A Review of Web Document Clustering Approaches

N. Oikonomakou, M. Vazirgiannis,
Dept. of Informatics,
Athens University of Economics & Business
Patision 76, 10434, Greece
{oikonomn, mvazirg}@aueb.gr

Nowadays, the Internet has become the largest data repository, facing the problem of information overload. Though, the web search environment is not ideal. The existence of an abundance of information, in combination with the dynamic and heterogeneous nature of the Web, makes information retrieval a difficult process for the average user. It is a valid requirement then the development of techniques that can help the users effectively organize and browse the available information, with the ultimate goal of satisfying their information need. Cluster analysis, which deals with the organization of a collection of objects into cohesive groups, can play a very important role towards the achievement of this objective. In this paper, we present an exhaustive survey of web document clustering approaches available on the literature, classified into three main categories: text-based, link-based and hybrid. Furthermore, we present a thorough comparison of the algorithms based on the various facets of their features and functionality. Finally, based on the review of the different approaches we conclude that although clustering has been a topic for the scientific community for three decades, there are still many open issues that call for more research.

1. Introduction

Nowadays, the internet has become the largest data repository, facing the problem of information overload. In the same time, more and more people use the World Wide Web as their main source of information. The existence of an abundance of information, in combination with the dynamic and heterogeneous nature of the Web, makes information retrieval a tedious process for the average user. Search engines, meta-search engines and Web Directories have been developed in order to help the users quickly and easily satisfy their information need.

Usually, a user searching for information submits a query composed by a few keywords to a search engine (such as Google (http://www.google.com) or Lycos

(http://www.lycos.com)). The search engine performs exact matching between the query terms and the keywords that characterize each web page and presents the results to the user. These results are long lists of URLs, which are very hard to search. Furthermore, users without domain expertise are not familiar with the appropriate terminology thus not submitting the right (in terms of relevance or specialization) query terms, leading to the retrieval of more irrelevant pages.

This has led to the need for the development of new techniques to assist users effectively navigate, trace and organize the available web documents, with the ultimate goal of finding those best matching their needs. One of the techniques that can play an important role towards the achievement of this objective is *document clustering*. The increasing importance of document clustering and the variety of its applications has led to the development of a wide range of algorithms with different quality – complexity tradeoffs.

The contribution of this paper is a review and a comparison of the existing web document clustering approaches. A comparative description of the different approaches is important in order to understand the needs that led to the development of each approach (i.e. the problems that it intended to solve) and the various issues related to web document clustering. Finally, we determine problems and open issues that call for more research in this context.

2. Motivation for document clustering

Clustering (or cluster analysis) is one of the main data analysis techniques and deals with the organization of a set of objects in a multidimensional space into cohesive groups, called clusters. Each cluster contains objects that are very similar to each other and very dissimilar to objects in other clusters (Rasmussen, 1992). An example of a clustering is depicted in figure 1. The input objects are shown in figure 1a and the existing clusters are shown in 1b. Objects belonging to the same cluster are depicted with the same symbol. Cluster analysis aims at discovering objects that have some representative behaviour in the collection. The basic idea is that if a rule is valid for one object, it is very possible that the rule also applies to all the objects that are very similar to it. With this technique one can trace dense and sparse regions in the data space and, thus, discover hidden similarities, relationships and concepts and to group large datasets with regard to the common characteristics of their objects. Clustering is a form of *unsupervised classification*, which means that the categories into which the collection must be partitioned are not known, and so the clustering process involves the discovering of these categories.

In order to cluster documents, one must first choose the type of the characteristics or attributes (e.g. words, phrases or links) of the documents on which the clustering algorithm will be based and their representation. The most commonly used model is the Vector Space Model (Salton et al., 1975). Each document is represented as a feature vector whose length is equal to the number

of unique document attributes in the collection. Each component of that vector has a weight associated to it, which indicates the degree of importance of the particular attribute for the characterization of the document. The weight can be either 0 or 1, depending on if the attribute characterizes or not the document respectively (binary representation). It can also be a function of the frequency of occurrence of the attribute in the document (tf) and the frequency of occurrence of the attribute in the entire collection (tf-idf). Then, an appropriate similarity measure must be chosen for the calculation of the similarity between two documents (or clusters). Some widely used similarity measures are the Cosine Coefficient, which gives the cosine of the angle between the two feature vectors, the Jaccard Coefficient and the Dice Coefficient (all normalized versions of the simple matching coefficient). More on the similarity measures can be found in Van Rijsbergen (1979), Willet (1988) and Strehl et al. (2000).

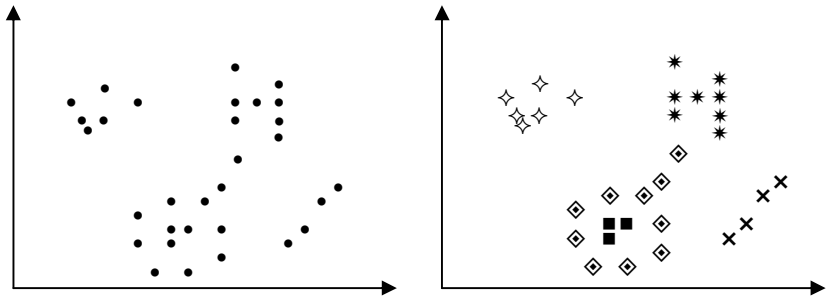

Figure 1. Clustering Example: a) input and b) clusters

Many uses of clustering as part of the Web Information Retrieval process have been proposed in the literature. Firstly, based on the cluster hypothesis, clustering can increase the efficiency and the effectiveness of the retrieval (Van Rijsbergen, 1979). The fact that the user's query is not matched against each document separately, but *against each cluster* can lead to an increase in the effectiveness, as well as the efficiency, by returning more relevant and less non relevant documents. Furthermore, clustering can be used as a very powerful mechanism for browsing a collection of documents or for presenting the results of the retrieval (e.g. suffix tree clustering (Zamir and Etzioni,1998), Scatter/Gather (Cutting et al., 1992)). A typical retrieval on the Internet will return a long list of web pages. The organization and presentation of the pages in small and meaningful groups (usually followed by short descriptions or summaries of the contents of each group) gives the user the possibility to focus exactly on the subject of his interest and find the desired documents more quickly. Furthermore, the presentation of the search results in clusters can provide an overview of the major subject areas related to the user's topic of interest. Finally, other applications of clustering include query refinement (automatic inclusion or exclusion of terms from the user's query in order to increase the effectiveness of

the retrieval), tracing of similar documents and the ranking of the retrieval results (Kleinberg, 1997 & Page et al.,1998).

3. Web Document Clustering Approaches

There are many document clustering approaches proposed in the litterature. They differ in many parts, such as the types of attributes they use to characterize the documents, the similarity measure used, the representation of the clusters etc. Based on the characteristics or attributes of the documents that are used by the clustering algorithm, the different approaches can be categorized into *i. text-based*, in which the clustering is based on the content of the document, *ii. link-based*, based on the link structure of the pages in the collection and *iii. hybrid* ones, which take into account both the content and the links of the document.

Most algorithms in the first category were developed for use in static collections of documents that were stored and could be retrieved from a database and not for collections of web pages, although they are used for the later case too. But, contrary to traditional document retrieval systems, the World Wide Web is a *directed graph.* This means that apart from its content, a web page contains other characteristics that can be very useful to clustering. The most important among these are the hyperlinks that play the role of citations between the web pages. The basic idea is that when two documents are cited together by many other documents (i.e. have many common incoming links) or cite the same documents (i.e. have many common outgoing links) there exists a semantic relationship between them. Consequently, traditional algorithms, developed for text retrieval, need to be refitted to incorporate these new sources of information about documents associations. In the Web Information Retrieval literature there are many applications based on the use of hyperlinks in the clustering process and the calculation of the similarity based on the link structure of the documents has proven to produce high quality clusters.

In the following we consider n to be the number of documents in the document collection under consideration.

3.1. Text-based Clustering

The text-based web document clustering approaches characterise each document according to its content, i.e. the words (or sometimes phrases) contained in it. The basic idea is that if two documents contain many common words then it is likely that the two documents are very similar.

The text-based approaches can be further classified according to the clustering method used into the following categories: *partitional*, *hierarchical*, *graph-based*, *neural network-based* and *probabilistic*. Furthermore, according to the way a clustering algorithm handles uncertainty in terms of cluster overlapping, an algorithm can be either *crisp* (or hard), which considers non-overlapping

partitions, or *fuzzy* (or soft), with which a document can be classified to more than one cluster. Most of the existing algorithms are crisp, meaning that a document either belongs to a cluster or not. It must also be noted that most of the mentioned approaches in this category are general clustering algorithms that can be applied to any kind of data. In this paper, though, we are interested in their application to documents. In the following paragraphs we present the main text-based document clustering approaches, their characteristics and the representative algorithms of each category. We also present a rather new approach to document clustering, which relies on the use of ontologies in order to calculate the similarity between the words that characterize the documents.

3.1.1 Partitional Clustering

The partitional or non-hierarchical document clustering approaches attempt a flat partitioning of a collection of documents into a *predefined* number of *disjoint* clusters. Partitional clustering algorithms are divided into iterative or reallocation methods and single pass methods. Most of them are iterative and the single pass methods are usually used in the beginning of a reallocation method, in order to produce the first partitioning of the data.

The partitional clustering algorithms use a feature vector matrix[1] and produce the clusters by optimising a criterion function. Such criterion functions are the following: maximize the sum of the average pairwise cosine similarities between the documents assigned to a cluster, minimize the cosine similarity of each cluster centroid to the centroid of the entire collection etc. Zhao and Karypis (2001) compared eight criterion functions and concluded that the selection of a criterion function can affect the clustering solution and that the overall quality depends on the degree to which they can correctly operate when the dataset contains clusters of different densities and the degree to which they can produce balanced clusters.

The most common partitional clustering algorithm is k-means, which relies on the idea that the center of the cluster, called *centroid*, can be a good representation of the cluster. The algorithm starts by selecting k cluster centroids. Then the cosine distance[2] between each document in the collection and the centroids is calculated and the document is assigned to the cluster with the nearest centroid. After all documents have been assigned to clusters, the new cluster centroids are recalculated and the procedure runs iteratively until some criterion is met. Many variations of the k-means algorithm are proposed, e.g. ISODATA (Jain et al., 1999) and bisecting k-means (Steinbach et al., 2000). Another approach to partitional clustering is used in the Scatter/Gather system.

[1] Each row of the feature vector matrix corresponds to a document and each column to a term. The ij-th entry has a value equal to the weight of the term j in document i.

[2] K-means does not generally use the cosine similarity measure, but when applying k-means to documents it seems to be more appropriate.

Scatter/Gather uses two linear-time partitional algorithms, Buckshot and Fractionation, which also apply HAC logic[3]. The idea is to use these algorithms to find the initial cluster centers and then find the clusters using the assign-to-nearest approach. Finally, the single pass method (Rasmussen, 1992) is another approach to partitional clustering which is based on the assignment of each document to the cluster with the most similar representative is above a threshold. The clusters are formed after only one pass of the data and no iteration takes place. Consequently, the order in which the documents are processed influences the clustering.

The advantages of these algorithms consist in their simplicity and their low computational complexity. The disadvantage is that the clustering is rather arbitrary since it depends on many parameters, like the values of the target number of clusters, the selection of the initial cluster centroids and the order of processing the documents.

3.1.2 Hierarchical Clustering

Hierarchical clustering algorithms produce a sequence of nested partitions. Usually the similarity between each pair of documents is stored in a nxn similarity matrix. At each stage, the algorithm either merges two clusters (agglomerative methods) or splits a cluster in two (divisive methods). The result of the clustering can be displayed in a tree-like structure, called a *dendrogram*, with one cluster at the top containing all the documents of the collection and many clusters at the bottom with one document each. By choosing the appropriate level of the dendrogram we get a partitioning into as many clusters as we wish. The dendrogram is a useful representation when considering retrieval from a clustered set of documents, since it indicates the paths that the retrieval precess may follow (Rasmussen, 1992).

Almost all the hierarchical algorithms used for document clustering are agglomerative (HAC). The steps of the typical HAC algorithm are the following:

1. Assign each document to a single cluster
2. Compute the similarity between all pairs of clusters and store the result in a similarity matrix, in which the ij-th entry stores the similarity between the i-th and j-th cluster
3. Merge the two most similar (closest) clusters
4. Update the similarity matrix with the similarity between the new cluster and the original clusters

[3] Buckshot and Fractionation both use a cluster subroutine that applies the group average hierarchical clustering method.

5. Repeat steps 3 and 4 until only one cluster remains or until a threshold[4] is reached

The hierarchical agglomerative clustering methods differ in the way they calculate the similarity between two clusters. The existing methods are the following (Rasmussen, 1992; El. Handouchi and Willet, 1989; Willet, 1988):

- *Single link*: The similarity between a pair of clusters is calculated as the similarity between the two most similar documents, one of which is in each cluster. This method tends to produce long, loosely bound clusters with little internal cohesion (chaining effect). The single link method incorporates useful mathematical properties and can have small computational complexity. There are many algorithms based on this method. Their complexities vary from O(nlogn) to $O(n^5)$. Single link algorithms include van Rijsbergen's algorithm (Van Rijsbergen, 1979), SLINK (Sibson, 1973), Minimal Spanning Tree (Rasmussen, 1992) and Voorhees's algorithm (Voorhees, 1986).
- *Complete link*: The similarity between a pair of clusters is taken to be the similarity between the least similar documents, one of which is in each cluster. This definition is much stricter than that of the single link method and, thus, the clusters are small and tightly bound. Implementations of this method are the CLINK algorithm (Defays, 1977), which is a variation of the SLINK algorithm, and the algorithm proposed by Voorhees (Voorhees, 1986).
- *Group average*: This method produces clusters such that each document in a cluster has greater average similarity with the other documents in the cluster than with the documents in any other cluster. All the documents in the cluster contribute in the calculation of the pairwise similarity and, thus, this method is a mid-point between the above two methods. Usually the complexity of the group average algorithm is higher than $O(n^2)$. Voorhees proposed an algorithm for the group average method that calculates the pairwise similarity as the inner product of two vectors with appropriate weights (Voorhees, 1986). Steinbach et al. (2000) used UPGMA for the implementation of the group average method and obtained very good results.
- *Ward's method*: In this method the cluster pair to be merged is the one whose merger minimizes the increase in the total within-group error sum of squares based on the distance between the cluster centroids (i.e. the sum of the distances from each document to the centroid of the cluster containing it). This method tends to result in spherical, tightly bound clusters and is less sensitive to outliers. Ward's method can be

[4] some examples of such threshold are the desired number of clusters, the maximum number of documents in a cluster or the maximum similarity value below which mo merge is done.

implemented using the reciprocal-nearest neighbor (RNN) algorithm (Murtagh, 1983), which was modified for document clustering by Handouchi and Willett (1986).

- *Centroid/Median Methods*: Each cluster as is it formed is represented by the group centroid/median. At each stage of the clustering the pair of clusters with the most similar mean centroid/median is merged. The difference between the centroid and the median is that the second is not weighted proportionally to the size of the cluster.

The HAC approaches produce high quality clusters but have very high computational requirements (at least $O(n^2)$). They are typically greedy. This means that the pair of clusters that is chosen for agglomeration at each time is the one, which is considered the best at that time, without regard to future consequences. Also, if a merge that has taken place is not appropriate, there is no backtracking to correct the mistake.

There are many experiments in the literature comparing the different HAC methods. Most of them conclude that the single link method, although the only method applicable for large document sets, doesn't give high quality results (El-Hamdouchi and Willett, 1989; Willett, 1988; Steinbach et al., 2000). As for the best HAC method, the group average method seems to work slightly better than the complete link and Ward's method (El-Hamdouchi and Willett, 1989; Steinbach et al., 2000; Zhao and Karypis, 2002). This may be because the single link method decides using very little information and complete link considers the clusters to be very dissimilar. The group average method overcomes these problems by calculating the mean distance between the clusters (Steinbach et al., 2000).

3.1.3 Graph based clustering

In this case the documents to be clustered can be viewed as a set of nodes and the edges between the nodes represent the relationship between them. The edges bare a weight, which denotes the strength of that relationship. Graph based algorithms rely on *graph partitioning*, that is, they identify the clusters by cutting edges from the graph such that the edge-cut, i.e. the sum of the weights of the edges that are cut, is minimized. Since each edge in the graph represents the similarity between the documents, by cutting the edges with the minimum sum of weights the algorithm minimizes the similarity between documents in different clusters. The basic idea is that the weights of the edges in the same cluster will be greater than the weights of the edges across clusters. Hence, the resulting cluster will contain highly related documents.

The different graph based algorithms may differ in the way they produce the graph and in the graph partitioning algorithm that they use. Chameleon's (Karypis et al., 1999) graph representation of the document set is based on the k-nearest neighbor graph approach. Each node represents a document and there

exists an edge between two nodes if the document corresponding to either of the nodes is among the k most similar documents of the document corresponding to the other node. The resulting k-nearest neighbor graph is sparse and captures the neighborhood of each document. Chameleon then applies a graph partitioning algorithm, hMETIS (Karypis and Kumar, 1999) to identify the clusters. These clusters are further clustered using a hierarchical agglomerative clustering algorithm and based on a dynamic model (Relative Interconnectivity and Relative Closeness) to determine the similarity between two clusters. So, Chameleon is actually a *hybrid* (graph based and HAC) text-based algorithm.

Association Rule Hypergraph Partitioning (ARHP) (Boley et al., 1999) is another graph based approach which is based on hypergraphs. A hypergraphs is an extension of a graph in the sense that each hyperedge can connect more than two nodes. In ARHP the hyperedges connect a set of nodes that consist a frequent item set. A frequent item set captures the relationship between two or more documents and it consists of documents with many common terms characterising them. In order to determine these sets in the document collection and to weight the hyperedge, the algorithm uses an association rule discovery algorithm (Apriori). Then the hypergraph is partitioned using a hypergraph partitioning algorithm to get the clusters. This algorithm is used in the WebACE project (Han et al., 1997) to cluster web pages that have been returned by a search engine in response to a user's query. It can also be used for term clustering.

Another graph based approach is the algorithm proposed by Dhillon (Dhillon, 2001) which uses iterative bipartite graph partitioning to co-cluster documents and words. The advantages of these approaches are that can capture the structure of the data and that they work effectively in high dimensional spaces. The disadvantage is that the graph must fit the memory.

3.1.4 Neural Network based Clustering

The Kohonen's Self-Organizing feature Maps (SOM) (Kohonen, 1995) is a widely used unsupervised neural network model. It consists of two layers: the input layer with n input nodes, which correspond to the n documents, and an output layer with k output nodes, which correspond to k decision regions (i.e. clusters). The input units receive the input data and propagate them onto the output units. Each of the k output units is assigned a weight vector. During each learning step, a document from the collection is associated with the output node, which has the most similar weight vector. The weight vector of that 'winner' node is then adapted in such a way that it will become even more similar to the vector that represents that document, i.e. the weight vector of the output node 'moves closer' to the feature vector of the document. This process runs iteratively until there are no more changes in the weight vectors of the output nodes. The output of the algorithm is the arrangement of the input documents in a 2-dimensional space in such a way that the similarity between the input documents is mirrored in terms of topographic distance between the k decision regions.

Another approach proposed in the literature is the *hierarchical feature map* (Merkl, 1998) model, which is based on a hierarchical organization of more than one self-organizing feature maps. The aim of this approach is to overcome the limitations imposed by the 2-dimensional output grid of the SOM model, by arranging a number of SOMs in a hierarchy, such that for each unit on one level of the hierarchy a 2-dimensional self-organizing map is added to the next level.

Neural networks are usually useful in environments where there is a lot of noise, and when dealing with data with complex internal structure and frequent changes. The advantage of this approach is the ability to give high quality results without having high computational complexity. The disadvantages are the difficulty to explain the results and the fact that the 2-dimensional output grid may restrict the mirroring and result in loss of information. Furthermore, the selection of the initial weights may influence the result (Jain et al., 1999).

3.1.5 Fuzzy Clustering

All the aforementioned approaches produce clusters in such a way that each document is assigned to one and only one cluster. Fuzzy clustering approaches, on the other hand, are non-exclusive, in the sense that each document can belong to more than one clusters. Fuzzy algorithms usually try to find the best clustering by optimising a certain criterion function. The fact that a document can belong to more than one clusters is described by a *membership function*. The membership function computes for each document a membership vector, in which the i-th element indicates the degree of membership of the document in the i-th cluster.

The most widely used fuzzy clustering algorithm is Fuzzy c-means (Bezdek, 1984), a variation of the partitional k-means algorithm. In fuzzy c-means each cluster is represented by a *cluster prototype* (the center of the cluster) and the membership degree of a document to each cluster depends on the distance between the document and each cluster prototype. The closest the document is to a cluster prototype, the greater is the membership degree of the document in the cluster. Another fuzzy approach, that tries to overcome the fact that fuzzy c-means doesn't take into account the distribution of the document vectors in each cluster, is the Fuzzy Clustering and Fuzzy Merging algorithm (FCFM) (Looney, 1999). The FCFM uses Gaussian weighted feature vectors to represent the cluster prototypes. If a document vector is equally close to two prototypes, then it belongs more to the widely distributed cluster than to the narrowly distributed cluster.

3.1.6 Probabilistic Clustering

Another way of dealing with uncertainty is to use probabilistic clustering algorithms. These algorithms use statistical models to calculate the similarity between the data instead of some predefined measures. The basic idea is the assignment of probabilities for the membership of a document in a cluster. Each

document can belong to more than one cluster according to the probability of belonging to each cluster. Probabilistic clustering approaches are based on finite mixture modeling (Everitt and Hand, 1981). They assume that the data can be partitioned into clusters that are characterized by a probability distribution function (p.d.f.). The p.d.f. of a cluster gives the probability of observing a document with particular weight values on its feature vector in that cluster. Since the membership of a document in each cluster is not known a priori, the data are characterised by a distribution, which is the mixture of all the cluster distributions. Two widely used probabilistic algorithms are Expectation Maximization (EM) and AutoClass (Cheeseman and Stutz, 1996). The output of the probabilistic algorithms is the set of distribution function parameter values and the probability of membership of each document to each cluster.

3.1.7 Using Ontologies

The algorithms described above, most often rely on *exact* keyword matching, and do not take into account the fact that the keywords may have some *semantic proximity* between each other. This is, for example, the case with synonyms or words that are part of other words (whole-part relationship). For instance a document might be characterized by the words “camel, desert” and another with the word “animal, Sahara”. By using traditional techniques these documents would be judged unrelated. Using an ontology can help capture this semantic proximity of the documents. An ontology, in our context, is a structure (a lexicon) that organizes words in a net connected according to the semantic relationship that exists between them. More on ontologies can be found in Ding (2001).

THESUS (Varlamis et al.) is a system that clusters web documents that are characterized by weighted keywords of an ontology. The ontology used is a tree of terms connected according to the IS-A relationship. Given this ontology and a set of document characterized by keywords the algorithm proposes a clustering scheme based on a novel similarity measure between sets of terms that are hierarchically related. Firstly, the keywords that characterize each document are mapped onto terms in the ontology. Then, the similarity between the documents is calculated based on the proximity of their terms in the ontology. In order to do that, an extension of the Wu and Palmer similarity measure is used (Wu and Palmer, 1994). Finally, a modified version of the DBSCAN clustering algorithm is used to provide the clusters. The advantage of using an ontology in clustering is that it provides a very useful structure not only for the calculation of document similarity, but also for dimensionality reduction by abstracting the keywords that characterize the documents to terms in the ontology.

3.2. Link-based clustering

Text-based clustering approaches were developed for use in small, static and homogeneous collections of documents. On the contrary, the www is a huge collection of heterogeneous and interconnected web pages. Moreover, the web pages have additional information attached to them (web document metadata, hyperlinks) that can be very useful to clustering. According to Kleinberg (1997), '*the link structure of a hypermedia environment can be a rich source of information about the content of the environment*'. The link-based document clustering approaches take into account information extracted by the link structure of the collection. The underlying idea is that when two documents are connected via a link there exists a semantic relationship between them, which can be the basis for the partitioning of the collection into clusters.

The use of the link structure for clustering a collection is based on citation analysis from the field of bibliometrics (White and McCain, 1989). Citation analysis assumes that if a person creating a document cites two other documents then these documents must be somehow related in the mind of that person. In this way, the clustering algorithm tries to incorporate the human judgement when characterizing the documents. Two measures of similarity between two documents p and q based on citation analysis that are widely used are: *co-citation*, which is the number of documents that co-cite p and q and *bibliographic coupling*, which is the number of documents that are cited by both p and q. The greater the value of these measures the stronger the relationship between the documents p and q is. Also, the length of the path that connects two documents is sometimes considered when calculating the document similarity.

There are many uses of the link structure of a web page collection in web IR. Croft's Inference Network Model (Croft, 1993) uses the links that connect two web pages to enhance the word representation of a web page by the words contained in the pages linked to it. Frei & Stieger (1995) characterise a hyperlink by the common words contained in the documents that it connects. This method is proposed for the ranking of the results returned to a user's query. Page et al.(1998) also proposed an algorithm for the ranking of the search results. Their approach, PageRank, assigns at each web page a score, which denotes the importance of that page and depends on the number and importance of pages that point to it. Finally, Kleinberg proposed the HITS algorithm (Kleinberg, 1997) for the identification of mutually reinforcing communities, called hubs and authorities. Pages with many incoming links are called authorities and are considered very important. The hubs are pages that point to many important pages.

As far as clustering is concerned, one of the first link-based algorithms was proposed by Botafogo & Shneiderman (1991). Their approach is based on a graph theoretic algorithm that found strongly connected components in a hypertext's graph structure. The algorithm uses a *compactness* measure, which indicates the interconnectedness of the hypertext, and is a function of the average

link distance between the hypertext nodes. The higher that compactness the more relevant the nodes are. The algorithm identifies clusters as highly connected sub-graphs of the hypertext graph. Later, Botafogo (1993) extended his idea to include also the number of the different paths that connect two nodes in the calculation of the compactness. This extended algorithm produces more discriminative clusters, with reasonable size and with highly related nodes.

Another link-based algorithm was proposed by Larson (1996), who applied co-citation analysis to a collection of web documents. Co-citation analysis begins with the construction of a co-citation frequency matrix, whose ij-th entry contains the number of documents citing both documents i and j. Then, correlation analysis is applied to convert the raw frequencies into correlation coefficients. The last step is the multivariate analysis of the correlation matrix using multidimensional scaling techniques (SAS MDS), which mirrors the data onto a 2-dimensional map. The interpretation of the 'map' can reveal interesting relationships and groupings of the documents. The complexity of the algorithm is $O(n^2/2\text{-}n)$.

Finally, another interesting approach to clustering of web pages is trawling (Kumar et al., 1999), which clusters related web pages in order to discover new emerging cyber-communities that have not yet been identified by large web directories. The underlying idea in trawling is that these relevant pages are very frequently cited together even before their creators realise that they have created a community. Furthermore, based on Kleinberg's idea, trawling assumes that these communities consist of mutually reinforcing hubs and authorities. So, trawling combines the idea of co-citation and HITS to discover clusters. Based on the above assumptions, Web communities are characterized by dense directed bipartite subgraphs[5]. These graphs, that are the signatures of web communities, contain at least one core, which are complete directed bipartite graphs with a minimum number of nodes. Trawling aims at discovering these cores and then applies graph-based algorithms to discover the clusters.

3.3. Hybrid Approaches

The link-based document clustering approaches described above characterize the document solely by the information extracted from the link structure of the collection, just as the text-based approaches characterize the documents only by the words they contain. Although the links can be seen as a recommendation of the creator of one page to another page, they do not intend to indicate the similarity. Furthermore, these algorithms may suffer from poor or too dense link structures. On the other hand, text-based algorithms have problems when dealing with different languages or with particularities of the language (synonyms,

[5] A bipartite graph is a graph whose node set can be partitioned into two sets N_1 and N_2. Each directed edge in the graph is directed from a node in N_1 to a node in N_2

homonyms etc.). Also, web pages contain other forms of information except text, such as images or multimedia. As a consequence, hybrid document clustering approaches have been proposed in order to combine the advantages and limit the disadvantages of the two approaches.

Pirolli et al. (1996) described a method that represents the pages as vectors containing information from the content, the linkage, the usage data and the meta-information attached to each document. The method uses spreading activation techniques to cluster the collection. These techniques start by 'activating' a node in the graph (giving a starting value to it) and 'spreading' the value across the graph through its links. In the end, the nodes with the highest values are considered very related to the starting node. The problem with the algorithm proposed by Pirolli et al. is that there is no scheme for combining the different information about the documents. Instead, there is a different graph for each attribute (text, links etc.) and the algorithm is applied to each one, leading to many different clustering solutions.

The 'content-link clustering' algorithm, which was proposed by Weiss et al. (1996), is a hierarchical agglomerative clustering algorithm that uses the complete link method and a hybrid similarity measure. The similarity between two documents is taken to be the maximum between the text similarity and the link similarity:

$$S_{ij}=\max\left(S_{ij}^{terms}, S_{ij}^{links}\right) \qquad (1)$$

The text similarity is computed as the normalized dot product of the term vectors representing the documents. The link similarity is a linear combination of three parameters: the number of Common Ancestors (i.e. common incoming links), the number of Common Descendants (i.e. common outgoing links) and the number of Direct Paths between the two documents. The strength of the relationship between the documents is also proportional to the length of the shortest paths between the two documents and between the documents and their common ancestors and common descendants. This algorithm is used in the HyPursuit system to provide a set of services such as query routing, clustering of the retrieval results, query refinement, cluster-based browsing and result set expansion. The system also provides summaries of the cluster contents, called content labels, in order to support the system operations.

Finally, another hybrid text- and link-based clustering approach is the toric k-means algorithm, proposed by Modha and Spangler (2000). The algorithm starts by gathering the results returned to a user's query from a search engine and expands the set by including the web pages that are linked to the pages in the original set. Each document is represented as a triplet of unit vectors (D, F, B). The components D, F and B capture the information about the words contained in the document, the out-links originating at the document and the in-links terminating at the document, respectively. The representation follows the Vector Space Model, mentioned earlier. The document similarity is a weighted sum of

the inner products of the individual components. Each disjoint cluster is represented by a vector called 'concept triplet' (like the centroid in k-means). Then, the k-means algorithm is applied to produce the clusters. Finally, Modha & Spangler also provide a scheme for presenting the contents of each cluster to the users by describing various aspects of the cluster.

4. Comparison

The choice of the best clustering methods is a tedious problem, firstly, because each method has its advantages and disadvantages, and also because the effectiveness of each method depends on the particular data collection and the application domain (Jain et al., 1999; Steinbach et al., 2000).

There are many studies in the literature that try to evaluate and compare the different clustering methods. Most of them concentrate on the two most widely used approaches to text-based clustering: partitional and HAC algorithms. As mentioned earlier, among the HAC methods, the single link method has the lowest complexity but gives the worst results whereas group average gives the best. In comparison to the partitional methods, the general conclusion is that the partitional algorithms have lower complexities than the HAC, but they don't produce high quality clusters. HAC, on the other hand, are much more effective but their computational requirements forbid them from being used in large document collections (Steinbach et al. 2000; Zhao et Karypis, 2002; Cutting et al., 1992). Indeed, the complexity of the partitional algorithms is linear to the number of documents in the collection, whereas the HAC take at least $O(n^2)$ time. But, as far as the quality of the clustering is concerned, the HAC are ranked higher. This may be due to the fact that the output of the partitional algorithms depends on many parameters (predefined number of clusters, initial cluster centers, criterion function, processing order of documents). Hierarchical algorithms are more efficient in handling noise and outliers. Another advantage of the HAC algorithms is the tree-like structure, which allows the examination of different abstraction levels. Steinbach et al. (2000), on the other hand, compared these two categories of text-based algorithms and drove to slightly different conclusions. They implemented k-means and UPGMA in 8 different test data and found that k-means produces better clusters. According to them, this was because they used an incremental variation of the k-means algorithm and because they run the algorithm many times. When k-means is run more than one times it may give better clusters than the HAC. Finally, a disadvantage of the HAC algorithms, compared to partitional, is that they cannot correct the mistakes in the merges. This leads to the development of hybrid partitional – HAC methods, in order to overcome the problems of each method. This is the case with Scatter/Gather (Cutting et al., 1992), where a HAC algorithm (Buckshot or Fractionation) is used to select the initial cluster centers and then an iterative partitional algorithm is used for the refinement of the clusters, and with bisecting k-means (Steinbach

et al., 2000), which is a divisive hierarchical algorithm that uses k-means for the division of a cluster in two. Chameleon, on the other hand, is useful when dealing with clusters of arbitrary shapes and sizes. ARHP has the advantage that the hypergraphs can include information about the relationship between more than two documents. Finally, fuzzy approaches can be very useful for representing the human experience and because it is very frequent that a web page deals with more than one topic. In the table that follows (Table 1) there are presented the main text-based document clustering approaches according to various aspects of their features and functionality, as well as their most important advantages and disadvantages.

The link-based document clustering approaches exploit a very useful source of information: the link structure of the document collection. As mentioned earlier, compared to most text-based approaches, they are developed for use in large, heterogeneous, dynamic and linked collections of web pages. Furthermore, they can include pages that contain pictures, multimedia and other types of data and they overcome problems with the particularities of each language. Although the links can be seen as a recommendation of a page's author to another page, they do not always intend to indicate the similarity. In addition, these algorithms may suffer from poor or dense link structures, in which case no clusters can be found because the algorithm cannot trace dense and sparse regions in the graph. The hybrid document clustering approaches try to use both the content and the links of a web page in order to use as much information as possible for the clustering. It is expected that, as in most cases, the hybrid approaches will be more effective.

5. Conclusions and Open Issues

The conclusion derived from the literature review of the document clustering algorithms is that clustering is a very useful technique and an issue that prompts for new solutions in order to deal more efficiently and effectively with the large, heterogeneous and dynamic web page collections. Clustering, of course, is a very complex procedure as it depends on the *collection* on which it is applied as well as the choice of the various *parameter values*. Hence, a careful selection of these is very crucial to the success of the clustering. Furthermore, the development of link-based clustering approaches has proven that the links can be a very useful source of information for the clustering process.

Although there is already much research conducted on the field of web document clustering, it is clear that there are still some open issues that call for more research. These include the achievement of better *quality-complexity tradeoffs*, as well as effort to deal with each method's disadvantages. In addition, another very important issue is *incrementality*, because the web pages change very frequently and because new pages are always added to the web. Also, the fact that very often a web page relates to more than one subject should also be considered and lead to algorithms that allow for *overlapping clusters*. Finally,

more attention should also be given to the description of the clusters' contents to the users, the *labeling issue*.

Name	Complexity Time Space	Input	Output	Similarity criterion	Type of clusters	Overlap	Handling Outliers	Advantages	Disadvantages
Single linkage	$O(n^2)$ $O(n)$ (Time: $O(n\log n)$ – $O(n^5)$)	Similarity matrix	Assign documents to clusters, dendrogram	Join clusters with most similar pair of documents	Few, long, ellipsoidal loosely bound, chaining effect	Crisp clusters	No	• Sound theoretical properties • Efficient implementations	• Not suitable for poorly separated clusters • Poor quality
Group Average	$O(n^2)$ $O(n)$	Similarity matrix	Assign documents to clusters, dendrogram	Average pairwise similarity between all objects in the 2 clusters	Intermediate in tightness between single and complete linkage	Crisp clusters	No	• High quality results	• Expensive in large collections
Complete linkage	$O(n^3)$ $O(n^2)$ (worst case) in sparse matrix less	Similarity matrix	Assign documents to clusters, dendrogram	Join cluster with least similar pair of documents	Small, tightly bound	Crisp clusters	No	• Good results (Voorhees alg.)	• Not applicable in large datasets
Ward's Method	$O(n^2)$ $O(n)$	Similarity matrix	Assign documents to clusters, dendrogram	Join clusters whose merge minimizes the increase in the total error sum of squares	Homogeneous clusters, symmetric hierarchy	Crisp clusters	No	• Good at discovering cluster structure	• Very sensitive to outliers • Poor at recovering elongated clusters
Centroid/ Median HAC	$O(n^2)$ $O(n)$	Similarity matrix	Assign documents to clusters	Join clusters with most similar centroids/ medians	-	Crisp clusters	No		• Small changes may cause large changes in the hierarchy
K-means	$O(nkt)$ $O(n+k)$ (k: initial clusters, t: iterations)	K, iter Feature vector matrix	Assign documents to clusters, refinement of initial clusters	Euclidean or cosine metric	Arbitrary sizes	Crisp clusters	No	• Efficient (no sim matrix required) • Suitable for large datasets	• Very sensitive to input parameters
Single-Pass	$O(n\log n)$ $O(n)$	Similarity threshold, Feature vector matrix	Assign documents to clusters	If distance to closest centroid> threshold assign, else create new cluster	Large	Crisp clusters	No	• Efficient • Simple	• Results depend on the order of document presentation to the algorithm
Cheme-leon	$O(nm+n\log n+m^2\log m)$ m: sub-clusters	k (for knn graph), MINSIZE, scheme for combining	Assign documents to clusters, dendrogram	Relative Interconnectivity, Relative Closeness	Natural, homogeneous, arbitrary sizes	Crisp clusters	Yes	• Dynamic modelling	• Very sensitive to parameters • Graph must fit memory • Cannot correct

Name	Complexity Time Space	Input	Output	Similarity criterion	Type of clusters	Overlap	Handling Outliers	Advantages	Disadvantages
		RI, RC							merges
ARHP	O(n) O(n)	Apriori, HMETIS parameters, confidence threshold	Assign documents to clusters	Min-cut of hyperedges	-	Crisp clusters	Yes	• Efficient • No centroid /similarity measure	• Sensitive to the choice of Apriori parameters
Fuzzy C-Means	O(n)	Initial c prototypes	Membership values for each document (u_{ik})	Minimize $\Sigma\Sigma u_{ik} d^2(x_k, u_i)$	hyperspherical, same sizes	Fuzzy clusters	No	• Handles uncertainty • Reflects the human experience	• Sensitive to initial parameters • Poor at recovering clusters with different densities
SOM	$O(k^2 n)$ (k: input units)	Weights (m_i)	Topological ordering of input patterns	$m_i(t+1) = m_i(t) + \alpha(t) * h_{ci}(t) * [x(t) - m_i(t)]$	hyperspherical	-	Yes	• Suitable for collections that change frequently	• Fixed number of output nodes limits interpretation of results
Scatter/ Gather	Buckshot: O(kn) Fractionation: O(nm)	k: number of clusters	Assign documents to clusters with short summary	Hybrid: first partitional, then HAC	-	Crisp clusters	No	• Dynamic Clustering • Clusters presented with summaries • Fast	• Must have a very quick clustering algorithm • Focus on speed but not on accuracy
Suffix Tree Clustering	O(n)	Similarity threshold for the merge of the base clusters	Assign documents to clusters	- Sim = 1 if $\|Bm \cap Bn\|/\|Bm\| >$ threshold and $\|Bm \cap Bn\|/\|Bn\| >$ threshold, else - Sim = 0	-	Fuzzy clusters	No	• Incremental • Captures the word sequence	• Snippets usually introduce noise • Snippets may not be a good description of a web page

Table 1. Comparison of the clustering algorithms

References

Bezdek, J.C., Ehrlich, R., Full, W. 1984. FCM: Fuzzy C-Means Algorithm. Computers and Geosciences

Boley, D., Gini, M., Gross, R., Han, E.H., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J. 1999. Partitioning-based clustering for web document categorization. Decision Support Systems, 27(3):329-341

Botafogo, R.A., Shneiderman, B. 1991. Identifying aggregates in hypertext structures. Proc. 3rd ACM Conference on Hypertext pp.63-74

Botafogo, R.A. 1993. Cluster analysis for hypertext systems. Proc. ACM SIGIR Conference on Research and Development in Information Retrieval, pp.116-125

Cheeseman, P., Stutz, J. 1996. Bayesian Classification (AutoClass): Theory and Results. Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, pp. 153-180

Croft, W. B. 1993. Retrieval strategies for hypertext. Information Processing and Management, 29:313-324

Cutting, D.R., Karger, D.R., Pedersen, J.O., Tukey, J.W. 1992. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. Proc. ACM SIGIR Conference on Research and Development in Information Retrieval, pp.318-329

Defays, D. 1977. An efficient algorithm for the complete link method. The Computer Journal, 20:364-366

Dhillon, I.S. 2001. Co-clustering documents and words using Bipartite Spectral Graph Partitioning. UT CS Technical Report # TR 2001-05 20(http://www.cs.texas.edu/users/inderjit/public_papers/kdd_bipartite.pdf)

Ding, Y. 2001. *IR and AI: The role of ontology*. Proc. 4th International Conference of Asian Digital Libraries, Bangalore, India

El-Hamdouchi, A., Willett, P. 1986. Hierarchic document clustering using Ward's method. Proceedings of the Ninth International Conference on Research and Development in Information Retrieval. ACM, Washington, pp.149-156

El-Hamdouchi, A., Willett, P. 1989. Comparison of hierarchic agglomerative clustering methods for document retrieval. The Computer Journal 32

Everitt, B. S., Hand, D. J. 1981. Finite Mixture Distributions. London: Chapman and Hall

Frei, H. P., Stieger, D. 1995. The Use of Semantic Links in Hypertext Information Retrieval. Information Processing and Management, 31(1):1-13

Han, E.H., Boley, D., Gini, M., Gross, R., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., Moore, J. 1997. WebACE: a web agent for document categorization and exploration. Technical Report TR-97-049, Department of Computer Science, University of Minnesota, Minneapolis (http://www-users.cs.umn.edu/~karypis/publications/ir.html)

Jain, A.K., Murty, M.N., Flyn, P.J. 1999. Data Clustering: A Review. ACM Computing Surveys, Vol. 31, No. 2

Karypis, G., Han, E.H, Kumar, V. 1999. CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modelling. IEEE Computer, 32(8):68-75

Karypis, G., Kumar, V. 1999. A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing, 20(1)

Kleinberg, J. 1997. Authoritative sources in a hyperlinked environment. Proc. of the 9th ACM-SIAM Symposium on Discrete Algorithms

Kohonen, T. 1995. Self-organizing maps. Springer-Verlag, Berlin

Kumar, S.R., Raghavan, P., Rajagopalan, S., Tomkins, A. 1999. Trawling the Web for Emerging Cyber-Communities. Proc. 8th WWW Conference

Larson, R.R. 1996. Bibliometrics of the World Wide Web: An Exploratory Analysis of the Intellectual Structure of Cyberspace. Proc. 1996 American Society for Information Science Annual Meeting

Looney, C. 1999. A Fuzzy Clustering and Fuzzy Merging Algorithm, Technical Report,CS-UNR-101-1999 (http://www.cs.unr.edu/~looney/cs479/newfzclst2.pdf)

Merkl, D. 1998. Text Data Mining. Dale, R., Moisl, H., Somers, H. (eds.), A handbook of natural language processing: techniques and applications for the processing of language as text, Marcel Dekker, New York

Modha, D., Spangler, W.S. 2000. Clustering hypertext with applications to web searching. Proc. ACM Conference on Hypertext and Hypermedia

Murtagh, F. 1983. A survey of recent advances in hierarchical clustering algorithms. The Computer Journal, 26:354-359

Page, L., Brin, S., Motwani, R., Winograd, T. 1998. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford (http://www.stanford.edu/~backrub/pageranksub.ps)

Pirolli, P., Pitkow, J., Rao, R. 1996. Silk from a sow's ear: Extracting usable structures from the Web. Proc. ACM SIGCHI Conference on Human Factors in Computing

Rasmussen, E. 1992. Clustering Algorithms. Information Retrieval, W.B. Frakes & R. Baeza-Yates, Prentice Hall PTR, New Jersey

Salton, G., Wang, A., Yang, C. 1975. A vector space model for information retrieval. Journal of the American Society for Information Science, 18:613-620

Sibson, R. 1973. SLINK: an optimally efficient algorithm for the single link cluster method. The Computer Journal 16:30-34

Steinbach, M., G. Karypis, G., Kumar, V. 2000. A Comparison of Document Clustering Techniques. KDD Workshop on Text Mining

Strehl, A., Joydeep, G., Mooney, R. 2000. Impact of Similarity Measures on Web-page Clustering. Proc. 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search, pp.30-31

Van Rijsbergen, C. J.. 1979. Information Retrieval. Butterworths

Varlamis, I., Vazirgiannis, M., Halkidi, M., Nguyen, B. THESUS: Effective Thematic Selection And Organization Of Web Document Collections based on Link Semantics. To appear in the IEEE Transactions on Knowledge And Data Engineering Journal

Voorhees, E. M. 1986. Implementing agglomerative hierarchic clustering algorithms for use in document retrieval. Information Processing & Management, 22:465-476

Weiss, R., Velez, B., Sheldon, M., Nemprempre, C., Szilagyi, P., Gifford, D.K. 1996. HyPursuit: A Hierarchical Network Search Engine that Exploits Content-Link Hypertext Clustering. Proc. Seventh ACM Conference on Hypertext

White, D.H., McCain, K.W. 1989. Bibliometrics. Annual Review of Information Science Technology, 24:119-165

Willett, P. 1988. Recent Trends in Hierarchic document Clustering: a critical review. Information & Management, 24(5):577-597

Wu, Z., Palmer, M. 1994. Verb Semantics and Lexical Selection. 32nd Annual Meetings of the Associations for Computational Linguistics, pp.133-138

Zamir, O., Etzioni, O. 1998. Web document clustering: a feasibility demonstration. Proc. of SIGIR '98, Melbourne, Appendix-Questionnaire, pp.46-54

Zhao, Y., Karypis, G. 2001. Criterion Functions for Document Clustering: Experiments and Analysis. Technical Report #01-40. University of Minnesota, Computer Science Department. Minneapolis, MN (http://www-users.cs.umn.edu/~karypis/publications/ir.html)

Zhao, Y., Karypis, G. 2002. Evaluation of Hierarchical Clustering Algorithms for Document Datasets, ACM Press, 16:515-524