# Clustering of Web Search Results based on an Iterative Fuzzy C-means Algorithm and Bayesian Information Criterion


Carlos Cobos
Computer Science Department
Universidad del Cauca
Popayán, Colombia
ccobos@unicauca.edu.co

Martha Mendoza
Computer Science Department
Universidad del Cauca
Popayán, Colombia
mmendoza@unicauca.edu.co

Elizabeth León
Systems and Industrial Department
Universidad Nacional de Colombia
Bogotá, Colombia
eleonguz@unal.edu.co

Milos Manic
Department of Computer Science
University of Idaho at Idaho Falls
Idaho Falls, U.S.A.
misko@uidaho.edu

Enrique Herrera-Viedma
Department of Computer Science
and Artificial Intelligence
University of Granada
Granada, Spain
viedma@decsai.ugr.es



***Abstract*—The clustering of web search has become a very interesting research area among academic and scientific communities involved in information retrieval. Clustering of web search result 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 web document clustering already exist, but results show there is room for more to be done. This paper introduces a new description-centric algorithm for clustering of web results called IFCWR. IFCWR initially selects a maximum estimated number of clusters using Forgy's strategy, then it iteratively merges clusters until results cannot be improved. Every merge operation implies the execution of Fuzzy C-Means for clustering results of web search and the calculus of Bayesian Information Criterion for automatically evaluating the best solution and number of clusters. IFCWR was compared against other established web document clustering algorithms, among them: Suffix Tree Clustering and Lingo. Comparison was executed on AMBIENT and MORESQUE datasets, using precision, recall, f-measure, $SSL_k$ and other metrics. Results show a considerable improvement in clustering quality and performance.**



***Keywords-web document clustering; fuzzy c-means; bayesian information criterion***


## I. INTRODUCTION

In recent years, 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 (IR) and web search [2]. Web document clustering systems seek to increase the coverage (amount) of documents presented for the user to review, while reducing the time spent reviewing them [3]. In IR, these web document clustering systems are called web clustering engines and the main exponents in the field are Carrot[2] (www.carrot2.org), SnakeT (http://snaket.di.unipi.it), Yippy (http://yippy.com, originally named as Vivisimo and then as Clusty), iBoogie (www.iboogie.com), and KeySRC (http://keysrc.fub.it) [4]. Such systems usually consist of four main components: search results acquisition, preprocessing of input, ***cluster construction and labeling***, and visualization of resulting clusters [2].

To obtain good results in web document clustering the algorithms must meet the following specific requirements [2, 5]: Automatically define the number of clusters to be created; generate relevant clusters for the user and assign the documents to appropriate clusters; define labels or names for the clusters that are easily understood by users; handle overlapping clusters (this means that documents can belong to multiple clusters); handle short input data descriptions (document snippets); reduce the high-dimension that is presented in the management of document collections; handle the processing time (the algorithm must be able to work with snippets and not only with the full text of the document); and handle the noise that is very common in the collection of documents. Several algorithms for web document clustering already exist, but results show there is still much to be done. There are three types of algorithms [2]: data-centric, description-aware and description-centric. Each of these builds ***clusters*** of documents and most of them assign a label to each group.

All of these algorithms report quality of clustering values, represented by low values of F-measure, i.e. between only 0.5 and 0.58 for AMBIENT and MORESQUE datasets, when the goal is 1.0 and their cluster labels can be improved. This is the main motivation of the present work, in which a new algorithm


This work was supported by the University of Cauca, the National University of Colombia (Bogotá), and the Spanish Ministry of Public Works and Transport.

that obtains better results for web document clustering is put forward.

The remainder of the paper is organized as follows. Section II presents some related works. The new algorithm is described in detail in Section III. Section IV shows the experimental results. Finally, some concluding remarks and suggestions for future work are presented.

## II. RELATED WORKS

As aforementioned, there are three types of web document clustering algorithms [2]: data-centric, description-aware and description-centric. A brief review of these is presented here.

**Data-centric algorithms** are the algorithms traditionally used for data clustering (partitional, hierarchical, fuzzy, density-based, etc.) [2, 6-9]. They seek the best solution in data clustering, but are not so strong on the presentation of the labels or in the explanation of the groups obtained. They address the problem of web document clustering as merely another data clustering problem. In relation to web document clustering, the hierarchical algorithm that brings the best results in accuracy is called Unweighted Pair-Group Method using Arithmetic averages (UPGMA) [7, 8]. In partitional clustering, the most representative algorithms are: k-means, k-medoids, and Expectation Maximization. Of more particular interest is the Bisecting k-means [6, 10] algorithm, which combines the strengths of the hierarchical and partitional methods reporting better results concerning the accuracy and efficiency of the UPGMA and k-means algorithms. In fuzzy clustering, a new proposal using fuzzy transduction-based clustering algorithm called FTCA was presented in 2010 [11]. FTCA results are promising but they are not compared over recognized datasets, and neither do they use $SSL_k$ metric to compare results, which is necessary to correctly compare the algorithm's results.

**Description-aware algorithms** give greater weight to one specific feature of the clustering process than to the rest. For example, they make as their priority the quality of the labeling of groups and as such achieve results that are more easily interpreted by the user. Their quality drops, however, in the cluster creation process. An example of this type of algorithm is Suffix Tree Clustering (STC) [5], which incrementally creates labels easily understood by users, based on common phrases that appear in the documents.

**Description-centric algorithms** [2, 10, 12-16] are designed specifically for web document clustering, seeking a balance between the quality of clusters and the description (labeling) of them. An example of such algorithms is Lingo [12] (implemented by www.carrot2.org in 2001), which makes use of Singular Value Decomposition (SVD) to find the best relationships between terms, but groups the documents based on the most frequent phrases in the document collection. NMF (2003) is another example of these algorithms. It is based on the non-negative matrix factorization of the term-document matrix of the given document corpus [17]. This algorithm surpasses the Latent Semantic Indexing (LSI) and the spectral clustering methods in document clustering accuracies but does not care about cluster labels. Another approach was proposed by the Pairwise Constraints guided Non-negative Matrix Factorization (PCNMF) algorithm [18] (2007). This algorithm transforms the document clustering problem from an un-supervised problem to a semi-supervised problem, using must-link and cannot-link relations between documents.

Finally, in partitional clustering from an evolutionary approach, in 2007 a hybridization between the Harmony Search (HS) [19] and k-means algorithms was proposed. This proposal is a data-centric algorithm [2] but it does not define the number of clusters automatically and does not show appropriate cluster labels. Later in 2009, a Self-Organized Genetic algorithm [20] was devised for text clustering based on the WordNet ontology. In this algorithm, a modified LSI model was also presented, which appropriately gathers the associated semantic similarities. This algorithm outperforms the standard genetic algorithm [21] and the k-means algorithm for web document clustering in similar environments. In 2010, two new algorithms were put forward. The first, called IGBHSK [22] was based on global-best harmony search, k-means and frequent term sets. The second one, called WDC-NMA [23] was based on memetic algorithms with niching techniques. In 2011, HHWDC [24] was presented. HHWDC was designed from a hyper-heuristic approach and allows defining the best algorithm for web document clustering based on several low-level heuristics and replacement strategies. These latest three researches outperform results obtained with STC and Lingo, evaluate different document representations models (term-document matrix and frequent term-document matrix) and use the Bayesian Information Criterion (BIC) for evaluating quality of solutions.

## III. THE NEW ALGORITHM

The new algorithm, called Iterative Fuzzy C-means Algorithm for Clustering of Web Results (IFCWR), is an iterative version of Fuzzy C-means guided by Bayesian Information Criterion.

IFCWR includes an initial stage related to document pre-processing, in which Lucene (http://lucene.apache.org) is used. The pre-processing stage includes: tokenize, lower case filtering, stop word removal, Porter's stemming algorithm and the building of the Term by Document Matrix (TDM with N documents by D dimensions or terms). Dimensions (columns) with a range equal to zero (0) are also removed.

Then, IFCWR starts randomly selecting a set of initial clusters based on Forgy's strategy [25]. Then it evaluates the fitness (quality) of the solution using Bayesian Information Criterion (see Eq. (1)). It goes on to merge the most similar centroids measured by cosine similarity (see Eq. (2)), to apply Fuzzy C-means on the new solution and calculate the BIC.

$$BIC = n \times Ln\left(\frac{SSE}{n}\right) + k \times Ln(n)$$

$$SSE = \sum_{j=1}^{k}\sum_{i=1}^{n} P_{i,j}\left\|x_i - c_j\right\|^2 \qquad (1)$$

Where n is the total number of documents, k is the number of clusters and SSE is the sum of squared error from the similarities of the different clusters. In SSE, $x_i$ represents the document i, $c_j$ is the centroid of the cluster j, and $P_{i,j}$ is equal to 1 if the document i belongs to cluster j, or 0 otherwise.

$$Sim(d,q) = \frac{\sum_{i=1}^{D}(W_{i,d} \times W_{i,q})}{\sqrt{\sum_{i=1}^{D} W^2_{i,d}} \times \sqrt{\sum_{i=1}^{D} W^2_{i,q}}} \quad (2)$$

Where d is a document represented in a multidimensional space of D dimensions (in this case terms or pre-processed words), q is a query also represented in a multidimensional space of D dimensions, $W_{i,d}$ is the weight of the term i in the document, and $W_{i,q}$ is the weight of the term i in the query.

If the new solution has a better fitness (lower value of BIC than the previous solution) the merge operation is repeated until the maximum processing time is reached. If the new solution is worse than the previous one and there is any time to continue processing information, the entire process is repeated and the best solution is selected (measured by BIC). A high-level overview of the IFCWR algorithm is provided in Fig.1. In the following, some additional information on these steps is provided.

| | |
|---|---|
| 01 | **Initialize algorithm parameters**: Maximum Execution Time (MET) or Maximum Number of Improvisations (MNI) |
| 02 | **Document preprocessing**: Tokenize, Lower case filtering, Stop word removal, Porter's stemming algorithm, Term-Document matrix (TDM) building, and Elimination of dimensions with a range equal to zero |
| 03 | **Initialize solution**: select randomly a set of centroids (Forgy's strategy) as the initial solution |
| 04 | **Execute Fuzzy C-means** for the initial solution |
| 05 | **Calculate fitness** (BIC) for the initial solution. |
| 06 | **Merge most similar clusters** based on cosine similarity and create a new solution (current solution) with this new configuration of centroids |
| 07 | **Execute Fuzzy C-means** for current solution |
| 08 | **Calculate fitness** (BIC) for current solution. |
| 09 | **Check stopping criterion**: if the MNI is satisfied or the MET is satisfied or BIC for the new solution is worse than the previous, iteration is terminated. Otherwise, Steps 06, 07 and 08 are repeated. |
| 10 | **Store best solution of current improvement**. From current improvement (steps 03 to 09) select the best solution, normally the last solution or the previous one and store in a list (list of best). |
| 11 | **Check stopping criterion**: if the MNI is satisfied or the MET is satisfied, iteration is terminated. Otherwise, Go to Step 03 and repeat the process for a new initial solution. |
| 12 | **Select best solution** from the list of best. |
| 13 | **Assign labels to clusters** based on frequent phrases in each cluster |
| 14 | **Overlap clusters** when labels are similar |

Figure 1. Summary of IFCWR algorithm

**Initialize algorithm parameters.** The algorithm only needs to know the Maximum Execution Time (MET) or the Maximum Number of Iterations (MNI). These parameters control the execution of the iterative process in the algorithm. For clustering of web results usually a MET value is 2 seconds.

In **Document preprocessing** a TDM representation of document is used. TDM is the most widely-used structure for document representation in IR. It is based on the vector space model [3, 6]. In this model, the documents are designed as bags of words, the document collection is represented by a matrix of D-terms by N-documents, each document is represented by a vector of normalized frequency term ($tf_i$) by the document inverse frequency for that term (its TF-IDF expressed by equation (3)), and the cosine similarity is used for measuring the degree of similarity between two documents, or between a document and the cluster centroid or between a document and the user's query.

$$w_{i,j} = \frac{freq_{i,j}}{\max(freq_i)} \times \log\left(\frac{N}{n_j}\right) \quad (3)$$

Where $freq_{i,j}$ is the observed frequency of the term j in document i, $\max(freq_i)$ is the maximum observed frequency in the document i, N is the total number of documents in collection, and $n_j$ is the number of documents where term j is presented.

When the algorithm creates the **Initialize Solution**, select an initial number of clusters based on the number of documents. This values is equal to $\lceil\sqrt{N}\rceil$, where N is the number of documents, but this value cannot be less than eight (8) neither greater than the number of documents.

**Select the best solution**: Find and select the best solution from the "list of best". The best solution is the solution with the lowest fitness value (minimize *BIC*). Then return this solution as the best clustering solution (centroids and fitness).

**Assign labels to clusters**: The algorithm uses a Frequent PHrases (FPH) approach for labeling each cluster. This step corresponds to step 2 "Frequent Phrase Extraction" in Lingo [12] (with some modifications), but in IFCWR this method is used for each cluster generated in best solution. The labeling of each cluster works as follows:

1: **Conversion of representation scheme**: Each document in the current cluster is converted from character-based to word-based representation. It takes into account the user query for this process.

2: **Document concatenation**: All documents in the current cluster are concatenated and a new document with the inverted version of the concatenated documents is created.

3: **Complete phrase discovery**: Right-complete phrases and left-complete phrases are discovered in the current cluster. The right-complete phrases and left-complete phrases are then alphabetically sorted and finally combined into a set of complete phrases.

4: **Final selection**: Terms and phrases whose frequencies exceed the ***Term Frequency Threshold*** are selected for the current cluster.

5: **Building of the "Others" label and cluster**: If documents fail to reach the Term Frequency Threshold then they are sent to the "other" cluster.

6: **Cluster label induction**: In the current cluster, a term-document matrix is built. Then, using cosine similarity, the best candidate terms or phrases for the cluster (which optimize SSE) are selected.

**Overlap clusters**: Finally, two or more clusters are merged if the labels generated in step 13 are the same. Document order is defined by the similarity to the centroid of the new merged cluster.

## IV. Experimentation

### A. Data Sets for Validation

To validate our algorithm we use two traditional benchmarking datasets in the clustering of web results: AMBIENT and MORESQUE.

AMBIENT (AMBIguous ENTries) consists of 44 queries extracted from ambiguous Wikipedia entries. Each query has a set of subtopics (meanings) and a list of one hundred (100) ranked search results collected from Yahoo! and manually annotated with document-level relevance judgments per subtopic. Most of the queries are a single word. The average number of subtopics for each AMBIENT query is 7.91, with an average number of relevant results per retrieved subtopic equal to 7.72. This dataset can be downloaded at http://credo.fub.it/ambient.

MORESQUE (MORE Sense-tagged QUEry results), consist of 114 ambiguous queries which we developed as a complement to AMBIENT. This dataset tests the behavior of Web search algorithms on queries of different lengths, ranging from 1 to 4 words. MORESQUE provides dozens of queries of length 2, 3 and 4, together with the 100 top results from Yahoo! for each query annotated. The average number of subtopics for each MORESQUE query is 3.82, with an average number of relevant results per retrieved subtopic equal to 19.43. This dataset can be downloaded at http://lcl.uniroma1.it/moresque.

### B. Compared systems

**Lingo** [12]: a web clustering engine implemented in the Carrot[2] open source framework that clusters the most frequent phrases extracted using suffix arrays; and **Lingo3G**, a commercial web clustering engine also available on Carrot[2].

**STC** [5]: the original Web search clustering approach based on suffix trees. STC and Lingo implementations are provided by the free open source Carrot[2] Document Clustering Workbench.

**KeySRC** [26]: a Web clustering engine built on top of STC with part-of-speech pruning and dynamic selection of the cut-off level of the clustering dendrogram.

**OPTIMSRC** [1]: a web document clustering algorithm based on generation of the meta partition with stochastic hill climbing followed by meta labeling (based on Lingo, STC, and KeySRC labels).

**Yahoo!**: the original search results returned by the Yahoo! search engine. In reference [1] SSL results for Yahoo! on AMBIENT dataset are presented.

### C. Ground-truth validation

Ground-truth validation is aimed at assessing how good a clustering method is at recovering known clusters (referred to as classes) from a gold standard partition. There are many different methods proposed for measuring the quality of a generated clustering compared to an ideal clustering. Three of the best known are precision, recall, f-measure, fall-out, and accuracy commonly used in information retrieval and classification tasks [19]. In this research, the weighted version of these measures is used to evaluate the quality of solution (measures commonly used by Weka [27]).

Given a collection of clusters, $\{C_1, C_2, \dots C_k\}$, to evaluate its weighted Precision, weighted Recall and weighted F-measure with respect to a collection of ideal clusters $\{C_1^i, C_2^i, \dots C_h^i\}$, these steps are followed: (a) find for each ideal cluster $C_n^i$ a distinct cluster $C_m$ that best approximates it in the collection being evaluated, and evaluate $P(C, C^i)$, $R(C, C^i)$, and $F(C, C^i)$ as defined by (4) and (5). (b) Calculate the weighted Precision (P), weighted Recall (R) and weighted F-measure (F) based on (6). Weighted Fall-out (FO) and weighted accuracy (Rand index, RI) are calculated in a similar way.

Table I shows results of each measure for each dataset and algorithm. On the AMBIENT dataset, IFCWR outperforms other algorithms (Lingo and STC) in recall, F-measure and Accuracy. Fall-out is also competitive in this dataset. Results on the MORESQUE dataset are favorable for STC. IFCWR and Lingo have similar reports in all measures except for fall-out, when IFCWR has better results (lower value than Lingo).

$$P(C, C^i) = \frac{|C \cap C^i|}{|C|} \text{ and } R(C, C^i) = \frac{|C \cap C^i|}{|C^i|} \quad (4)$$

Where C is a cluster of documents and cluster $C^i$ is an ideal cluster of documents

$$F(C, C^i) = \frac{2 * P(C, C^i) * R(C, C^i)}{P(C, C^i) + R(C, C^i)} \quad (5)$$

$$P = \frac{1}{T}\sum_{j=1}^{h} |C_j^i| * P(C_m, C_j^i), \quad R = \frac{1}{T}\sum_{j=1}^{h} |C_j^i| * R(C_m, C_j^i), \quad \text{and } F = \frac{2*P*R}{P+R} \text{ where } T = \sum_{j=1}^{h} |C_j^i| \quad (6)$$

TABLE I. Ground-truth validation results

| Dataset | Algorithm | Estimated K | P | R | F | FO | RI |
|---|---|---|---|---|---|---|---|
| AMBIENT K real: 7.91 | IFCWR | **7.36** | 80.65 | **59.48** | **63.14** | 2.58 | **83.39** |
| | Lingo | 20.86 | **86.75** | 50.21 | 58.68 | 2.65 | 80.43 |
| | STC | 11.00 | 72.40 | 53.14 | 55.38 | **2.54** | 81.89 |
| MORESQUE K real: 3.82 | IFCWR | **6.86** | 89.00 | 41.14 | 50.87 | **4.29** | 59.28 |
| | Lingo | 20.16 | **90.50** | 39.35 | 50.55 | 6.13 | 59.18 |
| | STC | 11.17 | 82.83 | **49.96** | **57.18** | 12.76 | **65.45** |

### D. User behavior evaluation

As an evaluation measure for the user behavior, the Subtopic Search Length under k document sufficiency ($SSL_k$) was used [1, 26, 28].This measure is defined as the average number of items (cluster labels or search results) that must be examined before finding a sufficient number (k) of documents relevant to any of the query's subtopics, assuming that both cluster labels and search results are read sequentially from top to bottom, and that only cluster with labels relevant to the

subtopic at hand are opened. $SSL_k$ allows an evaluation of full-subtopic retrieval (i.e., retrieval of multiple documents relevant to any subtopic) rather than focusing on subtopic coverage (i.e., retrieving at least one relevant document for some subtopics). $SSL_k$ also allows a realistic modelization of the user search behavior because the role played by cluster labels is taken into account.

The systems were tested over all queries in both datasets and the performance of the corresponding output was evaluated using $SSL_k$, with k = 1, 2, 3, 4. The results, averaged over the set of queries, are reported in Table II.

IFCWR obtained the best results in the Subtopic Search Length under k document sufficiency (SSL) measure. It outperformed all algorithms by between 3% and 34.9%.

The number of clusters (Estimated k value) is defined better in IFCWR than in Lingo and STC. AMBIENT has on average 7.91 sub-topics, IFCWR finds on average 7.36 while Lingo finds 20.86 and STC finds 11. The difference between real and estimated k value on AMBIENT is 6.95% for IFCWR, 163.72% for Lingo and 39.06% for STC. In MORESQUE a similar behavior is found. MORESQUE has on average 3.82 sub-topics, IFCWR finds on average 6.86 while Lingo finds 20.16 and STC finds 11.17. The difference between real and estimated k value on MORESQUE is 79.58% for IFCWR, 427.75% for Lingo and 192.41% for STC With a high number of clusters, the algorithm increases precision but it is more difficult for users to find the required information.

TABLE II. USER BEHAVIOR EVALUATION

| Dataset | Algorithm | $SSL_1$ | $SSL_2$ | $SSL_3$ | $SSL_4$ | Sum of $SSL_k$ |
|---|---|---|---|---|---|---|
| AMBIENT | IFCWR | **15.88** | **26.82** | **33.23** | **37.78** | **113,71** |
| | OPTIMSRC* | 20.56 | 28.93 | 34.05 | 38.94 | 122,48 |
| | Lingo* | 24.40 | 30.64 | 36.57 | 40.69 | 132,3 |
| | KeySRC* | 24.07 | 32.39 | 38.19 | 42.13 | 136,78 |
| | Lingo3G* | 24.00 | 32.37 | 39.55 | 42.97 | 138,89 |
| | Yahoo!* | 21.60 | 35.47 | 41.96 | 47.55 | 146,58 |
| MORESQUE | IFCWR | **11.61** | **19.02** | **24.59** | **28.21** | **83,43** |
| | Lingo | 16.51 | 26.44 | 33.86 | 39.20 | 116,01 |
| | STC | 19.60 | 32.26 | 40.19 | 45.19 | 137,24 |

* Take it from [1].

## V. CONCLUSIONS AND FUTURE WORK

The IFCWR algorithm has successfully been designed, implemented and evaluated. IFCWR is a description-centric algorithm for web document clustering based on Fuzzy C-means with the capacity of automatically defining the number of clusters. IFCWR uses Bayesian Information Criterion to decide which solution is better than the others. IFCWR shows promising experimental results in standard datasets and comparison with well known algorithms, but it still needs to be evaluated with users.

There are several tasks for future work. Among them: applying the IFCWR algorithm to other data sets (Text Retrieval Conference-TREC, other data sets based on Open Directory Project like ODP-239, High Accuracy Retrieval from Documents – HARD, Track of Text Retrieval Conference, among others); comparing the new algorithm with other traditional and evolutionary algorithms; making use of WordNet to work with concepts instead of terms and using disambiguation techniques in order to improve quality of cluster results.

## ACKNOWLEDGMENT

This paper was supported by the University of Cauca under Project VRI-2822 and the National University of Colombia. It has also been supported by the Projects of Spanish Ministry of Public Works and Transport 90/07 and 2009/91, and Excellence Andalusian Projects TIC-5299 and TIC-5991.

## REFERENCES

[1] C. Carpineto and G. Romano, "Optimal meta search results clustering," presented at the Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, Geneva, Switzerland, 2010.

[2] C. Carpineto, *et al.*, "A survey of Web clustering engines," *ACM Comput. Surv.*, vol. 41, pp. 1-38, 2009.

[3] R. Baeza-Yates, A. and B. Ribeiro-Neto, *Modern Information Retrieval*: Addison-Wesley Longman Publishing Co., Inc., 1999.

[4] C. Carpineto, *et al.*, "Evaluating subtopic retrieval methods: Clustering versus diversification of search results," *Information Processing & Management*, vol. 48, pp. 358-373, 2012.

[5] Z. Oren and E. Oren, "Web document clustering: a feasibility demonstration," presented at the Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, Melbourne, Australia, 1998.

[6] K. Hammouda, "Web Mining: Clustering Web Documents A Preliminary Review," ed, 2001, pp. 1-13.

[7] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*: Prentice-Hall, Inc., 1988.

[8] M. Steinbach, *et al.*, "A comparison of document clustering techniques," in *KDD workshop on text mining*, Boston, MA, USA., 2000, pp. 1-20.

[9] P. Berkhin, *et al.*, "A Survey of Clustering Data Mining Techniques," in *Grouping Multidimensional Data*, ed: Springer-Verlag, 2006, pp. 25-71.

[10] Y. Li, *et al.*, "Text document clustering based on frequent word meaning sequences," *Data & Knowledge Engineering*, vol. 64, pp. 381-404, 2008.

[11] T. Matsumoto and E. Hung, "Fuzzy clustering and relevance ranking of web search results with differentiating cluster label generation," in *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, 2010, pp. 1-8.

[12] S. Osiński and D. Weiss, "A concept-driven algorithm for clustering search results," *Intelligent Systems, IEEE*, vol. 20, pp. 48-54, 2005.

[13] D. Zhang and Y. Dong, "Semantic, Hierarchical, Online Clustering of Web Search Results," in *Advanced Web Technologies and Applications*, ed, 2004, pp. 69-78.

[14] B. Fung, *et al.*, "Hierarchical document clustering using frequent itemsets," in *Proceedings of the SIAM International Conference on Data Mining*, 2003, pp. 59-70.

[15] G. Mecca, *et al.*, "A new algorithm for clustering search results," *Data & Knowledge Engineering*, vol. 62, pp. 504-522, 2007.

[16] F. Beil, *et al.*, "Frequent term-based text clustering," in *KDD '02: International conference on Knowledge discovery and data mining (ACM SIGKDD)*, Edmonton, Alberta, Canada, 2002, pp. 436-442.

[17] X. Wei, *et al.*, "Document clustering based on non-negative matrix factorization," presented at the Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada, 2003.

[18] Z. Zhong-Yuan and J. Zhang, "Survey on the Variations and Applications of Nonnegative Matrix Factorization," in *ISORA'10: The Ninth International Symposium on Operations Research and Its Applications*, Chengdu-Jiuzhaigou, China, 2010, pp. 317–323.

[19] M. Mahdavi and H. Abolhassani, "Harmony K-means algorithm for document clustering," *Data Mining and Knowledge Discovery*, vol. 18, pp. 370-391, 2009.

[20] W. Song, *et al.*, "Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures," *Expert Systems with Applications*, vol. 36, pp. 9095-9104, 2009.
[21] W. Song and S. Park, "Genetic Algorithm-Based Text Clustering Technique," in *Advances in Natural Computation*, ed, 2006, pp. 779-782.
[22] C. Cobos, *et al.*, "Web document clustering based on Global-Best Harmony Search, K-means, Frequent Term Sets and Bayesian Information Criterion," in *2010 IEEE Congress on Evolutionary Computation (CEC)*, Barcelona, Spain, 2010, pp. 4637-4644.
[23] C. Cobos, *et al.*, "Web Document Clustering based on a New Niching Memetic Algorithm, Term-Document Matrix and Bayesian Information Criterion," in *2010 IEEE Congress on Evolutionary Computation (CEC)*, Barcelona, Spain, 2010, pp. 4629-4636.
[24] C. Cobos, *et al.*, "A hyper-heuristic approach to design and tuning heuristic methods for web document clustering," in *2011 IEEE Congress on Evolutionary Computation (CEC)*, New Orleans, USA., 2011, pp. 1350-1358.
[25] E. W. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," *Biometrics*, vol. 21, pp. 768-769, 1965.
[26] A. Bernardini, *et al.*, "Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering," in *WI-IAT '09: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies*, 2009, pp. 206-213.
[27] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*: Morgan Kaufmann Publishers Inc., 2005.
[28] U. Scaiella, *et al.*, "Topical clustering of search results," presented at the Proceedings of the fifth ACM international conference on Web search and data mining, Seattle, Washington, USA, 2012.