DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

The 1996 Mississippi State University Conference on

Speech Recognition

What: EE 8993 - 02 Project Presentations
Where: 432 Simrall, Mississippi State University
When: May 1, 1996 — 1:00 to 4:00 PM

SUMMARY

The Department of Electrical and Computer Engineering invites you to attend a mini-conference on Speech Recognition, being given by students in EE 8993 — Fundamentals of Speech Recognition. Papers will be presented on a wide range of topics including signal processing, Hidden Markov Models, search, and language modeling.

Students will present their semester-long projects at this conference. Each student will give a 12 minute presentation, followed by 3 minutes of discussion. After the talks, each student will be available for a live-input real-time demonstration of their project. These projects account for 100% of their course grade, so critical evaluations of the projects are welcome.

Session Overview

1:00 PM — 1:10 PM:	J. Picone, Introduction
1:15 PM — 1:30 PM:	**J. Trimble,** “Front-end of a Speech Recognizer”
1:30 PM — 1:45 PM:	**L. Webster**, “Front End Modeling with Special Emphasis on FFTs, LPC, and Feature Selection”
1:45 PM — 2:00 PM:	**R. Seelam**, “Implementation of Statistical Modeling Techniques and Channel Adaptation Techniques”
2:00 PM — 2:15 PM:	**A. Ganapathiraju**, “Implementation of Viterbi Beam Search Algorithm”
2:15 PM — 2:30 PM:	**N. Deshmukh**, “Efficient Search Algorithms for Large Vocabulary Continuous Speech Recognition”
2:30 PM — 2:45 PM:	**O. LaGarde**, “Language Modeling and Grammar Construction for an HMM Continuous Speech Recognition System”
2:45 PM — 3:00 PM:	**S. Given**, “Development of an N-Gram Based Language Model for Continuous Speech”
3:00 PM — 4:00 PM:	Demonstrations in 414 Simrall

AUTHOR INDEX

Volume I

Speech Recognition

Table of Contents

Front-end of a Speech Recognizer

by

J. Trimble
trimble@isip.msstate.edu

Department of Electrical and Computer Engineering
Mississippi State University

ABSTRACT

This proposal describes a plan to design and implement the front-end of a speech recognition sytem. The front end must derive a smooth spectral estimate of a signal in order to produce feature vectors that are compatible with the acoustic models of the system. Linear prediction provides an efficient and simple means of computing these feature vectors. Its basic purpose is to as accurately predict currents values of a signal based on a weighted sum of the signal's previous values. In addition, an even better spectral estimator, cepstral analysis, will be implemented.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

This presentation is temporarily missing. It will be restored shortly.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

THE POWER OF THREE: FFTs, LP TRANSFORMATIONS, AND FEATURE SELECTION

by

Leigh Anne Webster webster@isip.msstate.edu

The Speech Processing Group
Mississippi State University

ABSTRACT

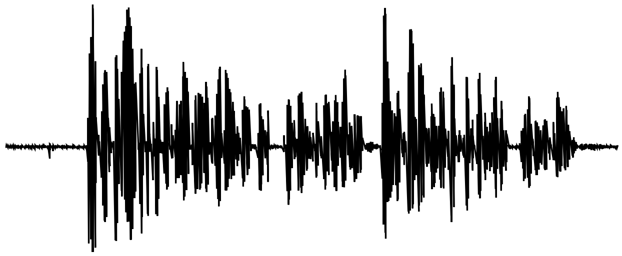
Communication is a key factor in life. In order to be a productive communicator, speech must be generated and comprehended, by fully understanding the speech signal. Information theory states that speech can be represented in terms of its message content. Another way of describing speech is in terms of an acoustic waveform, the signal relaying the message content. A typical speech recognizer is composed of three main elements: the front-end or acoustic model, search, and language modeling. In this project, the first element of a speech recognizer, the front-end model will be discussed with special emphasis given to fast Fourier transform (FFT) based measurements, linear prediction coefficient (LPC) transformations, and feature selection. The over-all purpose of this project is to design and implement the specific aspects of the acoustic model mentioned above with the final goal of incorporating them into a speech recognition system.

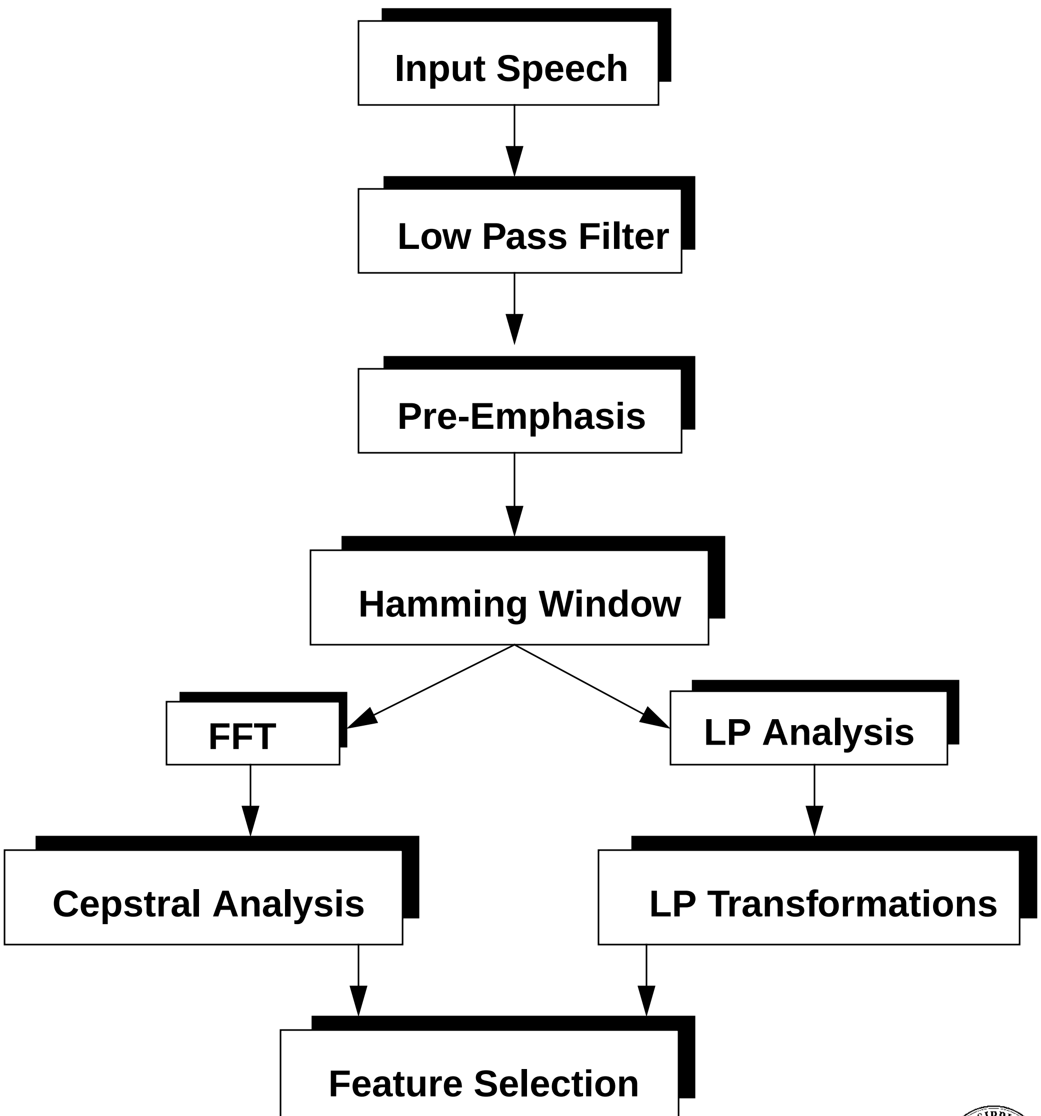
Accurate Modeling of the Front End
Why we strive for sound models...

- Applications of Speech Recognition
  - Telecommunications assistants
    - voice dialing
    - automated operator services
  - Computer applications
    - navigate your pc with your voice
    - aid the physically challenged
    - security (passwords)
  - Banking/Shopping
    - ATMs
    - pay bills
    - make purchases

- Obstacles
  - Cost
  - Computational considerations
  - Training sets

The Speech Waveform





Parameter Selection:

Sample Frequency = 12000 Hz

Pre-Emphasis Factor = 0.95

Frame Duration = 20 msec

Window Duration = 30 msec

FFT Analysis

- Advantages
  - easy way to compute a filter bank model
  - very efficient computationally
- Disadvantages
  - non-linear frequency mappings have to be adjusted to match the FFT fs/N frequencies
- The DFT

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi k \frac{n}{N}} \qquad k = 0, 1, \ldots, N-1$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi k \frac{n}{N}} \qquad n = 0, 1, \ldots, N-1$$

- make the DFT a power of 2 and use the FFT, zero pad if needed
- the radix-2 algorithm (see ~webster/ee_8993/project/fft.cc)

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Cepstral Analysis

❒ Current approaches in speech recognition are primarily focusing on modeling the vocal tract characteristics.

❒ Computing the cepstrum:

- find the log spectral magnitudes
- find the inverse FFT of the log spectrum

$$c(n) = \frac{1}{Ns}\sum_{k=0}^{Ns-1} \log 10|Savg(k)| e^{\left(2\pi k \frac{n}{Ns}\right)}$$

- MEL scale spaced cepstrum

❒ c(0) is the average value of the spectrum (discarded because absolute power measures of the signal are somewhat unreliable)

Linear Prediction Analysis

- ❑ Based on the least mean squared error theory
- ❑ If correct, predicts future values of the signal based on current measurements
- ❑ If error is small, model is good
  - ➔ Relation of speech samples to the excitation

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + Gu(n)$$

  - ➔ Linear predictor output with coefficients ak

$$\tilde{s}(n) = \sum_{k=1}^{p} a_k s(n-k)$$

  - ➔ Predictor Error

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k)$$

lpc algorithm (see ~webster/ee_8993/lpc/*.*)

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

LP Transformations

Linear prediction (LP) is one of the most powerful speech analysis tools, especially in estimating the basic speech parameters of pitch, formats, spectra, etc. This method offers very good estimates of speech parameters in addition to efficient computation.

For this reason, it is desirable to transform between various sets of parameters without loss of information.

Parameters of particular interest are:

1. FFT filter bank amplitudes
2. Reflection coefficients
3. Predictor coefficients
4. Cepstrum coefficients
5. Area ratios
6. Autocorrelation coefficients

Table 1: Transformation Routes

	FFT	RC	PC	CC	ALAR	RN
FFT	*****	xx		x		xx x
RC	x	*****	step		x	x
PC		step	*****			
CC	x			*****		
ALAR		x			******	
RN	x	x				*****

EXAMPLE

generate RC from RN

LEGEND

FFT = FFT filter bank amplitudes

RC = reflection coefficients

PC = predictor coefficients

CC = cepstral coefficients

ALAR = area ratios

RN = autocorrelation

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Feature Selection

- ❐ Once various sets of parameter coefficients have been generated, a feature vector can be formed
- ❐ Fewer than 6 types of coefficients offer poor recognition

SUMMARY

- ❒ Mel based cepstral coefficients are desirable to perceptually model a system
- ❒ FFT based parameters offer better performance than LPC based parameters
- ❒ Selection of initial parameter set is a vital role in speech recognition
- ❒ Signal modeling is a fundamental problem in this field of research
- ❒ Software technology for speech recognition is on the rise
- ❒ Robustness to noise is an impending aspect of speech recognition that will keep researchers busy well into the next century
- ❒ What does the future hold?

  Day by day, researchers and consumers are finding the need for robust speech recognition systems. The end result of the research going on today will greatly benefit mankind in the future...whether it is simply making life easier for the physically challenged or making everyday tasks (i.e., ATM withdrawals) easier to perform.

REFERENCES

1. Picone, J., "Signal Modeling Techniques in Speech Recognition," Proceedings of the IEEE, vol. 81, no. 9, pp. 1215-1246, Sept. 1993.
2. Rabiner, L.R. and R.W. Schafer, *Digital Processing of Speech Signals,* Prentice-Hall, Inc., 1978.
3. Proakis, J. G. and D. G. Mandakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, Macmillian Publishing, 1988.
4. Markel, J.D. and A.H. Gray, Jr., *Linear Prediction of Speech,* Springer-Verlag Publishers, 1982.
5. Deller, J. R., Jr., J. G. Proakis, and J. H.L. Hansen, *Discrete-Time Processing of Speech Signals,* Macmillian Publishing, 1993.
6. Flanagan, J. L. and L.R. Rabiner, *Speech Synthesis*, Dowden Hutchinson and Ross, Inc., 1973.
7. http://www.uninova.pt/~tr/home/tooldiag.html
8. http://www.speech.su.oz.au/comp.speech/Section6/Q6.5.html
9. ftp://ftp.cs.cmu.edu/project/fgdata/speech-compression/LPC/
10. http://www.lhs.com/
11. Pan, R. and C.L. Nikias, "The Complex Cepstrum of Higher Order Cumulants and Nonminimum Phase System Identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 36, no. 2, pp. 186-205, Feb. 1988.
12. Chen, Y., "Cepstral Domain Talker Stress Compensation for Robust Speech Recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 36, no. 4, pp. 433-439, Apr. 1988.
13. Lee, C.H., "On Robust Linear Prediction of Speech," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 36, no. 5, pp. 642-650, May 1988.
14. Furui, S., "Comparison of Speaker REcognition Methods Using Statistical Features and Dynamic Features," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 29, no. 3, pp. 448-449, June 1981.
15. Cheung, R.S. and B.A. Eisenstein, "Feature Selection via Dynamic Programming for Text-Independent Speaker Identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing,* vol. 26, no. 5, pp. 397-402, Oct. 1978.

IMPLEMENTATION OF STATISTICAL MODELING TECHNIQUES AND CHANNEL ADAPTATION TECHNIQUES

by

Raja Shekhar R Seelam seelam@isip.msstate.edu

Institute of Signal and Information Processing
Mississippi State University

ABSTRACT

Implementation of various Statistical Modeling Techniques is necessary for the building of a Speech Recogniser. Statistical Modeling is done to learn the nature of the multi-variate random process generating the signal parameters. In this direction, pre-whitening transformations were performed on the parameters to eliminate redundancy and to make the analysis easier.

The transformations were performed on the input feature vector to produce an uncorrelated Gaussian random vector, containing only “information-bearing” parameters. For some algorithmically complex computations such as the computation of the eigen values and eigen vectors, existing software was used.

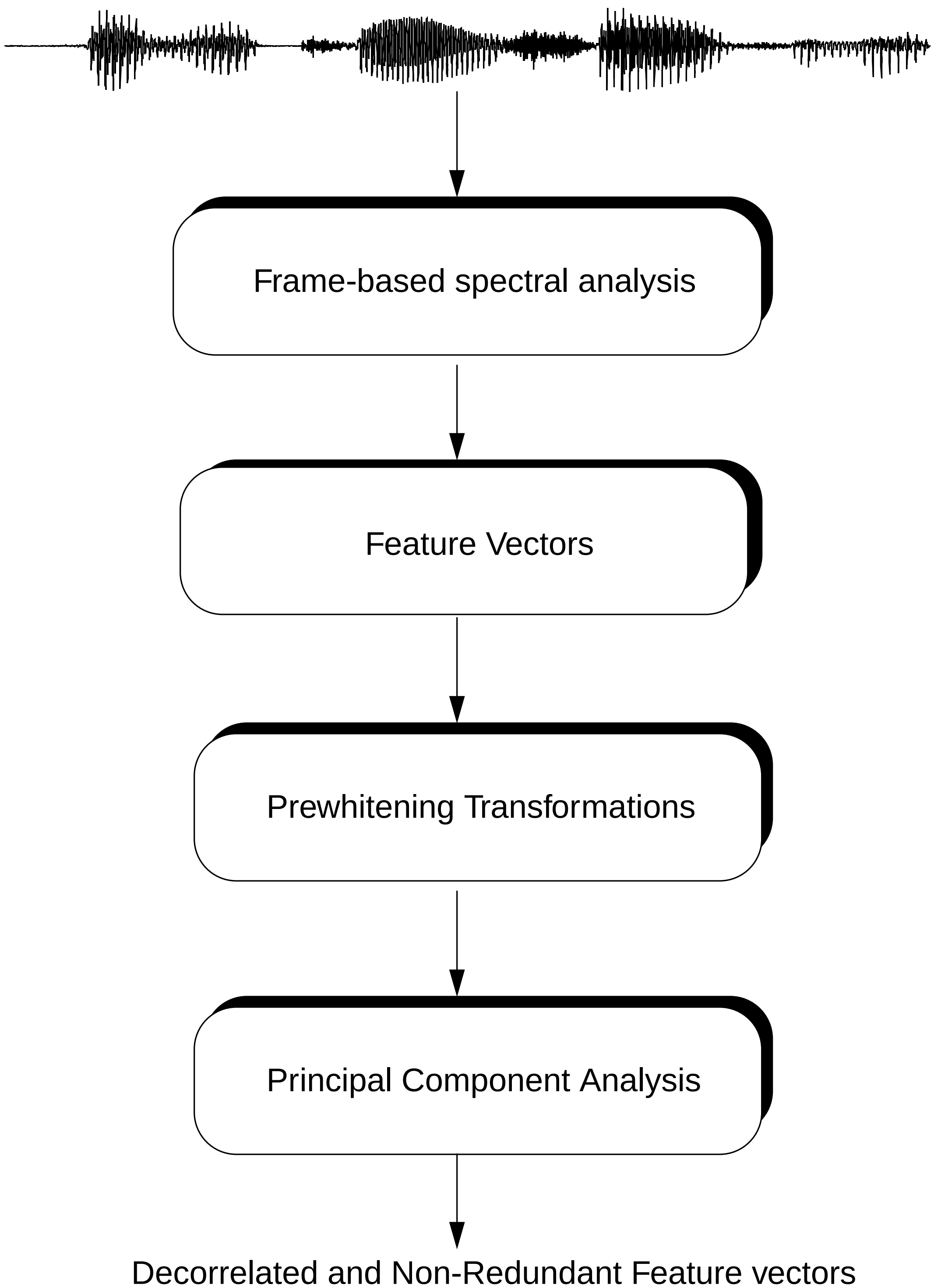
Channel adaptation techniques were implemented so as to make the parameters robust to changes in the acoustical environment. For this purpose, two particularly simple, but effective algorithms, Cepstral Mean Normalization/Subtraction and RASTA were chosen.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

THE PURPOSE

- Principal Component Analysis:
  - To learn the nature of the multi-variate random process generating the signal parameters.
  - To remove correlation and to eliminate redundancy from the features.
- Channel Adaptation:
  - Adapt the system to different kinds of acoustical environments.
  - The final aim is to make the Speech recognizer robust to different types of environments.

A SIMPLE OVERVIEW OF THE FRONT-END



THE MATHEMATICS

$\upsilon\ ,\ \mu_m$

Feature vector and its mean

$$C_v(i, j) = \frac{1}{N_f}\sum_{m=0}^{N_{ff}-1}(v_m(i)-\mu_v(j))(v_m(j)-\mu_v(j))$$

Covariance matrix

Λ, Φ

Eigen values and Eigen vectors of the Covariance matrix

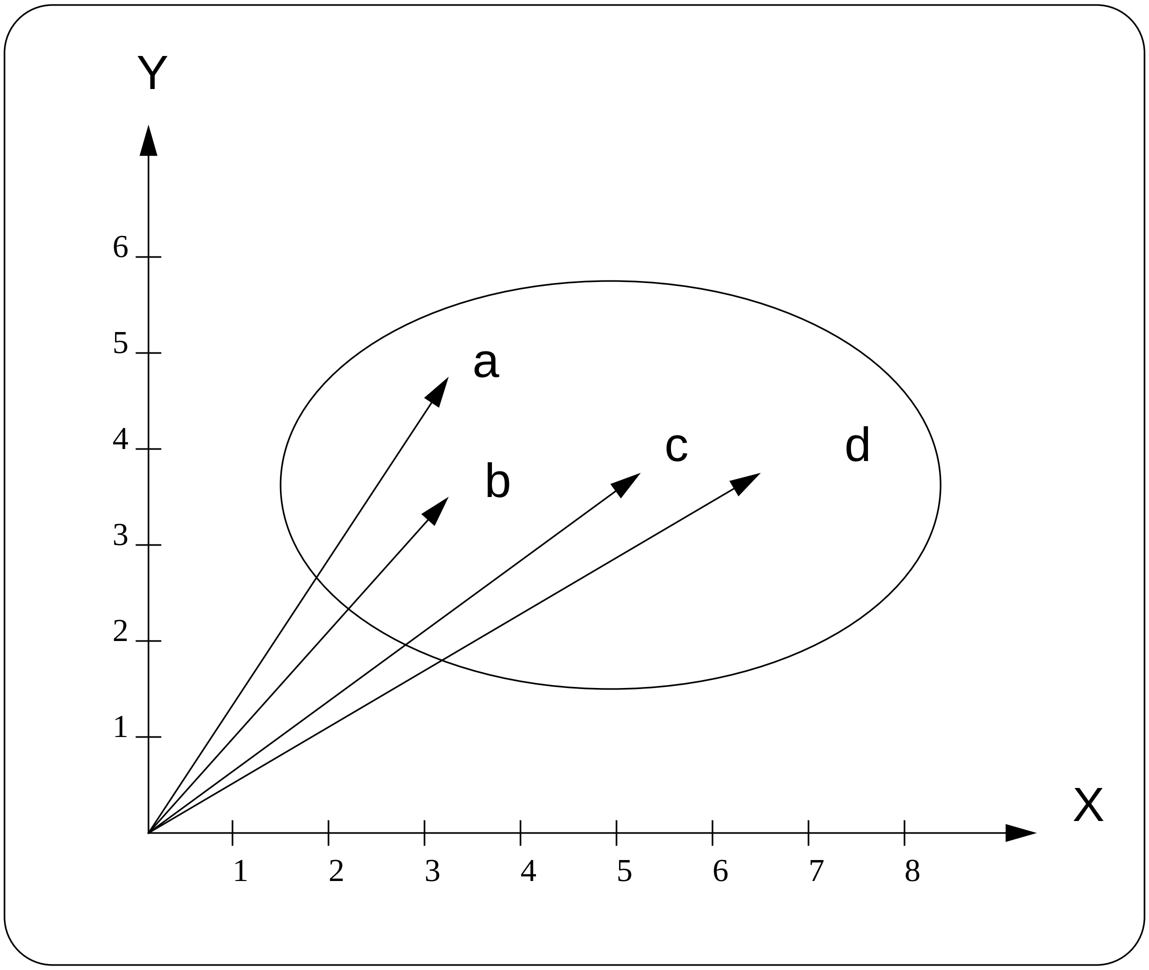
$$\Psi = \Lambda^{-1/2}\Phi$$

Transformation matrix

$$\bar{y} = \Psi(\bar{v}-\bar{\mu}_v)$$

Transformed Vector

VARIANCE WEIGHTING



Perceptually, the distance from a to b is larger than the distance from c to d even though both are shown to be the same, due to the fact that the distance a to b is a larger percentage of the variance in the vertical direction.

CHANNEL ADAPTATION

❒ The methods that have been implemented for this project:

- ❖ Cepstral Mean Normalisation (CMN)
- ❖ Relative Spectral Processing (RASTA)

❒ These two methods are simple to incorporate in the system and provide effective robustness.

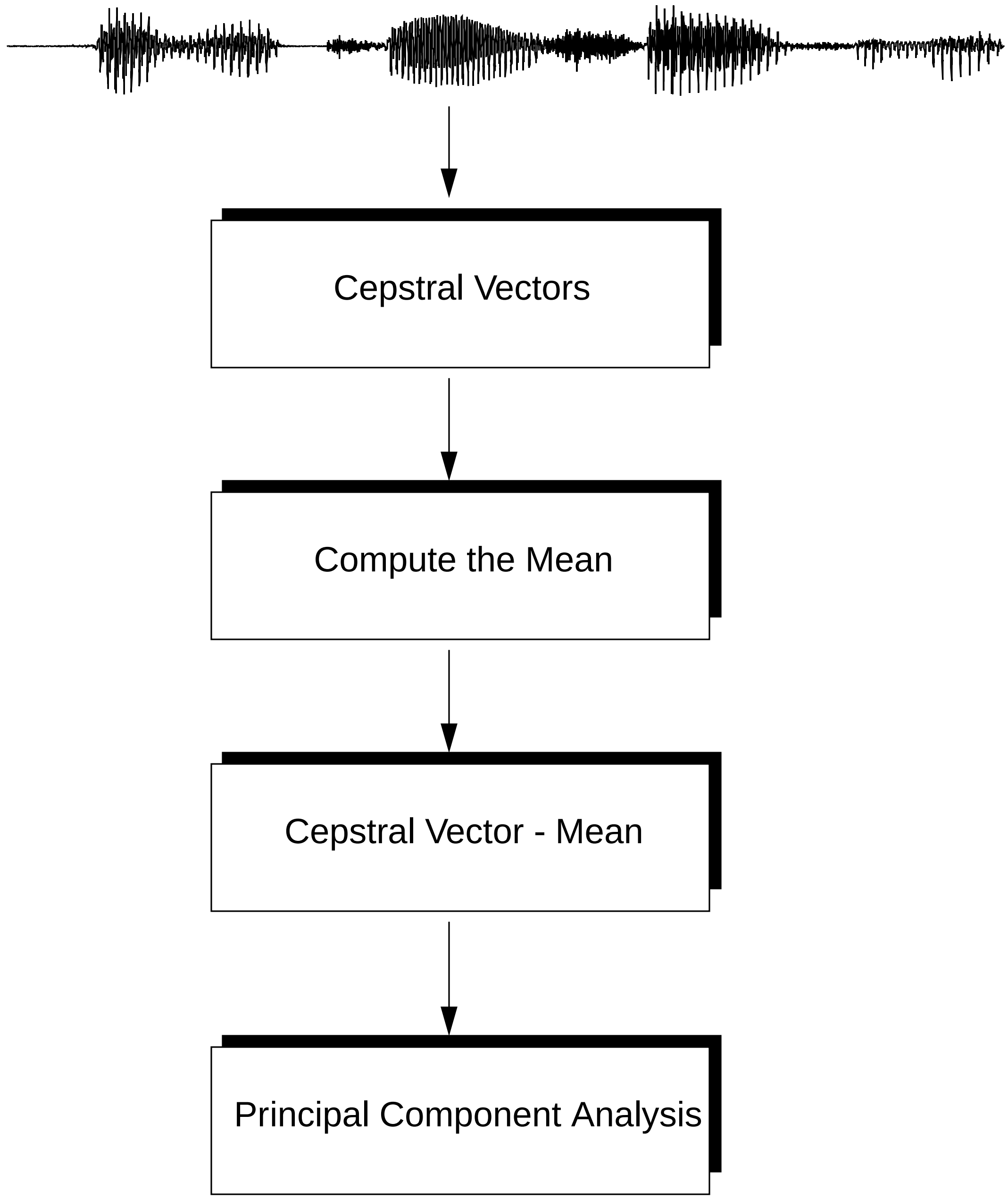
❒ Robust recognition is gaining importance due to its varied applications. The aim is to make the performance of the recognizer independent of the acoustical environment.

❒ Other methods that are popular include:

- ❖ RASTA - PLP
- ❖ CDCN (Developed by CMU)

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

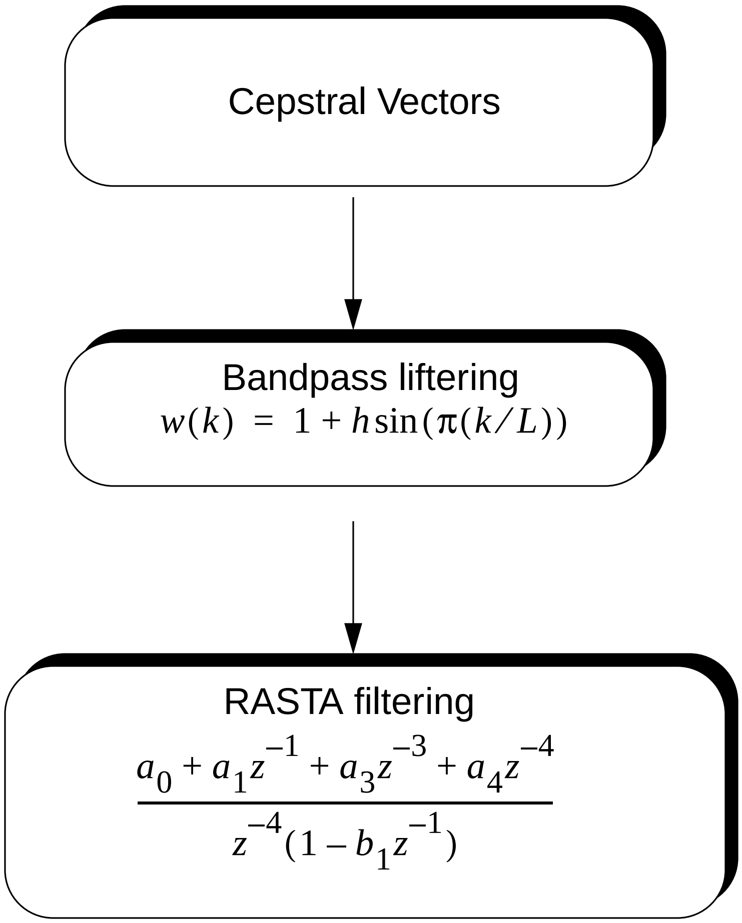
CEPSTRAL MEAN NORMALISATION



This has to be performed on clean data first and then on data obtained from noisy channels.

RASTA

(**R**el**A**tive **S**pec**T**r**A**l Processing)



RASTA, when used along with Bandpass liftering, has been proven to perform better than RASTA alone.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SUMMARY

- ❒ Basic software that can perform principal component analysis has been developed. The code has been developed keeping in mind that it has to be integrated with the other modules of the Speech recognizer.

- ❒ Extensive experimentation is to be done so as to make the code more generic and to arrive at the ideal number of features in the feature vector.

- ❒ I plan to develop a simple toolbox on the web using the code developed for this project. (On the lines of Cornell University's LASSP tools demos)

- ❒ Simple code to accomplish Cepstral mean normalization and RASTA exists. Detailed experimentation has to be done on this code.

- ❒ Extensive research is going on in the field of robust speech recognition and an ideal solution is yet to emerge.

REFERENCES

Principal Components:

[1] J. Picone, "Signal Modeling Techniques in Speech Recognition", in Proc. IEEE, vol. 81, no. 9, pp. 1215-1247, Sep. 1993.

[2] E.L. Bocchieri and G.R. Doddington, "Frame specific statistical features for speaker independent speech recognition", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, no.4, pp 755-764, Aug. 1996.

[3] J.R. Deller, J.G. Proakis, and J.H.L. Hansen, Discrete Time Processing of Speech signals. New York: MacMillan, 1993.

Channel Adaptation:

[4] Y.Kao, J.S. Baras, P.K. Rajasekaran, "Robustness study of free-text speaker identification and verification", Proc ICASSP, pp 379-382, Apr 1993.

[5] A. Acero, R.M. Stern, "Environmental robustness in Automatic speech recognition", Proc ICASSP, pp 849-852, Apr 1990.

[6] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP)", Proc of EUROSPEECH '91, pp 1367-1370, Genova, 1991.

[7] R.A. Gopinath, M. Gales, P.S. Gopalakrishnan, S. Balakrishnan-Aiyer and M.A. Picheny, Proc of ARPA Spoken Language Systems, 1994

[8] H. Hermansky, N. Morgan, and H.G. Hirsch, "Recognition of speech in additive and convolutional noise based on RASTA spectral processing", Proc ICASSP, vol. 2, pp 83 -85, 1993.

Software:

[9] W.H. Press, B.P. Flannery, S. A. Teukolsky, and W.T. Vettering, Numerical Recipes in C: The Art of Scientific Programming. New York: Cambridge Univ. Press, 1988.

Implementation of Viterbi Search Algorithm

by

Aravind Ganapathiraju ganapath@isip.msstate.edu

Department of Electrical and Computer Engineering
Mississippi State University

ABSTRACT

Speech Recognition can be treated in a very general sense as a structured search problem. Correct recognition is defined as outputting the most likely word sequence given the language model, the acoustic model and the observed acoustic data.

This work involves the implementation of a commonly used search algorithm, Viterbi Search. The implementation uses continuous observation HMMs to represent its word models. The algorithm is provides the most likely word sequence that could have produced the observed acoustic data. The code is object oriented and the structure has been made very generic so as to allow for using other search algorithms such as, Viterbi Beam Search, at a later stage.

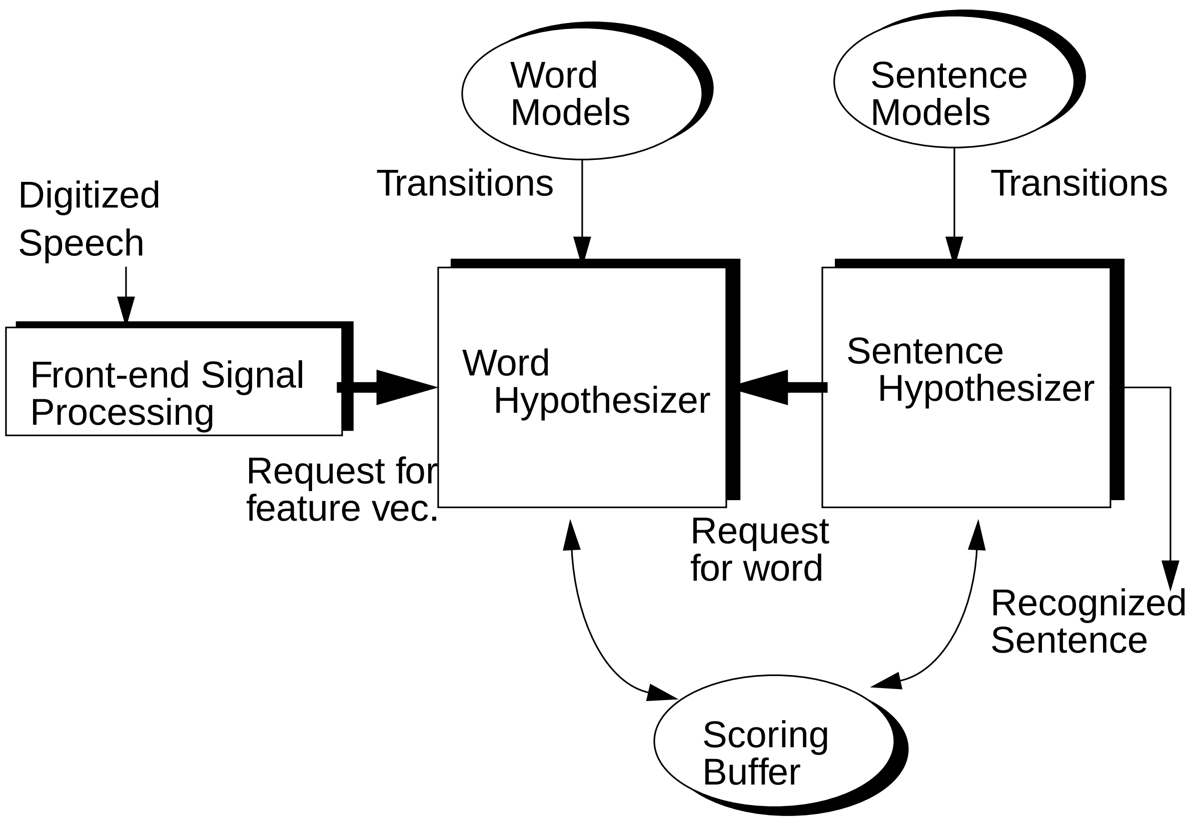
The design allows for integration of the search engine with various other modules of a speech recognizer including the language model, and the front-end signal processor. For experimentation a small language model has been created and dummy HMM models have been used. The Viterbi algorithm has been found to give the optimum solution to the search problem. It is not efficient in terms of memory. This basic framework will now be used to develop other efficient search algorithms.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

MOTIVATION

- ❒ Present day speech recognition technology is driven by applications involving speech understanding and multimedia
- ❒ Memory size is the primary obstacle to achieving good recognition performance in practical systems
- ❒ When we talk of real-time systems computational loads on hardware are very high
- ❒ A basic framework required to base future research in building a continuous speech recognition system
- ❒ This framework can then be extended to other algorithms and performance be compared

GENERIC RECOGNITION ARCHITECTURE



❒ Signal Modelling (Front-end Signal Processor)

❒ Acoustic Decoder (Word Hypothesizer)

❒ Language Model (Sentence Hypothesizer)

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

THE SEARCH PROBLEM

- ❒ Decoding strategy to find the most likely word sequence given the acoustic and language models
- ❒ Exhaustive search is computationally impractical
- ❒ The recognition problem can be formulated as a Maximum Likelihood (ML) path searching in a statistical network
- ❒ Most recognition systems use subword models and this further increases the number of possible word hypothesis and hence the computation
- ❒ Search space reduction possible by using syntactic and semantic constraints

SEARCH ALGORITHMS

❒ Viterbi Search

- ☞ Time synchronous search strategy
- ☞ Size of state space makes this impractical for large vocabularies
- ☞ Viterbi Beam Search employed to reduce the search space
- ☞ Breadth-first approach to the problem

❒ Stack Decoding

- ☞ State-synchronous search strategy
- ☞ Depth-first search strategy

❒ N-Best Search

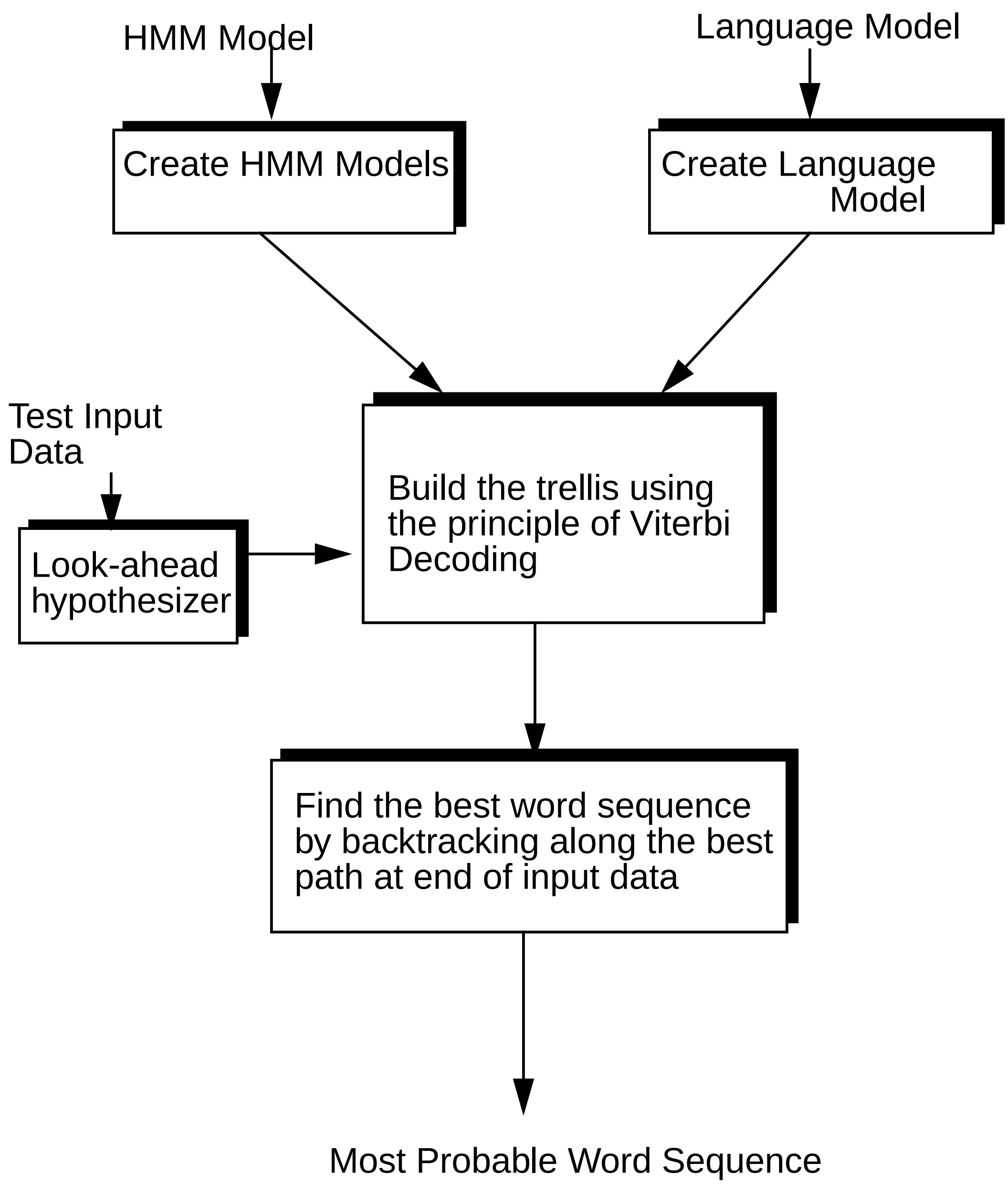
- ☞ Based on the Viterbi Search
- ☞ Keeps track of all hypotheses with different histories at each state
- ☞ Then allows N-top scoring hypotheses to propagate to next state
- ☞ This state level pruning is independent of global Viterbi pruning

❒ Forward- Backward Search

- ☞ Uses approximate search in forward direction by using simple acoustic models and language models
- ☞ A more complex search in backward direction performed

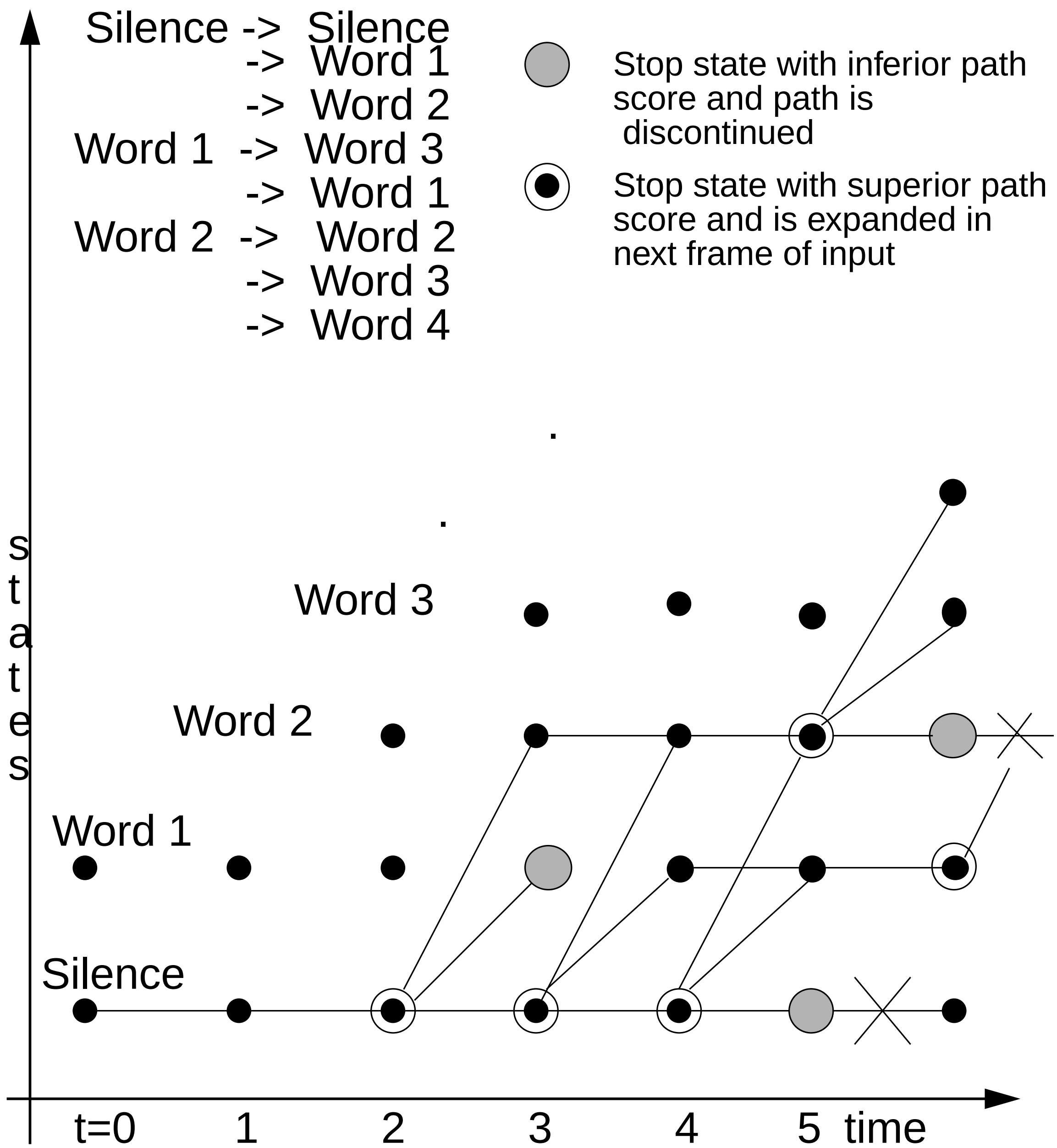
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

BASIC VITERBI IMPLEMENTATION



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

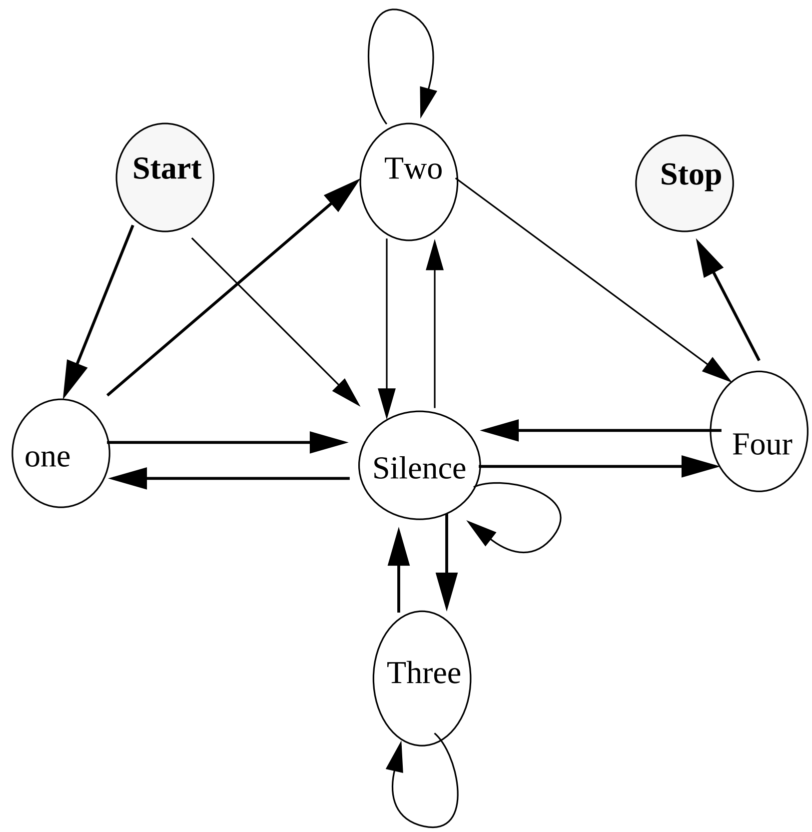
❐ State Representation of Viterbi Algorithm:



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

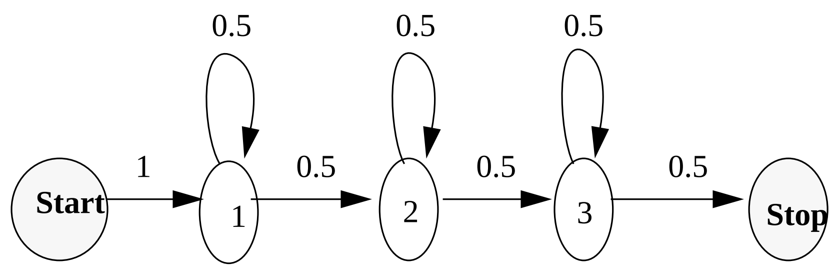
THE EXPERIMENTAL SETUP

❒ The Language Model:



❒ The Acoustic Model:

Left to Right Model Without Skip States



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

❒ Sample Output:

☞ The best word sequences and their scores at the end of test input

<Silence> three <Silence> <Silence> four <Silence>
21.779583

<Silence> four <Silence> <Silence> <Silence> four <Silence>
28.028919

<Silence> four <Silence> <Silence> <Silence> four <Silence>
30.028919

☞ The Viterbi algorithm as a best-path approach :

Word	**Frame**	**Score**
<Silence>	17	15.667695
three	17	-3.607835
four	17	0.200641
three	17	-3.775326
four	17	0.577218

Same words with different path histories

best :<Silence> 15.667695

Only the <Silence> model is allowed to expand into other words at the next input frame

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SALIENT FEATURES OF THE CODE

❒ Object Oriented

- ☞ Intuitive representation of the problem
- ☞ Objects for HMM-Models, Hypothesis scores, Matrices

❒ Generic in Nature

- ☞ Many of the data-structures like the linked lists etc. are generic
- ☞ No data is hard coded, to allow for easy debugging and development

❒ Data Driven

- ☞ Data provided by the user to build the models - there dimensions, topology etc. are provided by the user

❒ Expandability

- ☞ The present code can be easily adapted to accommodate the other modules of the recognizer such as , front-end and the language model

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SUMMARY & FURTHER RESEARCH

- A simple and a basic form of a Viterbi search engine has been built and tested on synthesized data.
- Viterbi Search Guarantees us with the best possible word sequence given the input data.
- Viterbi algorithm is inefficient in terms of computational efficiency and memory as seen from simulation results
- Improvements in terms of memory usage have to be incorporated by allowing for reusing data structures
- The existing basic tools will be used as building blocks for a full fledged continuous speech recognizer
- Other search strategies such as beam search, N-Best ,Multi-pass search etc. will be developed.

REFERENCES

1. J.R. Deller, J.G. Proakis, and J.H.L. Hansen, *Discrete Time Processing of Speech Signals*, MacMillan, New York, New York, USA, 1993.
2. L. Rabiner and B.H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1993.
3. J. Picone, "Continuous Speech Recognition Using Hidden Markov Models," *IEEE ASSP Magazine*, vol. 7, no. 3, pp. 26-41, July 1990.
4. Steve Young, "Large Vocabulary Continuous Speech Recognition: A Review", to appear in the IEEE Signal Processing magazine.
5. H.Ney,D.Mergel, A.Noll and A.Paeseler,"A Data-Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition", Proc. ICASSP, pp. 132-126, April 1987.
6. S.B. Lippman, *C++ Primer, 2nd Edition*, Addison-Wesley, Reading, Massachusetts, USA, 1991.
7. *The C Programming Language, 2nd Edition*, Prentice Hall, Englewood Cliffs, New Jersey, USA, 1991.

EFFICIENT SEARCH ALGORITHMS FOR LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

by

Neeraj Deshmukh

Institute for Signal and Information Processing (ISIP)
Mississippi State University
Mississippi State, MS 39762
deshmukh@isip.msstate.edu

ABSTRACT

Automatic speaker-independent speech recognition has made significant progress from the days of isolated word recognition. A major component of this advancement in technology is due to recent advances in search techniques that support efficient, sub-optimal decoding over large search spaces and complex statistical models. Moreover, these evaluation strategies are capable of dynamically integrating information from a number of diverse knowledge sources to determine the correct word hypothesis.

In this project we propose to implement two major classes of such decoding algorithms viz. multipass N-best search and search using acoustic models that employ a weighted mixture of Gaussian probabilities as density functions. This will later be integrated with other software modules implementing a language model and a speech signal-processing front-end to build a complete LVCSR system. The performance of this LVCSR system will be evaluated on speech data available in the public domain and compared with that of other recognizers as a benchmark. We will first evaluate the search algorithm modules in isolation using statistical measures and synthetic data. The final software will be placed in the public domain at the Institute of Signal and Information Processing (ISIP).

Keywords: Gaussian Mixtures Models, Forward-Backward N-Best Search

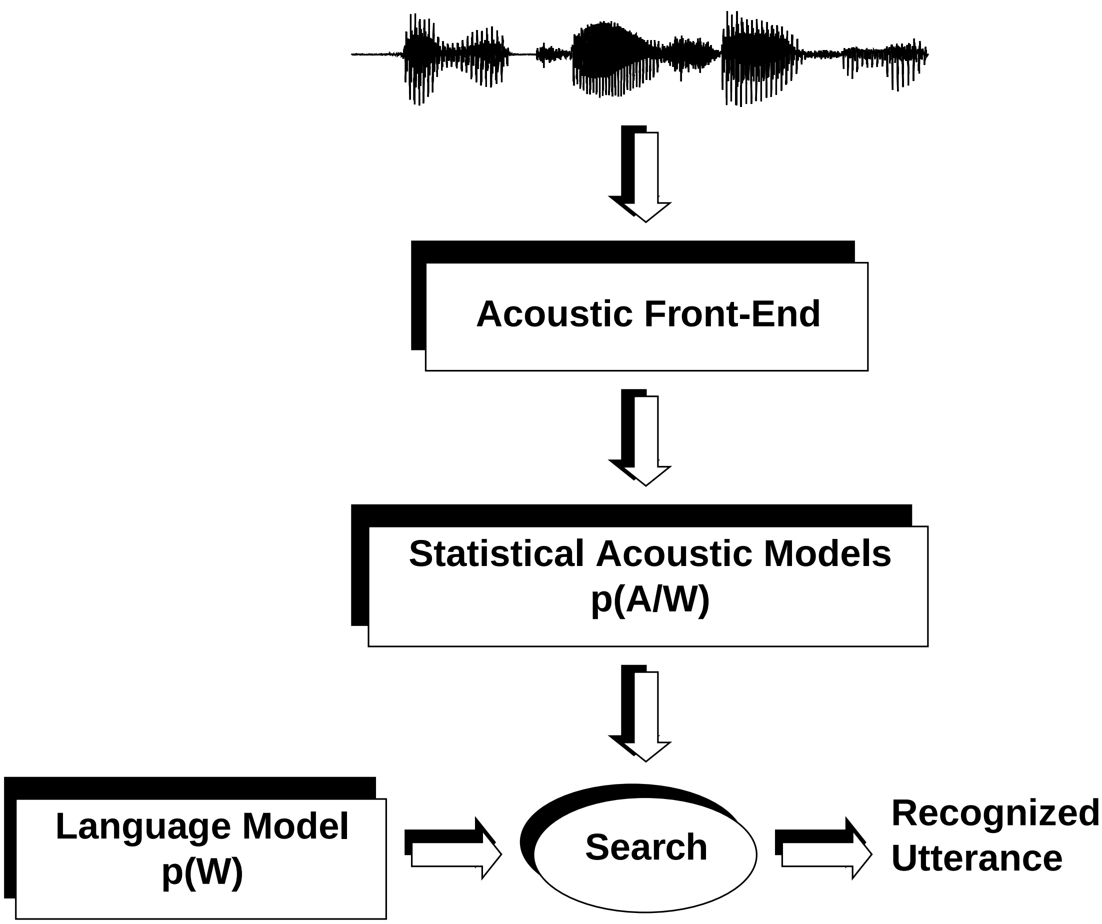
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

THE SPEECH RECOGNITION PARADIGM

- Bayesian Statistical Pattern Matching

$$\widehat{W} = \underset{W}{\arg max} \quad p(W)p\langle A|W\rangle$$

- A Typical Speech Recognition System

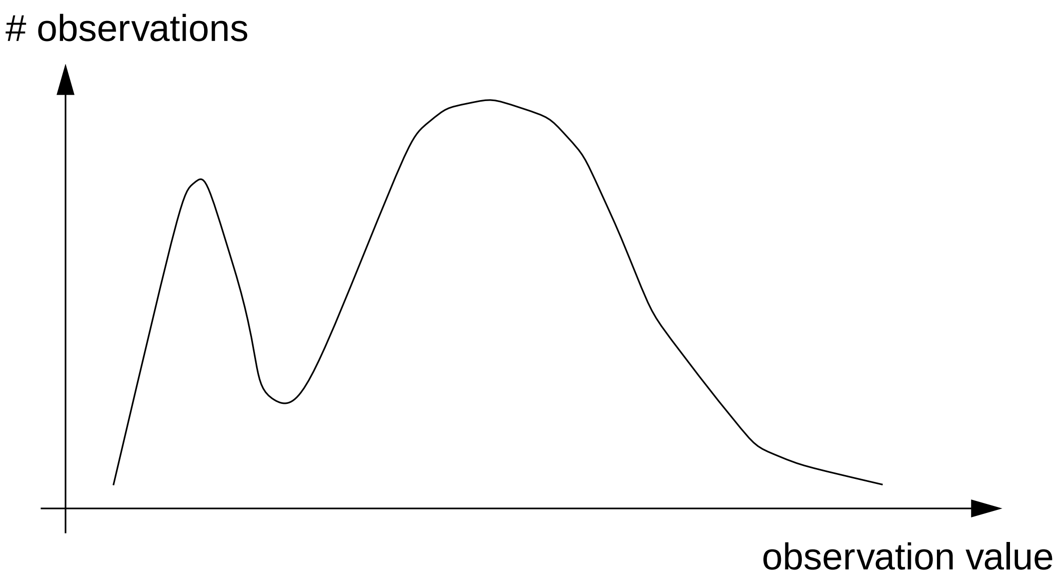


DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

TOWARDS BETTER ACOUSTIC MODELS

- Hidden Markov Models
  - continuous density
  - context dependence
  - phone models vs. word models

- Different modalities of same phonemes



  - No single distribution function can model this well
  - Linear combination of component distributions to model the net density function

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

GAUSSIAN MIXTURE MODELS

- Gaussian Mixture Distributions

$$f_{Y/X}\langle \xi | i \rangle = \sum_{m=1}^{M} c_{im} \aleph(\xi, \mu_{im}, C_{im})$$

where $$\sum_{m=1}^{M} c_{im} = 1$$

- Typical systems use 8 - 15 mixtures per state
  - computational overhead for training
  - added storage requirements

- State-tying and Mixture-tying
  - models share similar states
  - states share similar mixture components

- Training
  - Baum-Welch Forward-Backward Algorithm

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SEARCH IN SPEECH RECOGNITION

❒ Search Paradigm

☞ Choose the hypothesis with the highest likelihood score for the acoustic and language models given the observed data

❒ Motivation for Strategy

☞ exhaustive search is impractical

❒ Approaches to Restructure Search

☞ optimize hypothesis generation

☞ reduce the problem space using transformations

☞ dynamic programming / maximum-likelihood

☞ apply external knowledge / heuristics

❒ Sub-optimal choices

☞ no significant effect on error rate

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

N-BEST SEARCH

❒ Basic Algorithm

- ☞ employs the Viterbi beam search
- ☞ maintains all hypotheses within beam
- ☞ propagates top N hypotheses at every state
- ☞ N is independent of beam width

❒ Practical Issues

- ☞ Means to integrate information from diverse sources
- ☞ Multi-pass applications
- ☞ Partial to shorter hypotheses

❒ Generalized Multi-pass N-Best Search

- ❍ *Lattice N-Best*
  - ☞ build a lattice of word hypotheses in first pass
  - ☞ downsize lattice in subsequent passes
  - ☞ choose N top hypotheses using recursive backtrace
- ❍ *Forward-Backward N-Best*

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

FORWRAD-BACKWARD SEARCH

❒ Forward Pass

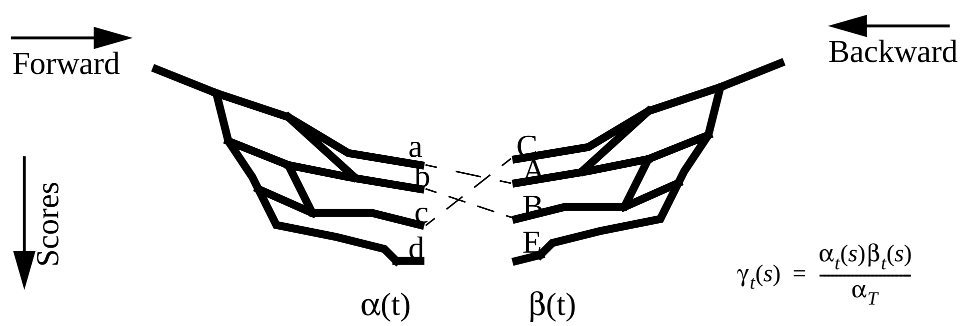
- fast search
- cheap, efficient models; simple grammar (unigram)
- determine possible word end-points

❒ Backward Pass

- detailed beam search
- reduced search space
- detailed models and more constrained grammar
- determine possible word beginnings

❒ Compiling the Two Passes

- combine scores if both passes yield good scores
- trace the N best-scoring hypotheses



$$\gamma_t(s) = \frac{\alpha_t(s)\beta_t(s)}{\alpha_T}$$

IMPLEMENTATION FEATURES

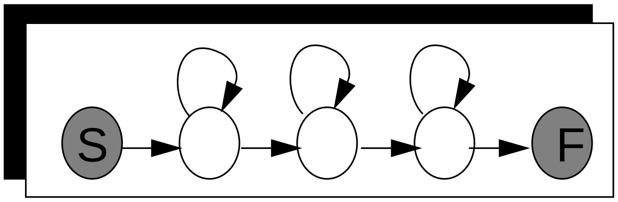
- Object-oriented Thrust
  - data-driven parameters
  - linked-list based implementation for flexibility
  - modular architecture to plug into variety of HMM-based applications
  - integration with other modules (front-end, language model) to form a complete LVCSR system

- Public-domain Software Development
  - software available from ISIP web site

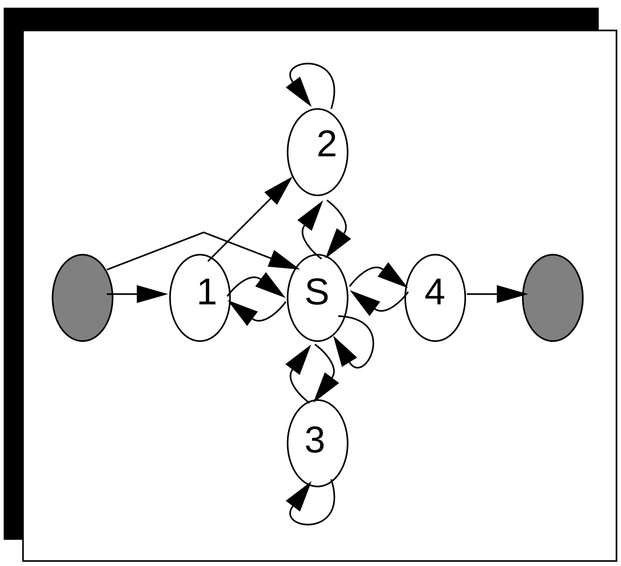
- Salient Features
  - user-defined HMM topology
  - no constraint on number of states, transitions or mixture components
  - user-defined grammar

- Experiments
  - synthetic data
  - testing as stand-alone modules

EXPERIMENTAL SETUP



The HMM topology used in experiments



Key:
word models
1 one
2 two
3 three
4 four
S <silence>

The language grammar used in experiments

- digit-string recognition
- five word models of fixed topology
- five states per model
- arbitrary (two / three) mixtures per state
- forward pass N-best

EXPERMINTAL RESULTS

❒ Forward-pass search with mixture models

- two or three component mixtures
- utterance constrained to start with silence
- synthetically generated data (with option to add noise)
- Euclidean distance as model scoring criterion

❍ **Reference sentence**

<Silence> one two four three <Silence>

<Silence>	one	two	four	three	<Silence>	
6	6	4	5	8	3	duration (frames)

❍ **Recognized Sentence Hypotheses**

<Silence> <Silence> one two four <Silence> three <Silence>

Score: 114.37

<Silence> <Silence> one two four <Silence> three <Silence> <Silence>

Score: 111.87

<Silence> <Silence> one two four <Silence> three <Silence>

Score: 115.36

CONCLUSION

❒ Summary

- ☞ Efficient search techniques and better acoustic models are vital for improved performance on LVCSR tasks
- ☞ The techniques implemented in this project represent the current state of the art
- ☞ This software is intended to be part of a larger project dedicated to provide a flexible public-domain LVCSR system
- ☞ A demonstration of this algorithm is available

❒ Future Directions

- ☞ Complete the implementation of the backward pass search
- ☞ Integrate the search modules with the other components to complete the recognition system
- ☞ Train and test the system on real data
- ☞ Compare recognition performance with other LVCSR systems on similar tasks

REFERENCES

1. Deller J.R.,Proakis J.G. and Hansen J.H.L., *Discrete-Time Processing of Speech Signals*, Macmillan Publishing, New York, 1993.
2. Rabiner L.R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *in Proceedings of IEEE*, Vol. 77, No. 2, pp. 257-285, 1989.
3. Lippman R.P. and Gold B., "Neural-Net Classifiers Useful for Speech Recognition", i*n Proceedings of the 1st IEEE International Conference on Neural Networks*, Vol. IV, pp. 417-425, San Diego, CA, 1987.
4. Viterbi A.J., "Error Bounds for Convolutional Codes and an Asymptotically Optimal Decoding Algorithm", *in IEEE Transactions on Information Theory*, Vol. IT-13, pp. 260-269, April 1967.
5. Paul D.B., "An Efficient A* Stack Decoder Algorithm for Continuous Speech Recognition with a Stochastic Language Model", *in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 405-409, 1992.
6. Chow Y.L. and Schwartz R.M., "The N-Best algorithm: An Efficient Procedure for Finding Top N Sentence Hypotheses", *in Proceedings of DARPA Speech and Natural Language Workshop*, pp. 199-202, October 1989.
7. Schwartz R.M. and Austin S.,"Efficient, High-Performance Algorithms for N-Best Search", *in Proceedings of the DARPA Speech and Natural Language Workshop*, pp. 6-11, June 1990.
8. The WWW Resources.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Language Modeling and Grammar Construction for a Hidden Markov Model Continuous Speech Recognition System
by

Owen LaGarde lagarde@isip.msstate.edu

EE 8993 Language Modeling Group
Mississippi State University

ABSTRACT

Language models compose a performance-critical segment of the HMM continuous speech recognition system, providing context sensitive embedding of *a priori* knowledge for the problem domain. The language model also aids in constriction of the search space and thereby defines, to a large degree, maximum performance for the global system. Two distinct though closely related approaches to modeling language via a structured grammar can be derived from a single set of base resources which in turn are generated through simple counting methods applied to training text. More complex access and maintenance methods are required, however, to efficiently employ such models in real-time systems due to size and operational speed constraints; a typical model will enclose more information than can be easily loaded in conjunction with the recognition system, thereby requiring a resource management and data structure scheme. In addition, limited preparation and formatting of data sets is required to achieve good results and meaningful evaluation of the finished model independent of the target recognition system can be difficult.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Objectives

- Prediction of Future Language Domain Events
  - representative of state space for a specific language and usage domain
  - evaluation and ordering of possible solutions based on probability of correctness
- Operation in Conjunction with Real-Time Systems
  - efficient data structures extensible for alternate or variant models
  - resource management schema tunable for size and speed

Three Standard Approaches

- Deterministic Grammars
  - probability of transformation -- production rules

$$A \Rightarrow b \qquad or \qquad A \Rightarrow aB$$

  - top-down parsing -- expansion of general to specific hypothesi

$$G = (V_n, V_t, P, S)$$

  - ARPA and Natural Language Understanding

- Stochastic Grammars
  - probability of transition -- Ngrams

$$A \rightarrow \beta \qquad P\langle A\beta\rangle$$

  - bottom-up or left-right parsing -- best-path extension of hypothesized solution paths
  - the CMU SLM Toolkit

- Characteristic Grammars and Hybrids
  - deterministic derived from stochastic
  - class grammars as grouping constructs

Elements of the Solution

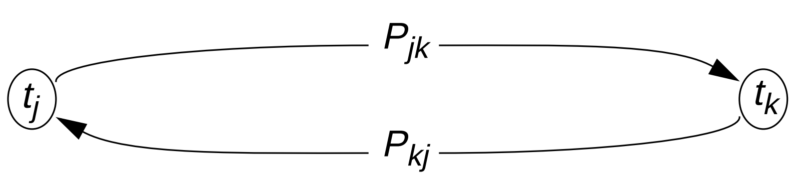
- Basic Stochastic Units
  - Unigrams, Bigrams, and Trigrams
  - Construction methods reduce to counting functions

$$P(t_i) = \frac{F(t_i | V^*)}{F(t^* | V^*)}$$

- Generalization of Ngrams
  - Probabilities derived from Bigram Lists

$$P(a,d) = P(a,b) \cdot P(b,c) \cdot P(c,d)$$

- Mapping to Network Representation
  - Models representable as weighted directed graphs



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Basic Stochastic Units

- Unigrams -- Probabilities for Symbols
  - Simple Frequency Function
  - Basis for a Vocabulary

$$P(t_i) = \left.\frac{F(t_i | V^*)}{F(t^* | V^*)}\right| \qquad \exists t^*$$

- Bigrams -- Symbol-to-Symbol Transitions
  - Probabilities for transitions vs. transformations
  - Unigrams, Bigrams, and Trigrams

- Trigrams -- The Beginnings of Context Encoding
  - Transitions after specific neighbor symbols

$$P\left(\omega_j^{\ i}, V^*\right) = \left.\frac{F\left(\omega_j^{\ i}, V^*\right)}{F\left(\omega_m^{\ n}, V^*\right)}\right| \qquad \omega \subseteq V^*, (m \le N), (n \le N)$$

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Generalization of Ngrams

❒ Application of Transitive Properties

$$P(a,d) = \prod P(i, i+1) \qquad i = [a, \ldots, (d-1)]$$

- ☞ Verifiable via comparison with Trigrams
- ☞ Constant factors allow tuning of output probabilities for long sequences
- ☞ Ngram model representable via *N* x *N* implicitly indexed matrix

	T_0	T_b	T_c	T_d	T_N
T_0					
T_a		P_{ab}			
T_b			P_{bc}		
T_c				P_{cd}	
T_N					

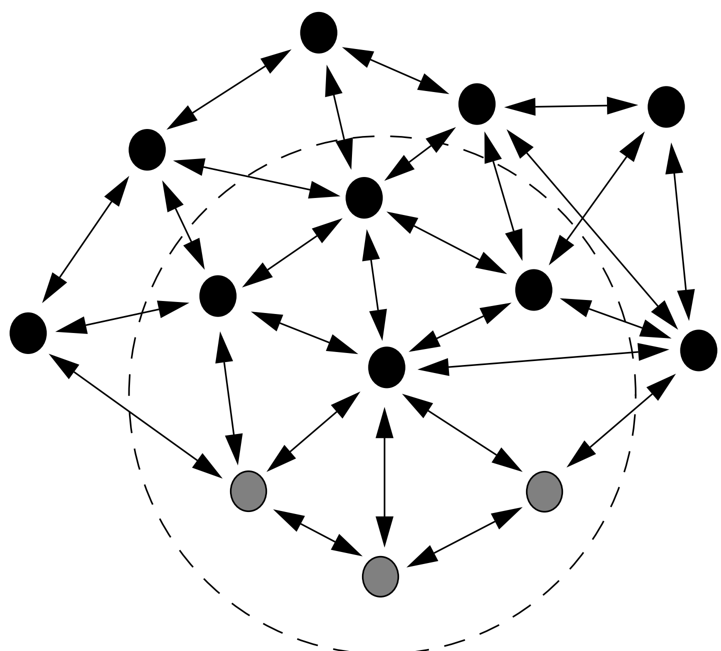
- ☞ Longer sequences may need emphasis relative to series length to keep fitness values within data type ranges

$$P(a,d) = (P(a,b) \cdot P(b,c) \cdot P(c,d)) \cdot K$$

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Network Representation

- Directed weighted graphs
  - Prune zero probability transition from the Bigram Matrix
  - Partial load levels defined by depth of traversal, maximum number of nodes = <number of symbols>$^{\text{<load level>}}$
  - Predict update requirements based on search engine's current position in the model
  - Advanced matching criteria



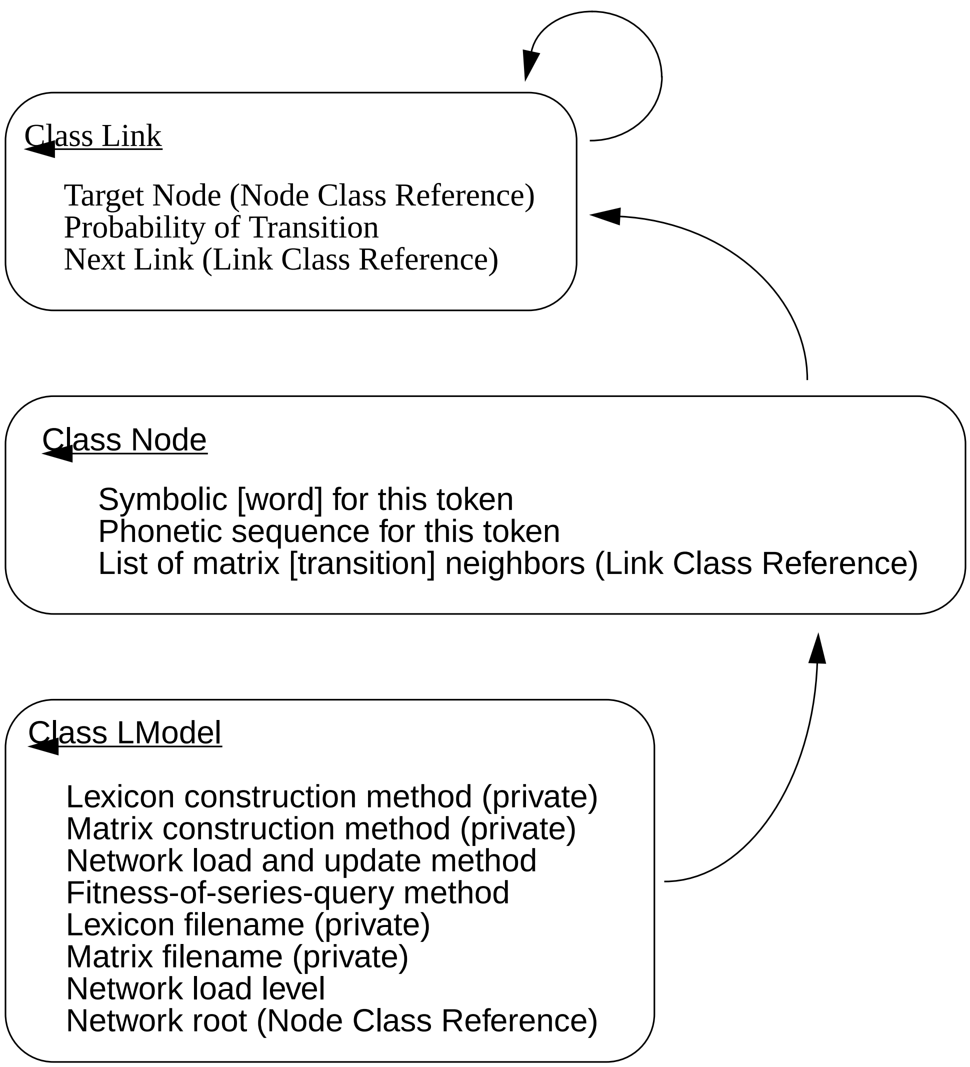
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

External Resources -- File Structures

- Word Frequency Matrix (WFM) -- probabilities of symbol sequences
  - **<implicit symbol *i* index> <implicit symbol *j* index> <P(*i,j*)>**
- Word Frequency List (WFL) -- probabilities of individual symbols
  - **<implicit symbol *i* index> <P(*i*)>**
- Word-Token List (WTL) -- symbol - to - index token map
  - **<implicit symbol *i* index> <text for symbol *i*>**
- Word-Phone List (WPL) -- symbol - to - phone sequence map
  - **<implicit symbol *i* index> <Worldbet phone sequence for symbol *i*>**
- Model Lexicon -- combination of WFL, WTL, WPL records
  - **<implicit index> <unigram> <token> <phones>**

Internal Resources -- Data Structures

- "Network" pointer management maintained by node and link objects
- Constructor requires training text source and generates system resources



DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Summary -- Points to Ponder

- Desirability of many side effects of the stochastic approach is defined by individual implementation
  - Basic resources can be constructed using shell scripts
  - Ngram generation from Bigram table is relatively fast, accurate, and compact
  - Implicit indexing, possible because basic units are functions of counting functions over lists, allows for highly condensed model representation
  - Token-based stochastic model can easily support a class grammar structure as a grouping methodology
  - Use of matrix introduces need for smoothing functions -- for both explicit zeros in the matrix and implicit zeros for symbols not in the training text
  - Requires knowledge of search engine output to optimize management of the model-as-network
  - Requires a word-phone dictionary to supply the HMM search engine and acoustic model with phone equivalents of current states
  - Training text requires formatting -- removal of punctuation, substitution of abbreviations, tagging of sentence boundaries, etc., prior to model construction.

References

1. **Deller, J. R., J. G. Proakis, and J. H. Hansen. 1993.** *Discrete-Time Processing of Speech Signals*. Macmillan Publishing, New York, pp. 677-804
2. **Rosenfeld, Ronald. 1995.** *CMU Statistical Language Modeling Toolkit: Manual*. Carnegie Mellon University Internet WWW Service, online URL "*http://www.cs.cmu.edu/afs/cs//user/roni/WWW/toolkit-SLT95-revised.ps*"
3. **Rosenfeld, Ronald. 1995.** *Optimizing Lexical and N-gram Coverage Via Judicious Use of Linguistic Data*. Carnegie Mellon University Internet WWW Service, online URL "*http://www.cs.cmu.edu/afs/cs//user/roni/WWW/vocov-eurospeach95-proc.ps*"
4. **Rosenfeld, Ronald. 1995.** Adaptive Statistical Language Modeling: A Maximum Entropy Approach. Ph.D. Thesis, Carnegie Mellon University, Technical Report CMU-CS-94-138, Carnegie Mellon University Internet WWW Service, online URL "*http://www.cs.cmu.edu/afs/cs//user/roni/WWW/thesis.ps*"
5. **Picone, Joseph. 1990.** *Continuous Speech Recognition Using Hidden Markov Models*. IEEE ACASSP Magazine, July 1990:26-40, IEEE Society Publications 1990
6. **Picone, Joseph. 1994.** *Context-Sensitive Statistical Signal Processing: Toward User Configurable Speech Recognition.* Systems and Information Science Laboratory, Texas Instruments, presentation March 22 1994
7. **Chomskey. 1959.** *On certain formal properties of grammars*. Information and Control 2:137-16, Dellar Associates
8. **Ries, Klaus, Finn Dag Buo, and Ye-Yi Wang. 1995.** *Improvised Language Modeling by Unsupervised Acquisition of Structure*. Interactive Systems Laboratories, Carnegie Mellon University, ICASSP Lecture Notes May 1995, Carnegie Mellon University Internet WWW Service, online URL "*http://www.cs.cmu.edu/afs/cs//project/cmt-38/ries/www/icassp_95.html*"
9. **Rudnicky, A., F. Lee, and A. G. Hauptmann. 1994.** *Survey of Current Speech Technology.* Communications of the ACM 37(3):52-57, Carnegie Mellon University Internet WWW Service, online URL "*http://www.speech.cs.cmu.edu/rspeach-1/air/papers/cacm94.ps*"

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Development of an N-Gram Based Language Model for Continuous Speech Recognition

by

Steve Given given@isip.msstate.edu

Language Modeling Group
Mississippi State University

ABSTRACT

An essential element of any speech recognition system is the language model. A language model attempts to identify and make use of the regularities in natural language to better define language syntax for easier recognition. One major obstacle in speech recognition is variability and uncertainty of message content. This, coupled with inherent noise, distortion and losses that occur in speech, emphasize the need for a good language model[1].

Several different types of language modeling techniques exist. This project will concern itself mainly with statistical language modelling. Statistical language modelling uses large amounts of text to automatically compute the model's parameters. This is called training. Language models can be compared using standard measures such as perplexity and recognition or word error rate. This project will use perplexity as a benchmark.

An ideal language model will provide *a priori* probabilities for all possible queries that the search algorithm may request. Hence, the complexity of the model is directly related to the size of the corpus upon which it is trained.

NGRAM STATISTICAL LANGUAGE MODELS

❒ Statistical language model.

☞ What is an N-Gram SLM?

☞ Where does it fit in?

☞ How do we train our model?

☞ What is coverage?

☞ What is smoothing?

☞ How do we benchmark a language model?

❒ CMU-SLS Language Modeling Toolkit

☞ What is the CMU toolkit?

☞ Why consider the CMU toolkit?

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

What is an N-Gram SLM?

❒ A language model that gives the probability of a correct hypothesis given the history of (N-1) previous words.

$$\hat{W} = w_1, w_2, \ldots, w_N \tag{1}$$

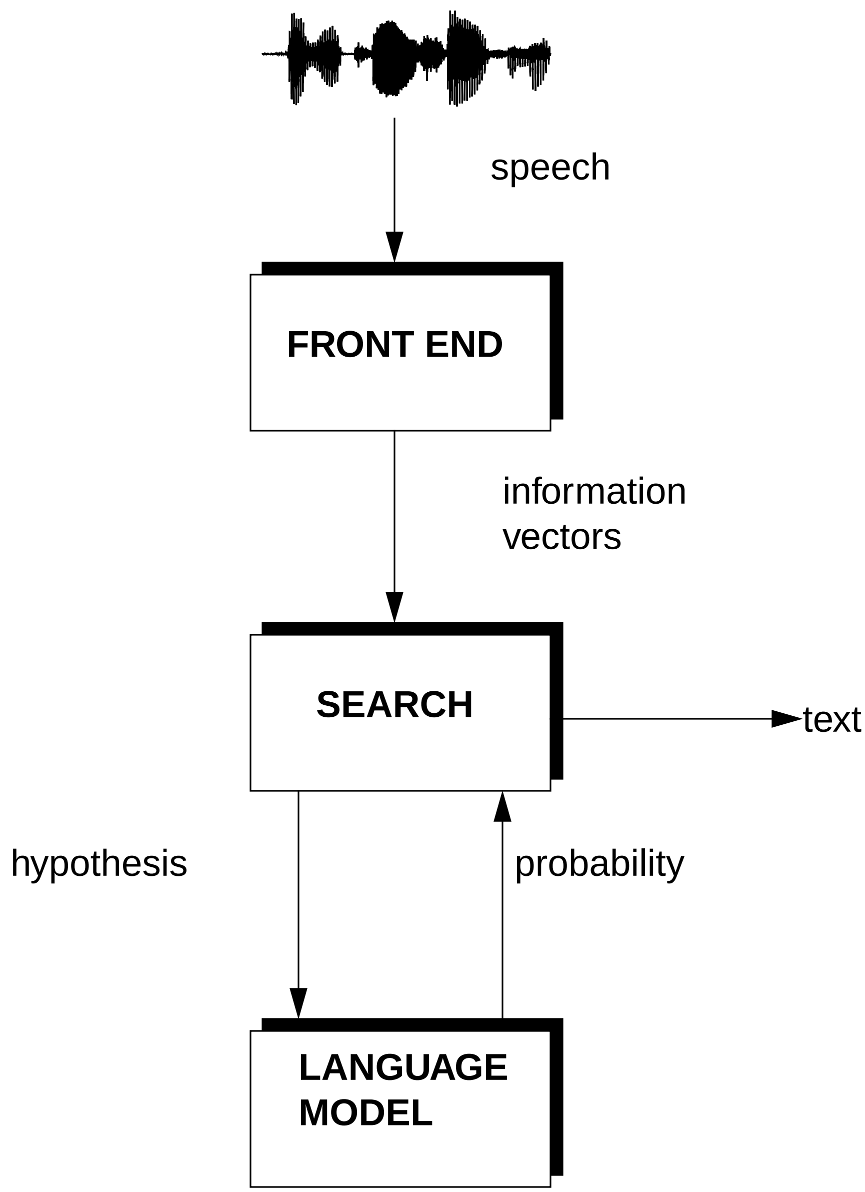
$$p\langle \hat{W} | Y \rangle = \max_{W} p\langle \hat{W} | Y \rangle \tag{2}$$

$$W = \operatorname{arg}\max_{W} p(W) p\langle Y | W \rangle \tag{3}$$

$$p(W) = \prod_{i=1}^{N} p\langle w_i | w_1, w_2, \ldots, w_{i-1} \rangle \tag{4}$$

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

THE BIG PICTURE- WHERE DOES THE LANGUAGE MODEL FIT IN?



TRAINING

- The process of building a language model from a text database.

- Building the corpus is one of the most time consuming tasks.

- Some text from the CMU corpus.

  <s> HAVE DROPPED THE CASE </s>

  <s> I'M BETTING ON THAT CHANCE </s>

  <s> ALEC SMIRKED </s>

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

COVERAGE

- ❒ Coverage describes how well the data is modeled.
- ❒ Coverage problems arise when outliers are not accounted for properly.
- ❒ High out-of-vocabulary(OOV) rate means poor performance.

SMOOTHING

- Smoothing is a general term which means adjusting the distribution so that P(W) != 0
- Deleted (Linear)Interpolation is one method for smoothing.
- Why smooth?

BENCHMARKING

- Perplexity is essentially the "branch-out" factor of the language model.

$$Q(\underline{w}) = 2^{\hat{H}(\underline{w})} \approx \frac{1}{\sqrt[N]{\hat{P}\left(w_1^N\right)}} \qquad \text{perplexity}$$

$$\hat{H}(\underline{w}) = -\lim_{N \to \infty} \log \hat{P}\left(\underline{w}_1^N = \underline{w}_1^N\right) \qquad \text{entropy}$$

- Word error rate is another common measure of a language model's(and speech system in general) worth.

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

CMU-SLS Language Modeling Toolkit

- ❒ Provides the ability to generate everything from basic word counts to Tri-Grams.
- ❒ Allows the user to build ARPA formatted language models.
- ❒ Allows the user to build LDC formatted language models.
- ❒ Can be used as the only tool for generating language models or can be used to build more sophisticated models.

REFERENCES

1. Rosenfeld, R. *Adaptive Statistical Language Modelling: A Maximum Entropy Approach*, Ph.D. thesis, Carnegie Mellon University, April 1994.
2. Lafferty, S. and Suhm, B., *Cluster Expansions and Iterative Scaling of Maximum Entropy Language Models*, in Fifteenth International Workshop on Maximum Entropy and Bayesian Methods, Kluwer Academic Publishers, 1995.
3. Koppleman, J., *A Statistical Approach to Language Modelling in the ATIS Domain*, MIT Department of Electrical Engineering and Computer Science, January, 1995.
4. Deller, J. R., Proakis, J. G., and Hansen, J. H.L., *Discrete-Time Processing of Speech Signals*. Macmillan Publishing Co., New York, 1993.
5. Rudnicky, A. T. and Hauptmann, A. G. *Survey of Current Speech Technology.* School of Computer Science Carnegie Mellon University, Pittsburg PA, 1993.
6. Lafferty, J., Sleator, D., and Temperley, D. *Grammatical Trigrams: A Probablistic Model of Link Grammar.* presented to 1992 AAAI Fall Symposium on Probablistic Approaches to Natural Language.
7. Gillett, J., Lafferty, J., Pietra, S. D., Pietra, V. D., Printz, H., and Ures, L. *Inference and Estimation of a Long-Range Trigram Model.* IBM, T.J. Watson Research Center, Yorktown Heights, NY.
8. Carter, D. *Improving Language Models by Clustering Training Sentences.* SRI International, 1994.
9. Segal, J., Stolcke, A. *Precise n-gram Probabilities from Stochastic Context-free Grammars.* International Computer Science Institute, Berkeley, California, appeared in ACL-94.
10. Omohundro, S. M., Andreas, S. *Best-first Merging for Hidden Markov Model Induction.* International Computer Science Institute, Berkeley, California, 1994.
11. Kita, K. *A Study on Language Modeling for Speech Recognition.* KITA Laboratories, 1992.
12. http://sls-www.les.mit.edu/SLSpubs.html
13. http://www.cs.cmu.edu/
14. http://www.ri.cmu.edu/
15. http://www.mambo.ucsc.edu
16. http://www.speech.cs.cmu.edu/
17. http://www.ll.mit.edu