The 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
15-17 September 2011, Prague, Czech Republic

The Design and Validation of an Automatic Email Clustering System Based on Semantics

Alexandra Suzana Cernian, Iuliana Florea, Dorin Carstoiu, Valentin Sgarciu
University Politehnica of Bucharest/Automatic Control and Computer Science Faculty, Bucharest, Romania
alexandra.cernian@aii.pub.ro, iulia_flor3a@yahoo.com, cid@aii.pub.ro, vsgarciu@aii.pub.ro

Abstract—**Everybody has at least one email address nowadays. We keep in touch with family, friends, business partners by email. We receive news, notifications and advertisements by email. We subscribe to different newsletters that we also receive by email. But don't you hate it when your inbox gets all crowded with messages and you're having trouble finding exactly the one email you need? We all hate those moments. Common email clients allow us to sort emails by date, by sender or even by subject. But that is not always enough. This paper proposes the design and validation of an automatic system for clustering email messages based on semantic meaning. Our system is based on the clustering by compression technique, which has proven its capabilities during previous tests.**

Keywords—semantics; emails; clustering by compression; normalized compression distance; FScore

I. Introduction

Emails have become a basic way to communicate with others. A user sends and receives tens or hundreds of emails on a daily basis. Managing the emails we receive can become a time consuming task and can even get annoying if we have to do it manually. Let's assume you remember at some point, you received an email with a diet you would like to start, but you remember neither who sent it to you, nor when. The search would be much easier if you had a Diet folder, containing all related messages. Therefore, it would be useful to provide some automatic approach which places each email in a folder with other messages on the same topic.

The current work presents a novel approach to grouping the emails in an inbox based on the semantic content they share. Of course, there have been other attempts that had the same purpose, some of them with very good results. The novelty we propose is the use of the clustering by compression [1] in order to find the similarities in the content of the emails.

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 [1]. The formula of the NCD is based on the sizes of the compressed files, taken singly and concatenated 2 by 2. Afterward, the similarity matrix is interpreted using a clustering algorithm, which generates the clustering tree. The innovation of this method is the NCD.

We consider the following: C - a compressor (e.g ZIP), x, y - two files, xy – the concatenation of x and y C(x) - the size of the x compressed with C, C(y) - the size y compressed with C, C(xy) - the size of xy compressed with C. The NCD is defined as follows [1]:

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

There are three fundamental steps for using the clustering by compression technique: choosing a compression algorithm for compressing the dataset, using the NCD for creating the similarity matrix, and choosing a clustering algorithm to interpret the similarity matrix and generate the clusters.

II. System Architecture

Most email servers offer their users the possibility to automatically group the emails they receive based on the sender or on some criteria related to keywords found in the subject of messages. The system we propose in this work aims at clustering the emails in an inbox based on semantic criteria, by using both the subject and the body of the emails. Let's consider the following scenario: you love going on vacation and you have subscribed to the newsletters of several tourism agencies. Therefore, you would like all these messages to be placed in a dedicated folder, called Tourism or Vacations. You would know exactly where to go when you plan on finding your next destination.

Fig. 1 shows the architecture of our system. For the moment, the application can only connect to the Gmail server [2] and works for the English and Romanian languages. It retrieves the subject and the body of the emails from the Inbox or from another folder defined for a specific user account. The emails are saved on the local computer as text files, in a predefined location. The weight of the subject is increased 10 times in the text file. Afterward, the text documents are passed to the Clustering Engine.

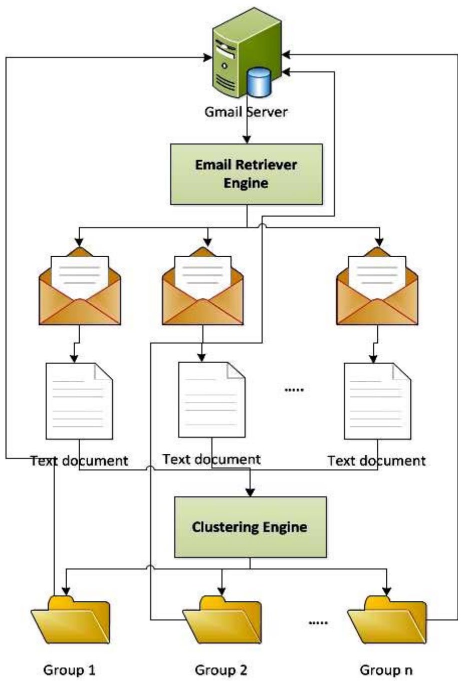

Figure 1. System architecture

The Clustering Engine is composed of the following elements:

- The text processing engine
- The BZIP2 compression algorithm [3]
- The UPGMA clustering algorithm [4]

The text processing engine is supposed to clean the text files, so that only the most representative parts remain. In every collection of documents there are words that recur very frequently. Words that occur in over 80% of the documents have very limited discrimination power in the clustering process. These words, such as *"the"*, *"for"*, *"of"* are called stop words [5] and are normally excluded by using dedicated algorithms. One such algorithm is also used in our application. The second text processing algorithm is a stemming algorithm [6] for the English or Romanian language, which replaces all the words with their stems. Moreover, the text processing engine incorporates a function which determines the frequency of the words, which will be further used to establish the label of each folder.

This new set of processed files will be subject to the clustering by compression procedure. The results of previous tests [7], [8] encouraged us to use BZIP2 as compression algorithm and UPGMA as clustering algorithm. The files are compressed and then the similarity matrix is computed using the NCD. Afterward, UPGMA generates the clusters, based on its interpretation of the distance matrix. For each group of files, a new folder is created on the local computer, in a predefined location. The word with the highest frequency in each group will determine the name of the corresponding folder. Afterward, the folders with the original version of the emails they contain are sent back to the Gmail inbox, in clustered format.

Fig. 2 presents the UML use case diagram [10] of the system. A UML use case diagram shows the main functionalities of a system.

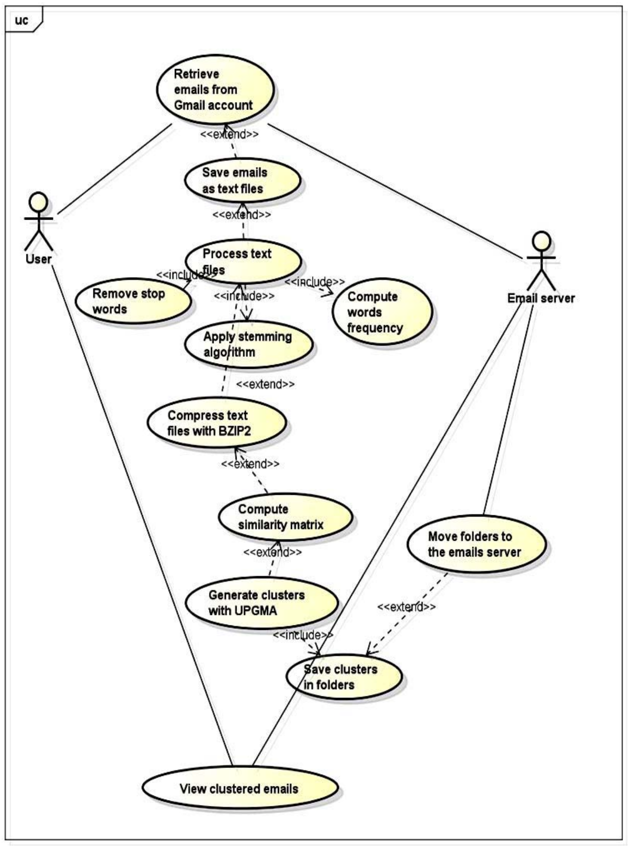

Figure 2. UML use case diagram

III. Experimental Results

So far, in order to validate the system proposed in this paper, we have used 2 datasets: a set of 50 emails written in Romanian and a set of 50 emails written in English. The datasets are made out of online magazine articles and the content of the emails is solely text. The articles in Romanian belong to the following categories: home decorations, diet, fashion, astrology, relationships, recipes and tourism. The articles in English belong to the following categories: celebrities, nature, fashion, relationships, technology and tourism. The purpose of the experimental validation is to see if the application is capable to detect these categories and to group the emails based on common content, placing them in the correct category.

In each case, we know what the right clustering would be. To assess the quality and robustness of the classification process, the FScore quality measure will be calculated [9]. If L_r is a particular predefined class of size n_r and S_i is a particular cluster of size n_i, then the FScore of this class and cluster is defined to be:

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

where n_{ri} documents in the cluster S_i belong to L_r, $R(L_r,S_i)=\frac{n_{ri}}{n_r}$ 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. 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 to this cluster make it in?". A perfect clustering solution produced by an application will be the one in which generated cluster has an identical match in the predefined solution. In this case, the FScore will be equal to 1. The higher the FScore value, the better the clustering solution is.

When clustering the newsletter items, the following cases are treated:

- The body of the emails is not processed **(a)**
- The body of the emails is processed by removing stopwords **(b)**
- The body of the emails is processed by removing stopwords and applying a stemming algorithm **(c)**
- The body of the emails is processed by removing stopwords, applying a stemming algorithm and increasing the weight of the keywords **(d)**

We will further refer to each of these scenarios as **(a), (b), (c)** and **(d)**.

The first round of experiments focused on the set of email written in Romanian. Table I presents the values of the FScore obtained in each of the 4 situation described above.

TABLE I. FSCORE FOR NEWSLETTER EMAILS IN ROMANIAN

	Unprocessed (a)	Stop words (b)	Stemming (c)	Complete (d)
FScore	0.70	0.71	0.69	0.93

When using the raw emails, the FScore is 0.70, which indicates an average clustering quality. The FScore is not much improved after removing the stopwords from the emails. It is only increased with 0.01, which is insignificant. The explanation resides in the fact that the Romanian language is complex, and the removal stopwords algorithm is not performant enough. In case (c), the result is even worse; the Fscore decreases to 0.69. This proves once again the complexity of the Romanian language, which makes it difficult to define accurate rules for eliminating suffixes and prefixes. The FScore increases significantly when augmenting the weight of the keywords in each email. The FScore value of 0.93 proves a good quality of the clustering results. 6 clusters were generated in this case, as we had expected.

The second round of experiments focused on the set of email written in English. Table II presents the values of the FScore obtained in each of the 4 situation described above.

TABLE II. FSCORE FOR NEWSLETTER EMAILS IN ENGLISH

	Unprocessed (a)	Stopwords (b)	Stemming (c)	Complete (d)
FScore	0.79	0.82	0.84	0.95

When using the raw emails, the FScore is 0.79, which indicates an average clustering quality. Although it is not a very high score, it is 0.09 higher than the score obtained for the emails written in Romanian, without applying any text processing techniques. After removing the stopwords, the FScore increases with 0.03, reaching a value of 0.82. This proves that stopwords removal algorithm for the English language works better than the one for the Romanian language, when the FScore only increased with 0.01. In case (c), the result is slightly improved; the Fscore reaches the value of 0.84. FScore increases significantly when augmenting the weight of the keywords in each email. The FScore value of 0.95 proves a good quality of the clustering results.

Fig. 3 presents a graphical comparison of the FScore values obtained for clustering newsletter items in Romanian and in English.

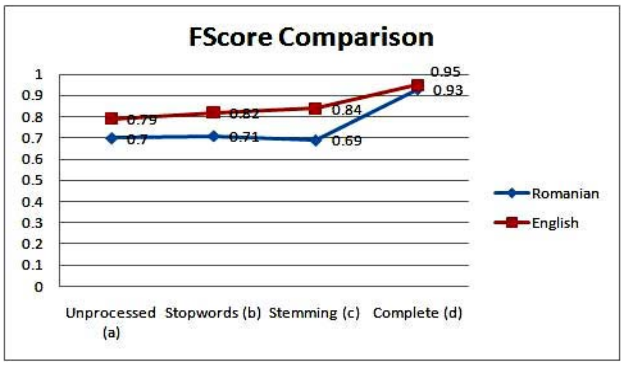

Figure 3. FScore comparison for clustering newsletter items

The FScore values for clustering the emails in English are constantly higher than those for clustering the emails in Romanian. In both cases (Romanian and English), the results are not significantly improved when applying the most well known text processing techniques: stopwords removal and stemming. The most important improvement is obtained when increasing the weight of several keywords used in each email.

To conclude, the results obtained for the emails in English were better than the results produced by our application when using the Romanian language. So, naturally, we started looking for an explanation. The most probable reason for this issue is the fact that both the algorithm for removing stop words and the stemming algorithm for the Romanian language have a poorer performance than those for the English language. Therefore, the text processing phase plays a very important role in the clustering process.

IV. Conclusion

This paper presented an integrated system for the automatic clustering of the email messages in a Gmail inbox. The application we propose integrates the clustering by compression technique and aims at detecting the semantic content of the emails. Therefore, the messages are grouped together based on common topics. For the moment, the application works for texts written in English or in Romanian and it has been tested using 2 sets of 50 emails, one in English, and the other one in Romanian. The FScore values obtained proved the potential of the clustering by compression to correctly interpret the informational content of the data. The FScore value for clustering the emails written in English was 0.96, which indicates an almost perfect solution, which leads to the conclusion that the semantic content embedded in the subject and the body of the emails was correctly extracted and used in the clustering process.

References

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

[2] Gmail Server: www.gmail.com .

[3] BZIP2 home page: http://bzip.org / ,last accessed on 14.01.2011.

[4] M. Murty A. Jain and P. Flyn, "Data Clustering: A Review," *ACM Computing Surveys*, 31(3), 1999.

[5] Stop words: http://en.wikipedia.org/wiki/Stop_words.

[6] Coveo Knowledge Base, "Unserstanding Stemming," http://www.coveo.com/en/Support/articles/Information%20-%20CES4-060330-3%20-%20Understanding%20Stemming.htm.

[7] A. Cernian, V. Sgarciu and D. Carstoiu, "Experimental Validation of the Clustering by Compression Technique," *Buletinul Stiintific UPB*, 2011 submitted for evaluation.

[8] A. Cernian, D. Carstoiu, V. Sgarciu, "On Using Metadata and Compression Algorithms to Cluster Heterogeneous Documents From a Semantic Point of View*," Proceedings of the ICSEA Conference*, 2010.

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

[10] A. D. Ionita, A. Cernian, *Notiuni Aplicate de Inginerie a Sistemelor de Programe*, Editura MATRIXROM, 2009.