Bootstrapping a Spoken Language Identification System Using Unsupervised Integrated Sensing and Processing Decision Trees

Shuai Huang [†1], Damianos Karakos [†‡2], Glen A. Coppersmith [†‡3], Kenneth W. Church [†‡4], Sabato Marco Siniscalchi [*5]

[†] *Center for Language and Speech Processing, Johns Hopkins University*
[‡] *Human Language Technology Center of Excellence, Johns Hopkins University*
[1−4] {shuaihuang,damianos,coppersmith,Kenneth.Church}@jhu.edu

[*] *Kore University of Enna, Enna, Italy*
[5] marco.siniscalchi@unikore.it

***Abstract*—In many inference and learning tasks, collecting large amounts of labeled training data is time consuming and expensive, and oftentimes impractical. Thus, being able to efficiently use *small* amounts of labeled data with an abundance of unlabeled data—the topic of semi-supervised learning (SSL) [1]—has garnered much attention. In this paper, we look at the problem of choosing these small amounts of labeled data, the first step in a bootstrapping paradigm. Contrary to traditional active learning where an initial trained model is employed to select the unlabeled data points which would be most informative if labeled, our selection has to be done in an *unsupervised* way, as we do not even have labeled data to train an initial model.**

We propose using unsupervised clustering algorithms, in particular integrated sensing and processing decision trees (ISPDTs) [2], to select small amounts of data to label and subsequently use in SSL (e.g. transductive SVMs). In a language identification task on the CallFriend[1] and 2003 NIST Language Recognition Evaluation corpora [3], we demonstrate that the proposed method results in significantly improved performance over random selection of equivalently sized training data.

I. Introduction

The goal of spoken language identification (LID) is to identify the language of a segment of spoken utterance automatically, based on features extracted from the speech [4]. As a motivating application, consider a customer service for a multilingual population which needs to route calls to an appropriate operator. Training a LID system consists of extracting features from speech appropriate for discriminating between languages and learning to weight those feature appropriately.

There are cases, however, where a LID system encounters speech from new (surprise) languages that it was not trained on. In such cases, one should be able to quickly acquire labels for a few utterances and be able to bootstrap a new LID system for those languages. Once an initial seed set has been collected, additional large amounts of unlabeled data can be used to improve the models, as is done in semi-supervised learning (SSL) [1]. Acquiring labeled data is the topic of *active learning* (see [5] and references therein). Traditional active learning assumes the existence of a learned model which guides the selection of unlabeled data to annotate, but here we are concerned with the (harder) task of coming up with an initial set of labels—we assume that we start without any supervision. We also assume that we have a limited budget for annotation and therefore must proceed with minimal labeled data in the domain of interest; we hope that using additional unlabeled data in a semi-supervised setting will allow us to get around this limitation. As we describe in a later section, our approach is to utilize an unsupervised clustering algorithm, and then select data to annotate based on the confidence of the clustering. The motivation behind this approach is consistent with the cluster assumption of SSL, which requires that data which belong to different classes tend to form clusters.

The attribute-based approach proposed by [6] (which we make use of in this paper) explores the universal acoustic phonetic features of speech with state-of-the-art performance results. Specifically, using HMMs and a language-dependent to language-independent mapping procedure, raw speech is converted into "manner" and "place" of articulation attributes which are subsequently used to train HMM-based and SVM-based identification systems. LID is essentially performed by detecting high-probability sequences (n-grams) of such features that correlate with each language. As mentioned above, training a LID system in a semi-supervised setting is done with collections of labeled and unlabeled data. Note that by "training data" we do not mean data useful for learning how to generate the articulatory features; in fact, the approach mentioned in [6] does not require the articulatory attribute generator to be trained on the languages that it will eventually be applied on.

The paper proceeds as follows: a review of the attribute-based approach [6] is given in Section II. A description of the proposed active learning approach is provided in Section III. Unsupervised integrated sensing and processing decision trees (ISPDTs) and other clustering algorithms are

[1] http://www.ldc.upenn.edu/Catalog/byType.jsp#speech.telephone

described in Section IV. Experimental results on the 2003 NIST Language Recognition Evaluation corpora using ISPDTs and other unsupervised clustering algorithms appear in Section V. Concluding remarks appear in Section VI.

II. Attribute-based approach for Spoken Language Identification

In [6], "manner" and "place" of articulation "attributes" were proposed as a universal acoustic characterization of all spoken languages. Manner contains six items: vowel, fricative, nasal, approximant, stop, and silence; and place contains ten items: coronal, dental, glottal, high, labial, low, mid, palatal, silence, and velar. Attribute transcriptions are obtained from phonemic transcripts via application of a phoneme-to-manner mapping table and a phoneme-to-attribute mapping table as is done in [6]. For each document, the feature vector is then created by collecting up to 4-gram statistics of the "attributes" items. Specifically, manner-based and place-based transcriptions are obtained for each document (D_i for $i \in (1, \cdots, N)$), and each transcription is then converted into an M-dimensional feature vector V_i by concatenation of the two term-count vectors, V_i^m and V_i^p; therefore[2], $M = M_m + M_p = 12664$. The feature vectors are also weighted according to the formula in [6],

$$w_{i,j} = \left[1 + \frac{1}{\log N} \sum_{i=1}^{N} \frac{n_{ij}}{n_{\cdot j}} \log \frac{n_{ij}}{n_{\cdot j}}\right] \frac{n_{ij}}{n_{i\cdot}} \tag{1}$$

where n_{ij} is the number of times term j occurs in document D_i, and $n_{\cdot j}$ is the number of times that term j appears in all the N documents, and $n_{i\cdot}$ is the number of terms in document D_i.These document vectors are then used to train spoken language classifiers (i.e., SVM).

The LID system employed in this paper is the same as the one used in [6], where a 1-versus-all multi-class SVM system is trained, such that for an individual target language, a separate SVM is trained with positive class consisting of the target language and the negative class consisting of all other languages. For LID tasks, the language identity is decided based on the maximum positive distance from the separating hyperplane [7].

III. Unsupervised Bootstrapped LID

Traditionally in active learning, the task is to annotate new data to improve upon an existing classifier. However, as we mention in Section 1, we aim to annotate an initial set of data to *bootstrap* a classifier; the selection of data to annotate will have to be done in an unsupervised manner. The baseline (naïve) way of doing this would be to select data randomly—we expect this to be a viable solution only when it is acceptable to annotate a large amount of data. Under a limited data annotation budget, however, the random sampling may introduce bias (or fail to correct for) in the learned labels.

[2]Since we are collecting up to 4-gram statistics of the attributes, the length of the attribute vector $M' = p + p^2 + p^3 + p^4$, where p is the number of attributes. For manner and place respectively we have $p_m = 6$ and $p_p = 10$, thus $M_m = 1554$ and $M_p = 11110$.

Our approach for obtaining labels is to first cluster the unlabeled data and then select data to annotate based on this clustering. Our argument in favor of this approach is similar to the *cluster assumption*[3] [1], but goes in the opposite direction: once the data have been clustered, the most confidently clustered data (e.g., those closest to each cluster centroid) are the ones with the strongest likelihood of belonging to different classes and are thus the "best" candidates for labeling.

This proposed approach differs significantly from the usual practice in active learning of selecting data to be labeled close to the boundary between classes. Since we lack a pre-existing classifier, we lack boundaries, so this approach is not possible. The (hypothesized) boundary between the clusters only *loosely* corresponds to the true boundary between the classes. Furthermore, in the case of bootstrapping with very small amounts of labeled data, picking data close to the boundary between the clusters one risks annotating data that lie on *opposite sides of the boundary*[4]. Bootstrapping a LID system with such labeled data will most certainly lead to learning the wrong model, one that classifies the (+)'s as (-)'s and vice-versa.

Unsupervised active learning has also been done in [8]. Our approach differs from [8] where the unsupervised clustering is done on data for which labels are *known*, and the active learning is done by selecting additional data which do not fall into any of the clusters. Despite noted similarities to [8], we do not assume the existence of labeled data; even if we applied the method of [8] using clusters that belong to a different set of languages (for which we do have labels), there is no guarantee that unlabeled data that do not fall into any of the clusters are the most informative.

In our experimental setup, we assume that the unlabeled data $v_1, v_2, \cdots, v_N$ are a mixture of L different languages: $l_1, \ldots, l_L$. We have experimented with $L = 2$ (binary classification) and $L = 3$ (three-way classification); thus, we do binary (resp. three-way) clustering in the first step. This can be easily extended to the case where there are more than 3 languages present, where a 1-versus-all multi-class SVM is trained. Once the unlabeled data have been clustered into L clusters, $C_1, \ldots, C_L$, we compute the centroids as follows:

$$c_i = \frac{1}{|C_i|} \sum_{v_k \in C_i} v_k, i \in \{1, \ldots, L\}. \tag{2}$$

Next, based on these centroids, we select the data to annotate $C'_1, \ldots, C'_L$ according to a similarity measure function; specifically, we employ the Euclidean distance as the similarity measure $s(k, i) = \|v_k - c_i\|, i \in \{1, \ldots, L\}$ between the data point v_k and the centroid c_i.

[3]As mentioned in [1], one of the reasons for the success of SSL is the *cluster assumption*, i.e., the fact that data that belong to different classes tend to form *clusters*. Thus, once some supervised data are available, the SSL learner is in a position to confidently *"propagate"* these labels to other (unlabeled) data that belong to the same cluster as the labeled data.

[4]That is, a data point with a (+) label which happens to be on the (-) side, and vice-versa (lots of points of that sort are close to the boundary–these are the misclassifications).

The K data points $v'_1, v'_2, \cdots, v'_K$ that are closest to the two centroids are "annotated" (their labels are revealed). Additionally, we have also considered a variant, where, apart from the K annotated data points, we also use another U additional data points that are close to the centroid as (noisy) labeled data, i.e., we assign labels to these U data points according to the label of the majority of the K annotated data in the same cluster C'_i.

IV. Unsupervised Clustering Algorithms

A. Unsupervised ISPDTs

Unsupervised integrated sensing and processing decision trees (ISPDTs) [2] are an unsupervised method to hierarchically cluster together data points that have similar empirical distributions by minimizing the expected value of a loss function (this is notably similar to regression trees). The main difference between ISPDTs and regular decision trees is that in ISPDTs the data are projected onto a lower-dimensional space at each internal node of the tree before splitting. The projection depends only on the data present at the corresponding node, and is optimized jointly with the "question" asked at that node, based on some local criterion.

One common non-tunable choice [9] for the local optimization criterion is minimization of $l(x^n, Q) = -\frac{1}{n}\log\left(Q\left(x_1, \cdots, x_n\right)\right)$, the negative normalized likelihood of the sequence x^n under the distribution Q. We instead prefer a tunable version of mutual information, the *Jensen-Rényi divergence* (as used in [10], [11]); the tunable parameter allows greater flexibility and more robustness to sparsity (which can yield state-of-the-art results in document categorization [10], [11]).

B. Other Unsupervised Clustering Algorithms

We compared ISPDTs to several other well known unsupervised clustering algorithms: repeated bisection (RB), globally optimal repeated bisection (GRB), agglomerative clustering (AG) [12] and information bottleneck (IB) [13]. Specifically, we choose the first three algorithms so that the resulting clustering solution optimizes the following criterion function:

$$\text{maximize } \mathcal{H} = \frac{\mathcal{I}}{\mathcal{E}} = \sum_{r=1}^{k} \frac{\|D_r\|^2}{n_r} \Bigg/ \sum_{r=1}^{k} n_r \frac{D_r^t D}{\|D_r\|} \qquad (3)$$

where k is the number of clusters, $D = \sum_{i=1}^{N} x_i$ is the sum of all data points, $D_r = \sum_{x_i \in S_r} x_i$ is the sum of all data points in the $r-$th cluster S_r, n_r is the size of S_r. This is the case where we use cosine function as the similarity measure between documents.

RB first clusters all the data points into two groups, then chooses one from the groups and bisects it. This process is repeated until all the k clusters are found; in each step, the data are bisected such that the criterion function $\mathcal{H}$ is locally maximized. The greedy nature of this procedure implies that it is only locally optimal and does not necessarily lead to globally optimal clusterings. GRB is similar to RB, yet with a globally optimized overall solution at the end. AG is a bottom-up algorithm in that it tries to find the clusters by first considering each data point to be a cluster and repeatedly merging pairs of clusters until there are k of them left; the data points are merged in such a way that the criterion function $\mathcal{H}$ is maximized.

For the information bottleneck (IB) method, the goal is to find clusters such that the mutual information between data X^n and the assigned cluster indices is as large as possible, under the constraint on the number of clusters. In this paper we follow the procedure for finding the maximizing clustering by choosing the one with the highest mutual information among many random clusterings.

V. Experimental Results

The annotation (active learning) and subsequent bootstrapping of the LID system are done using the CallFriend corpus, which is a collection of unscripted telephone conversations in 12 languages: Arabic, English, Farsi, French, German, Hindi, Japanese, Korean, Mandarin, Spanish, Tamil, Vietnamese. We use 40 sessions of each of the above languages in the experiments, and split the 12 languages into 2 sets, S_1 contains: English, German, Hindi, Japanese, Mandarin and Spanish, and S_2 contains the rest of the languages. The data in S_1 are labeled and are used in *clustering experiments* to find out which clustering algorithm works best, and those in S_2 are used in *LID experiments*. Note that our data selection approach is considered to be *unsupervised* with respect to the languages in S_2 because we do not use any of the S_2 labels to guide the selection process.

All results are reported on the 2003 NIST Language Recognition Evaluation test data corpus [3], and were performed using the 30-second setting, yielding 1280 sessions – 240 English sessions, 160 Japanese sessions, and 80 sessions for all other languages. Accordingly, the NIST corpus is divided into two sets: T_1 contains the languages in S_1 and is used as test set in clustering experiments; T_2 contains the languages in S_2 and is used as test set for the resulting LID system.

In the following experiments, we used the acoustic features from [6] to transform each spoken utterance into a high dimensional feature vector and perform LID tasks.

A. Unsupervised Clustering Experiments

In the unsupervised clustering experiments, we performed 2-way clustering of all possible 15 language pairs in T_1 using ISPDTs, with a resulting mean error rate $e_2 = 0.111$, with standard deviation $\sigma_2 = 0.104$. Likewise, for the 3-way clustering of all possible 20 language pairs, we obtain a mean error rate $e_3 = 0.199$, with standard deviation $\sigma_3 = 0.084$. The results of the 2-way clustering, by language pairs, are shown in Fig. 1, with the red line representing the baseline error rate obtained after placing all data into a single cluster (i.e., assigning the most probable label to all data).

For the other unsupervised clustering algorithms: repeated bisection (RB), globally optimal repeated bisection (GRB), agglomerative clustering (AG) and information bottleneck (IB), the first three algorithms are implemented using the Cluto

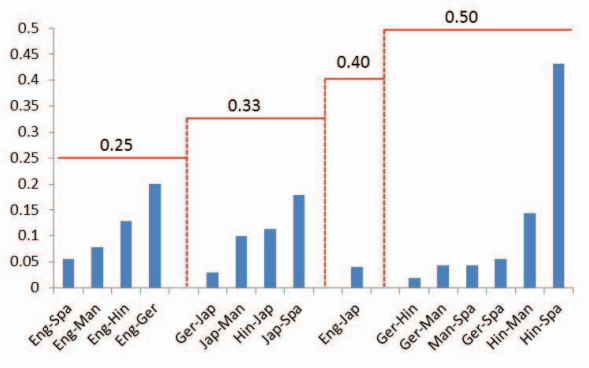


Fig. 1. Error rates from 2-way clustering using unsupervised ISPDTs, the x-axis corresponds to the 15 language pairs and y-axis corresponds to error rates.

TABLE I
ERROR RATES OF VARIOUS UNSUPERVISED CLUSTERING ALGORITHMS

Algorithm	2-way	3-way
ISPDTs	0.111 ± 0.104	0.199 ± 0.084
IB	$0.140^{*} \pm 0.117$	$0.225^{*} \pm 0.095$
RB	$0.128^{*} \pm 0.100$	$0.237^{\dagger} \pm 0.105$
GRB	$0.128^{*} \pm 0.100$	$0.230^{\dagger} \pm 0.096$
AG	$0.282^{\dagger} \pm 0.082$	$0.428^{\dagger} \pm 0.057$

clustering library [12]; each algorithm is run 10 times with cosine similarity as the similarity metric and the optimal clustering is selected according to the provided criterion function $\mathcal{H}$ mentioned above. Information bottleneck is implemented using the MATLAB package provided by [14], where we choose the parameter beta to be infinity (this setting performed best). The *mean error rate* $\pm$ *the standard deviation* ($\mu \pm \sigma$) for the above 4 algorithms along with ISPDTs are shown in Table I. The results are shown in Table I, all algorithms are statistically significantly worse than ISPDTs, as computed with a paired t-test. A $*$ indicates a statistically significant difference at the $p = 0.05$ level or lower. A $\dagger$ indicates a statistically significant difference at the $p = 0.005$ level or lower.

B. Annotation and LID Experiments

As is shown above, ISPDTs outperform the other clustering algorithms on set T_1. It is thus justifiable to use unsupervised ISPDTs to cluster the unlabeled data on set T_2 and then use the resulting clusters to collect labeled data to bootstrap the LID system.

Specifically, in the case of a mixture of 2 (resp. 3) languages, for the proposed approach, we choose K data points that are closest to each centroid after we cluster the CallFriend data into 2 (resp. 3) clusters. Furthermore, we also tried two variant approaches: (1) we annotated the K data points that were farthest away from each centroid. On average, we expect that such points will be spread more uniformly over the dataset, thus preventing biases introduced by the centroid proximity constraint; (2) we annotate the K data points that are close to the boundary.

To compare with the proposed approach, we randomly choose K utterances to annotate and do m-fold cross-validation, where the K utterances chosen in one fold are distinct from those chosen in the other fold. In this case we are using *random selection* to bootstrap our LID system, as is done in [15]. In the 2-way case, we ran the random selection experiments 1600 times for each language pair and computed the mean and standard deviation of the error rates across all language pairs.

Similar experimental settings are also used when there is a mixture of 3 languages in the training and test data, except that we ran the random selection experiments 300 times.

The experimental results of the proposed approach and random selection in mean error rate $\pm$ the standard deviation ($\mu \pm \sigma$) are shown in Table II. Fig. 2 shows a comparison of the proposed approach and the random selection (baseline), the error bars represent $\mu \pm \frac{\sigma}{2}$. Error rates of 0.0633 ± 0.0311 and 0.108 ± 0.019 are obtained in the 2-way case and 3-way case respectively when all the training data are used to train the SVMs; which can be considered as a lower bound on the error rate obtainable with the active learning and semi-supervised methods in each case. We can see that when we choose a small amount of training data to annotate in the active learning, the proposed approach gives a better result compared to the case where the selection is done randomly. This result is statistically significant with an overall p-value 2.14×10^{-4}.

For the proposed approach, in addition to the K annotated data points, we picked another U data points that are also closest to each centroid and assign labels according to the annotated data points. As it turns out, for the range of K of interest (2.5% and 5% of the dataset) the different values of U (ranging from 2.5% to 75% of the dataset) do not significantly affect the transductive SVM results.

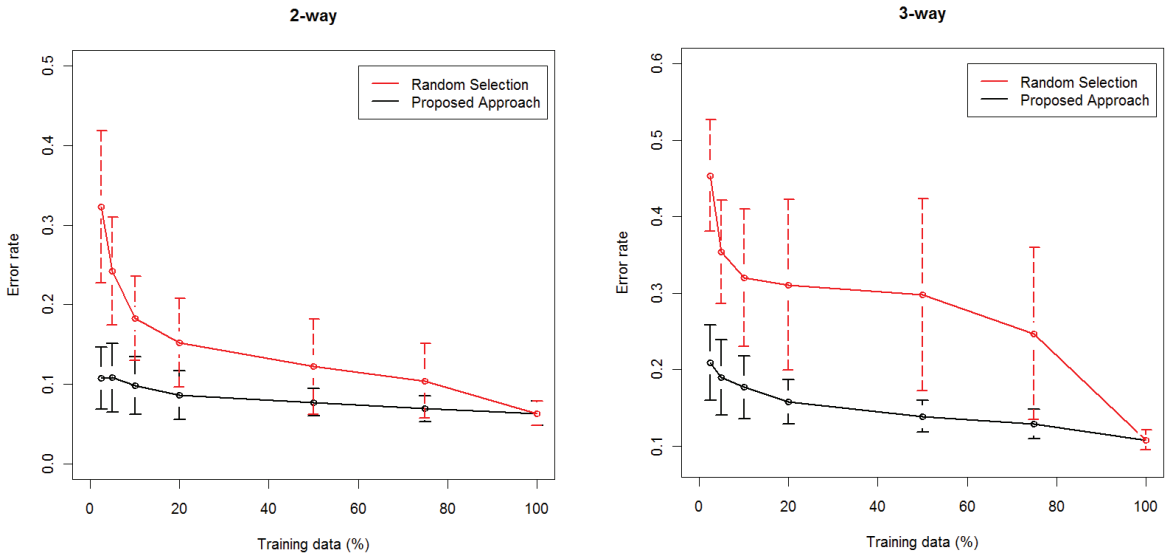
The results of the two variant approaches are shown in Table III, both variant approaches perform worse compared to the proposed approach.

VI. CONCLUSIONS

This paper proposes to use unsupervised clustering, specifically unsupervised ISPDTs [2], as a way to select small amounts of data to be annotated and used in bootstrapping a LID system. As is shown in Section IV-A, unsupervised ISPDTs give the best results on a set of development languages, and they are thus used to guide the computation of the different clusters in the remaining 6 "test" languages. We have shown in Section V that the proposed approach gives results which, in the case of very limited annotation, are superior to selecting data to annotate at random. For future work, we would like to expand the set of attributes to increase the acoustic resolution as is done in [16], use other speech features, and run experiments on more recent NIST Language Recognition Evaluation tasks. We also would like to try other approaches for active learning, using some kind of "closed-loop": once the initial data have been collected and used for bootstrapping the LID system, use the subsequent

TABLE II
ERROR RATES OF PROPOSED APPROACH AND RANDOM SELECTION

Training data	2-way		3-way	
	Proposed Approach	Random Selection	Proposed Approach	Random Selection
2.5%	0.107 ± 0.078	0.323 ± 0.191	0.209 ± 0.099	0.454 ± 0.146
5.0%	0.108 ± 0.087	0.242 ± 0.135	0.190 ± 0.099	0.354 ± 0.136
10%	0.098 ± 0.072	0.183 ± 0.106	0.177 ± 0.082	0.320 ± 0.180
20%	0.086 ± 0.061	0.152 ± 0.112	0.158 ± 0.058	0.311 ± 0.223
50%	0.077 ± 0.034	0.122 ± 0.120	0.139 ± 0.042	0.298 ± 0.252
75%	0.069 ± 0.033	0.104 ± 0.094	0.129 ± 0.038	0.247 ± 0.225
100%	0.063 ± 0.031		0.108 ± 0.019	



(a) (b)

Fig. 2. Comparison of the proposed approach and random selection (baseline)

TABLE III
ERROR RATES OF 2 VARIANT APPROACHES

Training data	2-way		3-way	
	Variant Approach 1	Variant Approach 2	Variant Approach 1	Variant Approach 2
2.5%	0.288 ± 0.170	0.364 ± 0.164	0.377 ± 0.136	0.454 ± 0.131
5.0%	0.234 ± 0.154	0.363 ± 0.167	0.312 ± 0.116	0.411 ± 0.096
10%	0.225 ± 0.147	0.303 ± 0.184	0.253 ± 0.103	0.340 ± 0.110
20%	0.192 ± 0.158	0.219 ± 0.158	0.210 ± 0.090	0.276 ± 0.105
50%	0.091 ± 0.039	0.164 ± 0.118	0.157 ± 0.056	0.274 ± 0.137
75%	0.095 ± 0.076	0.145 ± 0.115	0.136 ± 0.051	0.227 ± 0.185
100%	0.063 ± 0.031		0.108 ± 0.019	

automatic labels to suggest a more informed annotation effort that focuses on the *boundary* between the different classes.

ACKNOWLEDGEMENTS

We would like to thank Aren Jansen, Sanjeev Khudanpur and Scott Novotney for their very helpful suggestions. The first and second authors would like to acknowledge support by the National Science Foundation grant No CCF-0728931.

REFERENCES

[1] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006. [Online]. Available: http://www.kyb.tuebingen.mpg.de/ssl-book

[2] C. E. Priebe, D. J. Marchette, and D. M. Healy, "Integrated sensing and processing decision trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 699–708, 2004.

[3] A. F. Martin and M. A. Przybocki, "Nist 2003 language recognition evaluation," in *Proceedings of Eurospeech*, Geneva, Switzerland, 2003.

[4] M. A. Zissman and K. M. Berkling, "Automatic language identification," *Speech Comm.*, vol. 35, pp. 115–124, 2001.

[5] T. Kamm, *Active Learning for Acoustic Speech Recognition Modeling, Ph.D. Thesis*, Johns Hopkins Univ., 2004.

[6] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Exploring universal attribute characterization of spoken languages for spoken language recognition," in *Proceedings of Interspeech*, Brighton, UK, 2009.

[7] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*,

vol. 13, no. 2, pp. 415 –425, mar 2002.

[8] W. Hu, W. Hu, N. Xie, and S. Maybank, "Unsupervised active learning based on hierarchical graph-theoretic clustering," *IEEE Transactions on Systems, Man and Cybernetics-Part B*, vol. 39, no. 5, pp. 1147–1161, 2009.

[9] P. A. Chou, "Optimal partitioning for classification and regression trees," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 340–354, 1991.

[10] D. Karakos, S. Khudanpur, J. Eisner, and C.E.Priebe, "Iterative denoising using Jensen-Renyi divergences with an application to unsupervised document categorization," in *Proc. of ICASSP*, April 2007.

[11] D. Karakos, J. Eisner, S. Khudanpur, and C. E. Priebe, "Cross-instance tuning of unsupervised document clustering algorithms," in *Proc. 2007 Conference of the North American Chapter of the Assoc. for Computational Linguistics (NAACL-HLT 2007)*, April 2007.

[12] G. Karypis, "A software package for clustering high-dimensional data sets," University of Minnesota, Dept. of Computer Science, Technical Report 02-017, 2003.

[13] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. of the 37th Allerton Conference on Comm. and Comput.*, 1999, pp. 368–377.

[14] N. Slonim, "Iba 1.0: Matlab code for information bottleneck clustering algorithms," 2003, http://www.cs.huji.ac.il/ñoamm.

[15] D. Farris, C. White, and S. Khudanpur, "Sample selection for automatic language identification," in *ICASSP*, 2008, pp. 4225 –4228.

[16] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Exploiting context-dependency and acoustic resolution of universal speech attribute models in spoken language recognition," in *Proceedings of Interspeech*, Makuhari, Japan, 2010.