Speed improvements in a Missing Data-based speech recogniser by Gaussian selection

Y. Wang, H. Van hamme
ESAT Department, Katholieke Universiteit Leuven, Belgium,
Email: yujun.wang@esat.kuleuven.be, hugo.vanhamme@esat.kuleuven.be

Abstract

Speech recognition performance in noisy environments such as cars is degraded due to the mismatch between the feature vector and the speech model. To improve the noise robustness, we apply Missing Data Techniques (MDT) to Hidden Markov Models (HMM) using a mixture of Gaussians for the emission densities. Traditionally, MDT uses diagonal Gaussian covariance matrices to model spectral features, but due to correlation in the feature vector, this leads to a loss of accuracy at high SNR. To overcome this problem, we previously proposed an MDT based system that takes the feature correlation into account to regain the accuracy. However, Gaussian evaluation in the acoustic model now requires solving a Non Negative Least Square problem, which increases the computational requirements by an order of magnitude. Hence several Gaussian selection paradigms are extended to be used in an MDT-based speech recognizer and subsequently compared. Hereto, a modified Symmetric Kullback-LeiblermDivergence (KLD) metric is proposed for Gaussian selection methods based on clustering. Experimental results over Dutch and Flemish databases show that on average, only about 35% of the mixtures need to be evaluated and about 60% CPU time are saved, while maintaining the accuracy of a system that evaluates all Gaussians.

Introduction

Missing Data Techniques (MDT) increase the noise robustness of a speech recogniser by reducing the mismatch between the acoustic model and the noisy features, without having to model the noise. The first application of MDT to Hidden Markov Model (HMM) was formulated in the log spectral domain (SMDT [1]). Here, speech is represented by the log-energy outputs of a filter bank and modelled by a mixture of Gaussians with diagonal covariance. However, the filter bank outputs are highly correlated and poorly modelled with a diagonal covariance Gaussians. Cepstral MDT reduces the correlations with the Discrete Cosine Transformation (DCT) on the spectral features. There are two reasons which cause a CMDT [2] recogniser to be slow. First of all, for a context dependent HMM-based speech recogniser with Gaussian mixture output probability density functions, at every time frame, significant computational load is in matching the incoming observation with each Gaussian mixture. Furthermore, for CMDT, evaluating a Gaussian gets more expensive because the Maximum Likelihood Estimation (MLE) over a mixture implies solving a Non-negative Least Square (NNLSQ) problem. The latter reason could be alleviated by introducing the PROSPECT features [3], while the former can be surmounted by using Gaussian selection. However, since the MDT modifies the acoustic model, existing Gaussian selection methods need to be revisited. Approaches such as [5] and [8] can be adapted to our present needs. Besides Gaussian selection, subspace clustering [6] was also reported as an efficient method to save both computation and model storage and is also considered in this paper.

Gaussian Selection Overview

The purpose of Gaussian selection is to remove unlikely mixtures during the decoding phase of speech recognition. In [8], the author gave an *L*-Cluster-*M*-Best scheme. Each Gaussian is assigned to one of *L* clusters. Each cluster is in turn represented by a newly created Gaussian. During decoding, if the cluster Gaussian matches the incoming observation well, its member Gaussians are further calculated. Otherwise, members of a cluster Gaussian can be coarsely evaluated by assigning the matching score of the cluster Gaussian to every member. The term "well" here is translated as "to appear in the *M*-Best list of all cluster Gaussian likelihoods". Unlike [8], in [5], the author gave a neighbourhood structure. After all Gaussians are clustered as code words or cluster Gaussians, neighbourhoods are created surrounding the cluster Gaussians. The neighbourhoods are overlapping sets, i.e. each Gaussian is assigned to one or more neighbourhoods. In the decoding phase, only the best neighbourhood is selected and its members are calculated exactly, while the others are only coarsely evaluated.

In subspace clustering [6], computational and memory savings can be achieved by dividing the whole feature space into several small streams that can each be modelled by a smaller number of Gaussians. However, in MDT, missing data needs to be imputed based on the stream's reliable data, which leads to an unallowable accuracy loss.

Missing Data Techniques with the PROSPECT Features

In SMDT [1], when the speech signal is contaminated by additive noise, a spectral mask indicates at each time frame which spectral components are labelled as missing or unreliable (dominated by noise) and which are reliable (dominated by speech). Hence, a *D*-dimensional vector of spectral observations $\mathbf{y}$ can be split into an unreliable part $\mathbf{y}_u$ and a reliable part $\mathbf{y}_r$:

$$\mathbf{y}' = [\mathbf{y}_u' \quad \mathbf{y}_r'] \tag{1}$$

Data imputation uses the reliable part as evidence to estimate $\mathbf{s}_u$, the unreliable part of the clean speech $\mathbf{s}$ within it noisy observation $\mathbf{y}_u$ using the acoustic models of the current decoder speech hypothesis. Thanks to the imputation, the acoustic model can be evaluated on complete spectral data.

Data imputation is carried out in the form of Gaussian-wise MLE, where the cost function is

$$(\mathbf{s}-\boldsymbol{\mu}_s)'\Sigma_s^{-1}(\mathbf{s}-\boldsymbol{\mu}_s) \quad (2)$$

$\boldsymbol{\mu}_s$ and Σ_s are the spectral mean and diagonal covariance of the mixture respectively. $\mathbf{s}$ contains $\mathbf{s}_r$ to be imputed in order to minimize (2) given the constraint $\mathbf{s}_u \le \mathbf{y}_u$, where $\mathbf{y}_u$ is the noisy observation .

In CMDT [2], the DCT matrix $\mathbf{C}$ is applied to the log spectral features and thereby applied to the cepstral covariance $\Sigma_\mathbf{c}$. Equation (2) is changed to

$$(\mathbf{s}-\boldsymbol{\mu}_s)'\mathbf{C}'\Sigma_c^{-1}\mathbf{C}(\mathbf{s}-\boldsymbol{\mu}_s) = (\mathbf{y}-\mathbf{x}-\boldsymbol{\mu}_s)'\mathbf{C}'\Sigma_c^{-1}\mathbf{C}(\mathbf{y}-\mathbf{x}-\boldsymbol{\mu}_s) \quad (3)$$

where $\mathbf{x}$ is the non-negative difference between $\mathbf{y}$ and $\mathbf{s}$. Minimizing (3) subject to $\mathbf{x} \ge 0$ does not have an analytic solution and requires iteration due to the non-diagonal precision matrix $\mathbf{C}'\ \Sigma_c^{-1}\mathbf{C}$.

PROSPECT features aim to reduce computation in cepstral MDT by working with a linear transforms that can be factorized with matrices of small size. Details and motivation can be found in [3]. From a statistical perspective, it implies modelling spectral correlations in the lower order cepstrum only. Let $\mathbf{C}_K$ be the K by D orthonormal DCT matrix. The transformation applied to the log-spectrum $\mathbf{s}$ is then:

$$\mathbf{p} = \begin{bmatrix} \mathbf{C}_K \\ \mathbf{P}_\perp \end{bmatrix} \mathbf{s} = \mathbf{B}\mathbf{s} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} \quad (4)$$

K cepstral components are kept in $\mathbf{c}$, while $\mathbf{d}$ is the spectral residual after removing the spectrum that is captured by $\mathbf{c}$:

$\mathbf{d}=\mathbf{s}-\mathbf{C_K}'\mathbf{c}$ hence $\mathbf{P}_\perp=\mathbf{I}-\mathbf{C}_K'\ \mathbf{C}_K$. Vector $\mathbf{p}$ of dimension $K+D$ is referred to as the PROSPECT feature vector.

The mixture-wise likelihood is formulated as:

$$\mathrm{F} = \mathrm{N}_c \mathrm{N}_d^{\ \alpha} \quad (5)$$

The cepstral part is

$$\mathrm{N_c} = \frac{1}{(2\pi)^{\frac{K}{2}} \prod_{k=1}^{K} \sigma_{ck}} \exp\left(-\frac{1}{2} \sum_{k=1}^{K} \frac{(c_k - \mu_{ck})^2}{\sigma_{ck}^{\ 2}} \right) \quad (6)$$

and the projection part is

$$\mathrm{N_d} = \frac{1}{(2\pi)^{\frac{D}{2}} \prod_{j=1}^{D} \sigma_{dj}} \exp\left(-\frac{1}{2} \sum_{j=1}^{D} \frac{(d_j - \mu_{dj})^2}{\sigma_{dj}^{\ 2}} \right) \quad (7)$$

α is the projection stream weight and is set to 0.5. σ_{ck}, μ_{ck} and c_k denote the diagonal covariance, mean and observation of k-th component of the cepstral part of the PROSPECT feature; σ_{dj}, μ_{dj} and d_j denote the j-th component of the projection part. The log likelihood to be evaluated in the PROSPECT MDT is:

$$(\mathbf{s}-\boldsymbol{\mu})'[\mathbf{C}'\boldsymbol{\Sigma}_c^{-1}\mathbf{C} + \alpha\mathbf{P}_\perp{}'\boldsymbol{\Sigma}_d^{-1}\mathbf{P}_\perp](\mathbf{s}-\boldsymbol{\mu}) \quad (8)$$

Gaussian Clustering in the PROSPECT Domain

The K-means algorithm is adopted for clustering. Gaussian selection and subspace clustering share the same clustering method. The two essentials of K-means are the distance metric and the estimation of cluster Gaussians.

Gaussian Distance Metric

The symmetric Kullback-Leibler Divergence (KLD) is used to measure the distance between two N-dimensional Gaussian mixtures with diagonal covariance matrix [7].

$$\mathrm{d(f,g)} = \mathrm{KLD(f \parallel g)} + \mathrm{KLD(g \parallel f)} = \int \mathrm{f} \log \frac{\mathrm{f}}{\mathrm{g}} dx + \int \mathrm{g} \log \frac{\mathrm{g}}{\mathrm{f}} dx$$

$$= \frac{1}{2} \sum_{i=1}^{N} \left(\frac{\sigma_{gi}^2}{\sigma_{fi}^2} + \frac{\sigma_{fi}^2}{\sigma_{gi}^2} + \frac{(\mu_{gi} - \mu_{fi})^2}{\sigma_{fi}^2} + \frac{(\mu_{gi} - \mu_{fi})^2}{\sigma_{gi}^2} \right) - N \quad (9)$$

However, some care must be taken: F formulated by equation (5) does not integrate to unity because of the stream exponent α, which compensates for unmodelled correlations in the PROSPECT features. Hence it does not fall in the concept of the KLD in information theory. Function F can be decomposed into product of a coefficient H and a strict PDF f, where

$$\mathrm{H} = (2\pi)^{\frac{(1-\alpha)D}{2}} \prod_{j=1}^{D} \frac{\sigma_{dj}^{\ 1-\alpha}}{\sqrt{\alpha}} \quad (10)$$

and

$$\mathrm{f} = \frac{1}{(2\pi)^{\frac{K+D}{2}} \prod_{k=1}^{K} \sigma_{ck} \prod_{j=1}^{D} \frac{\sigma_{dj}}{\sqrt{\alpha}}} \times \exp\left\{ -\frac{1}{2} \left(\sum_{k=1}^{K} \frac{(c_k - \mu_{ck})^2}{\sigma_{ck}} + \sum_{j=1}^{D} \frac{(d_j - \mu_{dj})^2}{\left(\sigma_{dj} / \sqrt{\alpha}\right)^2} \right) \right\} \quad (11)$$

H is a mixture-dependent coefficient but will be approximated by a constant in the sequel. When substituting the means and diagonal covariance components in the strict PDF shown in Equation (11) into Equation (9), the divergence becomes:

$$\mathrm{d}(\mathrm{f}_c, \mathrm{g}_c) + \alpha\, \mathrm{d}(\mathrm{f}_d, \mathrm{g}_d) + A - (1-\alpha)D \quad (12)$$

(which is symmetrical and strictly positive for $\mathrm{f} \neq \mathrm{g}$) where

$$\mathrm{d}(\mathrm{f}_c, \mathrm{g}_c) = \frac{1}{2} \sum_{k=1}^{K} \left(\frac{\sigma_{gck}^2}{\sigma_{fck}^2} + \frac{\sigma_{fck}^2}{\sigma_{gck}^2} + \frac{(\mu_{gck} - \mu_{fck})^2}{\sigma_{fck}^2} + \frac{(\mu_{gck} - \mu_{fck})^2}{\sigma_{gck}^2} \right) - K \quad (13)$$

$$\mathrm{d}(\mathrm{f}_\mathrm{d}, \mathrm{g}_\mathrm{d}) = \frac{1}{2} \sum_{j=1}^{D} \left(\frac{\sigma_{gdj}^2}{\sigma_{fdj}^2} + \frac{\sigma_{fdj}^2}{\sigma_{gdj}^2} + \frac{(\mu_{gdj} - \mu_{fdj})^2}{\sigma_{fdj}^2} + \frac{(\mu_{gdj} - \mu_{fdj})^2}{\sigma_{gdj}^2} \right) - D \quad (14)$$

$$A = \frac{1}{2} \sum_{j=1}^{D} \left(\frac{(1-\alpha)\sigma_{gdj}^2}{\sigma_{fdj}^2} + \frac{(1-\alpha)\sigma_{fdj}^2}{\sigma_{gdj}^2} \right) \quad (15)$$

μ_{fck} and σ_{fck} are the k-th component of the mean and diagonal covariance of the cepstral part of f; μ_{gck} and σ_{gck} are the counterparts of g. μ_{fdj} and σ_{fdj} are j-th component of the mean and diagonal covariance of the projection part of f; μ_{gdj} and σ_{gdj} are the counterparts of g. We have observed that omitting A from equation (12) leads to a better balancing of cluster sizes and a better computation/accuracy trade off. When computing the distance between a Gaussian (smaller variance) and a cluster candidate (larger variance) we

observe from (15) that due to A, a cluster with a large variance in the projection part may be disfavoured. Hence the metric becomes

$$\mathrm{d}(\mathrm{f}_c, \mathrm{g}_c) + \alpha\, \mathrm{d}(\mathrm{f}_d, \mathrm{g}_d) \tag{16}$$

With the work of [4], both the output PDF and symmetric KLD can be written in forms of static and dynamic streams. Each stream can be divided into a cepstral and a projection part. Consequently, there are six streams with stream weights α_i (0.5 or 1). The expression for the distance metric combining information from S different streams with diagonal covariance and with different stream weights is

$$\sum_{i=1}^{S} \alpha_i\, \mathrm{d}(\mathrm{f}_i, \mathrm{g}_i) \tag{17}$$

Parameter Estimation of Cluster Gaussians

Like in [7], the cluster centre is chosen by unweighted matching of first and second order moments with the clustered Gaussians. Hence, its mean and diagonal covariance are given by:

$$\overline{\mu_i} = \frac{1}{W}\sum_{k=1}^{W} \mu_{ki} \tag{18}$$

$$\overline{\sigma_i}^2 = \frac{1}{W}\sum_{k=1}^{W} (\sigma_{ki}{}^2 + \mu_{ki}{}^2) - \overline{\mu_i}^2 \tag{19}$$

where W is the number of Gaussians belonging to the specific cluster. μ_{ki} and σ_{ki} are the i-th component of mean and diagonal covariance of k-th member Gaussian. Finally, the PROSPECT means in (13) and (14) are transformed to the spectral domain by multiplying it with the pseudo-inverse of **B** (as defined in Equation (4)) such that cluster Gaussians can be evaluated on spectral feature vector using the cost function (8).

Experiments

Both the L-Cluster-M-Best and the neighbourhood methods of Gaussian selection for PROSPECT MDT were implemented and tested in our experiments, as well as the subspace clustering. The experiments are carried out on AURORA-4 [11] in-car database which is a large vocabulary continuous speech recognition task. Since the noise in AURORA-4 is artificially added, we also evaluate on the SpeechDat [12] in-car Flemish database, to which we added SNR classification.

Experiments on AURORA-4

AURORA-4 is a 5k-word dictation task. The PROSPECT acoustic model set is trained using single pass retraining from a 21037 tied mixture model set, where K=4 and D=22 using the clean training set. A VQ mask [10] is computed from the noisy signal. Both the L-Cluster-M-Best and the neighbourhood Gaussian selection are evaluated. The percentage of Gaussians calculated of the former method is controlled by the M to L ratio, while that of the latter is controlled by the average neighbourhood size to mixture number ratio which is in turn controlled by a threshold Θ. A mixture i will be regarded as a member of the neighbourhood of cluster Gaussian j if:

$$\mathrm{Divergence}(mixture_i, cluster_j) < \Theta.\bar{E}_j$$

where divergence is defined by (16) and $\bar{E}_j$ is the average quantization error of cluster j. Θ is initiated with 1 and is increased to fulfill the predefined average neighborhood size. It is observed in AURORA-4 that the performance does not get much better as the number of clusters L is increased. We found that L = 110 is a reasonable value. In our experiments, replacing the score of the unselected member Gaussians with a very small value gives better results than assigning their scores with those of their cluster Gaussians in both L-Cluster-M-Best and neighborhood implementation as [8] did.

In Figure 1, the percentages of Gaussians to be calculated give an indication of the computational efficiency of decoding. For L-Cluster-M-Best, it is calculated by:

$$100(T.L + \sum_{t=1}^{T}\sum_{k=1}^{M} w_k)/T/G \tag{20}$$

where T is the number of frames, w_k is the size of cluster k and G is the total number of Gaussians. Curve (a) shows the tradeoff between Word Error Rate (WER) and the percentage of Gaussians calculated of 110-Cluster-M-Best Gaussian selection method, where M=(11, 22, … ,99, 110) yielding the ten data points along the curve. Curve (b) shows that of the neighborhood method. The average ratio of neighborhood size to the number of mixtures equals 10%, 20%… 90%. The L-Cluster-M-Best method selects M clusters around the observation, while the neighbourhood method selects Gaussians in a wide area, but not centred around the observation. Therefore, the former is more effective method, as is observed from our experiments. Curves (a) and (b) reach the same point when 100% of the Gaussians are calculated. It also shows that this point is above most points along curve (a), a phenomenon also observed in [6]. Curve (c) is the performance of subspace clustering with 4096, 6144 and 8192 clusters in each of the static and dynamic streams.

In order to achieve an efficient computation without losing much accuracy (or even gaining accuracy), we choose L-Cluster-M-Best in further experiments on MIDAS Flemish in-car database as the Gaussian selection method.

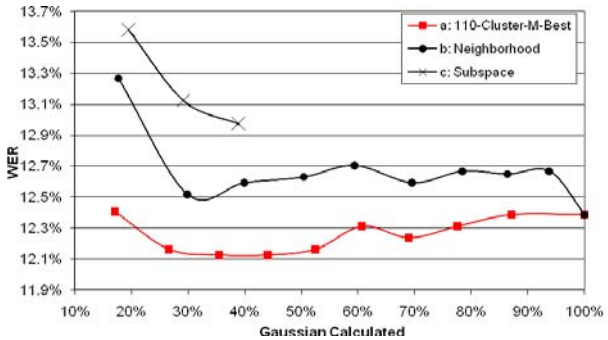

Figure 1: WER with percentage of Gaussian calculated of L-Cluster-M-Best, neighbourhood Gaussian selection and subspace clustering in AURORA-4 experiments.

Experiments on the SpeechDat Car Flemish Database

The car data in SpeechDat Flemish database [12] includes utterances recorded in different driving conditions. It consists of four channels from close, medium and far field microphones. The Flemish PROSEPECT acoutic model set is again trained using single pass retaining from triphone HMM with 28917 tied Gaussians. . The training data are

taken from the read speech component of the Flemish CGN database [ref]. The *L*-Cluster-*M*-Best Gaussian selection is tested, where *L*=170, *M*=34. Figure 2 shows the accuracy results on the recognition task with 637 active words or commands. The small gain in accuracy due to Gaussian selection is also observed here.

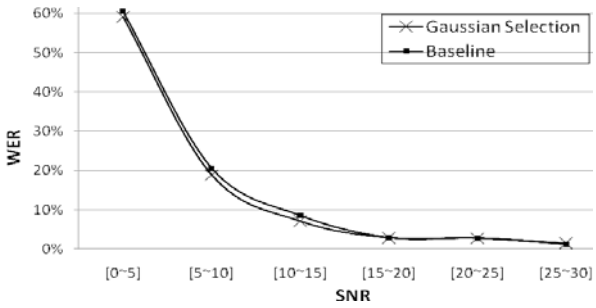

Figure 2: WER per SNR of 170-Cluster-34-Best Gaussian selection and the baseline recogniser without Gaussian removal on isolated words grammar of MIDAS Flemish in-car data

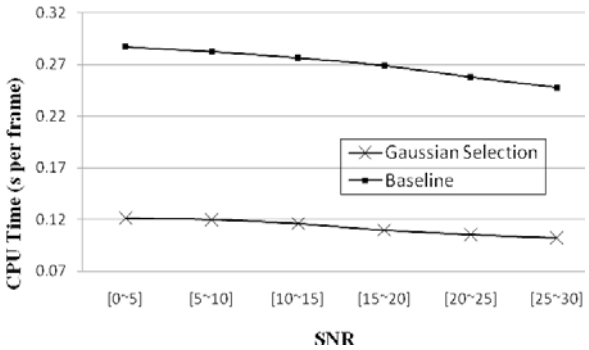

Figure 3: CPU time per frame of 170-Cluster-34-Best Gaussian selection and the baseline recogniser without Gaussian removal on isolated words grammar of SpeechDat Flemish in-car data.

Figure 3 compares the CPU time per frame between 170-Cluster-34-Best Gaussian selection and the baseline recogniser without removing Gaussians from evaluation on the isolated words task. Gaussian selection saves 60% CPU time. Both the CPU time of the baseline system and Gaussian selection are increased as the SNR is decreased, which is attributed to the increase in number of unreliable time-frequency cells, making the NNLSQ more expensive to solve. Furthermore, as more time-frequency cells are unreliable and the noisy observations are constraining the NNLSQ problem less ($\mathbf{s} \leq \mathbf{y}$) at lower SNR, the larger clusters tend to end up more in the *M*-best list, resulting in more Gaussians to be evaluated.

Conclusions

We have implemented the *L*-Cluster-*M*-Best and neighbor-hood Gaussian selection methods to speed up the PROSPECT MDT recogniser by excluding a large fraction of mixtures from evaluation. The clustering is performed in the PROSPECT domain and the covariance matrices of both cluster Gaussians and member Gaussians are assumed to be diagonal. The weighted KLD is shown as a valid distance metric in K-means clustering. As the number of cluster gets larger, the experiments do not show much better results. With the same computational load, the *L*-Cluster-*M*-Best method performs better than the neighborhood method. Subspace clustering cannot give results as good as Gaussian selection. The Gaussian selection is worthwhile because it brings great efficiency with tiny performance degradation as a trade off.

Acknowledgements

This research is financed by the MIDAS project of the Nederlandse Taalunie under the STEVIN programme.

References

[1] Cooke, M., Green, P., Josifovski, L. Vizinho, A., "Robust Automatic Speech Recognition with Missing and Unreliable Acoustic Data", Speech Communication, volume 34, no. 3, 267–285, 2001.

[2] Van hamme, H., "Robust Speech Recognition Using Missing Feature Theory in the Cepstral or LDA Domain", Proc. Eurospeech, 3089-3092, 2003.

[3] Van hamme, H., "PROSPECT Features and their Application to Missing Data Techniques for Robust Speech Recognition", Proc. ICSLP, volume I, 101-104, 2004.

[4] Van hamme, H., "Handling Time-Derivative Features in a Missing Data Framework for Robust Automatic Speech Recognition", Proc. ICASSP, 293-296, 2006.

[5] Bocchieri, E., "Vector quantization for efficient computation of continuous density likelihoods", Proc. ICASSP, Volume 2, 692–695, 1993.

[6] Bocchieri, E., Mak, B.K.-W., "Subspace Distribution Clustering Hidden Markov Model", IEEE Trans. Speech and Audio Proc., 9(3):264-275, 2001.

[7] Myrvoll, T.A., Soong, F.K., "Optimal Clustering of Multivariate Normal Distributions Using Divergence and Its Application to HMM Adaptation", Proc. ICASSP, I-552- I-555, 2003.

[8] Watanabe, T., Shinoda, K., Takagi, K., Iso, K.-I., "High Speed Speech Recognition Using Tree-Structured Probability Density Function", Proc. ICASSP, Volume I, 556 – 559, 1995.

[9] Shinoda, K., Lee, C.-H., "A structural Bayes approach to speaker adaptation", IEEE Trans. Speech and Audio Proc., 9(3):276-287, 2000.

[10] Van Segbroeck, M., Van hamme, H., "Vector-Quantization Based Mask Estimation For Missing Data Automatic Speech Recognition", Proc. Interspeech, 910-913, 2007.

[11] Parihar, N., Picone, J., "An Analysis of the Aurora Large Vocabulary Evaluation," Proc. Eurospeech, 337-340, 2003.

[12] Moreno, A., Lindberg, B., Draxler, C., Richard, G., Choukri, K., Allen, J., Euler. S., "Speechdat-car: A large speech database for automotive environments". LREC, 2000.