Web Metasearch Result Clustering System

Adina LIPAI
Academy of Economic Studies, Bucharest, România
adina_lipai@yahoo.com

The paper presents a web search result clustering algorithm that was integrated in to a desktop application. The application aims to increase the web search engines performances by reducing the user effort in finding a web page in the list of results returned by the search engines.

Keywords: *clustering, web search, search engines.*

Introduction

The paper presents a web search results clustering system. The purpose of the application is to minimize the user effort in finding the required web page between the result list, given at a common query. The system is a desktop application that creates a series of subject orientated groups, based on the subject of the web pages. Semantically similar pages will be placed together in the same group. The task of finding the appropriate web page will be reduced to finding the subject cluster in which it was placed. Each cluster will be identified by a descriptive label.

A web document search result clustering system, has the following functions: user query retrieval, interrogation of one of several online search engines, query results retrieval, web documents processing, web documents clustering and results visualisation. In the following pages we will present the architectural components that achieve this functionality.

Main architectural components

User interface: it has the task to retrieve the user query, and other user preferences like search engines to be used in the metasearch, visualization preferences, and clustering parameters. Figure 1 represents the user interface of the application. The starting of the clustering process can be made after all search engines are loaded into the browser and the query was passed to them. The results offered by the search engines are parsed from the web browsers present in the application, using a series of text parsing operations.

The applications offers a few specific user preferences for both level of processing visualisation and for clustering parameters. Few of this are: language preference, number of search engines to be used, detailed visual output, number of clusters, and others.

Processing modules: the application has three major processing modules: document processing module, document vector space representation module, and clustering module.

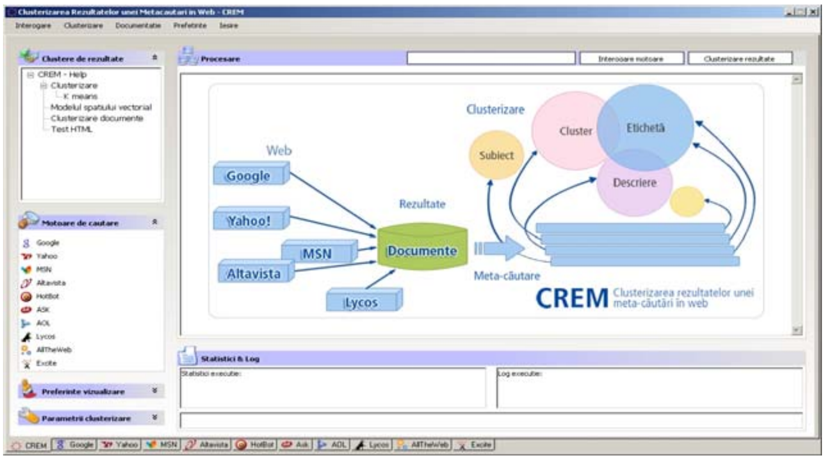


Fig.1. Web result clustering software: user interface

Document processing: the task is essential in obtaining high quality results. The main function of this module is obtaining of a condensed description of the web results, by eliminating the words and characters that do not have an informative value. De description will be used to obtain the term index vector. The term index vector will be used to obtain the *document-word* matrix, that will be used in the clustering process.

The *document-index term* matrix, contains the documents in a numeric form, obtained by applying the vector space model transformation. This representation will be used for the clustering process. There will be used two different formulas for calculating the weight of a word in the document: term frequency – inverse document frequency and term frequency – inverse liniar document frequency (Osiński, 2003]). The clustering process is made using the k-means clustering algorithm.

Application functioning principles

In the picture below we have represented the main functioning principles of the application ([Lipai, 2007]). The user has access to the application interface, which he uses to provide a query. The k-means algorithm used for clustering is adapted for document processing, and will calculate the similarity measure as distance evaluator. The application will perform 6 clusters by default, but it can form 2 – 10 clusters, according to user preferences. The N + 1 cluster will be formed from the unclassified instances.

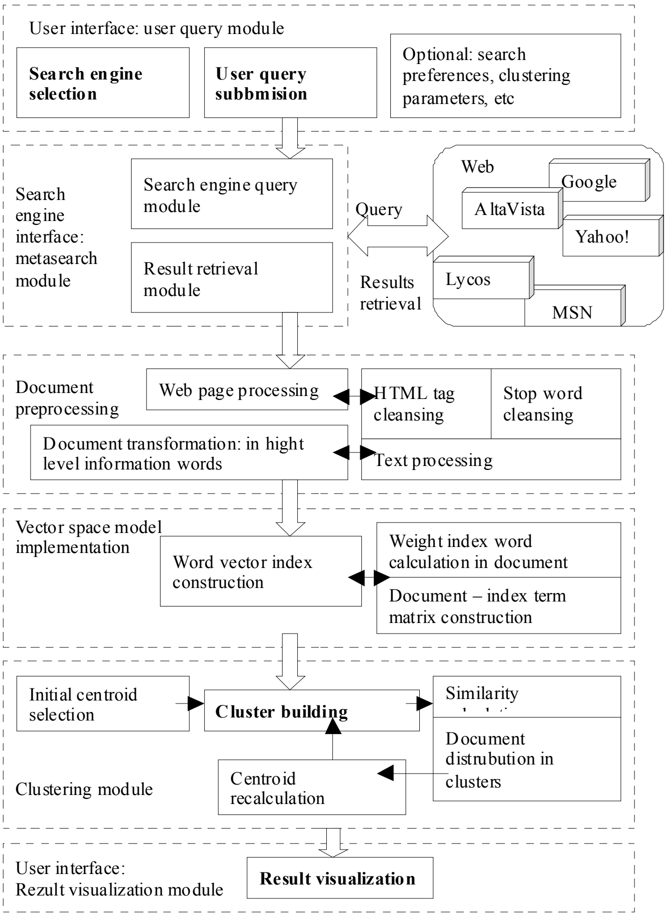

User interface: user query module
Search engine selection
User query subbmision
Optional: search preferences, clustering parameters, etc
Search engine interface: metasearch module
Search engine query module
Result retrieval module
Query
Results retrieval
Web
Google
AltaVista
Yahoo!
Lycos
MSN
Document preprocessing
Web page processing
HTML tag cleansing
Stop word cleansing
Document transformation: in hight level information words
Text processing
Vector space model implementation
Word vector index construction
Weight index word calculation in document
Document – index term matrix construction
Initial centroid selection
Cluster building
Similarity
Document distrubution in clusters
Centroid recalculation
Clustering module
User interface: Rezult visualization module
Result visualization

Fig.2. Application normal execution schema

Document pre-processing

Results are obtained by retrivieng the result list from more search engines. Document processing implies: HTML tag elimination, elimination of duplicate sites, elimination of stop words and stop characters, root extraction and other natural language processing. The root extraction module was made using basic Romanian grammatical rules.

Vector space model implementation

Implementing vector space model for a set of documents retrieved as a result of web search consist of transforming the string of words that make up a snippet in to a equivalent numeric vector. One document will be represented by n numeric elements, where n is the number of unique words present in all retrieved documents. The value of each element in the document will be given by the presence of that word in the document corpus.

First step in implementing the vector space model consist in the construction of the index term vector. The index term vector is constructed from all the individual unique words of the snippet and title of the retrieved documents. Each of the retrieved result will be divided into component words and added to the index vector ([Weiss, 2001]).

Second step consist of document vectorisation: each document will be transformed into its numeric vector representation. In the vector space model each document will be represented by a numeric vector of lengh *n,* where n is the dimension of the index term vector. The value of each word in the document corpus will be calculated with two different formulas: *term frequency-inverse document frequency* and *term frequency – linear inverse document frequency* ([Osiński, 2003], [Wróblewski, 2003]).

Term frequency-inverse document frequency:

$$w_{ij} = tf_{ij} \log_n \frac{n}{df_i} \quad \text{(Formula 1)}$$

Term frequency – linear inverse document frequency:

$$idfl_i = 1 - \frac{df_i - 1}{n-1} = \frac{n - df_i}{n-1} \quad \text{(Formula 2)}$$

Where:

- w_{ij} : represents the weight of index term c_j from document D_i;
- tf_i: represents the frequency of term c_i;
- df_i: represents the number of documents in which appears the term c_i;
- n: total number of documents.

After document vectorisation and calculation of word weight, the words that appear in only one document will be eliminated. The last step in representing the documents according with the vector space model is constructing the *document-term* matrix. The document – term matrix will be used by the clustering algorithm.

In this matrix structure we have the documents represented by lines, in the form of weighted vectors, and on columns we have the term words.

Clustering algorithm implementation

Initial centroid calculation

One of the major disadvantages of the k-means clustering algorithm is that it has a high computation time, because it makes to many iterations. This disadvantage can be unacceptable in web results clustering applications, where the computation time has to be very low. In this application we have implemented an initial centroid calculation method that has the purpose of reducing the computation time. The algorithm consists in dividing the data set in k groups, where k is the number of clusters we want to obtain. For each cluster, we will calculate average of each index term. This average vector will represent the initial centroids. The algorithms has basic principles in the partitioning clustering algorithms.

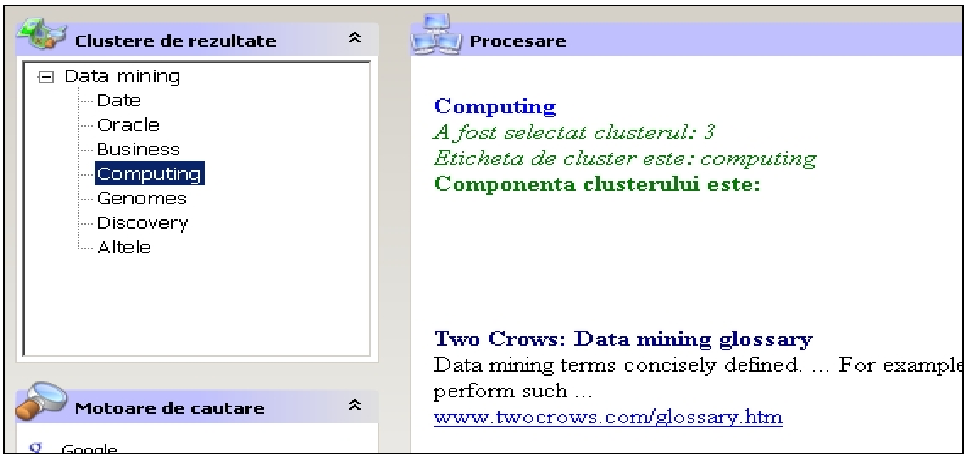
Implementation of the clustering algorithm

The clustering task consists in calculating the similarity between each document and each cluster. Each document will be distributed to the closest centroid. The most important adaptation of the k-means algorithm that had to be done for web document processing are: calculation of similarity measure as distance measure, use of soft assignation for cluster forming. In the end we will have a set of k + 1 overlapped clusters, where the last cluster consists of unclassified documents.

Cluster representation

After distributing each document to its cluster, we have to visualize the clusters. First step in forming a visual representation is label formation. The label can be made up of one or more representative words. The label will be chosen from the centroid, and it will consist of the index terms with the highest weight.

In the picture below we have represented the output returned by the application, as a result of user query “refractie”, showing in detail the label representation.



Computing
A fost selectat clusterul: 3
Eticheta de cluster este: computing
Componenta clusterului este:

Two Crows: Data mining glossary
Data mining terms concisely defined. ... For exampl
perform such ...
www.twocrows.com/glossary.htm

Fig.3. Detailed representation of cluster labels: application output

Conclusion

The paper presented a web metasearch result clustering system. And clustering application has the purpose to reduce the user effort in finding an exact web document between the hundreds of documents returned by a search engine at a common query. The paper presented the main architectural components of a clustering system, and its functionality. The steps necessary to cluster a result list were also presented, in the end showing the visualisation of the clusters.

Bibliography

1. [Osiński, 2003] **Osiński, S**.- *An Algorithm for Clustering of Web Search Results*, master thesis, Poznań University of Technology, Polonia, 2003.

2. [Weiss, 2001] **Weiss, D** .- *A clustering interface for web search results in polish and english*. Master's thesis, Poznan University of Technology, Poland, June 2001.

3. [Wróblewski, 2003] **Wróblewski**, **M., -** *A Hierarchical WWW Pages Clustering Algorithm Based on the Vector Space Model.* Master thesis, Department of Computing Science, Poznań University of Technology, Polonia, 2003.

4. [Lipai, 2007] **Lipai, A.** - *Tehnici de clusterizare a rezultatelor returnate într-o căutare web*, in procedeengs of The Eighth Ineternational Conference on Informatics In Economy, Informatics in Knowledge Society, ASE, Bucureşti, mai 2007.