A Modified Fuzzy ART for Soft Document Clustering

Ravikumar Kondadadi and Robert Kozma
Division of Computer Science
Department of Mathematical Sciences
University of Memphis, Memphis, TN 38152

ABSTRACT

Document clustering is a very useful application in recent days especially with the advent of the World Wide Web. Most of the existing document clustering algorithms either produce clusters of poor quality or are highly computationally expensive. In this paper we propose a document-clustering algorithm, KMART, that uses an unsupervised Fuzzy Adaptive Resonance Theory (Fuzzy-ART) neural network. A modified version of the Fuzzy ART is used to enable a document to be in multiple clusters. The number of clusters is determined dynamically. Some experiments are reported to compare the efficiency and execution time of our algorithm with other document-clustering algorithm like Fuzzy c Means. The results show that KMART is both effective and efficient.

1. INTRODUCTION

Clustering is an important tool in data mining and knowledge discovery. The ability to automatically group similar items together enables one to discover hidden similarity and key concepts. Also clustering enables one to summarize a large amount of data into a small number of groups. This serves as an invaluable tool for users to comprehend a large amount of data. The World Wide Web search engines serve as a good example for this. Clustering is used in many different fields, like data mining [5], image compression [15] and information retrieval [16]. Reference [10] provides an extensive survey of various clustering techniques.

The World Wide Web is a large repository of many kinds of information. The sheer size of it makes it hard for any user to find information relevant to him/her. Nowadays many search engines exist to allow users to query the Web, usually via keyword search. However, since each keyword is associated with many different subjects, and the typical amount of information (web documents) returned is very large, the user is not able to have a good grasp of the output. Usually the search results are listed by some sort of relevance measure. However, even documents of vastly different subjects can share the same high relevance scores. Thus, one needs a way to cluster the results from the web search engine to facilitate users.

Some search engines have pre-defined subjects that are used to categorize the output of search engines (for instance, yahoo.com). However, few search engines (like Teoma.com, wisenut.com) provide a dynamic clustering mechanism – i.e. clustering algorithms are applied only to the resulting documents of the query. We believe that this is an important service for any search engine over the Web and is highly beneficial to users.

While there are many traditional clustering algorithms available, document clustering brings along many distinctive issues to deal with. One such issue is representation. A document is typically represented as a vector (document vector), where each dimension corresponds to a term (word), and the value denotes whether a term is present or not. In addition, similarity between documents is typically measured by some non-Euclidean measure between the vectors. This means that a document vector cannot be manipulated like normal vectors. For instance, we cannot “average” document vectors. This implies that algorithms that require a “cluster center” like K-means [9,19] need to be modified significantly.

There are multiple ways of looking at the clustering problem. According to [11], there are four different kinds of clustering algorithms: agglomerative hierarchical algorithms, partition algorithms, model fitting and density based.

Agglomerative hierarchical clustering algorithms [7] use a bottom-up methodology to merge smaller clusters into larger ones, using techniques such as minimal spanning tree. Partition algorithms such as K-means try to divide data into subgroups such that the partition optimizes certain criteria, like inter-cluster distance or intra-cluster distances. They typically take an iterative approach. Model fitting algorithms attempt to fit the data as a mixture of easily parameterized distributions (e.g. multivariate normal) and estimate their parameters. Density-based algorithms, such as DBSCAN [8], view clustering as locating high-density regions.

The goal of document clustering is to categorize the documents so that all the documents in a cluster are similar. Most of the early work [9,19] applied traditional clustering algorithms like K-means to the sets of documents to be clustered. Willett [24] provided a survey on applying hierarchical clustering algorithms into clustering documents.

Cutting et al. [6] proposed speeding up the partition-based clustering by using techniques that provide good initial clusters. Two techniques, Buckshot and Fractionation are mentioned. Buckshot selects a small sample of documents to pre-cluster them using a standard clustering algorithm and assigns the rest of the documents to the clusters formed. Fractionation splits the N documents into ‘m’ buckets where each bucket contains N/m documents. Fractionation takes an input parameter ρ, which indicates the reduction factor for each bucket. The standard clustering algorithm is applied so that if there are ‘n’ documents in each bucket, they are clustered into n/ρ clusters. Now each of these clusters are treated as if they were individual documents and the whole process is repeated until there are only ‘K’ clusters.

Most of the algorithms above use a word-based approach to find the similarity between two documents. In [26] a phrase-based approach called STC (suffix-tree clustering) was proposed. STC is a linear-time clustering algorithm. This allows STC to form clusters depending not only on individual words but also on ordering of the words.

0-7803-7278-6/02/$10.00 ©2002 IEEE

In [18], a new method was proposed for clustering related documents using association rules and hyper-graph partitioning. This method first finds set of terms that occur frequently together in documents using the Apriori algorithm [1]. These frequent item sets are then used to group items into hyper-graph edges, and a hyper-graph partitioning algorithm is used to find the item clusters. The similarity among items is captured implicitly by the frequent item sets. The main advantage of this method is that it does not require any distance measure to find the similarity between documents.

The clustering techniques above can be categorized as hard clustering, as every item is clustered into a single cluster. Soft clustering allows each item to associate with multiple clusters, by introducing a membership function W_{ij} between each cluster-item pair to measure the degree of association.

In this paper, we propose a soft document-clustering algorithm using a modified Fuzzy Adaptive resonance theory network [4]. A brief description about soft clustering and some of the soft document clustering algorithms is given in the next section. In the rest of this paper, we discuss about ART networks briefly and then we discuss our proposed algorithm, together with our experimental results. We show that our clustering technique overcomes the problems of standard hard clustering algorithms mentioned above, without paying any price in efficiency.

2. SOFT DOCUMENT CLUSTERING

A single document very often contains multiple themes. For example, this paper can be classified into the fields “fuzzy clustering” as well as “Neural networks”. Many clustering algorithms mentioned above assign each document to a single cluster, thus making it hard for a user to discover such information.

To remedy the above situation, we can employ soft clustering. That is, each document can belong to multiple clusters, and there is a measure to determine the association between each cluster and each document. This has the following advantages:

- A document can belong to multiple clusters, thus we can discover the multiple themes for a document.
- Clusters that contain combination of themes. For instance, in our experiments, when the document set has documents related to baseball, movies and baseball-movies respectively, KMART formed three clusters for documents about baseball, movies and baseball movies where as hard clustering algorithms like k-means failed to produce a cluster for baseball-movies.
- The measure associated between clusters and documents can be used as a relevance measure to order the document appropriately.

Many soft clustering algorithms employ the idea of fuzziness in their methods. One of the most common fuzzy clustering algorithms is Fuzzy C-means (FCM). It was first reported by Dunn in 1972 and subsequently generalized by Bezdek [3]. FCM is based on the Partition clustering algorithm, iterating over the data sets until the values of the membership function stabilizes. FCM has been used in many applications like medical diagnosis, image analysis, irrigation design and automatic target recognition. Other fuzzy algorithm techniques such as Self-Organizing Maps [14], also abounds. Baraldi and Blonda [2] provides a good survey of such algorithms.

However, one drawback of fuzzy algorithms is that they are slow compared to non-fuzzy algorithms. Fuzzy clustering algorithms tend to be iterative, and typical fuzzy clustering algorithms require repeatedly calculating the associations between every cluster/document pair.

SISC and WBSC [12,13] are two soft document-clustering algorithms developed by one of the authors of this paper. SISC uses a modified Fuzzy C Means algorithm to cluster documents. It uses a randomization approach that enables it to avoid lot of computations needed in a traditional fuzzy clustering algorithm. At each iteration, it computes a similarity measure between a cluster and a document with a probability proportional to the proximity of the similarity measure to the threshold measure. It also has a robust outlier-handling mechanism.

WBSC [13] uses a word-based approach. It starts with each term as a cluster and clusters the terms depending on the documents they appear in. It is a hierarchical clustering algorithm.

There has also been work done on applying Self-organizing maps to cluster documents. For instance, [20] discusses an approach called “Adaptive approach” which uses self-organizing maps to cluster documents and also takes feedback from the user and re-clusters the documents. Approaches based on neural networks include one based on an adaptive bilinear retrieval model [25], and a hierarchical model based on fuzzy adaptive resonance theory [17].

In this paper, we propose a modification to the traditional Fuzzy ART algorithm, which is a hard clustering algorithm, to make it a soft clustering algorithm. This also cuts down some iterative search process in Fuzzy ART making it much faster than some of the existing document-clustering algorithms. We discuss briefly about ART networks in the next section.

3. ART NETWORKS

ART (Adaptive Resonance theory) neural networks are developed by Grossberg [4] to address the problem of stability-plasticity dilemma.

A network is plastic, if it can adapt to the inputs indefinitely. A network is not stable if it can with stand to noise. A traditional neural network uses the training data to adapt to the input, but does not do it for test data. So it is not plastic. Also if the training data contains some erroneous information it adapts according to that erroneous data. So it is not stable. The stability-plasticity dilemma can be proposed as follows: How can a learning system be designed to remain plastic or adaptive and at the same time remain stable to irrelevant events?

The ART networks proposed by Grossberg solve this problem. It is an incremental algorithm. So it adapts to new

0-7803-7278-6/02/$10.00 ©2002 IEEE

inputs indefinitely. At the same time, it wont let new inputs to change any stored patterns until the input pattern matches the stored pattern with in a certain tolerance. This means that an ART network has both plasticity and stability; new categories can be formed when the environment does not match any of the stored patterns, but the environment cannot change stored patterns unless they are sufficiently similar.

The general structure of an ART network is shown in the figure 1.

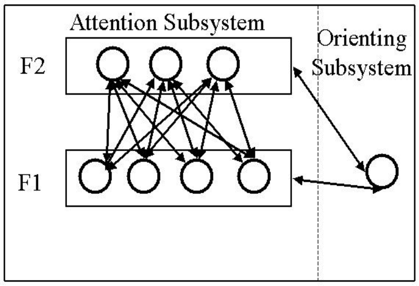

Figure 1: Architecture of an ART network

A typical ART network consists of two layers: an input layer (F1) and an output layer (F2). There are no hidden layers. The input layer contains N nodes, where N is the number of input patterns. The number of nodes in the output layer is decided dynamically. Every node in the output layer has a corresponding prototype vector. The networks dynamics are governed by two sub-systems: an attention subsystem and an orienting subsystem.

The attention subsystem proposes a winning neuron (or category) and the orienting subsystem decides whether to accept it or not. The network is said to be in a resonant state when the orienting system accepts a winning category (i.e. when the winning prototype vector matches the current input pattern close enough.)

There are many versions of ART algorithms: ART1, ART2, ARTMAP, Fuzzy ART, Fuzzy ART MAP etc. ART1 is the basic ART network that is used for binary data. Fuzzy ART is an extension of ART1 for analog data. It uses Fuzzy AND operator instead of the crisp operator. The basic Fuzzy ART algorithm was described below:

The Fuzzy ART takes three input parameters: choice parameter ($\beta > 0$), vigilance parameter ($0 \le \rho \le 1$) and learning rate ($0 \le \lambda \le 1$).

Step1: Initialization:

- Initialize all the parameters.

Step 2: Apply input pattern

- Let I:=[next input vector]
- Let P:= be the set of candidate prototype vectors

Step 3: Category choice

- Find the closest prototype vector ($P_i \in P$) that maximizes

$$\frac{\| \vec{I} \wedge \vec{P}_i \|}{\beta + \| \vec{P}_i \|} \quad (1)$$

β acts as a tie breaker when multiple prototype vectors are subsets of the input pattern and favors larger magnitude prototypes.

Step 4: Vigilance Test

- The prototype selected in the previous step undergoes a vigilance test that compares the similarity between the winning prototype and the current input pattern against a user-defined vigilance parameter as follows

$$\frac{\| \vec{I} \wedge \vec{P}_i \|}{\| I \|} \succ \rho \quad (2)$$

If the prototype passes the vigilance test, it is adapted to the given input pattern (Step 5). Otherwise, the current prototype is deactivated for the current input pattern and other prototypes in the F2 layer are also undergone the vigilance test until one of the prototypes passes the test.

If none of them passes the test, a new prototype is created for the current input pattern. Go to step 2 to continue for the next input.

Step 5: Matched prototype update:

- The matched prototype is updated to move closer to the current input pattern according to the following equation

$$P_i = \lambda(\vec{I} \wedge \vec{P}_i) + (1-\lambda)\vec{P} \quad (3)$$

λ is the learning rate. If λ is 1, it is called fast learning.

After the update, all the prototypes are reactivated and the algorithm continues with the next input (step 2).

The Fuzzy ART algorithm mentioned above is a hard clustering algorithm. We modified the Fuzzy Art to make it a soft clustering algorithm. The algorithm is called KMART (Kondadadi & Kozma Modified ART) algorithm. In the next section we present KMART.

4. KMART

Although Fuzzy ART has the name “fuzzy” in it, it is used to work with Fuzzy data. But it categorizes a given set of data items into different partitions. (i.e. it is a hard clustering algorithm). So it cannot be used for document clustering effectively.

The algorithm can be broadly divided into three stages; Pre-processing, cluster building and keyword selection.

4.1 Pre-processing:

In this stage, stop words are removed from all the documents. The algorithm maintains a common list of stop words like articles, propositions, verb auxiliaries etc. Then all the words in all documents are combined and redundant terms are removed to form a list of unique words in all the documents together. Document vectors are formed for each document. The length of the vector is the total number of unique words in all documents and the value of the vector is the frequency of the word if the word appears in the documents and zero otherwise.

4.2 Cluster Building:

A modified version of Fuzzy ART was used for cluster building.

We propose a change to the existing Fuzzy ART algorithm to make it a soft clustering algorithm. Instead of choosing a maximum similarity category and applying the vigilance test to check if it is close enough to the input pattern, we can check every category in the F2 layer and apply the vigilance test and if the category passes the vigilance test, the input document is put into that particular category. The similarity measure computed in the vigilance test defines a degree of membership of the given input pattern to the current cluster. This enables the document to be in multiple clusters with varying degrees of memberships. All the prototypes that pass the vigilance test are updated according to (3). This modification also has other advantages apart from allowing soft clustering.

- Fuzzy ART is generally time consuming because it involves some iterative search while searching for a winning category that satisfies the vigilance test. In our modification, there is no search because every F2 node is checked. This makes it computationally less expensive.
- Another advantage is that by eliminating the category choice step, we are avoiding the use of choice parameter, there by reducing the number of user-defined parameters in the system.

This modification also does not violate the underlying principle of ART networks i.e. to avoid stability- plasticity dilemma. KMART still is an incremental clustering algorithm, thus plastic and also before learning a new input it checks the input and the input pattern is learned only if it matches any of the stored patterns with in a certain tolerance.

4.3 Keyword selection:

The final step in KMART is to display representative keywords for each cluster formed in the previous stage. This allows users to distinguish among different clusters. For each cluster, we rank the words in that cluster according to the number of documents in the cluster the word appears and the similarity of the documents (defined by vigilance test) in which the word appears. We generally display the first 7-10 words as keywords.

5. EXPERIMENTS

In this section, we describe the results of the various experiments conducted and analyze the results. We compared our experiments with both soft clustering algorithms like SISC [12] and also hard clustering algorithms like k-means [19] and Fractionation [6].

5.1 Data & Experimental Environment:

We downloaded 2000 documents from the World Wide Web manually that belong to different categories like food, agents, virus, cricket, football, genetic algorithms etc. we also downloaded another 2000 documents from the UCI KDD archive [22] which has various documents from different newsgroups.

All the experiments are carried out on a 733 MHz, 256 MB RAM PC. We ran the algorithm to get the clusters and compared the quality of clusters formed. We also compared the execution times of all the algorithms for document sets of different sizes. To be more accurate, we actually ran all the algorithms on different document sets. Since except ours all other clustering algorithms take number of clusters as input, we made all of them to produce same number of clusters. All the results shown are averages taken over 20 different runs.

5.2 Quality of the Clusters:

We compared the clusters formed by the documents against the documents in the original categories and matched the clusters with the categories one-to-one. The number of matches can be used to measure the quality of the clusters formed. The matching was computed using a bi-partite matching algorithm [21]. Figure 2 compares the quality of the clusters formed by KMART to Fuzzy ART, SISC, K-means and Fractionation.

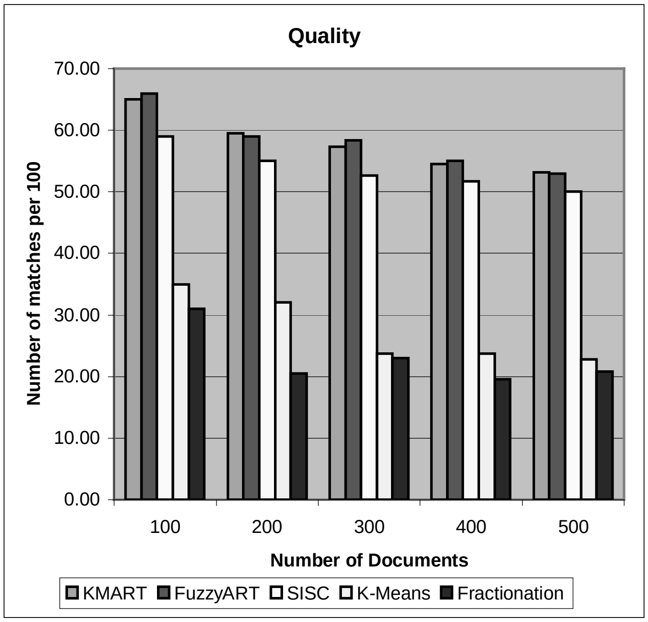

Figure 2: Comparison of quality of the clusters

As we can clearly see from the figure, KMART formed clusters of better quality compared to all other algorithms and almost comparable to the traditional Fuzzy ART.

5.3 Execution time:

We also compared the execution times of our approach with Fuzzy ART, SISC, K-Means and Fractionation. Figure 3 compares the execution time of KMART with other algorithms.

The execution time of KMART is linear with the number of documents. It can be clearly seen from the figure that our algorithm runs much faster than all the hard clustering algorithms and its execution time is almost comparable to that of SISC. KMART also runs much faster than Fuzzy ART. This is because KMART avoids the expensive time consuming search in the category choice step by eliminating that step from the Fuzzy ART algorithm.

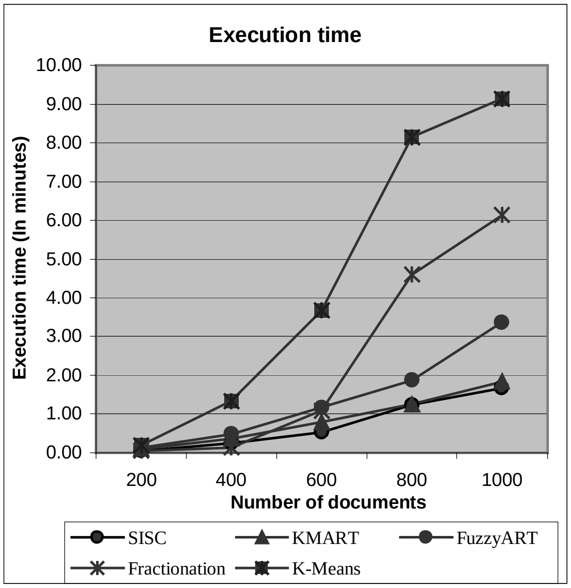

Figure 3: Comparison of execution times

This shows that KMART is very effective and efficient both in terms of quality of the clusters and also the execution time.

6. CONCLUSIONS AND FUTURE WORK

We proposed a modification to the traditional Fuzzy ART to adapt it to the document-clustering domain that makes it a soft clustering algorithm and also reduces the execution time. The experimental results show that our approach forms clusters of better quality and also faster compared to other algorithms. The main advantage of KMART over most of other fuzzy clustering algorithms is that the number of clusters is decided dynamically. Currently it's practical to work with around 1500 documents from web search perspective. Our future work involves making it more efficient and reducing the response time by adapting better data structures. We are also considering ways of automatically tuning the values of the vigilance and learning rate parameters depending on the input document set deriving a parameter-free Fuzzy ART network.

References:

[1] Rakesh Agrawal and Ramakrishnan Srikant, Fast Algorithms for Mining Association Rules in Large Databases, In Proceedings of the 1994 International Conference on Very Large Databases, pp 487-499, 1994.

[2] A. Baraldi, P. Blonda, A survey of fuzzy clustering algorithms for pattern recognition, Technical Report TR-98-038, International Computer Science Institute, Berkeley, CA, Oct 1998.

[3] J.L. Bezdek, Pattern Recognition With Fuzzy Objective Function Algorithms, Plenum Press, Nyew York, NY. 1981.

[4] Carpenter,G.A., Grossberg,S., Rosen,D. "Fuzzy ART: Fast Stable Learning of Analog Patterns by an Adaptive Resonance System.", Neural Networks, 4, 759-771.

[5] M.S. Chen, J. Han, and P.S. Yu, Data Mining: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, 8(6): 866-883, 1996.

[6] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, John W. Tukey, Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, In Proceedings of the Fifteenth Annual International ACM SIGIR Conference, pp 318-329, June 1992.

[7] F. Murtagh. A survey of recent advances in hierarchical clustering algorithms. The Computer Journal, 26(4): 354--359, 1983.

[8]Martin Ester, Hans-Peter Kriegel, Jorg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining ({KDD}-96)}, pages 226--231. AAAI Press, 1996.

[9] D. R. Hill, A vector clustering technique, in: Samuelson (Ed.), Mechanized Information Storage, Retrieval and Dissemination, North-Holland, Amsterdam, 1968.

[10] A.K. Jain, M.N. Murty and P.J. Flynn, Data Clustering: A Review, ACM Computing Surveys. 31(3): 264-323, Sept 1999.

[11] W.J. Krzanowski and F.H. Marriott, Multivariate Analysis: Classification, Covariance Structures and Repeated Measurements. Arnold, London, 1998.

[12] King-Ip Lin, Ravikumar Kondadadi, A Similarity based Soft clustering algorithm for documents, In proceedings of 7th international conference on Database systems for advanced applications (DASFAA-2001), pp 40-47, April 2001.

[13] King-Ip Lin, Ravikumar Kondadadi, "A Word based soft clustering algorithm for documents", In proceedings of 16th International conference on computers and their applications (CATA-2001), pp 391-395, March 2001.

[14] T. Kohonen, The self-organizing map, Proceedings of the IEEE, 78(9): 1464-1480, 1990.

[15] Y. Linde, A. Buzo and R.M. Gray, An Algorithm for Vector Quantization Design, IEEE Transactions on Communications, 28(1), 1980.

[16] M.N. Murty and A. K. Jain, Knowledge-based clustering scheme for collection management and retrieval of library books, Pattern recognition 28, 946-964, 1995.

[17] Alberto Munoz, Compound key word generation from document databases using a Hierarchical clustering ART Model, Intelligent Data Analysis, 1(1), Jan 1997. http://www-east.elsevier.com/ida/browse/96-5/ida96-5.htm

[18] Jerome Moore, Eui-Hong (Sam) Han, Daniel Boley, Maria Gini, Robert Gross, Kyle Hastings, George Karypis, Vipin Kumar, and Bamshad Mobasher, Web Page Categorization and Feature Selection Using Association Rule and Principal Component Clustering, In Proceedings of seventh Workshop on Information Technologies and Systems (WITS'97), December 1997.

[19] J. J. Rocchio, Document retrieval systems – optimization and evaluation, Ph.D. Thesis, Harvard University, 1966.

[20] Dmitri Roussinov, Kristine Tolle, Marshall Ramsey and Hsinchun Chen, "Interactive Internet search through Automatic clustering: an empirical study", In Proceedings of the International ACM SIGIR Conference, pages 289-290, 1999.

[21] Robert E. Tarjan, Data Structures and Network Algorithms, Society for Industrial and Applied Mathematics, 1983.

[22]UCI,http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html

[23] P.Willett, V. Winterman and D. Bawden, "Implementation of Nearest Neighbour Searching in an Online Chemical Structure Search System", Journal of Chemical Information and Computer Sciences, 26, 36-41,1986.

[24] P.Willett, Recent trends in hierarchical document clustering: a critical review, Information processing and management, 24: 577-97, 1988.

[25] Wong, S.K.M., Cai, Y.J., and Yao, Y.Y, Computation of Term Association by neural Network. In Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 107-115, 1993.

[26] O.Zamir, O.Etzioni, Web document clustering: a feasibility demonstration, in Proceedings of 19th international ACM SIGIR conference on research and development in information retrieval (SIGIR 98), 1998, pp 46-54.