Analysis of the Impact of the Audio Database Characteristics in the Accuracy of a Speaker Clustering System

Jesús Jorrín-Prieto[1]*, Carlos Vaquero*[2]*, Paola García*[1]

[1]Agnitio S.L., Madrid, Spain
[2]Cirrus Logic International, Madrid, Spain
{jjorrin, pgarcia}@agnitio-corp.com
carlos.vaquero@cirrus.com

Abstract

In this paper, a traditional clustering algorithm based on speaker identification is presented. Several audio data sets were tested to conclude how accurate the clustering algorithm is depending on the characteristics of the analyzed database. We show that, issues such as the size of the database, the number speakers, or how the audios are balanced over the speakers in the database significantly affect the accuracy of the clustering task. These conclusions can be used to propose strategies to solve a clustering task or to predict in which situations a higher performance of the clustering algorithm is expected. We also focus on the stopping criterion to avoid the worsening of the results due to mismatch between training and testing data while using traditional stopping criteria based on maximum distance thresholds.

1. Introduction

Research in the field of speech technologies has encouraged the collection of voice databases for several tasks. These databases may have large number of audio files; listening to them and extract conclusions is not feasible. To avoid such manual inspection, speaker clustering techniques are introduced. Clustering techniques cover algorithms to find, in an automatic manner, groups among a set of objects so that elements included in the same group are very similar and at the same time different from elements contained in other groups. The aim of speaker clustering is to classify speech segments into groups so that each group contains segments of a unique speaker and every segment of a speaker is contained in the same group.

Speaker clustering is a powerful tool which finds application in uncountable cases. For example, it can be used to find links between speakers in large databases for law enforcement and surveillance applications [1]. Speaker clustering has also motivated studies in automatic annotation and rich transcription of meetings, broadcast news or multimedia documents [2,3]. Other recently studied application is the use of speaker clustering to adapt a Speaker Recognition system to a new domain [4,5].

We find many recent works in which speaker clustering approaches are tested [4, 5, 6, 7]. However, most of these techniques are evaluated using databases where the distribution of the number of audios[1] per speaker is always very similar. i.e., implicitly, the same prior in the number of audios per speaker is assumed when these techniques are evaluated. NIST SRE datasets are among the most used to test speaker clustering approaches [4, 5, 6, 7]. Such datasets have a specific distribution of audios per speaker [7] that may influence the results of the clustering task; care should be taken while extrapolating the clustering results to other databases. We can find previous works [8] which state that the characteristics of the database can affect severely the accuracy of the clustering task.

In this work, we analyze the influence of the database characteristics in the accuracy of the clustering task. We show how variations on the distribution of audios per speaker affect the clustering task, both in terms of speaker and clustering purity and in terms of determining the actual number of speakers in the database.

This paper is organized as follows. Section 2 gives an overview of the clustering algorithm we considered. Section 3 describes the audio database, the performed experiments focusing on the analyzed variables and the measures used to evaluate the results. The results are presented in Section 4. Finally, Section 5 summarizes the conclusions extracted from all the experiments.

2. Clustering Algorithm

We consider a bottom-up Agglomerative Hierarchical Clustering (AHC). This is a greedy algorithm that starts with a number of clusters equal to the number of speech segments and it merges the closest clusters iteratively until a stopping criterion is met. We use an approach as the one presented in [8], which only requires a pairwise distance matrix. The advantage of this method is that before running the clustering algorithm we only need to compute the distance between all speech segments.

In our system we define the distance between two speech segments j and k as the score provided by a PLDA system [9]:

$$d(j,k) = -score_{PLDA}(j,k) \tag{1}$$

We use a gender independent i-vector PLDA system that uses a mixture of two gender dependent PLDA models, as the one described in [10]. We utilize i-vectors of 400 dimensions. The PLDA model considers a full covariance matrix to model the session component and a low rank matrix of dimension 120 to span the speaker subspace in the i-vector space. The system performs i-vector centering and length normalization.

While merging two clusters, the distance between the new cluster and the remaining ones must be updated. We consider minimum distance, or maximum score, criterion: when a new cluster m is obtained after merging two clusters j and k, the score for this cluster against any other cluster n is computed as the maximum score, or minimum distance, of the cluster j against n and the cluster k against n. Thus, the updated scores for the new cluster m against all the other clusters are:

[1] From now on we will use the shorthand "audio" to refer to audio files, audio collection or audio data.

doi: 10.21437/Odyssey.2016-57

$$d(m,n) = \min\{d(j,n), d(k,n)\}, \forall n \qquad (2)$$

One critical point while working with clustering algorithms is the stopping criterion. One of the most common is maximum distance [11]. This criterion stops the clustering algorithm when the distance between any clusters and all its neighbors is bigger than a certain threshold.

The desired stopping iteration is the one in which the number of clusters is equal to the number of speakers in the clustering task.

We consider two approaches of a maximum distance stopping criterion.

Maximum distance with raw scores (MD-R): To get the threshold value for the stopping criterion, the clustering algorithm runs over a labeled database. In such execution, the algorithm is configured to stop when there are as many clusters as the number of speakers. The score associated to the last merge will be the threshold value.

Maximum distance with unsupervised score calibration (MD-USC): Algorithm described in [12] allows us to obtain calibrated scores from unsupervised data. Calibrated scores, instead of those obtained from the PLDA system, are used to run the cluster algorithm. A labeled database is used to estimate the prior distributions required and threshold value is set to 0.

3. Experimental set-up

In this work, we designed 4 different experiments, each composed of several clustering tasks. Solving a clustering task implies running the clustering algorithm over a set of audios. The experiments analyze how the distribution of audios per speaker in the database affects the clustering task. Each experiment will focus in one variables that characterizes an audio database, isolated from the rest of them.

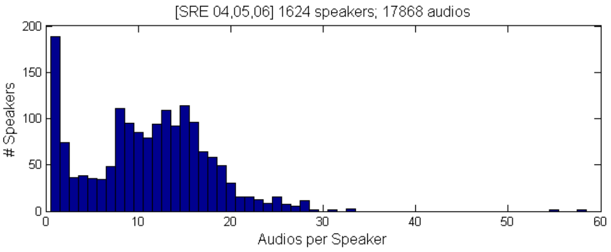

Figure 1: *Distribution of audios per speaker for the global set of audios (GSA)*

3.1. Audio database

We consider one initial set of 17868 audios and 1624 speakers obtained from NIST SRE’04, SRE’05 and SRE’06. This initial set will be named global set of audios (GSA) in the rest of the document. GSA has the distribution of audios per speaker showed in Figure 1.It is common to have databases with a distribution of audios per speaker similar to the one showed in Figure 1 [7].

3.2. Size of the task

The purpose of this experiment is analyzing the impact of the audio database size, which consists of 6 groups of clustering tasks. Each group will contain certain number of tasks with constant size. To constitute the audios used in each task of a specific group we divide GSA into subsets of constant size, so all audios are distributed among the different subsets. By doing so, GSA is fragmented into subsets containing around 10, 100, 1000, 3000, 9000 and 18000 recordings. Since all the audios are distributed among the different subsets for each size, each group of tasks will have different number of them. These numbers are shown in Table 1.

Table 1: *Number of clustering tasks for each group of tasks depending on the size of the task*

Size 10	Size 100	Size 1000	Size 3000	Size 9000	Size 18000
1800	180	18	6	2	1

The subsets of audios were designed to have distributions of audios per speaker similar to showed in Figure 1: For subsets of sizes 10 about 50% of the speakers have between 1 and 2 audios and the other 50% have more than 3 audios. For subsets of sizes 100 and 1000 40% of the speakers have between 1 and 2 audios, other 50% have between 3 and 9 and the other 10% have more than 10 audios. For subsets with more than 1000 audios about 40% of the speakers have between 1 and 5 audios, other 50% have between 6 and 15 and the other 10% have more than 16 audios.

All the generated subsets will be used to run a clustering task. No stopping criterion is used for this experiment. This means the clustering algorithm is run until we have a single cluster.

3.3. Number of speakers

We define R as the ratio between the number of speakers and the number of audios in a clustering task. R ϵ (0,1]. Upper limit values match the scenario in which there are as many speakers as the number of audios, while the lower one refers to those cases where only one speaker is found.

To analyze the impact of R and, in consequence, the number of speakers, we consider 6 groups of 40 clustering tasks. Clustering tasks of each group will have 5, 10 20, 50, and 80 speakers respectively. We fix the size of the tasks to 100 and the number of audios per speaker, so all the speakers have similar number of audios. This means that we have a uniform distribution of audios per speaker. We consider a subset of audios from GSA to run the clustering tasks. All audios of this subset are distributed among the different tasks for a group. The subset of audios from GSA is the same for all the groups of tasks. Table 2 shows the number of clustering tasks for each group.

Table 2: *Number of clustering tasks for each group of tasks depending on R*

R=0.05 (5 spks & 100 audios)	R=0.1 (10 spks & 100 audios)	R=0.2 (20 spks & 100 audios)	R=0.5 (50 spks & 100 audios)	R=0.8 (80 spks & 100 audios)
40	40	40	40	40

No stopping criterion is used for this experiment. The clustering algorithm is run until we have a single cluster.

3.4. Balance of speakers

How the audios are balanced over the speakers in the database is other variable that characterizes an audio database. The objective of this experiment is to study the impact of the presence of predominant speakers in the clustering task (the speaker who has the largest quantity of audios).

We define P as the percentage of audios belonging to the predominant speaker. To evaluate how P influences the results of a clustering task, we consider groups of 62 clustering tasks in which the predominant speaker will have the 12%, 25%, 35% and 50% of the audios respectively. There will be the same number of speakers in all tasks, no matter the group. Speakers other than the predominant one will have about the same number of audios, and this number will vary depending on the group. We fix the size of the tasks to 40 for all the tasks.

No stopping criterion is used for this experiment. This means the clustering algorithm is run until we have a single cluster.

3.5. Stopping criterion estimation

Both MD-R and MD-USC require a set of labeled data. The former to estimate the threshold value and the latter to estimate the prior distributions required by the unsupervised score calibration process. We performed experiments to measure the accuracy of both algorithms depending on how similar the labeled dataset is to the one used in the clustering algorithm. The labeled set contains 100 audios in which the distribution of audios per speaker is the same as the described in section 3.2 for tasks of size 100 (♦). Additionally, we consider 4 groups of unlabeled of audios. Each of those groups will contain 10 sets of 100 audios with the same distribution of audios per speaker. Distributions are showed in Table 3.

Table 3: *Number of clustering tasks, size of the tasks and audios per speaker distribution for each group of tasks.*

Group	Audios per speaker distribution	size	Number of clustering tasks
A	(♦)	100	10
B	Uniform with 5 speakers with 20 audios	100	10
C	Uniform with 20 speakers with 5 audios	100	10
D	Uniform with 50 speakers with 2 audios	100	10

Each audio data set is used to run two clustering tasks, each with a different stopping criterion. To evaluate the results, per clustering task, the number of clusters when the algorithm stops and difference with the real number of speakers in the task is computed (recall that the goal of the stopping criterion is to stop in the iteration in which the number of clusters is equal to the number of speakers in the clustering task). Later, we compute mean (μ) and standard deviation (σ) of such values for all the tasks of a group.

3.6. Performance measures:

Throughout this document we evaluated the approaches in terms of speaker impurity (SI) and clustering impurity (CI) [8]. Speaker impurity measures how spread a speaker is among the clusters, while cluster impurity measures how corrupted a cluster is from speakers not belonging to such cluster. CI and SI are used to build impurity trade-off (IT) curves. Each point of the IT curves shows CI and SI at each iteration of the clustering process: low cluster impurity zone is identified with the first iterations of the process, while high cluster impurity corresponds to the last iterations of the process. In such graphs the point in which the SI and the CI are equal is denoted as equal impurity rate (EI). The EI is a reference working point in many experiment results. In the experiments described in the previous sections, we had groups of clustering tasks in which the databases of the tasks for a specific group had similar characteristics. Since we want to compare results between tasks with different characteristics, results from all the tasks for a certain group will be evaluated using one single IT curve. To do so, clustering results for all tasks are grouped together to estimate the accuracy for all the subsets. We use CI and SI values of each step of the individual processes to compute global CI and SI values for all the tasks jointly. Common speakers that belong to different subsets will be treated as independent to compute SI values. By doing so, we are able to compare different groups of clustering task using one single TI curve per group.

4. Results

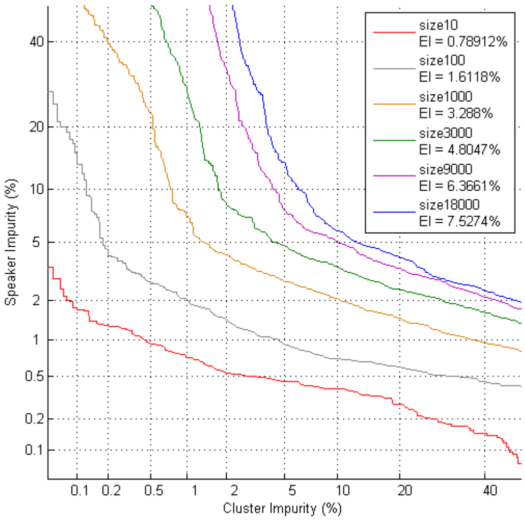

Figure 2*: Evaluation of a general clustering task: Impurity trade-off curves for several subset sizes.*

4.1. Size of the task

Figure 2 shows several IT curves. Each of them groups the results for all the clustering tasks of a specific size. Based on Figure 2, we can observe that the larger the subset the worse the accuracy of the clustering task.

Table 4: *Equal Impurity rates of curves in Figure 2*

Size of the task	10	100	1000	3000	9000	18000
EI(%)	0.789	1.61	3.28	4.804	6.36	7.52

EI values from Figure 2 are presented in Table 4. Based on Table 4, we observe that there may be a relation between the rise of the size in the task and the increase of the EI value (each time the size of the task is increased by a factor 10, the EI approximately doubles). We define a reduction size factor as the ratio between the size of the biggest task and the size of the smaller one. We can find the best function that describes the increase of the EI:

$$f(r)=\frac{EI_2}{EI_1} \tag{3}$$

where EI_1 and EI_2 are the equal impurity rates of the smaller and biggest task respectively, r the reduction size factor and $f(r)$ is a two terms power series of the form:

$$f(r)=a\cdot r^b+c \tag{4}$$

From Table 4 we can build several points with the form $\{f(r),r\}$. For example, if we select size 10 and 100 we have $\{\frac{1.61}{0.789},\frac{100}{10}\}=\{2.04,10\}$. We can check that such points fit equation (4), as shown in Figure 3. $f(r)$ coefficients used to plot Figure 3 are: $a=1.284$, $b=0.2759$ and $c=-0.4081$.

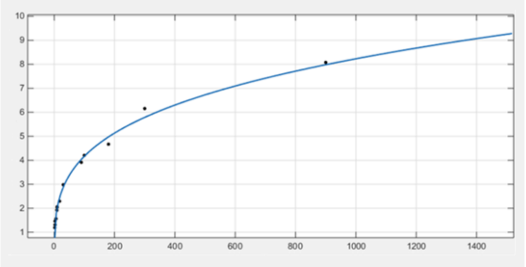

Figure 3: *Fit of working points computed from Table 4 within f(r)*

These results are interesting since they allow us to predict how accurate the clustering algorithm behaves in large datasets with just a sample of the complete database. It must be noticed that this sample must be large enough to provide significant measures. If so, tasks with different sizes must be designed and with the obtained results $f(r)$ can be stated. Note that the results were obtained using a specific database, so we must be cautious to extrapolate the results, since $f(r)$ depend on the distribution of audios per speaker of the analyzed database.

4.2. Number of speakers

Figure 4 shows several IT curves. Each of them groups the results for all the clustering tasks with a specific number of speakers. From Figure 4 we can conclude that the worst case is the 50 speaker's scenario. However it would be useful to have another way of representing the data to extract clearest conclusions. We get a better representation of the experiments' results if we plot SI rates for a specific CI value dependent on R. This alternative method is shown in Figure 5.

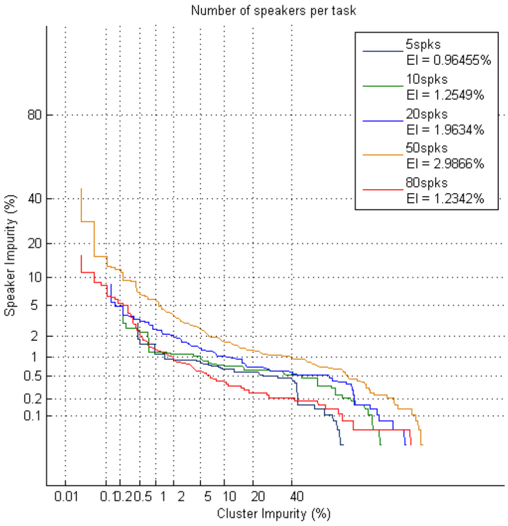

Figure 4:*Evaluation of clustering task with constant audios per speaker: Impurity trade-off for several numbers of speakers per task of 100 session.*

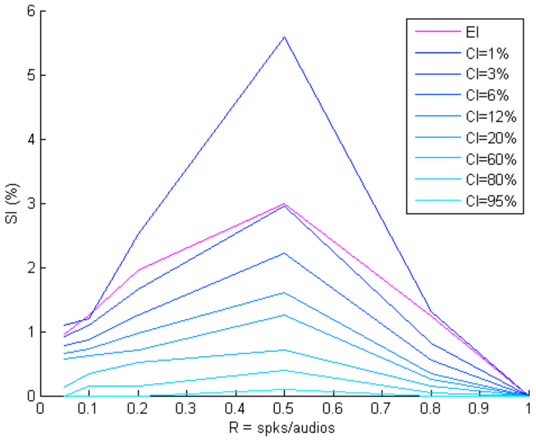

Figure 5: *Evaluation of clustering task with constant audios per speaker: Values of SI dependent on R for different values of CI and tasks of 100 sessions.*

Based on Figure 5 we can conclude that there is a difference between values of SI -for a constant CI rate- depending on R. We can observe SI is maximum for central values of R and from this point, SI decreases as R decreases and also as R increases. This implies that clustering tasks with R of about 0.5 will be harder to resolve than others with lower or higher R, because to get a certain CI rate, higher SI values are needed. We also reviewed that the dependence of SI with R decreases as CI increases. The variation between values of SI is lower with high values of CI. R has been defined as the number of speakers in the task divided by the number of audios, but it also helps us to know the level of the clustering dendogram in which the optimal solution[1] is found. If we have n as the number of speakers in the task, we should stop after the last n^{th} clustering merge. As we have as many possible

[1] As with the AHC algorithm not all the partitions are analyzed, the optimal solution (the one which has EI=0%) may not be reached. In this context, we refer to the optimal solution as the one which guess the correct number of speakers.

merges as the number of total audios (N), the optimal iteration to stop the clustering algorithm is

$$it_{opt} = N - n = N - N \cdot R = N \cdot (1 - R) \qquad (5)$$

With this definition and based on Figure 5, we can state that the most difficult tasks are those which optimal solution is found in the middle of the clustering process and the easiest are those where the optimal solution is on the top or on the bottom of the clustering dendogram. To demonstrate this statement we introduce the Stirling number of second kind, defined as the number of ways to partition a set of n objects into k non-empty subsets and is denoted by $S(n,k)$ or $\left\{ {n \atop k} \right\}$

$$S(n,k) = \left\{ {n \atop k} \right\} = \frac{1}{k!}\sum_{j=0}^{k}(-1)^{k-j}\binom{k}{j}j^{n} \qquad (6)$$

Given a clustering task, defined by the number of audios and the number of speakers, the Stirling number of second kind allow to compute the number of possible partitions with a specific number of clusters. In our analysis k is the number of speakers (as the optimal solution has one cluster per speaker) and n the audios in the task. Figure 5 shows $S(n,k)$ for some n and k values.

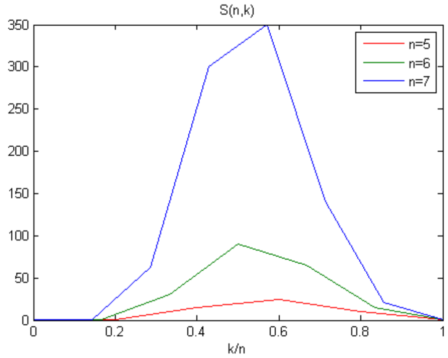

Figure 6: *Stirling numbers of second kind for several combinations of k and n.*

Based on Figure 6 we observe that the number of partitions follow the same trend as the results shown in Figure 5. It makes sense that both graphs behave similar, due to the fact that the greater the number of possible partitions, the most difficult to find the good one. So in those cases where more partitions are evaluated, errors occur more often than in the cases where few partitions are possible, obtaining in consequence worse results.

It must be noticed that this analysis was focused on how easy is to pass through the optimal solution while building the clustering dendogram, later we should be able to stop at that point using an accurate stopping criterion. This topic will be studied in section 4.4.

4.3. Balance of speakers

We consider 3 initial groups of tasks (P=12%, P=25% and P=50%) with 20 speakers (R=0.5). Results for all the tasks for each group are plotted in Figure 7.

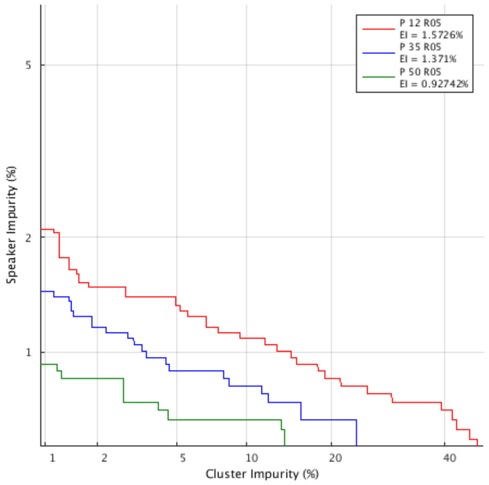

Figure 7: *Impurity trade-off curves for tasks in which the main speaker has the 12%, 35% and 50% of all the audios; in tasks of 40 sessions with R=0.5*

Based on Figure 7 we can clearly see that the amount of audios of the predominant speaker, P, affects the results, obtaining higher accuracy when we have a predominant speaker with most of the audios. The reason behind is that all the points of the IT curve for P=50% are under the rest of curves.

Considering only experiments that have R=0.5 may not be enough to extract clear conclusions, since in this case the difference between the number of audios for the predominant speaker and the rest of them is big enough no matter the value of P. For this reason, we consider other 3 groups of tasks but this time with 8 speakers (R=0.2). Results are shown in Figure 8.

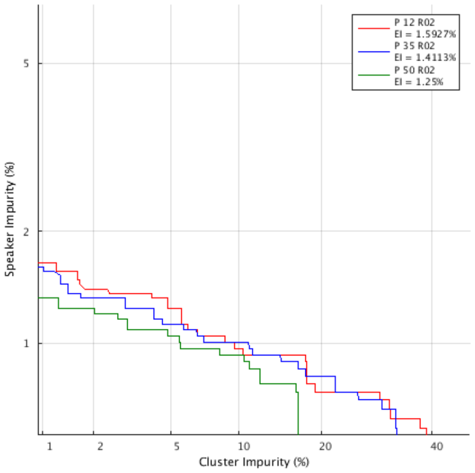

Figure 8: *Impurity trade-off curves for task in which the main speaker has the 12%, 35% and 50% of all the audios; in tasks of 40 sessions with R=0.2*

From Figure 8, we can state that similar results are obtained for all values of P, since for this case the difference between the number of audios for the predominant speaker and the rest of them is not big enough. One last experiment was carried out, because based on the results showed in Figure 7 and 8 we cannot ensure that differences found are only due to R, without considering the value of P. Hence, we must check the following hypothesis: fixing the number of speakers (R) and assuming that the results are dependent on the percentage of audios of the predominant speaker, P, there are ranges of P in which the results do not vary. Assuming that tasks with R bigger than 0.5 have different results depending on the value of R, we want to find P threshold value that under it the results remain similar. Based on figure 7, we can state that from P=35%, results will be P dependent, so if a threshold value exists, it should be under P=35%. To check this, we add a new group with P=25% to the results showed in Figure 7.

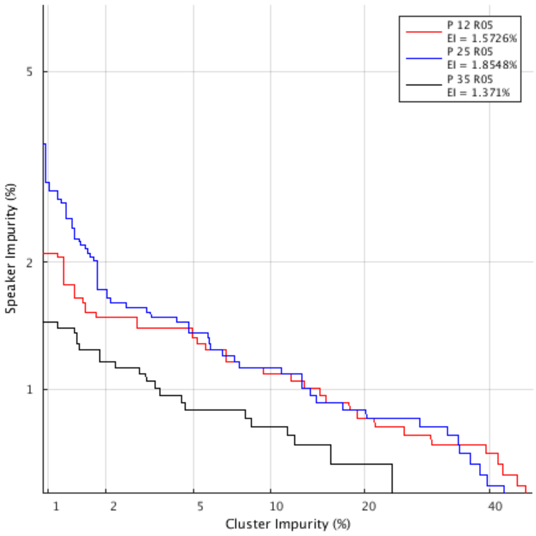

Figure 9: *Impurity trade-off curves for tasks in which the main speaker has the 12%, 25% and 35% of all the audios; in tasks of 40 sessions with R=0.5*

From Figure 9, we can verify our hypothesis, as the results for P=25% and 12% are quite similar.

Based on Figure 7, 8 and 9 we conclude that: given R, we find a specific value of P from which the results are better and under it all P offers similar results. This P threshold is R dependent, having higher values as R decreases. There may be cases in which the threshold value is so high that no matter the number of audios of the main speaker, the results don't change at all. This situation is more common if we work with low values of R. On the other hand, we could find cases in which for all values of P, the higher the value of P the better the results of the experiments. This situation will be related with those scenarios with high values of R.

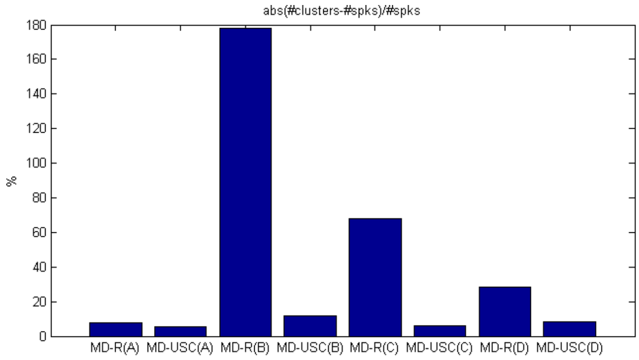

Figure 11: *Relative difference between the correct number of speakers and the real number of speakers*

4.4. Stopping criterion

Table 5 shows the mean number of clusters of each group of tasks after the stopping criterion is met. Since the objective is to stop when the number of clusters is equal to the number of speakers, the difference between these numbers is used as performance measure. Figure 11 shows the accuracy of each approach depending on the evaluated group.

Table 5: *Clustering tasks results: number of clusters and difference between real number of speakers and clusters when the clustering algorithm stops.*

Group	Stopping criterion	#Clusters	#clusters - #speakers
A	MD-R	μ=49,17 σ=7,61	μ=4,26 σ=3,43
A	MD-USC	μ=44,48 σ=7,05	μ=2,72 σ=3,09
B	MD-R	μ=13,90 σ=4,90	μ=8,90 σ=4,90
B	MD-USC	μ=5,40 σ=0,96	μ=0,60 σ=0,84
C	MD-R	μ=33,6 σ=4,71	μ=13,6 σ=4,71
C	MD-USC	μ=20,80 σ=1,39	μ=1,20 σ=1,03
D	MD-R	μ=64,20 σ=2,74	μ=14,20 σ=2,74
D	MD-USC	μ=54,10 σ=1,66	μ=4,10 σ=1,66

Based on Figure 11, the accuracy of MD-USC keeps, while MD-R offers a great variability. For all the groups, MD-R also offers poor results compared with those obtained with MD-USC having a relative difference between number of speakers and number of clusters that goes from 7.9 % to 178%, while such relative difference for MD-USC goes from 5.7% to 12%. Group A is the only case in which MD-R performs similar results to those obtained with MD-USC, although they are still worse (7.9% versus 5.7%). This happen since the audio database used to run the clustering algorithm is quite similar to the one used to compute the threshold. In this way, accurate results while using MD-R should be expected only if the training and testing set have similar audios per speaker distribution. On the other hand MD-USC offers accurate results even when there is a decalibration between training and testing set. It is true that the best results are obtained when there is no such decalibration, but if it exists, the accuracy of the system do not get as worse as in the case of MD-R.

5. Conclusions

We have analyzed the impact of the database characteristics in a) the accuracy of the speaker linking process, in terms of clustering and speaker purity, and b) in the accuracy of the stopping criterion. We have shown that the size of the database, the number speakers for a fixed amount of audios, and how the audios are balanced over the speakers in the database are variables that affect significantly the accuracy of a clustering task.

We demonstrate that there is a function that characterizes the worsening of the results in terms of EI as the size of the task increases. i.e., that smaller tasks will generate more accurate results than larger ones. We have also shown that, under the assumption of having a fixed distribution of audios per speaker, it may be possible to estimate the accuracy of a clustering task for a database (employing several measures to perform a regression to a two term power series). Given a small sample of a labeled database, that represents a much larger database; we may be able to estimate the accuracy of the clustering task in the full database. Further research should analyze and consider different distributions of the number of audios per speaker, to ensure this affirmation holds.

We have also shown that the accuracy of the clustering task also depends on the number of speakers, finding that tasks in which the number of speakers is about half of the number of audios will generate worse results. These results are supported by a theoretical analysis on the number of possible partitions for a given number of speakers in a clustering task. This help us to have prior knowledge on how accurate a clustering task is or how accurate it will be if we know the number of speakers in advance, or after the clustering task has provided a number of speakers.

We have presented that the balance of audios per speaker affects significantly the clustering task: If we have a predominant speaker, that is, there is at least one speaker with a number of audios much larger than the rest of them, more accurate results can be expected. As in the previous case we can predict how accurate the results are, if we find a predominant speaker in the solution offered by the algorithm or if we have such prior knowledge.

We showed that the prior distribution considered in the stopping criterion affects its accuracy severely. Assuming that the distribution of audios per speaker in the database is unknown, unsupervised techniques must be used. We showed that the use of unsupervised calibration can improve significantly the accuracy of a stopping criterion in this situation.

Finally, we want to emphasize that conclusions extracted from our experiments are not only useful to measure accuracy dependent on our database, but also they can influence the design of future algorithms of clustering. For example, from our analysis of the size of the task we concluded that errors occur less frequently for smaller tasks than for bigger ones. These results suggest that instead of running the clustering algorithm in the complete database, dividing the initial set into partitions of smaller sizes will guarantee better results. For this reason, approaches based on that divide and conquer could be used to compensate the effect of the size of the task.

6. References

[1] Anil Alexander, Oscar Forth "Blind Speaker Clustering Using Phonetic and Spectral Features in Simulated and Realistic Police Interviews". In International Association for Forensic Phonetics and Acoustics (IAFPA) conference, Santander, Spain, August 2012.

[2] Qin Jin, Kornel Laskowski, Tanja Schultz, and Alex Waibel, "Speaker Segmentation and clustering in Meetings". In Proc. ICASSP-2004 Meeting Recognition Workshop, Montreal, Canada, May 2004.

[3] Grégor Dupuy, Sylvain Meignier, Paul Deléglise, Yannick Estève, "Recent Improvements on ILP-based Clustering for Broadcast News Speaker Diarization". In Odyssey workshop, Joensuu, Finland, June 2014, pp. 187-193.

[4] Stephen H. Shum, Douglas A. Reynolds, Daniel Garcia-Romero, Alan McCree, "Unsupervised Clustering Approaches for Domain Adaptation in Speaker Recognition Systems". In Odyssey workshop, Joensuu, Finland, June 2014, pp. 265-272.

[5] Daniel Garcia-Romero, Alan McCree, Stephen Shum, Niko Brümmer, Carlos Vaquero, "Unsupervised Domain Adaptation for I-Vector Speaker Recognition". In Odyssey workshop, Joensuu, Finland, June 2014, pp. 260-264.

[6] Elie Khoury, Laurent El Shafey, Marc Ferras, Sébastien Marce, "Hierarchical speaker clustering methods for the NIST i-vector Challenge". . In Odyssey workshop, Joensuu, Finland, June 2014, pp. 254-259.

[7] Stephen H. Shum, William M. Campbell, and Douglas A. Reynolds, "Large-scale community detection on speaker content graphs," in Proceedings of ICASSP, 2013.

[8] David A. van Leeuwen, "Speaker linking in large datasets," In Odyssey2010, the Speaker Language and Recognition Workshop, Brno, Czech Republic, June 2010, pp. 202–208.

[9] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in 11th International Conference on Computer Vision, 2007, pp. 1–8.

[10] Senoussaoui, M., Kenny, P., Brummer, N., de Villiers, E., and Dumouchel, P., "Mixture of PLDA Models in I-Vector Space for Gender-Independent Speaker Recognition", In Proc. Interspeech, Florence, Italy, 2011.

[11] Marijn Huijbregts and David van Leeuwen, "Large scale speaker diarization for long recordings and small collections". In IEEE Transactions on Audio, Speech, and Language Processing, 2010.

[12] Niko Brümmer and Daniel Garcia-Romero. "Generative modeling for unsupervised score calibration". In ICASSP 2014.