Clustering Documents Using Tagging Communities and Semantic Proximity

Elisabete Cunha[1,2,3], Álvaro Figueira[1,4] and Óscar Mealha[2,5]
[1]CRACS&INESC TEC, [2]CETAC.MEDIA
[3]Instituto Politécnico de Viana do Castelo, [4]Universidade do Porto, [5]Universidade de Aveiro
Porto, Portugal
elisabetecunha@ese.ipvc.pt, arf@dcc.fc.up.pt, oem@ua.pt

Abstract—**Euclidean distance and cosine similarity are frequently used measures to implement the k-means clustering algorithm. The cosine similarity is widely used because of it´s independence from document length, allowing the identification of patterns, more specifically, two documents can be seen as identical if they share the same words but have different frequencies. However, during each clustering iteration new centroids are still computed following Euclidean distance. Based on a consideration of these two measures we propose the k-Communities clustering algorithm (k-C) which changes the computing of new centroids when using cosine similarity. It begins by selecting the seeds considering a network of tags where a community detection algorithm has been implemented. Each seed is the document which has the greater degree inside its community. The experimental results found through implementing external evaluation measures show that the k-C algorithm is more effective than both the k-means and k-means++. Besides, we implemented all the external evaluation measures, using both a manual and an automatic "Ground Truth", and the results show a great correlation which is a strong indicator that it is possible to perform tests with this kind of measures even if the dataset structure is unknown.**

Keywords-clustering, effectiveness, k-means, k-Communities, communitie detection, tagging, cosine similarity.

I. INTRODUCTION

The k-means algorithm [1] is one of the most popular partitional algorithms, considered one of the top 10 algorithms in data mining [2], mostly because of it's simplicity and efficiency [3-4]. However the results of the k-means algorithm depend largely on the initial seeds and on the number of clusters. Actually, depending on the chosen seeds, the resulting clusters may be different in each run. Many researchers have proposed alternatives in order to overcome this deficiency. David Arthur and Sergei Vassilvitskii proposed initializing the k-means algorithm using specific probabilities for the selected seeds, and they called to this method k-means++ [5]. Even though it improved the seed selection process, part of the problem persisted since the number of seeds remains unknown.

In order to overcome the generic k-means fault we propose a new algorithm called k-Communities, that will allow the choice of k specific seeds. For this, the collective intelligence that emerges from the users interactions, more specifically by tagging, will be used to see how the information must be related. Thus, the algorithm starts by implementing community detection on a network of tags. Moreover, taking into account the impact of similarity measures in detecting hidden patterns between documents we present a reflection on the measures Euclidean distance and cosine similarity, widely used to implement k-means algorithm. As a result of this reflection we implemented the new algorithm using cosine similarity and we present an alternative for computing the new centers in each iteration.

External clustering validation measures were implemented in 7 data sets in order to evaluate the efficacy of the k-Communities algorithm in comparison with the k-means or k-means++ algorithm.

To implement the external evaluation measures we use both a manual and an automatic structure. The automatic structure is obtained through an automatic "Ground Truth" algorithm [6], that combines the human classification given by tags with the information provided by the distance between documents.

II. K-MEANS AS INSPIRATION TO CREATE A NEW CLUSTERING ALGORITHM

A. *k-means algorithm*

The k-Means algorithm [1], partitions an initial set of documents into a set of clusters. Selecting k seeds the algorithm computes the distance from each document to each seed, grouping the documents which are closer to each seed. Next, the mean of the vectors in each cluster is computed, determining a new centroid and every document is once again associated to its nearest centroid. The process ends when convergence is achieved.

This algorithm is widely used because of its computational simplicity and efficiency [3-4]. The time complexity of each iteration is O(kn) but the number of iterations is usually very small.

B. *Reflection on similarity measures: Euclidean distance versus cosine similarity*

The model chosen to implement the k-means algorithm is Vector Space Model (VSM) considering each document as a vector in this space of words [7].

The method used to give weights to terms is the tfidf [8], given by (1), where N is the number of documents and n is the number of documents in which the term appears.

$$tfidf = tf \times Log\left(\frac{N}{n}\right) \qquad (1)$$

It allows combining the occurrence of a particular term in a document (tf) with the occurrence of this term in the whole collection and if a term appears with the same relevance in all documents of a collection it is no longer a relevant term to form clusters.

In a VSM a wide variety of distance functions and similarity measures have been used to compute the similarity between documents, such as Euclidean distance and cosine similarity. The default measure used to implement k-means algorithm is Euclidean distance. However the cosine similarity is also widely used.

In fact, Cosine similarity has one important propriety: it´s independence of document length. If we have two documents with exactly the same terms but with proportional frequencies, documents are treated as if they were the same document. For example: considering di = (2,0,1,0,3,0,0,1) and dj = (10,0,5,0,15,0,0,5) then dj=5di and the cosine similarity between this two documents will be 1, which means that the angle between this two documents is 0 and consequently they are seen as identical. This is especially true when a document corresponds to a summary of other, where several terms are common but with different frequencies. On the other hand, the Euclidean distance between di and dj is approximately 15.5, pointing out their differences.

As we can see, the k-means algorithm, using Euclidean distance (Fig.1) or cosine similarity (Fig. 2), originates different partitions.

Observe, in Fig. 1, that A and B are collinear with the origin of the referential, but were placed into different clusters. However, as shown in Fig. 2, using the cosine similarity, this two objects are placed on the same cluster because the angle between them is zero and consequently they are at the same distance to each centroid.

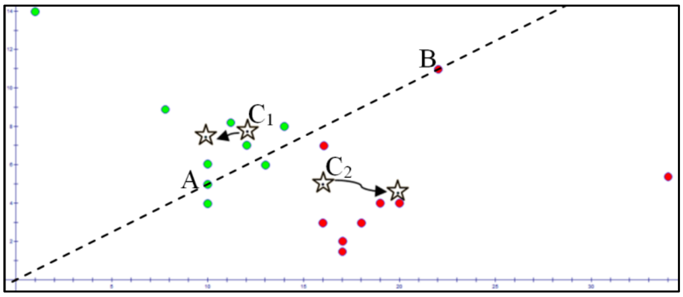


Figure 1. k-means algorithm using Euclidean distance

Nevertheless, the implementation of k-means using the cosine similarity is no guarantee of effectiveness. For example, C, D and E in Fig. 2, are closer to documents placed on cluster with seed C1 than to the documents of their own cluster.

When the cosine similarity is used to implement k-means algorithm the selection of new centroids will take into account the Euclidean distance, because the new center will be, on average, at the same distance (Euclidean distance) from all documents in the cluster. By consequence, the outliers influence the position of the new center and, as we can see in Fig.2, the angle between the new center and C, D and E, respectively, is greater comparatively with the initial seed C1.

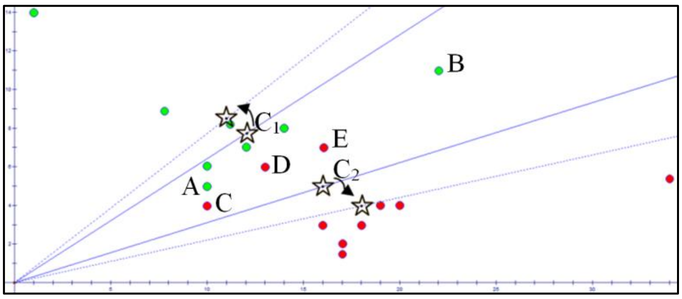


Figure 2. k-means algorithm using cosine similarity

It is easily perceived that the use of cosine similarity would be more consistent if the new center was the document more similar to the other documents inside its cluster.

C. *Community detection to select the initial seeds*

Clustering methods that are effective and efficient remain a challenge because the quality of the formed clusters isn’t always obtained throughout efficient clustering algorithms. k-means is well known for its efficiency but the partition quality depends on the initial seeds, because a random choice may result in a bad cluster optimization. In order to improve performance, many methods have been proposed. Among them is the one proposed by David Arthur and Sergei Vasilvitskii [5], where the seeds are selected with specific probabilities before k-means algorithm is run, providing an upgrade to the number of necessary iterations until convergence is achieved. This algorithm is called k-means++ and the time complexity is O(log k) [5].

However the number of partitions is still a problem. Taking into account that the clusters must satisfy people’s interests, we propose the analysis of tags associated to the documents by the users. Thus, community detection will be used to see how the information can be related, and consequently the number of partitions that reflect the collective intuition about the clustering structure.

The algorithm chosen to implement community detection was the one proposed by Girvan and Newman [9]. The steps of algorithm are:

LISTING I. COMMUTITY DETECTION ALGORITHM (GIRVAN-NEWMAN)

1. Compute the betweenness centrality of each edge.
2. The edge with largest centrality is removed. In case of a tie, randomly remove one of the edges.
3. The centralities of the new graph are recalculated; go to step 2.

This algorithm is historically important because it is a landmark on the field of community detection [10]. However, since it is not scalable for large datasets, we predict that in the future it will be possible to integrate other algorithms which are more efficient on large datasets. Once community detection is implemented, the number of communities (with more than one document) will be k, and the seeds will be the documents that have a greater degree within their community (random choice is used whenever there is a tie with other documents). Based on the k-means algorithm, we propose a new clustering algorithm, k-Communities.

D. *k-Communities clustering algorithm*

The k-Communities (k-C) algorithm starts with k seeds, each one coincident with the vectors of documents and new centroids are the documents that are more similar to other documents inside its cluster. Clearly, this has a price in terms of time complexity, since it is necessary to compare the distance between all documents inside each cluster in each iteration. The worst case is O(n^2). However, we expect our careful selection of initial seeds to keep the number of iterations very small.

Due to the complexity involved, the step 1 of the algorithm will only be recalculated when the system identifies the entry of new tags that justifies the recalculation. The steps of the algorithm are:

LISTING II. K-COMMUNITIES ALGORITHM

```
1. Select k seeds using community detection: each seed is the
   document that has greater degree inside it's community.
2. Compute the distance between each document and all seeds.
   (a) If the cosine similarity between a document and all centroids
       is zero then stop calculating, go to step 1 and add this document to
       the seeds set.
   (b) Elsif generates clustering by assigning each document to its
       closest seed.
       (i) If a document is closer to more than one seed, associates
           it to all seeds. To decide in which cluster it should stay,
           calculate the cosine similarity between this document and
           all the documents of the tie clusters and choose the cluster
           which has the most similar documents.
3. Compute the new centroid for each cluster, choosing the
   document that is more similar to the other documents in the
   cluster. Thus, the cosine similarity between each document and
   all documents of each cluster is computed, and the chosen
   document is the one who gets maximum sum as shown in
   Equation (2) (random choice if there is a tie between
   documents).
       $max \sum_{j=1}^{n} \cos(d_i, d_j)$        (2)
4. Go to step 2. The process ends when convergence is achieved,
   i.e., no more changes occur.
```

E. *Overlaping Communitty structure to improve text clustering*

Recently Cravino et al [11] have suggested a weighted cosine similarity proximity measure that takes into account the network of tags.

The user can set the degree to which the social aspect influences the grouping of documents, using a parameter called Social Slider, that assigns weights to tags. Additionally, related tags are identified through the overlapping community structure of the global network of tags. Each documents vector is constructed and the k-means clustering algorithm is implemented using cosine similarity as distance measure. The experimental results obtained by the authors do not identify significant improvements.

Using the same data set we want to assess if the k-C algorithm present better results in comparison with a method where the tags are integrated in the document according the suggested method.

III. EXTERNAL CLUSTERING VALIDATION MEASURES

Guaranteeing the effectiveness of a clustering algorithm is one of the most challenging issues. External criteria is one of the techniques used to evaluate the clustering results and involves comparing two structures: the one that resulted from the implementation of the algorithm and the one achieved from human intuition. Many external measures have been proposed [8], such as Purity, F measure and Rand Index which will be described in this section as well as how to obtain the "Ground Truth" when the structure of the data set is unknown.

A. *Obtaining the "Ground Truth"*

The implementation of external evaluation measures depends on external information, and it isn't always possible to manually organize documents in order to obtain the "Ground Truth" (especially if the data set is very large). Therefore, we also use a method called automatic "Ground Truth" [6]. This algorithm aims to find a structure that reflects the collective intuition of how the documents should be organized into clusters and integrate the information in the data, namely the similarity between documents, using the idea that, in general, each document and it's nearest document should belong to the same cluster.

The algorithm considers: (a) community detection implemented on an undirected network of documents, where the documents are nodes and edges are the connections between nodes that share at least one tag; (b) a directed network, where each document is connected to it's closest document; and (c) an algorithm to integrate steps (a) and (b). The communities are updated whenever the distance between two documents justifies the swap of community. Each pair is placed in one of the following three groups: 50% of pairs of closest documents; pairs of documents that are between 50% and 75% more closer or the 25% of more distant pairs.

B. *Measures*

The **Purity** measure [4] compares the "Ground Truth" classes with the clusters obtained through the clustering algorithm, selecting for each class the most similar cluster. The percentage of common documents is given by (3), where L={L1,L2,…,Lm} is the set of classes and C={C1,C2,…,Cm} is the set of clusters.

$$Purity(C,L) = \frac{1}{n}\sum_k max_j |C_k \cap L_j| \qquad (3)$$

Thus, Purity is always a number between 0 and 1. Clustering with a Purity close to 0 is a bad clustering and a Purity 1 corresponds to a perfect clustering [8].

F1 measure: the calculation of F1 measure is based on the pairs of documents in the collection. A collection of n documents has $n(n-1)/2$ pairs of documents and there are 4 possible connections between them as shown in the following contingency table.

TABLE I. CONTENGENCY TABLE

	Same Cluster	Different clusters
Same Class	True Positives (TP)	False Negatives (FP)
Different classes	False Positives (FP)	True Negatives (TN)

So the F_1 Measure [8] corresponds to the harmonic mean of Recall and Precision. **Precision** is the percentage of pairs of documents which are properly placed in the same cluster among the pairs of documents that are part of the cluster (4).

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

Recall is the percentage of pairs of documents which are properly placed in the same cluster among the pairs of documents that are or should be in the same cluster (5).

$$Recall = \frac{TP}{TP+FN} \quad (5)$$

Accordingly, F_1 is computed as shown in equation (6)

$$F_1 = \frac{2\times Recall\times Precision}{Precision+Recall} \quad (6)$$

Rand Index: Considering all pairs of documents in the collection, the Rand Index [8] measures the percentage of correct decisions, penalizing the false negatives and false positives as shown equation (7).

$$RI = \frac{TP+TN}{TP+TN+FN+FP} = \frac{2\times(TP+TN)}{n^2-n} \quad (7)$$

IV. EXPERIMENTAL RESULTS

In this paper, we present six case studies, which implement the k-means++ algorithm and the k-C algorithm.

We partitioned a dataset with 142 scientific papers in three data sets (D1, D2 and D3) collected from our personal library and from our University's Digital Library, which is hierarchically organized (we selected some Faculties and then collected papers from several areas). We then formed other three data sets (DA1, DA2 and DA3) which included only the abstracts of each paper. We used as tags the key words given by the authors.

Additionally, we used another data set (DClips) which has 124 news clips collected by users in the scope of another project, where each one can aggregate fragments of online news and associate tags. The k-means algorithm was implemented using only text or text+tags, through a weighted cosine similarity proximity measure. So, we intend to compare this results with the results of the k-C algorithm.

In each data set we can find documents from six classes, as we can see in Table II.

The manual "Ground Truth" and the automatic "Ground Truth" (described in Section 3.1) were both compared with the partitions generated by the clustering algorithms, using the external measures described in Section III.

TABLE II. MANUAL CLASSES OF EACH DATA SET

Data set	Classes
D_1 and DA_1	Clustering, Alpha Cronbach, Mathematics, History, Sport and Biology
D_2 and DA_2	Clustering, Cross Validation, Health, Sport, Biology and Mathematics
D_3 and DA_3	Clustering, usability, Health, Sport, Biology and Mathematics
D_{Clips}	Libya, US Tax, World Debt Crisis, Italy Downgrading, Greece and Other

A. Scientific Papers and Abstract Papers

In Table III we present the results of the average of the external measures, F_1, Precision, Recall, Rand Index and Purity for data sets D_1, D_2 and D_3.

TABLE III. AVERAGE OF THE EXTERNAL MEASURES FOR DATA SETS D1, D2 AND D3, USING AUTOMATIC CLASSES (AC) AND MANUAL CLASSES (MC)

External Measures	**Clustering Algorithm**			
	k-means++		**k-Communities (k-C)**	
	AC	**MC**	**AC**	**MC**
F_1	0.59	0.53	**0.78**	**0.80**
Precision	0.53	0.46	**0.88**	**0.88**
Recall	0.67	0.64	**0.70**	**0.73**
Rand Index	0.83	0.82	**0.93**	**0.94**
Purity	0.82	0.78	**0.85**	**0.83**

In average the results of the implementation of k-C algorithm are better than the results of k-means++ algorithm. This occurs both in comparison with the classes obtained manually (MC) or automatically (AC) (using the automatic "Ground Truth" algorithm).

The average Recall value indicates that the k-C algorithm has a lowest number of False Negatives, in other words, there are in average a lower number of pairs that belong to different clusters and that should be part of the same cluster.

Looking at the contents of each cluster we can see that the k-C algorithm provides the best results for the average Precision value, i.e., there are more pairs of documents that are properly associated in the same cluster. There is an improvement of over 35%, comparing with k-means++.

We can also observe that on average there is an increase of approximately 10% correct decisions (Rand Index), i.e. True Positives and True Negatives, when using the k-C algorithm. Finally, the average Purity value also shows an increase when using the k-C algorithm, showing that this clusters are more similar to the manual clusters or automatic clusters obtained through automatic "Ground Truth" algorithm.

Analyzing the results obtained for the abstracts of papers, Table IV, when using k-means++, the results are a lot worse when compared to those obtained for full text paper (Table III). Less information appears to have influence in the clustering. However, the new algorithm k-C shows consistent results when using abstracts and full papers. Looking at Table IV we can still see that the best results are obtained by k-C algorithm for all data sets and the results are very similar to those obtained when using full text, indicating that the new algorithm is more stable.

TABLE IV. AVERAGE OF THE EXTERNAL MEASURES FOR DATA SETS DA1, DA2 AND DA3, USING AUTOMATIC CLASSES (AC) AND MANUAL CLASSES (MC)

External Measures	Clustering Algorithm			
	k-means++		k-Communities (k-C)	
	AC	MC	AC	MC
F$_1$	0.42	0.38	**0.75**	**0.70**
Precision	0.37	0.34	**0.85**	**0.72**
Recall	0.49	0.51	**0.70**	**0.70**
Rand Index	0.75	0.76	**0.92**	**0.89**
Purity	0.70	0.66	**0.81**	**0.82**

B. *Clips*

Using the results obtained by Cravino et al [11] in a study based on a small data set (DClips) of news clips, we intend to compare them with the k-C algorithm results.

As we can see in Table V, k-C algorithm presents best results in comparison with k-means for text or for text+tags (even though they are generally lower to those obtained in the others data sets).

TABLE V. EXTERNAL MEASURES FOR DATA SET DCLIPS, USING AUTOMATIC CLASSES (AC) AND MANUAL CLASSES (MC)

External Measures	Clustering Algorithm (k-C)			
	k-means		k-Communities	
	Text	Text + Tags	AC	MC
F$_1$	0.27	0.26	0.47	0.49
Precision	0.23	0.24	0.54	0.56
Recall	0.32	0.30	0.42	0.44
Rand Index	0.66	0.67	0.81	0.82

As we can see in Fig. 3 (A) the generated communities show that the user has not seen relationship between the clips that are in different communities. However, as shown Fig. 3 (B), when we merge the graph of tags with the distance graph, where each document is linked to it´s closest document, we find that the nearest document to a document isn't always in the same community. Moreover, taking into account the thickness of the inter-communities connections there are very similar clips placed in different communities.

These results only reflect the similarity between two groupings, leading to the conclusion that the clips were arranged in a different way from the "Ground Truth" proposed by the user.

It is only natural that a user can't see all the relationships between the clips, but the fact of not seeing them does not mean they do not exist. Hence, if more users attribute tags, more connections will be discovered and thus the construction of the "Ground Truth" will be based on collective intelligence.

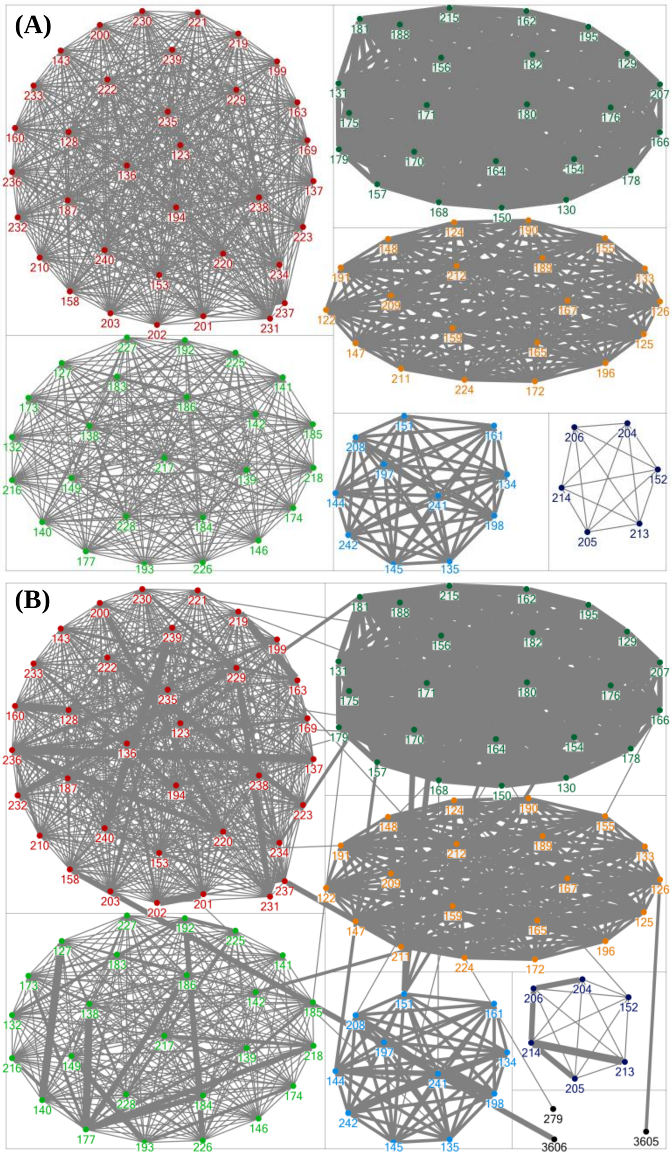


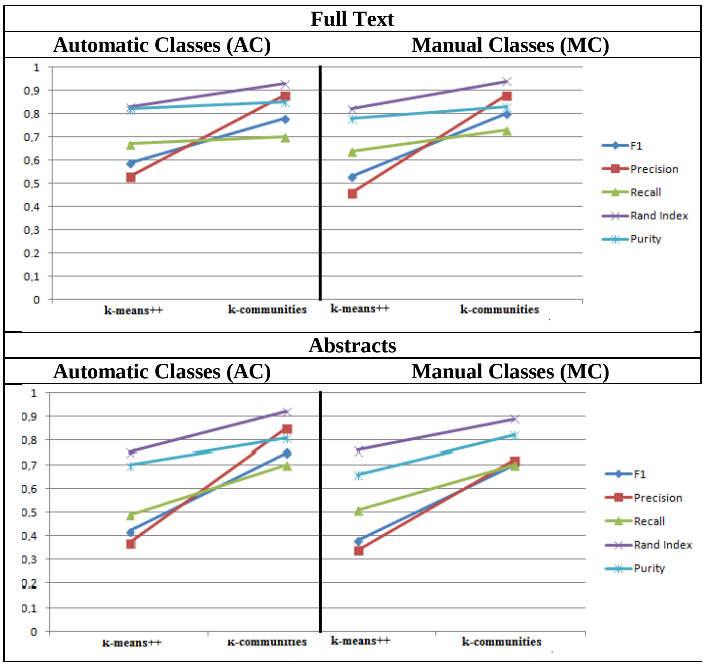
Figure 3. (A) Communities of tags graph (B) Merge of tags graph and distance graph

C. *"Ground Truth"*

By observing Table VI, using the results of the full papers and their abstracts, we can visually see that there is a strong correlation between the results obtained when comparing the results of clustering algorithms with manually organized groups and the groups that were automatically determined by the implementation of "Ground Truth" algorithm. In other words, the graphics on the right column are very similar to those on the left column.

For the data set Dclips, Automatic Classes (AC) have been used only in the k-C algorithm and we found that the results are very similar to those obtained for Manual Classes (MC), as seen on Table V. These results indicate that it is acceptable to use the “Ground Truth” algorithm to determine the structure of the data set when it is unknown.

TABLE VI. VISUAL CORRELATION RESULTS BETWEEN AUTOMATIC CLASSES (AC) AND MANUAL CLASSES (MC) FOR FULL AND ABSTRACTS PAPERS



V. CONCLUSION

In this paper, we proposed a new clustering algorithm called k-C algorithm to implement in a tagging system. The proposed clustering method is based on the k-means algorithm and the k initial seeds are, according to tags associated with each document, the center of each community detected in the tags graph.

Using the manual “Ground Truth” and the automatic one, we compared the results of the external measures of the k-means++ algorithm with the results of the external measures of the k-C algorithm. This comparison shows that in average, the k-C algorithm creates clustering which are closer to both manual and automatic "Ground Truth".

Concerning the algorithms performance, further work tests in larger datasets are required. These results will provide the information needed to determine the k-Communities application contexts.

ACKNOWLEDGMENT

This work is funded by the Portuguese Government through FCT — Fundação para a Ciência e a Tecnologia. Reference: SFRH/BD/46616/2008.

REFERENCES

[1] J.B. MacQueen, Some Methods for Classification and Analysis of MultiVariate, in Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability. 1967, University of California Press. p. 281-297.

[2] X. Wu, et al., Top 10 algorithms in data mining. Knowl. Inf. Syst., 2007. 14(1): p. 1-37.

[3] S. Theodoridis, and K. Koutroumbas, Pattern Recognition, Fourth Edition. Fourth Edition ed. 2009: Academic Press. 961.

[4] R. Feldman, and J. Sanger, The Text Mining Handbook Advanced Approaches in Analyzing Unstructured Data. 1.ª ed. 2007: Cambridge University Press. 410.

[5] D. Arthur, and S. Vassilvitskii, k-means++: the advantages of careful seeding, in Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. 2007, Society for Industrial and Applied Mathematics: New Orleans, Louisiana. p. 1027-1035.

[6] E. Cunha, and A. Figueira. Automatic Clustering Assessment through a Social Tagging System. in 2012 IEEE 15th International Conference on Computational Science and Engineering. 2012. Paphos, Cyprus.

[7] G. Salton, A. Wong, and C.S. Yang, A vector space model for automatic indexing. Commun. ACM, 1975. 18(11): p. 613-620.

[8] C. Manning, P. Raghavan, and H. Schütze, An Introduction to Information Retrieval. 2009: Cambridge University Press. Cambridge, England.

[9] M. Girvan, and M.E.J. Newman, Community structure in social and biological networks. Proceedings of the National Academy of Science, 2002. Nr. 12: p. pp 7821–7826.

[10] S. Fortunato, and C. Castellano, Community Structure in Graphs, in Encyclopedia of Complexity and Systems Science. 2009. p. 1141-1163.

[11] N. Cravino, J. Devezas, and Á. Figueira. Using the Overlapping Community Structure of a Network of Tags to Improve Text Clustering. in In Proceedings of the 23rd ACM Conference on Hypertext and Social Media (HT 2012). 2012. Milwaukee, WI, USA.