Assigning Web News to Clusters

Christos Bouras
Computer Engineering and Informatics Department, University of Patras and Research Academic Computer Technology Institute, N. Kazantzaki, Panepistimioupoli Patras, 26500 Greece
+30 2610 996954
bouras@cti.gr

Vassilis Tsogkas
Computer Engineering and Informatics Department, University of Patras, 26500, Greece
+30 2610 996954
tsogkas@ceid.upatras.gr

***Abstract*— The Web is overcrowded with news articles, an overwhelming information source both with its amount and diversity. Assigning news articles to similar groups, on the other hand, provides a very powerful data mining and manipulation technique for topic discovery from text documents. In this paper, we are investigating the application of a great spectrum of clustering algorithms, as well as similarity measures, to news articles that originate from the Web and compare their efficiency for use in an online Web news service application. We also examine the effect of preprocessing on clustering. Our experimentation showed that k-means, despite its simplicity, accompanied with preliminary steps for data cleaning and normalizing, gives better aggregate results when it comes to efficiency.**

Keywords- Web News Articles, Document Clustering, k-means, k-means++, Hierarchical Clustering

I. INTRODUCTION

News articles are flooding the Web every day from an extreme amount of major or minor news portals from around the globe. It's utterly impossible for a single individual to be able to keep track of an event, or a series of related events, from an unbiased and truly informative point of view. Clustering of news articles on the other hand, can help dealing with this situation by depicting the underneath content hierarchy of a huge amount of articles within the reach of a single individual. Consequently, clustering can provide to information retrieval (IR) systems with the potential to alleviate users while browsing and detecting quickly the needed information.

Using clustering techniques on news articles from the Web is not new. In [12] the authors study the formation of useful news clusters from a structural point of view by utilizing the links between Web pages containing similar information. This approach is however online, focusing on speed and not efficiency. We solve the problem differently: given the availability of data kept offline by our system, PeRSSonal [11], we aim at applying data manipulation techniques, along with clustering algorithms in order to find hidden relationships among the data. Discovering clusters within this overwhelming amount of data is expected to provide significantly improved results compared to offline clustering. This approach is similar to gene [13] and multimedia [14] clustering.

In this paper, we are describing a variety of document clustering techniques, and evaluating their application on our data set: news articles originating from the Web. Our aim is to compare the resulting clusters and determine which technique is best fitted for the extreme amount and diversity of news articles that an indexing system needs to address.

The rest of the paper is organized as follows. The next section gives a short establishment of the related work on document clustering in general. In Section III, we give a brief overview of our system which we are enhancing with clustering techniques, while in Section IV we elaborate more on our experimental approach towards the clustering methodologies used. Section V presents our evaluation results and Section VI concludes this paper with some remarks about the future work that is underway.

II. CLUSTERING METHODOLOGIES

Clustering data in general has been heavily researched by the scientific community over the last 20 years. Especially for document clustering, a huge variety of techniques has been proposed. A major goal of document clustering is to improve the results of information retrieval systems in terms of precision / recall, and thus serve better filtered and adequate results to their users, helping in essence the decision making process.

Two generic categories of the various clustering methods exist: agglomerative hierarchical and partitional. Typical hierarchical techniques [5] generate a series of partitions over the data, which may run from a single cluster containing all objects, to n clusters each containing a single object, and are widely visualized through a divisive (root to leaves) or agglomerative (leaves to root) tree structure. On the other hand, partitional algorithms typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering. For partitional techniques, a global criterion in most commonly used, the optimization of which drives the entire process.

We will now briefly elaborate more on the techniques that are applied within the scope of this paper to our clustering experimental approach.

A. Hierarchical Clustering

Divisive hierarchical methodologies generate a nested sequence of partitions, with a single, all-inclusive cluster at the top and singleton clusters of individual points at the bottom [1]. The vice-versa procedure occurs with agglomerative methodologies: the algorithm starts by considering each data point as a cluster of each own and proceeds by merging together tree nodes that share a certain degree of similarity.

In the above sense, hierarchical techniques require a cluster similarity or distance measure, in order to successively split clusters or merge data points belonging to different clusters. Most commonly, a similarity (distance) matrix is computed whose ij_{th} element expresses the distance between the i_{th} and j_{th} cluster. This matrix is updated on each step, where subsequent nodes are created by pairwise joining (for agglomerative) or splitting (for divisive) of nodes until the process is complete. The result of the above techniques is a tree-like structure, a dendrogram, displaying the merging process, and the intermediate clusters that occur during the procedure can be taken by "cutting" the tree at the required precision level. The aforementioned procedure is deterministic, compared with the ones described in the following subsection for partitional techniques. However, as explained in [2], sequential agglomerative hierarchical non-overlapping (SAHN) clustering methods, feature an average complexity of at least $O(n^2)$ and most commonly $O(n^3)$ - on the input size n - which in many cases is aversive for use with large datasets.

There are several flavors of hierarchical clustering techniques that we are evaluating in this paper. Their difference lies in how the distance between clusters is defined in terms of their members - articles. Typically, pairwise single, maximum, average, and centroid linkage distances between clusters are considered. For pairwise single linkage, the shortest among the pairwise distances of the clusters is considered as the inter-cluster distance, whereas for pairwise maximum linkage this is the longest among them. Moreover, for pairwise average linkage the mean of the pairwise distances is defined as the inter-cluster similarity (i.e. distance). Finally, for the centroid linkage, each cluster is represented by its centroid which is calculated on each step of the algorithm and the inter-cluster distance is the distance between the cluster centers.

B. Partitional Clustering

Contrary to hierarchical clustering, partitional techniques produce a single-level division of the data. Given the number of desired clusters, let k, partitional algorithms find all k clusters of the data at once, such that the sum of distances over the items to their cluster centers is minimal. Moreover, for a clustering result to be accurate, besides the low intra-cluster distance, high inter-cluster distances, i.e. well separated clusters, is desired. Typical partitional algorithms are: k-means, k-medians and k-medoids. These algorithms are based on the notion of the cluster center, a point in the data space, usually not existent in the data themselves, which represents a cluster. Their difference consists in how the cluster center is defined in each case. Following we will briefly describe each approach as well as some of their variations in the literature.

In k-means clustering, the cluster center is defined as the mean data vector averaged over all items in the cluster. In k-medians, instead of the mean, the median is calculated for each dimension in the data vector. Finally, in k-medoids the cluster center is defined as the item which has the smallest sum of distances to the other items in the cluster. K-medoids has the advantage of better handling of the outliers existing in data, while it does not depend on the order in which the objects are examined. The family of k-means partitional clustering algorithms [7] usually tries to minimize the average squared distance between points in the same cluster, i.e. if d1, d2,…, dn are the n documents and c1, c2,…, ck are the k clusters centroids, k-means tries to minimize the global criterion function:

$$\sum_{i=1}^{k}\sum_{j=1}^{n} sim(d_j, c_i) \qquad (1)$$

Typically, all those algorithms share the following Expectation Maximization (EM) [8] steps:

1. Randomly Select K points as the initial centroids
2. Assign all data to the closest centroid
3. Calculate the new centroids for each cluster
4. Repeat steps 2 and 3 until no reassignments of the centroids takes place.

Algorithm 1. Basic k-means EM algorithm

The EM algorithm suffers from frequently converging to local minima (or maxima), due to the random choice of the initial centroids. Computing thus a refined starting condition can yield significant improvements [3]. For example k-means++ [6], selects a point x as an initial cluster center, using a probability that is proportional to the square of the distance between each successive choice and the previous ones and then proceeds as k-means. This heuristic offers a significant boost compared with regular k-means as far as error and execution time are concerned. Another approach commonly used is multiple executions of the k-means algorithm, with different starting conditions, and finally keeping the best result; if a specific cluster assignment appears to be repeating, it is possible to be the best.

Bisecting k-means [4] introduces an alternative approach: initially the whole data set is treated as one cluster. A cluster is selected for split into two at each step by using a criterion such as the cluster size or the overall similarity. The split of the selected cluster is done using regular k-means and the procedure completes when the desired number of clusters is created. Consequently, unlike regular k-means, which splits the whole data set into k cluster at each iteration step, its bisecting variation splits only one existing cluster into two sub-clusters. The selection of which cluster to split can be based on its size, or on the centroid's neighbors network. Surprisingly, bisecting k-means is reported with a performance that generally beats k-means and even hierarchical approaches, while keeping the complexity linear.

The low complexity is commonplace for all of the previously mentioned partitional algorithms and thus they are best suited for clustering large document databases, as it is the case of this paper. Especially for Algorithm 1, the average complexity is linear in all relevant factors: iterations, number of clusters and number of documents, even though the worst case can get [8].

III. ARCHITECTURAL OVERVIEW OF PERSSONAL

Our system, PeRSSonal [11], features a staged and modular approach for performing the various tasks

concerning news articles that originate from the Web. Figure 1 gives an overview of the system's architecture.

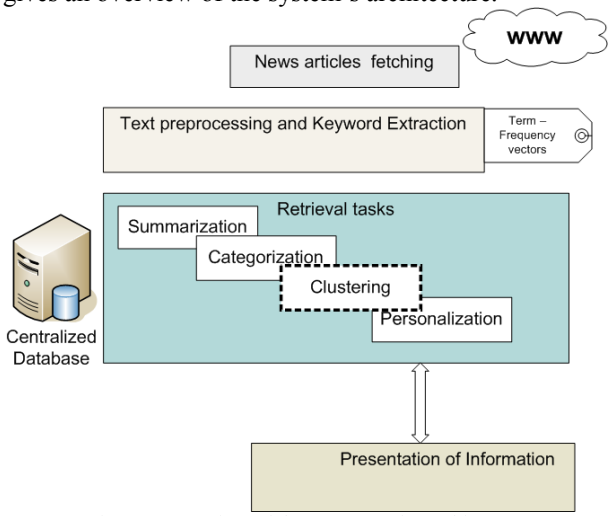

Figure 1 Overview of the PeRSSonal's architecture

At its input stage, PeRSSonal crawls and fetches news articles from major or minor news portals from around the world. This is an offline procedure and once articles as well as metadata information are fetched, they are stored in the centralized database from where they are picked up by the following procedures.

A key procedure of the system as a whole, which actually makes the rest of the steps viable, is text preprocessing on the fetched article's content, that results to the extraction of the keywords each article consists of. Analyzed in [10], keyword extraction applies several heuristics to come up with a weighting scheme that appropriately weights the keywords of each article based on information about the rest of the documents in our database. Keyword extraction in essence generates the term-frequency vector for each article that is used by the information retrieval techniques that follow. In this paper, we are using the results of this step, which is a weighted scheme of stemmed nouns existing in the original text, as input to various clustering approaches. Our target is to determine the effect of text preprocessing, as far as stemming and noun extraction are concerned, on the clustering process.

Text summarization, categorization of the articles on a predetermined set of classes, as well as personalization of the results, are some additional steps deployed in order to extract useful information from the data [9]. It is this level of the system that we are enhancing in this paper with the application of document clustering algorithms, in order to generate better results that the system's users view. Following the retrieval techniques, information is transmitted back to the end user.

IV. Clustering News Articles

A. Clustering Process

The overall clustering process as evaluated in this paper proceeds as depicted in Figure 2.

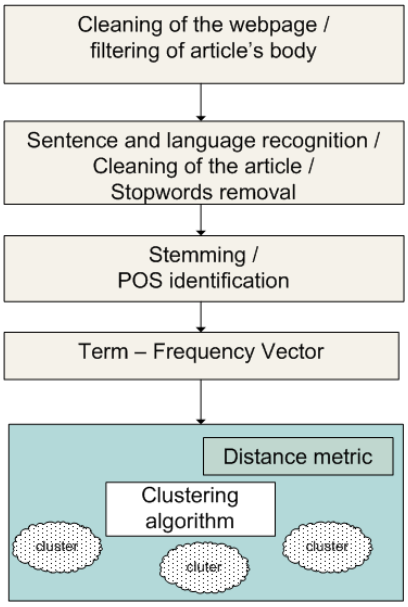

Figure 2 Flow of information for evaluating the clustering methodologies

Once the articles are fetched, we proceed with cleaning the originating webpage and keeping the article's body. This is afterwards analyzed by recognizing its language and its sentences. Following, the article's stopwords (i.e. words without any practical significance) are removed; the part of speech identification takes place [10] then, while a stemmer keeps only the stems of the words. Next comes the creation of the term - frequency vector for the article, which is given as input to the subsystem evaluated in this paper: the clustering kernel. By applying a variety of clustering algorithms and distance metrics, we try to determine whether preprocessing has an effect on the domain of clustering news articles and which approach benefits the most and would thus be applicable in our case. Most importantly, we try to estimate the effect of noun identification and stemming on each clustering approach and thus utilize the algorithm that will prove to be the most effective for the domain of Web news articles.

An important aspect that has to do with news articles in general, is their diversity and similarity at the same time. When fetching information from numerous news portals, it is normal to expect a certain degree of similarity, as far as the content is concerned, since a great amount of the published news articles are copied from other sources. However, it's important to be able to understand minor differences which may usually betray biases to certain opinions expressed in the articles. Moreover, when dealing with documents, the amount of terms that the system can possibly come across is limitless (even more when multiple languages are taken into consideration), compared for example with gene-clustering. The applied algorithms, as well as the similarity measures used should take into consideration the above.

Following, we describe the various similarity measures that are applied on high dimensionality sparse data within the scope of this paper.

B. Similarity Measures

All clustering methods described in Section 2 need to embed the documents to a suitable similarity space, thus share the notion of establishing the distance, i.e. similarity, between two data points, two clusters, or a data point and a cluster. In this paper, we are using the following distance functions for comparing the various methodologies:

- Euclidian, where the distance between two data points a and b is defined as:

$$d(a,b) = \frac{1}{n}\sum_{i=1}^{n}(a_i - b_i)^2 \qquad (2)$$

n being the dimensionality of the data. The Euclidean distance takes the magnitude of the input data into account and consequently preserves more information about them.

- City-block: $d(a,b) = \frac{1}{n}\sum_{i=1}^{n}|a_i - b_i|$ (3)
- Pearson correlation coefficient:

$$r(a,b) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{a_i-\bar{a}}{\sigma_\alpha}\right)\left(\frac{b_i-\bar{b}}{\sigma_b}\right) \qquad (4)$$

in which $\bar{a}$ and $\bar{b}$ are the sample mean of a and b respectively, and σ_α, σ_b are the sample standard deviation of a and b. The Pearson correlation coefficient can be thought as a measure for how well a straight line can be fitted to a scatterplot of a and b. The Pearson correlation coefficient is either +1 or −1 for the points in the scatterplot that lie on a straight line. Note that the Pearson distance is thus defined as: $d(a,b) = 1 - r$ (5)

- Cosine similarity: $d(a,b) = \cos(\theta) = \frac{a \bullet b}{|a||b|}$ (6)

where the similarity between the two data points is viewed by means of their angle in the n-dimensional space.

- Spearman-rank correlation ρ,

which is a non-parametric measure that performs well against outliers. It originates from the Pearson correlation by replacing every data value with its rank having the values firstly ordered. Due to the diminishing of the data values, there is no weight information taking place to the distance calculation compared to the previous – parametric similarity measures. The Spearman-rank distance between two data points a,b is defined as: $d(a,b) = 1 - \rho$ (7)

- Kendall's τ,

which is similar to the Spearman rank correlation, but using the relative ranks instead of the absolute ones. The Kendall's distance between two data points a,b is defined as:

$$d(a,b) = 1 - \tau \qquad (8)$$

Once the distance measure is defined, each clustering algorithm proceeds by calculating the distance matrix containing all the distances between the items that are being clustered. From the above distance functions, only Euclidian and City-block distance are true metrics since they satisfy the triangle inequality.

V. Evaluation

In the current section we are presenting the results of our experimental procedure for clustering news articles. In this frame we conducted a series of experiments on a predetermined set of news articles that are available in the system's database and have been offline analyzed as explained in section 3. Our dataset consists of 10000 randomly selected news articles originating from 20 major news portals, with a time span of 3 months. The news articles belong equivalently to seven basic domains: business, politics, health, education, science, sports and entertainment. After the preprocessing procedure described in Section 3, and most notably stemming and noun identification, we have kept for each article its list of stemmed nouns. Notice that duplicate articles originating from different sources have been removed from the dataset based on their title and main body.

On this dataset we applied the afore-mentioned clustering methodologies: single, maximum, linkage and centroid linkage hierarchical clustering, as well as regular k-means, k-medians and k-means++. For those, we utilized the open source clustering library described in [15] as well as the k-means++ implementation from [6]. Furthermore, for each of the above techniques, except k-means++ (which only supports Euclidian) we used the similarity measures described in section 2.3, i.e., Euclidian distance, city-block distance, Pearson correlation coefficient, cosine similarity, Spearman-rank correlation and Kendall's τ. For partitional algorithms, we used a 10 pass scheme with different starting conditions in order to avoid phenomena of local minima for the distance measures.

In order to determine the efficiency of each clustering method, we use the notion of Clustering Index (CI) as explained in [16]. Intuitively, since the most efficient clusters are the ones containing articles close to each other within the cluster, while sharing a low similarity with articles belonging to different clusters, CI focuses on increasing the first measure (intra-cluster similarity) while decreasing the second (inter-cluster similarity). The Clustering Index of each pass is defined as: $CI = \frac{\bar{\sigma}^2}{\bar{\sigma} + \bar{\delta}}$ (9)

where $\bar{\sigma}$ is the average intra-cluster similarity and $\bar{\delta}$ the average inter-cluster similarity. Furthermore, for determining the similarity between two articles we used the distance vector which is produced using the respective similarity measure per case. The results for each clustering methodology and distance measure run for a number of clusters from 100 to 1000, are depicted in Figures 3-8. The notions mentioned in the graphs are explained in Table 1.

TABLE 1 HIERARCHICAL CLUSTERING NOTATIONS

Type of distance	Distance Between two Clusters
Pairwise Maximum (complete) linkage	PCL
Pairwise Single linkage	PSL
Pairwise Centroid linkage	PKL
Pairwise Average linkage	PAL

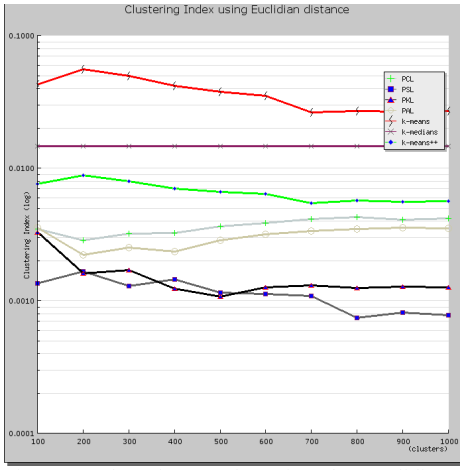

Figure 3 Clustering results using the Euclidian distance

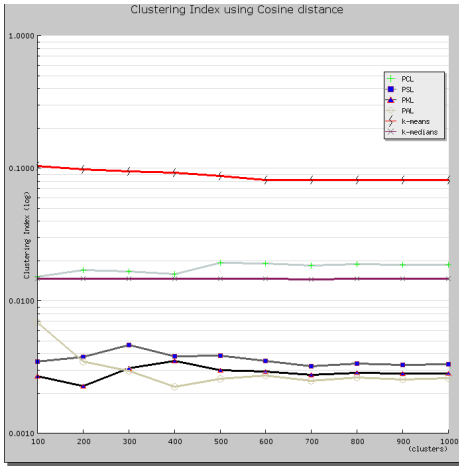

Figure 4 Clustering Results using the cosine distance

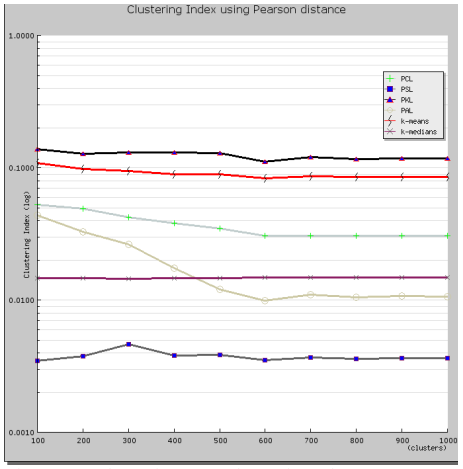

Figure 5 Clustering Results using the Pearson's distance

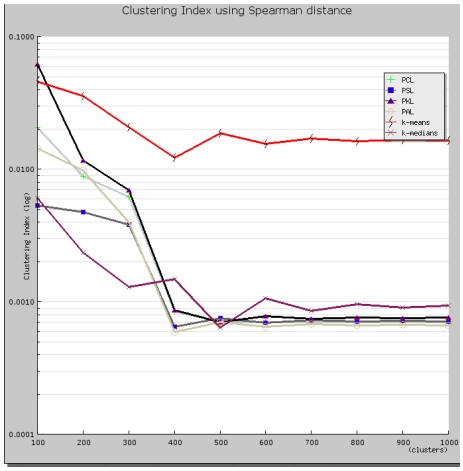

Figure 6 Clustering Results using the Spearman distance

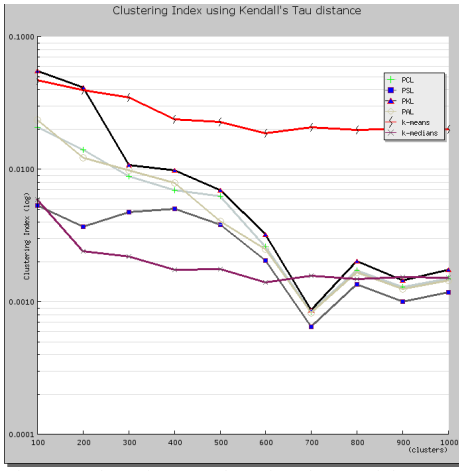

Figure 7 Clustering Results using the Kendal's tau distance

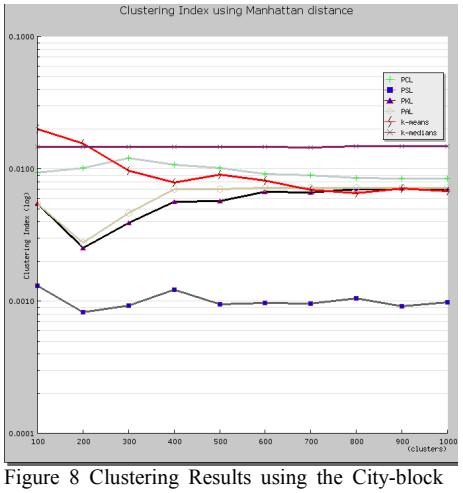

Figure 8 Clustering Results using the City-block distance

From the above graphs, k-means almost always outperforms any other clustering approach. Furthermore, cosine similarity and Euclidian distance proves better for k-means, since the clusters seem better connected, rather than with the city-block distance, which seems to be better fit to k-medians. Another observation is that the number of clusters directly affects the CI metric and that after a certain cluster threshold, each algorithm deteriorates in terms of CI. For example, the best CI for partitional algorithms is observed for k-means/cosine similarity and 100 clusters followed by k-means/Euclidian and 200 clusters. The best CI scores for hierarchical algorithms are observer for PKL and the Pearson's distance. Moreover, for most similarity measures we observed lower CI scores for hierarchical methodologies compared to partitional approaches. This originates from the manner that those algorithms operate when "cutting" the dendrogram: generation of many singleton clusters and a few clusters containing many articles.

As far as partitional clustering is concerned, k-means outperforms k-medians and even k-means++ which seems to deteriorate sooner as the number of clusters increases. Moreover, as Figure 9 shows, k-means++ is significantly slower than its counterparts given the number of clusters.

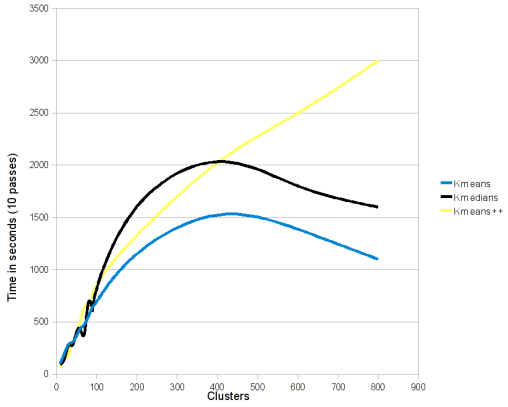

Figure 9 Average intra-cluster sum of distances for partitional clustering;

Following, we repeated the aforementioned experimentation omitting the steps of stemming and noun identification from the preprocessing procedure. The average

modification of the CI results is presented in Table 2. Clearly, stemming and noun identification on the article's keywords has a significantly beneficial effect for all methodologies, especially for k-means, explaining partly the CI results presented earlier in Figures 3-8.

TABLE 2 THE EFFECT OF PREPROCESSING ON CLUSTERING METHODOLOGIES

Clustering Method	Percent increase of CI when using stemming and noun identification
PCL	5%
PSL	6%
PKL	6%
PAL	5%
k-means	18%
k-medians	16%
k-means++	15%

Even though internal objective functions like CI are capable of giving a generic overview of the clustering process efficiency, an alternative approach is user-based evaluation. Based on this intuition, for our final set of experiments we tried to evaluate the generated clusters by using a group of 10 individuals. We requested that they grouped 50 random articles from the previous data set into 10 clusters according to their personal opinion. Afterwards we averaged their clustering selections and compared those results with the clustering passes of each of the various methodologies explained earlier using the Euclidian similarity distance. The evaluation metric at this case is the F measure, i.e. the weighted harmonic mean of the precision and recall observed between the users choices and the results generated by each clustering pass. The F results per clustering pass, depicted in Table 3 show that from a user based perspective, the resulting clusters produced by k-means are closer to what most of the users selected for the selected data set of articles.

TABLE 3 USERS' EVALUATION OF THE CLUSTERING METHODOLOGIES

Clustering Method	F result
PCL	0.42
PSL	0.42
PKL	0.43
PAL	0.41
k-means	0.61
k-medians	0.57
k-means++	0.51

VI. CONLCUSIONS

Within the scope of our indexing system, we have presented our evaluation results comparing some of the best clustering options currently available, applying them to the domain of news articles that originate from the Web. From the plethora of similarity measures that have been used, the appliance of Euclidian and cosine k-means produced the best results based not only on the internal CI function, but also on a real users' experimentation. More specifically, we have found that hierarchical clustering techniques resulted generally in worse CI scores, while partitional clustering, even though non-deterministic, can provide exceptional results. Another important finding is that preprocessing of the articles via stemming and noun identification can improve significantly the clustering results by a factor of 5-15% depending on the clustering algorithm.

Since we are only at the beginning of the clustering kernel of our application, we are aiming to its stability at first and its everyday use by the system daemons, in order to serve clusters of articles and not mere articles to the user. This has various future implications to the profile generation procedure for the system users. Moreover, we will be researching towards using the clustering kernel for clustering system users based on their dynamic profiles, and we will proceed with evaluating more extensively the clustering module with user feedback.

REFERENCES

[1] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques", Department of Computer Science and Engineering, University of Minnesota 2000

[2] W.H.E.. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods", J. Classification 7--24 (1984)

[3] P.S. Bradley and U. Fayyad, "Refining Initial Points for K-means Clustering", Proc. 15th Int'l Conf. Machine Learning, pp. 91--99, (1998)

[4] L. Yanjun and C. Soon, "Parallel bisecting k-means with prediction clustering algorithm". The Journal of Supercomputing, 39:19--37 (2007)

[5] A. El-Hamdouchi and P. Willett, "Comparison of hierarchic agglomerative clustering methods for document retrieval". The Computer Journal 32, pp. 220--227 (1989)

[6] D. Arthur and S. Vassilvitskii, (2007). "k-means++: the advantages of careful seeding". Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms: 1027--1035

[7] Y. Zhao and G. Karypi, "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering", Machine Learning, v.55 n.3, p.311--331, (2004)

[8] D. Arthur and S. Vassilvitskii, "On the Worst Case Complexity of the k-means Method". Technical Report. Stanford (2005)

[9] C. Bouras, V. Poulopoulos, and V. Tsogkas, "PeRSSonal's core functionality evaluation: Enhancing text labeling through personalized summaries", Data and Knowledge Engineering Journal, Elsevier Science, Vol. 64, Issue 1, pp. 330 – 345, (2008)

[10] C. Bouras and V. Tsogkas, "Improving text summarization using noun retrieval techniques", Lecture Notes in Computer Science. Knowledge-Based Intelligent Information and Engineering Systems, Vol. 5178/2008 pp. 593-600 (2008)

[11] PeRSSonal's Website, http://perssonal.cti.gr/

[12] D.R. Radev, S. Blair-Goldensohn, Z. Zhang, and R.S. Raghavan, "Interactive, Domain-Independent Identification and Summarization of Topically Related News Articles", Proceedings of the 5th European Conference on Research and Advanced Technology for Digital Libraries, pp. 225--238, (2001)

[13] M. Eisen, P. Spellman, P. Brown and D. Botstein, "Cluster analysis and display of genome-wide expression patterns". Proc. Natl. Acad. Sci., USA, 95, 14863–14868 (1998)

[14] A. Hinneburg and D. Keim "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", Proc. 4th Int. Conf. on Knowledge Discovery & Data Mining, New York City, NY, 1998

[15] M. de Hoon, S. Imoto and S. Miyano, "The C Clustering Library", Institute of Medical Science, Human Genome Center, University of Tokyo (2003).

[16] J. Taeho and L. Malrey "The Evaluation Measure of Text Clustering for the Variable Number of Clusters", Advances in Neural Networks ISNN 2007 Volume 4492/2007 pp. 871-879