Statistical Modeling for Speech Recognition

Article *in* World Applied Sciences Journal · January 2013

DOI: 10.5829/idosi.wasj.2013.21.mae.99935

Othman Omran Khalifa
International Islamic University Malaysia
480 PUBLICATIONS **2,929** CITATIONS
SEE PROFILE

Jamal Daoud
International Islamic University Malaysia
77 PUBLICATIONS **435** CITATIONS
SEE PROFILE

World Applied Sciences Journal 20 (Mathematical Applications in Engineering): 115-122, 2012
ISSN 1818-4952
© IDOSI Publications, 2012
DOI: 10.5829/idosi.wasj.2012.20.mae.99935

Statistical Modeling for Speech Recognition

Othman O. Khalifa, Khalid Khalil El-Darymli, Aisha-Hassan Abdullah and Jamal I. Daoud

Department of Electrical and Computer Engineering, International Islamic University Malaysia

Abstract: The demand of intelligent machines that may recognize the spoken speech and respond in a natural voice has been driving speech research. The challenging in speech recognition systems due to the language nature where there are no clear boundaries between words, the phonetic beginning and ending are influenced by neighbouring words, in addition to the variability in different speakers speech: male or female, young or senior, loud or low speech, read or spontaneous, emotional or formal, fast or slow speaking rate and the speech signal can be affected with environment noise. To avoid these difficulties the data driven statistical approach based on large quantities of spoken data is used. The performance of speech recognition systems is still far worse than that of humans. This is partly caused by the use of poor statistical models. In this paper, a comprehensive study of statistical methods for speech and language processing are presented. The role of signal processing in creating a reliable feature set for the recognizer and the role of statistical methods in enabling the recognizer to recognize the words of the spoken input sentence as well as the meaning associated with the recognized word sequence were presented.

Key words: Speech recognition % Speech coding % Statistical language modeling

INTRODUCTION

Speech recognition means a process that inputs human speech and tries to find out what was said. Speech is the primary means of communication between people and speech recognition is the conversion of speech into text. This is done by identifying characteristic acoustic features of human voice which is unique. This makes it possible to use the speaker`s voice to verify their identity and can be used to control access to services such as voice dialing, banking by telephone, database access service, security control for confidential information area, stock price quotations weather reports, Data entry, voice dictation, access to information: travel, banking, Commands, Automobile portal, speech transcription, Handicapped people, supermarket, railway reservations and remote access to computers etc [1]. Basically prior to a verification or identification session, users that are authorized to use the system will need to undergo enrollment session, where they provide speech samples that will be processed and stored in the database in order to be used later during the identification process. Here only certain important features that are unique to individuals are extracted while other redundancies are discarded. The basic concept of Speech Recognition (SR) systems is shown in Figure 1.

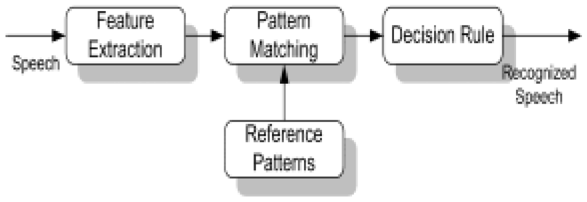


Fig. 1: Speech Recognition System Concept

Types of Speech Recognition: Speech recognition systems can be separated in several different classes by describing what types of utterances they have the ability to recognize [2, 3, 4]. These classes are classified as the following:

Isolated Words: Isolated word recognizers usually require each utterance to have quiet (lack of an audio signal) on both sides of the sample window. It accepts single words

Corresponding Author: Othman O. Khalifa, Department of Electrical and Computer Engineering, Faculty of Engineering, International Islamic University Malaysia, Jalan Gombak, Selangor, Malaysia.
E-mail: khalifa@iium.edu.my

or single utterance at a time. These systems have "Listen/Not-Listen" states, where they require the speaker to wait between utterances (usually doing processing during the pauses). Isolated Utterance might be a better name for this class.

Connected Words: Connected word systems (or more correctly 'connected utterances') are similar to isolated words, but allows separate utterances to be 'run-together' with a minimal pause between them.

Continuous Speech: Continuous speech recognizers allow users to speak almost naturally, while the computer determines the content. Basically, it's computer dictation). Recognizers with continuous speech capabilities are some of the most difficult to create because they utilize special methods to determine utterance boundaries.

Spontaneous Speech: At a basic level, it can be thought of as speech that is natural sounding and not rehearsed. An ASR system with spontaneous speech ability should be able to handle a variety of natural speech features such as words being run together.

Modules of Speech Recognition: Research in speech processing was motivated by people's desire to build mechanical models to emulate human verbal communication capabilities and to automate simple tasks which necessitates human machine interactions. Speech recognition technology has made it possible for computer to follow human voice commands and understand human languages. Based on major advances in statistical modeling of speech, automatic speech recognition systems today find widespread application in tasks that require human machine interface. Although many technological progresses have been made, still there are issues that need to be tackled [5, 8]. Figure 1 shows a schematic representation of speech recognition system. Figure 2 illustrates the major modules of an SR system. In feature extraction, signal processing techniques are applied to the speech signal in order to dig out the features that distinguish di?erent phonemes from each other. Given the features extracted from the speech, acoustic modeling provides probabilities for difierent phonemes at difierent time instants. Language modeling, on the other hand, defines what kind of phoneme and word sequences is possible in the target language or application at hand and what are their probabilities. The acoustic models and language models are used in

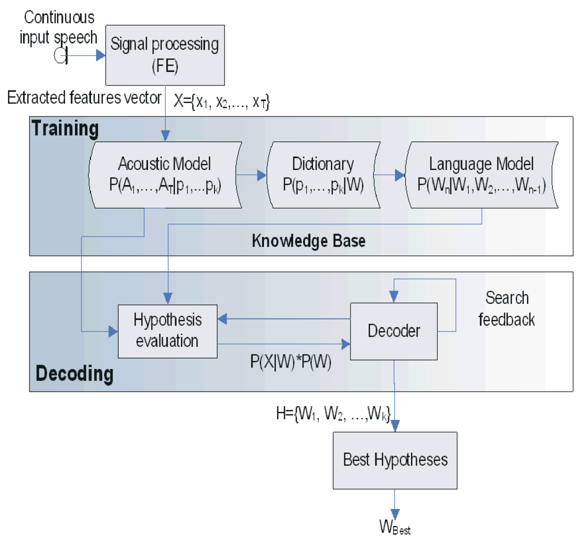


Fig. 2: The main components of an automatic speech recognition system

decoding for searching the recognition hypothesis that fits best to the models. Recognition output can then be used in various applications [4, 6, 7, 8].

The use of personal features, unique to all human being to identify or to verify a person`s identity is a field that is being actively researched. This can be seen in the usage of finger print, retina pattern and voice pattern in various applications to identify or to verify the identity of an individual. The speaker recognition system consists of the following steps: Speaker identification; in this system, when a user inputs a test utterance, the system will identify which of the speaker made the utterance according to the speech patterns stored in the database. Therefore here the output will be either the name of the user or the system will reject if the users speech pattern is not in the database.

Speech Modeling: Classical Approaches: Upon speaking in the microphone the speech pressure wave will be captured, converted to an electrical signal, sampled and quantized into 16 bits (N=16). This process is done inside the audio card. This digitized signal will be fed to a Front-End (FE) signal processing as depicted in Figure 3 [9, 10, 11].

Pre-Emphasis: The characteristics of the vocal tract define the current uttered phoneme. Such characteristics are evidenced in the frequency domain by the location of the formants. Although possessing relevant information, high frequency formants have smaller amplitude with

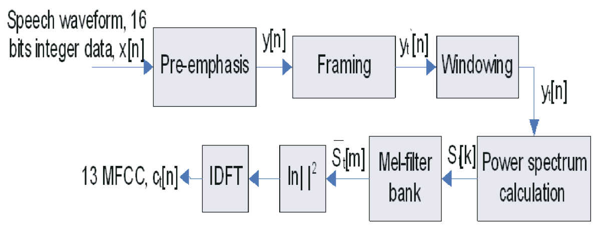


Fig. 3: Front-End (FE) signal processing.

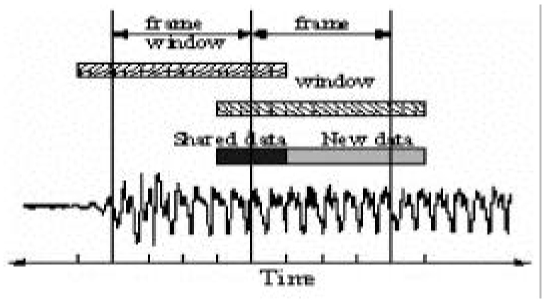


Fig. 4: Demonstration of framing and windowing

respect to low frequency formants. A pre-emphasis of high frequencies is therefore required to obtain similar amplitude for all formants. Such processing is usually obtained by filtering the speech signal with a first order Finite Impulse Response (FIR) filter whose transfer function in the z-domain is:

$$H[z] = 1 - ".zG^1\ 0 \# " \# 1 \qquad (1)$$

where " being the pre-emphasis parameter.

The following FIR pre-emphasis filter is applied to the input waveform *x*[n] [4]:

$$y[n] = x[n] - "x[n-1] \qquad (2)$$

Framing and Windowing: Each feature used by the recognizer must be calculated a user specified number of times per second (frame duration) and computed over some time interval (window duration). Typical frame duration in speech recognition is 10 ms, while typical window duration is 25 ms. This process is demonstrated in Fig. 4 below. Windowing is necessary since the voice signal is by nature a nonstationary signal whilst the power spectrum evaluation is reliable in the case of stationary signal [4].

$$\grave{y}_t[n] \equiv y[n - t.Q], \quad 0 \le n \le N, 1 \le t \le T \qquad (3)$$

$$y_t[n] \equiv w[n].\grave{y}_t[n] \qquad (4)$$

where: *Q* is a temporal length shift of each frame. And as it was shown earlier, normally N=16 bits. *w[n]* is the impulse response of the window. For SR applications, the most used window is the Hamming window which has the following impulse response:

$$w[n] = \begin{cases} 0.54 - 0.46\cos\left(\dfrac{2p\,n}{N-1}\right) & n = 0,\dots,N-1 \\ 0 & otherwise \end{cases} \qquad (5)$$

Power Spectrum: The Short Fourier Transform (SFT) for *y[n]* could be calculated using this formula:

$$Y_t\,[e^{jw}] = \sum_{n=-\infty}^{+\infty} y[n - t.Q].w[n]e^{-jwn} \qquad (6)$$

Computational complexity is greatly reduced if $Y_t[ej^T]$ is evaluated only for a discrete number of **9** values. If such values are equally spaced, for instance considering **9**=2**9***k/N*, then the Discrete Fourier Transform (DFT) of all frames of the signal is obtained:

$$Y_t[k] = Y_t[e^{j2pk/N}], k = 0,\dots,N-1 \qquad (7)$$

The phase information of the DFT samples of each frame is discarded since perceptually it was proven that phase does not carry useful information. Accordingly, the output of this stage is [20, 16, 4]:

$$S_t[k] = (real\,|Y_t[k]|)^2 + (imag(|Y_t[k]|)^2 \qquad (8)$$

The Mel Filterbank: Spectral analysis reveals those speech signal features which are mainly due to the shape of the vocal tract. Spectral features of speech are generally obtained as the exit of filter banks, which properly integrate a spectrum at defined frequency ranges. The *mel-scale* filterbank is the most used in SR. It is a series of *M* triangular band-pass filters that have been designed to simulate the band-pass filtering believed to occur in the auditory system. Refer to Fig. 5.

The transfer function of the triangular mel-weighting filters $H_m[k]$ is given by:

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])} & f[m-1] \le k \le f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m+1] - f[m])} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases} \qquad (9)$$

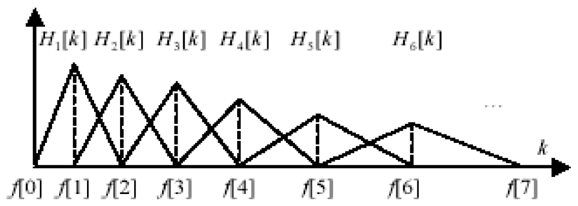

$H_1[k]$ $H_2[k]$ $H_3[k]$ $H_4[k]$ $H_5[k]$ $H_6[k]$
k
f[0] f[1] f[2] f[3] f[4] f[5] f[6] f[7]

Fig. 5: The mel-scale filterbank [8].

The mel-spectrum of the power spectrum is computed by multiplying the power spectrum by each of the triangular mel-weighting filters and integrating the result.

$$\bar{S}_t[m] = \sum_{k=0}^{N-1} S_t[k] H_m[k] \quad m = 0,1,...,M-1 \tag{10}$$

where, *k* is the DFT domain index, *N* is the length of the DFT and *M* is total number of triangular mel-weighting filters [8].

Log Energy Computation: The previous procedure has the role of smoothing the spectrum, performing a processing that is similar to that executed by the human ear. The next step consists of computing the logarithm of the magnitude of the coefficients which is performed by the ear as well. Magnitude discards the useless phase information while a logarithm performs a dynamic compression, making feature extraction less sensitive to variations in dynamics [4].

Mel Fraudulency Cepstarl Coefficients (MFCC) Computation: The MFCC is a representation defined as the real cepstrum of a windowed short-time signal derived from the FFT of that signal. The final procedure for MFCC computation consists of performing the inverse DFT on the logarithm of the magnitude of the filterbank output:

$$c_t[n] = \sum_{m=0}^{M-1} \ln\{|\bar{S}_t[m]|\}.\cos\left(n(m-\frac{1}{2})\frac{p}{M}\right), \quad n = 0,...,M-1 \tag{11}$$

where *M* varies for different implementation from 24 to 40. Typically for speech recognition only the first 13 coefficients are used [8].

Delta and Double Delta Coefficients: First and Second order differences may be used to capture the dynamic evolution of the signal. The first order delta MFCC computed from:

$$\Delta c_t = c_{t+2} - c_{t-2} \tag{12}$$

The second order delta MFCC computed from:

$$\Delta^* c_t = \Delta c_{t+1} - \Delta c_{t-1} \tag{13}$$

The final output of the FE processing would comprise 39 features vector (observations vector X_t) per each processed frame. Comprised features include 13 MFCC features c_t[n], 13 Δc_t[n] and 13 $\Delta\Delta c_t$[n] yields a total features vector of length equal to 13*3=39 [9].

Training: Training is the process of learning the *acoustic model* (AM) and the *language model* (LM) to construct the knowledge base of the speech recognizer. The knowledge base includes: AM, Dictionary and LM.

Acoustic Model: The *Acoustic Model* provides a mapping between a unit of speech and an HMM that can be scored against incoming features provided by the Front-End. It contains a pool of a Hidden Markov Models (*HMM*). Provides a mapping between a single unit of speech (within some context) and an HMM. The unit context can include: the surrounding units and a position within a word. For large vocabularies each word is represented as a sequence of phonemes, accordingly there has to be an AM per each phoneme, moreover, it has to be depending on the context (e.g. coarticulation) and even the context dependence may cross word boundary. Practically, the primary technique is to model the speech as a Hidden Markov Process (HMM) where each phoneme is a sequence of acoustic segments and each acoustic segment is one state in this Hidden Markov process, it is called *hidden* because when we observe the acoustics we do not directly infer what state the system is in. We will have to infer that with probabilistic inference computation. Phones are then further refined into context-dependent triphones, i.e., phones occurring in given left and right phonetic contexts. This depicted in Figure 6 [3, 9].

On Figure 6 above, a_{ij} is the transition probability from state *i* to state *j*, $b_i(k)$ is the output probability that is the probability of emitting symbol o_k when state *i* is entered. Practically they are represented by transition probability matrix and an output probability matrix respectively. The final state depicted in Figure 6 has no a transition probability since it is just a buffer for the last three states [12]. In addition to that, a complete HMM

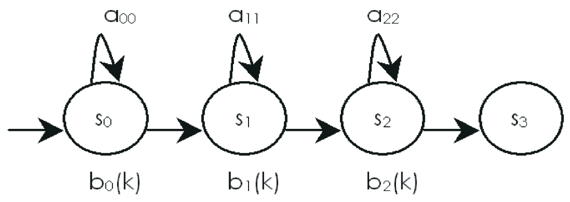


Fig. 6: 4-States per phoneme HMM model

model includes the initial state distribution **9**$_i$. All a_{ij}, $b_i(k)$ and **9**$_i$ matrices constitute the HMM model parameters **9**$=(A, B, B)$. For each acoustic segment, there is a probability distribution across acoustic observations $b_i(k)$. The leading technique is to represent the acoustic observations as a *mixture Gaussian distribution* or shortly *Gaussian Mixtures* (GM). If every state (senone) has it is own private GM then the AM is called continuous. If all the senones share a single codebook of Gaussian distributions, but each senone has its own set of mixture weights applied to that codebook, the AM is called semi-continuous or tied-mixture AM.

Continuous density acoustic models are computationally expensive to deal with, since they can contain hundreds of thousands of Gaussian densities that must be evaluated in each frame. To reduce this cost, one can use an approximate model that efficiently identifies the top scoring candidate densities in each Gaussian mixture in any given frame. The remaining densities can be ignored during that frame. Such approximate model is built by sub-vector quantizing the acoustic model densities [8, 18].

Dictionary: Dictionary is a file contains pronunciations for all the words of interest to the decoder. This file specifies word pronunciations. For large vocabulary speech recognizers pronunciations are specified as a linear sequence of phonemes [16].

Language Model (LM): For recognition of natural language it is a statistical LM where the speaker could be talking about any arbitrary topic. The main used model is the n-gram statistics and in particular *trigram* (n=3), $P(W_t|W_{t-1},W_{t-2})$ which finds the conditional probability for the current word given the two previous words. But this modeling create a problem which is if we have a vocabulary of 200,000 words it entails $200{,}000^3$ trigrams which is too large to have statistics to separately model each trigram. The solution to this problem is to model the most frequently occurring trigrams and to use some

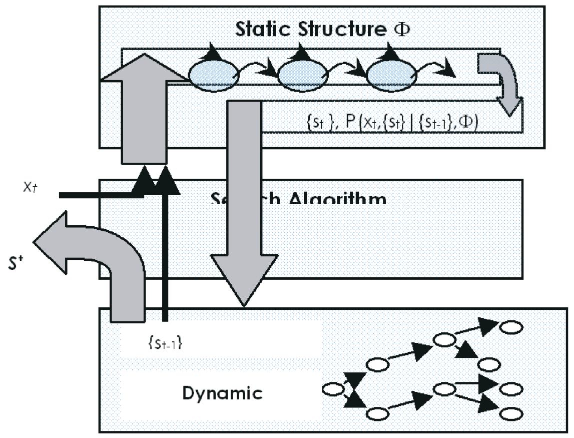


Fig. 7: Static and dynamic structures.

method of statistical smoothing and backing-off if we do not have enough instances of the actual trigram. To clarify, this means if we do not have enough data of the trigram we can back-off to the bigram which is modeling the current word giving only the previous word. Or we can even further back-off modeling the current word independent of the context which is called a unigram probability model [8, 19].

Recognition: The final goal in SR is recognition itself. Using the knowledge databases which include a well-trained AM, LM and the lexicon and given an input speech utterance the goal is to unveil the best hidden state sequence. By "*BEST*" we mean the sequence of states that maximizes the probability function.

Let $S=(s_1,s_2,\ldots,s_T)$ be the sequence of states that are recognized and x_t be the feature samples computed at time t, where the feature sequence from time 1 to t is indicated as: $X=(x_1,x_2,\ldots,x_t)$. Accordingly, the sequence of recognized states S^* could be obtained by:

$$S^*=ArgMax\ P(S,X|????? \qquad (14)$$

Note that **9** is the static structure which represents the HMM model parameters. The whole recognition issue it turned out to be a search issue.

Figure 7 depicts the search algorithm. Given x_t, the algorithm retrieves all the admissible states at time t-1: $\{s_{t-1}\}$. Then it finds all the possible successors $\{s_t\}$ of the states $\{s_{t-1}\}$ and computes the associated probability $P(x_t,\{s_t\}|\{s_{t-1}\},$**9**$)$ for any pair of admissible states $\{s_t\}$, $\{s_{t-1}\}$. Probabilities and successors $\{s_t\}$ are stored so that the probability of a sequence of admissible states (paths) can be retrieved. When all the observations are processed, the best path is the one with the highest probability [4]. The static structure contains 9and given $\{s_{t-1}\}$ and x_t returns the admissible successors $\{s_t\}$ and the associated probabilities. This structure representing an HMM, does not change configuration during recognition. The second structure at the bottom of Figure 7 stores the admissible states $\{s_t\}$. For each state s_t the predecessor state s_{t-1}, is memorized that has generated s_t and the probability to be in s_t given the observations and the previous states. This structure is essentially a tree graph grows at each time instant and therefore it is called dynamic. Data contained in the static structure are used to generate hypotheses on the possible word sequences.

Hypotheses are stored in the nodes of tree graph that implements the dynamic structure. The decoder finds the more likely sequence of words analyzing information contained in this tree and using the Veterbi Beam search [4].

The Veterbi Beam Search: We are looking for the state sequence $S=(s_1,s_2,\ldots,s_T)$ that maximizes P(S,X|9 To define the best-path probability:

$$V_t(i) = P(X_1^t, S_1^{t-1}, s_t = i \mid \Phi) \quad (15)$$

where: $V_t(i)$ is the probability of the most likely state sequence at time t, which has generated the observation X_1^t (until time t) and ends in state i. 9 = (A, B, B) is HMM model parameters. The Veterbi beam search algorithm is given below. N is the number of states in the model 9 and T is the number of sequence of observations. Note that b is called the beam search coefficient and I_b is the threshold for the cumulative score. Hypothesis tree which could be structured dynamically using the Veterbi beam search algorithm is depicted in Figure 8 [4, 6, 8].

Algorithm 1: The Veterbi beam search.

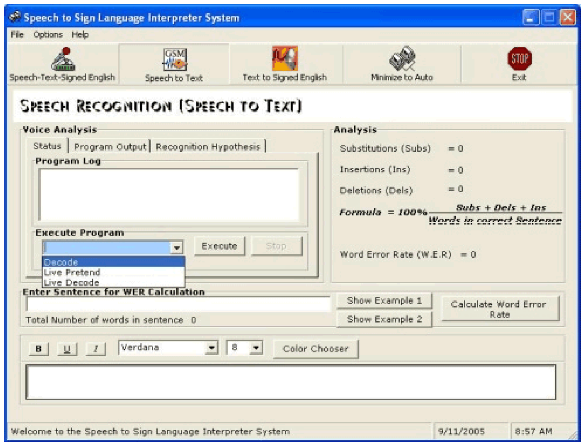
```
Initialization:
For 1 # i # N
$V_1(i) = B_i b_i (X_1)$
Goto XX

Recursive Steps:
For 2 # t # T        {
For 1 # k # N
$V_1(k) = V_{t-1}(j)\, a_{jk}\, b_k (X_t)$
Goto XX              }

Backtracking:
$s_t^* = ArgMax\ V_{t+1} (s^*_{t+1})\ t = T-1, T-2, ..., 1$
$S^* = (s^*_1, s_2^*, ..., s_T^*$ is the best sequence

XX:
For 1 # i # N
Find pt(st*)= Max[Vt(i)]
Calculate the threshold $J_b = p_t(s_t^*) b$
For 1 # j # N        {
If pt(st=j) $ $J_b$ Memorize both $V_t$(j) and path "j"
Else Discard Vt(j) }
Return
```

Other Models

Vector Quantization: Vector quantization (VQ) model also known as centroid model, is one of the simplest text-independent speaker models. It was introduced to speaker recognition in the 1980s and its roots are originally in data compression. Even though VQ is often used for computational speed-up techniques and lightweight practical implementations, it also provides competitive accuracy when combined with background model adaptation [10].

Gaussian Mixture Model: Gaussian mixture model (GMM) (Reynolds and Rose, 1999, Reynolds *et al*., 2000) is a stochastic model which has become the de facto reference method in speaker recognition. The GMM can be considered as an extension of the VQ model, in which the clusters are overlapping. That is, a feature vector is not assigned to the nearest cluster as in (4), but it has a nonzero probability of originating from each cluster. A GMM is composed of a finite mixture of multivariate Gaussian components. A GMM, denoted by k, is characterized by its probability density function:

Support Vector Machine: Support vector machine (SVM) is a powerful discriminative classifier that has been recently adopted in speaker recognition. It has been applied both with spectral, prosodic and high-level features. Currently SVM is one of the most robust classifiers in speaker verification and it has also been successfully combined with GMM to increase accuracy. One reason for the popularity of SVM is its good generalization performance to classify unseen data. The SVM, is a binary classifier which models the decision boundary between two classes as a separating hyperplane. In speaker verification, one class consists of the target speaker training vectors.



Substitutions (Subs) = 0

Insertions (Ins) = 0

Deletions (Dels) = 0

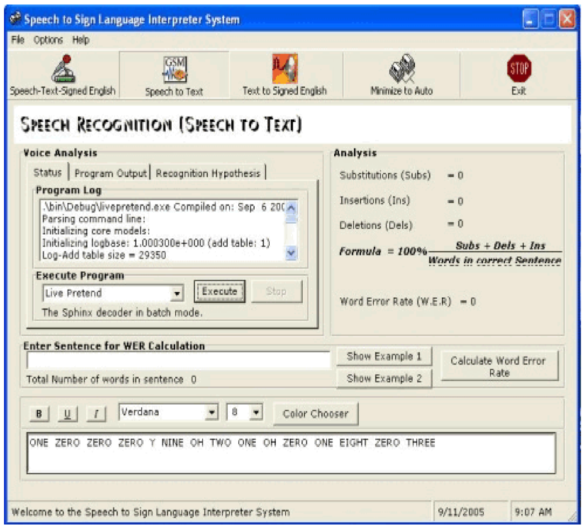
Formula = 100% $\frac{\text{Subs + Dels + Ins}}{\text{Words in correct Sentence}}$

Word Error Rate (W.E.R) = 0

Total Number of words in sentence 0

Welcome to the Speech to Sign Language Interpreter System 9/11/2005 8:57 AM

Fig. 8: Decode and Live Pretend) for batch mode) and (Live Decode) for live mode- to convert speech into text



```
.\bin\Debug\livepretend.exe Compiled on: Sep 6 200
Parsing command line:
Initializing core models:
Initializing logbase: 1.000300e+000 (add table: 1)
Log-Add table size = 29350
```

Live Pretend

The Sphinx decoder in batch mode.

Substitutions (Subs) = 0

Insertions (Ins) = 0

Deletions (Dels) = 0

Formula = 100% $\frac{\text{Subs + Dels + Ins}}{\text{Words in correct Sentence}}$

Word Error Rate (W.E.R) = 0

Total Number of words in sentence 0

ONE ZERO ZERO ZERO Y NINE OH TWO ONE OH ZERO ONE EIGHT ZERO THREE

Welcome to the Speech to Sign Language Interpreter System 9/11/2005 9:07 AM

Fig. 9: Demonstration of "Live Pretend" program. Live Pretend enables recognition of prerecorded speech

Result Analysis: In this paper, Speech to Text system is a large vocabulary, speaker independent and continuous speech recognizer. Figure 8, 9 show User can select one of the three included programs -(Decode and Live Pretend) for batch mode) and (Live Decode) for live mode- to convert speech into text.

CONCLUSION

In this paper, the overall process performed by Sphinx 3.5 could be shortly described as follows: upon speaking in the microphone the speech signal will be captured, digitized and applied to FE signal processor wherein a certain number of MFCC features will be extracted per each input frame. The static structure already residing as a knowledge base comprises AM, dictionary and LM. Using the observations (extracted features) a search graph will be constructed dynamically and the final recognition output will be obtained through the Veterbi beam search.

REFERENCES

1. Rabiner, L. and B.H. Juang, 1993, Fundamentals of Speech Recognition, Prentice Hall.
2. Gales, J.F. and S.J. Young, 2008. The application of hidden Markov models in speech recognition, Foundation and Trends in Signal Processing, 1(3): 195-304.
3. Becchetti, C. and L.R. Ricotti, 1999. Speech Recognition Theory and C++ Implementation. England: Wiley.
4. Gouvêa, E., 0000. The CMU Sphinx Group Open Source Speech Recognition Engines. http:// www.speech.cs.cmu.edu/sphinx/.
5. Huang, X., A. Acero, HW. Hon and R. Reddy, 2001. Spoken Language processing, a Guide to Theory, Algorithm and System Development. Prentice Hall PTR.
6. Arisoy, E., M. Sarac¸lar, B. Roark and I. Shafran, 2010. Syntactic and sub-lexical features for Turkish discriminative language models, in Proc. ICASSP'10, pp: 5538-5541.
7. Hwang, Mei-Yuh, 1993. Subphonetic Acoustic Modeling for Speaker Independent Continuous Speech Recognition. Ph.D. thesis, Computer Science Department, Carnegie Mellon University. Tech Report No. CMU-CS-93-230.
8. Jelinek, F., 1976. Continuous Speech Recognition by Statistical Methods. Proceedings of the IEEE, 64(4): 532-556.
9. Lin, E., 2003. A First Generation Hardware Reference Model for a Speech Recognition Engine. Master Thesis, Computer Science Department, Carnegie Mellon University.
10. Cemal Hanilci and Figen Ertas, 2011. VQ-Ubm Based Speaker Verification Through Dimension ReductionUsing Local PCA, 19th European Signal Processing Conference (EUSIPCO 2011) Barcelona, Spain, August 29 - September 2, 2011.

11. Ravishankar, M., 1996. 0000. Efficient Algorithms for Speech Recognition. Ph.D. dissertation, Carnegie Mellon University. Tech Report. No. CMU-CS-96 143.
12. Watanabe, S., T. Hori and A. Nakamura, 2010. Large vocabulary continuous speech recognition using WFST-based linear classifier for structured data, in Proc. Interspeech'10, pp: 346-349.
13. Ravishankar, M.K., 2004. Sphinx-3 s3.X Decoder (X=5). Sphinx Speech Group. School of Computer Science, CMU. http:// cmusphinx.sourceforge.net/ sphinx3/.