ON USING THE NORMALIZED COMPRESSION DISTANCE TO CLUSTER WEB SEARCH RESULTS

Alexandra Suzana Cernian, Liliana Dobrica, Dorin Carstoiu and Valentin Sgarciu
Faculty of Automatic Control and Computer Science, University Politehnica of Bucharest
313 Splaiul Independentei, Bucharest, Romania

Keywords: Organizing Web Search Results, Clustering by Compression, Normalized Compression Distance (NCD), Clustering Methods.

Abstract: Current Web search engines return long lists of ranked documents that users are forced to sift through to find relevant documents. This paper introduces a new approach for clustering Web search results, based on the notion of clustering by compression. Compression algorithms allow defining a similarity measure based on the degree of common information. Classification methods allow clustering similar data without any previous knowledge. The clustering by compression procedure is based on a parameter-free, universal, similarity distance, the normalized compression distance or NCD, computed from the lengths of compressed data files. Our goal is to apply the clustering by compression algorithm in order to cluster the documents returned by a Web search engine in response to a user query.

1 INTRODUCTION

Current Web search engines return long lists of ranked documents that users are forced to browse through to find relevant documents. Most of today's Web search engines (e.g., Google, Yahoo) follow this paradigm, thus making it difficult for users to find the information they are looking for.

This paper introduces a new approach for clustering Web search results, based on the notion of clustering by compression. On the one hand, compression algorithms allow defining a universal similarity measure based on the degree of common information shared by a number of documents. On the other hand, classification methods allow clustering similar data without any previous knowledge.

Our goal is to apply the clustering by compression algorithm (Cilibrasi and Vitanyi, 2005) for clustering the documents returned by a Web search engine in response to a user query. The method is based on the fact that compression algorithms offer a good evaluation of the actual quantity of information comprised in the data to be clustered, without requiring any previous processing. It defines the normalized compression distance (NCD), which can be used as distance metric with the clustering algorithms. The most important characteristic of the NCD is its universality.

The rest of the paper is structured as follows: in Section 2 we make a short review of some related work, in Section 3 we introduce some thematic aspects related to our work, in Section 4 we present the test platform we have set up in order to exemplify our solution, in Section 5 we present and validate our results and in Section 6 we draw a conclusion.

2 RELATED WORK

In recent years, link information began to play a key role in some web applications. Nowadays, the applications of clustering Web objects (including web-pages, purchase items, queries, users, and etc.) use the link information at different levels.

Traditional clustering methods ignore the link information and cluster objects based on content features. However, some of them treat link information as additional features. (Su et al., 2001) described a correlation-based document clustering method, which measures the similarity between web-pages based on their simultaneous visits. The agglomerative clustering approach described in

Cernian A., Dobrica L., Carstoiu D. and Sgarciu V. (2010).
ON USING THE NORMALIZED COMPRESSION DISTANCE TO CLUSTER WEB SEARCH RESULTS.
In *Proceedings of the 5th International Conference on Software and Data Technologies*, pages 293-298
DOI: 10.5220/0002926102930298

(Beeferman and Berger, 2000) is another approach, which uses the "click-through data" method, forming a bipartite graph of queries and documents. However, it does not take into account the content features of both query and document, leading to an ineffective clustering.

So far, several Web search results clustering systems have been implemented. We can mention four of them. Firstly, (Cutting et al., 1992) have created the Scatter/Gather system to cluster Web search results. This system is based on two clustering algorithms: *Buckshot* – fast for online clustering and *Fractionation* – accurate for offline initial clustering of the entire set. This system has some limitations due to the shortcomings of the traditional heuristic clustering algorithms (e.g. k-means) they used. Secondly, (Zamir and Etzioni, 1998) proposed in an algorithm named Suffix Tree Clustering (STC) to automatically group Web search results. STC operates on query results snippets and clusters together documents with large common subphrases. The algorithm first generates a suffix tree where each internal node corresponds to a phrase, and then clusters are formed by grouping the Web search results that contain the same "key" phrase. Afterwards, highly overlapping clusters are merged. Thirdly, (Stefanowski and Weiss, 2003) developed Carrot^2, an open source search results clustering engine. Carrot^2 can automatically organize documents (e.g. search results) into thematic categories. Apart from two specialized document clustering algorithms (Lingo and STC), Carrot^2 provides integrated components for fetching search results from various sources including YahooAPI, GoogleAPI, MSN Live API, eTools Meta Search, Lucene, SOLR, Google Desktop and more. Finally, (Zhang and Dong, 2004) proposed a semantic, hierarchical, online clustering approach named SHOC, in order to automatically group Web search results. Their work is an extension of O. Zamir and O. Etzioni's work. By combining the power of two novel techniques, key phrase discovery and orthogonal clustering, SHOC can generate suggestive clusters. Moreover, SHOC can work for multiple languages: English and oriental languages like Chinese.

3 THEORETICAL FOUNDATION

Clustering is one of the most useful tasks in the data mining process for discovering groups and identifying interesting distributions and patterns in the underlying data. The clustering problem is about partitioning a given data set into groups (clusters), so that the data points in a cluster are more similar to each other than points in other clusters. The relationship between objects is represented in a Proximity Matrix (PM), in which rows and columns correspond to objects. This idea is applicable in many fields, such as life sciences, medical sciences, engineering or e-learning.

3.1 Clustering by Compression

In 2004, Rudi Cilibrasi and Paul Vitanyi proposed a new method for clustering based on compression algorithms (Cilibrasi and Vitanyi, 2005). The method works as follows. First, it determines a parameter-free, universal, similarity distance, the normalized compression distance or NCD, computed from the lengths of compressed data files (singly and in pair-wise concatenation). Second, it applies a clustering method.

The method is based on the fact that compression algorithms offer a good evaluation of the actual quantity of information comprised in the data to be clustered, without requiring any previous processing. The definition of the normalized compression distance is the following: if x and y are the two objects concerned, and C(x) and C(y) are the lengths of the compressed versions of x and y using compressor C, then the NCD is defined as:

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

The most important advantage of the NCD over classic distance metrics is its ability to cluster a large number of data samples, due to the high performance of the compression algorithms.

The NCD is not restricted to a specific application. To extract a hierarchy of clusters from the distance matrix, a dendrogram (ternary tree) is determined by using a clustering algorithm. Evidence of successful application has been reported in areas such as genomics, virology, languages, literature, music, handwritten digits, and astronomy. The quality of the NCD depends on the performances of the compression algorithms.

3.2 Classification Methods

The goal of the classification methods is to group elements sharing the same information. Classification methods can be divided into the following three categories: distance methods, characters methods and quadruplets methods.

In order to achieve the objective of this work, only the methods of distance and the methods of

quadruplets (which also use criteria based on the notion of distance) were considered for analysis and implementation, for the reason that the normalized compression distance (NCD) can only be used with methods based on distance.

3.2.1 Distance Methods

Distance methods are based on the initial calculation of a distance matrix, which contains information about the resemblance between the elements to be classified. The matrix is built by using a distance metric and by calculating the similarity distance between all the elements, taken two by two. Afterwards, the classification algorithm uses the information contained in the distance matrix to build a binary tree or to group the elements into distinct groups. The two best known types of distance methods are the hierarchical methods of classification and the nonhierarchical methods of classification (K-Means) (Delahaye, 2004). In order to reach the goal of this work, only hierarchical methods were of interest. The disadvantage of the nonhierarchical methods consists in the fact that the number of classes must be known "a priori", which is big inconvenient when talking about queries launched on search engines.

The hierarchical methods of classification generate binary trees by unifying at each stage the two closest elements into a new partition. The elements to be classified will be the leaves of the resulting tree, while the internal nodes represent a network, which models the relationship between the elements. The closest elements are always children of same parents within the tree.

The most popular algorithm of the hierarchical classification is UPGMA ("Un-weighted Pair Group Method with Arithmetic Mean") (Carrot^2, 2002). It represents the reference method against which all new classification methods are compared.

3.2.2 Quadruplet Methods

The name of these methods comes from the structures called topologies of quadruplets (or simply quadruplets). A topology of quadruplet is a binary tree with 4 leaves and 2 internal nodes. These trees present some interesting properties which can be exploited in the process of classification. Given a set of 4 elements u, v, w, x there are always 3 possible distinct topologies of quadruplets which include these four elements in various configurations: uv|xw, ux|vw and uw|xv. A set of quadruplet topologies can be aggregated into a tree of quadruplets, which preserves the properties of the quadruplet topology: n leaves and (n-2) internal nodes, each internal node has exactly 3 neighbors, each leaf has exactly one neighbor (its parent), the tree is not rooted.

The methods of quadruplets are usually implemented in two stages (Boley et al., 1999) called the stage of inference and the stage of recombination or aggregation. The most known quadruplet methods are "Addtree" and "Quartet Puzzling".

4 TEST PLATFORM FOR CLUSTERING WEB PAGES

In order to evaluate the performance of the proposed method for clustering Web documents, we have developed a Java application which automatically organizes (clusters) small collections of documents, namely the search results, into thematic categories. The workflow of actions in the application is the following. By using Google SOAP Search API, the application retrieves the first 50 results returned by the Google search engine in response to a user request. These 50 Web pages are saved as text files, which will be further processed: stopwords are eliminated, a stemming algorithm is applied and HTML tags are removed. The processed files are then clustered using the Java actors we have implemented in order to test the clustering by compression technique. In order to assess the results, we have used manual clustering and Carrot^2 :

1. Manual clustering. We have launched a search on the Google search engine using the Rosetta keyword. Then, we looked at the first 50 results to see what was their thematic: space mission, famous stone, Unix software application, movie etc. We opened each link, looked at their content, and then decided into which group they fall. In our opinion, we distinguished the following groups: Stone (5), Space mission (9), Proteins (4), Software (6), Genomics (2), Music (2), Company (4), language learning (6), Movie (3), Books (2), Bio-software (2), Others (5).

After applying the clustering by compression technique to the Web search results, we compared the obtained clusters against the clusters deduced from the manual clustering.

2. Carrot^2. As described in Section 2, Carrot^2 is an Open Source search results clustering engine (Stefanowski and Weiss, 2003). It can automatically organize documents (e.g. search results) into thematic categories.

After applying the clustering by compression technique to the Web search results, we compared the clusters obtained against the clusters produced by Carrot[2].

4.1 Google SOAP Search API

The SOAP Search API (Google SOAP API, 2006) was created for developers and researchers who are interested in using Google Search as a resource in their applications. By integrating Google SOAP Search API Web service into their applications, software developers can access billions of Web pages directly from their own applications. Communication is performed via SOAP, an XML-based mechanism for exchanging information.

4.2 The Actors

For our current work, we have implemented in Java the following actors: *DistanceMatrixComputerActor* – depending on two parameters: (1) the compression algorithm to be used and (2) the distance metric to be used (Ionescu, 2005). The tests performed focused on ZIP and BZIP2 as compression techniques and on the normalized compression distance (NCD) for the second parameter. *ClusterTreeConstructorActor* - determining the type of classification method to be used. In order to conduct the tests, two classification algorithms have been chosen: UPGMA – belonging to the hierarchical class and MCQP ("Maximum Consistency based Quartet Puzzling") (Ionescu, 2005) – method based on the quadruplet topology. *ClusterTreeScoreActor* - having as input the distance matrix and issuing a value within the range 0 and 1, describing the quality of the binary tree obtained (Ionescu, 2005). *PresentationActor* – showing the graphical representation of the obtained binary tree, from which clusters will be identified (Ionescu, 2005).

These actors have been integrated into the Kepler Project (Altintas et. al., 2004), an open-source scientific system which allows scientists to design scientific workflows and execute them. The functionality of Kepler is based on the notion of actors. The actors are re-usable components which communicate with other actors through communication channels.

4.3 Techniques of Automatic Text Processing

The first text processing technique we have used is the elimination of stopwords., such as articles (e.g. *"the"*), prepositions (e.g. *"for"*, *"of"*) and pronouns (e.g. *"I"*, *"his"*). The second text processing technique we have used is stemming (Porter stemming, 2006), reducing different grammatical forms of a word to their base form. When using the entire Web pages for the clustering process, we also use regular expressions to eliminate HTML tags and punctuation marks, which could represent a source of noise when identifying the informational content of a document.

5 EXPERIMENTAL RESULTS

This section is a discussion regarding our experience when we performed clustering snippets and clustering entire Web pages.

5.1 Clustering Snippets

Since Google SOAP Search API allows launching a request on Google and retrieving the snippets returned by the Google search engine, the first thing we considered was clustering the Web pages based on these snippets (Table 1).

So, at this stage, the application performed the following actions:

- Launch a query on the Google search engine through the Google SOAP Search API. The keyword for our search was *Rosetta* (space mission, famous stone, Unix software application, movie etc);
- Retrieve the snippets for the first 50 Web documents returned by the search engine;
- Generate thematic clusters with the clustering by compression technique, using the ZIP and BZIP2 compression algorithms in conjunction with the UPGMA and MCQP classification methods.

Table 1: Clustering snippets.

	ZIP + UPGMA	BZIP2 + UPGMA	ZIP + MCQP	BZIP2 + MCQP
Speed	1s	1.2s	1.4s	1.5s
similarity with the manual clustering	72%	74%	71%	72%
similarity with Carrot[2]	66%	69%	67%	70%

After this first round of experiments, using the snippets returned by the search engine, the results were not satisfactory. We noticed that the clustering by compression algorithm did not seem to distinguish very precisely the thematic of the documents. We found that, for instance, our platform managed to gather the pages treating the Rosetta space mission, but, on the other hand, it mixed web pages about the Rosetta software with web pages about Rosette music band. Compared to our empirical evaluation, the number of clusters did not differ much, but the content of each cluster was not the one we had expected.

The main difference between Carrot^2 and our test platform is that Carrot^2 can place a Web page in several clusters, whereas our platform places each page in only one cluster. So, for our evaluation, we considered that the 2 clusterings are similar if they contain a certain Web page in clusters treating the same subject (for instance, Rosetta stone).

The problem now is to identify the cause of this unsatisfactory result. We considered using the factor analysis, which provides a classification of data and is recognized by the researchers for the relevance of its results. Factor analysis (Morrison, 1990) is a technique for data reduction. It tries to find a new set of variables, which can express the "common" information among the original variables. Widely used in applied statistics, these techniques are powerful tools for analysis. In our case, factor analysis has played the role of the classification algorithm, since it was based on the distance matrix, which it had to "translate" and project in a plane. The results obtained are as follows (figure 1):

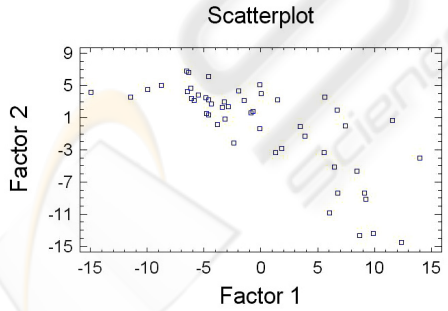

Figure 1: Factor analysis for snippets.

In the above representation, one can easily see that groups are not precisely formed, so we can now appreciate that the problem in the clustering we have obtained is due to the distance matrix and not the classification algorithm. But originally, this matrix is calculated using the sample of 50 snippets, on which we apply the compression algorithms chosen, as described by the NCD in Section 2. So, the problem could be generated either by the compression algorithm, or by the input data. Since the previous tests (Ionescu, 2005) have shown that ZIP and BZIP2 algorithms have good performance with the clustering by compression technique, compressors have been eliminated as a possible cause of the malfunction. Therefore, at this point, we can say that the cause of misclassification is the input data.

5.2 Clustering Web Pages

The next approach will use the entire Web page, not just a snippet, as input for the clustering procedure. From the textual information point of view, a Web page consists of: HTML tags, code entities (&amp), special characters (#, &, %), meaningful information and stop words.

So, our goal right now is processing Web pages in order to keep only meaningful words. Thus, using regular expressions, HTML tags will be removed. Stop words are removed from text using a Java implementation for eliminating stop words from documents written in English. The final processing step will be to apply the Porter stemming algorithm. Afterwards, we apply the clustering by compression technique on the processed files. At this point, we expect this process to be rather time consuming, due to the fact that the Web search results are processed before clustering. However, for the moment, we are interested in the clustering performances.

The results we have obtained in this case are depicted in the following table (Table 2):

Table 2: Clustering Web pages.

	ZIP + UPGMA	BZIP2 + UPGMA	ZIP + MCQP	BZIP2 + MCQP
Speed	2.7s	2.4s	3s	3.1s
similarity with the manual clustering	92%	94%	94%	95%
similarity with Carrot2	87%	89%	88%	88%

The projection in the plan of the matrix of distances calculated using the BZIP2 compressor and the UPGMA classification algorithm shows that specific groups are formed by thematic criteria (figure 2).

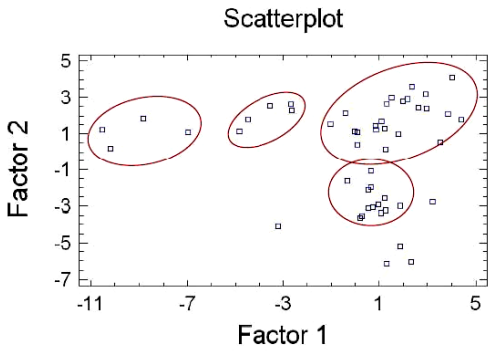

Figure 2: Factor analysis for Web pages.

Regarding the two compressors (ZIP and BZIP2) and the two classification algorithms used (UPGMA and MCQP), we can conclude the following:

- The BZIP2 compressor provided better results than ZIP.
- The MCQP algorithm is slower than UPGMA and the results are very similar.
- The best speed/performance ratio is provided by the BZIP2 + UPGMA combination.

6 CONCLUSIONS

Clustering by compression produced good results during our tests conducted in order to improve the relevance of the results of queries launched on the Web. The application can correctly classify Web documents without any "a priori" information. The NCD proved its capacity to objectively evaluate the distance between objects of various types by approximating the informational content. The best speed/performance ratio is provided by the BZIP2 - UPGMA combination. However, the results proved that the quality and the robustness of the process of clustering Web pages by compression depend on the pre-processing techniques applied to the Web documents. Consequently, as perspectives, we are considering finding better techniques to process the Web documents, in order to improve the speed of the clustering process, as well as to implement a Web interface which will allow the users to visualize the clustered data.

ACKNOWLEDGEMENTS

This work was supported by CNCSIS – UEFISCSU, project number PNII - IDEI 1238/2008.

REFERENCES

Cilibrasi, R., Vitanyi, P., 2005. Clustering by compression. *IEEE Transactions on Information Theory*, Vol. 51, No. 4, pp 1523–1545.

Grunwald, P., Vitanyi, P., 2004. Shannon Information and Kolmogorov Complexity.

Ionescu, T., 2005. Etude des méthodes de classification par compression , Supélec Gif-sur-Yvette, France.

Delahaye, J., 2004. Classer musique, langues, images, textes et genomes, *Pour la science,* n°317.

Altintas. I., Berkley, C., Jaeger E., Jones, M. Ludascher, B. and Mock, S., 2004. Kepler: an extensible system for design and execution of scientific workflows, *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*, p: 423-424.

Morrison, D. F., 1990. *Multivariate Statistical Methods*. New York: McGraw-Hill.

Carrot2, 2002: http://project.carrot2.org/

Zamir, O., Etzioni, O., 1998. Web document clustering: A feasibility demonstration. *Proceedings of SIGIR '98*, pp. 46--53.

Su, Z., Yang, Q., Zhang, H. J., X,u X., Hu, Y. H., 2001. Correlation-based Document Clustering using Web Logs, In *Proceedings of the 34th Hawaii International Conference On System Sciences(HICSS-34)*.

Beeferman, D., Berger, A., 2000. Agglomerative clustering of a search engine query log, In *Proceedings of the Sixth ACM SIGKDD*, pp. 407-416.

Google SOAP API, 2006: http:// code.google.com/apis/soapsearch/

Porter stemming algorithm, 2006: http:// tartarus.org/~martin/PorterStemmer/

Cutting, D., Karger, D., Pedersen, J., Tukey, J. W., 1992. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. *Proceedings of the 15th Annual International ACM/SIGIR Conference*, Copenhagen.

Stefanowski, J., Weiss, D., 2003. Carrot^2 and Language Properties in Web Search Results Clustering. In: Lecture Notes in Artificial Intelligence: Advances in *Web Intelligence, Proceedings of the First International Atlantic Web Intelligence Conference*, Madrid, Spain, vol. 2663 (—), pp. 240—249

Zhang, D., Dong, Y., 2004. Semantic, Hierarchical, Online Clustering of Web Search Results. In *Proceedings of the 6th Asia Pacific Web Conference (APWEB*), Hangzhou, China