Local feature selection in text clustering

Marcelo N. Ribeiro[1], Manoel J. R. Neto[2], and Ricardo B. C. Prudêncio[1]

[1] Centro de Informática, Universidade Federal de Pernambuco, Recife – PE – Brazil
[2] Instituto de Computação, Universidade Federal de Alagoas, Maceió – AL – Brazil

Abstract. Feature selection has improved the performance of text clustering. Global feature selection tries to identify a single subset of features which are relevant to all clusters. However, the clustering process might be improved by considering different subsets of features for locally describing each cluster. In this work, we introduce the method ZOOM-IN to perform local feature selection for partitional hierarchical clustering of text collections. The proposed method explores the diversity of clusters generated by the hierarchical algorithm, selecting a variable number of features according to the size of the clusters. Experiments were conducted on Reuters collection, by evaluating the bisecting K-means algorithm with both global and local approaches to feature selection. The results of the experiments showed an improvement in clustering performance with the use of the proposed local method.

1 Introduction

Clustering algorithms have been applied to support the information access in large collections of textual documents [7]. Such techniques may organize similar documents in clusters (groups) associated to different levels of specificity and different contexts. Text clustering has been applied to improve precision and recall of information retrieval systems as well as to provide interfaces for navigation between documents [7]. In [9], for instance, clustering algorithms are used to organize the results of user's queries to a search engine. The structure of clusters, properly labeled, offers a vision of what types of questions can be answered by the query results.

In order to accomplish the text clustering process, the documents are represented, in most cases, as a set of *terms of indexing* associated to numerical weights. Considering all existing terms in a collection brings some difficulties to the clustering algorithm. In fact, when the size of the feature space is very high, the distance between similar points is no very different than the distance between more distant points (i.e.,"curse of dimensionality") [8].

Considering the above context, text clustering usually contains a phase of dimensionality reduction of the vectors that represent the documents. Features in the reduced space may correspond to a subset of the original features (as performed by feature selection methods [3]), or they may be created by combining the original features (as performed by feature extraction methods [8]). In text clustering, feature extraction presents a disadvantage compared to feature

selection since each new feature is no longer associated with an existing term or word, which makes the formed clusters less comprehensive [8].

Feature selection can be classified as either global or local [1]. The global approach aims to select a single subset of features which are relevant to *all* derived clusters [5]. Despite the large use of global methods in literature, depending on the problem, it is possible that there are several different subsets of features that show good clusters. In order to overcome this limitation, local feature selection, in turn, tries to identify different subsets of features associated to each formed cluster. Although recent work has obtained good empirical results by evaluating local feature selection on benchmarking Machine Learning data (see [5]), there is no investigation of the use of local feature selection for text clustering.

In this work, we proposed the ZOOM-IN, a local feature selection method for partitional hierarchical clustering [10]. In this method, all the documents are initially allocated to a single top-level cluster which is recursively divided into small sub-clusters. At each division step, a feature selection criterion is applied to choose the features which are more relevant only considering the cluster being divided. The number of selected features is defined according to the cluster size. The result of our method is a hierarchy of clusters in which each cluster is represented by a different subset of features.

Experiments were performed on the Reuters collection [4], comparing the bisecting K-means algorithm (a partitional hierarchical algorithm) [7], with both the global and local feature selection approaches. The results revealed an improvement in precision when the local approach was compared to the global approach. The ZOOM-IN method eliminated irrelevant terms, at same time maintaining the quantity of information required for each division of the clusters.

Section 2 brings a brief introduction to text clustering. Section 3 presents feature selection approaches applied to text clustering, followed by Section 4 which presents the proposed method. Section 5 brings the experiments and results. Finally, Section 6 presents some final considerations and future work.

2 Text clustering

Text clustering is the process of grouping similar documents into clusters, in order to better discriminate documents belonging to different categories. A document in text clustering is described by a set of keywords, so-called *terms of indexing*, which is a vocabulary extracted in the collection of texts. A weight is associated to each term, defining an array of terms that represents the document. The term weights are commonly computed by deploying the Vector Space Model with the $TF\text{-}IDF$ weighting schema. In this model, each term weight $tfidf_j$ in the document d_j is given by:

$$tfidf_j = tf_j \log \frac{n}{DF_t} \tag{1}$$

where n is the number of documents in the collection, DF_t is the number of documents in the corpus where the term t occurs and tf_j is the frequency of

the term t in the document d_j. The proximity between two document vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$, represented in this model, is usually defined by the *cosine* measure as:

$$\cos(\boldsymbol{d}_1, \boldsymbol{d}_2) = \frac{\boldsymbol{d}_1 \cdot \boldsymbol{d}_2}{\|\boldsymbol{d}_1\| \times \|\boldsymbol{d}_2\|} \tag{2}$$

A hierarchical clustering algorithms groups the data in a hierarchy of clusters. For text clustering, the hierarchical solution has more advantages regarding a flat approach, since it divides the collection of documents on various levels of granularity and specificity, providing a better view of the collection [10].

The hierarchical clustering algorithms may be further categorized into agglomerative or partitional. Agglomerative clustering is a bottom-up approach which starts affecting each document to a distinct cluster and progressively joins similar clusters. Partitional clustering, in turn, is a top-down approach which starts with all documents in a single cluster and progressively divides the existing clusters. According to [10], in the agglomerative clustering, wrong decisions of combining clusters at the beginning of the algorithm execution tend to multiply errors as the clustering is executed. Partitional algorithms, in turn, have a more global vision of possible cohesive clusters, and hence, they will be the focus of our work. A widespread partitional algorithm is the bisecting K-means [7], in which the simple K-means algorithm is used to bisect the clusters (i.e., dividing each cluster in two sub-clusters) at each division step. The bisecting K-means has shown to be very competitive compared to agglomerative algorithms [7].

3 Feature selection for text clustering

Feature selection for text clustering is the task of disregarding irrelevant and redundant terms in the vectors that represent the documents, aiming to find the smallest subset of terms that reveals "natural" clusters of documents [3]. Searching for small subset of relevant terms will speed up the clustering process, while avoiding the curse of dimensionality. The methods commonly used to select features in text clustering deploy statistical properties of the data as a criterion to determine the quality of the terms [2, 8] (see Section 3.1). The selection is made with the use of a threshold or a fixed number of desired features.

3.1 Criteria for ranking features

In this section, we cited some criteria which will be later applied in our experiments:

Document Frequency (DF). The value DF_t of the term t is defined as the number of documents in which the term t occurs at least once in the collection of documents.

Term Frequency Variance (TfV). Let tf_j be the frequency of term t in the document d_j. The quality of term t is defined in the TfV method as:

$$TfV_t = \sum_{j}^{n} tf_j^2 - \frac{1}{n}\left[\sum_{j}^{n} tf_j\right]^2 \tag{3}$$

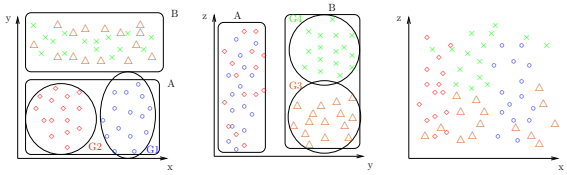
where n is the number of documents in the collection. In the experiments performed in [2], the TfV method has shown to maintaining the precision of the clustering process with up to 15% of the total number of features.

Mean of TF_IDF (TI). In [8], the quality of a term t is defined as the mean value of $tfidf_j$ across all documents ($j = 1, \ldots, n$) in the collection. The TI method has shown a performance superior to DF and similar to TfV [8].

3.2 Global and local feature selection

Feature selection for clustering may occur on either the global or the local approach. The global feature selection chooses the relevant features once by deploying a pre-defined ranking criterion, and uses the same subset of features in the whole clustering process. Global selection is the most investigated approach in the literature [1, 3, 8]. In local feature selection, a subset of features is chosen for each cluster. It assumes that the clusters may be better discriminated from each other by considering a different subset of features for each cluster.

Figure 1 illustrates a set of objects belonging to four clusters, which are described by the features x, y and z. The clusters G1 and G2 are only revealed when the attributes x and y are considered, i.e., the attribute z is irrelevant to distinguish between G1 and G2 (see Figure 1(a)). Figure 1(b), in turn, illustrates that features y and z are relevant to identify the clusters G3 and G4, i.e., feature x is irrelevant in this context. Finally, the attributes x and z corresponds to a irrelevant subset of features (Figure 1(c)). In such situation, any subset of features eventually returned by a global method would not be able to identify the four existing clusters. It is necessary to examine a feature in the context of different subsets before stating that the feature is actually irrelevant [3].



(a) in x and y (b) in y and z (c) in x and z

Fig. 1. Data of the clusters G1, G2, G3 and G4 for different features.

Compared to the global approach, there is few relevant work in the literature of clustering that investigated the local feature selection. In [5], for instance, the authors proposed a local feature selection method for clustering, by searching several subsets of features that show different clusters, and choosing the

most cohesive clustering based on a criterion of cluster evaluation. In [5], experiments were performed to evaluate the proposed local method for the K-means algorithm. By using the local method, the authors obtained an improvement in precision for clustering benchmarking problems from the UCI repository. We highlight here that, to the best of our knowledge, the local approach for feature selection has not been applied in any work for text clustering.

4 Proposed method

In this work, it is proposed an algorithm to perform partitional hierarchical clustering with local feature selection. In our proposal, it is expected that the privileged global vision of partitional algorithms can take advantage of a local vision offered by the local feature selection. We also expected that the variety of subsets of features selected to each division of the clusters might reveal hidden clusters in the data. The proposed algorithm for local feature selection using the bisecting K-means follows the steps:

1. Choose a cluster to divide, considering an initial cluster containing all the documents;
2. Select features for the chosen cluster by deploying a ranking criterion (as cited in Section 3.1). The features may be filter based on a pre-defined number of N required terms or based on a threshold τ on the ranking criterion;
3. Build 2 sub-clusters using the K-means algorithm;
4. Repeat steps 2 and 3 by $ITER$ times;
5. Repeat steps 1, 2, 3 and 4 until the required number of clusters is reached.

The problem in Figure 1 can be initially solved by selecting the subset of features (e.g., x and y) that best reveals clusters in the data (e.g., Figure 1(a)). Following, the algorithm generates sub-clusters to both clusters A (the data of G1 plus G2) and B (the data of G3 plus G4). By performing a new feature selection to each cluster, the cluster A remains with the features x and y and cluster B with the features y and z. The cluster A can now be broken into G1 and G2 (children of the cluster A). Cluster B, in turn, can be broken into G3 and G4 (children of the cluster B), thus revealing all clusters for these data.

An important aspect to be considered in our algorithm is the number N of terms to be selected for each cluster. As the algorithm is executed, the generated clusters become smaller, and hence, the number of distinct terms in documents also decreases. Thus, the choice of a large constant number N tends to minor the potential of the selection procedure since the number of selected terms will be similar to the number of distinct terms. The choice of a small constant number N, on other hand, will cause a lost of information when the clusters are large.

A solution to the above trade-off is to use a variable number of terms according to the size of the clusters and the number of distinct terms. For simplicity, in our work, it is proposed to choose the number of terms n_i for the cluster i as:

$$n_i = \left\lfloor \frac{N_T}{N_C} \cdot m_i \right\rfloor \tag{4}$$

where N_T is the number of different terms in the collection of documents, N_C is the size of the collection of documents and m_i is the size of the cluster i. N_T/N_C is the proportion of different terms revealed in each document of the collection. This procedure reduces the number of terms locally selected in each division of cluster. Because this reminds the setting of a binocular, this method is referred in this work as *ZOOM-IN* method. As it will be seen, we performed experiments with both the local feature selection with constant number of features and the ZOOM-IN to decide the number of selection terms per each iteration.

5 Experiments and Results

Section 5.1 describes the experiments performed to evaluate the viability of the proposed method. Section 5.2, in turn, presents the obtained results.

5.1 Experiments Description

In our experiments, we used a subset of documents in the Reuters-21578 collection [4] which were assigned to a single class (representing a total number of 1228 documents associated to 42 classes). The collected documents were initially processed in order to remove stopwords (prepositions and common words). We also applied the stemming operator with the Porter's algorithm.

The clustering algorithms were evaluated by deploying the *micro-averaged precision* measure, also used, for instance, in [6]. The micro-averaged precision assumes that each cluster formed by the clustering algorithm has a majority representative class c. Considering T the set of clusters and C the set of classes, the micro-averaged precision is given by [6]:

$$P(T) = \frac{\sum_{c \in C} \alpha(c, T)}{\sum_{c \in C} \alpha(c, T) + \beta(c, T)} \tag{5}$$

where $\alpha(c, T)$ is the number of documents correctly affected to c and $\beta(c, T)$ is the number of documents incorrectly affected to c.

For evaluating the hierarchy generated by the clustering algorithms, the clusters considered for computing the precision were those present in the leaves of the produced dendogram and the number of clusters specified for execution of the algorithms was equal to the number of classes of the collection. Finally, the micro-average precision was averaged over 30 different runs of the evaluated algorithms.

5.2 Results and discussion

Figure 2 presents the precision obtained for each evaluated algorithm (bisecting K-means with both the global and local feature selection), with constant number of selected features. As it may be seen, for all global methods, when are selected few terms the performance of the global methods falls. The criterion of ranking

that obtained best precision rates is the TfV, with performance similar to TI and better than the DF, as already observed in the work [8]. It is concluded that for a few number of selected terms, there is little information on the documents, which deteriorates the precision of the clustering.

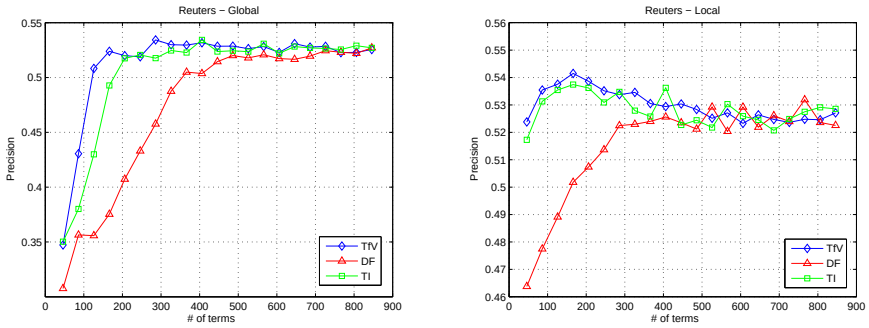


Fig. 2. Micro-averaged precision in relation to the number of terms used for the Reuters collection, with local and global feature selection.

The local approach, in turn, keeps the precision even with a very small quantity of terms, excepting for the method DF. Interestingly, the precision obtained for fewer selected terms are even better than the precision obtained when a large number of features is selected. This is due to the fact that for large values of the number of selected terms, the selective potential decreases locally (as discussed in Section 4). With a small amount of selected terms, the selective potential is kept during the divisions of the clusters, and the precision is improved.

Table 1. Micro-averaged precision with the ZOOM-IN method.

Collection	**Without method**	**TfV**	**DF**	**TI**
Reuters	0.527117	0.540988	0.527362	0.541395

However, a small amount of selected terms may undermine the amount of information needed at the beginning of the execution of the clustering, when the clusters are still large and the number of distinct terms as well. It is necessary to select the terms that reflect a real benefit to the clustering. In this context, we also performed an experiment using a variable amount of locally selected terms, which is called the ZOOM-IN method (as proposed in Section 4). The values of precision obtained by ZOOM-IN were even better that the results obtained by the local method with few features (see Table 1).

6 Conclusions

In this work, it was proposed the use of a local feature selection approach for partitional hierarchical text clustering. Each cluster derived by the proposed method is represented by a different subset of features. In the performed experiments, the local approach was compared to the global feature selection approach for the bisecting K-means. It was observed that the local approach obtained good precision even for few selected terms. We also performed experiments by using the ZOOM-IN method to automatically define the number of selected features in each iteration of the partitional algorithm. The results obtained by the ZOOM-IN were satisfactory, because it proved the need for feature selection in text clustering and showed the benefits in select features locally.

As future work, we intend to evaluate other criteria for ranking features, which use information from the similarity between documents, such as the ranking based on entropy [1]. Finally, the proposed method will be evaluated on other collections of documents.

References

1. M. Dash, K. Choi, P. Scheuermann, and H. Liu. Feature selection for clustering - a filter solution. In *Proceedings of the 2002 IEEE International Conference on Data Mining*, pages 115–122, 2002.
2. I. Dhillon, J. Kogan, and C. Nicholas. Feature selection and document clustering. In M. W. Berry, editor, *Survey of Text Mining*, pages 73–100. Springer, 2003.
3. J. G. Dy and C. E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, 2004.
4. D. D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. `http://www.daviddlewis.com`, 1999.
5. Y. Li, M. Dong, and J. Hua. Localized feature selection for clustering. *Pattern Recognition Letters*, 29(1):10–18, 2008.
6. N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th International ACM SIGIR Conference*, pages 129–136, 2002.
7. M. Steinbach, G. Karypis, and V. Kumar. A comparison of document clustering techniques. Technical Report, Department of Computer Science and Engineering, University of Minnesota, 2000.
8. B. Tang, M. Shepherd, E. Milios, and M. I. Heywood. Comparing and combining dimension reduction techniques for efficient text clustering. In *International Workshop on Feature Selection for Data Mining*, 2005.
9. O. Zamir, O. Etzioni, O. Madani, and R. M. Karp. Fast and intuitive clustering of web documents. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pages 287–290, 1997.
10. Y. Zhao and G. Karypis. Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 515–524, 2002.