Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Fast Tree-Based Search Algorithm for Cluster Search Engine

Chun-Wei Tsai*, Ko-Wei Huang†, Ming-Chao Chiang*, and Chu-Sing Yang‡
*Department of Computer Science and Engineering,
National Sun Yat-sen University, Kaohsiung, Taiwan, R.O.C.
†Department of Computer and Communication Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
‡Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
Email: cwtsai87@gmail.com, q3897106@mail.ncku.edu.tw, mcchiang@cse.nsysu.edu.tw, csyang@ee.ncku.edu.tw

***Abstract*—In this paper, we present an Intelligent Cluster Search Engine System, called ICSE. This system is motivated by the observation that traditional search engines present to the users a set of non-classified web pages based on its ranking mechanism, and the unfortunate results are that they usually can not satisfy the need of users. For this reason, ICSE provides to the user a set taxonomic web pages in response to a user's query, and thus it would greatly help the users filter out irrelevant or redundant information. The proposed system can be divided into two parts. The first is the knowledge base constructed by Open Directory Project and Yahoo! Directory. The second is the fast clustering algorithm described herein for clustering the web pages. In addition, in response to a user's query, the proposed system will first send the query to a meta-search engine. Then, it will create a clustered document set using the given knowledge base and the clustering algorithm of ICSE. Our simulation result showed that the proposed system can enhance the relevance and coverage of the search results that the users need compared with traditional search engines.**

***Index Terms*—clustering search engine, meta-search engine, document clustering.**

I. Introduction

Most people heard of the story about how successful Yang and Yahoo! are and how many of the computer companies are started at a garage or the like. In general, one of the key success factors of these companies is in that they find and create a useful and easy way for the internet user to access all the information available in the Internet. Using directory to search for the information a user needs is certainly one of the most popular services at the initial stages of Yahoo!. The advantage of directory search is in that it can provide more relevant information than search engines because the information is created by human beings instead of by search engines. However, the disadvantage is in that it may not be able to provide the most up-to-date information available in the Internet because it may take time for the human beings to update the directory.

Nowadays, most users of the Internet use search engine to find the information they need. This is because search engine can provide users of the Internet a great deal of information for which they are looking than directory search can provide. Lawrence [1] pointed out that there are *not* even a single search engine that can cover more than 16% of the information available in the Internet. But, in the daily life, we usually still use GYM (Google, Yahoo!, and MSN) or other search engines to find the information we need. The reality is that even though all of the search engines can only cover a low percentage of the information in the Internet, the users of the Internet can generally still find the information they need even if it takes time to filter out all the redundant information. This can be justified by the observation that most of the web pages collected by web spider or internet robot are not relevant to the query of the user. In addition, the available information is way too much to be handled efficiently, and most of the available information are irrelevant to the user. On the other hand, the response time, or the number of web pages needed to be transmitted, has become not as critical as it used to be because the network bandwidth and the storage technologies have been significantly improved. Instead, the most important question is how to present to the user of the Internet precisely the information they need.

We discussed a very simple example in our previous work [2], but the problem still exists in Google. The problem is when the keyword "cat" (meaning an animal) is given as a query to Google,[1] the first item returned is the company "Caterpillar: Home," which has nothing to do with the animal "cat." Of course, one of the possible reasons may be that the company paid Google for the advertisement so that their rank is set to the highest. However, the strange phenomenon also occurs in Yahoo! (with the second highest rank)[2] and MSN (with the highest rank too).[3] Now, it is possible that the company paid all of these three search engine companies. But an observation of these search engines shows that the web site of "Centre for Alternative Technology" is ranked the third on Google and MSN. No doubt, it really happens on these famous search engines. Under some circumstances, it is a waste of our time for trying to filter out the irrelevant information

[1] http://www.google.com/search?hl=en&q=cat, March 5, 2009.
[2] http://search.yahoo.com/search?p=cat, March 5, 2009.
[3] http://search.msn.com/results.aspx?q=cat, March 5, 2009.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

when such a search engine is used. But most of the time, it is the search engine, not the company, to be blamed for returning the irrelevant information, because most of the web sites use keywords or the like to ensure that the search engine ranking mechanism will put them on top. For example, the web site designer can add more keywords to make a search engine believe that this web site is highly related to the information the user is looking for, but their goal is usually to elicit more people to visit their web site. For these reasons, the cluster search engine provides another way to find the information, by returning a set of classified web pages to the user.

Fig. 1. The result of Kartoo.

Recently, many cluster search engines have been developed and put into use, which include Vivisimo [3], SnakeT [4], iBoogie [5], Kartoo [6], Copernic [7], Grokker [8], Dogpile [9] and Webclust [10]. It is worth mentioning that Kartoo (as shown in Fig. 1) and Grokker provide an interesting graphical interface to help the user understand the relations between the returned web pages. In [11], [12], [13], [14], [15], [16], more and more researchers are focused on improving the clustering capability of a cluster search engines (see Fig. 2). In [11], Paolo 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.

In this paper, we present an intelligent cluster search engine system, called ICSE. Not only do we focus on a novel and fast clustering algorithm, but we also apply the proposed algorithm to the search engine system we implemented. The clustering algorithm is built on a meta-directory tree that can not only enhance the relevance of the search result but it can also save the computation time.

The remainder of the paper is organized as follows. In Section II, we fist gives a detailed description of the proposed algorithm. Then, we explain how the system is designed and implemented. Finally, an example is given to show how the proposed system works. Section III presents the simulation result. Conclusion is given in Section IV.

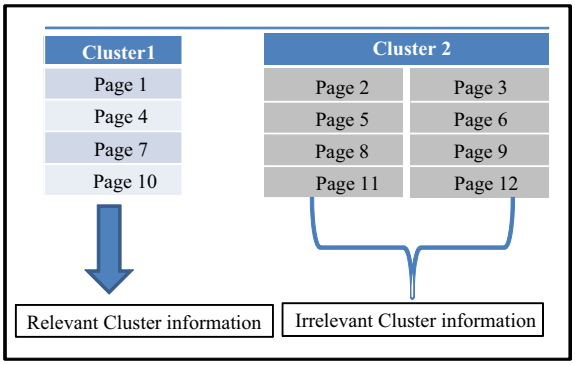

Fig. 2. Example showing how a clustering search engine filters out the irrelevant information.

II. The Proposed Method

In this section, we will give an introduction to the clustering algorithm on ICSE, called CA-ICSE. Then, we explain how ICSE is designed and implemented and how the proposed clustering algorithm is applied. Finally, an example is given to show how the system works and what it does.

A. The CA-ICSE Algorithm

```
1  begin
2    Let ℓ = 0;
3    for each web page D_i ∈ W do
4      for (ℓ ≠ L) do
5        for each node n_j^ℓ ∈ T do
6          Compute the similarity between D_i and n_j^ℓ using TF−IDF [17];
7        end for
8        Choose n_j^ℓ, which is most similar to D_i;
9        ℓ = ℓ + 1;
10     end for
11     Let ℓ = 0;
12   end for
13 end
```

Fig. 3. Outline of the CA-ICSE algorithm.

Fig. 3 gives an outline of the CA-ICSE algorithm. The input of the algorithm is a set of web pages and a meta-directory tree. The output is a classified set of the input web pages.

In Fig. 3, $W = \{D_1, D_2, \ldots, D_m\}$ represents the set of the input web pages where m is the number of web pages in W. $\mathcal{T}$ is the meta-directory tree. L is the maximal level of the meta-directory tree $\mathcal{T}$. In other words, $\mathcal{T}^\ell = \{n_1^\ell, n_2^\ell, \ldots, n_{k_\ell}^\ell\}$ where k_ℓ is the number of nodes at level ℓ where ℓ indicates the current level in $\mathcal{T}$ and n_j^ℓ represents the jth node at level ℓ. The CA-ICSE algorithm is used to find the most suitable taxonomy for all the web pages in W. Basically, it contains two major steps: (1) computing the similarity and (2) clustering the web pages based on the similarity computed in step (1). More precisely, as Fig. 3 shows, CA-ICSE start off setting ℓ to 0 on line 2 so that the search will begin with the root of the meta-directory tree $\mathcal{T}$. Then, for each web page D_i, the loop beginning on line 3 and ending on line 12 computes its

similarity with the nodes in $\mathcal{T}$. This loop will compute the similarity between D_i and every node in $\mathcal{T}$ level by level until the leaf is reached. At each level, the proposed algorithm will choose the node that is most similar to D_i to be the sub-category of D_i. The process will be repeated until the leaves of $\mathcal{T}$ are visited and the categories of all the web pages are determined.

In addition, the CA-ICSE algorithm can be used to reduce the clustering time of cluster search engine system. For example, as Fig. 4 shows, if there are 4 nodes at each level in the meta-directory tree, then the CA-ICSE algorithm can eventually cut the search space down to 25% of the whole search space because we only look at the most similar node and then iterate. In other words, we only look at 4 nodes at each level instead of all the nodes at each level. For this reason, this clustering algorithm can find the suitable categories (nodes) rapidly.

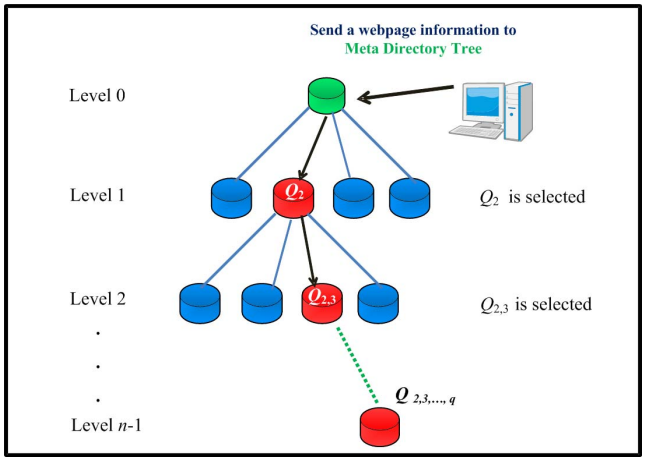

Fig. 4. Using CA-ICSE to classify the web pages and reduce the search space.

B. Design and Implementation of the System

The ICSE, as we described herein, is composed of the following four modules: (1) Meta-search engine, (2) Meta-directory tree, (3) Web pages clustering, and (4) Topic generation. Web pages clustering has been described earlier. In what follows, we will describe all of the remaining modules in turn.

An overview of the ICSE is shown in Fig. 5. The data grid and grid computing shown in the figure can be used to enhance the performance of the system. However, they are beyond the scope of the paper, and thus we will not discuss them any further. In addition, if a query, say, "oasis" is given to the meta-search engine for the second time, the ICSE will rapidly respond to the user by showing the cached results of the meta-search engine.

1) Meta-search engine: In this module, The extraction technology as described in [18] is used to parse the web pages and analyze the HTML tags. Then, we design a parser

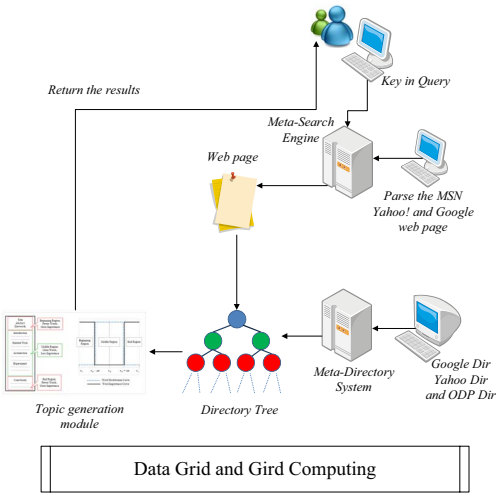

Fig. 5. Framework of the ICSE.

to extract the search results. By using the Stemmer [19] to discard the common morphological and inflectional endings and Stopword [20] to discard worthless words, the web pages will then be converted to a unified format. For example, when a query is given to the meta-search engine, the meta-search engine will first send the query to the underlying search engines, and then the web pages returned by the underlying search engines are collected. The ICSE uses the retrieval rules to extract the information in each web page. Fig. 6 shows the interface of ICSE.

Fig. 6. The user interface of ICSE.

2) Meta-directory tree: In order to cluster the returned web pages rapidly, we propose a novel clustering algorithm, which uses meta-directory trees as the knowledge base for both reducing the computation time required for clustering and enhancing the quality of the clustering results. The proposed algorithm uses an off-line processing to create and maintain the meta-directory tree. Then, the web pages can be rapidly classified into the most similarity clusters on-line. In this research, we design a web page *parser* to get the information that ICSE needs. The parser extracts the directory information, which includes web site, title, hyper link, abstract, ontology information, and relationship. All of the extracted directory

information can be used by the ICSE to efficiently classify the web pages. The title, link, and abstract are essentially the same as those of the search results. The directory information is used to keep track of the category to which the web page belongs. The association with other web pages can be easily understood using the relationship. The ODP offers a massive of text files that contains about 5 million hyper link records and more than 600,000 classified catalogs. In this research, we extract more than 10,000,000 (or 6GB) web pages from these three directory systems.

3) Web pages clustering: Section II-A gives a detailed description of the proposed clustering algorithm CA-ICSE. The underlying concept of CA-ICSE is to use directory trees as a knowledge base to find the taxonomy of the input data quickly. In other words, traditional clustering and classification technologies of data mining classify data without a knowledge base. For this reason, it generally takes a lot of computation time to find the classified results. Another problem is how each group of the classified results is labeled once it is classified. Because of these problems, the proposed algorithm uses a directory-tree based approach, which can not only cluster the web pages quickly but also assign a more meaningful label to each group of the classified results. Moreover, CA-ICSE can greatly reduce the computation time required by the clustering module of a clustering search engine, thus making it much more scalable for large data sets.

4) Topic generation: The topic generate module [21] assumes that the words in the web page at the beginning and end parts are more important than those in the middle part. After the web pages are split and the scores are computed, the module selects some of the lexicons that are more important than the others to be the topic.

C. An Example

In this section, an example is given. Fig. 7 shows that when a query *oasis* is given to the ICSE system, then the following steps will be taken.

1) Meta-search engine module will send the query *oasis* to the meta-search engine (e.g., Google) to collect and extract the web pages W.
2) ICSE will then use the information available in the meta-directory tree $\mathcal{T}$ to cluster the set of web pages W into a set of classified web pages, which is referred to as R here.
3) Topic generation module will summarize the sets R and W to get the top 10 keywords, denoted T_R.
4) Finally, T_R will be returned to the user.

The result of sending the query *oasis* to our ICSE system is shown in Fig. 7. Fig. 7(a) shows the subgroup information of the clustering and summarization results. Fig. 7(b) shows the group information of the ICSE system upon receiving the query *oasis*. In addition, the keywords in Figs. 7(a) and (b) represent the topic words chosen to describe the information of each group.

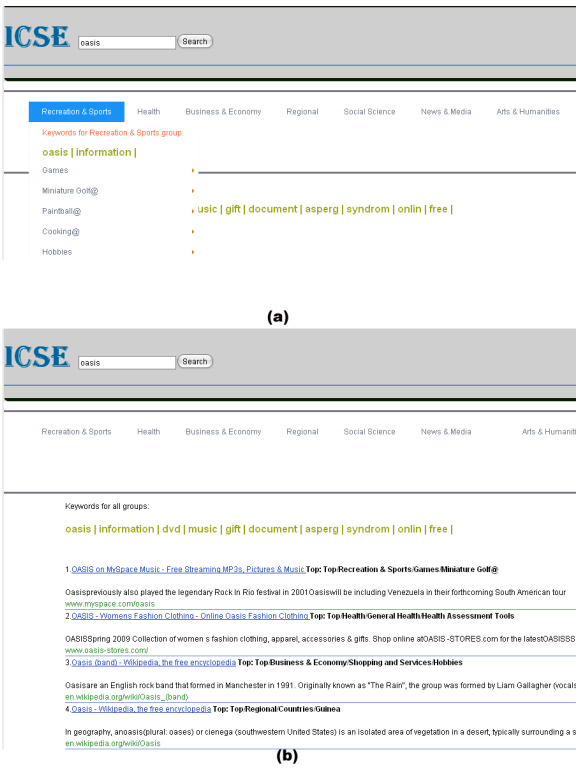

Fig. 7. An example illustrating how topic generation module works.

III. Experimental Result

A. Environment Setting

In this section, we evaluate the performance of the ICSE system. All experiment were performed on a 3.4GHz Intel Pentium D CPUs ASUS desktop with 1GB main memory and 160 GB hard disk running on Windows XP. All of the programs were written in Java (Java Servlet). The proposed system is available online at [22], [23]. We also use the Vivisimo [3] and iBoogie [5] for the purpose of comparison.

B. Experimental Results

The performance of the ICSE is estimated by using several common queries. Table I shows the total response time of twenty queries. The second and third column show the time taken, respectively, by the meta-search engine and the clustering module. The response time is defined to be the time elapsed when a query is sent to the ICSE for the first time and the response is received by the user. Our experimental results show that the average of the response time of the twenty queries is about $396.85\text{ms} + 108.9\text{ms} = 507.55\text{ms}$.

Table II compares the response time of each item[4] returned by Vivisimo, iBoogie, and the ICSE system. The response time

[4]The total response time divided by the number of items returned.

TABLE I
THE RUNNING TIME OF THE META-SEARCH ENGINE MODULE AND THE CLUSTERING MODULE OF THE PROPOSED SYSTEM IN RESPONSE TO A QUERY

QUERY	MetaSE Response Time	Clustering Time
amazon	391ms	125ms
apple	391ms	109ms
american eagle	406ms	125ms
Chien-Ming Wang	390ms	110ms
diesel	578ms	94ms
hamburger	453ms	125ms
IEEE	375ms	110ms
IEEE SMC	422ms	93ms
java	406ms	109ms
Lena	250ms	81ms
msn	235ms	110ms
oasis	422ms	141ms
Obama	437ms	78ms
shell	406ms	109ms
suede	375ms	110ms
taiwan	407ms	110ms
thread	485ms	141ms
USC	468ms	110ms
warcraft	250ms	94ms
wow	390ms	94ms
AVERAGE	396.85ms	108.9ms

MetaSE indicates Meta-Search Engine.

TABLE II
THE RESPONSE TIME OF THE PROPOSED SYSTEM, VIVISIMO, AND IBOOGIE IN RESPONSE TO A QUERY

QUERY	System Response Time			
	Vivisimo	iBoogie	$ICSE_1$	$ICSE_2$
amazon	4.41ms	16.50ms	20.64ms	1.19ms
apple	4.82ms	15.73ms	20.83ms	1.29ms
american eagle	7.51ms	17.38ms	23.09ms	1.38ms
Chien-Ming Wang	5.46ms	17.86ms	20.00ms	1.29ms
diesel	5.92ms	13.89ms	28.00ms	1.35ms
hamburger	4.38ms	17.65ms	25.13ms	1.95ms
IEEE	9.79ms	17.49ms	20.21ms	1.24ms
IEEE SMC	5.71ms	16.97ms	20.60ms	1.24ms
java	5.21ms	13.56ms	20.60ms	1.31ms
Lena	3.94ms	16.16ms	13.24ms	1.38ms
msn	6.20ms	16.72ms	13.80ms	1.23ms
oasis	9.37ms	13.63ms	23.46ms	1.28ms
Obama	11.38ms	21.76ms	22.39ms	1.29ms
shell	5.86ms	16.39ms	20.60ms	1.24ms
suede	4.70ms	16.45ms	21.09ms	1.34ms
taiwan	4.35ms	17.06ms	20.68ms	1.09ms
thread	4.27ms	13.53ms	26.08ms	1.27ms
USC	9.76ms	15.08ms	24.08ms	1.29ms
warcraft	4.25ms	13.69ms	15.64ms	1.48ms
wow	8.84ms	13.76ms	20.17ms	1.30ms
AVERAGE	6.31ms	16.06ms	21.02ms	1.32ms

of the ICSE can be divided into two parts: $ICSE_1$ and $ICSE_2$. $ICSE_1$ indicates the response time the query is given to the ICSE for the first time while $ICSE_2$ indicates the response time the same query is sent to the ICSE the second time or onward.

From the results, we can easily see that $ICSE_2$ is more than about 5 up to 20 times faster than other clustering search engines for the ICSE system will cache the clustering results in the knowledge database. Thus, when the same query is sent to ICSE, the response is almost instant.

IV. CONCLUSION

In this paper, we proposed a novel Intelligent Cluster Search Engine System, called ICSE, for online information search. This system combines the information coverage provided by metasearch engine systems and the relevance of information offered by directory search systems. In addition, we presented in detail how the system is designed and implemented. We also proposed a simple but fast algorithm for clustering the web pages that can handle the large scale clustering problem. In particular, the algorithm is fundamentally different from traditional clustering algorithms such as k-means that require a tremendous amount of computation time. For this reason, the tree search concept is employed by this algorithm, and it can significantly reduce the computation time required for clustering. In the future, our goal is to improve both the coverage rate and the relevance rate and add the concepts of ontology and web 2.0 to our system.

ACKNOWLEDGMENT

This work was supported in part by National Science Council, R.O.C., under Contact No. 95-2221-E-006-509-MY2.

REFERENCES

[1] S. Lawrence and C. L. Giles, "Accessibility of information on the web," *Nature 400*, pp. 107–109, 1999.
[2] 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.
[3] Vivisimo. [Online]. Available: http://search.vivisimo.com/
[4] Snaket. [Online]. Available: http://snaket.di.unipi.it/
[5] Iboogie. [Online]. Available: http://www.iboogie.com/
[6] Kartoo. [Online]. Available: http://www.kartoo.com/flash04.php3
[7] Copernic. [Online]. Available: http://find.copernic.com/copernic.html
[8] Grokker. [Online]. Available: http://www.grokker.com/
[9] Dogpile. [Online]. Available: http://www.dogpile.com/
[10] Webclust. [Online]. Available: http://www.webclust.com/
[11] 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.
[12] 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.
[13] 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.
[14] 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.
[15] F. Giannotti, M. Nanni, D. Pedreschi, and F. Samaritani, "Webcat: Automatic categorization of web search results," *Proceedings of SEBD'2003*, pp. 507–518, 2003.
[16] S. Osinski and D. Weiss, "Conceptual clustering using lingo algorithm: Evaluation on open directory project data," *Intelligent Information Systems*, pp. 369–377, 2004.
[17] R. Baeze-Yates and B. Ribeiro-Neto, "Modern information retrieval," *ACM Press*, 1999.
[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] M. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.
[20] Stop word list. [Online]. Available: http://www-fog.bio.unipd.it/waishelp/stoplist.html

[21] H. Y. Hsu, C. W. Tsai, M. C. Chiang, and C.-S. Yang, "Topic generation for web document summarization," *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3702–3707, 2008.
[22] The icse system. [Online]. Available: http://140.116.216.156/se
[23] Introduction to the icse system. [Online]. Available: http://cwtsai.ee.ncku.edu.tw/NSNP.htm