Unsupervised Clustering of Spontaneous Speech Documents

Edgar Gonzàlez, Jordi Turmo

TALP Research Center
Universitat Politècnica de Catalunya, Barcelona, Spain
{egonzalez,turmo}@lsi.upc.edu

Abstract

This paper presents an unsupervised method for clustering spontaneous speech documents. The approach uses a hierarchical algorithm to automatically determine the number of clusters and a starting model for a subsequent iterative algorithm. We have evaluated this method on the Switchboard corpus and compared it to a set of supervised and other unsupervised methods. The results show that our method significantly outperforms the rest of the approaches.

1. Introduction

Every day a huge amount of speech data is produced. The grouping of documents coming from speeches, presentations or meetings according to their similarity can be helpful to a human user who needs to browse collections of these data, as well as a previous step for further automatic processing. In this context, document clustering techniques can be applied.

From the whole range of clustering techniques, iterative refinement algorithms are extremely popular due to their good performance, relative simplicity and good theoretical foundations. Some examples of the use of iterative refinement algorithms for speech document clustering are [1] and [2]. The first one applies a hierarchical tree clustering method to a collection of 100 documents in 10 topics from the Switchboard corpus, using both manual and automatic transcripts. The second one measures the degradation in performance experimented by a modification of the K-means algorithm when porting from manual to automatic transcripts, carrying an evaluation on the TDT2 corpus of broadcast news.

An example of non-iterative refinement algorithm is presented in [3]. It uses the Non-negative Matrix Factorization method and applies it to subsets of up to 10 categories from the TDT2 corpus.

However, all these methods are supervised. They require the number of clusters to be known *a priori*, and some of them need an initial model, to which they can be sensitive.

This paper proposes an unsupervised approach which automatically detects the number of clusters and gives a good starting model to iterative algorithms. The rest of the paper is organized as follows: section 2 presents our method, section 3 describes the framework of the evaluation used for our experiments. The set of experiments and the results are included in section 4. Lastly, section 5 concludes the paper.

2. Our Unsupervised Clustering Approach

An overview of our method is depicted in figure 1. It first uses a hierarchical clustering method to obtain a dendrogram, or hierarchical tree representation of the subsumption of clusters in the collection [4], and then it tries to find a set of clusters in this tree that can be used as initial model for an iterative algorithm. A detailed description can be found in [5]. The following sections briefly introduce every step in the process.

2.1. Hierarchical Clustering

In the first step, a hierarchical algorithm is used to generate a complete dendrogram. In this process, there is no need to chose either an initial model or a number of clusters. Concretely, we use Hierarchical Agglomerative Clustering (HAC), a simple algorithm which has been reported to show good performance on real-world collections [4]. HAC works in a bottom-up fashion: it starts placing every document in a cluster by itself, and then it repeatedly merges the two closest clusters. The concept of closest depends on the distance function used. Our previous experiments, as well as published evaluations of HAC [4], pointed to group average distance (UPGMA) as the most suitable distance in HAC context. UPGMA distance between clusters c_i and c_j is defined as the average of pairwise distances between documents d_r and d_s from each cluster:

$$dist_{UPGMA}(c_i, c_j) = \frac{1}{n_i n_j} \sum_{d_r \in c_i} \sum_{d_s \in c_j} dist(d_r, d_s) \quad (1)$$

where n_i and n_j are the sizes of clusters c_i and c_j, respectively.

2.2. Generation of Initial Model Candidates

The clusters of the resulting dendrogram are scored and ranked according to a quality function of the clusters they represent, and then the maximal set of top scored clusters that covers a quantity of documents of less than or equal to a fraction γ of the collection is selected. These clusters are filtered to remove those that are included inside other higher-ranked clusters. The final set is what we call an initial model candidate.

The cluster quality function is crucial to the algorithm. To find a proper function we observed what properties should *good* clusters have:

Minimum within distance: The distances between the documents in a cluster should be small if they belong to the same category. $W(c_i)$ of cluster c_i is defined as the average of the pairwise distances within documents from the cluster:

$$W(c_i) = \frac{1}{n_i(n_i - 1)} \sum_{d_r \in c_i} \sum_{d_s \in c_i, s \neq r} dist(d_r, d_s) \quad (2)$$

Maximum between distance: The distances between the documents in different clusters should be large if they belong to different categories. $B(c_i)$ of cluster c_i is defined as the average

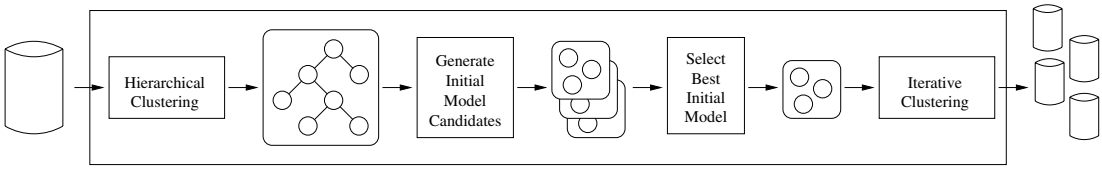

Figure 1: *The Hybrid clustering procedure*

Name	Formula	Name	Formula
W	$1/W$	GW	$1/GW$
WB	B/W	GWB	B/GW
WN	N/W	GWN	N/GW

Table 1: Quality functions used by the algorithm

of the pairwise distances between documents from the cluster and documents from the other clusters:

$$B(c_i) = \frac{1}{n_i(n-n_i)} \sum_{d_r \in c_i} \sum_{d_s \notin c_i} dist(d_r, d_s) \quad (3)$$

Maximum between distance in the neighborhood: The previous measure has the disadvantage that in certain circumstances the whole collection mass can introduce *noise* to the value. That is why a between distance in the cluster vicinity is useful. Using the dendogram representation, we approximate the neighborhood of a cluster by its sibling. Hence, $N(c_i)$ of cluster c_i is defined as the UPGMA distance between c_i and its sibling:

$$N(c_i) = dist_{UPGMA}(c_i, sibling(c_i)) \quad (4)$$

Minimum cluster growth: Lastly, to cope with clusters with different densities, the property of cluster *growth*, $G(c_i)$, is proposed, intuitively defined as the cluster expansion at the last dendrogram join, relative to the internal density of its two children, c_{i1} and c_{i2}:

$$w\_sum(c_i) = \sum_{d_r \in c_i} \sum_{d_s \in c_i, s \neq r} dist(d_r, d_s) \quad (5)$$

$$w\_children(c_i) = \frac{w\_sum(c_{i1}) + w\_sum(c_{i2})}{n_{i1}(n_{i1}-1) + n_{i2}(n_{i2}-1)} \quad (6)$$

$$G(c_i) = \frac{dist_{UPGMA}(c_{i1}, c_{i2})}{w\_children(c_i)} \quad (7)$$

Taking into account these four properties, we have considered six quality functions which clusters should try to maximize, and which are listed in table 1.

2.3. Selection of the Best Initial Model

To avoid manual selection of the quality function to use and of the aforementioned $gamma$ parameter, the initial model candidates for all quality functions and values of γ from 0.05 to 0.95, in steps of 0.05, are obtained and evaluated using the Calinski and Harabasz C score [6]. The C score is a normalized ratio of the distances between clusters and the distances within every cluster. We have used this score function taking into account that the Calinski and Harabasz method outperformed other approaches in a previous evaluation of several unsupervised approaches for the evaluation of the quality of cluster models [7].

The candidate with the best C score will be chosen. However, as the method has a preference for higher model dimensions, it is not the global maximum that is chosen. For each quality function, the first local maximum of the C scores as γ decreases from 1.0 to 0.0 is found. After that, it is the maximum of all the maxima that is taken.

2.4. Iterative Clustering

The resulting best candidate is used as initial model for an iterative refinement algorithm, and this produces the final complete clustering solution.

The algorithm we use in this step is Expectation Maximization (EM) [8], the most popular algorithm among them. EM starts with an initial model, and then it repeatedly determines the membership of each document to a cluster according to the current model (Expectation) and reestimates the parameters of the model using the new assignment of the documents to the clusters (Maximization), until the quality of the new model is not better than that of the previous one.

The parameters of the model for EM are estimated using Naive Bayes, as in [9].

3. Evaluation Framework

3.1. Evaluation Corpora

The corpus used to evaluate our approach is Switchboard [10]. This is a collection of 2438 spontaneous telephone conversations among 543 speakers (302 male, 241 female) for which manual transcripts are available[1]. The participants in every conversation were chosen by a computer operator, who also introduced a topic to talk about from a set of 67. For our purposes, we use these topics as categories.

We used two different collections extracted from Switchboard Corpus: The first one, **SWB**, consists of all the documents in the corpus, a total of 4876, one for each side of the conversations. The second one, **SWB-22**, consists of only those documents belonging to those categories from 100 documents up. This is a collection of 22 categories and 2682 documents, coming from 1341 conversations. The aim of this second collection is to allow comparisons of the behavior of the different methods when working on data sets with less categories and a more similar number of documents in all of them: in SWB the number of documents in each category ranges from 156 to 8, whereas in SWB-22 the range is from 156 to 100.

3.2. Evaluation Metrics

In order to compare the quality of the clustering solutions resulting from different approaches, we use the following four measures:

[1] `http://www.isip.msstate.edu/projects/switchboard/index.html`

a) *Purity* evaluates the degree to which each cluster contains documents from a single category. The purity P of a cluster c_i is the fraction of the cluster size n_i that the largest category of documents assigned to c_i represents [11]. The overall purity is the weighted average of all cluster purities:

$$P(c_i) = \frac{1}{n_i} \max_j n_i^j \quad (8)$$

$$Purity = \sum_{i=1}^{k} \frac{n_i}{n} P(c_i) \quad (9)$$

Intuitively, the larger the purity value, the better the clustering solution is.

b) *Entropy* is an information-theoretic measure which can be used to analyze the distribution of categories in each cluster. The overall entropy of the solution is the weighted average of all cluster entropies, and it is defined as [11]:

$$E(c_i) = -\frac{1}{\log q} \sum_{j=1}^{q} \frac{n_i^j}{n_i} \log \frac{n_i^j}{n_i} \quad (10)$$

$$Entropy = \sum_{i_1}^{k} \frac{n_i}{n} E(c_i) \quad (11)$$

where q is the number of categories in the collection. Because entropy measures the amount of disorder in a system, the smaller the entropy value, the better the clustering solution is.

c) *Category recall (CR)* indicates the percentage of categories present in the clustering solution. We consider a category to be present if there is at least one cluster with the majority of its documents assigned to this category. As the goal is to extract all the categories in the collection, the larger the category recall, the better the clustering solution is. Additionally, if CR is closer to k, more clusters represent a different category. This means that the number of category splittings is lower, and should be a property of good clustering solutions.

d) *Estimated number of clusters (k).* The closer k is to the number of categories in the collection q, the better the initialization algorithm is.

3.3. Evaluation Procedure

To obtain statistically significant measures, we use a layout similar to a stratified 10-fold cross validation: we randomly split every collection in 10 folds, keeping in each fold the same distribution of categories as in the whole collection. We then use every group of 9 of these folds. The fact that the task of document clustering does not use separate training and test sets accounts for the fact that we do not use the spare fold.

4. Experiments and Results

We carried out two sets of experiments. In the first one, we compared the performance of our method against its two constituent algorithms. We ran our algorithm on the collections and then compared its results against those of HAC and an average across five runs of randomly initialized EM (EM5). As both are supervised algorithms, we fed them the estimated number of clusters, k, determined by our method.

In the second set of experiments, we compared our method against other unsupervised ones which allow the detection of the number of clusters. Specifically, we used Calinski and Harabasz method [6]. This method consists in generating a clustering solution for values of k from 1 onwards until a local maximum of C score is found. We have used HAC and EM for the clustering, the measures on the latter being again an average of 5 randomly initialized runs. We then compared the accuracy of the detected model dimension k as well as the quality of the clustering solution this k led to.

The documents were represented as *tfidf* vectors of words, and the distance metric between two documents was the cosine of the vectors.

The results of these experiments are described below.

Method	k	CR	Purity	Entropy
SWB Collection				
Ours	27.90 (1.66)	17.90 (2.88)	55.70 (2.31)	30.20 (1.40)
HAC	*id*	13.30 (0.94)	35.90 (2.34)	43.10 (2.47)
EM5	*id*	17.84 (2.04)	15.74 (3.35)	67.28 (5.92)
SWB-22 Collection				
Ours	17.10 (2.38)	15.60 (3.60)	77.30 (10.18)	20.00 (7.15)
HAC	*id*	10.00 (0.94)	51.90 (4.48)	35.70 (4.99)
EM5	*id*	9.92 (1.81)	30.94 (6.06)	57.06 (6.26)

Table 2: Comparison of method performances

4.1. Comparison of method performances

Table 2 shows the results for the first set of experiments, aimed at comparing the performance of the approaches. The mean and the standard deviation (in parentheses) are shown for each metric. Our method will be referred to as Ours.

Comparing the average values among all folds, we can see that the quality of the solutions proposed by Ours clearly outperforms those proposed by its constituents HAC and EM5, when using the same k found by Ours. This is a promising result considering that our approach is unsupervised. Concretely, purity and CR are higher and entropy is lower in both collections using Ours than using either HAC or EM5. Moreover, the measures for Ours are better even if we take standard deviation into account: decreasing (or increasing in the case of entropy) the values for Ours by a standard deviation, they are still better than the values for HAC or EM5 increased (resp. decreased) by a standard deviation. The only exception is CR in SWB, whose values for Ours and EM5 do not differ significantly. However, the other measures show that the quality of the clustering solution proposed by the former is clearly higher than the quality of that proposed by the latter.

It is also worth noticing that the CR achieved by Ours on SWB-22 is quite close to the found k. As mentioned before in section 3.2, this means that the number of split categories is low, which is a property of good clustering solutions.

4.2. Comparison of model dimension detection

3 shows the results for the second set of experiments, aimed at comparing the ability to detect a good estimation of the number of clusters. Here the differences between Ours and the other

Method	k	CR	Purity	Entropy
SWB Collection				
Ours	27.90 (1.66)	17.90 (2.88)	55.70 (2.31)	30.20 (1.40)
HAC + Calinski	5.80 (2.39)	3.50 (1.84)	7.40 (3.20)	87.30 (8.81)
EM5 + Calinski	3.90 (0.95)	0.76 (0.80)	8.78 (2.05)	77.28 (5.17)
SWB-22 Collection				
Ours	17.10 (2.38)	15.60 (3.60)	77.30 (10.18)	20.00 (7.15)
HAC + Calinski	5.30 (1.25)	3.50 (0.71)	21.60 (6.52)	73.00 (11.66)
EM5 + Calinski	4.50 (1.04)	1.36 (1.08)	18.54 (3.95)	69.86 (6.49)

Table 3: Comparison of model dimension detection

methods are even larger. HAC and EM5 dramatically underestimate k with respect to Ours. Even with standard deviation into account, the values for the worst cases of Ours (mean k decreased by its standard deviation) are significantly closer to the true number of categories in the collection than the best cases of HAC and EM5 (mean k increased by its standard deviation).

The comparison of the number of cluster k detected by Ours against the real number of categories in the collections shows a difference between the results on the two collections. In SWB-22 the k of 17.10 is quite close to the number of categories, 22, whereas for SWB the result of 27.90 is an underestimation of the real number of categories, 67. The greater difficulty of SWB probably accounts for this underestimation.

Underestimation of the number of clusters k produces clustering solutions with clusters containing documents from several categories. For this reason, clustering solutions with underestimated k have lower purity, lower CR and higher entropy.

4.3. Comparison of collections

Comparing both collections, SWB-22 seems easier to be clustered than SWB. All the methods obtain better results on the former than on the latter. The reason could lie in the structure of the collections: SWB-22 consists of less categories than SWB, and the sizes of SWB-22 categories are more similar than those of SWB. We think that this difference is the reason why better results are obtained on SWB-22 than on SWB. On the one hand, in SWB there are some categories with few documents. This may prevent the algorithm from finding evidence to identify the category as a cluster. On the other hand, some of these low-sized categories are highly ambiguous with respect to the rest of them. Consequently, they probably become merged with similar higher-sized categories into a single cluster.

4.4. Robustness to outliers

Tables 2 and 3 show that values of standard deviation for Ours executed on SWB-22 are quite higher than for the other methods. Looking at the results for every fold, we found that all folds have similar values, around the mean of the 10 folds, except for a single one, which shows a significantly worse behavior. In this one, the value for k was 11, and as a consequence the quality of the solution decreased (purity was 51%, entropy 38% and CR 7). Similarly, the performance of HAC and EM5 in this fold was also the lowest of all folds.

We think that the reason for this lies in the presence of outliers, which induce to confusion of similar categories. For instance, categories *Crime* and *Gun Control* had been merged into a single cluster. Even if our method is still the best ranked in situations with outliers, we think that additional work in the treatment of outliers could be done to improve its performance.

5. Conclusions

This paper proposes an unsupervised approach for speech document clustering, and presents the evaluation on two different datasets from Switchboard corpus. We compare our method against supervised approaches and other unsupervised ones. Our method clearly outperforms these approaches, so for purity as for entropy, category recall and estimated number of clusters.

We believe that these results are encouraging to consider future research on unsupervised clustering approaches as highly reliable for dealing with spontaneous speech document collections.

6. Acknowledgements

This work has been partially funded by the European CHIL Project (IP-506909), the *Departament d'Universitats, Recerca i Societat de la Informació* and the Spanish Ministry of Science and Technology (TIN2004-0171-E).

7. References

[1] Carlson, B.A., "Unsupervised Topic Clustering of Switchboard Speech Messages", in proceedings of the IEEE ICASSP'96, 1996.

[2] Grangier, D. and Vinciarelli, A., "Effect of Recognition Errors on Text Clustering", IDIAP-RR 04-82, 2004.

[3] Xu, W., Liu, X. and Gong, Y. "Document Clustering Based On Non-negative Matrix Factorization", in proceedings of the ACM SIGIR'03, 2003.

[4] Zhao, Y. and Karypis, G. "Evaluation of hierarchical clustering algorithms for document datasets", in proceedings of CIKM'02, 2002.

[5] Surdeanu, M., Turmo, J. and Ageno, A., "A hybrid unsupervised approach for document clustering", in proceedings of KDD'05, 2005.

[6] Calinski, T. and Harabasz, J. "A dendrite method for cluster analysis", Communications in Statistics, 3:1-27, 1974.

[7] Milligan, G.W. and Cooper, M.C. "An examination of procedures for determining the number of clusters in a data set", Psychometrica, 50, 1985.

[8] Dempster, A.P. Laird, N.M. and Rubin, D.B. "Maximum likelihood from incomplete data via the EM algorithm", Royal Statistical Society, Series B, 39(1), 1977.

[9] Nigram, K. McCallum, A. Thrun, S. and Mitchell, T. "Text classification from labeled and unlabeled documents using EM", Machine Learning, 39(2/3), 2000.

[10] Godfrey, J.J. Holliman, E.C. and McDaniel, J. "SWITCHBOARD: telephone speech corpus for research and development", in proceedings of the IEEE ICASSP'92, 1992.

[11] Zhao, Y. and Karypis, G. "Empirical and theoretical comparisons of selected criterion functions for document clustering", Machine Learning, 55(3), 2004.