Information Sciences 281 (2014) 248–264

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Clustering of web search results based on the cuckoo search algorithm and Balanced Bayesian Information Criterion

Carlos Cobos [a,b,*], Henry Muñoz-Collazos [a], Richar Urbano-Muñoz [a], Martha Mendoza [a,b], Elizabeth León [c], Enrique Herrera-Viedma [d,e]

[a] Information Technology Research Group (GTI) Members, Universidad del Cauca, Sector Tulcán Office 422 FIET, Popayán, Colombia
[b] Computer Science Department, Electronic and Telecommunications Engineering Faculty, Universidad del Cauca, Colombia
[c] Systems and Industrial Engineering Department, Engineering Faculty, Universidad Nacional de Colombia, Colombia
[d] Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
[e] Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 26 April 2013
Received in revised form 16 April 2014
Accepted 21 May 2014
Available online 2 June 2014

Keywords:
Cuckoo search algorithm
Clustering of web result
Web document clustering
Balanced Bayesian Information Criterion
k-Mean

ABSTRACT

The clustering of web search results – or web document clustering – has become a very interesting research area among academic and scientific communities involved in information retrieval. Web search result clustering systems, also called Web Clustering Engines, seek to increase the coverage of documents presented for the user to review, while reducing the time spent reviewing them. Several algorithms for clustering web results already exist, but results show room for more to be done. This paper introduces a new description-centric algorithm for the clustering of web results, called WDC-CSK, which is based on the cuckoo search meta-heuristic algorithm, *k*-means algorithm, Balanced Bayesian Information Criterion, split and merge methods on clusters, and frequent phrases approach for cluster labeling. The cuckoo search meta-heuristic provides a combined global and local search strategy in the solution space. Split and merge methods replace the original Lévy flights operation and try to improve existing solutions (nests), so they can be considered as local search methods. WDC-CSK includes an abandon operation that provides diversity and prevents the population nests from converging too quickly. Balanced Bayesian Information Criterion is used as a fitness function and allows defining the number of clusters automatically. WDC-CSK was tested with four data sets (DMOZ-50, AMBIENT, MORESQUE and ODP-239) over 447 queries. The algorithm was also compared against other established web document clustering algorithms, including Suffix Tree Clustering (STC), Lingo, and Bisecting *k*-means. The results show a considerable improvement upon the other algorithms as measured by recall, *F*-measure, fall-out, accuracy and SSL_k.

1. Introduction

In recent years, web result clustering has become a very interesting research area among academic and scientific communities involved in information retrieval (IR) and web search [13]. This is because it is most likely that results relevant to the user are close to each other in the document space, thus tending to fall into a relatively small number of clusters

* Corresponding author at: Information Technology Research Group (GTI), Universidad del Cauca, Sector Tulcán Office 422 FIET, Popayán, Colombia. Tel.: +57 2 8209800x2119; fax: +57 2 8209810.
E-mail address: ccobos@unicauca.edu.co (C. Cobos).

http://dx.doi.org/10.1016/j.ins.2014.05.047
0020-0255/

[47] and thereby achieving significant reductions in search time. In IR, these web result clustering systems are called web clustering engines and the main exponents in the field are Carrot2 (www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy (http://yippy.com, originally known as Vivisimo and later as Clusty), iBoogie (www.iboogie.com), and KeySRC (http://keysrc.fub.it) [12]. Such systems usually consist of four main components: search result acquisition, preprocessing of input, cluster construction and labeling, and visualization of resulting clusters [13] (see Fig. 1).

The **search result acquisition** component begins with a query defined by the user. Based on this query, a document search is conducted in diverse data sources – in this case in such traditional web search engines as Google, Yahoo! and Bing. In general, web clustering engines work as meta-search engines and collect between 50 and 200 results from traditional search engines. These results contain as a minimum a URL, a snippet and a title [13].

Preprocessing of search results comes next. This component converts each of the search results (as snippets) into a sequence of words, phrases, strings or general attributes or characteristics, which are then used by the clustering algorithm. A number of tasks are performed on the search results, including removal of special characters and accents, conversion of strings to lowercase, stop word removal, stemming of word, and control of terms or concepts allowed by a vocabulary [13].

Once preprocessing is finished, **cluster construction and labeling** is commenced, making use of three types of algorithm [13]: data-centric, description-aware and description-centric. Each of these builds clusters of documents and assigns a label to the groups.

Finally, in the **visualization** step, the system displays the results to the user in hierarchically organized folders. Each folder seeks to have a label or title that represents well the documents it contains and that is easily identified by the user. As such, the user simply scans the folders that are actually related to their specific needs. The presentation folder tree has been adopted by various systems such as Carrot2, Yippy, SnakeT, and KeySRC, because the folder metaphor is already familiar to computer users. Other systems such as Grokker and Kart004 use a different display scheme based on graphs [13].

To obtain good results in web document clustering the algorithms must meet the following specific requirements [13,54]: (1) automatically define the number of clusters that are going to be created; (2) generate relevant clusters for the user and assign the documents to appropriate clusters; (3) define labels or names for the clusters that are easily understood by users; (4) handle overlapping clusters (documents can belong to more than one cluster); (5) reduce the high dimension of document collections; (6) handle the processing time, i.e. less than or equal to 2 s; and (7) handle the noise that is frequently found in documents.

Another important aspect when studying or proposing an algorithm to perform web document clustering is the document representation model. The most widely used models are [38]: Vector space model [6,31], Latent Semantic Indexing (LSI) [6,57], Ontology-based model [41,67], N-gram [54], Phrase-based model [54], and Frequent Word (Term) Sets model [41,75]. In the Vector Space Model (VSM), the documents are designed as bags of words. Document collection is represented by a matrix of D-terms by n-documents. This matrix is commonly called Term by Document Matrix (TDM). In TDM, each document is represented by a vector of normalized term frequency by inverse document frequency for that term, in what is known as the TF-IDF value. In VSM, the cosine similarity is used for measuring the degree of similarity between two documents or between a document and the user query. In VSM, as in most of the representation models, a process of stop word removal and stemming [6] should be done before re-presenting the document. Stop word removal refers to the removal of very common words (such as articles and prepositions) and can yield a reduction of over 40% on TDM matrix dimensionality), while stemming refers to the reduction of words to their canonical stem or root form. A reduction of allowed terms or concepts by a vocabulary can also be executed [13].

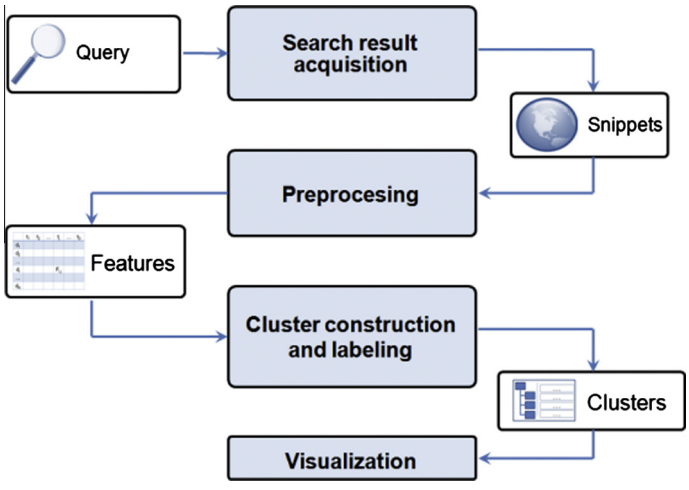

Fig. 1. The components of a web clustering engine (adapted from [13]).