SEMI-SUPERVISED ACCENT DETECTION AND MODELING

Shilei Zhang, Yong Qin

IBM Research - China, Beijing 100193
{slzhang, qinyong}@cn.ibm.com

ABSTRACT

In this paper, we propose an iterative refinement framework for semi-supervised accent detection, where the accent labels of training corpus were generated by the user's self-judgement with poor accuracy. Firstly, we get the initial accent detection models based on cross-validation (CV) method, and then select the pure accent samples iteratively based on cost criterion derived from neighbor function, which is sensitive to the accent class purity. SVM based accent recognition approach is applied as the basic accent detection method which assumes that certain phones are realized differently across accents. Finally, we update the accent specific acoustic models via adaptation based on the detected specific accent data. The efficiency of the proposed method is demonstrated with experiments on English dictation database.

Index Terms— Accent detection, cross-validation, semi-supervised method, neighbor function

1. INTRODUCTION

One of the key challenges in practical automatic speech recognition (ASR) is to improve the recognition accuracy on accented speech. Speakers with accents differ from native speakers and from each other in many dimensions of the linguistic spectrum, including morphological, lexical, syntactical, and phonological ones. Cross-accent experiments in [1] show that accent problem is very dominant in real speech recognition application. A common problem in speech recognition of accented speech is that there is not enough training data for training an accent-specific or a speaker-specific recognizer.

Several techniques for accent/dialect detection have been successfully studied. Gaussian Mixture Models (GMMs) and Hidden Markov Model (HMM) based system have been widely applied in both practice and literature for accent identification task [2, 3, 4, 5], and discriminative training procedure can further improve the performance compared with maximum likelihood training [6, 7]. Some previous approaches used on the speaker verification task, such as Phone Recognition and Language Model (PRLM) [8, 9, 10], and Gaussian Mixture Models – Universal Background Model (GMM-UBM) with shifted delta cepstral [11] have been shown to be effective in accent/dialect recognition. In recent work [12, 13, 14, 15], researchers have shown that phone-type-based SVM kernel approach that relies on the hypothesis that certain phones are realized differently across dialects achieves state-of-the-art performance for multiple dialect and accent detection tasks. Usually, in the above work, we need collect pure accent data to train detection models, and then use them for accent identification. However, in our application, the field speakers claim their accents when applying speech application engine, which are not reliable. So we need to start with unreliable user claimed accent data, and then refine the accent detection model iteratively, finally we can re-label the accent training set and further update the accent specific acoustic models. Therefore, we call this process semi-supervised accent detection and modeling method. We propose a flexible semi-supervised framework which consists of iterative accent data purification and modeling using the cost criterion based on a neighbor function, which can effectively evaluate the accent class purity. The main benefit of this work is that we can get effective accent detection model and then we can collect large amount of accent data automatically for accent speech recognition with minimum human effort.

The rest of the paper is organized as follows: Section 2 describes the data set information. Section 3 introduces the SVM based accent detection approach. Section 4 describes the method and framework of semi-supervised accent detection. Section 5 presents the experimental results. Section 6 concludes the paper.

2. DATA SETS

Three American English accent classes (Native/Southern/Hispanic) as training data were used in the experiments, all of which were real user data collected from an English dictation system at different periods of time. The native accent data were composed of 412 speakers in 52.4 hours; the Southern accent data were composed of 3303 speakers in 504.8 hours; and the Hispanic accent data were composed of 580 speakers in 90.9 hours. The time duration of each speaker in these data sets is from 2 minutes to 100 minutes. The field accent labels come from users' own selection, which are not very accurate. Some users think they have an accent but actually they don't or their accents are very mild. So using those field data to train the accent acoustic models (AM) is not very reliable. According to our manually random checking, only about 50% of field labels match human labels. From the further analysis, we see that most of the confusions in both Southern/Hispanic data are against native.

Two sets of test data were used in the experiments. Test set 1 is used for accent detection evaluation including 3 accent subsets: Native accent set includes 87 speakers in 23.3 hours; Southern accent set includes 47 speakers in 10.2 hours; Hispanic accent set includes l7 speakers in 3.2 hours. The accents in this data set are manually labeled by human. Test set 2 with obvious Southern accent is used for accent modeling experiments, which were composed of 12 labeled southern accent speakers in 9.4 hours. The adaptation data of each speaker was 4 minutes long, recorded by the speaker in the enrollment stage reading the prompts.

3. SVM BASED ACCENT DETECTION METHOD

Support vector machines trained on Gaussian "supervectors" have been successfully used for the speaker verification and dialect recognition tasks [14, 16]. A GMM supervector (GSV) is

constructed by stacking the weighted means of the mixture model. GSV-based approaches map each speech utterance to a high-dimensional vector space. Support Vector Machines (SVMs) are generally used for classification of test vectors within this space. In this work, we apply the GSV with SVM approach to the basic accent detection problem.

3.1. Feature Extraction

Automatic time-aligned context-independent (CI) phone segmentation is generated by an English speech recognition system, where the acoustic model consists of 5k tied-states and 200k Gaussian components, trained on 2000 hours of data. The extracted features were 24-dimension vectors computed via an LDA+STC projection from 48-dimsion MFCC features (the static cepstra plus the 1st, 2nd and 3rd order derivatives). SAT training was first performed on the features, where the speaker-specific transforms were estimated via Constrained MLLR (CMLLR), and then feature-space and model-space MPE training was performed based on the SAT model. The language model used in the experiments was a general purpose trigram model.

3.2. GMM Supervector Extraction

The GMM supervector can be considered as a mapping between an utterance and a high-dimensional vector. This concept fits well with the idea of a SVM sequence kernel. The GMM-universal background model (UBM) is a single GMM model that represents the distribution of speaker independent features. This is done in order to deal with the variability that characterizes accent recognition. The GMM supervector extraction process is as follows:

a) Build a GMM-UBM for each phone type: After removing the non-speech phone, our phone inventory includes 50 CI phones; The GMM-UBM for each phone consists of 16 Gaussian components with diagonal covariance matrices, which is trained using the frames aligned to the same phone type from all training utterances.
b) Adapt the GMM-UBM and extract GMM Supervectors at the phone level:
    1. Take 2-minute speech utterances chunks labeled with accent labels (such as “native” or “southern”);
    2. MAP Adapt the means of the corresponding phone GMM-UBM on each chunk;
    3. The adapted phone GMM forms a super-vector for each chunk by stacking all the weighted Gaussian means in one super-vector. Such a supervector summarizes the acoustic-phonetic characteristics of each phone in one vector.
    4. Treat these super-vectors as SVM training features.

The resultant projection vectors naturally reflect the relationship of voice similarities among all specific accent utterances chunks, and hence are robust against interference from non-speaker factors.

3.3. Phone-type-based SVM Method

Support vector machines (SVMs) have been proved to be an effective method for pattern recognition. SVMs perform a nonlinear mapping from an input space to an SVM feature space. Linear classification techniques are then applied in this potentially high-dimensional space. The main design component in an SVM is the kernel, which is an inner product in the SVM feature space. Since inner products induce distance metrics and vice versa, the basic goal in SVM kernel design is to find an appropriate metric in the SVM feature space relevant to the classification problem. The linear SVM kernel proposed by [14] in equation (1) based on the upper bound of Kullback-Leibler (KL) divergence is used here to train the target accent detection models implemented with the LIBSVM tool [17].

$$K(S_{Ua}, S_{Ub}) = W_a^T W_b \tag{1}$$

Where $S_{Ua} = \{f_\phi'\}_{\phi\in\Phi}, S_{Ub} = \{g_\phi'\}_{\phi\in\Phi}$ are the adapted phone-GMM sets of utterance chucks U_a and U_b, Φ is phone inventory. ω_ϕ', μ_ϕ' and Σ_ϕ' are the weight, mean and diagonal covariance matrix of the adapted phone-GMMs for phone type ϕ, respectively. while ω_ϕ, μ_ϕ and Σ_ϕ are the weight, mean and diagonal covariance matrix of corresponding phone GMM-UBM, respectively. Then each utterance chuck S_{Ux} can be represented as a single supervector W_x, which is formed by stacking the mean vectors of adapted phone-GMM (after scaling by $\sqrt{\omega_\phi}\,\Sigma_\phi^{-1/2}$ and subtracting the corresponding μ_ϕ) in some fixed order. The supervector representation can be viewed as the phonetic finger-print of the speaker with accent.

For binary classifier, such as native vs. southern, the native supervector set is used to provide the positive examples, whereas the non-native set is used as background or negative samples. During test, the supervectors of the testing speech utterances chunks are used by the binary classifier to generate a classification score. On the other hand, LIBSVM implements the “one-against-one” approach for multi-class classification. If k is the number of classes, then k(k-1)/2 classifiers are constructed and each one is trained from two classes. Voting strategy is used in classification stage: the class voted by maximum number of multiple binary classifiers will be the classification decision. LIBSVM also provides classification probability estimates, which is important to the following semi-supervised accent modeling work. Another important function in LIBSVM is cross-validation, which is used in the accent model initialization in our proposed method. In k-fold cross-validation, we first divide the training set into k subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining k-1 subsets.

4. SEMI-SUPERVISED ACCENT DETECTION

4.1. Accent Characteristics and Variation

An accent is a certain form of a language spoken by a subgroup of speakers of that language which is defined by phonological features. It is intuitive that different accents have large difference in pronunciation. It is also possible that the spatial and intensity information in feature space and distribution pattern has large difference within and across different accents. In this section, we investigate how to develop a reliable cost measurement of inter-accent and intra-accent similarity based on neighbor function (NF) criterion.

4.2. Neighbor Function Criterion

The NF is a measurement of neighboring relationship [18, 19, 20]. It takes into consideration of both spatial and intensity information and their distribution pattern, which is effective to data structure with different forms.

4.2.1 Neighbor Relationship

The neighbor relationship between a couple of points (p_i, p_j) in points set $\Phi = \{p_1, p_2, \cdots p_n\}$ can be described as follows: p_i is p_j 's s^{th} neighbor, and on the other hand p_j is p_i 's t^{th} neighbor, that means there are $s-1$ neighbors of p_j are closer than p_i, and $t-1$ neighbors of p_i are closer than p_j, normally $t \neq s$. In this paper, a cost of the couple (p_i, p_j) is defined as: $a_{ij} = s + t - 2, i \neq j$.

4.2.2 Neighbor Function

In points set Φ, if points p_i and p_j are conjoint, the neighbor function c_{ij} of (p_i, p_j) is defined as $c_{ij} = a_{ij}$, it is nonnegative because $a_{ij} \geq 0$; if points p_i and p_j are not conjoint, the neighbor function $c_{ij} = 0$. Here, neighbor function also can be called as conjoint cost.

$$c_{ij} = \begin{cases} a_{ij} & if(p_i \sim p_j) \\ 0 & otherelse \end{cases} \tag{2}$$

where $(p_i \sim p_j)$ denotes that p_i and p_j are conjoint. As shown in Fig.1, class ω_1 (solid circles) are dense, class ω_2 (hollow circles) are sparse. $p_i \in \omega_2$ is more reasonable than $p_i \in \omega_1$ by human's intuition, but p_i 's nearest neighbor is p_k and $p_k \in \omega_1$. So p_i would be classified in ω_1 if judged by Euclidean distance. But if judged by neighbor function, p_k is p_i 's closest neighbor, s=1; p_i is p_k 's sixth neighbor, t=6. The neighbor function of (p_i, p_k) is $c_{ik} = 1 + 6 - 2 = 5$, similarly, the neighbor function of (p_i, p_j) is $c_{ij} = 2 + 1 - 2 = 1$, c_{ij} is smaller than c_{ik}, so $p_i \in \omega_2$.

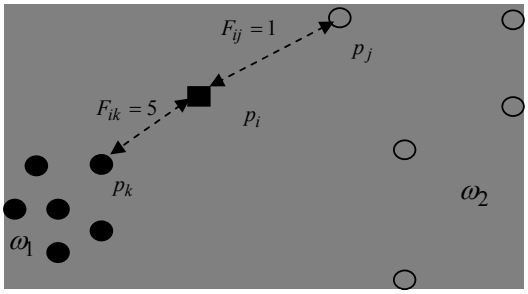

Fig. 1 Illustration of neighbor function: although p_i 's nearest neighbor is $p_k \in w_1$, $p_i \in \omega_2$ according to neighbor function

4.2.3 Cost function

When we have defined the cost of conjoint samples, we can compute the cost of intra-class and inter-class. We can define the cost of intra-class as equation (3).

$$L_{IA} = \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij} \tag{3}$$

Where N is the number of samples.

Define γ_{ij} as the minimal neighbor function value between class ω_i and class ω_j, i.e. calculate all the neighbor function values of sample pairs between class ω_i and class ω_j, and γ_{ij} is the minimal of the values. Obviously, the minimal neighbor function value between class ω_i and other $c-1$ classes can be defined as following:

$$\gamma_i = \min_{j \in \theta, j \neq i} \gamma_{ij}, \ \theta = \{1,2,\cdots,c\} \tag{4}$$

Let k be the class with minimal neighbor function value between ω_i and other $c-1$ classes. $\alpha_{i\max}$ represents the maximal conjoint cost within the class ω_i and $\alpha_{k\max}$ represent the maximal conjoint cost within the class ω_k. Then we can define the inter-class cost of ω_i in follows:

$$\beta_i = \begin{cases} -[(\gamma_i - \alpha_{i\max}) + (\gamma_i - \alpha_{k\max})] & \begin{cases} \gamma_i > \alpha_{i\max} \\ \gamma_i > \alpha_{k\max} \end{cases} \\ \gamma_i + \alpha_{i\max} & \begin{cases} \gamma_i \leq \alpha_{i\max} \\ \gamma_i > \alpha_{k\max} \end{cases} \\ \gamma_i + \alpha_{k\max} & \begin{cases} \gamma_i > \alpha_{i\max} \\ \gamma_i \leq \alpha_{k\max} \end{cases} \\ \gamma_i + \alpha_{i\max} + \alpha_{k\max} & \begin{cases} \gamma_i \leq \alpha_{i\max} \\ \gamma_i \leq \alpha_{k\max} \end{cases} \end{cases} \tag{5}$$

Then the total inter-class cost of all classes is defined by

$$L_{IR} = \sum_{i=1}^{c} \beta_i \tag{6}$$

From the above equation, we can see that if the minimal neighbor function value between ω_i and ω_k is smaller than the maximal conjoint cost within them, β_i is positive. From the discriminative perspective, we need maximize γ_i and minimize $\alpha_{i\max}$, $\alpha_{k\max}$ to make all the inter-class cost negative.

Taking both inter-class cost and intra-class cost into account, we define the cost function criterion as

$$J_{NN} = L_{IA} + L_{IR} \tag{7}$$

4.2.4 Accent Class Purity Measurement based on Cost Function

As large difference in feature space and distribution pattern within and across different accents, cost function is very fitful to measure the accent class purity in the iterative semi-supervised accent detection and classification process. We should minimize the cost function step by step, which can be treated as automatic stop criterion in the iterative process.

4.3. Semi-supervised Accent Detection

As mentioned in section 3, we extracted adapted GMM for each utterance chunk, which is treated as a sample point here. To compute the cost function, we firstly generate the neighbor function matrix of training data as follows:

a) Compute distance matrix Δ; each element : $\Delta_{ij} = \Delta(y_i, y_j)$ describe the distance between y_i and y_j. Kullback-Leibler divergence is employed as the distance measurement in the vector space of two sample points.
b) According to Δ, compute the neighbor matrix M, each element M_{ij} is neighbor relationship value of sample y_j to y_i .Generally M is non-negative matrix.
c) Generate neighbor function matrix L, where each element

is $L_{ij} = M_{ij} + M_{ji} - 2$. If points p_i and p_j are conjoint, L_{ij} is their neighbor function value. We can set the diagonal value $L_{ii} = 2N$ or larger value, where $i = 1,2,\cdots,N$.

On the other hand, we need train seed models based on selected data using 5-fold cross validation SVM accent detection system. We first divide the each accent training set into 5 subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining 4 subsets. We only use the "correct" detection data to train the initial seed accent model, where the "correct" means the selected data have the same accent label between detection output and field label (user's self claim). We select the accent data at the sample level but not the speaker level to guarantee representative samples from different speakers can be used in the model training.

The semi-supervised accent detection method is formally implemented by the following iterative procedure:

1) Accent detection and labeling of every accent training data by comparing the probability output with threshold based on the current accent detection model, where probability threshold will be increased by step 5% from 50% at each following iterative step.
2) Only select the data in training set with the same label between detection output and field label.
3) Compute the cost function value on accent class of selected data as equation (7), which will be normalized by total selected sample number.
4) Continue next step until the process can not decrease the cost function value.
5) Update the accent detection models based on selected data.

The above process can support binary and multi-class accent classifier task.

5. EXPERIMENTAL RESULTS

5.1. 3-class Accent Detection Task

In this experiment, we concentrate on 3-class accent detection task: Southern/Native/Hispanic. To avoid biasing to specific accent, the training set here includes the entire native accent training data in 52.4 hours, Southern accent data with same amount data from 308 speakers and Hispanic accent data with same amount data from 341 speakers selected randomly Firstly, we use two traditional methods to train the accent classification models directly on the above accent training sets. 1) accent classification use the likelihood scores based on the accent AMs adapted from the native AM. 2) phone type based SVM method. The testing is on *Test set 1*, where the speaker's accent classification result is decided by the voting of their samples' accent label output.

Table 1. Accent detection performance comparison of two systems

Classification Accuracy	*Native*	*Southern*	*Hispanic*
SVM based method	79.1%	60.4%	62.5%
AM Likelihood score	96.6%	21.6%	29.4%

From Table 1, we can see the accent detectors easily bias to native label, mainly because many speakers in Southern/Hispanic data set actually are native ones, even they think themselves are with some accent. Table 2 shows the classification accuracy results using Native/Southern/Hispanic 3-class classifier based on our semi-supervised accent data selection and modeling process. We can see our semi-supervised method can achieve good and unbiased accent detection results.

Table 2. Performance using semi-supervised accent detection

	Classification Accuracy
Test set 1 (Southern/Native/Hispanic)	95.3%/87.3%/85.7%
Test set 2	100.0%

5.2. Acoustic Models Adaptation with Detected Accent Data

The experiments are based on an English speech recognition system as mentioned in section 3.1. In this section, we will focus on southern accent data to perform accented AMs adaptation experiments. Many speakers in Southern accent training data set actually are native ones, so we will train the Southern/Native binary detector based on semi-supervised process using Southern/Native accent training data to further identify pure southern accent data from the corresponding accent training set. Finally, we select 49% speakers with about 200 hours data from southern accent training set, which is used to build southern accent specific adapted acoustic models. For comparison, we also build new SVM based detectors using *Test set 1* (with manually labeled accent test data) to identify the southern accent training data. Compared with these two detected accent speaker list, there are about 93% matched speaker distribution. For the above selected accented speakers, we verify the accuracy of accent detection results by manually labeling about 100 southern speakers, and we found 95% speaker have medium to high southern accents, others also have low accents.

Table 3. Performance on Southern accented AMs

Test Set 2	*Baseline (Native Model)*	*Accent Model (Adaptation set A)*	*Accent Model (Adaptation set B)*
	WER	*WER (WERR)*	*WER (WERR)*
ne	11.78%	11.02% (6.46%)	10.69% (9.27%)
4m	10.36%	10.21% (1.45%)	9.88% (4.67%)

The performance comparison results using Southern accented AMs are shown in Table 3, where non-enrollment case is unsupervised speaker adaptation, referred to as *ne* here; while in 4-minute enrollment case, we use the 4 minutes standard enrollment data to do feature space and model space speaker adaptation, referred to as *4m* here. In the accented AMs building, 2 sets of adaptation data are employed for comparison: 1) entire southern accent training set, refer to as *set A* here; 2) detected southern accent speakers by our semi-supervised method mentioned above, refer to *set B* here. We can see from the results, above relatively 9% gain with MAP based accented AMs adaptation and half of the gain survived after 4 minutes enrollment data speaker adaptation, which also indicates that the accent data is quite different from the native data, and accent issue has great impact in practical applications.

6. CONCLUSIONS

In this paper, we proposed a flexible framework to effectively select pure accent data and build the accent detection model from the unreliable accent labeling. The performance can be improved by the accented AMs based on the selected accent data in ASR experiments. For real application, a typical approach to improve the performance is to integrate an accent detector followed by a corresponding accent-specific recognizer. Our semi-supervised accent detection method is a kind of active learning algorithm, which is valuable in the area where large amount of unlabeled data is available.

7. REFERENCES

[1] C. Huang, T. Chen and E. Chang, "Accent Issue in Large Vocabulary Continuous Speech Recognition," *International Journal of Speech Technology*, 7: 141-153, 2004.

[2] C. Teixeira, I. Trancoso and A. Serralheiro, "Accent Identification," in *ICSLP*, vol.3, pp.1784-1787, Philadelphia, PA, USA, 1996.

[3] J.H.L. Hansen and L.M. Arslan, "Foreign Accent Classification Using Source Generator Based Prosodic Features," in Proc. *ICASSP*, vol.1, pp. 836-839, Detroit, Michigan, USA, 1995.

[4] P. Fung and W.K. Liu, "Fast Accent Identification and Accented Speech Recognition," in Proc. *ICASSP*, vol.1, pp. 221-224, Phoenix, Arizona, USA, 1999.

[5] T. Chen, C. Huang, E. Chang, and J. Wang, "Automatic accent identification using Gaussian mixture models," in *ASRU*, pp. 343-346, Italy, Dec. 2001.

[6] B. Burget, P. Matejka, and J. Cernock, "Discriminative training techniques for acoustic language identification," in *ICASSP*, vol. I, pp. 209-212, Toulouse, France, 2006.

[7] P. Matejka, L. Burget, P. Schwarz, and J. Cernocky, "Brno university of technology system for NIST 2005 language recognition evaluation," in *Odyssey*, San Juan, Puerto Rico, June, 2006.

[8] M.A. Zissman, T. Gleason, D. Rekart, and B. Losiewicz, "Automatic dialect identification of extemporaneous conversational, Latin American Spanish speech," in *ICASSP*, vol. 2, pp. 777 –780, Atlanta, USA, 1996.

[9] O. Koller, A. Abad, and I. Trancoso, "Exploiting variety-dependent Phones in Portuguese Variety Identification," in *Odyssey*, pp. 279-285, Brno, Czech Republic, 2010.

[10] N.F. Chen, W. Shen, and J.P. Campbell, "A linguistically-informative approach to dialect recognition using dialect-discriminating context dependent phonetic models," in *ICASSP*, pp. 5014-5017, Dallas, Texas, USA, 2010.

[11] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19 - 41, 2000.

[12] F. Biadsy, J. Hirschberg, and M. Collins, "Dialect recognition using a phone-GMM-supervector-based SVM kernel," in *Interspeech*, pp. 753-756. Makuhuari, Sep., 2010.

[13] F. Biadsy, J. Hirschberg, and D. Ellis, "Dialect and Accent Recognition using Phonetic-Segmentation Supervectors," in *Interspeech*, pp. 745-748, Florence, Italy, Aug., 2011.

[14] F. Biadsy, "Automatic Dialect and Accent Recognition and its Application to Speech Recognition," in *PhD. Thesis*, Columbia University, 2011.

[15] H. Soltau, L. Mangu and F. Biadsy, "From Modern Standard Arabic to Levantine ASR: Leveraging GALE for dialects," in *ASRU,* pp. 266-271, Hawaii, USA, Dec., 2011.

[16] W.M. Campbell, D.E. Sturim, and D.A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, pp. 308-311, 2006.

[17] C. C. Chang and C.J. Lin, "LIBSVM: a library for support vector machines," Software available at http://www.csie.ntu.edu.tw/~cjlin /libsvm.

[18] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification," Second Edition. New York: Wiley Interscience, 2000.

[19] Z. Q. Bian and X. G. Zhang, "Pattern Recognition," Beijing, Tsinghua University Press, 2000.

[20] J. Chen and J.Tian, "Reclassification of Segmentation Boundary base on Neighboring Function," *Proceedings of SPIE Symposium on Medical Imaging*, Volume 6914, pp. 69143M-1-69143M-8, USA, February, 2008.