Lecturer Alexandra CERNIAN, PhD
E-mail: Alexandra.cernian@aii.pub.ro
Professor Dorin CARSTOIU, PhD
E-mail: dorin.carstoiu@aii.pub.ro
Lecturer Adriana OLTEANU, PhD
E-mail: Adriana.olteanu@aii.pub.ro
Professor Valentin SGARCIU, PhD
E-mail: vsgarciu@aii.pub.ro
"Politehnica" University of Bucharest

ASSESSING THE PERFORMANCE OF COMPRESSION BASED CLUSTERING FOR TEXT MINING

Abstract. *The nature of the human brain is to find patterns in whatever surrounds us. Thus, we are all developing models of our personal universe. In an extended form, a constant preoccupation of philosophers has been to model the universe. Clustering is one of the most useful tools in the data mining process for discovering groups and identifying patterns in the underlying data. This paper addresses the compression based clustering approach and focuses on validating this method in the context of text mining. The idea is supported by the evidence that compression algorithms provide a good evaluation of the informational content. In this context, we developed an integrated clustering platform, called EasyClustering, which incorporates 3 compressors, 4 distance metrics and 3 clustering algorithms. The experimental validation presented in this paper focuses on clustering text documents based on informational content.*

Keywords*: clustering, compression, text mining, EasyClustering, FScore.*

JEL Classification: O30

1. Introduction

Clustering is one of the most powerful tools used for discovering groups and identifying patterns in the underlying data. Clustering methods aim to discover similarity between elements and group the most similar elements together. Clustering is unsupervised process (Marmanis and Babenko, 2009), since there is no predefined structure or pattern of the data. Clustering is applicable in many domains, ranging from biology and medicine to finance and marketing. It is used infields such as data mining, pattern recognition, information retrieval, image analysis, market analysis, statistical data analysis and so on.

The origin of this work is the clustering by compression technique (Cilibrasi and Vitanyi, 2005). The compression based clustering approach leans on the following concepts: the Kolmogorov complexity (Li and Vitanyi, 2008), the normalized information distance (Vitanyi et. al., 2009) and the approximation of the amount of information provided by compressors. The technique uses a universal distance metric called the Normalized Compression Distance (NCD). Therefore, we consider it is worth investigating the benefits that compression algorithms could bring to the overall clustering domain. For instance, what if the performance of combining a distance metric such as Jaro or Levenstein with a clustering algorithm like UPGMA could be significantly improved by compressing the input data?

In order to evaluate the performance of compression based clustering, we developed a clustering platform, called EasyClustering. The clustering platform integrates 3 compression algorithms (ZIP, bzip2 and GZIP), 4 distance metrics (NCD, Jaro, Jaccard and Levenstein) and 3 clustering algorithms (UPGMA, MQTC and k-means). The EasyClustering platform aims at facilitating a comparative analysis of the clustering results produced by various combinations of compression algorithms, distance metrics and clustering algorithms. Through this comparative evaluation, the user can objectively evaluate the benefits of the compression based clustering approach. For the particular work discussed in this paper, we focus on the performance of the compression based clustering approach in the context of text mining, based on semantic criteria.

The rest of the paper is structured as follows: Section 2 presents the theoretical background and some related work, Section 3 describes the EasyClustering platform, Section 4 presents experimental results for validating the capabilities of compression based clustering in the context of text mining, and Section 5 draws the conclusions for this work.

2. Background and related work

When talking about data clustering, there are a few basic concepts which need to be discussed, such as distance metric, similarity matrix and clustering algorithms. Conventional clustering methods mainly consist of two parts: the construction of a similarity matrix between documents and the construction of clusters using a clustering algorithm.

A distance metric (Li et. al., 2003) is defined as a function which establishes the distances between the elements of a data set X. Once a distance metric has been chosen for measuring the distances between the elements of a dataset, the similarity or distance matrix is computed, containing the distances among the n objects, taken two by two (Grünwald and Vitanyi, 2004). It is a symmetric $n \times n$ matrix containing positive real numbers, normalized between 0 and 1.

2.1 Clustering Methods

The purpose of clustering methods is to group similar elements together. The similarity is established through specific distance metrics, based on which similarity or distance matrix are computer (Aggarwal, 2013). Afterward, clustering algorithms interpret the matrix and create clusters. There are three main clustering methods categories: partitional methods, hierarchical methods and quartet methods.

2.1.1 Hierarchical clustering

Hierarchical clustering algorithms produce a hierarchy of nested groups. Each step involves the assignment of an element to a specific cluster. The most well-known hierarchical clustering algorithm is UPGMA (**U**nweighted**P**air **G**roup **M**ethod with **A**rithmetic Mean), which produces a *dendrogram*as output (Milligan, 1996).

The main steps of an agglomerative hierarchical clustering algorithm for clustering a set *X* of *n* objects are the following:

Step 0: Each object forms its own cluster.
Step 1: The distance matrix in computed. Using a distance metric *d*.
Step 2:A new element is formed by merging the two closest elements. Now, we have *n-1* clusters.
Step 3: Now, we will want to merge the two closest clusters using one of the linkage methods described above.
Step 4: At step *t,* we have *n - (t - 1)* clusters, and we want to join the closest clusters as describes in Step 3. The process is repeated until all the objects belong to a single cluster.

2.2.2 Partitional clustering

Partitional algorithms create a fixed, predefined number of clusters (Jain and Dubes, 1998). They attempt to directly decompose the data set into disjoint clusters. These techniques start with a randomly chosen or user-defined clustering, then optimize the clustering according to some validity measurement. Perhaps the most famous partitional algorithm is K-MEANS(Hartigan and Wong, 1979).

Let *X*be a group of *n* objects to cluster. Assume that *d* is a distance metric for and we want to generate *k* clusters of objects.
Step 1: We randomly determine *k* initial centres for the clusters, called *centroids*.
Step 2: For each object, determine its distance to the centroids.
Step 3: Group the objects based on minimum distance to a centroid.
Stop: The process is stopped when two consecutive steps produce the same division of the objects.

2.2.3 Quartet clustering

The quartet method of clustering was introduced by Cilibrasi and Vitanyiin (Cilibrasi and Vitanyi, 2005). The output of the quartet method is called *quartet topology* (Cilibrasi and Vitanyi, 2005). The authors propose a quartet clustering method called *Minimum Quartet Tree Cost (MQTC),* described in what follows (Cilibrasiand Vitanyi, 2005):

Step 1: A quartet tree with *2n -2* nodes is randomly created.
Step 2: A new quartet tree is created through simple mutations.
Step 3: The score of the new tree is calculated.
Step 4: If the score of the new tree is greater than the score of the original tree, then we replace the original tree with the new tree.
Stop: If after a long time no change of the cost occurs, then the algorithm stops.

2.2 Clustering by Compression

In 2005, Cilibrasi and Vitanyi proposed a new method for clustering based on compression algorithms (Cilibrasi and Vitanyi, 2005). The method works in 2 steps, as follows:
Step 1: It uses the universal metric NCD to compute the similarity matrix
Step 2: It uses a clustering method to interpret the distance matrix.

Then the NCD is defined as follows (Cilibrasi and Vitanyi, 2005):

$$\mathrm{NCD}(x,y) = \frac{C(xy) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \qquad (1)$$

Where: *C* is a compressor, *x, y* are two files, *xy* is teh file obtained by concatenating x and y, *C(x*), *C(y)* and *C(xy)* are the sizes of the compressed files.

2.3 Clustering test platforms

In order to evaluate the solution proposed in this paper, it is important to be objectively informed with regard to other similar approaches. Thus, this section provides an overview of clustering platforms we have found during our research.

- **Cluster 3.0**. Cluster 3.0 (de Hoon et. al., 2004) is an open source clustering software dedicated to genomic datasets analysis, such as DNA microarrays. It implements hierarchical clustering, the k-means clustering algorithm, k-medians clustering and 2D self-organizing maps.
- **CompLearn**. CompLearn Toolkit (Cilibrasi, 2003) was especially developed for the validation of the clustering by compression technique and for the quartet method proposed by Cilibrasi and Vitanyi in (Cilibrasiand Vitanyi, 2005). CompLearn uses the NCD as distance metric,

several compressors (gzip, bzip2, PPMZ), and the MQTC quartet method for constructing trees.

- **RapidMiner and Weka**. RapidMiner (RapidMiner, 2011) and Weka (Weka, 2011) are 2 data mining platforms which also integrate some clustering components, but they require an experimented user to use them.
- **ClusTIO**. ClusTIO (ClusTIO, 2009) is a java command-line tool implementing several distance metrics (among which also the NCD), several compressors (ZIP, BZIP2, arithmetic compressor) and several clustering algorithms (UPGMA, MTCQP, MQTC). MTCQP is a quartet-based algorithm proposed by the author of ClusTIO. This platform takes a distance matrix in PHYLIP format as input and outputs a tree in Newick format (ClusTIO, 2009).
- **CVAP**. CVAP (Wang, 2009) is a MATLAB cluster validity and analysis tool. It includes 18 validity indices and 5 clustering algorithms. CVAP uses the Euclidean distance and Pearson correlation for calculating the distance matrix.

Section 3 presents the design, UML modelling and implementation of the EasyClustering platform.

3. The EasyClustering test platform – an integrated clustering system

In order to objectively assess the performance of the compression based clustering method, we have developed a test platform called EasyClustering(Cernian, 2011). The design of the platform is presented in what follows.

3.1 The architecture of the clustering system

Figure 1illustrates the architecture of the clustering platform. The workflowfor using the EasyClustering platform is the following:

- Select the dataset to be clustered and upload it onto the platform.
- For text documents, pre-processing techniques are available and they include stemming and eliminating stopwords.
- The Compression Engine compresses the files (for the NCD distance metric).
- The Distance Matrix Engine computes the similarity matrix according to the distance metric chosen by the user.
- The Clustering Engine generates the clusters using one of the integrated algorithms.
- The Output Engine displays the clusters received from the Clustering Engine in a graphical format. Moreover, the clustering solution is also saved in a predefined XML format and in text format.

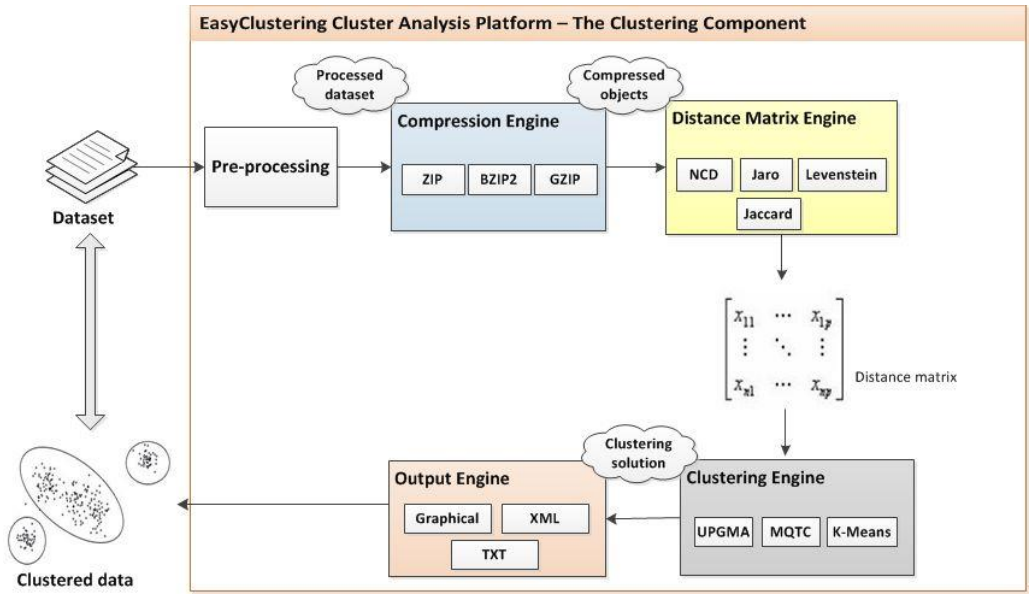

Figure 1. The architecture of the EasyClustering clustering platform

The Compression Engine integrates the following 3 compressors: ZIP, BZIP2 and GZIP.

The Distance Matrix Engine contains the following distance metrics: NCD, Jaro, Jaccard and Levenstein. The NCD is discussed in Section 2.2. The Jaro distance (Jaro, 1989) measures the similarity between two strings. The higher the Jaro distance is, the more similar two strings are. The Levenstein distance (Levenstein, 1966)determines the number of differences between two entries. The Jaccard distance (Jaccard Similarity,2014) is a measure of the dissimilarity between the datapoints in a dataset. These metrics have been chosen because they proved to work best under universality conditions.

The Clustering Engine contains 3 clustering algorithms: K-Means, UPGMA and MQTC, which have been discussed in Section 2.1 of this paper.

3.2 The UML model of the clustering platform

Figure 2 presents the use UML case diagram (Booch et. al., 2010) for the EasyClustering test platform.

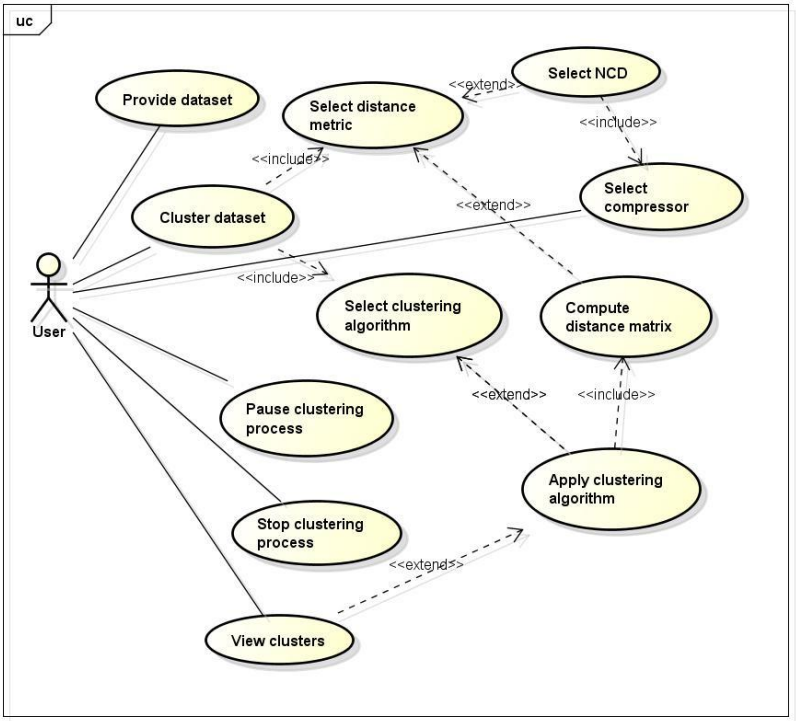

Figure 2. The UML use case diagram the EasyClusteringplatform

1. **The user provides the dataset to be clustered**
    a. The user starts the application.
    b. The user presses the Load Dataset button and a new dialog box appears. There are 2 options:
        i. The user can choose a folder and the application will automatically select all the files inside that folder;
        ii. The user can provide as input a text file containing the paths to the documents to be clustered.
    c. The application return to the main window.

2. **The application performs the clustering of the dataset**

Precondition: The dataset must have been loaded.

    a. The user selects the distance metric to be used for computing the distance matrix.
        i. If the user chooses the NCD distance metric, the user must select the compressor that will be used from the Compressors region.

ii. Otherwise, the compression of the elements to be clustered in optional.

b. The user must select the clustering algorithm that will be used for interpreting the distance matrix.
c. The user presses the Cluster button in order to start the clustering process.
d. The application computes the distance matrix based on the distance metric specified.
    i. If the application is using the NCD to compute the distance matrix, the compressed versions of the documents will be generated, using the compressor previously specified by the user.
e. The distance matrix is displayed in a pop-up window and can be saved as a text file.
f. The application uses the clustering algorithm specified by the user on the distance matrix in order to generate the corresponding clusters.
g. The clusters generated by the clustering algorithm are displayed in graphical format.
h. The clusters generated by the clustering algorithm are exported in predefined XML format and text format.

3. The user can view the result of a clustering process.

Precondition: The clustering process must have been performed with no errors.

a. The user can view the clusters generated by the clustering algorithm in a graphical format.
b. The user can save the hierarchy of clustered documents as a text file.

4. The user can pause the clustering process.

*Precondition***:** The clustering process must be in progress.

a. The user can press the Pause button in order to temporarily stop the clustering process.
b. When the user presses the Cluster button, the process will be restarted from the point where it was when it was interrupted.

5. The user can stop the clustering process.

*Precondition***:** The clustering process must be in progress.

The user can press the Stop button in order to stop the clustering process. Any information will be lost regarding the current clustering.

The EasyClustering platform covers the clustering aspects, providing the user with a visual representation of clusters. After a complete series of tests on a particular dataset, the user can draw an objective conclusion regarding which distance metric or clustering algorithm performed better, as well as an evaluation of the results with and without compressing the dataset before starting the clustering process.

4. Compression based clustering in the context of text mining

The purpose of text mining is to process unstructured textual information in order to extract meaningful information. Text mining usually involves the following steps: structuring the input text (data cleaning), identifying patterns within the structured data, and the interpretation of the output (Berry, 2010). Text mining has gained popularity in a wide range of applications, such as: Web mining, marketing applications, text processing, opinion mining, and sentiment analysis. In this context, text clustering is one of the most frequently used tasks.

For testing the performance of the compression based clustering approach for text clustering, we used a selection of 50 scientific paper abstracts from IEEExplore (IEEExplore, 2014). The 50 elements of the dataset belong to six categories. The accuracy of the clustering solutions has been evaluated with the FScore measure (Aggarwal, 2013).

4.1Clustering the unprocessed text files

The first round of experiments concerning the clustering of text files focused on using the unprocessed dataset. Table 1 presents the top 5 results obtained for the classification of the text files.

Table 1. Top 5 results for clustering text files

	BZI2+ UPGMA + NCD	**ZIP/GZIP+ UPGMA+ NCD**	**BZIP2+ MQTC+ NCD**	**GZIP/ZIP+ MQTC+ NCD**	**BZIP2+ KMEANS+ NCD**
Fscore	0.97	0.88	0.87	0.81	0.80

These top 5 results prove that high quality results were produced by the EasyClustering platform for clustering scientific abstracts. An FScore value of 0.97 indicates an almost perfect clustering solution. However, besides this clustering solution for which we have obtained a very high FScore value, the other 4 FScores show a decrease of the quality of the clustering solutions obtained, which seems to be stabilized around the average value of 0.83.

In what follows, we will analyse the results obtained from several perspectives: from the compression algorithms perspective, from the distance metrics perspective and from the clustering algorithms perspective.

The no compression approach produced only 2 interesting results for this dataset: an FScore of 0.69 produced by the combination of Jaro and UPGMA and an FScore of 0.62 produced by the combination of Levenshtein and K-Means. Using the uncompressed dataset with the Jaccard distance metric lead to poor results, with FScore values around 0.40.Compressing the files before using the Jaro, Jaccard or Levenshtein distance metrics did not improve the results.

From the compression algorithms point of view, the best results were produced by BZIP2, followed by GZIP and then by ZIP.

As shown in Table 1, the best results were produced by the NCD distance metric. For the NCD, the FScore values ranged between 0.51 and 0.97, the lowest value being produced by the association between ZIP and GZIP compressors, K-Means and NCD. However, when using the association between BZIP2, K-Means and NCD, the FScorehad a value of 0.80.

The Levenshtein distance metric produced average results when the distance matrix was interpreted with the K-Means clustering algorithm. When using the UPGMA and MQTC clustering algorithms, there were no clearly defined clusters. Thus, the results were rather uncertain and an appreciation of the quality would not be extremely relevant. The best FScore value for the Levenshtein distance metric was obtained when the dataset was not compressed and the K-Means algorithm was used to produce the clustering solution: 0.62. Similar values, 0.57, were obtained in the same conditions, when the dataset was compressed with the BZIP2 and GZIP algorithms.

The Jaccard distance metric did not produce good results for this dataset. The average FScore was 0.30 and the visual examination of the solutions obtained confirm that the clusters are not correctly formed.

The Jaro distance metric produced its best results when the input dataset was not compressed, with a top FScore value of 0.69 (no compression + UPGMA), which denotes a rather good quality of the clustering solution. The same FScore value was obtained for BZIP2 + UPGMA + Jaro. A significant drop in interpreting the distance matrix produced by Jaro without compressing the files is noticed for the K-Means algorithm; in this case, the FScore is 0.28. The MQTC algorithm did not produce good results for interpreting distance matrices computed with the Jaro distance metric.

As a general statistics for this dataset from the clustering algorithms perspective, the best results were produced by the UPGMA algorithm, which lead to a value of 0.97 and 2 values of 0.88 for the FScore. The MQTC clustering algorithm also produced high quality results, with the highest FScores between 0.81 and 0.87. As in the previous cases, K-Means generated solutions of a significantly lower quality, with a top FScore value of 0.80.

4.2Clustering text files based on increased weight of keywords

The second round of experiments for text clustering was based on the following approach:

- Within each text abstract, we identified a set of 3 keywords;
- We increased the weight of the keywords, 10 times for each keyword.

In this way, meaningful information should be easier identified and the clustering solutions should be more accurate.

Table 2 presents the top 5 results obtained for this set of experiments.

Table 2.Top 5 results for clustering text files with increased keywords' weight

	BZI2+ UPGMA + NCD	**BZIP2+ MQTC+ NCD**	**NULL+ K-Means+ Levenshtein**	**NULL+ UPGMA+ Levenshtein**	**GZIP+ MQTC/UPGMA+ NCD**
Fscore	1	1	1	1	0.94

The results depicted in Table 2 prove that the quality of the text clustering was significantly improved by identifying the keywords and increasing their weight. We obtained 4 solutions which were a perfect match to the reference clustering, and a large number of solutions with FScore values greater than 0.80. For this dataset, both NCD and Levenshtein distance metrics produced very good results.

In what follows, we will analyse the results obtained from several perspectives: from the compression algorithms perspective, from the distance metrics perspective and from the clustering algorithms perspective.

The no compression approach produced very good results for the Levenshtein distance metrics, with 2 FScore values of 1 (K-Means and UPGMA). The solution obtained with the MQTC clustering algorithm is also very good, having an FScore of 0.93. Compressing the input dataset before computing the distance matrix with Levenshtein led to a significant decrease of the FScores, which dropped to an average value of 0.48.

Judging by the best results obtained, from the compression algorithms point of view, the best results were produced by BZIP2, followed by GZIP and then by ZIP.

As shown in Table , the best results were produced by the NCD distance metric. For the NCD, the FScore values ranged between 0.52 and 1, the lowest value being produced by the association between ZIP and GZIP compressors, K-Means and NCD. When using the association between BZIP2, K-Means and NCD, the FScore attained a value of 0.72, which is lower than the score obtained in

section 4.1 for the same combination of algorithms (0.80). However, the rest of the FScore values are better than those produced by clustering the unprocessed dataset.

The Levenshtein distance metric also produced high quality results when the dataset was not compressed. The second clustering solution had an FScore of 1 (perfect match) and thethird had an FScore of 0.93. When the dataset was compressed, the results lost their accuracy and the FScore values dropped significantly.

The Jaccard distance metric did not produce good results for this dataset. The average FScore was 0.30 and the visual examination of the solutions obtained confirm that the clusters are not correctly formed.

The Jaro distance metric produced rather confusing results. The best FScore it produced was 0.69 (BZIP2 + UPGMA + Jaro), which indicates an average clustering solution. In many of the solutions generated with Jaro, clusters were very mixed up or difficult to identify.

As a general statistics for this dataset from the clustering algorithms perspective, the best results were produced by the UPGMA algorithm, followed by MQTC and, finally, by K-Means. However, we must notice that K-Means managed to interpret very well the distance matrix produced by the Levenshtein distance matrix, when the dataset was not compressed. In this case, the FScore was 1.

5. Conclusion

Figure 3 presents a comparison of the best 10 FScore values obtained for text clustering during the validation process.

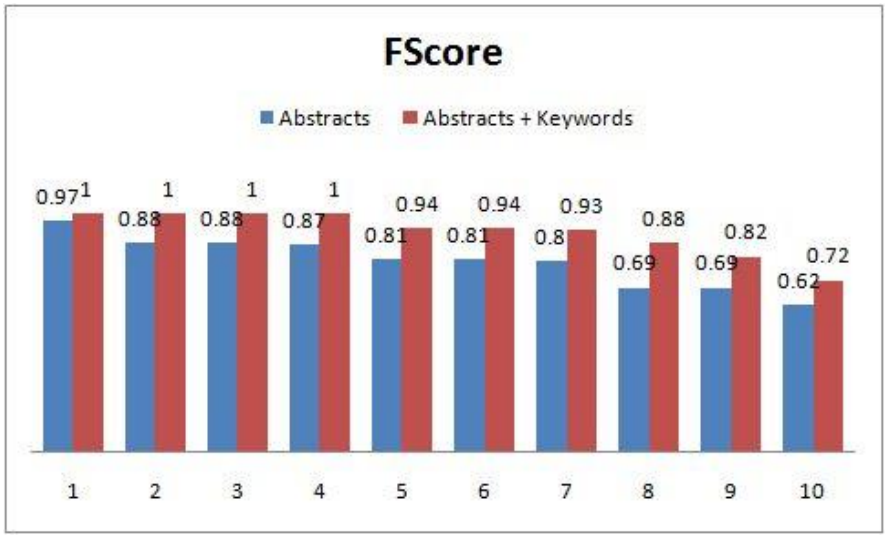

Figure 3.FScore comparison for text clustering

The values obtained during the second round of tests, when using the increased weight for keywords, are significantly better than the FScores obtained when using the unprocessed text files. Thus, increasing the weight of the keywords (10 times * 3 keywords) facilitated the identification of the relevant pieces of information within the text files, and, consequently, led to a more accurate clustering process.

Another interesting conclusion of the text clustering experimental results was that the Levenshtein distance metric produces very good results when noise is eliminated from the text files and the relevant information is easily identified and retrieved.

To conclude, the experimental validation conducted for compression based clustering in the context of text mining produced good results, the clusters generated based on informational content being highly accurate.

REFERENCES

[1] **Booch, G., Jacobson, I., Rumbaugh,J. (2010),*OMG Unified Modeling Language Specification***;*Version 2.3, First Edition*: March 2010. Retrieved 19.01.2011.http://www.omg.org/spec/UML/2.3/;
[2]**BZIP2 home page**: http://bzip.org/, last accessed 23.06.2014.
[3] **Cilibrasi, R. (2003),** ***The CompLearn Toolkit***, available at: http://www.**complearn**.org/;
[4] **Cilibrasi R, Vitányi,Paul M.B. (2005)**, ***Clustering by Compression***; *IEEE Transactions on Information Theory*, volume 51, pp. 1523-1545;
[5] **ClusTIO (2009),** Available at: http://www.softpedia.com/get/Science-CAD/ClusTIO.shtml.
[6] **De Hoon, M. J. L., Imoto, S., Nolan, J. and Miyano, S. (2004)**, ***Open Source Clustering Software***; *Bioinformatics*, **20** (9): 1453—1454;
[7**] Charu C. Aggarwal, Chandan K. Reddy (2013)**, ***Data Clustering: Algorithms and Applications***; CRC Press;
[8] **Grünwald, P., Vitányi, P. (2004),*Shannon Information and Kolmogorov Complexity***;
[9] **GZIP home page (2003)**: http://www.gzip.org/, last accessed 20.06.2014.
[10] **Hartigan, J. and Wong, M. (1979),** ***A k-means Clustering Algorithm***; *Journal of Applied Statistics*, 28;
[11]**JaccarSimilarity**http://www.code10.info/index.php?option=com_content&view=article&id=60:article_jaccard-similarity&catid=38:cat_coding_algorithms_data-similarity&Itemid=57, last accessed 20.06.2014;

[12] **Jain , A., Dubes, R. (1998)**, ***Algorithms for Clustering Data***; *Prentice Hall*;
[13] **Jaro, M. A. (1989)**, ***Advances in Record Linkage Methodology as Applied to the 1985 Census of Tampa Florida***; *Journal of the American Statistical Society*, 84 (406): 414–20;
[14] **Levenshtein VI (1966),** ***Binary Codes Capable of Correcting Deletions, Insertions and Reversals*****;** *Soviet Physics Doklady*, 10: 707–10;
[15] **Li, M., Chen, X, Li, X., Ma, B., Vitányi, Paul M.B. (2003)**, ***The Similarity Metric***; 14th ACM-SIAM Symp. Discrete Algorithms;
[16**] Li, M., Vitanyi, P. (2008),** ***An Introduction to Kolmokorov Complexity and its Applications – 3rd Edtition***, *Spinger*, ISBN: 978-0-387-33998-6;
[17] **Marmanis,H and Babenko D. (2009)**, ***Algorithms of the Intelligent Web***; *Manning Publications*;
[18] **Murty, M., Jain, A. and Flyn, P. (1999)**, ***Data Clustering: A Review***; *ACM Computing Surveys*, 31(3);
[19] **Milligan, G. W. (1996)**, ***Clustering Validation: Results and Implications for Applied Analyses***; *World Scientific Publ*;
[20] **Quartet Method** - ***The Quartet Method of hierarchical clustering***: http://www.sergioconsoli.com/Quartet.htm, last accessed on 27.01.2014.
[21] **RapidMine**r: http://rapid-i.com/content/view/181/196/, last accessed 10.06.2014.
[22] **Wang, K. (2009)**, ***CVAP: Cluster Validity Analysis Platform*** *(cluster analysis and validation tool)*, available at: http://www.mathworks.com/matlabcentral/fileexchange/14620-cvap-cluster-validity-analysis-platform-cluster-analysis-and-validation-tool;
[23] **WEKA**: http://www.cs.waikato.ac.nz/ml/weka/, last accessed 10.06.2014;
[24] **ZIP file format**: http://en.wikipedia.org/wiki/ZIP_(file_format), last accessed 15.06.2014;
[25] **IEEExplore**: www.ieeexplore.org, last accessed 20,06.2014;
[26] **Xu R., Wunch D.C (2009)**, ***Clustering***; *John Wiley & Sons*;
[27] **Michael W. Berry , Jacob Kogan (2010),*****Text Mining: Applications and Theory;*** *Wiley;*
[28] **Cernian Alexandra, Sgarciu Valentin and Carstoiu Dorin** (2011), ***Experimental Validation of the Clustering by Compression Technique***; *U.P.B. Scientific Bulletin*; Series C, Vol. 73, Iss. 3, 2011, ISSN 1454-234x, pp. 61-74.