BLIND CLUSTERING OF SPEECH UTTERANCES BASED ON SPEAKER AND LANGUAGE CHARACTERISTICS

D.A. Reynolds, E. Singer, B.A. Carlson, G.C. O'Leary, J.J McLaughlin and M.A. Zissman

M.I.T. Lincoln Laboratory
Lexington, MA 02420-9108 USA
{dar,es,gco,jackm,maz}@sst.ll.mit.edu

ABSTRACT

Classical speaker and language recognition techniques can be applied to the classification of unknown utterances by computing the likelihoods of the utterances given a set of well trained target models. This paper addresses the problem of grouping unknown utterances when no information is available regarding the speaker or language classes or even the total number of classes. Approaches to blind message clustering are presented based on conventional hierarchical clustering techniques and an integrated cluster generation and selection method called the d^* algorithm. Results are presented using message sets derived from the Switchboard and Callfriend corpora. Potential applications include automatic indexing of recorded speech corpora by speaker/language tags and automatic or semiautomatic selection of speaker specific speech utterances for speaker recognition adaptation.

1 INTRODUCTION

This paper addresses the general task of automatic grouping of unlabeled speech messages based on either the identity of the speaker of the message or the language spoken. In the conventional speaker or language recognition tasks, models for specific speakers or languages are trained using data from a labeled corpus and the likelihoods of unknown test utterances are computed from the models. The class of the unknown utterance is selected based on a comparison of the likelihood scores. In this paper we generalize the task to the case where no knowledge of the composition of the set of utterances, in terms of either the attribute classes (speaker identities or languages) or even the number of classes, is available. The goal of the system is to automatically partition the unlabeled speech messages into clusters based on a common attribute class. Ideally, each cluster would comprise all messages spoken by a specific speaker (for the speaker clustering task) or all messages spoken in a specific language (for the language clustering case). Applications of blind message clustering include automatic indexing of recorded speech corpora by speaker/language tags and automatic or semiautomatic selection of speaker-specific speech utterances for speaker recognition adaptation.

This work was supported by the Department of the Air Force. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Air Force.

2 SYSTEM OVERVIEW

The block diagram in Figure 1 shows the major components of the blind message clustering system. The task of the system can be formally described as follows: Given a collection of N speech messages $m_1, m_2, ..., m_N$ from N_s attribute classes (that is, N_s speakers or N_s languages), produce a partitioning of the messages into a set of N_c clusters $c_1, c_2, ...c_{N_c}$. Ideally, $N_c = N_s$ with each cluster containing all the messages associated with one and only one attribute class. The system first produces an inter-message distance matrix based on the attribute of interest (speaker or language in this paper). A set of candidate clusters is generated from the distances, and a final partition of the input messages is obtained by searching through the candidates and selecting the set that maximizes an appropriate evaluation metric. We now describe each of the system components in greater detail.

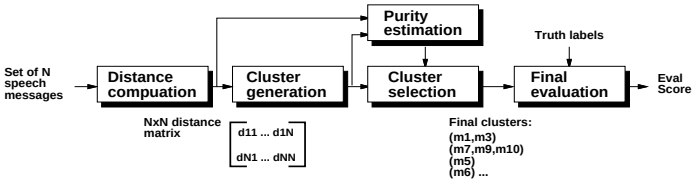

Figure 1: Block diagram of blind clustering tasks.

2.1 Distance computation

Techniques such as Gaussian mixture modeling for speaker recognition and interpolated language modeling of phone sequences for language identification have yielded state-of-the-art performance in supervised recognition tasks and so formed the basis of the inter-message distance computation. The goal of this system component is to generate inter-message distances (or more properly, dissimilarities) which are small when the messages are of the same attribute class and large otherwise.

2.1.1 Speaker attribute

The distance computation for the speaker attribute is based on the adapted GMM speaker recognition method described

in [1]. For blind clustering, adapted GMMs are obtained from the cepstral representations of all messages and likelihoods $l(m_i|\lambda_j)$ are computed for all (m_i, λ_j) pairs, where m_i is message i and λ_j is the GMM formed from message j. An inter-message distance matrix is formed using the symmetric Cross Likelihood Ratio d_{ij} where

$$d_{ij} = \log \frac{l(m_i|\lambda_B)}{l(m_i|\lambda_j)} + \log \frac{l(m_j|\lambda_B)}{l(m_j|\lambda_i)} \quad (1)$$

and λ_B is the Universal Background Model [1]. The Cross Likelihood Ratio distance measure has the property that $d_{ij} = d_{ji}$; however, $d(i,i) \neq 0$ and there is no guarantee that the triangular inequality is obeyed.

2.1.2 Language attribute

The distance computation for the language attribute uses the Phone Recognition followed by Language Modeling technique of language recognition described in [2]. Language models are obtained from phone sequence representations of all messages and duration-normalized likelihoods $l(m_i|\lambda_j)$ are computed for all (m_i, λ_j) pairs, where m_i is message i and λ_j is the interpolated language model formed from message j. Inter-message distances are computed using the symmetric Cross Entropy distance d_{ij} where

$$d_{ij} = \log \frac{l(m_i|\lambda_i)}{l(m_i|\lambda_j)} + \log \frac{l(m_j|\lambda_j)}{l(m_j|\lambda_i)} \quad (2)$$

For the Cross Entropy distance measure, $d_{ij} = d_{ji}$ and $d(i,i) = 0$; however, the triangle inequality is not obeyed.

2.2 Cluster generation and selection

The inter-message distance matrix is the input to the cluster generation and selection module where the messages are clustered subject to an evaluation measure which scores the "goodness" of a partition. In this work we use the BBN metric [3] which is designed to give better scores to partitions having large, pure clusters than to ones having smaller, impure clusters. The BBN metric is given by

$$I_{BBN} = \sum_{i=1}^{N_c} n_i p_i - Q N_c \quad (3)$$

where n_i is the number of messages in candidate cluster i, N_c is the number of candidate clusters in the selected partition, and p_i is the purity of cluster i. Purity is defined as $p_i = \sum_{j=1}^{N_s} (\frac{n_{ij}}{n_i})^2$, where N_s is the number of attribute classes (number of speakers or languages) and n_{ij} is the number of messages of attribute class j in cluster i.

The variable Q is a system design parameter that controls the degree to which fewer, larger clusters are favored at the expense of decreased purity. With small values of Q, greater weight is assigned to cluster purity and messages of the same class will tend to be dispersed among several clusters. For higher values of Q, greater weight is attached to limiting the total number of clusters; however, each cluster is more likely to contain messages of different attribute classes. Throughout this paper the value of Q will be set to 0.5. Two approaches to the problem of cluster generation and selection will be presented in Section 3.

2.3 Evaluation

The BBN metric is useful for comparing systems on a specific data set. To compare performance across data sets we define the *clustering efficiency* as a normalized BBN metric given by

$$\text{Clustering efficiency} = \frac{I_{BBN}(C) - F}{O - F} \quad (4)$$

where $I_{BBN}(C)$ is the BBN metric value for partition C, F is the BBN metric value for the full (one message per cluster) search, and O is the BBN metric value for the optimum (true) partition[1]. Clustering efficiency thus describes system performance as a fraction of $O - F$. Use of the term "efficiency" is somewhat inappropriate since it is possible to select a partition whose clustering efficiency is negative (i.e., $I_{BBN}(C) < F$).

2.4 Purity estimation

Evaluation of the BBN metric requires knowledge of the internal composition of the clusters in order to compute the cluster purities p_i. In practice, of course, the data is unlabeled and the purity is not known. To evaluate the BBN metric on unlabeled data we use the nearest neighbor purity estimator described in [3]. Briefly, the algorithm operates as follows: For each of the n_i elements of candidate cluster i, identify the nearest neighbors based on distance and calculate the fraction of the n_i nearest neighbor messages that are members of cluster i. Then $\hat{p}_i$, the nearest neighbor purity estimate of cluster i, is the average of the n_i fractions. In practice it has been observed that this purity estimator underestimates purities for small clusters and overestimates purities for large clusters. Recently, BBN has obtained improved clustering efficiency performance by adding a size dependent bias to the purity estimates [6].

3 CLUSTER GENERATION AND SELECTION

3.1 Hierarchical clustering

Hierarchical clustering is a well-known method for generating candidate clusters from a distance matrix [4]. In agglomerative clustering, all messages start as singleton clusters and are iteratively combined in a minimum-distance, pairwise fashion until only a single cluster containing all the messages exists. In divisive clustering, all messages start in a single cluster which is iteratively split using a best-split criterion until each cluster contains exactly one message.

A tree produced by either type of hierarchical clustering contains exactly $2N-1$ candidate clusters (this includes the N singletons clusters). Two methods were evaluated for selecting the best partition of the candidate clusters. The simplest method, called level cutting, is equivalent to slicing the tree horizontally at each merge level. The partition selected is the one whose BBN metric, using estimated purity values, is maximum.

Level cutting does not necessarily identify the best possible partition of a set of candidate clusters since the best partition does not, in general, occur at a horizontal cut.

[1] It is easy to show that $F = N(1-Q)$ and $O = N - N_s Q$, where N is the total number of messages [3].

Better scoring partitions may be obtained by combining clusters (nodes) occurring at different levels of the tree, but consideration of all possible combinations of nodes in a tree is computationally prohibitive. To make the non-level cutting computationally tractable, we implemented a sequential, best-first search technique, wherein the "best" scoring node is selected from the tree, the node's descendants and ancestors are removed from further consideration, and the selection continues until no nodes are left in the tree. To preserve the effect of the parameter Q when comparing clusters, we have defined the score of each cluster in the tree as the per-message cluster value $\hat{p}_i - Q/n_i$. Using the actual contribution of a cluster, $n_i\hat{p}_i - Q$, tends to favor very large but impure clusters in selection. We expect that this non-level cutting algorithm will identify a partition whose clustering efficiency is at least as good as, and potentially better than, that produced by level cutting.

3.2 d^* clustering

The d^* clustering algorithm is an integrated cluster generation and selection strategy that partitions the data into hyperspheres of radius d. The algorithm is based on the notion that the distribution of distances between messages of the same attribute class will be well separated from the distribution of distances between messages of different attribute classes. Thus, in general, same-attribute-class messages should be tightly grouped together. If we can locate a centroid for one of these groups, we should be able to create a pure cluster by retaining all messages which lie within a radius of d from the center, where d sets the tradeoff between not capturing same-attribute-class messages (misses) and accepting different-attribute-class messages (false alarms). In the first version of the algorithm, called the fixed d^* algorithm, it is assumed that an "optimum" value of d (d^*) has been selected *a priori*, perhaps by analyzing distance distributions from some development messages. Every message is treated as a candidate centroid with cluster members consisting of messages within radius d^*. We identify the first cluster as the one with the highest per-message cluster value $\hat{p}_i - Q/n_i$. The members of this first cluster are removed from further consideration and these steps are repeated with the remaining $N - n_1$ messages to identify the second cluster, continuing until all messages are clustered.

The variable d^* algorithm is a generalization of the fixed approach and does not require choosing a radius *a priori*. Each message is assumed to be the centroid of several candidate clusters whose radii vary from d_{min} to d_{max}. In our experiments, d_{min} is set to the $0.1th$ percentile and d_{max} to the $10th$ percentile of the distance distribution with 10 equal-spaced candidate radii between them. With N objects and n_d values of d, there will be $N*n_d$ initial candidate clusters from which the one with the highest $\hat{p}_i - Q/n_i$ is selected. Members of the selected cluster are removed from consideration and the next best cluster is selected. The algorithm proceeds until all messages are clustered. Results presented in this paper are for the variable d^* clustering algorithm only. Note that d^* clustering is similar to the partitioning class of clustering algorithms [4].

4 DATABASES

4.1 Speaker

The speech data used for speaker attribute clustering consisted of 1369 messages chosen from the 1996 NIST speaker recognition evaluation target test set [5] which is derived from the Switchboard telephone speech corpus. The messages are nominally 30 seconds in duration and contain speech from one of 397 speakers (225 males in 853 messages and 172 females in 516 messages). The number of messages spoken by each speaker ranged from 27 (2 speakers) to 1 (41 speakers).

4.2 Language

The speech data use for language attribute clustering consisted of 783 10-minute messages chosen from the Callfriend telephone speech corpus [5]. The messages were spoken in 12 languages by both male and female speakers and the conversation sides were summed. Thus, each message contained 2 male speakers, 2 female speakers, or 1 male and 1 female speaker. No speaker appeared more than once in the data set. Phone sequences were obtained by processing the messages with an HMM-based English phone recognizer trained on the OGI Language-ID speech corpus [2]. The phones were further labeled as being of long or short duration using the tagging scheme described in [2]. We tested the blind message clustering system on the full 12-language 783-message set ("ALL") as well as on two 4-language subsets ("ASIA" and "EURO"); see Figure 2.

LANGUAGE	MESSAGES	SUBSETS		
		"ALL"	"ASIA"	"EURO"
ARABIC	56	X		
ENGLISH	92	X	X	X
FARSI	55	X		
FRENCH	59	X		X
GERMAN	58	X		X
HINDI	58	X		
JAPANESE	59	X	X	
KOREAN	55	X		
MANDARIN	91	X	X	
SPANISH	91	X		X
TAMIL	54	X		
VIETNAMESE	55	X	X	
TOTAL		783	297	300

Figure 2: Composition of language subsets.

5 EXPERIMENTS AND RESULTS

5.1 Distance measure evaluation

The key to successful clustering is the ability of the distance measure to produce smaller values for distances between messages of the same attribute class and larger values for distances between messages of different attribute classes. We have found it useful to evaluate the distance measures we are using by plotting the separation of these two types of distances using the Detection Error Tradeoff (DET) curve. For this purpose all distances were classified as either same (distance between messages of the same attribute class) or different (distance between messages of different attribute classes). We then swept out an attribute-class independent

threshold over the ordered distances and plotted false rejection vs. false acceptance probabilities. The distributions and DET curves for both the speaker and language sets are shown in Figure 3. These plots lead us to expect much better clustering performance for the speaker attribute than for the language attribute.

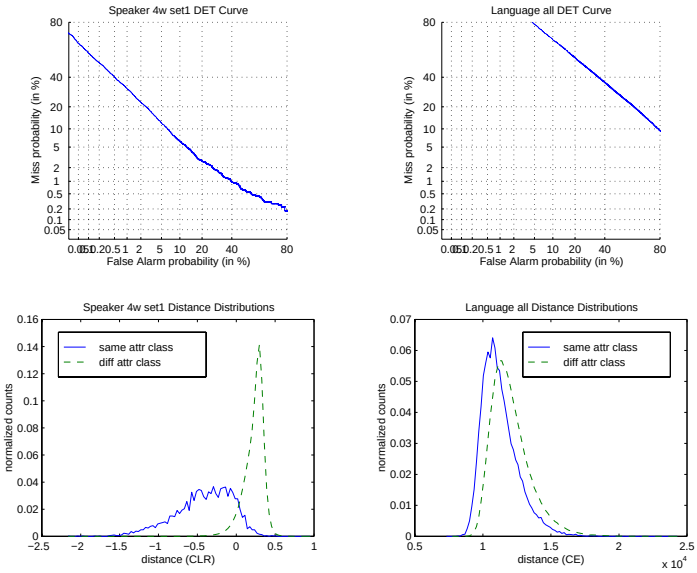

Figure 3: DET curves and distributions for speaker and language data. DET curves plot miss vs. false alarms. Distributions show same (solid) vs different (dashed) distances.

5.2 Clustering efficiency

Table 1 presents the clustering efficiency results for three blind message clustering approaches using the 1369-message Switchboard speaker set (SW_1369) and the three Callfriend language sets (CF_all, CF_asia, CF_euro). We used the agglomerative method of hierarchical clustering for these tests since experiments indicated that it performed better than the divisive method. Non-level cutting results are shown only for the speaker attribute, as explained in Section 6. Clustering performance for the speaker attribute is much better than performance for language attributes, as predicted by the DET curves. Results show that the best clustering performance as measured using the BBN metric was obtained using the d^* clustering algorithm[2].

Attribute	Data	Hierarchical Level	Hierarchical Non-level	d^*
Speaker	SW_1369	0.509	0.574	0.611
Language	CF_all	0.044	–	0.122
	CF_asia	0.069	–	0.182
	CF_euro	0.065	–	0.242

Table 1: Clustering efficiency for speaker and language data.

[2]Using the improved biased purity estimator, BBN has obtained efficiencies for level cutting hierarchical clustering that are higher than the scores for d^* shown in Table 1 [6].

6 DISCUSSION

In both the d^* algorithm and the non-level cutting version of hierarchical clustering, estimated purities are employed as part of the sequential selection of the final cluster set. Because of the nature of the nearest neighbor purity estimator, additional constraints must be imposed on the selection process in order to avoid selecting poor, but apparently pure, clusters. This problem arises because a cluster comprising a large subset of messages, regardless of its composition, will be relatively isolated with respect to the entire set of messages and will be regarded as "pure" by the purity estimator.

For the non-level cutting approach this problem can be avoided by selecting a maximum tree level above which searches for candidate clusters are forbidden. This restriction translates to selecting a minimum number of clusters for the partition. Although the approach worked well for the speaker case, it was difficult to quantify and did not generalize well for language clustering. For this reason, the non-level cutting approach was not pursued further. For the variable d^* algorithm this problem is avoided by rejecting any candidate cluster containing more than 10%[3] of the messages in the data set. Resulting partitions will contain at least 10 clusters, a suboptimal outcome for data containing fewer than 10 attribute classes. The impact of this rule depends on the value of Q: For small Q, fragmentation of messages of the same attribute class into several clusters may have little effect (see Eq. 3); for large Q, fragmentation may be costly.

7 CONCLUSIONS AND FUTURE WORK

This paper has examined an extension of the classic supervised speaker and language recognition tasks to an unsupervised clustering task. Application of distance measures based on state-of-the-art speaker and language recognition techniques along with the integrated cluster generation and selection d^* algorithm were found to be very effective (in terms of clustering efficiency) for the speaker attribute but rather limited for the language attribute. Gains in language clustering will likely require better language attribute distance measures and purity estimators. Future work for speaker clustering will focus on extending the current approach to work with multi-speaker utterances.

REFERENCES

[1] D. A. Reynolds, "Comparison of background normalization methods for text-independent speaker verification," *Proc. Europseech '97*, pp. 963-966, September 1997.

[2] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech", *IEEE Trans. Speech and Audio Proc.*, SAP-4(1):31-44, January 1996.

[3] A. Solomonoff, A. Mielke, M. Schmidt, H. Gish, "Clustering speakers by their voices", *Proc. ICASSP '98*, 757-760, May, 1998.

[4] L. Kaufman and P. J. Rousseuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley, 1990.

[5] LDC, http://www.ldc.upenn.edu; 1996 NIST Speaker Recognition Evaluation Kit, CallFriend Language ID Corpus.

[6] Michael Schmidt, personal communication.

[3]Empirically determined value.