Speaker Utterances tying among speaker segmented audio documents using hierarchical classification: towards speaker indexing of audio databases

Sylvain Meignier, Jean-François Bonastre, Ivan Magrin-Chagnolleau

SPEAKER UTTERANCES TYING AMONG SPEAKER SEGMENTED AUDIO DOCUMENTS USING HIERARCHICAL CLASSIFICATION: TOWARDS SPEAKER INDEXING OF AUDIO DATABASES

Sylvain Meignier[*(1)], *Jean-François Bonastre*[(1)], *Ivan Magrin-Chagnolleau*[(1)(2)]

[(1)]LIA / CERI

Université d'Avignon - Agroparc - BP 1228 - 84911 Avignon Cedex 9 - France

[(2)]Laboratoire Dynamique Du Langage - UMR 5596

Université Lumière Lyon 2 & CNRS - 14, avenue Berthelot - 69363 Lyon Cedex 07 - France

{sylvain.meignier, jean-francois.bonastre}@lia.univ-avignon.fr - ivan@ieee.org

ABSTRACT

Speaker indexing of an audio database consists in organizing the audio data according to the speakers present in the database. It is composed of three steps: (1) segmentation by speakers of each audio document; (2) speaker tying among the various segmented portions of the audio documents; and (3) generation of a speaker-based index. This paper focuses on the second step, the speaker tying task, which has not been addressed in the literature. The result of this task is a classification of the segmented acoustic data by clusters; each cluster should represent one speaker. This paper investigates on hierarchical classification approaches for speaker tying. Two new discriminant dissimilarity measures and a new bottom-up algorithm are also proposed. The experiments are conducted on a subset of the Switchboard database, a conversational telephone database, and show that the proposed method allows a very satisfying speaker tying among various audio documents, with a good level of purity for the clusters, but with a number of clusters significantly higher than the number of speakers.

1. INTRODUCTION

Speaker indexing of an audio database consists in organizing the audio data according to the speakers present in the database. It is composed of three steps (Figure 1). The first step is the segmentation of each audio document by speakers. The segmentation produces a set of speaker-based segmented portions that we will refer to as *speaker utterances* in the following. The second step consists in tying the various speaker utterances among several previously segmented audio documents. During this stage, one label is attributed to all the speaker utterances matched together. The last stage corresponds to the creation of a speaker-based index.

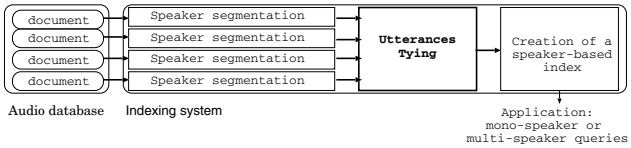

Fig. 1. *Block-diagram of a speaker indexing system.*

The speaker segmentation problem (Figure 2) is usually addressed by one of the two following methods. The first one (described in [1][2][3]) relies on a speaker change detection followed by a clustering step. The second one (see for instance [4][5]) does the segmentation and the clustering simultaneously using a hidden Markov model. In both cases, the system has to determine the set of speakers present within a given audio document as well as the boundaries of each intervention. No *a priori* information on speakers is used in these approaches, *i.e.* the speaker utterance models are built during the segmentation process. However, a world / background model may have been trained on a different data set and used to adapt the speaker models and / or normalize the scores computed with each speaker model.

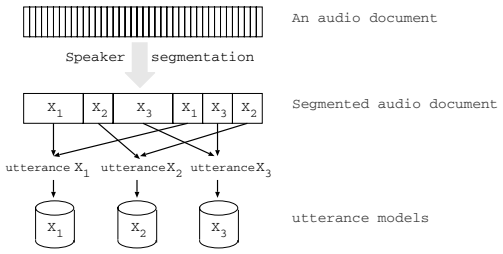

Fig. 2. *Speaker segmentation and generation of speaker utterance models.*

Speaker utterances tying is a classification problem similar to speaker clustering [2][3]. Speaker clustering is usually done inside one audio document, whereas, in speaker utterances tying, utterances are matched among several audio documents (Figure 3). However, similar segments inside one audio document are already matched during the preliminary segmentation process. Moreover, the matched utterances are longer than the segments used in speaker clustering. But the great number of channels in the set of audio documents represents an additional difficulty.

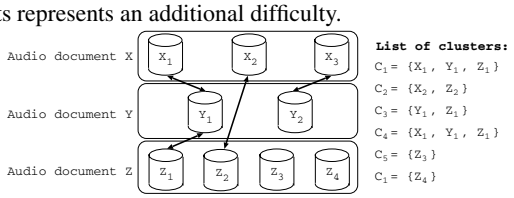

Fig. 3. *Example of speaker utterances tying.*

The last step of speaker indexing is the creation of a speaker-based index which remains an open and difficult problem for real applications. The aim of a speaker-based index is to organize the matched speakers to make the search in a database more efficient.

In this paper, we focus on the speaker utterances tying step, which has not been addressed in the literature, assuming that the

* RAVOL project: financial support from Conseil général de la région Provence Alpes Côte d'Azur and DigiFrance.

segmentation has been accurately done: we use the segmentation given with the Switchboard database and this segmentation has not been evaluated again. An accurate segmentation is a segmentation in which each intervention is labeled by the right speaker and all speakers have been found. The boundaries have also accurately been set. We also suppose that models have been computed on every speaker utterance during the segmentation phase. At the end of the speaker utterances tying step, all the speaker utterances previously segmented are grouped into clusters, each cluster corresponding to a speaker identity. We also assume that the number of speakers (that is, the number of clusters) in the audio database is unknown. However, in the framework of an audio information retrieval application, this number will usually be high. Therefore, it is not possible to re-calculate all the models when an additional audio document (that is, additional speaker utterances) is added to the audio database. The chosen technique will have to allow the addition of new speakers with a low cost.

The methods used in speaker clustering (described in [1][2][3] for instance) can be applied to the speaker utterances tying problem. Hierarchical classification is the main method proposed for speaker clustering. It is an iterative agglomerative method. At each stage, the algorithm groups the two closest clusters, according to a chosen dissimilarity measure. The result of hierarchical classification is generally represented by a dendrogram which illustrates the consecutive groupings of clusters (Figure 4).

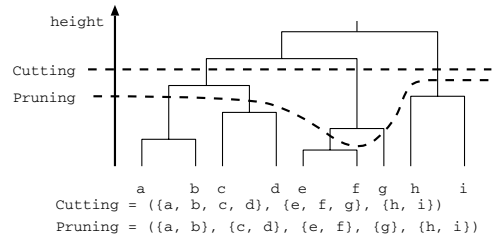

Fig. 4. *Example of dendrogram: cluster selection methods.*

In this paper, we propose to apply a hierarchical classification approach to the new problem of speaker utterances tying. We apply this technique to an audio database composed of conversational telephone speech (a subset of Switchboard II).

2. SPEAKER UTTERANCES TYING ALGORITHM

A hierarchical classification method is defined by:

- a measure of dissimilarity between clusters;
- an agglomerative method based on the chosen measure to group clusters together;
- a dendrogram pruning method to select the final set of clusters.

2.1. Dissimilarity Measures

A dissimilarity measure expresses the closeness between clusters composed of speaker utterances. Let $d(u, v)$ be a dissimilarity measure between clusters u and v. We will assume in the following that d is symmetric, that is, $d(u, v) = d(v, u)$.

Here are a few dissimilarity measures classically proposed in the literature for speaker clustering:

- Methods which require models to be trained at each utterances grouping as the generalized likelihood ratio [2] or the Bayesian information criterion (BIC) [1]. These two methods have a heavy computational cost.
- Methods which do not require models to be trained at each utterances grouping as the cross likelihood ratio (d_{clr}) [3] or the symmetric Kullbach-Leibler distance [2].

Notation

Let X_i be the set of utterances corresponding to speaker i in the audio document X.

Let $S_X = \{X_1, \ldots, X_i, \ldots, X_n\}$ be the set of speaker utterances in the audio document X.

Let $\lambda(X_i)$ be the utterance model corresponding to data X_i.

Let $\overline{X_i} = S_X - \{X_i\}$ be the utterances not corresponding to speaker i.

Let $l(v|\lambda(u))$ be the likelihood of the data v given the model $\lambda(u)$. $l(v|\lambda(u))$ is normalized by the number of speech frames in the data v.

Let $l(v|\lambda(\overline{X_i})) = \max_{x \in \overline{X_i}} l(v|\lambda(x))$.

Let $l(\overline{X_i}|\lambda(u)) = \max_{x \in \overline{X_i}} l(x|\lambda(u))$.

Let $\lambda(W)$ be the background model.

Let $r(v|\lambda(u))$ be the ratio between $l(v|\lambda(u))$ and $l(v|\lambda(W))$.

Classic Dissimilarity Measures

The cross likelihood ratio [3] is expressed in terms of dissimilarity as:

$$d_{clr}(X_i, Y_j) = \frac{l(Y_j|\lambda(W))}{l(Y_j|\lambda(X_i))} \cdot \frac{l(X_i|\lambda(W)}{l(X_i|\lambda(Y_j))}$$

This measure does not discriminate X_i and Y_j from the other utterance models $\overline{X_i}$ and $\overline{Y_j}$.

Proposed Dissimilarity Measures

We propose two measures which use explicitly all kind of information present in the audio documents X and Y (utterances or models). If utterances X_i and Y_j are produced by the same speaker, then speaker i can not have produced $\overline{Y_j}$ and speaker j can not have produced $\overline{X_i}$.

The first proposed dissimilarity measure uses data from the other utterances present in the audio documents X and Y:

$$d_1(X_i, Y_j) = \frac{f(\overline{Y_j}|\lambda(X_i)) + f(\overline{X_i}|\lambda(Y_j))}{f(Y_j|\lambda(X_i)) \cdot f(X_i|\lambda(Y_j))}$$

f can be the likelihood (l) or the likelihood ratio (r).

The second one uses the other utterances models (that is, $\lambda(\overline{Y_j})$ and $\lambda(\overline{X_i})$) corresponding to the data present in the audio documents X and Y:

$$d_2(X_i, Y_j) = \frac{f(Y_j|\lambda(\overline{X_i})) + f(X_i|\lambda(\overline{Y_j}))}{f(Y_j|\lambda(X_i)) \cdot f(X_i|\lambda(Y_j))}$$

Again, f can be the likelihood (l) or the likelihood ratio (r).

Dissimilarity Matrix

Based on one of these measures, we define a dissimilarity matrix composed of the dissimilarity measures between all pairs of utterances. But grouping utterances from the same document is not relevant, since the utterances of a given audio document are produced by different speakers. Therefore, $d(X_i, X_j) = +\infty$ for all i, j in a given audio document X.

2.2. Agglomerative Methods

Agglomerative methods are abundantly documented in the literature [2][6][7]. We just remind the formulae of the two algorithms that we used in our experiments.

Let $P_n = \{C_1, \ldots, C_i, \ldots, C_j, \ldots, C_n\}$ be the partition composed of n clusters.

Let c_i^a be an element of the cluster C_i.

Let c_j^b be an element of the cluster C_j.

In the single link method, the dissimilarity between two clusters is the minimum of the dissimilarity between all pairs of utterances drawn from the two clusters. The formula is: $d(C_i, C_j) = \min_{a,b} d(c_i^a, c_j^b)$.

In the complete link method, the dissimilarity between two clusters is the maximum of all pairwise dissimilarities between utterances in the two clusters: $d(C_i, C_j) = \max_{a,b} d(c_i^a, c_j^b)$.

2.3. Tree Pruning

At the end of the hierarchical classification algorithm, a dendrogram is built in which each node corresponds to a cluster. The cutting (or pruning) of the dendrogram produces a partition composed of all the utterances. Several techniques exist in the literature [2][7] for selecting a partition. These techniques consist in cutting the dendrogram at a given height or in pruning the dendrogram by selecting clusters at different heights (see Figure 4).

Classic Pruning Method

In our experiments, we selected the partition by pruning the dendrogram. The method that we call $Best$ is described in [3]. It is based on the estimated purity[1] $\hat{p}_i$ of the cluster i (see for instance [2][3]). At each stage, the best $score_i = \hat{p}_i - \frac{Q}{n_i}$ is selected. The $score_i$ is computed for each node i corresponding to the cluster i composed of n_i elements. The descendants and ancestors of i are then removed from the dendrogram. The algorithm is continued until the dendrogram is empty.

Proposed Pruning Method

We proposed a new method called Asc adapted from the previous one. The $score_i$ is also computed for each node. The tree is traversed from the leaves to the root. When $score_i$ stops growing between node i and its son-nodes, the two son-nodes are included in the partition. The node i and its ancestors are then removed from the dendrogram. The algorithm is continued until there is no leaf left in the dendrogram.

We noticed that the first method ($Best$) promotes partitions composed of large clusters while the second one (Asc) produces small clusters.

3. EXPERIMENTS AND RESULTS

3.1. Database

The proposed approaches were experimented on a subset of the *2-speakers* data used during NIST 2001 evaluation campaign [8]. The *2-speakers* segmentation reference for each test file was available. This subset was composed of 408 telephone speech conversations extracted from the Switchboard II corpus. The number of speakers was 319 (132 males and 187 females). Each speaker appeared in 1 to 4 tests (see Table 1). Each speaker utterance had a duration of 31 seconds on average (Min =∼14 s, Max =∼53 s.). The total duration of the database was close to 422 minutes.

Speakers appearing in	1 test	2 tests	3 tests	4 tests
Number of speakers	72	90	64	93

Table 1. *Number of speakers appearing in 1 to 4 tests.*

For information, in the NIST 2001 evaluation, the best segmentation system did ∼10% of error rate for the *2-speakers* task. In our experiments, we used the *2-speakers* segmentation given by NIST as reference, that is, corresponding to a system which would do 0% of error rate.

A world/background model was also trained on a different data set and used to adapt the speaker models and normalize the scores computed with each speaker model. This subset was composed of 472 Switchboard II tests uttered by 100 speakers (both males and females).

3.2. Automatic Speaker Recognition System

The acoustic parameterization (16 cepstral coefficients and 16 Δ-cepstral coefficients) was carried out using the SPRO module developed by the ELISA consortium [9].

The speaker models and likelihoods were calculated by the AMIRAL automatic speaker recognition system developed at LIA [10]. The speakers were modeled by Gaussian Mixture Models (GMM) with 128 components and diagonal covariance matrices [11], adapted from the world/background model. The adaptation scheme was based on the *maximum a posteriori* method (MAP).

3.3. *1-Speaker* Experiments

1-Speaker Verification Evaluation

We first evaluated the accuracy of the dissimilarity measures. Therefore, a cross-verification test was performed between two speaker utterances. The similarity score, used as the verification score, was expressed as the opposite of the dissimilarity measure, and calculated between each available pairs of speaker utterances: $s(u, v) = -d(u, v)$. This score was calculated for each dissimilarity measure available (d_{clr}, d_1 with $f = l$ or $f = r$, and d_2 with $f = l$ or $f = r$). The verification score was also computed only if the utterances u and v came from different audio documents. i.e. $d(u, v) \neq +\infty$.

Results and Discussion

The *1-Speaker* verification results shown in Figure 5 were obtained with 332,112 tests. The similarities calculated with $f = l$ were not normalized by a background model. Therefore, log-likelihood results are worse than log-likelihood ratio results.

All error rates are higher than the results obtained for the *1-speaker* verification task of the NIST 2001 evaluation (equal error rate of ∼10%). There are two major differences between our experiments and the NIST 2001 *1-Speaker* verification results. First, in our experiments, the duration of training utterances was 31 seconds on average, whereas, for the NIST campaign, 2 minutes of speech were available to train each speaker model. Second, for the NIST verification results, a well known normalization (H-norm) was performed before the scoring phase.

s_1 (with $f = r$) gives better results than s_2 (with $f = r$). Using the data of the other utterances present in the two tested audio documents improves the performance of the similarity measure.

Finally, s_1 (with $f = r$) outperforms s_{clr} at the equal error rate.

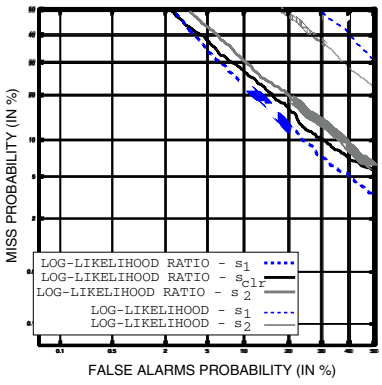

Fig. 5. *DET curve of 1-Speaker Verification test based on similarity measures.*

[1] called "Nearest Neighbor Purity Estimator"

3.4. Speaker Utterances Tying Experiments

Evaluation

The evaluation of speaker utterances tying was processed on the partition obtained after the pruning step using the two error rates proposed in [1].

Let N_c be the number of clusters in partition P.

Let n_i be the number of utterances in cluster i.

Let IN_i be the number of utterances of the main speaker[2] inside the cluster i.

Let OUT_i be the number of utterances of the main speaker outside the cluster i.

The type-I error rate is expressed by:

$$e_I = \frac{1}{N_c} \sum_{i \in P} \frac{n_i - IN_i}{n_i}$$

The type-II error rate is expressed by:

$$e_{II} = \frac{1}{N_c} \sum_{i \in P} \frac{OUT_i}{IN_i + OUT_i}$$

Type I and II error rates are summarized by : $e = e_I + e_{II}$

Results and Discussion

In our experiments, we note that e and N_c scores are linked according to the parameter Q used in the pruning methods. A decreasing of the cluster number N_c produces more errors (i.e. e grows).

Table 2 presents the results of the two agglomerative methods. In both cases, e scores between complete linkage and single linkage are close, but N_c, the number of clusters, is nearest to the real number (319) with complete linkage.

	Complete Linkage				Single Linkage			
	Asc		Best		Asc		Best	
	e	N_c	e	N_c	e	N_c	e	N_c
0.0	63.7	596	68.6	519	64.3	643	65.1	636
0.5	67.5	537	72.8	449	70.0	598	70.6	588
1.0	70.0	505	83.5	350	77.0	544	84.5	484

Table 2. Complete vs. Single linkage; for d_{clr} measure with $Q \in \{0, 0.5, 1\}$; $e = e_I + e_{II}$ reported in %.

Table 3 presents the results of the dissimilarity measures. The cross likelihood ratio d_{clr} is the dissimilarity that produces the best e and the number of clusters the closest to the real number. The number of clusters ($N_c = 596$) is important compared to the real number (319) but e_I is low ($\sim 7\%$). Choosing a higher value for Q produces a partition composed of less clusters (see table 2). Although d_1 performs better than d_{clr} for the *1–Speaker* experiments, this is not the case anymore for the speaker utterances tying experiments.

The results of the pruning methods are expected: the Asc method produces smaller clusters and the $Best$ method larger clusters. Moreover, the error e is very close for a given number of clusters, whatever the pruning method is. The choice of the pruning method will depend on the application.

[2]the speaker identity which minimizes r.

	Pruning Asc				Pruning Best			
	e_I	e_{II}	e	N_c	e_I	e_{II}	e	N_c
d_2 llk	1.4	66.5	67.9	790	27.7	63.0	90.7	540
d_1 llk	0.6	67.2	67.8	808	28.1	63.5	91.6	546
d_2 llr	6.6	63.8	70.4	708	21.4	60.3	81.7	563
d_1 llr	5.5	59.7	65.2	646	14.6	54.6	69.2	529
d_{clr}	7.4	56.3	63.7	596	15.1	53.5	68.6	519

Table 3. *Results of the various dissimilarity measures for complete linkage. $Q = 0$ and e, e_I, e_{II} in %. llk = log-likelihood, llr = log-likelihood ratio.*

4. CONCLUSION AND PERSPECTIVES

In this paper, we evaluated the potentiality of hierarchical classification approaches for speaker utterances tying, in the framework of speaker indexing. We applied hierarchical classification approaches classically used in speaker clustering to the new problem of speaker utterances tying. We also proposed two new dissimilarity measures to take into account all the information present in one speaker-based segmented audio file and a new bottom-up classification algorithm. The best system produces a speaker utterance tying with a very good cluster purity but with a high number of clusters.

Future work will focus on normalizing discriminant dissimilarity measures. Experimentation of a complete indexing system, associating an automatic speaker segmentation phase and a speaker tying phase is also planned.

5. REFERENCES

[1] S. Chen, J.F. Gales, P. Gopalakrishnan, R. Gopinath, H. Printz, D. Kanevsky, P. Olsen, and L. Polymenakos, "IBM's LVCSR system for transcription of broadcast news in the 1997 HUB4 english evaluation," in *DARPA speech recognition workshop*, 1998, http://www.nist.gov/speech/publications/darpa98/html/bn20-/bn20.htm.

[2] A. Solomonoff, A. Mielke, M. Schmidt, and H. Gish, "Clustering speakers by their voices," in *Proceedings of ICASSP 98*, 1998.

[3] D.A. Reynolds, E. Singer, B.A. Carlson, J.J. McLaughlin G.C. O'Leary, and M.A. Zissman, "Blind clustering of speech utterances based on speaker and language characteristics," in *Proceedings of ICSLP 98*, 1998.

[4] Sylvain Meignier, Jean-François Bonastre, and Stéphane Igounet, "E-HMM approch for learning and adapting sound models for speaker indexing," in *2001 : a Speaker Odyssey*, June 2001, pp. 175–180.

[5] L. Wilcox, D. Kimber, and F. Chen, "Audio indexing using speaker identification," *Proceedings of SPIE 94*, pp. 149–157, 1994.

[6] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, Sept. 1999.

[7] B.S. Everitt, *Cluster Analysis*, Oxford University Press, New York, third edition, 1993.

[8] NIST, "The NIST 2001 speaker recognition eveluation plan," http://www.nist.gov/speech/tests/spk/2001/doc/2001-spkrec-evalplan-v53.pdf, Mar. 2001.

[9] Ivan Magrin-Chagnolleau, Guillaume Gravier, and Raphaël Blouet for the ELISA consortium, "Overview of the ELISA consortium research activities," in *2001 : a Speaker Odyssey*, June 2001, pp. 67–72.

[10] Corinne Fredouille, Jean-François Bonastre, and Teva Merlin, "Amiral: a block-segmental multi-recognizer approach for automatic speaker recognition," *Digital Signal Processing*, vol. 10, no. 1–3, pp. 172–197, January/April/July 2000.

[11] Douglas A. Reynolds, "Speaker identification and verification using gaussian mixture speaker models," *Speech Communication*, pp. 91–108, Aug. 1995.