Clustering Text Documents: An Overview

Radu CREȚULESCU*, Daniel MORARIU*, Lucian VINȚAN*

** "Lucian Blaga" University of Sibiu, Romania, Engineering Faculty, Emil Cioranst. no. 4, 550025 Sibiu,
Phone +40-269-217928, Fax +40-269-212716, e-mail {radu.kretzulescu, daniel.morariu,
lucian.vintan}@ulbsibiu.ro*

Abstract: Clustering is an important process in text mining used for groping documents based on their contents in order to extract knowledge. In data mining process, clustering and cluster analysis can be used as stand-alone application for data distribution analysis, in order to observe certain characteristics of each cluster (group). In this paper we will present some requirements for clustering algorithms, and then we will present a possible taxonomy of the commonly used clustering algorithms. In the last section we will discuss the external and internal techniques for clustering validation**.**

Key words: Text Mining, Unsupervised leaning, Clustering, Cluster Validation

1. Introduction

In the recent years, significant increases in using the Web and the improvements of the quality and speed of the Internet have transformed our society into one that depends strongly on the information. The huge amount of data that is generated by this process of communication represents important information that accumulates daily and that is stored in form of text documents, databases etc. The retrieving of this data is not simple and therefore the data mining techniques were developed for extracting information and knowledge that are represented in patterns or concepts that are sometimes not obvious.

As mentioned in [5, 9], machine learning software provides the basic techniques for data mining by extracting information from raw data contained in databases. The process usually goes through the following steps:

- transform the data into a suitable format
- data cleaning
- deduction or conclusions on the extracted data.

Machine learning techniques are divided into two sub domains: supervised learning and unsupervised learning. Under the category of unsupervised learning, one of the main tools is data clustering. This paper attempts to provide taxonomy of the most important algorithms used for clustering. For each algorithm category, we selected the most common version of this entire family. Below we present algorithms used in context of document clustering.

2. Unsupervised versus supervised learning

In supervised learning, the algorithm receives data (the text documents) and the class label for the corresponding classes of the documents (called labeled data). The purpose of supervised learning is to learn the concepts that correctly classify documents for given classification algorithm. Based on this learning the classifier will be able to predict the correct class for unseen examples. Under this paradigm, it is also possible the appearance of the over-fitting effects. This will happen when the algorithm memorizes all the labels for each case.

The outcomes of supervised learning are usually assessed on a disjoint test set of examples from the training set examples. Classification methods used are varied, ranging from traditional statistical approaches, neural networks to kernel type algorithms [6].

The quality measure for classification is given by the accuracy of classification.

In unsupervised learning the algorithm receives only data without the class label (called unlabeled data) and the algorithm task is to find an adequate representation of data distribution.

The central point of this paper is to present the clustering as a key aspect in unsupervised learning.

Some researchers have combined unsupervised and supervised learning that has emerged the concept of semi-supervised learning [4]. In this approach is applied initially an unknown data set in order to make some assumptions about data distribution and then this hypothesis is confirmed or rejected by a supervised approach.

3. Cluster Analysis

Cluster analysis is an iterative process of clustering and cluster validation facilitated by clustering algorithms and cluster validation methods. Cluster analysis includes two major aspects: clustering and cluster validation.

Clustering refers to grouping objects according to certain criteria. To achieve this goal researchers have developed many algorithms [5, 13, 14, 15]. Since there are no general algorithms that can be applied to all types of cases it becomes necessary to apply a validation mechanism so that the user will find an algorithm suitable for its particular case.

Cluster analysis is a process of discovery through exploration. It can be used to discover structures without the need for interpretation [13].

The validation of the clusters becomes a cluster quality evaluation process.

3.1 Definitions

Def. 4.1: Clustering is the process of grouping physical or abstract objects into classes of similar properties. [14]

A cluster is a collection of objects that are similar to each other from the same group and are dissimilar to objects that are form different groups.

Def. 4.2 Conceptual clustering – is a process of grouping where the objects in a group will form a class only if they can be described by one concept. This approach differs from conventional clustering where the dissimilarity measure is based on mathematical distances.

The basic idea for clustering, which is the starting point for both previous presented approaches, is to find those clusters with high inter-cluster similarity and very small intra-cluster similarity.

The main directions of clustering research are focused more on distance-based cluster analysis. Several algorithms have been developed such as partitional algorithms like k-Means and k-Medoids, or hierarchical algorithms. Further investigations have demonstrated the usefulness of other approaches such as algorithms based on suffix trees [20, 27], the bio-inspired algorithms, such as those based on the behavior of ants in search of food and creation of cemeteries [1, 7, 17], particles swarms (particle swarm optimization) [21] and ontologies [3, 12].

3.2. General requirements for clustering algorithms

Typical requirements for clustering algorithms in data mining and text documents [9] are:

a. Scalability of the algorithms: Many clustering algorithms work very well with small data sets. However, large databases containing millions of objects and using a large sample of these sets would lead to inconclusive results.
b. Ability to use different types of attributes: Many clustering algorithms use numerical data as input. In some cases it is necessary to apply clustering algorithms on string data, binary data, ordinal data or a combination thereof.
c. Clusters of arbitrary shape recognition: Many algorithms find clusters using geometric distances such as Euclidean distance or Manhattan distance. These algorithms, however, tend to find spherical clusters with similar size and density. In reality, clusters can be in any form.
d. Setting input parameters based on minimal knowledge of the field: Many clustering algorithms require certain parameters such as the number of clusters that need to be determined (e.g. *k*-means). The result of clustering can be significantly different depending on the input parameters set. The input parameters for data sets containing high dimensional objects are difficult to be determined.
e. Ability to use data that contain noise: Many real data sets containing missing data, unknown or outliner. Some algorithms are sensitive to such data and clustering quality becomes poor.
f. Insensitivity to the order of data processing: Some clustering algorithms may produce different results for different order of the inputs (e.g., Single Pass, BIRCH- Balanced Iterative Reducing and Clustering).
g. High dimensionality: Data may contain a lot of dimensions and attributes. Many clustering algorithms fail to produce good results even for data with small dimensions. The human eye fails to assess the quality of a cluster by up to three dimensions.
h. Interpretability and usability: Users want that the clustering results shall be useful, interpretable and understandable.

4. Data used in clustering

4.1 Data representation

A data matrix represents for the clustering algorithm a matrix with *n* objects and each has *p* attributes. The representation of the data will be a matrix of *n* × *p* attributes.

$$\left[\begin{array}{c|cccc} x_{11} & \cdots & x_{1j} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{ip} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{np} \end{array}\right] \qquad (4.1)$$

4.2 Dissimilarity Matrix

This matrix will be a *N x N* square matrix and contains the dissimilarity measures of all pairs of objects for the clustering. Since *d(i, j) = d (j, i)* and *d(i, i) = 0* we obtain the following dissimilarity matrix

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \cdots & \cdots & \cdots & 0 & \\ d(n,1) & & & d(n,n-1) & 0 \end{bmatrix} \qquad (4.2)$$

where *d(i, j)* is the measure of dissimilarity between two objects.

Objects are clustered according to the similarity between them or by their dissimilarity.

4.3 Dissimilarity versus similarity

The dissimilarity *d(i, j)* is a positive number close to 0 when the documents *i* and *j* are close to each other and increases when the distance between i and j increases. The dissimilarity between two objects can be obtained by subjective evaluation made by experts based on direct observation or can be calculated based on correlation coefficients.

The similarity *s(i, j)* is a positive number close to 0 when the two documents are not similar and increases when the documents are more similar.

If the similarity and the dissimilarity coefficients are in the range [0, 1] the following equation is true:

$$d(i,j) = 1 - s(i,j) \qquad (4.3)$$

4.4. Common formula used for dissimilarity

To calculate the dissimilarity of objects we calculate the distance between every two objects. The distances must satisfy the following properties:

$d(i,j) \geq 0$ non-negativity

$d(i,i) = 0$

$d(i,j) = d(j,i)$ Symmetry

$d(i,j) \leq d(i,h) + d(h,j)$ Triangle rule

Commonly used formulas for distance are:

a. Euclidean distance:

$$d_E(x_i,x_j) = \sqrt{\sum_{k=1}^{p}(x_{ik} - x_{jk})^2} \qquad (4.4)$$

b. Manhattan distance

$$d_{Ma}(x_i,x_j) = \sum_{k=1}^{p}\left|x_{ik} - x_{jk}\right| \qquad (4.5)$$

c. Minkowski distance

$$d_{Mi}(x_i,x_j) = \sqrt[q]{\sum_{k=1}^{p}(x_{ik} - x_{jk})^q} \qquad (4.6)$$

d. Cosine distance

$$d_{\cos} = \frac{\sqrt{\sum_{k=1}^{p} x_{ik}^2} \cdot \sqrt{\sum_{k=1}^{p} x_{jk}^2}}{\sum_{k=1}^{p}(x_{ik} \cdot x_{jk})} \qquad (4.7)$$

e. Canberra distance

$$d_{CAN}(i,j) = \sum_{k=1}^{n} \frac{\left|x_{ik} - x_{jk}\right|}{\left|x_{ik}\right| + \left|x_{jk}\right|} \qquad (4.8)$$

5. A possible Taxonomy for clustering algorithms

Arranging data in different groups can be made using different strategies. Based on this, the clustering strategy can be divided into several categories: techniques based on data partitioning, clustering techniques based on hierarchical methods [5], methods based on attributes order, methods based on density, grid-based methods and methods based on models. [9].

5.1 Partitioning Methods

If *n* objects must be grouped into *k* groups, then a partitioning method constructs *k* partitions of the objects, each partition is represented by a cluster with $k \leq n$. The clusters are formed taking into account the optimization of a criterion function. This function expresses the dissimilarity between the objects, so that the objects that are grouped into a cluster are similar and objects from different clusters are dissimilar. For this type of grouping method the clusters must satisfy two conditions:

- Each cluster must contain at least one object;
- Each object must be included in a single cluster.

The basic idea of this type of methods is that the algorithm initially starts with a given number *k* groups representing the number of partitions (clusters) and then it applies a partitioning method that recalculates the *k* clusters. The method also uses an iterative relocation technique that attempts to improve the partitioning by moving objects from one group to another. The moving criterion needs to respect the condition that the objects of the same cluster are similar (close). There are different criteria to evaluate the quality of clusters. These will be presented in Section 6.

We can identify three different types of partitioning algorithms:

- K-Means clustering algorithms; for this type of algorithm each cluster is represented by the average value of all objects from the same cluster.
- K-Medoids clustering algorithms: for this type of algorithm each cluster is represented by the objects which are closest to the center (medoid) of the cluster.
- Probabilistic algorithms for clustering: a probabilistic method assumes that the data come from a mixture of populations whose distributions and probabilities must be determined.

These algorithms can be used in collections of small to medium data when the resulted clusters can be found in spherical forms.

5.1.1.K-Means algorithm

K-Means algorithm [10], [11], [18] is the most popular algorithm used in scientific and industrial applications. The name of the algorithm comes from the representation of *k*-clusters C_j whose weights (means) c_j are calculated as the centroid of the points which are grouped in cluster C_j. The value for the parameter *k* is set at the start of the algorithm and represent the number of cluster that want to be obtained. The similarity between the items from the same cluster C_j is calculated in relation to the centroid. The error criterion used is defined in the formula

$$E(C) = \sum_{j=1}^{k} \sum_{x_i \in C_j} \left\| x_i - c_j \right\|^2 \qquad (5.1)$$

Where *x* is the given entry object and c_j the centroid of C_j.

The scope is to achieve a greater intra-cluster similarity and very small inter-cluster similarity so that we get *k* clusters as compact and simultaneously separated as possible. It is obvious that this type of algorithm works with numerical variables. The algorithm performs the following steps:

1. Generate *k* random centers in *n*-dimensional space, which represents the initial centroids.
2. For each object compute the distance between it and all the centroids, then the object is assigned to the closest centroid for computing the distance there are several formulas presented in section 4.4.
3. Once all objects have been properly assigned to the centroids, the positions of the *k* centroids are recalculated as the center of all samples assigned to each centroid individually.
4. Repeat steps 2 and 3 until the centroids no longer change their positions.
5. The objects assigned to the centroids represent the contents of the final clusters.

5.1.2 K-Medoids method

K-medoids algorithm is an adaptation of k-Means algorithm. Instead of computing the centroid of each group, the algorithm chooses a representative element called medoid for each cluster at each iteration. The medoid for each group is calculated by finding an item in the cluster that minimizes the sum:

$$\sum_{j \in C_i} d(i,j) \qquad (5.2)$$

Where C_i is the cluster that contains the object *i* and *d(i,j)* the distance between object *i* and object *j*.

There are two advantages by using existing objects as cluster centers. First, a medoid can describe usefully a cluster. Second, it is not necessary to calculate the distance between the objects for each iteration of the algorithm. The distances can be computed once and then saved in a so called distances matrix.

The steps of the k-Medoids algorithm can be summarized as follows:

1. Choose k objects random to be the original medoids of the clusters.
2. Assign each object to the nearest cluster associated to the medoid.
3. Recompute the positions for the k-medoids.
4. Repeat steps 2 and 3 until the medoids doesn´t change.

Kaufman and Rousseeuw presented in [16] the PAM (Partition Around Medoids) algorithm which is an iteratively implementation of the k-Medoids algorithm.

The K-Medoids algorithm is more robust in terms of noise comparing to the k-Means algorithm. Since k-Medoids is working directly with medoids it is not so much influenced by the outliners like the centroid calculation influences the k-Means algorithm. However in terms of computational cost the k-medoids algorithm has a higher cost because for a single iteration the computational cost is $O(k(n-k)^2)$. The K-medoids algorithm is efficient on small data sets. On large data sets, due to its high computational cost, the algorithm becomes quite slow. For applying the PAM algorithm to large data sets Rousseew and Kaufman (1990) [16] developed the algorithm CLARA (Clustering Large Applications) who relies on the PAM algorithm which is applied only to a sample of data extracted from a given set. CLARA will discover in this reduced set the *k*-medoids. The complexity of each iteration is $O(kS^2+k(kn))$ where *S* is the data set size chosen (number of samples), *k* the number of clusters and *n* the total number of points. In [15, 22, 25] is proposed to choose the value $S = 40+2k$.

5.2 Hierarchical methods

These types of methods provide a hierarchical structure of the set of objects. There are two approaches of hierarchical methods:

Agglomerative -These methods have a "bottom-up" approach. At the beginning of the algorithm each object is a cluster, and then in the next steps the clusters are merged together based on similarity measures creating a hierarchical structure until all clusters are joined into a single cluster or until another stopping condition (given number of clusters, time, etc.) is reached.

Divisive - These methods have a "top-down" approach. First all objects are considered to be contained in a single cluster, then after successive iterations each cluster is divided into smaller clusters until each object is a cluster or until a stopping condition is reached.

A major problem of hierarchical clustering algorithms is that once the division or merging step has been made it cannot be canceled. This issue also represents a major advantage of this type of calculations due to reduction algorithms avoiding calculating the various combinations of possibilities.

AGGLOMERATIVE ALGORITHMS

Depending on the method of computing the similarity between the clusters in the hierarchical agglomerative approaches there can be distinguished more models of this category of clustering algorithms [19]:

Single Link

The similarity between two clusters is given by the minimum distance between the most similar two documents contained in those clusters.

$$sim(A,B) = \min_{a \in A, b \in B} sim(a,b) \qquad (5.3)$$

where *A* and *B* are the clusters to be merged and *a, b* are documents with $a \in A$ and $b \in B$.

Complete link

The similarity between two clusters is given by the similarity of the most dissimilar documents from the two clusters. This is equivalent to choosing pairs of clusters for merging with the smallest diameter.

$$sim(A,B) = \max_{a \in A, b \in B} sim(a,b) \qquad (5.4)$$

Average link

The similarity between two clusters is calculated as the average sums of distances between each element from the first cluster and all elements from the second cluster.

$$sim(A,B) = \frac{1}{|A||B|}\sum_{a\in A, b\in B} sim(a,b) \qquad (5.5)$$

Centroid Link

The similarity between two clusters is calculated as the distance between the cluster centroids.

$$sim(A,B) = \|C_A - C_B\| \qquad (5.6)$$

Ward's method

Ward's method [31] attempts to minimize the Sum of Squares (SS) of any two (hypothetical) clusters that can be formed at each step:

$$\Delta(A,B) = \sum_{i\in A\cup B}\|x_i - m_{A\cup B}\|^2 - \sum_{i\in A}\|x_i - m_A\|^2 - \sum_{i\in B}\|x_i - m_B\|^2 = \frac{n_a \cdot n_B}{n_a + n_b}\|m_A - m_B\|^2 \qquad (5.7)$$

Where m_j is the centroid of cluster j and n_j is the number of objects in cluster j.

If the two pairs of cluster centers will have identical distances using the Ward's method then the smaller clusters will be merged.

DIVISIVE ALGORITHMS

The divisive hierarchical clustering approach is conceptually more complex than the agglomerative. If the first step of an agglomerative approach can have $\frac{n\cdot(n-1)}{2}$ possibilities to merge two objects then in the divisive approach there are $2^{n-1}-1$ possibilities to divide data into two groups. This number is considerably higher than that in case of agglomerative methods.

Kaufmann and Rousseeuw have presented in [15] a divisive clustering algorithm DIANA (Divisive Analysis).

Algorithms like BIRCH (Balanced Iterative Reducing and Clustering) proposed in [28] and CURE (Clustering Using Representatives) proposed in [8] combines the partitioning methods with the hierarchical ones.

5.3 Method based on the order of words

Suffix Tree Clustering (STC)

The algorithm presented in [27] does not require a vector representation of objects (documents) like the algorithms presented in the previous sections. The STC algorithm uses the Suffix Tree Document Model (STDM) to represent the documents. This algorithm will take into account the words order, and creates clusters based on words or groups of consecutive words from the documents. Thus can be a major advantage because it takes account of word order in sentences.

A suffix tree of a document *d* is a compact tree containing all suffixes *s* of that document. In our case a suffix is a string consisting of one or more words.

Rules for building the suffix tree:

1. Each internal node other than root must have at least two children and each edge leaving a node is labeled with a nonempty substring *n*.
2. Any two edges that start from the same node cannot start with the same word.

Suffix tree construction

Let $d = w_1w_2w_3 \ldots w_m$ be the representation of a document as a sequence of words in same order from as in the document with $i = \overline{1, m}$ and S a set of n documents. The suffix tree for a set S containing n strings, each of them having the length m_n is a tree that contains exactly one root node and Σm_n leaf nodes.

Since each node contains at least two children it will contain the common part of at least two suffixes. Each leaf of the tree can represent one or more documents and all edges from the root to the leaf nodes represent an entire document or a substring from a document. Two documents with many common nodes tend to be similar.

Selecting the base nodes

After creating the suffix tree some nodes will become clusters. The basic rules for a node to become a cluster are:

a. Each node in the suffix tree that has at least two children is the considered a basic node and receives a score *S(B)* according to equation (5.8)
b. After ordering the nodes in descending order based on the obtained score the algorithm retains the first *k* nodes (a given threshold). These basic nodes will continue to be used in the next step.
c. Nodes that receive a score lower than predetermined threshold are eliminated.

Formulas used:

Let $|B|$ be the number of documents contained in cluster B and $|P|$ the number of words in the sentence P.

A cluster score is calculated as

$$S(B) = |B| \cdot f(|P|) \qquad (5.8)$$

where the weighting function for the length of a sentence is

$$f(|P|) = \begin{cases} 0.5, & |P| = 1 \\ |P|, & 2 \le |P| \le 6 \\ 6, & |P| > 6 \end{cases} \qquad (5.9)$$

This function penalizes the sentences which contain a single word, is linearly increasing for sentences containing two to six words and become constant for sentences longer than six words.

The similarity between two clusters is calculated:

$$SIM(B_i,B_j) = \begin{cases} 1, & \left(\frac{|B_i \cap B_j|}{|B_i|} > \alpha\right) \wedge \left(\frac{|B_i \cap B_j|}{|B_j|} > \alpha\right) \\ 0, & otherwise \end{cases} \qquad (5.10)$$

where α is a chosen threshold.

In [27] threshold for α is 0.5 and in [26] it is 0.8. Regarding the formula for calculating the similarity we can choose to use the Jaccard's coefficient, which gives us the degree of similarity between two nodes. The higher the value the greater is the similarity between nodes (the value is in the interval [0, 1] - with 1 for identical nodes):

$$J\left(B_i,B_j\right)=\frac{B_i \cap B_j}{B_i \cup B_j} \tag{5.11}$$

Combining the base nodes

a. The similarity between two base nodes is calculated according to formula (5.10).

b. If the similarity between two nodes exceeds a certain threshold then they will merge and after merging the resulted node will become a cluster.

5.4 Density-based methods

A topological space can be decomposed into its connected components. Starting from this point of view we can define a cluster as a related component that is constructed in the direction where the density is higher. This is the reason why the density based algorithms are able to discover clusters of arbitrary shapes. Also constructing towards higher density the algorithm protects itself against noise (outliner). However, these methods have the disadvantage that if a cluster consists of two adjacent areas of different density but larger than a given threshold the result is not very informative [9].

Important algorithms of this method include: DBSCAN (Density-Based Spatial Clustering with Noise Algorithm) [2], which produces clusters with a threshold density. OPTICS (Ordering Points to Identify the Clustering Structure) [2] extends the DBSCAN algorithm and computes the distances ε_i less given threshold ε with $0 \leq \varepsilon i \leq \varepsilon$. The only difference from the OPTICS algorithm DBSCAN is that the objects are not assigned to a cluster but the algorithm saves the order in which objects were processed.

5.5 Grid- based methods

Grid-based methods are methods that quantify the representation of the object space into a finite number of cells that form a grid structure. Once this representation is made a clustering algorithm is applied. The great advantage is that the processing time depends only on the number of cells in each dimension of the transformed space and not by the number of objects contained in the cells. Typical algorithms for this type of method are STING (Statistical Information Grid) [23], DENCLUE (DENsity-based CLUstEring), Clique, MAFIA (MAximal Frequent Itemset Algorithm)

5.6 Model - based methods

Methods based on models are assuming that there is a model for each cluster and they try to find data that fit best with the model. A model-based algorithm can discover clusters by constructing a density function that reflects the spatial distribution of the data. However these algorithms can lead to an automatic discovery of clusters, when the numbers of clusters are based on standard statistical methods taking into account the noise. Algorithms that fall into this category are: AutoClass (Automatic Classification), COWEB, Class. Also in this category are used as a model and algorithms such as SOM neural networks.

It may also be included in this category, algorithms that rely on ants' behavior and particle swarms.

6. Cluster validation

Each clustering algorithm applied on the same set of data will group the data in different ways depending on the similarity metric used. This makes analysis for the efficiency of clustering algorithms very difficult.

To evaluate the performance or quality of clustering algorithms, objective measures must be established. There are three types of quality measures:

a. external, when there is an a priori knowledge about the clusters (we have pre-labeled data);
b. internal, which have no information about clusters;
c. relative assessing group differences between different solutions.

External measures are applied to both classification and clustering algorithms, while internal and relative measures are applied only to the clustering algorithms.

EXTERNAL VALIDATION MEASURES

External validation measures require pre-labeled data sets for clustering analysis. Because clustering data can be done from many points of view and the labels may vary from these points of view, the comparison is made to the group containing the most data in a specific category.

Some of the external evaluation measures:

Precision is the percentage of retrieved documents which are really relevant to that category. The value for the precision is in the interval [0,1], with 1 as best. For a cluster C_i and a known class S_j we compute the precision:

$$precision(C_i,S_j)=\frac{\left|C_i \cap S_j\right|}{\left|C_i\right|} \tag{6.1}$$

Recall is the percentage of documents that are relevant to that category and are indeed grouped into that category. The value for the recall is in the interval [0,1], with 1 as best.

$$recall(C_i,S_j)==\frac{\left|C_i \cap S_j\right|}{\left|S_i\right|} \tag{6.2}$$

Accuracy is the percentage of documents that are correctly grouped into categories depending on the labels of documents (needs labeled documents).

Fmeasure is a measure that combines precision and recall site and is calculated according to formula (6.3)

$$Fmeasure(C_i,S_j)=\frac{2\cdot precizia(C_i,S_j)\cdot recall(C_i,S_j)}{precizia(C_i,S_j)+recall(C_i,S_j)} \tag{6.3}$$

For each class it will select only the cluster with the highest Fmeasure. In the final Fmeasure for overall measure of a clustering solution is weighted by the size of each cluster.

INTERNAL VALIDATION MEASURES

In [31] are presented four such metrics:

Compactness–This measure expresses how similar are the data from a given cluster

$$compactness(C) = \sum_{i=1}^{n_C} (\vec{c} - \vec{x}_i)^2 \qquad (6.4)$$

Where C is the current cluster, n_c the number of elements in the cluster, c the cluster centroid and x_i is an element of the cluster. In other words, these metric measures how "close" the documents within a cluster are. The value is in the interval $[0, \infty)$ and as the lower the better is the measure

Separabillity–This measure between the clusters expresses how dissimilar the clusters are.

$$separability = \sum_{i,j=1}^{C} \min(dist(\vec{c}_i, \vec{c}_j)) \qquad (6.5)$$

Thus for each cluster, the nearest cluster is determined. Based on this metric we seek clusters that maximize this value, so that the clusters are very dissimilar.

Balance - The balance is computed as the clusters are formed and expresses how “well-balanced” the formed clusters are.

$$balance = \frac{n / k}{\max_{i=\overline{1,k}}(n_i)} \qquad (6.6)$$

Where n is the total number of documents, n_i the number of documents in the cluster I (the formula takes into consideration the biggest formed cluster), k is the number of formed clusters.

This measure can take values in the interval [0,1], the maximum value 1 is achieved when all clusters have the same number of documents. A value close to 0 is obtained when the number of documents contained in clusters varies greatly.

7. Conclusions

Clustering in text documents is an unsupervised learning method of classifying documents with a certain degree of similarity using different metrics based on distance. Basically, the clustering algorithm must identify (must find) the clusters in which documents are grouped and/or some patterns (rules) that separate one group from another group. There is no predefined taxonomy. It is established when the clustering algorithm run to the set of documents.

Cluster evaluation is an next important task. The formulas presented for internal and external evaluation of clusters can give a measure for evaluate the quality of clustering process. Still it is very hard to compare different clustering results.

8. Acknowledgment

This work was partially supported by CNCSIS-UEFISCSU, project number PN II-RU code PD_670/2010.

9. References

[1] Abraham, A., Ramos, V. - *"Web Usage Mining Using Artificial Ant Colony Clustering and Genetic Programming"* Proc. of the Congress on Evolutionary Computation (CEC 2003), Canberra, pp. 1384-1391, IEEE Press. 2003

[2] Ankerst, M., Breuning, M., Kriegel, H.-P., Sander, J.,- Oprics: "Ordering points to identify the clustering structure" In. Proc. 1999 ACM-SIGMOD, Int. Conf. Management of Data, Philadelphia, 1999

[3] Batet, M., Valls, A., Gibert, K. - *"Improving classical clustering with ontologies"*, In Proc. IASC 2008, Yokohama, 2008

[4] Bennett, K. P., Demiriz A., - *"Semi-supervised support vector machines."* In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, Advances in Neural Information Processing Systems, pages 368-374, Cambridge, MA, 1998. MIT Press.

[5] Berkhin, P., - *"A Survey of Clustering Data Mining Techniques"*, Kogan, Jacob;Nicholas, Charles; Teboulle, Marc (Eds.) Grouping Multidimensional Data, Springer Press, pp. 25-72 (2006)

[6] Burges, C. J. C., - "*A tutorial on support vector machines for pattern recognition.*" Data Mining and Knowledge Discovery, 2(2):121-167, 1998.

[7] Dorigo, M., Bonabeau, E., Theraulaz, G., - *"Ant Algorithms and stigmergy"* Future Generation Computer Systems Vol.16, 2000.

[8] Guha, S. Rastogi, R., Shim, K., - *"CURE: an efficient clustering algorithm for large databases."* in Proceedings of the 1998 ACM SIGMOD international conference on Management of data (SIGMOD '98), AshutoshTiwary and Michael Franklin (Eds.). ACM, New York, NY, USA1998

[9] Han, J., Kamber, M., - *"Data Mining: Concepts and Techniques"*, Morgan Kaufmann Publishers, 2001

[10] Hartigan, J. A., Wong, M., Algorithm as 136: A k-means clustering algorithm, Journal of the Royal Statistical Society. Series C (Applied Statistics), London, 1979

[11] Hartigan, J. A., - *"Clustering Algorithms"*, New York: John Wiley & Sons, Inc, 1975

[12] Hotho, A., Staab, S. Stumme,G., - *"Ontologies Improve Text Document Clustering"*, IEEE International Conference on, p. 541, Third IEEE International Conference on Data Mining (ICDM'03), 2003

Implementation, Morgan Kaufmann Press, 2000

[13] Jain, A., K., Dubes, R.,C. - *"Algorithms for Clustering Data"*, Prentice Hall, Englewood Cliffs, NJ. 1988

[14] Jain, A. Murty, M. N., Flynn, P. J., - *"Data Clustering: A Review"*, ACM Computing Surveys, Vol. 31(3), pp. 264-323 (1999)

[15] Kaufman, L. and Rousseeuw, P.J. - *"Finding Groups in Data: An Introduction to Cluster Analysis",* Wiley-Interscience, New York (Series in Applied Probability and Statistics), 1990

[16] Kaufman, L., Rousseeuw, P. J., - *"Clustering by means of medoids, in Statistical Data Analysis based on the L, Norm"*, edited by Y. Dodge, Elsevier/North-Holland, Amsterdam, 1987

[17] Labroche, N., Monmarché, N., Venturini G., - *"AntClust: Ant Clustering and Web Usage Mining, In Genetic and Evolutionary Computation"*, GECCO 2003 Lecture Notes in Computer Science, Volume 2723/2003, 201, 2003.

[18] MacQueen, J. B., - *"Some Methods for classification and Analysis of Multivariate Observations"*, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1:281-297, 1967

[19] Manning, C., - *"An Introduction to Information Retrieval"*, Cambridge University Press, 2009

[20] Meyer,S., Stein, B., Potthast, M., - *"The Suffix Tree Document Model Revisited"*, Proceedings of the I-KNOW 05, 5th International Conference on Knowlegdge Management, Journal of Universal Computer Science, pp.596-603, Graz, 2005

[21] van der Merwe, D.W., Engelbrecht, A.P., - *"Data clustering using particle swarm optimization, in: The 2003 Congress on Evolutionary Computation"*,. CEC '03, Canberra, 2003

[22] Ng, R., Han, J. - *"Efficient and Effective Clustering Methods for Spatial Data Mining"*, Proceedings of International Conference on Very Large Data Bases,Santiago, Chile, Sept. 1994

[23] Wang, W., Yang, J., Muntz, R., - *"STING: A statistical information grid approach to spatial data mining",* In Proc. 1997 Int. Conf. Very Large Data Bases, Athens, 1997

[24] Ward, J. H., - *"Hierachical grouping to optimize an objective function"*, J. Am. Statist. Assoc. 58, 236-244, 1963

[25] Wei, C. P., Lee, Y. H., Hsu, C. M., - *"Empirical Comparison of Fast Clustering In Algorithms for Large Data Sets"*, In Experts Systems with Applications vol. 24, pp. 351– 363 - 2003

[26] Wen, H., - *"Web SnippetsClustering Based on an Improved Suffix Tree Algorithm"*, Proceedings of FSKD 2009, Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, 14-16 August 2009

[27] Zamir, O, Etzoni, O., - *"Web Document Clustering: A Feasibility Demonstration"*, Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, 1998

[28] Zhang, T. Ramakrishnan, R. Livny, M., - *"BRICH: an efficent data clustering method for very large databases"*, in Proceedings 1996 ACM-SIGMOD Int. Conf. Management of Data, Montreal, 1996

[29] Zhang, B., - "Generalized k-harmonic *Means – Dynamic Weighting of Data in Unsupervised Learning"*, In Proceedings of the 1st SIAM ICDM, Chicago, 2001

[30] Zhao, Y. Karypis, G., - *"Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering"*, Machine Learning, Vol. 55. No. 3, 2004

[31] Metrics for evaluating clustering algorithms- http://www.scribd.com/Clustering/d/28924807- accesat 25.08.2010