An Efficient Clustering Search Engine Using Directory Tree

Chun-Wei Tsai,* Heng-Yao Hsu,† Chu-Sing Yang,‡ and Ming-Chao Chiang§

*§Department of Computer Science and Engineering, National Sun Yat-sen University
†‡Department of Electrical Engineering, National Cheng Kung University
*cwtsai87@gmail.com, †q3695122@mail.ncku.edu.tw,
‡csyang@ee.ncku.edu.tw, §mcchiang@cse.nsysu.edu.tw

摘要

在本篇論文中，我們提出一個分群式搜尋引擎，稱之爲CSES。此系統的基礎設計不同於傳統的搜尋引擎GYM (Google, Yahoo!, and MSN)等。傳統的搜尋引擎提供給使用者的回傳網頁並未經過分門別類，而CSES提供使用者一個分類過後的網頁資訊，以減少使用者在過濾無用及多餘的資訊所花費的時間。這個系統利用Open Directory Project,Yahoo!及Google的分類目錄資訊建立一個分類目錄樹(meta-directory tree)。在這個研究中，我們建立一個分類目錄樹合併方法去有效的進行網頁的分門別類，稱之爲MDTBC。這個方法將輸入到匯總式搜尋引擎所回傳的查詢結果，利用分類目錄樹進行分群並輸出一個分門別類的資訊給使用者。實驗結果證明這個系統比傳統的分群演算法更適合在這個問題之上，並且可以提供更合適的資訊給使用者。

關鍵字：分類式搜尋引擎，匯總式搜尋引擎，文件分群

ABSTRACT

In this paper, we present a Clustering Search Engine System, called CSES. This system is fundamentally different from traditional search engines, such as GYM (Google, Yahoo!, and MSN), in that GYM presents to the user non-classified web pages whereas CSES provides to the user taxonomic web pages that would greatly help reduce the time required by the user to filter out irrelevant or redundant information. The system uses the meta-directory information made available by Open Directory Project, Google Directory, Yahoo! Directory to create a meta-directory tree and then builds on the meta-directory tree an efficient method for clustering the web pages, which we refer to as meta-directory tree based clustering, or MDTBC. This method takes as input the query result of the meta-search engine module of CSES and produces as output a clustered (or categorized) document set based on the information available in the meta-directory tree. Our simulation result showed that the proposed system outperforms both the traditional clustering algorithms and the traditional search engines, in terms of, respectively, the relevance of the search result and the information coverage.

Keywords: clustering search engine, meta-search engine, document clustering.

1. INTRODUCTION

Over the past decades, the increase in the computing power of computers and the advance in internet technology have been phenomenal. As such, a tremendous amount of traditional printed material has been transformed into digital material. One recent study [1] found that over 90% of the information in use today is born in electronic form since the introduction of the first web search engine Wandex [2], a web crawler developed by Matthew Gray at MIT in 1993. In the same year, another search engine Aliweb made its debut to the public. After then until 2000 or so, many search engines, namely, WebCrawler, Northern Light, Google, Vivisimo, Yahoo! and so on had appeared and become popular. In [3], Lawrence pointed out that the six search engines (HotBot, AltaVista, Northern Light, Excite, Infoseek, and Lycos) can cover up to about 60% of the internet information. But recently, because of the explosion of the online information, Lawrence [4] indicates that there are *not* even a single search engine that can cover more than 16% of the internet information. After 2000, the GYM (Google, Yahoo!, and MSN) has gradually become the most popular search engines.

In general, directory search provides more relevant information than search engine. However, most internet users still rely on search engine to find the information they need. This is because search engine can provide the user of the internet a large number of web pages that contain terms specified in a given query. For instance, given the query "*oasis*" to the Yahoo! Directory and Yahoo! Search Engine, the number of web pages returned by the Yahoo! Directory is 3,419[1] whereas the number of web pages returned by the Yahoo! Search Engine is more than 37,600,00[2]. Even though search engine can return a large number of web pages for a given query, there exists an important problem that needs to be solved. This problem is the relevance of the returned information. The irrelevant information returned by a search engine is not an error of the search engine. Instead, it is, to a large extent, due to the fact that the query itself is ambiguous. That is, keywords in a given query may have numerous meanings [5]. For example, if "mp3" was given to a search engine, it could mean an "mp3 music file" or an

[1] http://search.yahoo.com/search/dir?h=c&p=oasis, May 25,2007.
[2] http://search.yahoo.com/search?p=oasis, May 25,2007.

"mp3 player." Another example is when the keyword "cat" (meaning a cat) is given as a query to the Google search engine,[3] the first item returned is the company "Caterpillar, Inc.," which has nothing to do with the animal "cat."

For these reasons, in 1997, Northern Light[4] provides a search engine that can automatically classify the search result. According to the classified information, the internet user can easily find the information they need. Recently, many clustering search engines have been launched, which include Vivisimo [6], iboogie [7], and grokker [8].

In this paper, we present an alternative in the design and implementation of a clustering search engine system, called CSES. Besides, as part of the system, we propose a novel and fast clustering algorithm—built on the notion of a meta-directory tree—that can not only enhance the relevance of the search result but also reduce the computation time of clustering. The focus of this research is primarily on reducing the computation time of clustering, and the technologies used include information retrieval (document relationship analysis), information extraction (meta-search engine and directory information collection) and data clustering (classification of web pages).

The remainder of the paper is organized as follows. Section 2 introduces the clustering search engine and technologies used in the design and implementation of the system. Section 3 gives the problem definition and explains how the system is designed and implemented. Section 4 gives a detail description of the proposed system. Section 5 presents the simulation result. Conclusion is given in Section 6.

2. Related Works

2.1 Information Retrieval and Information Extraction

Information retrieval (IR) [9] encompasses multiple disciplines involving researches in dependence analysis of a group of files, clustering of files, and classification of files. Information extraction (IE) is one of the IR researches. The role of IE is to extract useful information blocks embedded in web pages and remove useless information such as advertisements. Besides, IE can convert different web page formats into a unified format. In general, the role of IR is to analyze the relationship of file (document) contents to identify similarities, and such research includes Vector Space Model [10] and Document Clustering [11]. The major difference between information retrieval and information extraction is that the former analyzes the relationship between files while the latter focuses on the recognized contents in a single file.

IE is an automatic wrapper method [12], which does not require human labeling of specific regions in web pages to mark them as interesting [13]. It supposes that most information regions in a web page are arranged in records. Recently, many researches [14], [15], [16], [13], [17], [18] have been proposed to extract these records and transform them into a program readable format. The CSES described herein uses a simple wrapper process to extract the useful information blocks. The following section will give a brief introduction to the IE technology employed in the CSES.

2.2 Clustering Search Engine

Traditional search engines such as Google, Yahoo and MSN provide a way for the user to search for information in the internet by specifying a query that may contain one or more terms. However, as we discussed earlier, none of them can cover more than 16% of the internet information. For this reason, meta-search engines are developed to enhance the information coverage by sending the query to—and combining all the information returned by—all the underlying (traditional) search engines. As such, the information coverage is greatly enhanced. But it also means that they will always return a long list of web pages [19]. Unfortunately, if the query entered by the user is itself ambiguous, the search result will include information that belongs to more than one category. In this case, clustering search engine provides a better way to help the user find the information quickly, by grouping the web pages into distinct categories that would greatly help the user get the information they are looking for. In other words, clustering search engine would help the user filter out all the useless web pages.

Over the past few years, many clustering search engines have been developed and put into use, which include Vivisimo [6], SnakeT [20], IBoogie [7], Kartoo [21], Copernic [22], Grokker [8], Dogpile [23] and Webclust [24]. These clustering search engines all provide a taxonomy for the search result to help the user find the needed information quickly. Kartoo and Grokker even provide a graphical interface to help the user understand the relations between the returned web pages.

Recently, many studies [25], [19], [26], [27], [28], [29] have been conducted, and many ways have been proposed to enhance the clustering capability of clustering search engines. Paolo [25] divides clustering search engines into five categories: (1) Single words and flat clustering, (2) Sentences and flat clustering, (3) Single words and hierarchical clustering, (4) Sentences and hierarchical clustering, and (5) Personalized ranking algorithms. Giannotti [28] uses k-means for clustering the web pages. Tsai [5] uses the genetic algorithm with k-means and multiple search method for improving the clustering result. SnakeT [25] uses an innovative bottom-up hierarchical clustering algorithm for providing a more useful web page information to user. Besides, Ferragina [25] uses two knowledge bases (anchor texts and DMOZ classifies) for improving the accuracy rate. In [19], five measurements are used for web document clustering, and Zeng uses three methods for improving the clustering result. In this paper, we present a novel clustering algorithm for grouping the search result returned by the meta-search engine into different topics. The following sections will introduce the proposed system.

[3] http://www.google.com.tw/search?hl=zh-TW&q=cat, March 5, 2006.
[4] http://www.northernlight.com/, March 5, 2006.

3. Design and Implementation of the System

3.1 Problem Definition

Although search engines, such as Yahoo! and Google, can offer much more information to the user than directory search systems, they also produce excessive irrelevant or redundant information. The cluster search engine system CSES described herein provides a solution that combines the information coverage offered by search engines and the relevance of information made available by directory search systems. Tsai [5] indicates that if search engine can further cluster the search result into different categories, it would save the user a great deal of time—by giving him/her the desired contents quickly, without overwhelming by irrelevant or redundant information. In this research, our focus will be primarily on the clustering technology that can be used to efficiently classify the search result and thus provide the user with the most relevant information, by arranging the search result into categories so that the user can easily filter out unwanted web pages and find the web pages they need.

To simplify our discussion of the problem definition and the proposed algorithm, the following notations will be used throughout the rest of this paper except where no confusion is possible.

- $\mathcal{S}$ The set of search engines from which a meta-search engine is constructed, i.e., $S = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_m\}$ where m is the number of search engines used in the construction of the meta-search engine.
- Q The query given to the meta-search engine $\mathcal{S}$ which may contain one or more terms.
- $\mathcal{D}_i$ The set of web pages (documents) returned by search engine $\mathcal{S}_i$ in response to the query Q.
- N_i The number of web pages returned by search engine $\mathcal{S}_i$ in response to the query Q, i.e., $N_i = |D_i|$.
- N The total number of web pages returned by the meta-search engine $\mathcal{S}$ in response to the query Q. That is, $N = \sum_{i=1}^{m} N_i$.
- D The set of web pages returned by the meta-search engine $\mathcal{S}$ in response to the query Q. That is, $D = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \cdots \cup \mathcal{D}_m = \{D_1, D_2, \ldots, D_N\}$ where m and N are as defined above, and D_i is the ith web page returned by the meta-search engine $\mathcal{S}$, which consists of the *title*, *hyper link*, *abstract*, and *rank* of the ith web page.
- T The meta-directory tree—an ordered tree labeled by web pages. We further let $T_{\ell,j}$ represent the jth node at level ℓ of T.[5] Moreover, each node $T_{\ell,j}$ contains $P_{\ell,j}$ web pages where $P_{\ell,j} \geq 1$, and $T_{\ell,j,k}$ represents the kth web page in the jth node at level ℓ of T.

The problem is defined as follows:

Input: A set of brief introductions or descriptions D returned by the meta-search engine $\mathcal{S}$ in response to a given query Q.

Output: A classified document set R.

[5]Note that we are assuming the root of the tree T has level 0.

3.2 The CSES Framework

Fig. 1 shows the framework of CSES, which is made up of three modules: *a meta-search engine*, *a meta-directory tree*, and *a similarity computation module*. The meta-search engine module is responsible for collecting and extracting the web page information from GYM (Google, Yahoo! and MSN) when given a query Q. The meta-directory tree module can be considered as an offline processor that takes the responsibility for creating and updating the meta-directory tree T by extracting the taxonomy information from Google, Yahoo! and ODP. In other words, the meta-directory tree module builds a knowledge base that provides the classified information of the web pages to the CSES. For example, when given a query Q, the meta-search engine module will forward the query to the Google, Yahoo! and MSN search engines to collect and extract the web page information D. Then, the CSES will use the knowledge base to determine to which cluster (or topic) each returned web page belongs. Finally, a classified document set R is returned to the user, which would greatly reduce the time required by the user to filter out irrelevant or redundant web pages. Besides, if a query Q is given to the meta-search engine for the second time and onward, the CSES will immediately return to the user the cached result if they are not expired yet. The data grid and grid computing shown in Fig. 1 are reserved for future work and are not the focus of this paper. However, it is worth mentioning in passing that the storage capability provided by data grid and the computation power offered by grid computing combined can be used to greatly enhance the performance of the system.

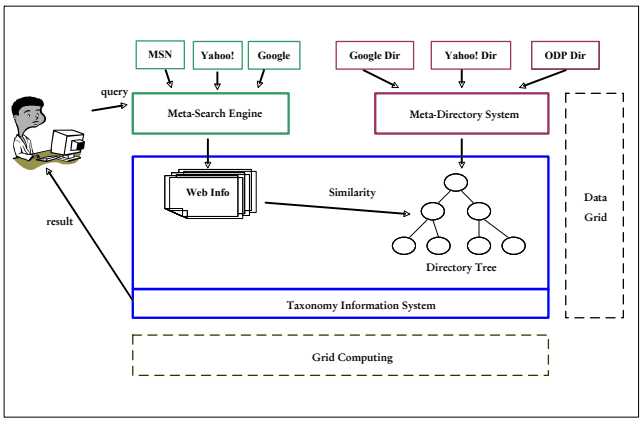

Fig. 1. Framework of the CSES.

3.3 Design of the Meta-Search Engine

To make CSES capable of recognizing common web pages, we utilize the same technology as given in [18] to analyze the structure of a web page. First, we design a wrapper program (Fig. 2) to extract the search result. In other words, as soon as a query Q is given to the meta-search engine $\mathcal{S}$, the meta-search engine will send that query to all

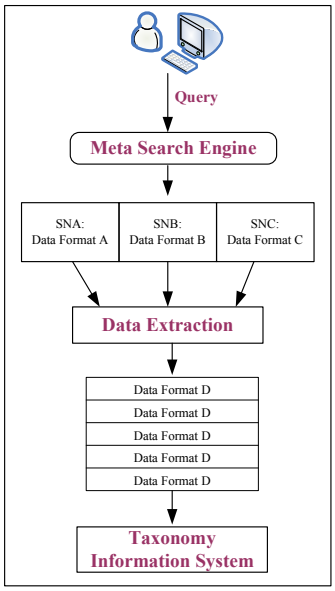

Fig. 2. Meta-search engine.

the underlying search engines $\mathcal{S}_i$ and collect the web pages returned by these search engines. The CSES will then use the pre-defined rules to extract the web page information and build records. In this study, the meta-search engine $\mathcal{S}$ also takes the responsibility for converting the web pages collected into a unified format. The extract module uses Stemmer [30] to remove the common morphological and inflectional endings and Stopword [31] to delete useless words. In short, the meta-search engine module provides the interface for the user to enter a query and then convert the search result into a unified format for later processing.

3.4 Creating and Updating the Meta-Directory Tree

To reduce the computation time of clustering, we present in this paper a novel clustering method, called meta-directory tree based clustering or MDTBC for short. This method is built on a meta-directory tree. In other words, the method has been designed specifically for the classification of web pages based on the web page information available in the meta-directory tree. It uses an offline processor to create and update the meta-directory tree and then uses that tree to classify the web pages into the most suitable clusters. Before we proceed to introduce the MDTBC, we need to explain how the meta-directory tree is created.

In this research, we design a web page *parser* to parse the meta-directory information retrieved from the public available directory systems that the CSES needs. This parser extracts the directory information in a top-down fashion. The information needed includes title, hyper link, abstract, taxonomy, and relationship. The first three fields are essentially the same as those for the search result. The fourth field is used to keep track of the catalog to which a web page information belongs. The fifth field is used to describe the relationship with other web page information. These directories provide the CSES abundant of information for the web page classification. For instance, ODP provides a large text file that contains 4,822,096 hyper link records and more than 580,000 classified catalogs. Insofar as the system described herein is concerned, the meta-directory tree is built from three directory systems: Yahoo![6], Google[7] and Open Directory Project[8]). In this research, we extract more than 10,000,000 (or 6GB) web page information from these three directory systems.

4. The Proposed Clustering Algorithm

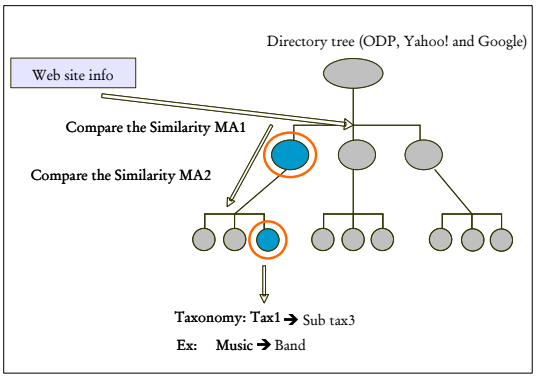

Fig. 3. Using MDTBC to classify web pages into clusters.

Insofar as an online information search system is concerned, the response time is probably one of the most important issues to be addressed. Fig. 3 shows the web pages returned by the meta-search engine and how they are clustered in CSES. Each web page information D_i is compared with entries in the meat-directory tree T to find the most suitable category. Before the computation of the similarity between D and T can be done, each web page D_i needs to be cleaned up by Stemmer's algorithm [30], and then stop word list [31] is used to remove all the useless words. Fig. 4 gives a detail description of MDTBC, assuming that the meta-search tree has been built. First, the similarity between D_i and the nodes (also called categories, topics, or clusters) in the root of T, i.e., T_ℓ where $\ell = 0$, is computed, and the most similar node $T_{\ell,j}$ is found. The features in node $T_{\ell,j}$ is the union of the features in $T_{\ell,j,1}$ to $T_{\ell,j,P_{\ell,j}}$, where $P_{\ell,j}$ is the number of web pages in node $T_{\ell,j}$. That is,

$$T_{\ell,j} = \bigcup_{t=1}^{P_{\ell,j}} T_{\ell,j,t}.$$

Then, the MDTBC assigns the category at this level to the web page D_i. The sub-category of D_i is then computed based on the similarity between D_i and the children of node $T_{\ell,j}$. This process is repeated until the leaves of the tree T are considered. The MDTBC can then find the category, subcategory, sub-subcategory, and so on of each web page D_i. In other words, if there are 10 categories at each level in T, then after the similarity computation at that level is

[6]http://http://dir.yahoo.com/, May 25,2007.
[7]http://www.google.com/dirhp, May 25,2007.
[8]http://dmoz.org/, May 25,2007.

done, the MDTBC can eventually cut the search space down to 10%. Clustering the web pages in this way is very similar to a tree search. For this reason, this method can handle a large document set and find the suitable categories (topics) quickly.

MDTBC1. Let $\ell = 0$.
MDTBC2. For each web page D_i returned by the meta-search engine $\mathcal{S}$, compute the similarity between D_i and each node at level ℓ of the tree T.
MDTBC3. Use the most similar node to determine the catalog at level ℓ and reduce the search space.
MDTBC4. If ℓ is less than the maximum level of the tree T, let $\ell = \ell + 1$.
MDTBC5. If the topic, subtopic, subsubtopic, and so on of the web page D_i have been determined, then terminate; otherwise, return to step MDTBC2.

Fig. 4. Outline of MDTBC.

MDTBC is a method for finding the most suitable cluster for an input web page. Insofar as the MDTBC is concerned, several methods such as VSM [9] and the phrase based methods [32], [33] can be used to compute the similarity between documents. In this paper, we will only present a quick and simple method as given in Eq. (1) for computing the similarity between documents. In Eq. (1), we use $S_{a,b}$ to represent the similarity between documents a (web page returned by the meta-search engine $\mathcal{S}$) and b (the web page information in a node of the directory tree) in terms of the abstract of each document. Moreover, we use n to represent the number of terms in both documents a and b, i.e., $n = |a \cap b|$. In Eqs. (2) and (3), a_i and b_i represent, respectively, the ratio of the frequency of the ith term in documents a and b (denoted, respectively, $tf_{a,i}$ and $tf_{b,i}$) to the total number of terms in documents a and b (denoted, respectively, $|a|$ and $|b|$).

$$S_{a,b} = \sum_{i=1}^{n} a_i b_i \tag{1}$$

$$a_i = \frac{tf_{a,i}}{|a|} \tag{2}$$

$$b_i = \frac{tf_{b,i}}{|b|} \tag{3}$$

5. Experimental Result

In this section, we will discuss the simulation result. For those of you who are interested, the proposed system is available [34], [35] for the public to try out. The response time and accuracy analysis of CSES will be given in the subsection where the simulation result are shown.

5.1 Experimental Result

The simulation can be divided into two parts which account for both the response time and the accuracy rate. Insofar as the response time is concerned, several different queries are tested to measure the performance of the CSES. The response time is defined to be the time difference between the time a query is given to the CSES and the time the clustered search result is returned, i.e., the time required by the meta-search engine to extract the web pages plus the time required for clustering. Table 1 shows the response time of twenty queries. When a query is given to the CSES for the first time, it requires 5468.8ms (2872.2ms+2596.6ms) in average for the twenty queries tested. For the second time and onward, the clustered search result can be quickly retrieved from the database, and thus the response is almost instant.

Table 1. The response time of queries given to the system for the first time

Query	Meta-Search Engine	Classification
oasis	2781ms (G:250ms, Y:1015ms, M:1516ms)	2377ms
shell	3625ms (G:344ms, Y:1187ms, M:2094ms)	3030ms
IEEE	2969ms (G:469ms, Y:1125ms, M:1375ms)	2289ms
suede	2640ms (G:390ms, Y:1156ms, M:1094ms)	2022ms
msn	2578ms (G:219ms, Y: 859ms, M:1500ms)	2767ms
diesel	2766ms (G:266ms, Y:1016ms, M:1484ms)	2999ms
apple	3343ms (G:250ms, Y:1359ms, M:1734ms)	2718ms
taiwan	3080ms (G:252ms, Y:1265ms, M:1563ms)	2801ms
java	3032ms (G:282ms, Y:1032ms, M:1718ms)	2673ms
hamburger	3125ms (G:344ms, Y:1188ms, M:1593ms)	2874ms
wow	2578ms (G:219ms, Y:1015ms, M:1344ms)	3250ms
polo	2268ms (G:253ms, Y:1265ms, M: 750ms)	2070ms
queen	3188ms (G:375ms, Y:1281ms, M:1532ms)	2120ms
thread	2672ms (G:297ms, Y:1313ms, M:1062ms)	2083ms
amazon	2438ms (G:250ms, Y: 859ms, M:1329ms)	2876ms
Average	2872.2ms	2596.6ms

Table 2. The response time of different clustering search engines

QUERY	Vivisimo	iBoogie	$CSES_1$	$CSES_2$
oasis	13.25ms	24.74ms	147.37ms	**0.72ms**
shell	12.64ms	23.09ms	141.60ms	**0.65ms**
IEEE	12.72ms	24.34ms	125.19ms	**0.58ms**
suede	13.74ms	23.56ms	103.60ms	**0.43ms**
msn	13.14ms	25.24ms	121.48ms	**0.55ms**
diesel	12.68ms	23.17ms	122.66ms	**0.51ms**
apple	13.04ms	24.56ms	137.75ms	**0.59ms**
taiwan	11.15ms	24.00ms	117.62ms	**0.65ms**
java	13.64ms	29.23ms	142.63ms	**0.31ms**
hamburger	13.35ms	23.79ms	133.31ms	**0.28ms**
wow	13.03ms	24.38ms	121.42ms	**0.28ms**
polo	13.04ms	24.79ms	117.24ms	**0.39ms**
queen	12.99ms	23.12ms	143.46ms	**0.43ms**
thread	12.49ms	22.05ms	132.08ms	**0.38ms**
amazon	13.55ms	29.69ms	120.77ms	**0.57ms**
Average	12.96ms	24.65ms	128.55ms	**0.49ms**

For the purpose of comparison, Table 2 shows the average response time of Vivisimo, iBoogie and CSES. The difference between $CSES_1$ and $CSES_2$ is that the former gives the response time when the query is given to the CSES for the first time while the latter shows the response time when the same query is given to the CSES for the second time and onward. It shows that $CSES_2$ is more than 26 times faster than other clustering search engines because the clustering results are cached in the database, and all the system needs to do is to retrieve the clustering results cached in the database if they are not expired and send them back to the user.

Table 3 shows the accuracy rate of CSES. The terms column gives the number of feature terms for each query given to the CSES, and the sites column represents the number of web sites collected by the meta-search engine. Table 3 shows that the accuracy rate of CSES is somewhere between 25.00% and 69.17%. In most cases, however, the accuracy rate is greater than 50%. Note that for the analysis

Table 3. ACCURACY RATE

QUERY	TERMS	SITES	ACCURACY
oasis	735	35	38.33%
shell	996	47	49.17%
IEEE	805	42	25.00%
suede	918	45	41.67%
msn	881	44	54.17%
diesel	921	47	49.17%
apple	862	44	63.33%
taiwan	960	50	64.17%
java	844	40	60.00%
hamburger	904	45	69.17%
wow	971	48	60.83%
polo	721	37	34.17%
queen	800	37	46.67%
thread	795	36	45.83%
amazon	945	44	55.83%
AVERAGE	870.533	42.733	50.50%

given here, the order of terms is not considered in the measure of the document similarity. In the future, one of our goals is to enhance the accuracy rate of the CSES.

6. CONCLUSION

In this paper, we proposed a novel Clustering Search Engine System, called CSES, for online information searches. This system combines the information coverage provided by search engines and the relevance of information offered by directory search systems. In this paper, we presented how the system is designed and implemented. Besides, we proposed a simple but novel algorithm for clustering the web pages. This algorithm is conceptually similar to a tree search that would greatly reduce the computation time for clustering and thus make it feasible for online information clustering—the response time of which is generally stringent. In particular, this algorithm is fundamentally different from traditional clustering algorithms that require a tremendous amount of computation time.

In the future, our goal is to keep improving the accuracy rate of document clustering process and enhancing the response time of the clustering search engine system described herein. However, as it shows in the paper, the CSES provides a framework to obtain more relevant information for the users of the internet. In conclusion, this system will greatly reduce the amount of time that the users need to waste filtering out irrelevant or redundant information.

Acknowledgements This work was supported in part by National Science Council, R.O.C., under contact number 95-2221-E-006-509-MY2.

REFERENCES

[1] K. Consulting, "Electronic discovery costs: Managing systems and information to control costs and improve results," *KAHN Consulting INC.*, pp. 1–12, 2005.
[2] Wandex. [Online]. Available: http://www.wikipedia.org/wiki/Search_engine
[3] S. Lawrence and C. L. Giles, "Searching the world wide web," *Science*, vol. 280.
[4] ——, "Accessibility of information on the web," *Nature 400*, pp. 107–109, 1999.
[5] C. W. Tsai, T. W. Liang, J. H. Ho, C. S. Yang, and M. C. Chiang, "A document clustering approach for search engines," *IEEE International Conference on Systems, Man and Cybernetics*, pp. 1050–1055, 2006.
[6] Vivisimo. [Online]. Available: http://search.vivisimo.com/
[7] Iboogie. [Online]. Available: http://www.iboogie.com/
[8] Grokker. [Online]. Available: http://www.grokker.com/
[9] R. Baeze-Yates and B. Ribeiro-Neto, "Modern information retrieval," *ACM Press*, 1999.
[10] G. Salton, "The smart retrieval system—experiments in automatic document processing," *Prentice Hall Inc.Englewood CliffsNJ*, 1971.
[11] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," *In Proceedings of the 26th annual international ACM SIGIR conference*, pp. 267–273, 2003.
[12] N. Kushmerick, D. S. Weld, and R. B. Doorenbos, "Wrapper induction for information extraction," *15th International Joint Conference on Artificial Intelligence*, pp. 729–737, 1997.
[13] B. Liu, R. Grossman, and Y. Zhai, "Mining data records in web pages," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 601–606, 2003.
[14] D. Embley, Y. Jiang, and Y.-K. Ng, "Record-boundary dsicovery in web documents," *ACM SIGMOD Internation Conference on Management of Data(SIGMOD'99)*, pp. 467–478, 1999.
[15] D. Buttler, L. Liu, and C. Pu, "A fully automated object extraction system for the world wide web," *21st International Conference on Distributed Computing Systems*, pp. 361–370, 2001.
[16] C. H. Chang and S. C. Lui, "IEPAD: Information extraction based on pattern discovery," *10th International Conference on World Wide Web (WWW-10)Hong Kong*, pp. 595–609, 2001.
[17] Y. Kusumura, Y. Hijikata, and S. Nishida, "Text mining agent for net auction," *ACM Symposium on Aied Computing*, pp. 1095–1102, 2004.
[18] C. W. Tsai, J. H. Ho, T. W. Liang, and C. S. Yang, "IWPS: An intelligent web portal system for web information region integration," *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3884–3889, 2005.
[19] H. J. Zeng, Q. C. He, Z. Chen, W. Y. Ma, and J. Ma, "Learning to cluster web search results," *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR'04)*, pp. 210 – 217, 2004.
[20] Snaket. [Online]. Available: http://snaket.di.unipi.it/
[21] Kartoo. [Online]. Available: http://www.kartoo.com/flash04.php3
[22] Copernic. [Online]. Available: http://find.copernic.com/copernic.html
[23] Dogpile. [Online]. Available: http://www.dogpile.com/
[24] Webclust. [Online]. Available: http://www.webclust.com/
[25] P. Ferragina and A. Gulli, "A personalized search engine based on web-snippet hierarchical clustering," *International World Wide Web Conference (WWW2005)*, pp. 801–810, 2005.
[26] S. Dumais, E. Cutrell, and H. Chen, "Optimizing search by showing results in context," *Proceedings of the SIGCHI conference on Human factors in computing systems (SIGCHI'01)*, pp. 277–284, 2001.
[27] X. He, C. H. Ding, H. Zha, and H. D. Simon, "Automatic topic identification using webpage clustering," *Proceedings of the 2001 IEEE International Conference on Data Mining*, pp. 195–202, 2001.
[28] F. Giannotti, M. Nanni, D. Pedreschi, and F. Samaritani, "Webcat: Automatic categorization of web search results," *Proceedings of SEBD'2003*, pp. 507–518, 2003.
[29] S. Osinski and D. Weiss, "Conceptual clustering using lingo algorithm: Evaluation on open directory project data," *Intelligent Information Systems*, pp. 369–377, 2004.
[30] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.
[31] Stop word list. [Online]. Available: http://www-fog.bio.unipd.it/waishelp/stoplist.html
[32] O. Zamir and O. Etzioni, "Grouper: A dynamic clustering interface to web search results," *Computer Networknos.11-16*, vol. 31, pp. 1361–1374, 1999.
[33] K. M. Hammouda and M. S. Kamel, "Efficient phrase-based document indexing for web document clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 10, pp. 1279–1296, 2004.
[34] The CSES system. [Online]. Available: http://cwtsai.ee.ncku.edu.tw/monkii/
[35] Introduction to the CSES system. [Online]. Available: http://cwtsai.ee.ncku.edu.tw/NSNP.htm