# Multiple factor hierarchical clustering algorithm for large scale web page and search engine clickstream data

**ARTICLE** *in* ANNALS OF OPERATIONS RESEARCH · AUGUST 2010

Impact Factor: 1.22 · DOI: 10.1007/s10479-010-0704-3



**2 AUTHORS**, INCLUDING:

Gang Kou
Southwestern University of …

**171** PUBLICATIONS **1,950**
CITATIONS

SEE PROFILE

Ann Oper Res
DOI 10.1007/s10479-010-0704-3

# Multiple factor hierarchical clustering algorithm for large scale web page and search engine clickstream data

**Gang Kou · Chunwei Lou**



**Abstract** The developments in World Wide Web and the advances in digital data collection and storage technologies during the last two decades allow companies and organizations to store and share huge amounts of electronic documents. It is hard and inefficient to manually organize, analyze and present these documents. Search engine helps users to find relevant information by present a list of web pages in response to queries. How to assist users to find the most relevant web pages from vast text collections efficiently is a big challenge. The purpose of this study is to propose a hierarchical clustering method that combines multiple factors to identify clusters of web pages that can satisfy users' information needs. The clusters are primarily envisioned to be used for search and navigation and potentially for some form of visualization as well. An experiment on Clickstream data from a processional search engine was conducted to examine the results shown that the clustering method is effective and efficient, in terms of both objective and subjective measures.

**Keywords** Information retrieval · Web page clustering · Multiple criteria decision making · Multiple factor hierarchical algorithm · Clickstream analysis · K-means algorithm

## 1 Introduction

The idea of automatically retrieving useful information from large amounts of electronic text was initiated by Vannevar Bush in his seminal essay "As We May Think" (Bush 1945). From 1950s to 1980s, various information retrieval models and techniques have been developed in the field of Information Retrieval (IR) (Singhal 2001). Information retrieval is defined as "finding material of an unstructured nature (usually text) that satisfies an information need from within large collections" (Manning et al. 2008, p. 3).

The developments in World Wide Web and the advances in digital data collection and storage technologies over the past two decades enable companies and organizations to store

G. Kou (✉) · C. Lou
School of Management and Economics, University of Electronic Science and Technology of China, Chengdu 610054, P.R. China
e-mail: kougang@yahoo.com

Published online: 14 February 2010

up huge amounts of electronic web pages (Heer and Chi 2001). The availability of real-life large collections of electronic text presents opportunities and challenges for those previously developed IR models and techniques (Foss et al. 2001). On the one hand, the abilities of these models and techniques to scale to very large collections of web pages can be evaluated (Shi 2009; Hong et al. 2008); on the other hand, deficiencies of many existing techniques were identified and the demands for new techniques that can effectively retrieve relevant web pages over large corpora are increasing (Gómez et al. 2008).

A Clickstream is the data stream generated when a computer user browses the Internet (Hu and Zhong 2008). Analyzing the clickstream helps us to understand users' information needs and provide them better browsing experiences. Data mining and machine learning techniques have been applied to record and analyze clickstream data (Nasraoui et al. 2003).

As an unsupervised learning method, clustering is used in many fields, such as statistics, data mining, machine learning, and bioinformatics. When clustering is used in information retrieval, the assumption is that web pages clustered together behave similarly with respect to relevance to information needs (Van Rijsbergen 1979; Cutting et al. 1992; Steinbach et al. 2000; Dhillon et al. 2001). Some typical applications of clustering in information retrieval include search result clustering, corpus analysis, scatter-gather, language modeling, and cluster-based retrieval (Manning et al. 2008; Lee and Lee 2008). Clustering can help information retrieval systems retrieve and rank web pages by grouping similar web pages into clusters. When the number of web pages is very large, many popular clustering algorithms, such as the k-means which require extensive computing resources to do the analysis, may fail to find optimal solutions. The goal of this study is to propose a multiple factor hierarchical clustering method that is able to retrieve relevant web pages from large corpora efficiently.

The paper is organized as follows. Section 2 introduces the clickstream data collected from a professional search engine. Section 3 describes the definition of the web page clusters in search engine from business perspective. Section 4 proposes a multiple factor hierarchical clustering algorithm for large scale text dataset. Section 5 presents an experiment that examines the clustering algorithm on clickstream data and evaluates the performance of the algorithm using both objective and subjective measures. The last section concludes the paper with discussions and future research directions.

## 2 Clickstream data collection

### 2.1 Data collection and statistics

The Clickstream data that used for this research was collected during a six-month-period in 2007 from a professional search engine's daily log. The data collected includes user queries, clicks, user sessions, search results, and web pages. This data includes approximately 100 million clicks on 6 million pages in about 20 million search sessions. In order to understand the Clickstream data, identify its quality, and discover insights into the data (CRISP-DM 1996), several statistical tests (Zhang and Segall 2008) were preformed on the data and the results are shown in Tables 1, 2, and 3.

Table 1 lists overall web page popularity as measured by number of clicks. Take the bolded row as an example. It represents pages that were clicked between 67 and 101 times. Notice that this group represents 2% of all pages, (i.e. 121,280 pages) and generates 10.15% of all clicks (i.e. 10,192,299 clicks). It also indicates that pages clicked 67 times or more represent 4.45% of all pages (i.e. 269,255 pages) and generates 46.97% of all clicks (i.e.

**Table 1** Page-click analysis

| Num clicks for pages in group | | Web page-centric statistics | | | | Click-centric statistics | | | |
|---|---|---|---|---|---|---|---|---|---|
| Minimum | Maximum | Number pages | Percent pages | Cumm. number pages | Cumm. percent | Number clicks | Percent clicks | Cumm. number clicks | Cumm. percent |
| 1 | 7 | 3963182 | 65.46% | 6053969 | 100.00% | 11147749 | 11.10% | 100385982 | 100.00% |
| 8 | 16 | 930467 | 15.37% | 2090787 | 34.54% | 10946635 | 10.90% | 89238233 | 88.90% |
| 17 | 28 | 449324 | 7.42% | 1160320 | 19.17% | 10729554 | 10.69% | 78291598 | 77.99% |
| 29 | 43 | 272123 | 4.49% | 710996 | 11.75% | 10212480 | 10.17% | 67562044 | 67.30% |
| 44 | 66 | 169618 | 2.80% | 438873 | 7.25% | 10196448 | 10.16% | 57349564 | 57.13% |
| **67** | **101** | **121280** | **2.00%** | **269255** | **4.45%** | **10192299** | **10.15%** | **47153116** | **46.97%** |
| 102 | 158 | 73791 | 1.22% | 147975 | 2.45% | 10116694 | 10.08% | 36960817 | 36.82% |
| 159 | 274 | 46214 | 0.76% | 74184 | 1.23% | 10087844 | 10.05% | 26844123 | 26.74% |
| 275 | 700 | 23344 | 0.39% | 27970 | 0.47% | 10067569 | 10.03% | 16756279 | 16.69% |
| 701 | 37246 | 4626 | 0.08% | 4626 | 0.08% | 6688710 | 6.66% | 6688710 | 6.66% |

**Table 2** Session-click analysis

| Num clicks for sessions in group | | Session-centric statistics | | | | Click-centric statistics | | | |
|---|---|---|---|---|---|---|---|---|---|
| Minimum | Maximum | Number sessions | Percent sessions | Cumm. number sessions | Cumm. percent | Number clicks | Percent clicks | Cumm. number clicks | Cumm. percent |
| 1 | 2 | 11635881 | 55.66% | 20905303 | 100.00% | 13945321 | 13.89% | 100385982 | 100.00% |
| 3 | 4 | 3482596 | 16.66% | 9269422 | 44.34% | 12118974 | 12.07% | 86440661 | 86.11% |
| 5 | 7 | 2426386 | 11.61% | 5786826 | 27.68% | 14231937 | 14.18% | 74321687 | 74.04% |
| 8 | 10 | 1175917 | 5.62% | 3360440 | 16.07% | 10517086 | 10.48% | 60089750 | 59.86% |
| **11** | **15** | **939600** | **4.49%** | **2184523** | **10.45%** | **11736640** | **11.69%** | **49572664** | **49.38%** |
| 16 | 22 | 592194 | 2.83% | 1244923 | 5.96% | 10969981 | 10.93% | 37836024 | 37.69% |
| 23 | 35 | 383262 | 1.83% | 652729 | 3.12% | 10753685 | 10.71% | 26866043 | 26.76% |
| 36 | 75 | 221345 | 1.06% | 269467 | 1.29% | 10490678 | 10.45% | 16112358 | 16.05% |
| 76 | 985 | 48122 | 0.23% | 48122 | 0.23% | 5621680 | 5.60% | 5621680 | 5.60% |

**Table 3** Session-search analysis

| Range of session length | Number of sessions | Percent of total sessions | Average session length | Avg number of searches | Percent of total searches |
|---|---|---|---|---|---|
| 0–11 secs | 4165037 | 19.92% | <4 secs | 1.07 | 2.55% |
| 11–203 secs | 4197045 | 20.08% | 89 secs | 2.11 | 5.07% |
| 204–637 secs | 4181457 | 20.00% | 407 secs | 5.87 | 14.06% |
| 638–1774 secs | 4181376 | 20.00% | 1089 secs | 11.23 | 26.90% |
| 1775–37231 secs | 4180388 | 20.00% | 5336 secs | 21.47 | 51.41% |

47,153,116 clicks). In this search engine, about 35% of the search results generate 90% of the total clicks.

Suppose that each search session represents a user of the search engine, the session-click statistics show different ways that users may use the search engine. Table 2 lists the overall session-click relationship. Take the bolded row as an example. It represents sessions with clicks between 11 and 15 times. This group represents 4.49% of all sessions (i.e. 939,600 sessions) and generates 11.69% of all clicks (i.e. 11736640 clicks). Although sessions with 11 and more clicks account for only 10.45% of sessions (i.e. 2,184,523 sessions), they represent 49.38% of all clicks (i.e. 49572664 clicks). In this search engine, about 28% of the sessions generate 74% of clicks.

Table 3 groups users by session length. It records the number of sessions, average session length, and average number of searches for each group. In addition, it shows how many percent of the total sessions and total searches each group represents.

Notice that the shortest user sessions (i.e. 0–11 seconds) accounted for 19.92% of the total sessions, but constituted only 2.55% of the total searches. The longest user sessions (i.e. 1,775–37,231 seconds) represented 20% of the total sessions, but accounted for more than half of the total searches.

From Tables 1, 2, and 3, we observed the following characteristics of this clickstream dataset:

1. There are more than 6 million web pages viewed for about 100 million times in 20 million user search sessions within the six month period.
2. 65.46% of the pages have seven or fewer clicks.
3. 55.66% users (i.e. sessions) have two or fewer clicks.
4. The most active 10% of users generate almost half of all clicks.
5. The most popular 2% of pages generate about 47% of clicks.
6. The average number of searches in each user session is 8.35 and the average session length is about 1,385 seconds.

## 3 Business understanding and cluster definition

In a data mining project, it is very important to understand the project objectives and requirements from a business perspective (CRISP-DM 1996). In order to find the good quality web page clusters, we need to define cluster in search domain from a business perspective. In this study, domain experts of the search engine provide four major requirements to assist the definition of a web page cluster.

First, each cluster should consist of the web pages from the result of the same search query and represent an interesting topic. Second, clusters are generated in advance or during run-time and web pages in the results of a search engine are assigned to relevant clusters. Membership in a cluster could be fixed or fuzzy/probabilistic. Third, within a cluster, users can use filters/facets to narrow Web pages. A single clustering scheme assumes that there exists a comprehensive set of clusters that are truly useful to a user. However, there is significant risk that the user's information need depends on other dimensions that are not represented in the cluster organizing scheme. Fourth, as web pages are added or updated, each page must be assigned to at least one cluster. This typically involves a comparison between a web page and a cluster's centroid.

Web pages could also be added based on other criteria. Following is a list of detailed requirements and implementation notes:

– For any given link there will be one query to get the nearby cluster IDs, and then another query to get the web page IDs after the user clicks on a link.
– When showing the list of nearby clusters, or selecting a default cluster to show, the application may want to order the results based on the web page's similarity to clusters, or perhaps based on some other cue in the user's session.
– Initial creation of the clusters would be done using a clustering algorithm and similarity measure. The approach will determine how hard it is to create an effective label and description.
– With multiple cluster universes, each universe has a different organizing principle and similarity measurement. This approach will enable the search results to model web page spaces that reflect the multiplicity of users' information needs.
– Pre-generated clusters make it easier for editors or users to enhance the cluster by rating cases or adding comments/summaries. An editor could also manually add web pages to a cluster.
– Over time, the addition of new web pages can cause the cluster to "shift", so the clusters originally created may become obsolete or inaccurate. These issues suggest we need the ability to:
  ○ Identify when we should re-cluster.
  ○ Re-cluster and generate labels and summarizations without losing all the editor or user added enhancements.
  ○ Compare the new scheme with the old one.

## 4 Multiple factor hierarchical clustering method for large scale text corpus

Current information in an IR system is stored in electronic web page repositories and retrievable only by text search. With the increase in the number of electronic web pages, it is hard to manually organize, analyze and present these web pages efficiently (Peng et al. 2008a). Two important criteria that have been widely used to measure the quality of an IR system are the relevance of returned results and the length of waiting time (Al-Aomar and Dweiri 2008). Clustering can group web pages into clusters that are coherent internally and different externally. The clusters can be used for search, navigation, and a potential form of visualization. Major clustering approaches can be categorized into partitioning, hierarchy, density-based, grid-based, and model-based algorithms (Han and Kamber 2006). This section describes some concepts and techniques that have been used in this paper. In particular, selected concepts and techniques of text preprocessing, retrieval model, and a clustering method are discussed in sequence.

### 4.1 Text preprocessing: tokenization, stop-words, and stemming

The goals of text preprocessing are to represent full-text web pages (Cooley et al. 1999) in a suitable format and to optimize the performance of text mining algorithms by discarding irrelevant data (Mathiak and Eckstein 2004). Text web pages are first divided into a set of *index terms* or *keywords*. This division process is called *tokenization*. These index terms are then used to represent full-text web pages (Zhang et al. 2009), which refer to the titles and abstracts in this study. Different index terms have varying importance and this difference is expressed using *weight*s. Each keyword in a web page is associated with a weight. In this paper, a self-developed C++ program was used to tokenize article titles and abstracts into keywords.

Tokenization normally results in thousands or even tens of thousands of keywords. Among these keywords are some common words that do not contribute to the retrieval task and thus should be removed from the keyword list. Stop-words and stemming are two prevalent keywords reduction methods. Regardless of topics and research areas, there are always common words that occur frequently in all web pages, such as articles, prepositions, and conjunctions. This type of words is refer to as *Stop-words* and is irrelevant for the purpose of retrieval. A *stem* is the portion of a word which is left after the removal of its affixes, i.e., prefixes and suffixes. Porter's stemming algorithm (Porter 1980) has been widely used and was selected by this project to further reduce the dimensionality.

### 4.2 Retrieval model: vector space model

After converting full-text web pages into a set of keywords and reducing indexing structure, retrieval models can be set up. In information retrieval, retrieval models can be classified into three categories: *Boolean model*, *vector space model*, and *probabilistic model*. *Boolean model* considers index terms are either present or absent in a web page, and hence can not recognize partial matches. *Vector space model* represents text using a vector of terms (Salton et al. 1975). *Probabilistic IR model* estimates the probability of relevance of web pages for a query based on the probabilistic principle (Singhal 2001). For different experiments, probabilistic model and vector space model may have different performance. Nevertheless, it has been shown that the vector space model is expected to outperform the probabilistic model with many cases.

According to *vector space model*, a web page can be represented as a vector:

$$\langle (d_{r1}, w_1), (d_{r2}, w_2), (d_{r3}, w_3), \ldots, (d_{rn}, w_n) \rangle,$$

where $d_{ri}$ denotes a keyword $i$ used to describe the web page $r$, and $w_i$ denotes the weight of the keyword $i$, which can be determined by frequency of use. A collection of $n$ web pages can be represented by a *term-web page matrix*. An entry in the matrix corresponds to the weight of a term in that web page; zero means the term doesn't exist in the web page or has no significance in the web page (Baeza-Yates and Ribeiro-Neto 1999).

Researchers have developed various weighting schemes to calculate weights of terms. A popular term weight is *tf-idf* weighting: $w_{ij} = tf_{ij} \times idf_i$, where $tf_{ij}$ is *term frequency* across the entire corpora: $tf_{ij} = f_{ij} / \max\{f_{ij}\}$ and $f_{ij}$ is the frequency of term $i$ in web page $j$; $idf_i$ is the *inverse web page frequency* of term $i$: $idf_i = \log_2(N/df_i)$, $N$ is the total number of web pages and $df_i$ is the web page frequency of term $i$, i.e. the number of web pages containing term $i$. A term in a web page with higher *tf-idf* weight is regarded as more indicative than terms with lower *tf-idf* weights. The reasoning behind *tf-idf* weighting is that a term occurring frequently in a web page but rarely in the rest of the collection is indicative. In this paper, the frequencies of each index term within each abstract were counted using SQL and *tf-idf* weights were computed and stored in a term-web page matrix.

### 4.3 The multiple factor hierarchical clustering algorithm

When dealing with very large corpus, clustering methods like the k-means (McQueen 1967) may fail to find optimal solutions because it requires extensive computing resources to do the analysis. As Hearst pointed out in his paper—*untangling text data mining*, a mixture of computationally-driven models and user-guided analysis may open the door to exciting new results (Hearst 1999, p. 6). Han and Kamber provide a comprehensive introduction of contemporary clustering techniques (Han and Kamber 2006). This study proposes a clustering

method that combines a hierarchical clustering algorithm and users' previous search behavior. It is a multiple factor clustering algorithm because it clusters web pages using not only the text similarities among the web pages, but also the user browsing activities in search session. Web pages were gathered from an anonymous text database together with web pages retrieval history (Park et al. 2007). For each index term remains, *tf*, *idf*, and *tf-idf* weights were calculated. Based on *tf*, *idf*, and *tf-idf* weights, a term-web page matrix was created. Based on search history, frequently retrieved web pages were selected as initial centers of clusters. The top *m* percent of web pages that are similar to a center web page are assigned to that cluster. Then the center of each cluster can be determined by calculating the mean value of the web pages for each cluster. A web page that has not been assigned to any cluster is associated to a particular cluster that has the shortest distance between the center of that cluster and the web page. This method is able to find optimal solutions for very large text collections because it reduces the computation time by including only the top *m* percent of web pages in the calculation of the centers of clusters. Inspired by Clustering by Committee algorithm (Pantel 2003), we propose the multiple factor hierarchical clustering algorithm for Clickstream data as follow:

*Input*: a list of web pages D to be clustered, a list of most clicked/viewed or editors/users specified web pages CD, a list of cluster committees CC

*Step 1*: Given a list of web pages D to be clustered, a list of most clicked/viewed or editors/users specified web pages CD, for each web page c in CD, compute the pairwise similarity between c and the web pages in D that share some given features such as co-citation, co-view, co-keyword. Thus we compute and find the top-k similar web pages of all web pages in CD.

*Step 2*: For each web page c in CD, cluster the top-k similar web pages of c using tf-idf clustering;

Compute the score = the size of web pages in each cluster discovered × average pairwise similarity between web pages in that cluster;

Store the cluster cl with the highest score in a list CL;

Sort the clusters in CL in descending order of their scores.

*Step 3*: Let CC be a list of cluster committees, initially empty.

For each cluster cl in CL, Compute the centroid of cl by averaging the frequency vectors of its web pages

If cl's centroid is not close to any of the committees previously added to CL by certain measure threshold, add cl to CL.

*Step 5*: For each web pages d in D, if d is not close to any committee in CL, add d to a list of residues R

*Step 6*: If R is empty, we are done and return CL. Otherwise, return the union of CL and the output of a recursive call to Step 1 using the input CC and replacing D with R and CD with a list of most clicked/viewed web pages CR in R.

*Output*: a list of cluster committees CC.

The list of cluster committees CC can be manually edited or tagged by editors. The number of clusters is not fixed. When there are web pages that can't be fit into any cluster, re-run the algorithm and new cluster committee will be generated if there is any. The re-cluster process will not lose any enhancements added by the editor or user since cluster committees CC will be part of the input of the process.

With the list of cluster committees CC, a future cluster request of web pages becomes a categorization problem and the incoming web pages will be assigned to its closest cluster committees. The assignment feature of this algorithm is similar to the K-means method,

however, it differs from the k-means in the way of calculating centroids of clusters. The centroid of each cluster committee in the proposed algorithm does not change when a new web page is assigned to that cluster because a newly assigned web page is not added to the cluster committee.

This algorithm works like this: Every topic/cluster is represented by a committee of web pages. Committees are generated by seed pages that are pre-labeled or defined as the most clicked pages from search history. We are looking for a group of all web pages that are related to a topic or cluster and we know there exists a seed page that must belong to that group. The group of web pages shares one of the following three relationships with a seed page:

1. Determine all relevant web pages of a seed page by finding all pages which have a citation relationship (citing or be cited by, co-citation) with the seed page;
2. Determine web pages sharing common interest with a seed page by finding web pages which frequently appeared in the same browsing session or clicked by a same user with the seed page;
3. Determine web pages that "looks like" a seed page by finding pages which have word similarities to the seed page.

Now we have a committee of relevant, connected, and similar web pages that are related to a seed page. The centroid of this committee would likely represent the cluster. Since the committee consists of the most relevant web pages, this algorithm is able to deal with noisy data and the centroid of the cluster would not be affected by outliers in the data.

It is important to leverage the past session logs of users accessing the search engine to improve the retrieval engine. At a minimum, we envision the research to result in the algorithm to predict useful attributes (e.g., topics of interest for the user, candidate words for query expansion) of the session data that can be used as features during retrieval.

## 5 Experimental results

An experiment that applies the proposed clustering algorithm on a collection of clickstream data was conducted and the results were described in this section. The experiment first analyzed clickthrough precision of the clustering results with the real search results. Secondly, three domain experts manually examined and evaluated 21 samples of clusters in different cluster quality levels and 21 randomly selected search results.

### 5.1 Clickthrough precision-based analysis

In order to verify whether the proposed clustering algorithm can help users to find interesting topic or web pages, we conducted an analysis of clickthrough precision of the top 10 items in both the original search results and the new clustering results (Kumar and Patel 2008). We also analyzed the precision at a particular rank. For example, rank #1 is the top web page in the results. Note that there is a strong natural bias working in favor of the original ranking. Users tend to click things that are higher on a list *regardless* of how good the recommendation is. Table 4 summarizes the results of the clickthrough analysis. In Table 4, *rank* is the rank in a list the row represents (#1 = top recommendation). *Num Instances* is the total number of items with that rank. *Original Rank Precision* is the clickthrough precision for the original ranking. *New Rank Precision* is the clickthrough precision for my new ranking. *Unchanged Rankings* is the percentage of *clicked* items that have the same original and new ranking.

**Table 4** Clickthrough precision comparison

| Rank | Num instances | Original rank precision | New rank precision | Unchanged rankings |
|---|---|---|---|---|
| 1 | 20372364 | 38.74% | 39.65% | 47.52% |
| 2 | 19938745 | 25.38% | 30.87% | 15.72% |
| 3 | 19720132 | 22.73% | 24.54% | 12.67% |
| 4 | 19487436 | 17.12% | 20.35% | 4.33% |
| 5 | 18837769 | 14.23% | 15.78% | 2.97% |
| 6 | 18723930 | 11.35% | 12.58% | 2.26% |
| 7 | 18587432 | 9.69% | 10.84% | 1.98% |
| 8 | 18003378 | 7.55% | 9.32% | 1.56% |
| 9 | 17920533 | 6.36% | 8.27% | 1.86% |
| 10 | 17755694 | 5.75% | 7.57% | 1.74% |

As seen in Table 4, the ranking of more than 90% of the items in the first 10 ranking has been changed in the clustering results and the average clickthrough precision of the new ranking by clusters is 12.90% higher than original search result ranking. This is a significant improvement since the total number of sessions is more than 20 million.

## 5.2 Domain experts review results

### *5.2.1 Evaluation criteria and computational metrics*

There are many types of metrics for assessing the goodness of clusters. Internal similarity, external similarity, and IR-related are three very important objective metrics:

**Internal similarity**: internal similarity of the web pages within a cluster is measured by standard distance functions such as cosine similarity;
**External similarity**: external similarity of the web pages between two clusters is measured by standard distance functions such as cosine similarity;
**IR-related metrics**: Precision, Recall, F-score

Besides the objective performance metrics, we also evaluate the quality of cluster subjectively by inviting three search engine domain experts to assess at least two types of quality:

(A) *Direct Quality* of a given cluster—the internal quality of a cluster
(B) *Comparative Quality* of a cluster (or set of clusters)—when compared to a competing cluster (or set of clusters)

Subjective measures of the quality of a cluster include:

**Coherence**: the internal coherence of the web pages within a candidate cluster
**Utility**: the usefulness of a candidate cluster to a user. A cluster may be coherent but useless. For instance, if the cluster was formed because the web pages in question shared the same file format, web pages returned by this cluster are useless to users.

In this study, 21 samples of clusters in different cluster quality levels and 21 search results pages are randomly selected. The coherence and utility score of the selected clusters and search results are assessed by three search engine domain experts independently. Whether the results were generated by clusters or search engine is hided from the domain experts. The details of the experiment are specified as follows:

**Table 5** Coherence and utility scores by reviewers

| | | Coherence | Utility |
|---|---|---|---|
| Clusters | Mean | 4.46 | 4.90 |
| 21 | Std. deviation | 0.43 | 0.30 |
| High quality clusters | Mean | 4.72 | 5.00 |
| 10 | Std. deviation | 0.26 | 0.00 |
| Medium quality clusters | Mean | 4.27 | 4.80 |
| 5 | Std. deviation | 0.39 | 0.45 |
| Low quality clusters | Mean | 4.19 | 4.83 |
| 6 | Std. deviation | 0.47 | 0.41 |
| Search results | Mean | 3.93 | 4.36 |
| 21 | Std. deviation | 0.52 | 0.50 |
| Total | Mean | 4.20 | 4.63 |
| 42 | Std. deviation | 0.54 | 0.49 |

1. Generate 21 random samples of clusters using the clustering algorithm and random selected seed page and 21 search results of query which contains the seed page.
2. For each seed page, create a spreadsheet listing the seed page, a description, and all the top $N$ web pages that can be directly associated with the seed (by citation/co-session for example). Give all 40 spreadsheets to domain experts for blind-grading. Information about which part is from the clusters and which is from the search result is not known by domain experts.
3. For each spreadsheet, domain experts are asked to answer two questions:
   - Does it correspond to an interesting and important topic? If so, write a short description of the topic.
   - Does it provide a considerably complete coverage of the topic?

#### *5.2.2 Coherence and utility score of the clusters and search results*

The average coherence score (in 5 points scale) of the 21 cluster is 4.46 and the utility score is 4.90, while the average coherence score of the 21 search results is 3.93 and the utility score is 4.36. The average coherence score of the 10 clusters which have high internal similarity is 4.72 and the average utility score is 5.00. The average coherence score of the 5 clusters which have medium internal similarity is 4.27 and the average utility score is 4.80. The average coherence score of the 10 clusters which have low internal similarity is 4.19 and the average utility score is 4.83. Clearly, the coherence and utility score of clusters is better than search results according to domain expert's assessment.

## 6 Conclusion and further research

This article presented a multiple factor hierarchical clustering algorithm for large scale text collections that combines domain knowledge such as user browsing and retrieval history. This algorithm is able to efficiently assign a cluster request of web pages to its closest cluster in the presence of a large collection of text data and deal with noisy data without affecting the computation of the centroid of clusters.

An experiment was conducted to examine the clustering algorithm using both objective measures, such as internal similarity, external similarity, and clickthrough precision and subjective measures, such as the judgments of coherence and utility of retrieved web pages by human reviewers. The result indicates that the clustering algorithm can improve the search result ranking with higher clickthrough precision and shows that the coherence and utility of the proposed clusters are better than search results provided by traditional search engines.

There are some open questions that are not addressed in this paper. Answers to these questions can further improve the performance of the algorithm. First, how to identify the ranking of web pages within the clusters? Currently, the web page is ranked according to its distance to the centroid of the cluster. This ranking criteria might not be appropriate and could be further improved. Second, how to generate the description of each cluster automatically? We may use the most frequently appeared key words of the cluster as the label, however, the frequent key words don't have clear meaning in many cases and should be re-examined. Third, how to define the relationships between different clusters? Other than the external similarity measure and distance between centroids of clusters, users could leverage a hierarchical structure of the clusters and Multi-Criteria Decision Making (Kou et al. 2003, 2005; Peng et al. 2008b; Shi et al. 2005) to achieve better searching and browsing experiences.

**Acknowledgements** A short 4-page version of this paper appeared previously at the NCM 2009, the 5th International Joint Conference on INC, IMS and IDC, Aug. 25–27, 2009, Seoul, Korea. The authors would like to thank three anonymous referees for the valuable suggestions and comments. This research has been partially supported by grants from the National Natural Science Foundation of China under the Grant No. 70901015, No. 70621001; the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

## References

Al-Aomar, R., & Dweiri, F. (2008). A customer-oriented decision agent for product selection in web-based services. *International Journal of Information Technology & Decision Making*, *7*(1), 35–52.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Wokingham: Addison-Wesley.

Bush, V. (1945). As we may think. *Atlantic Monthly*, *176*, 101–108.

Cooley, R., Mobasher, B., & Srivastava, J. (1999). Data preparation for mining world wide web browsing. *Journal of Knowledge Information Systems*, *1*(1), 5–32.

CRISP-DM (1996). CRoss industry standard process for data mining. http://www.crisp-dm.org/Overview/index.htm. Accessed 28 August 2009.

Cutting, D., Karger, D., Pedersen, J., & Tukey, J. (1992). Scatter/gather: a clusterbased approach to browsing large document collection. In *Proceedings of the 15th ACM SIGIR conference* (pp. 318–329), Copenhagen, Denmark.

Dhillon, I., Fan, J., & Guan, Y. (2001). Efficient clustering of very large document collections. In R. L. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, & R. R. Namburu (Eds.) *Data mining for scientific and engineering applications*. Dordrecht: Kluwer Academic.

Foss, A., Wang, W., & Zaane, O. (2001). A non-parametric approach to Web log analysis. In *1st SIAM ICDM, workshop on web mining* (pp. 41–50), Chicago, IL.

Gómez, S. A., Chesnevar, C. I., & Simari, G. R. (2008). Defeasible reasoning in web-based forms through argumentation. *International Journal of Information Technology & Decision Making*, *7*(1), 71–101.

Han, J. W., & Kamber, M. (2006). *Data mining: concepts and techniques* (2nd ed.). San Mateo: Morgan Kaufmann.

Hearst, M. A. (1999). Untangling text data mining. In *Proceedings of ACL'99: the 37th annual meeting of the association for computational linguistics*, University of Maryland, June 20–26.

Heer, J., & Chi, E. (2001). Identification of web user traffic composition using multimodal clustering and information scent. In *1st SIAM CDM, workshop on web mining* (pp. 51–58), Chicago, IL.

Hong, A., Katerattanakul, P., & Joo, S. J. (2008). Evaluating government website accessibility: a comparative study. *International Journal of Information Technology & Decision Making*, *7*(3), 491–515.

Hu, J., & Zhong, N. (2008). Web farming with clickstream. *International Journal of Information Technology & Decision Making*, *7*(2), 291–308.

Kou, G., Liu, X., Peng, Y., Shi, Y., Wise, M., & Xu, W. (2003). Multiple criteria linear programming to data mining: models, algorithm designs and software developments. *Optimization Methods and Software*, *18*(4), 453–473, Part 2.

Kou, G., Peng, Y., Shi, Y., Wise, M., & Xu, W. (2005). Discovering credit cardholders' behavior by multiple criteria linear programming. *Annals of Operations Research*, *135*(1), 261–274.

Kumar, M., & Patel, N. (2008). Using clustering to improve sales forecasts in retail merchandising, *Annals of Operations Research*. Published online: 17 September 2008.

Lee, J., & Lee, H. (2008). Strategic agent based web system development methodology. *International Journal of Information Technology & Decision Making*, *7*(2), 309–337.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press, 2008.

Mathiak, B., & Eckstein, S. (2004). Five steps to text mining in biomedical literature. In *Proceedings of the second European workshop on data mining and text mining in bioinformatics*, Italy (pp. 47–50).

McQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley symposium on mathematics, statistics and probability* (Vol. 1, pp. 281–298).

Nasraoui, O., Cardona, C., Rojas, C., & Gonzalez, F. (2003). Mining evolving user profiles in NoisyWeb clickstream data with a scalable immune system clustering algorithm. In *Proc. of KDD workshop on web mining as a premise to...*

Pantel, P. (2003). *Clustering by committee*. Ph.D. Thesis, University of Alberta.

Park, S., Seo, K., & Jang, D. (2007). Fuzzy art-based image clustering method for content-based image retrieval. *International Journal of Information Technology and Decision Making*, *6*(2), 213–233.

Peng, Y., Kou, G., Shi, Y., & Chen, Z. (2008a). A descriptive framework for the field of data mining and knowledge discovery. *International Journal of Information Technology and Decision Making*, *7*(4), 639–682.

Peng, Y., Kou, G., Shi, Y., & Chen, Z. (2008b). A multi-criteria convex quadratic programming model for credit data analysis. *Decision Support Systems*, *44*(4), 1016–1030.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137.

Van Rijsbergen, C. J. (1979). *Information retrieval*. London: Butterworth.

Salton, G., Wong, A., & Yang, C. S. (1975). A vector space model for information retrieval. *Communications of the ACM*, *18*(11), 613–620.

Shi, Y. (2009). Current research trend: information technology and decision making in 2008. *International Journal of Information Technology and Decision Making*, *8*(1), 1–5.

Shi, Y., Peng, Y., Kou, G., & Chen, Z. (2005). Classifying credit card accounts for business intelligence and decision making: a multiple-criteria quadratic programming approach. *International Journal of Information Technology and Decision Making*, *4*(4), 1–19.

Singhal, A. (2001). Modern information retrieval: a brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, *24*(4), 35–43.

Steinbach, M., Karypis, G., & Kumar, V. (2000). A comparison of document clustering techniques. In *6th ACM SIGKDD, world text mining conference*, Boston, MA.

Zhang, Q., & Segall, R. (2008). Web mining: a survey of current research, techniques, and software. *International Journal of Information Technology and Decision Making*, *7*(4), 683–720.

Zhang, W., Yoshida, T., & Tang, X. (2009). Distribution of multi-words in Chinese and English documents. *International Journal of Information Technology and Decision Making*, *8*(2), 249–265.