EXPERIMENTAL VALIDATION OF THE CLUSTERING BY COMPRESSION TECHNIQUE

Alexandra Suzana CERNIAN[1], Valentin SGÂRCIU[2], Dorin CÂRSTOIU[3]

În zilele noastre, oamenii se confruntă cu o cerere din ce în ce mai mare de cunoştinţe şi informaţii. În acest context, clasificarea datelor este esenţială pentru obţinerea de informaţii structurate ca răspuns la interogările utilizatorilor. În această lucrare vom evalua rezultatele produse de o nouă tehnică de clasificare – clasificarea prin compresie - atunci când se aplică asupra unor seturi diferite de date. Procedeul de clasificare prin compresie se bazează pe o distanţă universală de similitudine, numită distanţă normală de compresie sau NCD, calculată pe baza dimensiunii fişierelor de date comprimate. Rezultatele experimentale arată că se pot clasifica corect fişiere de diferite tipuri, fără nici o informaţie prealabilă. NCD a dovedit capacitatea de a evalua distanţa dintre obiectele de diferite tipuri, prin aproximarea distanţei normale de informaţie (NID), o metrică universală, care există doar la nivel teoretic.

Nowadays, people have to deal with great demand on knowledge and information. In this context, data clustering is essential to getting structured information in response to user queries. In this paper, we assess the results of a new clustering technique – clustering by compression – when applied to different sets of data. The clustering by compression procedure is based on a universal distance metric, the normalized compression distance or NCD, computed from the lengths of compressed files. Experimental results show that it can correctly cluster files of different types without any prior information. The NCD has proven its ability to evaluate the distance between objects of different types, by approximating the Normalized Information Distance (NID), a theoretical universal metric.

Keywords: clustering by compression, normalized compression distance, FScore

1. Introduction

In 2005, two researchers from Holland, Paul Vitányi and Rudi Cilibrasi, proposed a new idea for clustering data, based on compression algorithms [1].

The origin of their idea has the following three seeds: the Kolmogorov

[1] PhD student, Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania, e-mail: alexandra.cernian@aii.pub.ro

[2] Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania

[3] Prof., Faculty of Automatic Control and Computers, University POLITEHNICA of Bucharest, Romania

complexity [5], the normalized information distance [7] and the fact that compressors provide a good approximation of the amount of information contained in a data sample.

The outcome of their work is a new distance metric, called the Normalized Compression Distance (NCD), which is supposed to be a universal metric.

The aim of the current work is to assess whether the clustering by compression technique produces good output for different datasets, such as different types of genomes, text files, handwritten texts and heterogeneous files.

The outline of the rest of the paper is the following: Section 2 provides a theoretical background, discussing the clustering by compression, the NCD, which is the central point of this technique, and an overview of clustering algorithms; Section 3 presents the test platform, describing the system architecture and the validation procedure; Section 4 presents a detailed analyses of the test results; Section 5 draws the conclusions of this work.

2. Theoretical background

2.1. Clustering by compression

The clustering by compression method works as follows. First, it builds the similarity matrix based on a universal distance metric called the normalized compression distance or NCD, computed from the lengths of compressed data files (singly and in pairwise concatenation). Second, it applies a clustering method.

Let us consider the following: *C* - a compressor (e.g ZIP), *x, y* - two files, *xy* – a file obtained by concatenating x and y, *C(x*) - the size of the compressed version of x using C, *C(y)* - the size of the compressed version of y using C, *C(xy)* - the size of the compressed version of the concatenation xy using C. Then the NCD is defined as follows [1]:

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

The NCD is related to the following concepts: the Kolmogorov complexity [3], the normalized information distance [1] and the fact that compression algorithms provide a good approximation of the informational content.

The Kolmogorov complexity, *K(x)*, of an object x (such as a piece of text), is a measure of the computational resources needed to specify the object. It can be defined as the length of the shortest binary description of *x* [1]. *K(x)* can be considered as the length of the shortest program (in any programming language) which prints *x* and then stops [1]. Unfortunately, the Kolmogorov complexity is not computable. But, an efficient idea is to approximate *K(x)* with the length of the compressed version of the object *x*, after it has been compressed with a real compressor (such as ZIP, BZIP, etc.).

Based on the Kolmogorov complexity, a group of researchers proposed the Normalized Information Distance (NID) [9] between two objects, *x* and *y*. It is defined as follows:

$$NID(x,y) = \frac{\max\{K(x\,|\,y), K(y\,|\,x)\}}{\max\{K(x), K(y)\}} \tag{2}$$

where *K(y|x)* is the extra number of bits necessary to describe y apart from describing x.

Starting from formula (2) and approximating K(x) with the length of the compressed version of x, Cilibrasi and Vitanyi defined the NCD using (1).

The NCD is not restricted to a specific application. Evidence of successful application has been reported in areas such as genomics, virology, languages, literature, music, handwritten digits and astronomy [1].

2.1.1. Clustering methods

The goal of clustering methods is to group elements sharing common information. The key element of clustering is the concept of similarity [8]. The purpose of clustering is to gather the elements which are most similar between them, but less similar to all the others [10]. The members of a cluster must be very similar to each other and very dissimilar to the members of the other clusters. In the clustering process, there is no predefined structure of the data and no examples to show what kind of relations would be valid among the data. Consequently, it is perceived as an unsupervised process [10].

Clustering methods can be divided into the following three categories: partitional methods, hierarchical methods – agglomerative and divisive - and quartet methods.

The test platform used for the work presented in this paper only contains a hierarchical agglomerative algorithm (UPGMA) and a quartet method proposed by Vitanyi and Cilibrasi [1].

2.2.1. Hierarchical methods

Hierarchical clustering algorithms organize the data into a hierarchy of nested groups. In other words, these algorithms create a structure of clusters within clusters. A hierarchical algorithm for news articles could come up with four large groups that represent general topics, such as politics, sports, business, and technology. Within each group, there are several subgroups. For example, the sports news group might have the following subgroups: basketball news, football news, and so on.

These algorithms involve N-1 steps. At each step *s*, a new element is assigned to a cluster, based on the information produced in the previous steps. Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters. They are also called bottom-up. The most famous agglomerative hierarchical clustering method is UPGMA (**U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic Mean). The output of a hierarchical algorithm is called a *dendrogram* [10].

2.2.2. Quartet methods

The quartet method of clustering was first introduced in [1], where the authors presented the method in a formal way. If we have *n* objects we want to cluster, the quartet method accepts as input a distance matrix. Based on this distance matrix, the quartet method produces as output a *dendrogram* with a special topology, called *quartet topology* [1]. If we consider 4 objects - *a, b, c, d* - then a quartet topology is a tree of arity 3, with 4 leaves and 2 internal nodes, which means that the tree consists of two subtrees with two leaves each. Quartet topologies can be connected to form a quartet tree [1].

The goal of the quartet method is to find (or approximate as closely as possible) the tree that embeds the maximal number of consistent quartet topologies from a given set Q of quartet topologies [1], with minimum total cost. This is called the *Maximum Quartet Consistency (MQC)* problem. In [1], the authors propose a quartet clustering method called *Minimum Quartet Tree Cost (MQTC),* which is based on the optimization of the MQC problem.

3. The test platform

3.1. The system architecture

The main purpose of the current work is to have an exhaustive assessment of the performance of the clustering by compression method. Consequently, we have developed a test platform which includes several compressors, several distance metrics and several clustering algorithms. The platform is called *EasyClustering*. It is designed in a modular, object oriented manner, in order to facilitate any future developments, such as the integration of new compressors, distance metrics or clustering algorithms.

An overview of the architecture of the EasyClustering test platform is presented in Fig. 1. The purpose is to provide a flexible and scalable system, which can be used by researchers to perform a variety of tests using different clustering and compression algorithms. The platform is designed in a modular, object oriented manner, in order to facilitate any future developments, such as the integration of new distance metrics or clustering algorithms.

EasyClustering was developed in Java [11]. This insures a high degree of portability, the only requirement for the platform to run being the Java Runtime Environment (JRE).

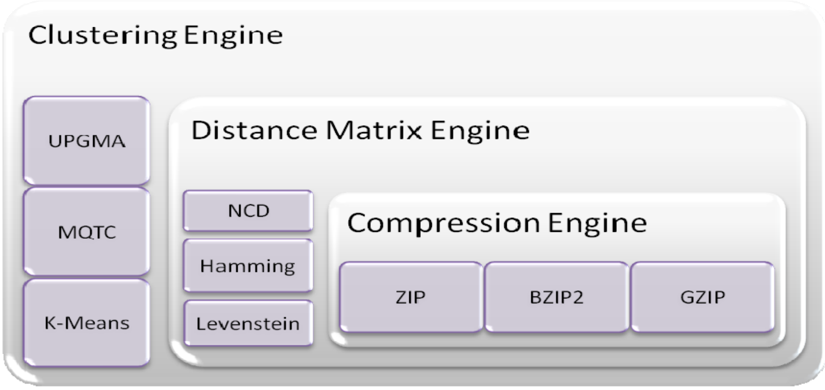


Fig 1. Overview of the test platform architecture

The steps of a clustering process are the following:

- The first step is to decide over a dataset we want to cluster.
- When using the NCD, all the elements in the dataset are compressed with Compression Engine singly and concatenated 2 by 2. When other distance metrics are used, this step is skipped.
- The Distance Matrix Engine computes the similarity matrix according to the distance metric chosen by the user.
- The Clustering Engine interprets the distance matrix according to the clustering algorithm selected by the user and generates the clusters.
- The clusters are displayed in HTML format.

3.2. The test method

The tests will cover 5 main areas of application of the clustering by compression: genomes clustering, text clustering, music clustering, image clustering and optical character recognition. Two clustering methods (UPGMA and MQTC) will be used in combination with the 2 distance metrics (NCD, Levenstein) and 3 compressors (ZIP, BZIP2, GZIP) and applied to 5 datasets. Thus, a total of 60 tests will be conducted and presented.

To assess the quality and robustness of the classification process, the FScore measure will be used [12]. Given a particular predefined class L_r of size n_r and a particular cluster *Si* of size n_i , suppose n_{ri} documents in the cluster S_i belong to L_r, then the FScore of this class and cluster is defined to be:

$$F(L_r, S_i) = \frac{2 * R(L_r, S_i) * P(L_r, S_i)}{R(L_r, S_i) + P(L_r, S_i)} \tag{3}$$

where $R(L_r, S_i) = \frac{n_{ri}}{n_r}$ is called the recall value for the class L_r and the cluster S_i

and $P(L_r, S_i) = \frac{n_{ri}}{n_i}$ is called the precision value for the class L_r and the cluster S_i. Roughly, the precision answers the question: “How many of the documents in this cluster belong there?”, whereas the recall answers the question: “Did all of the documents that belong in this cluster make it in?”

The FScore of the entire clustering solution is defined as the sum of the individual FScore for each class, weighted according to the class size: $FScore = \sum_{r=1}^{c} \frac{n_r}{n} F(L_r)$.

A perfect clustering solution will be the one in which every class has a corresponding cluster containing exactly the same documents in the resulting clustering. In this case, the FScore will be one. The higher the FScore value, the better the clustering solution is.

Fig. 2 presents the workflow of the test method.

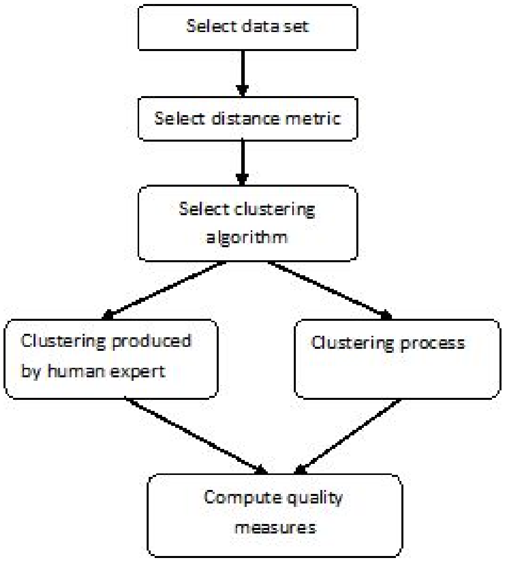


Fig. 2. Test method workflow

4. Experimental results

This section presents the detailed results which were obtained. A total of 75 tests were performed (3 compressors x 1 distance metrics x 5 clustering methods x 5 datasets). For each of the 5 test datasets, a screenshot of one of the best clustering results will be presented, as well as a table with the FScores and a discussion of the most interesting aspects of the results.

4.1. Mammals clustering

The scientific classification of animals is designed to classify the millions of animals into categories based on shared traits, in order to better understand how they are related to each other.

The evolutionary tree built from 24 mammals is also presented in [1]. The authors used the CompLearn platform [6] and they provide a very in-depth analysis of the reconstruction of the phylogenetic tree.

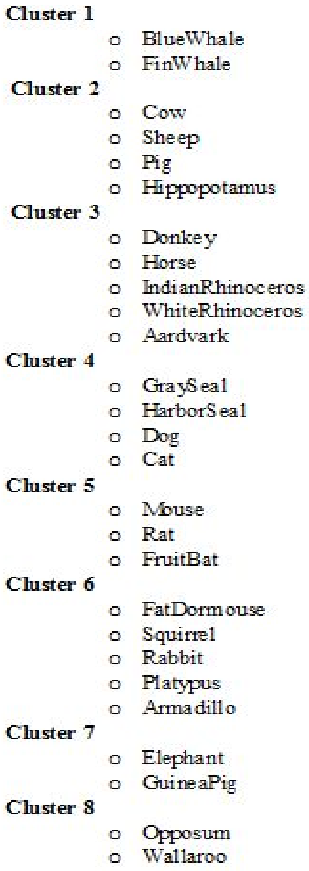
For this work, we have used a set of 32 mammals genomes. The dataset was taken from the download kit available with the CompLearn platform. The genomes are represented as complete DNA sequences and they are saved as text files. The following 32 genomes were used: cow, sheep, blue whale, fin whale, pig, hippopotamus, donkey, horse, gray seal, harbor seal, cat, dog, Indian rhinoceros, white rhinoceros, mouse, rat, fruit bat, fat dormouse, squirrel, rabbit, platypus, armadillo, guinea pig, elephant, opossum and wallaroo.

The following 19 current orders of placental mammals (Eutheria) are presents in [13]:

1) ***Artiodactyla*** (even-toed ungulates: antelope, deer, camels, wild pigs, wild cows, mt. sheep, hippos, etc.)
2) ***Carnivora*** (canines (coyotes, foxes, wolves), cats (bobcats, lynx, mountain lion) tigers, lions, bears (black bears, panda, polar bear, grizzly, etc.), weasels, minks, otters and pinnipeds (seals and sea lions), etc.)
3) ***Cetacea*** (whales, dolphins)
4) ***Chiroptera*** (bats)
5) ***Dermoptera*** (colugos or flying lemurs)
6) ***Edentata*** (Xenarthra) (toothless mammals –¬ armadillos, sloths, hairy anteaters)
7) ***Hyracoidae*** (hyraxes, dassies)
8) ***Insectivora*** (insect-eaters: hedgehogs, moles, shrews)
9) ***Lagomorpha*** (rabbits, hares, pikas)
10) ***Macroscelidea*** (elephant-shrews)
11) ***Perissodactyla*** (odd-toed ungulates: horses, rhinos, tapirs)
12) ***Pholidata*** (pangolins)
13) ***Pinnipedia*** (seals and walruses) - also included under carnivores.
14) ***Primates*** (apes, monkeys, lemurs, people)
15) ***Proboscidea*** (elephants)
16) ***Rodentia*** (rodents: rats, mice, squirrels, chipmunks, beaver, gerbils, hamsters, etc.)
17) ***Scandentia*** (19 tree shrews)
18) ***Sirenia*** (dugongs and manatees)
19) ***Tubulidentata*** (aardvarks)

The mammals classification above has been used as reference to validate the test results.

Fig. 3 presents the tree obtained using the BZIP2 compressor and the UPGMA clustering method.



Cluster 1
- BlueWhale
- FinWhale

Cluster 2
- Cow
- Sheep
- Pig
- Hippopotamus

Cluster 3
- Donkey
- Horse
- IndianRhinoceros
- WhiteRhinoceros
- Aardvark

Cluster 4
- GraySeal
- HarborSeal
- Dog
- Cat

Cluster 5
- Mouse
- Rat
- FruitBat

Cluster 6
- FatDormouse
- Squirrel
- Rabbit
- Platypus
- Armadillo

Cluster 7
- Elephant
- GuineaPig

Cluster 8
- Opposum
- Wallaroo

Fig. 3. Mammals / BZIP2 / UPGMA

Table 1 presents the FScore values for all the tests performed for clustering the 32 mammals genomes.

Table 1

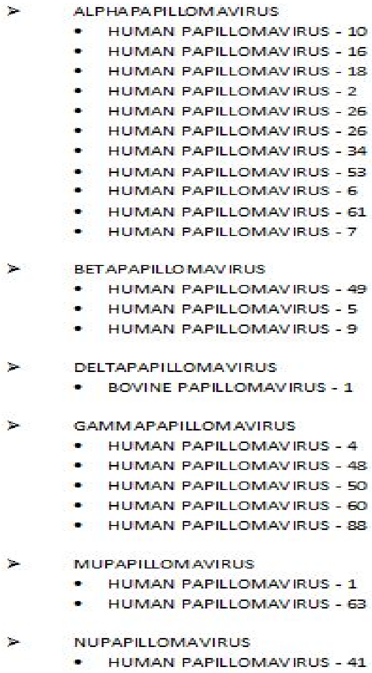
FScore for clustering mammals

	UPGMA + ZIP	**UPGMA + BZIP2**	**UPGMA + GZIP**	**MQTC + ZIP**	**MQTC + BZIP2**	**MQTC + GZIP**
FScore	0.90	0.95	0.92	0.86	0.95	0.92

Both UPGMA and MQTC clustering methods produced very high scores and generated correct clusters. In this case, the BZIP2 compressor had the best performance.

4.2 Classification of the human papillomavirus genomes

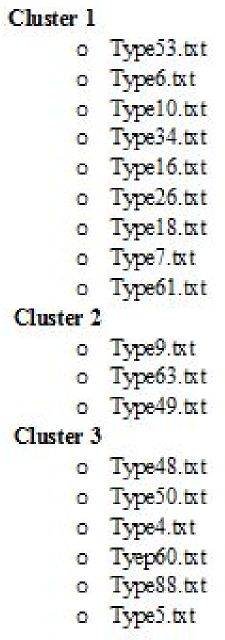
The human papillomavirus (HPV) is a member of the papillomavirus family of viruses capable of infecting humans. Currently, there are nearly 200 known types of HPV [14]. The official classification [15] presented in Fig. 4 is used as a reference in order to validate the results of our tests.



- ALPHAPAPILLOMAVIRUS
  - HUMAN PAPILLOMAVIRUS - 10
  - HUMAN PAPILLOMAVIRUS - 16
  - HUMAN PAPILLOMAVIRUS - 18
  - HUMAN PAPILLOMAVIRUS - 2
  - HUMAN PAPILLOMAVIRUS - 26
  - HUMAN PAPILLOMAVIRUS - 26
  - HUMAN PAPILLOMAVIRUS - 34
  - HUMAN PAPILLOMAVIRUS - 53
  - HUMAN PAPILLOMAVIRUS - 6
  - HUMAN PAPILLOMAVIRUS - 61
  - HUMAN PAPILLOMAVIRUS - 7
- BETAPAPILLOMAVIRUS
  - HUMAN PAPILLOMAVIRUS - 49
  - HUMAN PAPILLOMAVIRUS - 5
  - HUMAN PAPILLOMAVIRUS - 9
- DELTAPAPILLOMAVIRUS
  - BOVINE PAPILLOMAVIRUS - 1
- GAMMAPAPILLOMAVIRUS
  - HUMAN PAPILLOMAVIRUS - 4
  - HUMAN PAPILLOMAVIRUS - 48
  - HUMAN PAPILLOMAVIRUS - 50
  - HUMAN PAPILLOMAVIRUS - 60
  - HUMAN PAPILLOMAVIRUS - 88
- MUPAPILLOMAVIRUS
  - HUMAN PAPILLOMAVIRUS - 1
  - HUMAN PAPILLOMAVIRUS - 63
- NUPAPILLOMAVIRUS
  - HUMAN PAPILLOMAVIRUS - 41

Fig. 4. HPV genomes taxonomy

We tested the clustering of 18 HPV genomes. The source of the complete genomes was [16]. The genomes are the DNA sequences of the different type of virus and they are saved as text files. The text files were provided as input for the test platform. In this case, the best results were obtained when using the GZIP compressor and the MQTC clustering algorithm. The output is presented in Fig. 5.



Cluster 1

- Type53.txt
- Type6.txt
- Type10.txt
- Type34.txt
- Type16.txt
- Type26.txt
- Type18.txt
- Type7.txt
- Type61.txt

Cluster 2

- Type9.txt
- Type63.txt
- Type49.txt

Cluster 3

- Type48.txt
- Type50.txt
- Type4.txt
- Tyep60.txt
- Type88.txt
- Type5.txt

Fig. 5. HPV / GZIP / MQTC

Table 2 presents the FScore values for all the tests performed for clustering the 18 HPV genomes.

Table 2

FScore for clustering HPV genomes

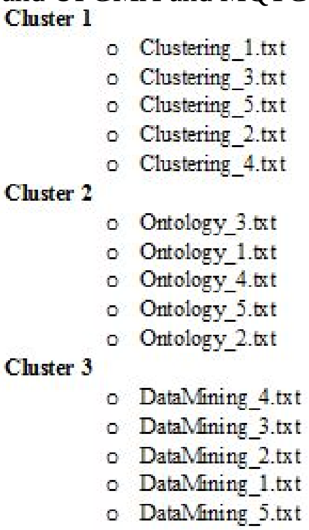
	UPGMA + ZIP	UPGMA + BZIP2	UPGMA + GZIP	MQTC + ZIP	MQTC + BZIP2	MQTC + GZIP
FScore	0.78	0.84	0.83	0.82	0.83	0.85

Both UPGMA and MQTC clustering methods produced very high scores and generated correct clusters when working in combination with the BZIP2 and GZIP compressors. The FScores in these cases are very close. In this case, the ZIP compressor produced the lower results. However, the highest FScore is 0.85, a lot lower than in the case of the 32 mammals genomes, when the FScores obtained were very close to 1.

4.3 Text clustering

For testing the performance of the clustering by compression technique with text files, we used a selection of 15 scientific paper abstracts from IEEExplore [17]. The 15 elements of the dataset belong to three categories: clustering, data mining and ontology. In order to build the dataset, we selected 5 papers from each category, copied their abstracts into text files saved locally on the computer, and named each file according the following pattern: CatgoryName_Index.txt.

Fig. 6 shows the results produced by the EasyClustering platform when using BZIP2 as compressor and UPGMA and MQTC as clustering algorithms.



Cluster 1

- Clustering_1.txt
- Clustering_3.txt
- Clustering_5.txt
- Clustering_2.txt
- Clustering_4.txt

Cluster 2

- Ontology_3.txt
- Ontology_1.txt
- Ontology_4.txt
- Ontology_5.txt
- Ontology_2.txt

Cluster 3

- DataMining_4.txt
- DataMining_3.txt
- DataMining_2.txt
- DataMining_1.txt
- DataMining_5.txt

Fig.6. Abstracts/BZIP2/MQTC/UPGMA

Table 3 presents the FScore values for all the tests performed for clustering the 15 scientific abstracts.

Table 3

FScore for clustering scientific abstracts

	UPGMA + ZIP	**UPGMA + BZIP2**	**UPGMA + GZIP**	**MQTC + ZIP**	**MQTC + BZIP2**	**MQTC + GZIP**
FScore	0.90	1	0.96	0.93	1	0.97

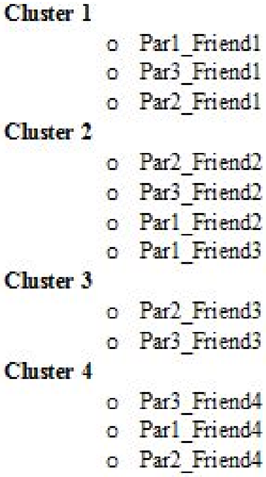
The explanation for the very high scores obtained in this case resides in the fact that the abstracts contain information of high relevance to the articles, as well as a large number of keywords specific for the domains they belong to.

4.4 Recognition of Handwritten Text

In [1], the authors presented the results of clustering handwritten digits using the clustering by compression technique implemented in the CompLearn tool. The results of their test were very good, the clustering having the score of 1 (100% correct).

In this work, we used handwritten text paragraphs. We selected 3 different paragraphs, each paragraph having 3 or 4 phrases. We asked 4 friends to write by hand each of these paragraphs, scan them, and give them to us. We used the scanned versions for the clustering process. The purpose was to see if the application places all the files produced by the same person in the same cluster, or if it will be able to interpret the meaning of the files and to cluster them based on their content.

Fig. 7 shows the results produced by the EasyClustering platform when using BZIP2 as compressor and UPGMA as clustering algorithm.



Cluster 1

- Par1_Friend1
- Par3_Friend1
- Par2_Friend1

Cluster 2

- Par2_Friend2
- Par3_Friend2
- Par1_Friend2
- Par1_Friend3

Cluster 3

- Par2_Friend3
- Par3_Friend3

Cluster 4

- Par3_Friend4
- Par1_Friend4
- Par2_Friend4

Fig. 7. Handwritten text/ BZIP2 /UPGMA

Table 4 presents the FScore values for all the tests performed for clustering the 12 handwritten texts.

Table 4

FScore for clustering handwritten text

	UPGMA + ZIP	**UPGMA + BZIP2**	**UPGMA + GZIP**	**MQTC + ZIP**	**MQTC + BZIP2**	**MQTC + GZIP**
FScore	0.83	0.91	0.88	0.86	0.89	0.87

4.5 Clustering of Heterogeneous Files

The last test presented in this paper involves clustering heterogeneous files with the clustering by compression technique. In order to do this test, we have established a heterogeneous dataset consisting of 20 files having different extensions: .doc, .txt, .pdf, .ppt, and .jpg. The purpose was to cluster the documents based on common content, and not on the type of the file.

Table 5 presents the FScore values for all the tests performed for clustering the 20 heterogeneous files.

Table 5

FScore for clustering heterogeneous files

	UPGMA + ZIP	**UPGMA + BZIP2**	**UPGMA + GZIP**	**MQTC + ZIP**	**MQTC + BZIP2**	**MQTC + GZIP**
FScore	0.73	0.82	0.79	0.80	0.84	0.80

The results of the tests showed that clustering by compression is not completely able to interpret the content of documents independently of their type. For instance, it always manages to correctly cluster the text files, but it tends to group the .pdf files based on their type, and not on their content. This happens because the .pdf file is represented and interpreted by the computer in a totally different manner than text files, thus inducing some noisy elements to the clustering process.

5. Conclusions

The clustering by compression technique produced good results during all tests which were conducted in this work. It proved that it can correctly cluster files of different types (genomes, text, handwritten text) without any prior information. So, the NCD has proven its capability to evaluate the distance between objects of different types. Therefore, we can appreciate that the NCD provides a good evaluation on the NID presented in [5].

As an overview of the tests presented in this paper, we may conclude that the BZIP2 compressor produced the best results, in combination with both UPGMA and MCQP as clustering method. The GZIP compressor also produced very good results, not far from BZIP2. Regarding the clustering algorithms, they both produced clustering results with very similar FScores. The only disadvantage of MTQC might be the fact that it is much slower than UPGMA.

In conclusion, the clustering by compression has proven its potential. Finding new areas of application of the method would be of great interest to fully exploit the capabilities of this method, from a research or commercial point of view.

Acknowledgements.

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour, Family and Social Protection through the Financial Agreement POSDRU/6/1.5/S/16.

REFERENCES

[1] *R. Cilibrasi, P.M.B. Vitányi*, "Clustering by Compression", in IEEE Transactions on Information Theory, 2005

[2] *R.Cilibrasi, P. Vitányi, R. de Wolf*, "Algorithmic Clustering of Music Based on String Compression", Computer Music Journal of the MIT, 2004

[3] *P. Grünwald, P. Vitányi*, "Shannon Information and Kolmogorov Complexity", 2004

[4] *G.W. Milligan*, "Clustering Validation: Results and Implications for Applied Analyses", World Scientific Publ., 1996

[5] *Ming Li, Xin Chen, Xin Li, Bin Ma, P.M.B. Vitányi*, "The Similarity Metric", 14th ACM-SIAM Symp. Discrete Algorithms, 2003

[6] *R. Cilibrasi*, CompLearn, http://www.complearn.org

[7] *J. Delahaye*, “Classer musique, langues, images, textes et genomes”, in Pour la science, **n°317**, 2004

[8] ****Distance and similarity measures: http://reference.wolfram.com/mathematica/guide/DistanceAndSimilarityMeasures.html

[9] *P.M.B. Vitányi, F.J. Balbach, R.L. Cilibrasi, Ming Li*, Chapter 3 “Normalized Information Distance”, Information Theory and Statistical Learning, pp. 35-71, 2009.

[10] *M. Murty A. Jain, P. Flyn*, “Data clustering: A review”, ACM Computing Surveys, **31(3),** 1999

[11] *Kathy Sierra, B. Bates*, “Head First Java – Second Edition”, O’Reilly, 2009

[12] *C.J. van Rijsbergen*, Information Retrieval (2nd ed.), Butterworth, 1979

[13]**** Mammalian phylogeny: http://www.exploringnature.org/db/detail.php?dbID=87&detID=1193

[14] Centers for Disease Control and Prevention, “Genital HPV Infection - CDC Fact Sheet". (CDC), 2008. http://www.cdc.gov/std/HPV/STDFact-HPV.htm

[15] ****Human papillomavirus taxonomy: http://www.metapathogen.com/papillomavirus/

[16] ****HPV genomes: http://www.ncbi.nlm.nih.gov/nuccore

[17]**** IEEExplore Digital Library: www.ieeexplore.org