We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Practical Issues of Building Robust HMM Models Using HTK and SPHINX Systems

Juraj Kacur and Gregor Rozinaj
Slovak University of Technology, Faculty of Electrical Engineering
and Information Technology, Bratislava
Slovakia

1. Introduction

For a couple of decades there has been a great effort spent to build and employ ASR systems in areas like information retrieval systems, dialog systems, etc., but only as the technology has evolved further other applications like dictation systems or even automatic transcription of natural speech (Nouza et al., 2005) are emerging. These advanced systems should be capable to operate on a real time base, must be speaker independent, reaching high accuracy and support dictionaries containing several hundreds of thousands of words.

These strict requirements can be currently met by HMM models of tied context dependent (CD) phonemes with multiple Gaussian mixtures, which is a technique known from the 60ties (Baum & Eagon, 1967). As this statistical concept is mathematically tractable it, unfortunately, doesn't completely reflect the physical underlying process. Therefore soon after its creation there have been lot of attempts to alleviate that. Nowadays the classical concept of HMM has evolved into areas like hybrid solutions with neural networks, utilisation of different than ML or MAP training strategies that minimize recognition errors by the means of corrective training, maximizing mutual information (Huang et. al., 1990) or by constructing large margin HMMs (Jiang & Li, 2007). Furthermore, a few methods have been designed and tested aiming to suppress the first order Markovian restriction by e.g. explicitly modelling the time duration (Levinson, 1986), splitting states into more complex structures (Bonafonte et al., 1996), using double (Casar & Fonollosa, 2007) or multilayer structures of HMM. Another vital issue is the robust and accurate feature extraction method. Again this matter is not fully solved and various popular features and techniques exist like: MFCC and CLPC coefficients, PLP features, TIFFING (Nadeu & Macho, 2001), RASTA filter (Hermasky & Morgan, 1994), etc.

Even despite the huge variety of advanced solutions many of them are either not general enough or are rather impractical for the real-life employment. Thus most of the currently employed systems are based on continuous context independent (CI) or tied CD HMM models of phonemes with multiple Gaussian mixtures trained by ML or MAP criteria. As there is no analytical solution of this task, the training process must be an iterative one (Huang et al., 1990). Unfortunately, there is no guarantee of reaching local maxima, thus lot of effort is paid to the training phase in which many stages are involved. Thus there are some complex systems that allow convenient and flexible training of HMM models, where the most famous are HTK and SPHINX.

This chapter provides you with the description of some basic facilities and methods implemented by HTK and SPHINX systems and guides you through a thorough process of building speaker independent CDHMM models using the professional database MOBILDAT-SK (Darjaa et al., 2006). First, basic tolls for building practical HMM models are described using HTK and SPHINX facilities. Then several experiments revealing the optimal tuning of the training phase are discussed and evaluated ranging from: selecting feature extraction methods and their derivatives, controlling and testing the overtraining phenomenon, selecting modelled units: CI and CD phonemes vs. models of functional words, setting proper tying options for CD phonemes, etc. Further, the popular training procedures for HTK and SPHNIX systems will be briefly outlined, namely: REFREC (Linderberg et al., 2000) / MASPER (Zgank & Kacic, 2003) and SphinxTrain (Scriptman, 2000). After the presentation of both training schemes the newly suggested modifications are discussed, tested and successfully evaluated. Finally, the achieved results on both systems are compared in terms of the accuracy, memory usage and the training times.

Thus the following paragraphs should give you the guideline how to adjust and build both robust and accurate HMM models using standard methods and systems on the professional database. Further, if it doesn't provide you with the exact settings, because they may be language specific, at least it should suggest what may be and what probably is not so relevant in building HMM models for practical applications.

2. Systems for building robust HMMs

2.1 Hidden Markov Toolkit- HTK

The HTK system is probably the most widely employed platform for training HMM models. Its outputs (HMMs) adhering to the suggested training steps are regarded as a sort of standard and are believed to be eligible for real life, large vocabulary ASR systems. The latest version is 3.4, however, the results are related to 3.2.1 version (Young et al., 2002).

HTK is a complex tool that provides advanced and flexible means for any stage of the HMM training: speech and data processing, definition of HMM models (discrete, continuous and semi-continuous), dictionary related tasks, initializations and training methods, model enhancement and adaptation tools, it can use both finite state grammar (BNF) or statistical language models via N grams, has tools for online as well as offline recognition and implements various evaluation methods, etc. Furthermore, each release is accompanied with a very precise documentation.

As some of the facilities would be directly involved in our experiments let us mention them. HTK supports many speech extraction methods like: various filter banks, LPC and CLPC coefficients, PLP, and MFCC parameters. Except that several auxiliary features are available like: normalized energy, differential and acceleration coefficients, cepstral mean subtraction and vocal tract length normalization. It supports and process description files with or without time labels. When the time information is available the Viterbi training can be used for the initialization phase and the classical Baum-Welche algorithm for the training, otherwise the flat start method and the embedded training are the only options. Moreover, to speed up the training process and to eliminate possible error recordings, both forward and incremental backward pruning methods can be applied. HTK supports discrete (DHMM), continuous (CDHMM) and semi-continuous (SCHMM) models. Any structure of the transition matrix is allowed and can differ from model to model. Moreover there are two non- emitting states at both ends of each model which allow the construction of the so called

T model. Furthermore, each element of the model (means, variances, mixtures, etc.) can be tied to the corresponding elements of other models. In HTK there are implemented 2 methods for parameter's tying, namely: the data driven one and the decision trees. The decoder supports forced alignment for multiple pronunciations, and the time alignment that can be performed on different levels and assess multiple hypotheses as well. To ease the implementation for online systems a separate recognition tool called ATK has been released. Of course an evaluation tool supporting multiple scoring methods is available.

2.2 SPHINX

The SPHINX system is eligible for building large vocabulary ASR systems since late 80ties (Lee et al., 1990). Currently there are SPHINX 2, 3, 3.5 and 4 decoder versions and a common training tool called SphinxTrain. The latest updates for SphinxTrain are from 2008, however, here mentioned options and results will refer to the version dated back to 2004. Unfortunately, the on-line documentation is not extensive, so the features mentioned here onwards are only those listed in manuals dated back to 2000 (Scriptman, 2000).

SphinxTrain can be used to train CDHMM or SCHMM for SPHINX 3 and 4 decoders (conversion for version 2 is needed). SphinxTrain supports MFCC and PLP speech features with delta or delta-delta parameters. Transcription file contains words from a dictionary, but neither multilevel description nor time labels are supported. There are 2 dictionaries, the main for words (alternative pronunciations are allowed but in the training process are ignored), and the second one is the so called filler dictionary where non-speech models are listed. The main drawback is the unified structure of HMM models that is common to all models both for speech and non-speech events. At the end of each model there is one non-emitting state, thus no "T" model is supported. Further, it is possible to use only the embedded training and the flat start initialization processes. Observation probabilities are modelled by multi mixture Gaussians and the process of gradual model enhancement is allowed. SphinxTrain performs tying of CD phonemes by constructing decision trees; however no phoneme classification file is required as the questions are automatically formed. Instead of setting some stoppage conditions for the state tying, the number of tied states must be provided by the designer prior to the process which expects deep knowledge and experience. Unlike HTK, only whole states can be tied. Apart of the SphinxTrainer there is a statistical modelling tool (CMU) for training unigrams, bigrams and trigrams.

3. Training database MOBILDAT–SK

A crucial aspect to succeed in building an accurate and robust speaker independent recognition system is the selection of the proper training database. So far there has been designed and compiled many databases following different employment assumptions and designing goals, like: AURORA, TIMIT, SPEECHDAT, SPEECON, etc. Further, the task of recognition is more challenging in adverse environments and requires more steps, additional pre-processing and more sophisticated handling. Since we want to demonstrate to the full extend the capabilities, options, modification and pitfalls of the HMM training process, we decided to use the Slovak MOBILDAT database (Darjaa et al., 2006) which was recorded over GSM networks and generally provides more adverse environments (wider range of noises, lower SNRs, distortions by compression techniques and short lapses of connections). The concept of MOBILDAT database is based on the widely used structure of

the SPEECHDAT database, whose many versions have been built for several languages using fix telephone lines and are regarded as professional databases.

The Slovak MOBILDAT-SK database consists of 1100 speakers that are divided into the training set (880) and the testing set (220). Each speaker produced 50 recordings (separate items) in a session with the total duration ranging between 4 to 8 minutes. These items were categorized into the following groups: isolated digit items (I), digit/number strings (B,C), natural numbers (N), money amounts (M), yes/no questions (Q), dates (D), times (T), application keywords (A), word spotting phrase (E), directory names (O), spellings (L), phonetically rich words (W), and phonetically rich sentences (S, Z). Description files were provided for each utterance with an orthographical transcription but no time marks were supplied. Beside the speech, following non- speech events were labeled too: truncated recordings (~), mispronunciation (*), unintelligible speech (**), filed pauses (fil), speaker noise (spk), stationary noise (sta), intermitted noise (int), and GSM specific distortion (%). In total there are 15942 different Slovak words, 260287 physical instances of words, and for 1825 words there are more than one pronunciation listed (up to 5 different spellings are supplied). Finally, there are 41739 useable speech recordings in the training portion, containing 51 Slovak phonemes, 10567 different CD phonemes (word internal) and in total there are slightly more than 88 hours of speech.

4. Robust and accurate training process

Our final goal is to choose a proper training scheme and adjust its relevant parameters in order to get robust and accurate HMM models that can be used in practical large vocabulary applications like: dialog systems, information retrieval systems and possibly dictation or automatic transcription systems. However, issues regarding the construction of stochastic language models represented by unigrams, bigrams or generally N- grams are out of the scope of this document. Thus in this section, aspects like: eligible speech features, HMM model structures and modelled units, overtraining phenomenon, and the tying of states will be discussed.

4.1 Feature extraction for speech recognition

One of the first steps in the design of an ASR system is to decide which feature extraction technique to use. At the beginning it should be noted that this task is not yet completely solved and a lot of effort is still going on in this area. The aim is to simulate the auditory system of humans, mathematically describe it, simplify for practical handling and optionally adapt it for a correct and simple use with the selected types of classification methods.

A good feature should be sensitive to differences in sounds that are perceived as different in humans and should be "deaf" to those which are unheeded by our auditory system. It was found (Rabiner & Juan, 1993) that the following differences are audible: different location of formants in the spectra, different widths of formants and that the intensity of signals is perceived non-linearly. On the other hand, following aspects do not play a role in perceiving differences: overall tilt of the spectra like: $X(\omega)\omega^\alpha$, where α is the tilt factor and $X(\omega)$ is the original spectra, filtering out frequencies laying under the first formant frequency, removing frequencies above the 3rd format frequency, and a narrow band stop filtering.

Furthermore, features should be insensitive to additive and convolutional noises or at least they should represent them in such a way that these distortions are easy to locate and

suppress in the feature space. Finally, when using CDHMM models it is required for the feasibility purposes that the elements of feature vectors should be linearly independent so that a single diagonal covariance matrix can be used. Unfortunately, yet there is no feature that would ideally incorporate all the requirements mentioned before.

Many basic speech features have been designed so far, but currently MFCC and PLP (Hönig et al., 2005) are the most widely used in CDHMM ASR systems. They both represent some kind of cepstra and thus are better in dealing with convolutional noises. However, it was reported that some times in lower SNRs they are outperformed by other methods, e.g. TIFFING (Nadeu & Macho, 2001). Furthermore, the DCT transform applied in the last step of the computation process minimize the correlation between elements and thus justifies the usage of diagonal covariance matrices. Besides those static features it was soon discovered that the changes in the time (Lee at al., 1990) represented by delta and acceleration parameters play an important role in modelling the evolution of speech. This is important when using HMMs as they lack the natural time duration modelling capability. Overall energy or zero cepstral coefficients with their derivations also carry valuable discriminative information thus most of the systems use them. Furthermore, to take the full advantage of cepstral coefficients, usually a cepstral mean subtraction is applied in order to suppress possible distortions inflicted by various transmission channels or recording devices. At the end we shall not forget about the liftering of cepstra in order to emphasise its middle part so that the most relevant shapes of spectra for recognition purposes would be amplified. Well, this appealing option has no real meaning when using CDHMM and Gaussian mixtures with diagonal covariance matrices. In this case it is simply to show that the liftering operation would be completely cancelled out when computing Gaussian pdf.
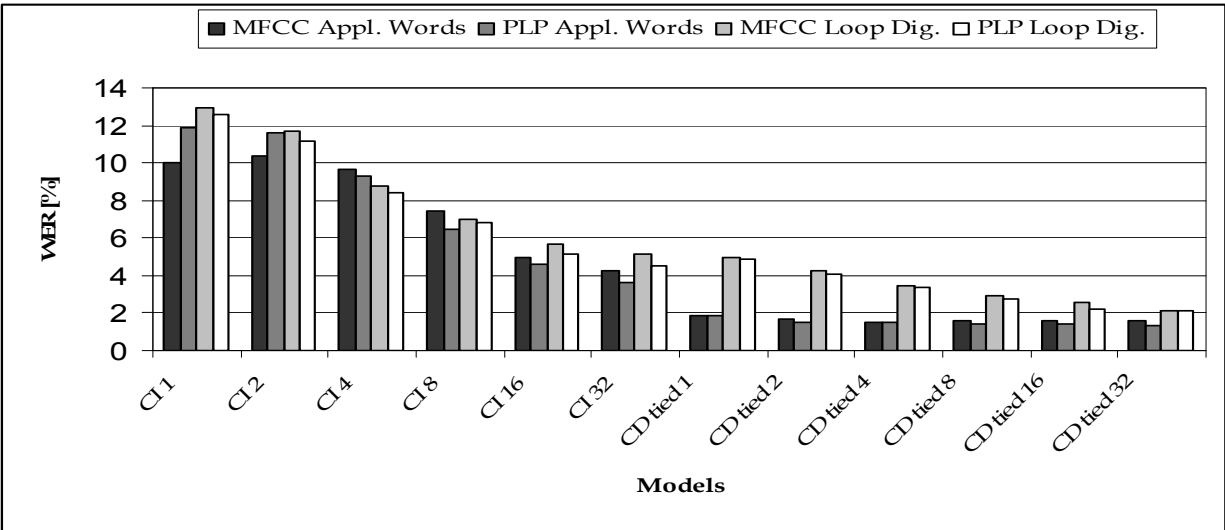

Fig. 1. Word error rates of PLP and MFCC features for application words and looped digits tests as a function of HMM models (CI and tied CD with multiple mixtures).

All the above-mentioned features and auxiliary settings were tested and evaluated on our database in terms of the recognition accuracy. Three tests were done on the test set portion of the database: single digits, digits in the loop, and application words. The training was based on the MASPER training procedure (will be presented later in the text) using the HTK system. In fig. 1 there are shown results for both MFCC and PLP features with delta, acceleration, and C0 coefficients, modified by the mean subtraction (this setting showed the

best results for both features). These were calculated over different HMM models (CI and tied CD phoneme models with multiple Gaussian mixtures) and both the application words and looped digit tests. From these 2 tests one can induce that slightly better results are obtained by PLP method, but in order to get a numeric evaluation of the average WER for all the models, both tests for MFCC and PLP were computed separately. These averaged errors over models and tests revealed that PLP is slightly better, scoring 20.34% while MFCC showed 20.96% of WER that amounts to a 3% drop in an average word error rate. Further, we investigated the significance of auxiliary features and modification techniques. For both methods the cepstral mean subtraction brought essentially improved results on average by 33.83% for MFCC and 21.28% for PLP. That reveals the PLP is less sensitive to the cepstral mean subtraction, probably, because it uses non linear operations (equal loudness curve, 0.33 root of the power, calculation of the all pole spectra) applied prior to the signal is transformed by the logarithm and cepstral features are calculated. Next the role of C0 (zero cepstral coefficient) was tested and compared to the solely static PLP and MFCC vectors, where it brought relative improvement by 19.7% for MFCC and 9.7% for PLP, again PLP showed to be less sensitive to the addition of a static feature or modification. Next the inclusion of delta coefficients disclosed that their incorporation brought down the averaged error by 61.15% for MFCC and 61.41% for PLP. If this absolute drop is further transformed to the relative drop calculated over a single difference coefficient (if all are equally important), it shows that one delta coefficient on average causes a 4.7% WER drop for MFCC and 4.72% for PLP. Finally, the acceleration coefficients were tested, and their inclusion resulted in a 41.16% drop of WER for MFCC and 52.47% drop for PLP. Again, if these absolute drops are calculated for a single acceleration coefficient it was found that one such a coefficient causes on average a 3.16% drop of WER for MFCC and a 4.03% for PLP. Interestingly enough, both dynamic features caused to be more significant for PLP than for MFCC in relative numbers, however, for the additional C0 (static feature) this was just the opposite. That may suggest that PLP itself (in this task) is better in extracting static features for speech recognition.

4.2 Discrete, continuous or semi-continuous HMM

The issue of the observation probability modelling will not be experimentally tested here. There have been many discussions and experiments on which type of HMM models is better in certain environments, etc. (Huang et al., 1990), but let us just in brief mention several fact and findings. The usage of discrete HMM (non-parametric modelling of the distribution) has clear advantage of being able to model any distribution, however, it requires huge amount of data to do so and moreover it uses the VQ procedure. This powerful technique introduces an irreversible distortion and is based on a vague notion of the acoustic distance (Rabiner & Juan, 1993). On the other hand, Gaussian mixture model (CDHMM) does not introduce anything like that, but requires lot of computing and for some complex models (lot of free parameters) it may not produce robust estimations. To eliminate both of those problems, semi-continuous HMM were introduced and showed better result in some applications (Huang et al., 1990). However, the fact that all Gaussain mixtures are trained on and share the same data which for some phonemes are from linguistic and physical pint of view completely different, poses an accuracy problem.

Even though the most successful systems are based on CDHMM, in some applications with higher degree of noise presence DHMM or discrete mixture HMM (DMHMM) were

reported to provide more accurate results (Kosaka et al., 2007). Usually this is explained by the inability of Gausian mixture pdf to model the occurrence of noisy speech. However, this is not the case as for example, the theoretical results from the artificial neural networks domain, namely the radial bases function (RBF), roughly say that a RBF network can approximate any continuous function defined on a compact set with the infinitely small error (Poggio & Girosi, 1990). Thus it poses as a universal approximator. If we compare the structure of a RFB network with N inputs (size of a feature vector), M centres (Gaussian mixtures) and one output (probability of a feature vector) we find out that these are actually the same. Generally, Gaussian mixtures can be viewed as an approximation problem how to express any continuous function of the type f: $R^N \rightarrow R$ by the means of sum of Gaussian distributions, which is just what the RBF networks do. Thus the derived theoretical results for RBF must also apply to this CDHMM case regarding the modelling ability of Gaussians mixtures. Unfortunately, the proof says nothing about the number of mixtures (centres). Therefore, based on these theoretical derivations we decided to use CDHMM without additional experiments.

4.3 Modeling of speech units

Another issue to be tackled before building up an ASR system is to decide which speech and non speech units to model. In the early times where only small vocabulary systems were needed the whole word approach was the natural choice and exhibited good results. This method is rather unpractical for open systems where new words should be easily added and totally infeasible for large vocabulary, speaker independent applications where several dozens of realizations for every word are required. The opposite extreme is to construct models only for single phonemes which would solve the flexibility and feasibility problems. However, pronunciations of phonemes are strongly dependent on the context they are uttered in, the so called coarticulation problem, which caused a sudden drop in the accuracy. Next natural step was the inclusion of the context for each phoneme, both the left and right equally, which resulted in the context dependent phonemes (Lee et al., 1990). This obviously increased the accuracy but because of their huge number the problem of robustness re-emerged. Theoretically there are (in Slovak) 51^3 CD phonemes but practically there are about 11000 of them in 80 hours of recorded speech (specially constructed database in order to contain phonetically rich sentences like MOBILDAT). Thus the latest step led to the building of tied CD phonemes where phonetically and /or feature similar logical states are mapped to the same physical one. To allow more degrees of freedom usually states of models are tied instead of the whole models, however, if all states of a model are tied to another one then, such a couple creates one physical model. Still, the option of modelling the whole words may be appealing especially for application words that are quite frequent and usually exhibit huge variation in their sound forms (Lee et al., 1990). Furthermore, frequent and critical words or even phrases that are vital from the accuracy point of view, like yes and no, digits etc. can be modelled separately as unique models (Huang et al., 1990). Except speech units also other non-speech events have to be modelled so that a whole conversation could be correctly expressed by the concatenated HMM models. Then those usual events in a real conversation must be identified and classified according to their location, origin of creation and physical character. It is common to use a general model of a background which should span different time intervals and thus must allow backward connections. Events that are produced by the speaker himself (exhaling, sneezing, coughing,

laughing etc.) can be modelled by a unique model, but to increase the modelling ability these events are further divided into groups e.g. sound produced unintentionally like coughing, sneezing, etc. and intentional sounds like laughing, hesitating- various filling sounds, etc.
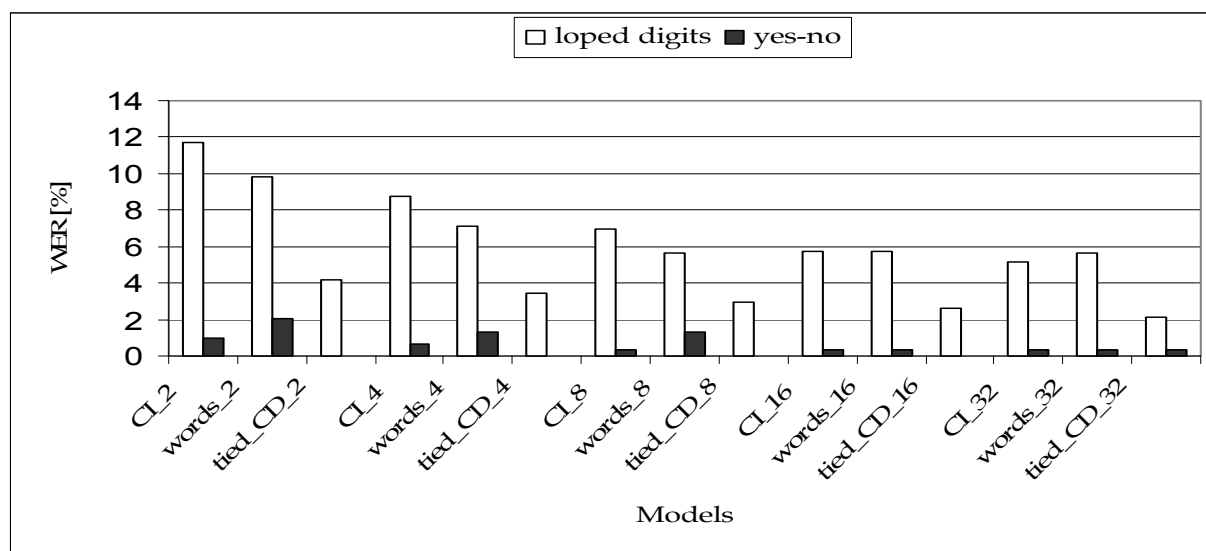

Fig. 2. Word error rates for CI, whole word, and tied CD phoneme models with different number of mixtures and both looped digit and yes-no tests. There are 3 states per a phoneme and a strictly left right structure of models.

To determine which models perform best and at what cost the following experiments were executed. CI and tied CD phoneme models were train with different number of Gaussian mixtures as was suggested by the REFREC or MASPER training schemes. To verify the effectiveness of the whole word models, models for digits and yes / no words were constructed as well. The whole word models consisted of the same number of states as their phoneme-concatenated counterparts and followed the same transition matrix structure (strictly left right, no skip). However, to utilize more efficiently the whole word models in mimicking the co-articulation effect, the HMM structure was enhanced so as to allow a one state skip. This structure was tested for whole word models as well as for CI and CD models created by the MASPER training scheme. In fig. 2 there are shown results for whole word models, CI and tied CD phoneme models with different number of mixtures, 3 states per a phoneme, and with a strictly left-right structure, for looped digits and yes / no tests. The same results for the left - right structure with one allowed state to be skipped and 3 states per a phoneme are depicted in fig. 3.

For the strict left right structure of HMM models there is surprisingly very little difference in terms of averaged WER between CI phonemes 4.09% and whole word models 3.94%. The tied CD phoneme models outperformed even the whole word models scoring on average only 1.57% of WER. Similar tests with the one state skip structure however, brought different results as seen in fig. 3. The averaged WER for CI models is 5.46%, tied CD models scored 3.85% and the whole word models 3.12%. These results deserve few comments. First an obvious degradation of WER for CI by 33.4% and tied CD phoneme models by 145% when moving from strictly left - right structure to the one state skip structure that potentially allows more modelling flexibility. By introducing additional skips the minimal

occupancy in a phoneme model has reduced to only one time slot (25ms) comparing to the original 3 (45ms) which is more realistic for a phoneme. By doing so some phonemes were in the recognition process passed unnaturally fast, that ended up in a higher number of recognized words. This is known behaviour and is tackled by introducing the so called word insertion penalty factor that reduces the number of words the best path travels through. In the case of short intervals like phonemes there is probably only a small benefit in increasing the duration flexibility of a model that is more obvious for CD models as they are even more specialized. On the other hand, when modelling longer intervals like words which have strong and specific co-articulation effects inside, the increased time duration flexibility led to the overall improved results by 4.5%. However, when comparing the best tied CD phoneme models with the best word models, the tied CD models still provided better results. This can be explained by their relatively good accuracy as they take in account the eminent context and their robustness because they were trained from the whole training section of the database. On the contrary, the word models were trained only from certain items, like digits and yes - no answers, so there might not have been enough realizations. Therefore the appealing option for increasing the accuracy by modelling whole functional words should not be taken for granted and must be tested. If there is enough data to train functional word models then the more complex structures are beneficial.
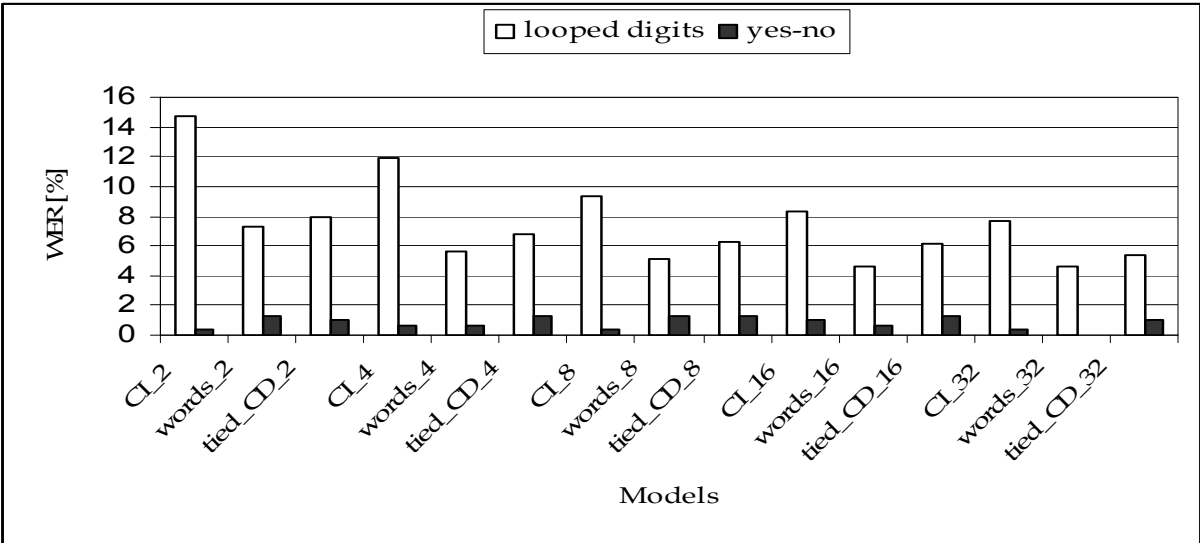

Fig. 3. Word error rates for CI, whole word, and tied CD phoneme models with different number of mixtures and both looped digit and yes-no tests. There are 3 states per a phoneme and a left right structure of HMM models with one allowed state to skip.

4.4 Overtraining phenomenon for HMM

Using the limited data which do not cover the whole feature space proportionally and being restricted to the models that only roughly approximate the physical underlying process, it was soon realized that during the course of training and despite to the assured convergence of the training algorithms, results started to deteriorate on the unseen data at the certain point. Although this phenomenon is ubiquities in all real classification and decision taking algorithms, some methods like artificial neural networks, decision trees, etc. are well known to be vulnerable. In brief, the overtraining is simply explained by the fact that as the training

converges on the training data, the models or classifiers are getting too specific about the training data and inevitably start to loose a broader view over a particular task (loosing the generalization ability). There are more methods to detect and eliminate the overtraining but let's mention some of them: the usage of test sets, restricting the complexity of models, gathering more general training data, setting floors for parameters, etc.
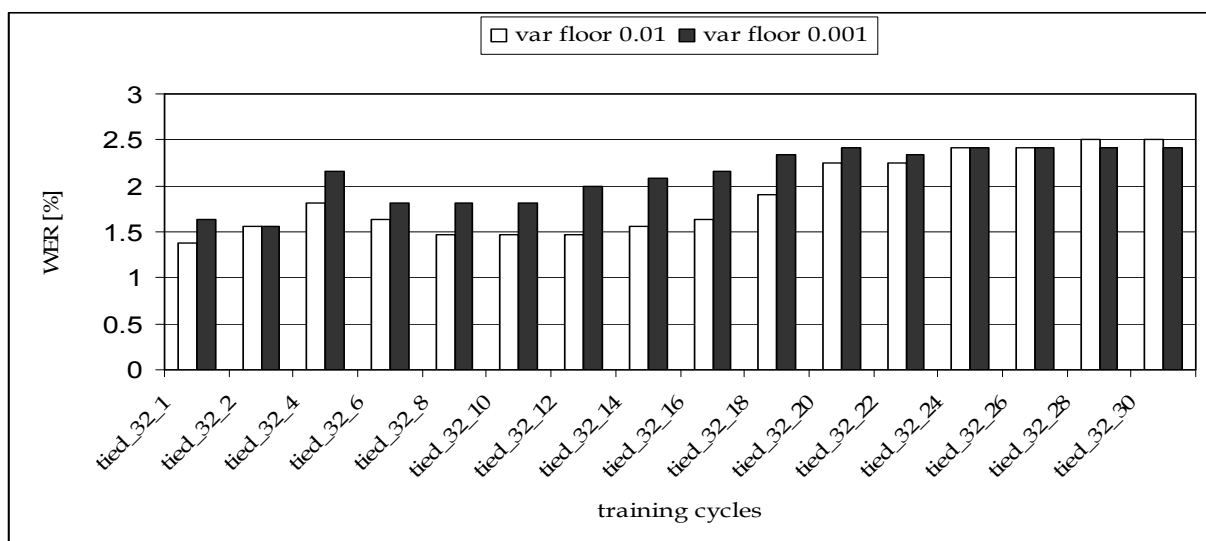

Fig. 4. WER as a function of the training cycles for tied CD phonemes with 32 mixtures, two variance floors (1% and 1‰), and evaluated for the application words test.

Although this phenomenon usually doesn't pose too serious problems regarding the HMM concept, in practical settings it must be dealt with. As the available data are limited, the most effective ways to prevent this phenomenon to happen are: restrict the number of training cycles, set up the variance floors for Gaussian pdf, tie similar means, variances, mixtures, states, and even models, do not construct models with too many Gaussian mixtures, and check the performance of HMM models on the test set. To examine and enumerate the above mentioned methods on the professional database, we decided to accomplish following test: we exposed models that are most prone to the overtraining (with more than 4 mixtures both CI and tied CD phoneme models), to 30 additional training cycles using 2 different variance floors (1% and 1‰ of the overall variance). Again, the training followed the MASPER training scheme for CI and tied CD phoneme models with 1 up to 32 Gaussian mixtures. In fig. 4 there is depicted the WER measure as a function of training cycles for tied CD phonemes with 32 mixtures and both variance floors (1% and 1‰) for the application words test. The same results but for CI phonemes are in fig. 5.

As it can be seen from fig. 4 additional trainings caused the rise of WER for both variance floors, however, WER got stabilized. Bit different situation was observed for CI phonemes were the extra training cycles caused the WER do drop further, but this decrease after 6 or 8 iterations stopped and remained at the same level, for both variance floors. This can be due to a large amount of samples for CI HMM models of phonemes. For the tied CD phonemes the higher sensitivity to the overtraining was observed, which is not a surprise as these models are much more specialized. In both cases the selected values for variance floors provided similar final results. This can be viewed that both floors are still rather low to completely prevent the overtraining given the amount of training samples and the

complexity of models. However, the experiments proved that the original training scheme and the settings on the given database are in eligible ranges and are reasonably insensitive to the overtraining. On the other hand, it was documented that the extensive training may not bring much gain, and it can even deteriorates the accuracy.
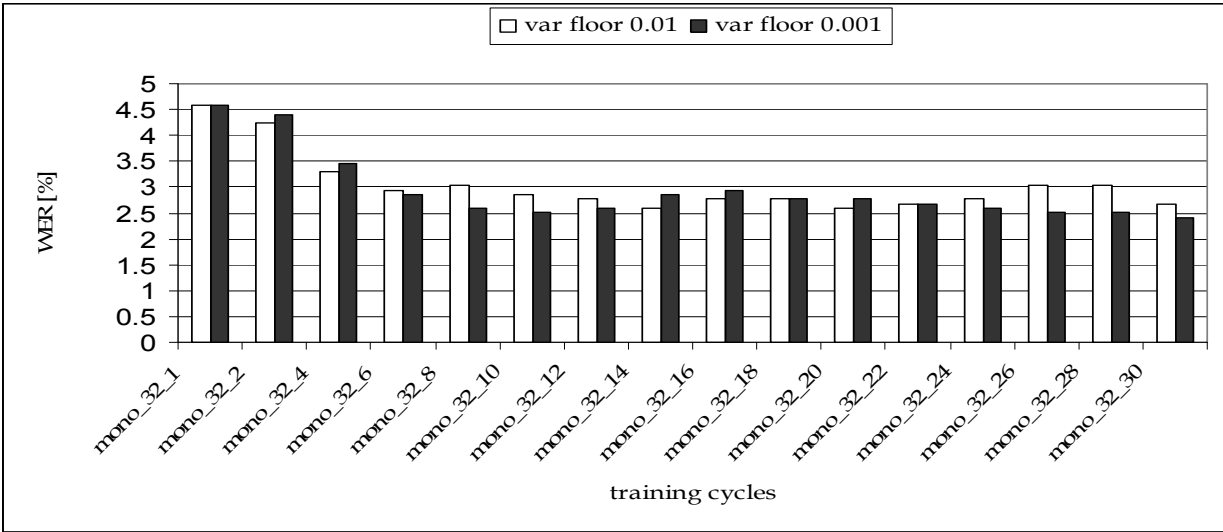

Fig 5. WER as a function of the training cycles for CI phonemes with 32 mixtures, two variance floors (1% and 1‰), and evaluated for the application words test.

4.5 Tying process for context dependent phoneme models

CD phonemes are the most frequently modeled sub-word units in practical large vocabulary systems. However, this would not be possible without tying similar mixtures, states or even whole models of phonetically and/or feature similar units. There are more method to do the tying but probably the most popular are the data based clustering and the decision tree method (Young et al., 2002). The data based clustering tries to merge predefined states by the designer so that the largest cluster reaches the maximum size (it is the largest distance between any two states in the cluster that are to be tied). Process begins with simple states and progress further by merging closest groups. Except the maximal distance, it is advantageous to set up the minimal number of frames per cluster (occupation count) that it will be trained from. On the other hand the decision tree clustering follows different strategy. It uses language questions based on predefined classification of phonemes which are language specific. These questions are asked about the left and right context of a CD phoneme. Based on each question the original cluster is divided according to the answer, positive or negative. Resulting two clusters are better in modeling the data than the original one and thus this separation causes the increase of the modeling probability of the data. Only those clusters are left which caused the highest increase. The process is stopped if the highest increase is less than the predefined constant. To prevent forming clusters with insufficient data the minimal occupation count is set. The greatest advantage of this method lies in the existence of the decision trees that can be preserved and later used to synthesized CD phonemes that were not present in the training phase.

Following the above mentioned discussion we decided to use the decision tree based tying method. However, to get the "optimal" result one must experiment with the proper settings. We were able to classify 40 different groups of phonemes in Slovak, in order to produce the

splitting questions. There is no harm in providing as many as possible questions because all are tested and only those which are really relevant take place. Thus there were two options to set: minimal log likelihood increase and the minimal number of the training frames per a cluster. In the MASPER procedure as well as in the HTK book example these were set to 350 for the minimal log likelihood increase and 100 for the minimal occupation count. As these options depend on each other we tested both, ranging from 50% less to 50% more than the suggested settings. These settings are language specific, and moreover, they depend on the size and type of the database (how many different CD phonemes are there, how many realizations, what are their acoustic dissimilarities, etc.). Increasing both values leads to more robust, but less precise models as well as to lower number of physical states. Of course, their decrease would have the opposite effect. Thus this is the place for experiments and the final tuning for most systems. First, in tab. 1 there are averaged WER and relative improvements for tied CD phoneme models over application words and looped digits tests. Originally suggested values were shifted in their values by: ±50%, ±25%, and 0%.

	-50%	-25%	original	25%	50%
settings	175 log prob. 50 occup.	280 log prob. 80 occup.	350 log prob. 100 occup.	420 log prob 120 occup.	525 log prob. 150 occup.
average WER	2.52	2.53	2.55	2.56	2.52
improvement %	1.12	0.97	0	-0.31	1.07

Table 1. Average WER for tied CD phoneme models for application words and looped digits tests as a function of the minimal log likelihood increase and the minimal occupation count.

Min. occupation count =100	Minimal log likelihood increase		
	100	200	350
average WER	2.49	2.51	2.55
relative improvement %	2.44	1.81	0

Table 2. Averaged WER and relative improvements for tied CD phoneme models over application words and looped digit tests, with the minimal occupation count set to 100.

These results don't clearly show the preferred direction for finding the optimal parameters, i.e. whether to increase the robustness or accuracy. Therefore, as the minimal number for the occupation count the 100 threshold value was accepted. This assures that even for the most complex models with 32 mixtures, there will be on average at lest 3 observations for a single mixture. This seems to be the lowest reasonable number from the robustness point of view. In table 2 there are listed results for several minimal log likelihood increases keeping the occupancy count fixed.

As it can be seen the best result is approximately 100 (minimal log likelihood increase) for the 100 threshold value of the minimal occupancy count. This suggests that making the HMM models more specialized (there are more splits) brought additional 2.4% decrease in the averaged WER comparing to the original settings.

5. Training of HMMs using HTK system

There are many successful ASR systems that were trained by the HTK tools. In our experiments with HTK we decided to use the MASPER (Zgank et. al., 2004) training scheme which is a cross-lingual counterpart of the reference recognition system REFREC 0.96 (Lindberg et al., 2000). Furthermore, both procedures closely cooperate with SPEECHDAT or MOBILDAT databases and handle all relevant aspects of building robust HMM.

5.1 MASPER / REFREC training procedures

The general concept of REFREC 0.96 is based on that one presented in the HTK documentation, however enhanced to serve for multilingual purposes. During the course of the run it produces following models: flat start CI phoneme models with 1 up to 32 Gaussians mixtures, CI models generated in the 2nd run that are initialized on the time aligned data, CD models (with only one Gaussian) and tied CD models with 1 to 32 Gaussian mixtures. On the evaluating part of the training there are 3 small vocabulary tests provided for all models involving: application words, single digits and digits in the loop. REFREC 0.96 uses MFCC speech features with C0, delta and acceleration coefficients that make up a vector with 39 elements. Utterances that are in a way damaged by the existence of GSM noise (%), unintelligible speech (**), mispronunciation (*) and truncation (~) are removed. For speaker produced noises (spk) and hesitations (fil) separate models are used while the other markers are ignored. The training starts with the flat start initialization and a variance floor is also calculated. This initialization and first cycles of embedded training are executed over the phonetically reach utterances with the creation of SIL (general background) and SP (short pause T model) models. Then the Viterbi forced alignment utilizing multiple pronunciations is done as well as the acoustically "suspicious" recordings are removed. Next, the process goes on in cycles of two training passes followed by a mixture incrementing stage by the factor of 2 as far as 32 mixtures are reached. These final models are used to do the time alignment over all utterances so that the Viterbi initialization and the single model training of CI phonemes can be done in the second stage of the training. CI phonemes derived in the second stage with 1 Gaussian mixture are used for cloning CD phonemes. In this way more accurate models of CI phonemes with 1 mixture are obtained than those trained in the first run. These single-mixture models are further used by the cloning and tying procedures in the construction of tied CD models. After the cloning, the CD models are trained in 2 cycles of the embedded training and then tied which is done by the decision tree algorithm. After the tying, the gradual process of two re-estimations passes interleaved by mixtures incrementing stage is repeated up to the 32 mixtures are reached. Finally, CI phoneme models from the second run are enhanced and trained in cycles using the embedded training up to 32 mixtures.

To enable an effective design of the multilingual and cross-lingual ASR systems some further modification must have been done to the REFREC 0.96, which resulted in the MASPER procedure. These changes are as follows: cepstral mean subtraction, modifications to the parameters of tree based clustering, and the production of the training statistics.

5.2 Proposed modification to the MASPER training scheme

As we can see, REFREC 0.96 or MASPER are advanced procedures for building mono, multi or cross-lingual HMM models for large vocabulary ASR systems. However, we discovered

some deficiency of these schemes in handling the training data, i.e. the removal of all utterances partially contaminated with truncated, mispronounced and unintelligible speech even though the rest of the recording may be usable. Thus in the following the modification to the MASPER procedure aiming to model the damaged parts of the speech while preserving useful information will be presented and evaluated.

Let's start with some statistic regarding the portion of damaged and removed speech. After the rejection of corrupted speech files there were in total 955611 instances of all phonemes. The same analysis applied just to the rejected speech files has discovered further 89018 realizations of usable phonemes, which amounts to 9.32% of all appropriate phoneme instances. More detailed statistic regarding the recordings, CI and CD phonemes used by MASPER and modified MASPER procedures on MOBILDAT –SK is summarized in table 3.

Statistics of the database	MASPER	modified MASPER	Absolute increase	Relative increase
recordings	40861	43957	3096	7,58%
CI phonemes	51	51	0	0%
CD phonemes	10567	10630	63	0,60%
instances of CI phonemes	955611	1044629	89018	9,32%
average number of instances per a CD phoneme	~90.4	~98.27	~7.84	~8.7%

Table 3. Statistics of CI and CD phonemes contained in MOBILDAT SK that are utilized by MASPER and modified MASPER procedures.

To be more specific, in fig. 6 there are depicted realizations of Slovak phonemes used by MASPER and modified MASPER procedures. The modified MASPER procedure preserves eligible data from the damaged recordings by using a unified model of garbled speech that acts as a patch over corrupted words. These words are not expanded to the sequence of phonemes, but instead, they are mapped to a new unified model of garbled speech, the so called BH model (black hole- attract everything). Then the rest of a sentence can be processed in the same way as in the MASPER procedure. The new model is added to the phoneme list (context independent and serves as a word break) and is trained together with other models. However, its structure must be more complex as it should map words of variable lengths spoken by various speakers in different environments.

Following this discussion about the need for a complex model of garbled words while having limited data, there are two related problems to solve: the complexity of such a model and its enhancement stages. From the modelling point of view the ergodic model with as many states as there are speech units would be the best, however, it would extremely increase the amount of parameters to estimate. As it was expected there must have been tested more options ranging from the simplest structures like a single state model to models with 5 states (emitting). At the end of the training it was assumed that this model should be ergodic, just to get the full modelling capacity, which is not strictly related to the time evolution of the speech.
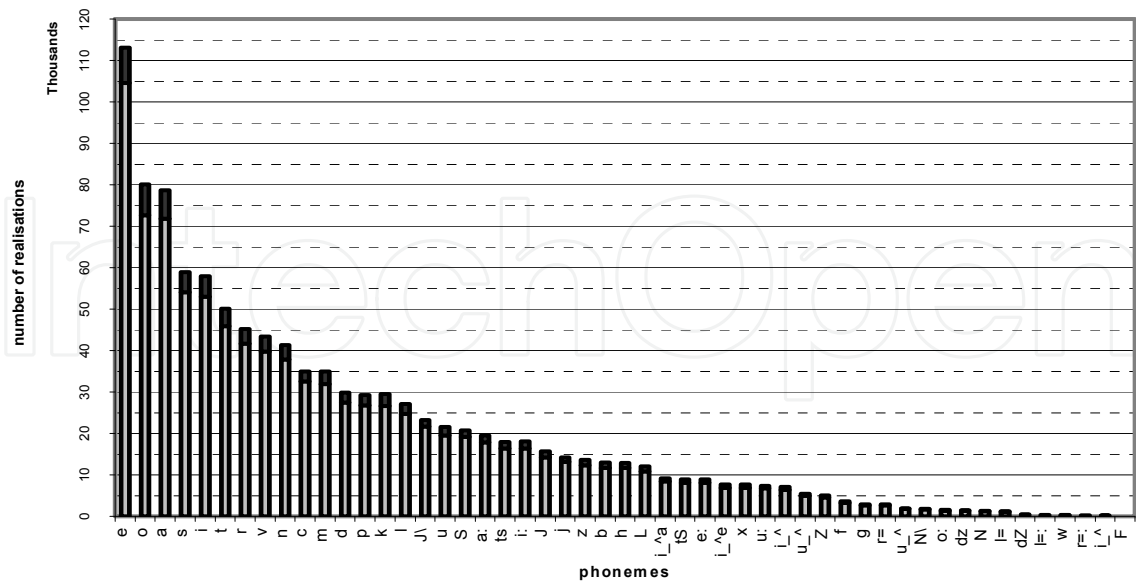
Fig. 6. Number of instances for each phoneme in MOBILDAT-SK processed by MASPER and modified MASPER (black tops)

Initial structure	Method of enhancement	Final structure
Stricly left-right, start and end in the first and last emitting states	Addition of all backward connection, no T model	ergodic
ergodic	no	ergodic
Left-right, with all forward connections	Addition of backward connections	ergodic
Left right, with all forward connections	Addition of backward connections, single model training of BH in the second stage	ergodic

Table 4. Tested initial structures, methods of enhancement and final structures for BH model.

Furthermore, there were more strategies how to enhance the original BH model so that in a final stage it would be ergodic. In table 4 there are summed up all the tested possibilities of original and final structures of the BH model as well as the applied modifications during the training. All BH models and updating methods were evaluated by standard tests used in MASPER (single digits, looped digits and application words). From all these experiments the 5 state BH model with initial structure allowing all forward connections showed to be slightly better then remaining ones. In fig. 7, there are shown results for CI and CD phoneme models and the looped digits test (perplexity= 10).
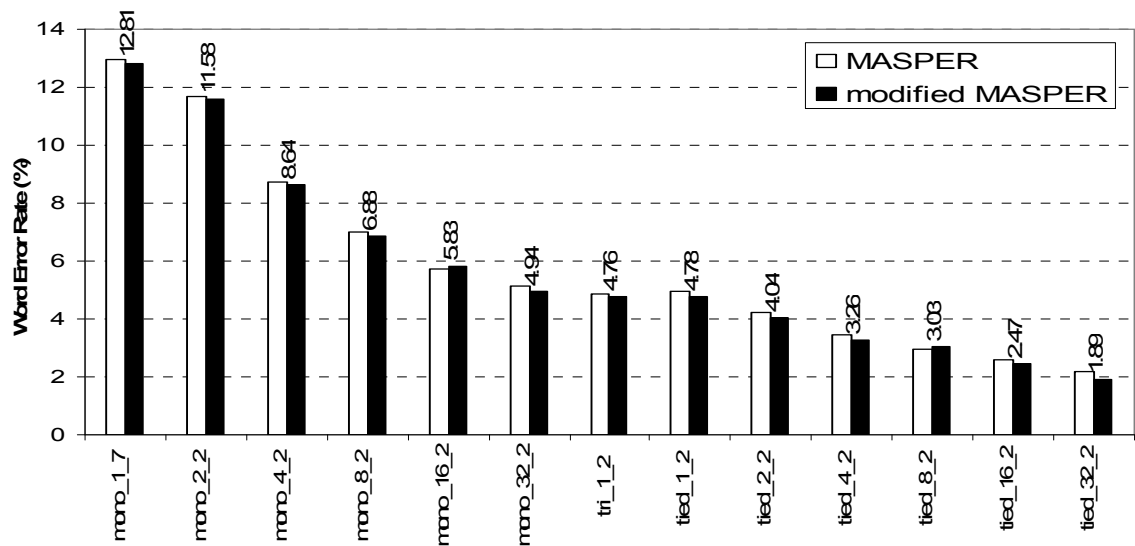
Fig. 7. Word error rates for the looped digits test and different CI and CD phoneme models, using MASPER and modified MASPER training methods.

Tests and models	SVIP AP		SVIP DIG		SVWL	
	Orig.	Mod.	Orig.	Mod.	Orig.	Mod
mono_1_7	10.03	10.21	4.13	5.05	12.96	12.81
mono_2_2	10.38	10.81	4.13	5.05	11.67	11.58
mono_4_2	9.69	9.17	5.05	5.05	8.73	8.64
mono_8_2	7.44	7.18	2.75	2.75	6.98	6.88
mono_16_2	4.93	5.02	1.83	1.83	5.71	5.83
mono_32_2	4.24	4.15	2.29	2.29	5.13	4.94
tri_1_2	1.99	1.99	1.38	1.38	4.86	4.76
tied_1_2	1.82	1.82	1.38	1.38	4.95	4.78
tied_2_2	1.64	1.56	1.38	1.38	4.21	4.04
tied_4_2	1.47	1.3	0.92	0.92	3.45	3.26
tied_8_2	1.56	1.21	1.38	1.38	2.96	3.03
tied_16_2	1.56	1.47	0.92	0.92	2.59	2.47
tied_32_2	1.56	1.3	0.92	0.92	2.16	1.89

Table 5. Word error rates for different CI and CD phoneme models using MASPER and modified MASPER and 3 tests: application words, single digits and looped digits.

Furthermore, in table 5 there are listed results for all tests and models. As it can be seen this modification brought on average improved results both for CI as well as CD phonemes; for tied CD models almost a 5% improvement was achieved. The main drawback of this approach is however, the longer training process which added 11% of extra computation.

6. Training of HMMs using SPHINX system

The HMM training tool whose output can be directly used with the SPHNIX decoders is called SphinxTrain. Thus let us in the following list some basic information about it.

6.1 SphinxTrain

It is an independent application which trains HMM models containing SphinxTrain tools and control scripts which actually govern the training (Scriptman, 2000).

The training process gradually undergoes following stages: verification of the input data, feature extraction (MFCC and PLP static coefficients plus standard auxiliary features are supported), vector quantization (used only for discrete models), initialization of HMM models (flat start), training of CI HMM models (no time alignment is supported, thus only the embedded training is available) with gradually incremented mixtures, automatic generations of language questions and building decision trees that are needed in the tying process for CD phonemes, pruning of the decision trees and tying similar states, training of tied CD phoneme models with gradually augmented number of mixtures.

To sum it up, as it can be seen, besides others there are several differences to the MASPER scheme: there are only two stages of the training, no single model training in the second stage, no alignment for multiple pronunciations, no sp model and no modification to the SIL structure (backward connection), number of training cycles is not fixed but is controlled by the convergence factor, non existence of predefined classes of phonemes, all models have the same structure, etc.

6.2 Proposed modifications to SphinxTrain procedure

Modifications that were done and tested can be divided into two categories: formal which involved conversions of data types and structures and the functional which affected the training process or the structure of HMM models. Functional changes and settings include following issues: selection of modelled non-speech events (so called fillers), set proper recordings for initialization and training phases, number of states per model, number of Gaussians mixtures, number of tied states for CD models, how to handle the problem of missing short pause model especially for the spelled items, etc. Thus in the following those issues will be addressed and tested on MOBILDAT- SK.

Unlike HTK a filer dictionary has to be constructed containing all non-speech events that should be modelled. This part of the design process is important as the non-speech models would share the same structure as the speech units (limitation of SphinxTrain). As a consequence, these models won't be so flexible (no backward connections etc.) thus they should rather be specialized to particular events. However, only few of the events were marked in the MOBILDAT database. Therefore we decided to use a general background model that includes all silences either within sentences or at the beginning and end of each recording. Two other models were added to the filer dictionary, one marking the noises produced by the speaker (spk) and a hesitation model (fil), as they were separately marked in the database.

Regarding the problem of usable recordings for the training process we decided to remove all items that contained damaged speech as no appropriate BH (garbled speech) model could be constructed following the tests with the modified MASPER procedure.

As most of the systems use multiple Gaussian mixtures ranging from 1 to 32 we decided to construct all of them for CI and tied CD phonemes and test their performance in all other experiments.

In order to find "optimal" setting for our system we performed tests regarding the number of states per model (either 3 or 5), number of tied CD phonemes that ranged from 2000 to

18000 and different training scenarios for spelled items to eliminate the missing short pause model. There were altogether 4 training scenarios for the spelled items. The original one ignored the problem and did not consider any background model between phonemes, even despite that there is a high probability of silence when the phonemes are spelled. The second one removed these recordings, just to avoid any incorrect transcription to be involved in the training. The 3rd scenario blindly assumed that there must be high a priory probability of pauses and thus inserted the silence model between all spelled items, and the last scenario uses the forced alignment tool from the SPHINX decoder (this was not included in the earlier versions of SphinxTrain scripts). This alignment does not produce any time marks, does not perform selection between multiple realizations of words, it just decides to place models from the filer dictionary between words. We applied this tool to all recordings (unmarked silences may occur also in other recordings) in the early stage of the training using CI models with 16 Gaussian mixtures. Tests were performed on the SPHINX 4 decoder (Walker et al., 2004) as it supports finite state grammar (JFSG) and the evaluation was done on the HTK system so that very similar conditions were maintained.

3 states	CD models, number of Gaussian mixtures			Average accuracy over different CD models for fix number of tied states
Number of tied states	8	16	32	
2000	97.67	97.87	98.14	97.89
5000	97.58	98.05	98.19	97.94
9000	97.67	97.88	97.96	97.83
12000	97.45	97.79	97.91	97.72
15000	97.5	97.88	98.16	97.84
180000	97.51	97.8	98.11	97.80

Table 6. The accuracy of CD HMM models with 3 states per a model and various numbers of mixtures for different number of tied states.

In table 6, results for 3 state models and different number of tied states for CD models are listed, the same results but for 5 state models are shown in table 7. As it can be seen the best results on average were obtained for 5000 tied states in the case of 3 state models, however, the differences in the widely tested ranges were interestingly small. For the test with 5 state models the best number was 18000 which is relatively high, but again the mutual differences were negligible. For 5 state models there were 5/3 times more different states thus it is natural these should be modelled with higher number of physical states. Comparing tables 6 and 7 it can be seen that on average the models with higher number of states (5) provided slightly better results, the average relative improvement is 0.21%. This is of no wander as they may have better modelling capabilities. On the other hand, there are more free parameters to be estimated which may produce more accurate but not robust enough models which in this test was apparently not the case. Finally, the 4 training scenarios were compared and the results are listed in table 8.

5 states	Number of Gaussian mixtures for CD models			Average accuracy over different CD models for fix number of tied states
Number of tied states	8	16	32	
2000	97.8	98.13	98.33	98.08
5000	97.85	98.28	98.11	98.08
9000	97.98	97.85	98.07	97.96
12000	97.84	97.98	98.07	97.96
15000	97.93	98.12	98.21	98.08
180000	97.97	98.17	98.3	98.14

Table 7. The accuracy of CD HMM models with 5 states per a model and various numbers of mixtures for different number of tied states.

Models	Scenarios			
	Original	Spelled items removed	SIL inserted into spelled items	Forced alignment
CI -4 Gaussians	94.49	94.85	95.02	95.26
CI -8 Gaussians	95.19	95.23	95.49	95.58
CI -16 Gaussians	95.96	96.08	96.34	95.96
CI -32 Gaussians	96.24	96.48	96.57	96.43
CD -4 Gaussians	97.31	97.62	97.63	97.46
CD -8 Gaussians	97.67	97.63	97.69	97.7
CD-16 Gaussians	97.88	98.15	97.82	97.7
CD-32 Gaussians	97.96	98.25	98.12	98.25
Average over models	96.58	96.78	96.83	96.82

Table 8. The accuracy of different tied CD and CI HMM models for 4 training scenarios.

As it can be seen the worst case is the original training scheme. On the other hand, the best results on average are provided by the "blind" insertion of SIL models between spelled phonemes. This suggests that there was really high incidence of pauses and the forced alignment was not 100% successful in detecting them.

7. Conclusion

Even though there are many new and improved techniques for HMM modelling of speech units and different feature extraction methods, still they are usually restricted to laboratories or specific conditions. Thus most of the systems designed for large vocabulary and speaker independent tasks use the "classical" HMM modelling by CDHMM with multiple Gaussian mixtures and tied CD models of phonemes.

In this chapter the construction of robust and accurate HMM models for Slovak was presented using 2 of the most popular systems and the training schemes. These were tested on the professional MOBILDAT –SK database that poses more adverse environment. In practical examples issues like: feature extraction methods, structures of models, modelled units, overtraining, and the number of tied states were discussed and tested. Some of the here suggested adjustments were successfully used while building Slovak ASR (Juhar, et al., 2006). Then the advanced training scheme for building mono, cross and multilingual ASR systems (MASPER based on HTK) that incorporates all the relevant training aspects was presented. Next, its practical modification aiming to increase the amount of usable training data by the BH model was suggested and successfully tested. Further, the training method utilizing the SPHINX system (SphinxTrain) was discussed and in the real conditions its "optimal" settings were found for the case of MOBILDAT –SK database. Finally, useful modifications for eliminating the problem of the missing short pause model in the case of spelled items were suggested and successfully tested. To compare both systems the best settings (modifications) for MASPER and SphinxTrain were used. Averaged word error rates were calculated over all models using application words and looped digits tests. Achieved results in table 9 are also listed separately for CI and tied CD models with 4, 8, 16, and 32 mixtures, the memory consumption and the training times are also included.

	Average WER		Memory consumption [MB]		Training time [hours]	
	Mod. Masper	Mod. SphinxTrain	Mod. Masper	Mod. SphinxTrain	Mod. Masper	Mod. SphinxTrain
All models	4.28	6.92	95	177	25h 8min	20h 58min
CD models	2.38	3.66	91.7	174	8h 48min	8h 53min
CI models	6.18	10.17	3.29	3.14	16h 20min	12h 5min

Table 9. Overall comparison of modified MASPER and modified SphinxTrain training procedures in terms of the accuracy, memory consumption and the training times. Word error rates were calculated for models with 4, 8, 16 and 32 mixtures.

As it can be seen, SphinxTrain scored on average worse by 38% in terms of WER evaluated over all models and executed tests. Its models occupy 86% more memory and are stored in 4 different files; however the training time for SphinxTrain is 20% shorter. If the averaged results are looked at separately, i.e. looped digits and application words tests, more precise image is obtained, see table 10. Models trained the by SphinxTrain procedure showed on average better results for the looped digits test than those on MASPER, on the other hand, SphinxTrain models were much less successful in the task of application words (perplexity 30), which contain richer set of CI and CD phonemes. That may suggest the tying and the training processes were not so effective. In the case of the MASPER procedure CI models were taken from the second run of the training so they were initialized and trained (only 7 initial cycles) on the time aligned recordings from the first run and thus they converged faster at the beginning. This fact is also used in the tying process of CD phonemes where models with only 1 mixture are taken in account. Finally, it should be noted that different decoders had to be used (HVite and SPHINX 4) during the evaluation. Despite the fact the same grammars and test sets were used, these decoders still have their specific settings

which may not have been optimized for particular tests, e. g. the insertion probability of fillers (SPHINX 4), pruning options, etc. Thus the results except the training phase partially reflect the decoding process as well, which was not the primary aim.

	looped digits		aplication words	
	Mod. Masper	Mod. SphinxTrain	Mod. Masper	Mod. SphinxTrain
All models	3.93	3.17	4.63	10.66
CD models	2.61	2.15	2.15	5.17
CI models	5.26	4.19	7.10	16.16

Table 10. Comparison of modified MASPER and modified SphinxTrain training procedures in terms of the accuracy, evaluated separately for looped digits and application words tests. Word error rates were calculated for models with 4, 8, 16 and 32 mixtures.

8. References

Baum, L. & Eagon, J. (1967). An inequality with applications to statistical estimation for probabilities functions of a Markov process and to models for ecology. *Bull AMS*, Vol. 73, pp. 360-363

Bonafonte, A.; Vidal, J. & Nogueiras, A. (1996). Duration modeling with expanded HMM applied to speech recognition, *Proceedings of ICSLP 96,* Vol. 2, pp. 1097-1100, ISBN: 0-7803-3555-4. Philadelphia, USA, October, 1996

Casar, M. & Fonllosa, J. (2007). Double layer architectures for automatic speech recognition using HMM, in book *Robust Speech recognition and understanding*, I-Tech education and publishing, ISBN 978-3-902613-08-0, Croatia, Jun, 2007

Darjaa, S.; Rusko, M. & Trnka, M. (2006). MobilDat-SK - a Mobile Telephone Extension to the SpeechDat-E SK Telephone Speech Database in Slovak, *Proceedings of the 11-th International Conference Speech and Computer (SPECOM'2006)*, pp. 449-454, St. Petersburg 2006, Russia

Hermasky, H. & Morgan, N. (1994). RASTA Processing of Speech, *IEEE Transactions on Speech and Audio Processing*, Vol. 2, No. 4, Oct. 1994

Hönig, F; Stemmer, G.; Hacker, Ch. & Brugnara, F. (2005). Revising Perceptual linear Prediction (PLP), *Proceedings of INTERSPEECH 2005,* pp. 2997-3000, Lisbon, Portugal, Sept., 2005

Huang, X.; Ariki, Y. & Jack, M. (1990). *Hidden Markov Models for Speech Recognition*, Edinburg university press, 1990

Jiang, H. & Li X. (2007) A general approximation-optimization approach to large margin estimation of HMMs, in book *Robust Speech recognition and understanding*, I-Tech education and publishing, ISBN 978-3-902613-08-0, Croatia, Jun, 2007

Juhar, J.; Ondas, S.; Cizmar, A; Rusko, M.; Rozinaj, G. & Jarina, R. (2006). Galaxy/VoiceXML Based Spoken Slovak Dialogue System to Access the Internet. *Proceedings of ECAI 2006 Workshop on Language-Enabled Educational Technology and Development and Evaluation of Robust Spoken Dialogue Systems*, pp.34-37, Riva del Garda, Italy, August, 2006

Kosaka, T.; Katoh, M & Kohda, M. (2007). Discrete-mixture HMMs- based approach for noisy speech recognition, in book *Robust Speech recognition and understanding*, I-Tech education and publishing, ISBN 978-3-902613-08-0, Croatia, Jun, 2007

Lee, K.; Hon, H. & Reddy, R. (1990). An overview of the SPHINX speech recognition system, *IEEE transactions on acoustics speech and signal processing*, Vol. 38, No. 1, Jan., 1990

Levinson, E. (1986). Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech and Language*, Vol. 1, pp. 29-45, March, 1986

Lindberg, B.; Johansen, F.; Warakagoda, N.; Lehtinen, G; Kacic, Z; Zgang, A; Elenius, K. & Salvi G. (2000). A Noise Robust Multilingual Reference Recognizer Based on SpeechDat(II), *Proceedings of ICSLP 2000*, Beijing, China, October 2000

Nadeu, C. & Macho, D. (2001). Time and Frequency Filtering of Filter-Bank energies for robust HMM speech recognition, *Speech Communication*. Vol. 34, Elsevier, 2001

Nouza, J.; Zdansky, J.; David, P.; Cerva, P.; Kolorenc, J. & Nejedlova, D. (2005). Fully Automated System for Czech Spoken Broadcast Transcription with Very Large (300K+) Lexicon. *Proceedings of Interspeech 2005*, pp. 1681-1684, ISSN 1018-4074, Lisboa, Portugal, September, 2005,

Poggio, T. & Girosi, F. (1990). Networks for approximation and learning, *Proceedings of the IEEE* 78, pp. 1481-1497

Rabiner, L. & Juan, B. (1993). *Fundamentals of speech recognition*, ISBN 0-13-015157-2, Prentice Hall PTR, New Yersey.

Scriptman (2000). Online documentation of the SphinxTrain training scripts, location: http://www.speech.cs.cmu.edu/sphinxman/scriptman1.html, last modification Nov. 2000

W. Walker, P. Lamere, P. Kwok (2004). Sphinx-4: A Flexible Open Source Framework for Speech Recognition, Report, location: http://research.sun.com/techrep /2004/smli_tr-2004-139.pdf

Young, S.; Evermann, G.; Hain, T.; Kershaw, D.; Moore, G.; Odell, J.; Ollason, D.; Povey, Dam.; Valtchev, V. & Woodland, P. (2002). The HTK Book V.3.2.1, Cambridge University Engineering Department, Dec. 2002

Zgank, A.; Kacic, Z.; Diehel, F.; Vicsi, K.; Szaszak, G.; Juhar, J.; Lihan, S. (2004). The Cost 278 MASPER initiative- Crosslingual Speech Recognition with Large Telephone Databases, *Proceedings of Language Resources and Evaluation (LREC)*, pp. 2107-2110, Lisbon, 2004

Speech Recognition

Edited by France Mihelic and Janez Zibert

ISBN 978-953-7619-29-9

Hard cover, 550 pages

Publisher InTech

Published online 01, November, 2008

Published in print edition November, 2008

Chapters in the first part of the book cover all the essential speech processing techniques for building robust, automatic speech recognition systems: the representation for speech signals and the methods for speech-features extraction, acoustic and language modeling, efficient algorithms for searching the hypothesis space, and multimodal approaches to speech recognition. The last part of the book is devoted to other speech processing applications that can use the information from automatic speech recognition for speaker identification and tracking, for prosody modeling in emotion-detection systems and in other speech processing applications that are able to operate in real-world environments, like mobile communication services and smart homes.

INTECH
open science | open minds