2.3. Query-biased scoring

The work of Tombros and Sanderson (1998) involved computing a ‘query score’ for each sentence in a document based upon its relevance to the query.

\[
\text{Query score} = \frac{n^2}{q}
\]  

(1)

where \( n \) is the number of query terms in the sentence and \( q \) is the total number of query terms.

This comes from the view that the larger the number of query terms in a sentence, the more relevant the sentence is likely to be. The motivation comes from the belief that if users could see the sentences in which their query terms appeared, they would be able to make a better assessment of document relevance.
2.4. Anchortext-biased scoring

Extending the concept of query-biased scoring, we suggest a new sentence scoring called anchortext-biased where an ‘anchortext score’ for each sentence in the destination page is computed based upon its relevance to the anchortext in the source page describing the destination page.

\[
\text{Anchortext score} = \frac{n^2}{a}
\]

where \(n\) is the number of anchortext terms in the sentence and \(a\) is the total number of anchortext terms.

This comes from the view that the larger the number of anchortext terms in a sentence, the more relevant the sentence is likely to be. The motivation comes from relating the relevant sentences in the destination page with the anchortext in the source page.

2.5. Semantic summaries

Semantic summaries will be created to better describe a given destination page by incorporating context information to eliminate possible semantic ambiguity and vagueness. Semantic summaries will be composed of the following five sentences as shown in Fig. 1:

- **Query-biased summaries**: Top 2 highest scoring sentences will be selected based on the query-biased sentence scoring for the source page, which helps to reflect the information need expressed in a query.
- **Anchortext-biased summaries**: Top 2 highest scoring sentences will be selected based on the anchortext-biased sentence scoring for the destination page, which helps to relate the relevant sentences in the destination page with the anchortext in the source page.
- **Anchortext sentence**: The sentence in the source page containing the anchortext describing the destination page (refer to Fig. 2), which helps to reflect the source page creator’s description of the destination page.

1. The Actual Web Document is split into its component sentences.
2. Create Query-biased Summaries for the Actual Web Document:
   => Compute Query Score for each sentence in the Actual Web Document.
   => Select top 2 highest scoring sentences.
3. Create Semantic Summaries (composed of 5 sentences) for each Actual Web Document and Back Link Document pair:
   {  
   3.1 The Back Link Document is split into its component sentences.
   3.2 Create Query-biased Summaries for the Back Link Document:
     => Compute Query Score for each sentence in the Back Link Document.
     => Select top 2 highest scoring sentences.
   3.3 Create Anchortext-biased Summaries for the Actual Web Document:
     => Locate Anchortext in the Back Link Document based on the Initial URL.
     => Compute Anchortext Score for each sentence in the Actual Web Document.
     => Select top 2 highest scoring sentences.
   3.4 Locate Anchortext Sentence in the Back Link Document based on the Initial URL.
   }

Fig. 3. The algorithm to create a Semantic Virtual Document.
2.6. Algorithm to create Semantic Virtual Documents

The algorithm to create a SVD given an Actual Web Document (together with its Initial URL) and some Back Link Documents is depicted in Fig. 3. Note that if an Actual Web Document has no in-bound links (i.e. No Back Link Documents), the SVD will consist of only the Query-biased Summaries for the Web Document (Top 5 highest scoring sentences will be selected instead based on the query-biased sentence scoring).

3. Clustering of Web Documents

In order to automatically reorganise the results returned by search engines, we will discuss and present content-based automatic clustering of Web Documents using the HAC technique and the proposed SVD. We will also describe our data structure together with our fast implementation technique of HAC to speed up the automatic document clustering process. In addition, we will also illustrate how to represent the computer-generated clusters with descriptive textual summaries.

3.1. Document clustering

Several important points should be considered in the development and implementation of algorithms for clustering Web Documents. These include identifying relevant features of documents and determining appropriate weights for each feature, selecting an appropriate clustering method to partition the collection into clusters of similar documents and establishing a pairwise measure of document similarity as well as computational and memory resources limitations.

Numerous document similarity measures such as Cosine, Dice and Jaccard have been proposed but the choice of similarity measure has less qualitative impact on clustering results than the choice of the clustering algorithm (Willett, 1988). We have chosen to use the cosine similarity measure since cosine has the nice property that it is 1.0 for identical vectors and 0.0 for orthogonal vectors.

Numerous clustering algorithms such as various HACs (Rasmussen, 1992) have been applied to produce a hierarchy of clusters. We have chosen to use the Single-Linkage HAC due to its simplicity and the availability of an optimal space and time algorithm for its computation (Voorhees, 1986). More discussions on the advantage of adopting HAC Single-Linkage will be given in Section 3.3.

3.2. Knowledge representation using Semantic Virtual Documents

It has been shown that clustering may be done with vectors that use only a few representative terms from a document where the quality of clustering is not adversely affected while offering a significant speed advantage compared to full-profile clustering (Schutze & Silverstein, 1997). Based on this, we propose to cluster the SVD created instead of the actual Web Documents. We may view the SVDs as truncated document vectors containing context-dependent summaries that are highly descriptive of the actual Web Documents contents.

Furthermore, the reduced dimension of the Feature Vectors created for the SVDs using the vector space model (Salton, Wong, & Yang, 1975) will also result in a faster and more efficient document clustering process.

3.3. Fast implementation of HAC Single-Linkage algorithm

Traditionally, a similarity matrix is constructed where row \( i \) and column \( j \) represents the similarity between Document \( i \) and Document \( j \). However, the similarity matrix can be reduced to a triangular matrix
for HAC Single-Linkage approach since the similarity value between Document \( i \) and Document \( j \) is the same as the similarity value between Document \( j \) and Document \( i \). Furthermore, it is different from other HAC algorithms, the inter-cluster similarity in the Single-Linkage HAC is defined in terms of the greatest inter-document similarity, which means that with a right procedure a HAC clustering can be performed without recalculating the similarities between each document or each newly formed cluster with any other documents/or newly formed clusters at each step in the clustering processing. Therefore instead of constructing a similarity matrix and update the similarity values at each clustering step (which is both memory and time intensive), we propose to construct a Vector of Similarity Beans, which store the similarities between each document with all other documents—the similarity values need to be computed once only in the entire clustering process, using the cosine similarity measure mentioned earlier. Our fast implementation of HAC Single-Linkage algorithm is depicted in Fig. 4. The similarity beans are sorted in a descending order which enable no-update in similarity values to be preformed during the clustering processing. Table 1 shows an example of the proposed data structure which contains a reference to Document A, a reference to Document B and the Similarity Value between them.

From Fig. 4, it is observed that in this implementation, there is no need to re-calculate or update the similarity between each newly formed cluster with any of existing clusters or documents at each clustering step. The hierarchical structure could be formed step by step purely based on the pre-sorted similarity beans

```
1. Create Singleton Clusters.
   => Each Document is placed in a Cluster with Similarity Value initialised to 1.0.

2. For each element in the Vector of Similarity Beans (Sorted in descending order), based on the similarity value between document A and B
   {
   2.1 If Document A and Document B are already in Clusters:
   {
   2.1.1 If Document A and Document B are in the same Cluster:
   => No action needs to be taken.
   2.1.2 If Document A and Document B are in different Clusters:
   => Merge the Clusters to form a new Cluster.
   }
   2.2 If Document A or Document B (but not both Documents) is in a Cluster:
   => Add the Document without a Cluster to the other Document's Cluster.
   2.3 If neither Documents (Document A and Document B) are in a Cluster:
   => Create a new Cluster containing both Documents.
   }
```

Table 1
Example of vector of similarity beans (sorted in descending order)

<table>
<thead>
<tr>
<th>Document A</th>
<th>Document B</th>
<th>Similarity value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0.9</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.7</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0.6</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.5</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>0.4</td>
</tr>
</tbody>
</table>

Fig. 4. Fast implementation of HAC Single-Linkage algorithm.
between every two documents. As a result, the document clustering process can be performed more efficiently and rapidly.

3.4. Creation of cluster summaries

After the clustering process, we need to represent the computer-generated clusters with descriptive textual summaries. Recall that the SVDs contain context-dependent summaries that are highly descriptive of the actual Web Documents contents. As such, we propose to extract keywords from the cluster’s centroid, which is defined as the average vector in the vector space of all the SVDs in the cluster (i.e. Summary vector of the cluster).

Keyword extraction relies on an evaluation of each term’s importance and this is captured in the Term Frequency-Inverse Document Frequency (TF-IDF) term weighting scheme. Thus, we suggest that a Cluster Summary to be composed of the top 5 highest scoring terms based on the TF-IDF term weighting scheme for a particular cluster.

4. “Tree-like” GUI—A visual interface for browsing

Most of the Web search engines are text based. They display search results from user queries as long lists of pointers with/without summaries of retrieved pages. Proposals for visualising the output of an information retrieval system were presented as early as in the 1960s (Sammon, 1969). Scatter/Gather (Cutting et al., 1992; Cutting et al., 1993; Hearst & Pedersen, 1996) and Vivisimo (Vivisimo, 2000; http://vivisimo.com) are examples of visual presentations of search results that allows users to iteratively refine their search by clustering documents interactively and browsing the results. Our “Tree-like” GUI interface presents the content-based automatic clustering results and also suggests a list of additional terms that can be added to the original query based on a selected cluster to overcome the ambiguity of short queries typically given by users.

4.1. “Tree-like” GUI

Fig. 5 illustrates our idea of a very intuitive “Tree-like” GUI similar to Windows Explorer interface to visualise and browse the document clustering results where only the cluster summaries are shown in the left frame initially. As a result, the user may view the cluster (i.e. topic) of interest quickly and easily (via the cluster summaries) instead of browsing through traditional search results based on relevance ranking (i.e. ranked titles/abstracts approach).

Upon “expanding” a cluster of interest (by clicking on +), the titles of all the actual Web Documents in the cluster will be shown. The contents of the SVD together with the URL of the actual Web Document will be shown in the right frame when a particular title is selected.

4.2. Cluster-biased automatic query expansion

When a cluster summary is selected, the system will automatically suggest a list of candidate terms for query expansion and the user is free to select some or none of these to refine the original query and conduct a new Web search again if desired. Nevertheless, if the user does not physically interact with the “Tree-like” GUI within an allocated timeout, the system will automatically expand (and select) the largest cluster where a list of additional terms can be added to the original query as shown in Fig. 6.
Fig. 5. “Tree-like” GUI.

Fig. 6. Cluster-biased automatic query expansion.
5. Pseudo-relevance feedback

To address the problem of word mismatch (Furnas, Landauer, Gomez, & Dumais, 1987) and short queries (Butler, 2000) typically used by search engine users, researchers have shown that query expansion using document summaries can be considerably more effective than using full-document expansion (Lam-Adesina & Jones, 2001). We use the vector space model instead of the probabilistic model for both term weighting and pseudo-relevance feedback in our system. As mentioned earlier, our system can perform automatic query expansion with (i.e. selecting a particular cluster of interest) or without user intervention depending on how the user interacts with the “Tree-like” GUI during browsing.

5.1. Automatic query expansion

Automatic query expansion techniques have a significant advantage over manual techniques such as relevance feedback (Salton & Buckley, 1990) and manual thesauri because they require no intervention by the user to identify which of the documents presented in response to an initial query are relevant. Xu and Croft (1996) provides a detailed review of the global and local techniques available for automatic query expansion. Recent results have shown that pseudo-relevance feedback is a simple yet effective automatic query expansion technique (Carpineto, Romano, & Giannini, 2002).

Rocchio’s algorithm (Rocchio, 1971) is the most employed method in the vector space model, which uses document vector addition and subtraction to build the new query $Q'$ from the old query $Q$ as follows:

$$Q' = Q + \frac{1}{n_1} \sum_{i=1}^{n_1} R_i - \frac{1}{n_2} \sum_{i=1}^{n_2} S_i$$

(3)

where $R$ contains the set of $n_1$ relevant document vectors and $S$ contains the set of $n_2$ non-relevant document vectors.

5.2. Cluster-biased automatic query expansion

We propose to use only the topic (i.e. cluster) of interest selected by the user from the “Tree-like” GUI during browsing for automatic query expansion since this is the cluster deemed most relevant to the user’s information needs. Assuming that the information about the non-relevant documents is absent and all the documents in the selected cluster are very similar (i.e. the cluster hypothesis) and relevant (Refer to Section 3), Rocchio’s formula can be reduced to the following:

$$Q' = Q + C$$

(4)

where $C$ is the centroid of the selected cluster defined as the average vector in the vector space of all the SVDs in the selected cluster.

Recent results have shown that term selection in pseudo-relevance feedback based on query-biased summaries is effective (Lam-Adesina & Jones, 2001). Therefore, we propose to present the top 10 highest scoring terms (excluding those terms already present in the original query) to the user as the candidate terms for query expansion. Furthermore, since SVDs were used in the document clustering process, the source of the expansion terms is not only restricted to the actual Web Documents alone.

6. Experimental results

In order to demonstrate the feasibility and effectiveness of the proposed SVD in Web content mining, we have developed a prototype system called iSEARCH and conducted several experiments with online Web
Documents. We first compared SVD with hypertext knowledge representation where only the actual Web Documents were used. Next, we also considered the effectiveness of only query-biased summaries created for the actual Web Documents as another form of knowledge representation technique.

For our experiments, we constructed several document collections by saving the results of different queries to the Google search engine via the Google Web APIs service. We chose not to use the standard information retrieval collections including TREC (TREC Web Track, n.d.; http://es.csiro.au/TRECWeb) and the Web Document databases (Sinka & Corne, 2002), as we were interested in usefulness and effectiveness of SVD in representing actual Web Documents with context-dependent summaries. The standard information retrieval collections including the TERC, Sinka and Corne’s database (2002) do not have the information of Back Link Documents. We will not be able to make use of those Web Documents from back-links in forming the proposed SVDs. Nine general short queries (”computer”, “jaguar”, “java”, “mouse”, “saturn”, “software”, “mobile phone”, “new zealand”, “wireless application protocol”) were used to construct nine corresponding document collections. It is noted that in addition to the heavy-loaded words such as Jaguar, less-loaded queries like mobile phone, wireless application protocol have also been used for evaluating the performance of iSearch. This is because when a heavy-loaded word is used, we could only quantitatively evaluate the grouping capability of the designed system on clear-cut groups, or the contents, which has significant differences. When a less-loaded word is used, it is expected that the Web pages retrieved from Internet are more related. Therefore a finer grouping capability is required in order to detect the groups with certain degree of overlaps in the contents. In our experiments, we want to quantitatively evaluate the grouping capability of iSearch from an easier grouping with a heavy-loaded query term to a more subtle finer grouping with a less-loaded query, so as to see if the grouping has been performed based on the context, and the outcome could be evaluated based on human judgment. The experimental results of the proposed system on the above nine query terms are quite consistent and therefore results on selected queries will be detailed and discussed later.

6.1. iSEARCH prototype system

Fig. 7 shows iSEARCH in which the research was carried out to demonstrate the feasibility and effectiveness of the proposed methods. The system uses the Google Web APIs service to access and query Google’s index of over 3 billion Web pages (Olsen, 2002) for each keywords/queries specified by a user. The search results from Google will be used to create SVDs to represent all the actual Web Documents. Related Web Documents represented by the SVDs will be grouped together automatically using HAC algorithm. “Tree-like” GUI Creator will create a “Tree-like” GUI to inform the user of the clustering results. Automatic Query Expansion helps the user to refine his/her search by automatically suggesting additional words that can be added to the original query when a particular topic is selected. The user can then decide whether a new search needs to be performed again based on the modified query to retrieve more relevant Web Documents.

6.2. Construction of document collections for evaluation

We first define several short queries e.g. “computer”, “wireless application protocol”, “mobile phone”, which are similar to those query terms used in TREC. The words appearing in each query were used as keywords for a Web search using Google. We collected the top 300 results (i.e. Initial URLs) for each query and downloaded its original document from the Web. In addition, we also collected the top 5 Back Links for each Initial URL and downloaded its original document from the Web.

Thus, for each query, we have a collection of 300 actual Web Documents and 1500 Back Link Documents. Based on our “human” relevance judgment, we manually assigned each actual Web Document to one of several top-level topics manually defined for each collection. A single term is given to describe each
topic covered under the query. For example, for “computer”, there are seven key topics. A single term is used for each of those topics as listed in Table 2.

6.3. Cluster precision and recall

In iSEARCH, evaluating the document clustering results is a tricky business. However, in situations where documents are already clustered, we can compare the computer-generated clusters with the corresponding “human” defined topics (based on “human” relevance judgment). For this comparison, we will introduce and use the measures of cluster precision and recall as follows.

Suppose $T$ be the number of documents that have been clustered into one cluster actually belong to this topic (based on “human” judgment), $S$ be the size of this cluster (i.e. number of documents in this cluster) and $R$ be the number of documents that should belong to this topic (based on “human” judgment).

<table>
<thead>
<tr>
<th>Topic</th>
<th>SVDCE CP (%)</th>
<th>SVDCE CR (%)</th>
<th>QSCE CP (%)</th>
<th>QSCE CR (%)</th>
<th>HTCE CP (%)</th>
<th>HTCE CR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Security</td>
<td>100.00</td>
<td>92.31</td>
<td>100.00</td>
<td>46.15</td>
<td>83.33</td>
<td>38.46</td>
</tr>
<tr>
<td>Museum</td>
<td>100.00</td>
<td>80.00</td>
<td>100.00</td>
<td>40.00</td>
<td>100.00</td>
<td>40.00</td>
</tr>
<tr>
<td>Product directories</td>
<td>80.00</td>
<td>100.00</td>
<td>80.00</td>
<td>100.00</td>
<td>66.67</td>
<td>100.00</td>
</tr>
<tr>
<td>Gaming</td>
<td>100.00</td>
<td>75.00</td>
<td>100.00</td>
<td>75.00</td>
<td>100.00</td>
<td>50.00</td>
</tr>
<tr>
<td>Publications</td>
<td>100.00</td>
<td>90.32</td>
<td>69.70</td>
<td>74.19</td>
<td>47.06</td>
<td>51.61</td>
</tr>
<tr>
<td>HCI</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
<td>50.00</td>
<td>66.67</td>
<td>12.50</td>
</tr>
</tbody>
</table>

Fig. 7. iSEARCH prototype system.
Therefore, cluster precision and recall can be defined as:

\[
\text{Cluster precision (CP)} = \frac{T}{S}
\]

\[
\text{Cluster recall (CR)} = \frac{T}{R}
\]

In other words, a high cluster precision value implies that the cluster is a “pure” subset of the dominant topic and a high cluster recall value implies that the cluster is comprised of entirely relevant documents.

6.4. Evaluation methodology

For each query, we performed three separate experiments in order to evaluate the effectiveness of the proposed SVD. In the first experiment, we applied clustering to the document collection consisting of only the 300 actual Web Documents. We will refer to this experiment as Hyper Text Clustering Experiment (HTCE). For each individual top-level topic, the cluster precision and recall will be computed and the suggested query expansion terms will also be recorded down. Next, we run experiments on creating query-biased summaries (in the absence of Back Link Documents) to represent each actual Web Document and then applying clustering. We will refer to this experiment as Query-biased Summaries Clustering Experiment (QSCE). Lastly, the proposed SVD is used to represent each actual Web Document and then applying clustering. We will refer to this experiment as SVD Clustering Experiment (SVDCE).

6.5. Evaluation of cluster precision and recall

Tables 2–5 depict the cluster precision and recall for “computer”, “jaguar”, “mobile phone” and “wireless application protocol” queries respectively. The single terms listed under “Topic” in each table refer to the top-level topics for the corresponding query. The terms are derived based on human judgment.

In general, it was observed that higher values in cluster precision and recall were obtained for SVDCE compared to QSCE and HTCE for almost all topics (except low cluster recall for Animal Topic of “jaguar”

### Table 3
Cluster precision and recall ("jaguar" query)

<table>
<thead>
<tr>
<th>Topic</th>
<th>SVDCE</th>
<th>QSCE</th>
<th>HTCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CP (%)</td>
<td>CR (%)</td>
<td>CP (%)</td>
</tr>
<tr>
<td>Apple</td>
<td>100.00</td>
<td>93.75</td>
<td>85.29</td>
</tr>
<tr>
<td>Car</td>
<td>100.00</td>
<td>90.00</td>
<td>66.67</td>
</tr>
<tr>
<td>Club</td>
<td>100.00</td>
<td>96.67</td>
<td>70.27</td>
</tr>
<tr>
<td>Animal</td>
<td>92.31</td>
<td>63.16</td>
<td>90.48</td>
</tr>
<tr>
<td>Atari console</td>
<td>100.00</td>
<td>87.50</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Table 4
Cluster precision and recall ("mobile phone" query)

<table>
<thead>
<tr>
<th>Topic</th>
<th>SVDCE</th>
<th>QSCE</th>
<th>HTCE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CP (%)</td>
<td>CR (%)</td>
<td>CP (%)</td>
</tr>
<tr>
<td>Health</td>
<td>100.00</td>
<td>87.50</td>
<td>100.00</td>
</tr>
<tr>
<td>News</td>
<td>78.57</td>
<td>78.57</td>
<td>100.00</td>
</tr>
<tr>
<td>Virus</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Charger</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
<tr>
<td>Games</td>
<td>100.00</td>
<td>100.00</td>
<td>75.00</td>
</tr>
<tr>
<td>Recycling</td>
<td>100.00</td>
<td>100.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>
This further illustrates the usefulness and effectiveness of SVD for knowledge representation compared to the others. Similar improvements in cluster precision and recall were also observed for QSCE compared to HTCE. This is consistent with the results obtained by White et al. (2003). Moreover, it was also observed that SVDCE and QSCE completed much faster compared to HTCE due to the reduced dimension of the feature vectors created for them. However, SVDCE cluster precision and/or recall may degrade slightly compared to HTCE and QSCE if both the actual Web Documents and corresponding Back Link Documents were mainly composed of multimedia information and cannot assist in revealing more textual information. This was observed for Animal Topic of “jaguar” query, where SVDCE cluster recall degraded over HTCE and QSCE but cluster precision improved. However, this is beyond the scope of the current research.

6.6. Evaluation of automatic query expansion

Table 6 depicts some suggested query expansion terms for “computer”. In general, it was noted that the query expansion terms suggested by SVDCE helped to clarify the topic much better compared to HTCE.
and QSCE as the source of the expansion terms is not only restricted to the actual Web Documents alone. For instance, notice that two new terms “imac” and “macintosh” were suggested by SVDCE (Apple Topic of “computer”) that were both absent in HTCE and QSCE. Thus, the suggested query expansion terms may come from the Semantic Summaries, which contain additional knowledge from the Back Link Documents. This further demonstrates the usefulness and effectiveness of SVD for representation. Besides, significant improvements in cluster precision and recall were also observed for the HCI Topic of QSCE compared to HTCE. This further demonstrates the usefulness of query-biased summaries in representing knowledge in actual Web Documents.

7. Conclusions and future work

We introduced a novel technique SVD for representation of actual Web Documents. In addition, we also discussed and presented a prototype system with SVD representation, HAC clustering, a “Tree-like” GUI and cluster-biased automatic query expansion techniques to enhance the relevance judgment process.

Experimental results have shown that SVD representation resulted in a faster and more accurate document clustering. Furthermore, term suggestion based on the proposed cluster-biased automatic query expansion via SVD is more effective than others.

The results presented in this paper have proved the usefulness and effectiveness of the newly proposed SVD for knowledge representation in Web content mining. However, this research is by no means definitive and there are a few areas on which future research could focus:

• Although we have focused on representing knowledge in HTML Web Documents, there are numerous other file formats that are publicly accessible on the Internet. Also, if both the actual Web Documents and corresponding Back Link Documents were mainly composed of multimedia information (e.g. graphics, audio, etc.), SVD will not be particularly effective in revealing more textual information. It would be worthwhile to research new techniques to include these file formats and multimedia information for knowledge representation.

• Though we have chosen HAC Single-Linkage for document clustering, there are numerous other clustering algorithms available. It would be interesting to observe whether these algorithms would yield better clustering results.

• The numerical constants used to select the number of terms for creating the cluster summaries and suggesting additional number of words for query expansion have only been selected via intuition and no research has been carried out to test their validity. Nevertheless, through testing, we were able to tailor the constants to produce what was thought to be the most effective. Similarly, the number of sentences together with the number of Back Link Documents have also been selected via intuition and testing to produce what was thought to be the most effective context-dependent summaries. However, the validity of these assumptions has not been fully tested and empirical research into their effectiveness may be necessary in the future.

References


