IDIAP RESEARCH REPORT

BASELINE SYSTEM FOR AUTOMATIC SPEECH RECOGNITION WITH FRENCH GLOBALPHONE DATABASE

Sandrine Revaz Milos Cernak

Idiap-RR-26-2012

AUGUST 2012

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Baseline System for Automatic Speech Recognition with French GlobalPhone Database

Revaz Sandrine and Milos Cernak (*supervisor)*

August 10, 2012

Abstract

This report presents one month trainee work on development of French Automatic Speech Recognition (**ASR**) system using a french part of multilingual database GlobalPhone_FR. The purpose of this report is to explain and give results of the training and testing of the ASR with this specific database. Two different methods are presented, the Hidden Markov Model (HMM) with MFCC/PLP features and tandem features from Multilayer Perceptron (MLP) phone posteriors. The report presents data preparation for GlobalPhone_FR ASR training, and compares the two different approaches. Word recognition accuracy achieved with MFCC features is 71.46% and the tandem features with 3-layer MLP improved the accuracy to 72.15%. We interpret this result as a baseline for the GlobalPhone_FR database.

1 Database

1.1 GlobalPhone

GlobalPhone is a multilingual text and speech database developed by XLingual in collaboration with the Karlsruhe Institute of Technology (Schultz (2002)). The goal of the GlobalPhone database collection was to provide read speech database suitable for different kinds of research in the areas of (1) multilingual speech recognition, (2) rapid deployment of speech processing systems to new or under-resourced languages, (3) language and speaker identification tasks, (4) multilingual speech synthesis or voice conversion, as well as (5) monolingual recognition in a large variety of languages. We are interested in the first one for Automatic Speech Recognition (ASR). The entire GlobalPhone corpus enables the acquisition of acoustic-phonetic knowledge of the most widespread languages of the world, 19 languages to be exact. In this report, we are interested in the French database, GlobalPhone_FR, which contains exactly 100 adult speakers and around 10′503 sentences. The speech data is available in PCM waveform files, 16bit resolution, 16kHz sampling rate, mono quality. The read texts were selected from national newspapers available via Internet to provide a large vocabulary (up to 65,000 words); the main topics are politics and economics which can restrict the vocabulary. For French database, it's the newspaper Le Monde which was selected. The transcriptions are internally validated and supplemented by special markers for spontaneous effects like stuttering, false starts, and non-verbal effects like laughing and hesitations. Speaker information such as age, gender, occupation, etc. as well as information about the recording setup complements the database.

GlobalPhone_FR: The GlobalPhone_FR's folder contains two important folders: /audio and /dbase. In the first one, there are all the audio files, '.wav' files, which belong to /adc's folder.
In the second, two folders are useful: /adc and /trl, and one is interesting for informations about speakers: /spk.

/adc contains the audio of one spoken turn TID of speaker SID. The audio format is PCM 16bit 16kHz byte-order low-high lossless compressed with the program "shorten" written by Tony Robinson[1]. Namimg format of a file is as follows:

/adc/SID/LIDSID_TID.adc.shn

where:

[1] http://www.softsound.com/Shorten.html

/adc/ = directory name
SID = speaker ID (e.g. ”001”)
LID = language ID (one of ”AR, CH, CR, .., VN”)
TID = turn ID (starting at ”1”)
adc.shn = compressed audio (shorten by T. Robinson)

In **/trl**, the transcription files, /trl/LIDSID.trl, contain the spoken utterances transcribed in language specific encoding (ISO8859-1 for French). Both, the speakers ID (SID) and the turnID (TID) are reported in the transcription file as comment lines started by ;. The file contains one turn per line preceded by the TID.
For example:

;SprecherID FR100
; 1:
A cette occasion on remettra aux étudiants une brochure qui regroupe les conseils personnalisés des enseignants qui assureront les cours et une première bibliographie
; 2:
Ainsi selon le point du réseau ou le virus a été découvert le type de réseau et le type de virus on décidera ou non de déconnecter physiquement les pc du réseau
...

The following table (1) describes the speakers characteristics such as the gender and age distribution for French database. The first table shows gender, age category, and smoking (y=smoker, n=nonsmoker) as well has health status (y=feels healthy, n=feels sick or has allergies). The category x indicates that information is not available.

LID	Spk	Gender			Age Category						Smoking			Healthy		
		F	M	x	<19	$20-29$	$30-39$	$40-49$	$\geqslant 50$	x	y	n	x	y	n	x
FR	100	51	49	0	3	52	16	13	14	2	0	0	100	0	0	100

Table 1: Speakers characteristics

1.2 Database Preparation

Before starting with the acoustic modeling, some database preparations are needed.

Lists: The list of speakers must be split up into 3 files: training list, development list and test list. In the database of GlobalPhone_FR, we have 100 different speakers so we decided to use the first 80 speakers for training, the next 10 for testing and the last 11 for developing.

Transcription: First, all the transcription were converted from ISO8859-1 to UTF-8 coding, and tokenized and normalized using in-house scripts. We convert all the '.trl' files supplied with GlobalPhone_FR into HTK's master label files ('.mlf') in the working directory, i.e. firstly we have for example:

;SprecherID FR001
; 1:
A ces trois sortes de jours sont associés deux prix par jour en fonction de la consommation en heures pleines ou creuses
; 2:
Ainsi les quartiers se débarrassent de ces populations sinistres et de ces bouges où la police ne met le pied que quand la justice l' ordonne
...

and the effect should be to convert the prompt utterances exampled above into the following form:

#!MLF!#
"/FR*001_1.*adc.lab"*
à
ces
trois
sortes
de
jours
sont
associés
deux
...

So, as can be seen, the prompt labels need to be converted into path names, each word should be written on a single line and each utterance should be terminated by a single period on its own. The first line of the file just identifies the file as a Master Label File (MLF).

Dictionaries: The first step in building a dictionary is to create a sorted list of the required words. Each word must be associated with its own phonetic, for this step, we are using the French dictionary, BDLEX[2]. The phonemes in the dictionaries are represented using the Speech Assessment Methods Phonetic Alphabet (SAMPA). SAMPA is based on the International Phonetic Alphabet (IPA), but features only ASCII characters[3]. BDLEX consists of a lexical database developed at Institut de Recherche en Informatique (IRIT) in Toulouse. The data cover lexical, phonological, and morphological information. It may miss some word in the BDLEX, so a list of unseen words is created. We used Phonetisaurus [4], a grapheme-to-phoneme tool that uses existing dictionaries to derive a finite state transducer based mapping of sequences of letters (graphemes) to their acoustic representation (phonemes). The transducer was then applied to the list of unseen words. In that way we completed the training dictionaries.

Two dictionaries are required for training: flat (basic) and main (alternative entries with sil and sp final phones) for alignment. The two files begin with this two lines:

</s>	[]	sil
</s>	[]	sil

which represent the three columns of the file. The first one is the word, the second in bracets is the written word if the first one is recognized and the last column is the phonetic transcription. The flat dictionary contains each word with this tree columns and the main dictionary is the same but each line is written twice, one ended by 'sil' and the other by 'sp':

</s>	[]	sil
</s>	[]	sil
-t-il	[-t-il]	t i l sil
-t-il	[-t-il]	t i l sp
a	[a]	a sil
a	[a]	a sp
abaissement	[abaissement]	a b E s m a^ sil
abaissement	[abaissement]	a b E s m a^ sp
...		

[2] http://catalog.elra.info/product_info.php?products_id=33
[3] http://www.phon.ucl.ac.uk/home/sampa/index.html
[4] http://code.google.com/p/phonetisaurus/

2 Hidden Markov Models (HMMs) with mel-frequency cepstral coefficients

2.1 Introduction

A hidden Markov model (**HMM**) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. An HMM can be considered as the simplest dynamic Bayesian network. HMMs can be trained automatically and are simple and computationally feasible to use.

A HMM consists of a number of states. Each state j has an associated observation probability distribution $b_j(o_t)$ which determines the probability of generating observation o_t at time t and each pair of states i and j has an associated transition probability a_{ij}. The Figure 1 shows a simple left-right HMM with five states in total. Three middle of these are emitting states and have output probability distributions associated with them.

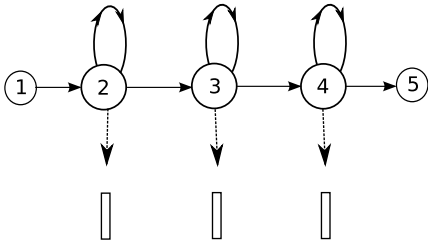


Figure 1: A simple left-right HMM with five states.

The Fig. 2 depicts a scheme of the training and testing process.

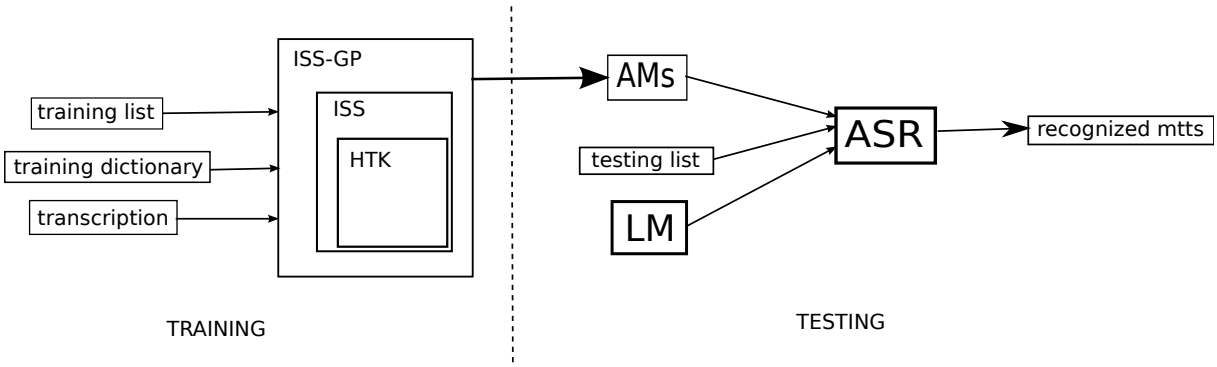


Figure 2: ASR training and testing schema. ISS stands for *Idiap Speech Script*, AMs for *Acoustic Models*, and LM for *Language Model*.

The first step (training list, training dictionary, transcription) is described in the Sec. 1.2 'Database Preparation'. Then, the acquisition of the AMs is explained in the training part. The third step is addressed in the test part. Finally the results are described at the end of this chapter.

2.2 Training Part

Training of AMs was performed using the HTS tools[5], wrapped into Idiap Speech Scripts.

Features Extraction First of all, we need to extract features. For this we used both the Perceptual Linear Prediction (**PLP**) and Mel-Frequency Cepstral Coefficients (MFCC) methods. Both parametrisations are based on the short-term spectrum of speech and use mel-frequency filterbanks. The feature extractor outputs a sequence of M-dimensional vectors (with M being a small integer, such as 10), outputting one of these every 10 milliseconds (known as frame shift parameter). The vectors consist of cepstral coefficients, which are obtained by taking a Fourier transform of a short time window of speech and

[5] http://hts.sp.nitech.ac.jp/

decorrelating the spectrum using a cosine transform, then taking the first (most significant) coefficients. As we parametrized static parameter vector with $M = 12$ plus energy coeffienct, finally we obtain 39 features; delta (+13) and acceleration (+13) coefficients. Ceptral mean normalization has been also applied.

Training To start, the training program aligns the words and its pronouciations; it's mean that a matching is create between the word and its phonemes. Then, an estimation of each phoneme (monophone) is given by a single Gaussian. After, a list of triphones is created and an reestimation is done, having still single Gaussian distributions. The HMM is trained to have in each state a statistical distribution that is a mixture of 16 Gaussians, which will give a likelihood for each observed vector. Finally to tie rare states, we applied a Minimum Decription Length (MDL) decision trees (Shinoda and Watanabe (1997)). The set of Gaussian's means and variances in each HMM state forms the AMs.

2.3 Testing

So, we have the AMs given by the training part, the test list given by the database preparation, and there is only the LM needed to run speech recognition. The LM is the set of the likelihoods of the appearance of each word (and word's sequence) in the text database. Now, using this formula:

$$\underset{i}{\operatorname{argmax}}\{P(w_i|\vec{o})\} = \frac{P(w_i)P(\vec{o}|w_i)}{P(\vec{o})}$$

where:

w_i is the i^{th} vocabulary word,
$\vec{o}$ is a sequence of speech vectors,
$P(w_i|\vec{o})$ is the probabilities to obtain w_i with $\vec{o}$ given; this is what we search
$P(w_i)$ is the probabilities given by LM,
$P(\vec{o}|w_i)$ is the probabilities given by AMs,
$P(\vec{o})$ is negligible because of the maximization.

we can test the ASR. The 3-gram LM with Kneser-Ney discounting was trained from all the text transcription belonging to the training part of AMs of the GlobalPhone_FR database using the SRILM language modelling toolkit (Stolcke (2002)).

2.4 Results with PLP features

First, we test the development list for adjusting the two parameters: LM_SCALE (Table 2) and WORD_PENALTY (Table 3).

LM_SCALE is a parameter which permit to compare the probabilites from LM and that of AMs. WORD_PENALTY permits to tune a numner of inserted words in the recognition result.

Development List - PLP				
WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
0	10	0.53	76.39	71.90
0	14	1.06	76.6	73.54
0	**16**	0.88	76.39	**73.68**
0	18	0.88	76.11	73.65
0	17	0.97	76.26	73.67
0	15	0.88	76.53	73.66

Table 2: Difference of results on development set depending on LM_SCALE

We see in the tables above that the best result is for LM_SCALE=16 and WORD_PENALTY=-7. This values and the test list give (Table 4):

Development List - PLP				
WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
-10	16	0.88		73.74
-5	16	0.97	76.08	73.78
-7	16	0.97	75.97	**73.82**

Table 3: Difference of results on development set depending on WORD_PENALTY

Test List - PLP				
WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
-7	16	0.57	74.36	**71.82**

Table 4: Results of PLP features with the test list.

2.5 Experiments

First experiment: variation of the 'TIE_FORCE_NSTATES' which determines the number of tied states (Table 5). The above results are based on TIE_FORCE_NSTATES=3000.

Here WORD_PENALTY=−7 and LM_SCALE=16.

Test List - PLP			
TIE_FORCE_NSTATES	SENT % Correct	Word % Correct	Accuracy
none	0.38	74.49	71.68
1000	0.0	66.23	63.30
2000	0.0	67.40	64.13
3000	0.57	74.36	71.82

Table 5: Difference of results depending on TIE_FORCE_NSTATES

As we can see, the result is better if there is no TIE_FORCE_NSTATES or if TIE_FORCE_NSTATES=3000. The models are more compact and robust with state tying.

Second experiment: variation of the a number of alternative word pronunciations n in the training dictionary (Table 6). The above results are based on n=1.

Here there is no TIE_FORCE_NSTATES, WORD_PENALTY=−7 and LM_SCALE=16.

The variation of n does not improve the result. The reason is that the alrernative pronunciations generated by Phonetisaurus is data-driven that probably does not correlate with confusions percieved by humans. We still try to tuned WORD_PENALTY and LM_SCALE with the condition n=6 and the development list but it did not give better results (Table 7). Here there is no TIE_FORCE_NSTATES.

Third experiment: We extract the features with Mel-frequency cepstral coefficients (**MFCCs**) method (Table 8).

Finaly the best result for the test list is given by PLP under this conditions (Table 9).

Test List - PLP			
n	SENT % Correct	Word % Correct	Accuracy
1	0.38	74.49	71.68
2	0.19	68.36	64.79
6	0.0	62.32	57.94

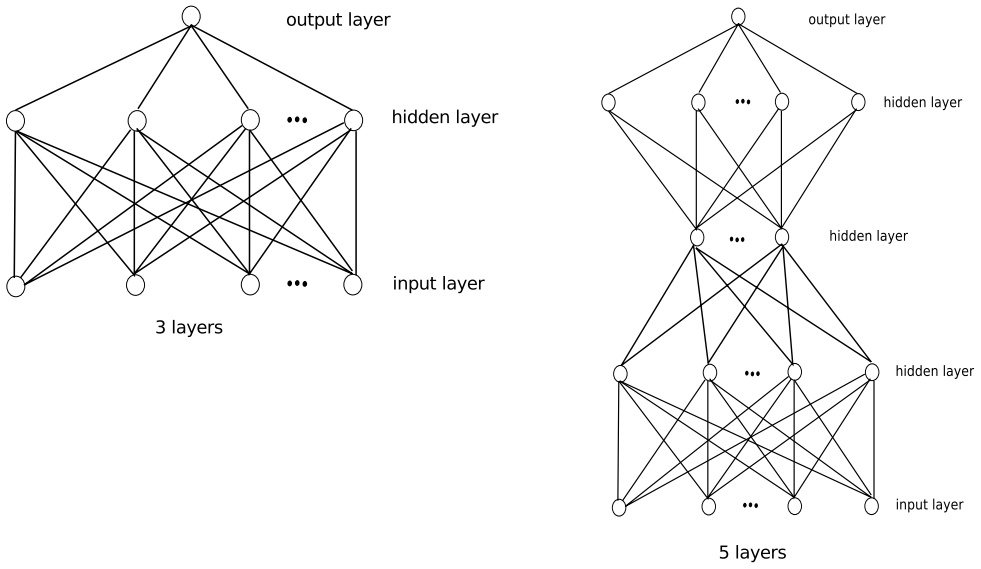
Table 6: Difference of results depending on n - the number of alternative word pronunciations.

Development List - PLP				
WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
0	18	0.0	65.57	61.08
0	15	0.0	65.01	59.81
0	20	0.0	65.60	61.47
-10	20	0.0	64.33	61.38
-5	20	0.0	64.92	61.42

Table 7: Tuned of WORD_PENALTY and LM_SCALE depending on n=6.

3 HMMs with tandem features from Multilayer Perceptron (MLP) phone posteriors

3.1 Introduction



A multilayer perceptron (**MLP**) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate output. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one (Figure above). Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training the network:

THE BACK PROPAGATION ALGORITHM (**BPA**):
Step 1: Initialization: Set $t = 0$ and choose initial weight matrices W for each layer.
Lets denote $w_{ij}^{k}(t)$ as the weighting coefficients connecting i^{th} input node in layer $k-1$ and j^{th} output node in layer k at time t.
Step 2: Forward Propagation: Compute the values in each node from input layer to output layer in a

Test List - MFCCs						
n	TIE_FORCE_NSTATES	WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
1	none	-7	16	0.48	74.28	**71.48**
1	3000	-7	16	0.48	73.97	**71.46**

Table 8: **Result with MFCC features.**

Test List - PLP						
n	TIE_FORCE_NSTATES	WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
1	3000	-7	16	0.57	74.36	**71.82**

Table 9: **Best Result achived with PLP features.**

propagating fashion, for $k = 1$ to K

$$v_j^k = sigmoid(w_{0j}(t) + \sum_{i=1}^{N} w_{ij}^k(t) v_i^{k-1}) \quad \forall j$$

where $sigmoid(x) = \frac{1}{1+e^{-x}}$ and v_j^k is denotes as the j^{th} node in the k^{th} layer.
Step 3: Back Propagation: Update the weights matrix for each layer from output layer to input layer according to:

$$\overline{w}_{ij}^k(t-1) = w_{ij}^k(t) - \alpha \frac{\partial E}{\partial w_{ij}^k(t)}$$

where $E = \sum_{i=1}^{s} ||y_i - o_i||^2$ and $(y_1, y_2, \ldots, y_s)$ is the computed output vector in Step 2. α is referred to as the learning rate and has to be small enough to guarantee convergence. One popular choice is $\frac{1}{(t+1)}$.
Step 4: Iteration: Let $t = t + 1$ Repeat Steps 2 and 3 until some convergence condition is met.

3.2 MLP-Training Part

The same training that in the chapter before is done except the extraction was done by PLP/MFCC features. But before, the new features was created by the BPA describes above. We can resume the contents of the BPA in this way:

- Presentation of a pattern to the network drive.
- Comparing the output with the output of the targeted network.
- Calculating the error at the output of each of the neurons of the network.
- Computing, for each of the neurons, the output value that would have been correct.
- Definition of the increase or decrease necessary to obtain this value (local error).
- Adjusting the weight of each connection to the local error is the lowest.
- Assigning blame to all previous neurons.
- Repeat from step 4 on previous neurons using blame as an error.

The input is 9 frames (1 central frame and its context: 4 before and 4 after) with is 39 features (13 MFCC coefficient + 13 delta + 13 acceleration) so 351 inputs.

So, for 3 layers, the first (input) contains 351 neurons and the output (the third) is a vector of 38 improved features (we had 38 phonemes in the system). The second is a hidden layer.

For 5 layers, the first (input) contains also 351 neurons. We add the constraint that the third one have only 50 neurons, resulting in a bottleneck arquitecture. The second and the forth are hidden layers and the last one (output) is a vector of 38 improved features.

3.3 MLP-Testing Part

The only difference there is with the previous chapter is a new feature set.

3.4 Results

The result for a **3-layers** perceptron is (Table 10):

Test List - MFCC						
n	TIE_FORCE_NSTATES	WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
1	3000	-7	16	0.57	75.30	**72.15**

Table 10: **Result of MLP with 3 layers.**

We test the development list for adjusting the two parameters: LM_SCALE and WORD_PENALTY (Table 11).

Dvl List - MFCC						
n	TIE_FORCE_NSTATES	WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
1	3000	0	10	0.97	76.32	70.98
1	3000	0	15	1.33	77.02	73.51
1	3000	0	17	1.24	76.80	73.72
1	3000	0	19	1.15	76.58	73.80
1	3000	0	20	1.15	76.49	73.82
1	3000	**-5**	**20**	1.15	76.28	**73.94**
1	3000	-7	20	1.15	76.19	73.90
1	3000	-3	20	1.15	76.34	73.87

Table 11: **Tuning of WORD_PENALTY and LM_SCALE with the dvl list using MLP with 3 layers.**

We see in the Tab. 11) that the best result is for LM_SCALE=20 and WORD_PENALTY=-5. This values and the test list give (Table 12).

Test List - MFCC						
n	TIE_FORCE_NSTATES	WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
1	3000	-5	20	0.67	74.97	**72.17**

Table 12: Result with the test list for 3-MLP

4 Conclusions

In conclustion, we use HMM for training the model with PLP features. Then, we test some variation of the number of tied states (TIE_FORCE_NSTATES) and the number of different prunonciation (up to 6 alternatives). We didn't obtained significant difference of the results. After, we also train the model with MFCCs features which didn't improved the results. Finally, we use the MLP method with 3 layers. With this method, we can see a small improvement but not significant (Table 13).

We began with MLP method with 5 layers that promises better performance, but time does not allow us to complete the training.

Test List - MFCCs - HMM						
n	TIE_FORCE_NSTATES	WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
1	3000	-7	16	0.48	73.97	**71.46**
Test List - MFCCs - MLP with 3 layers						
n	TIE_FORCE_NSTATES	WORD_PENALTY	LM_SCALE	SENT % Correct	Word % Correct	Accuracy
1	3000	-7	16	0.57	75.30	**72.15**

Table 13: **Comparison between HMM and MLP with 3 layers**

5 Acknowledges

I would like to thank Idiap to permit me to discover mathematics used in the field of artificial inteligence, it was really interesting. I learned a lot about informatic, preparation data, different kind of training and testing in speech recognition.

I want to thank Alexandre Nanchen for his time and his help with my understanding of mathematical concepts.

And finaly thank you for hosting the group of developers.

References

Tanja Schultz. Globalphone: A multilingual speech and text database developed at karlsruhe university. In *Proceedings of the ICSLP*, pages 345–348, 2002.

Koichi Shinoda and Takao Watanabe. Acoustic modeling based on the MDL principle for speech recognition. In George Kokkinakis, Nikos Fakotakis, and Evangelos Dermatas, editors, *EUROSPEECH*. ISCA, 1997.

Andreas Stolcke. Srilm-an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286, November 2002.