Journal of Computer Science 2 (7): 577-582, 2006
ISSN 1549-3636

Performance Comparison of Sub Phonetic Model with Input Signal Processing

[1]Dr.E.Ramaraj and [2]E. Chandra,
[1]SG Lecturer, Department of Computer Science and Engineering, Alagappa University, Karaikudi
[2]Senior Lecturer, Department of Computer Applications,
D.J.Academy for Managerial Excellence, Coimbatore, India

Abstract: The quest to arrive at a better model for signal transformation for speech has resulted in striving to develop better signal representations and algorithm. The article explores the word model which is a concatenation of state dependent senones as an alternate for phoneme. The Research Work has an objective of involving the senone with the Input signal processing an algorithm which has been tried with phoneme and has been quite successful and try to compare the performance of senone with ISP and Phoneme with ISP and supply the result analysis. The research model has taken the SPHINX IV[4] speech engine for its implementation owing to its flexibility to the new algorithm, robustness and performance consideration.

Key words: Signal transformation, speech engine, senonic baseform, new algorithm

INTRODUCTION

HMM signal representations: In statistically based automatic speech recognition, the speech waveform is sampled at a rate between 6.6 kHz and 20 kHz and processed to produce a new representation as a sequence of vectors containing values of what are generally called *parameters*. The vectors ($y(t)$ in the notation used in section) typically comprise between 10 and 20 parameters and are usually computed every 10 or 20 msec. These parameter values are then used in succeeding stages in the estimation of the probability that the portion of waveform just analyzed corresponds to a particular phonetic event that occurs in the phone-sized or whole-word reference unit being hypothesized. In practice, the representation and the probability estimation interact strongly: what one person sees as part of the representation another may see as part of the probability estimation process. For most systems, though, we can apply the criterion that if a process is applied to all speech it is part of the representation, while if its application is contingent on the phonetic hypothesis being tested it is part of the later matching stage.

Representations aim to preserve the information needed to determine the phonetic identity of a portion of speech while being as impervious as possible to factors such as speaker differences, effects introduced by communications channels and paralinguistic factors such as the emotional state of the speaker. They also aim to be as compact as possible.

The language representation: A speech recognizer converts the observed acoustic signal into the corresponding orthographic representation of the spoken sentence. The recognizer chooses its guess from a finite vocabulary of *words* that can be recognized. For simplicity, we assume that a word is uniquely identified by its spelling.

Dramatic progress has been demonstrated in solving the speech recognition problem via the use of a statistical model of the joint distribution $p(W,O)$ of the sequence of spoken words W and the corresponding observed sequence of acoustic information O. This approach, pioneered by the IBM Continuous Speech Recognition group, is called the *source-channel model*. In this approach, the speech recognizer determines an estimate $\hat{W}$ of the identity of the spoken word sequence from the observed acoustic evidence O by using the a posteriori distribution $p(O|W)$. To minimize its error rate, the recognizer chooses that word sequence that maximizes the a posteriori distribution:

$$\hat{W} = \underset{w}{\operatorname{argmax}}\, p(W|O) = \underset{w}{\operatorname{argmax}}\, \frac{p(W)\, p(O|W)}{p(O)}$$

where $p(W)$ is the probability of the sequence of n-words W and $p(O|W)$ is the probability of observing the acoustic evidence O when the sequence W is spoken. The a priori distribution $p(W)$ of what words might be spoken (the source) is referred to as a language model (LM). The observation probability model $p(O|W)$ (the channel) is called the acoustic model. We discuss in this section, various approaches and issues for building the language model.

The source-channel model has also been used in optical character recognition (OCR) where the observation sequence is the image of the printed characters, in handwriting recognition where the observation is the sequence of strokes on a tablet, or in machine translation (MT) where the observation is a sequence of words in one language and we represents

Corresponding Author: E. Chandra, D.J. Academy for Managerial Excellence, Coimbatore, India

the desired translation in another language. For all these applications, a language model is key. Therefore, the work on language modeling has a wide spectrum of applications.

Word models: Word models are able to capture within-word phonological variations since the same phone is assimilated by different models when it is realized differently in different words. For example, the expected acoustic realization of phone /T/ in *ten*, the flapped /T/ in *thirty* and the deleted /T/ in *twenty* are modeled by different parameters within the respective word models. Words are also the most natural units of speech because they are exactly what we want to recognize. For small vocabulary tasks, word models may be suitable since it is not difficult to collect enough utterance samples for each word. For large vocabulary speech recognition, word modeling becomes difficult because many repetitions of each word are needed to obtain reliable estimates. Moreover, acoustic examples of one word are exclusively used for training that particular word; no other word can benefit from them. Furthermore, when a new word is added to a system, there is no way to obtain a model for the new word without collecting utterance samples for this word.

Monophone models: To achieve parameter sharing across different words, subword units like phonetic models are often used because they are both trainable and sharable. Since there are only about 50 phones in English, phones can be sufficiently trained with just a few thousand sentences. Hence, unlike word models, monophone models have no training problem. Word models are formed by concatenating a sequence of phones, or networking a set of alternative phonetic pronunciations. Thus, different words that contain the same phone share the same phone model and thus utterances of one word provide training data for parts of the other words. A new word can be easily added to a system without collecting utterance samples. Despite these many advantages, monophone modeling assumes that a phone produced in one context is equivalent to the same phone in any other context. This is far from the truth . Although we may try to say each word as a concatenated sequence of independent phones, phones are not produced independently because our articulators cannot move instantaneously from one position to another. Thus, the realization of a phone is strongly affected by its surrounding phones.

Multi-phone models: One way to share parameters and at the same time keep detailed and consistent models is to use multi-phone units. Examples of these include syllables demisyllables and traditional diphones . Each of these models cover a sequence of phones that contain the most severe contextual effects. However, while the co-articulations in the middle portions of these units are well modeled, the beginning and ending portions are still susceptible to some contextual effects. Even when we define multiple-phone units by breaking word pronunciations at points where contextual effects are as small as possible, we find that the cost of the information we lose at these multiple-phone-unit boundaries is not recovered by the superior modeling of the middle portions . The large number of multi-phone models presents another considerable problem for large vocabulary tasks, as there are over 20,000 syllables in English. Although there are fewer syllable models than word models in a very large vocabulary, reliable estimation of these parameter values is still not possible with a typical amount of available data (20,000 sentences). Moreover, as with word models, there is no parameter sharing between different multi-phone models and thus adequate occurrences in the training data for every single unit must be guaranteed to obtain reliable estimates

Subphonetic models: With regard to choosing a speech unit as the basic modeling unit, we have reviewed word models, multi-phone models (including syllables), monophone and context-dependent phone models. IBM first proposed *senones* as front-end based subphonetic units in 1987. Meanwhile, Dragon presented subphonetic segments and PELs (phonetic elements) for constructing triphone or PIC (phoneme-in-context) model. Soon after, GEC-Marconi presented *phonicles*, or phonetic particles . In 1992 ATR InterpretingTelephony Research Laboratories in Japan proposed the hiddenMarkov network (HMnet) constructed by the SSS (Successive State Splitting) algorithm , which conceptually resembles our work but is very distinct in its implementation. Among the above work, senones and senones are probably most successful and well known.

Subphonetic modeling using senones: The subphonetic unit we investigate is the existing Markov state in phonetic HMMs transformed to senones for research purpose.

Generalization at a subphonetic level: The goal of generalized triphones—grouping acoustically similar triphones into a single model—is to reduce the number of free parameters in the system's acoustic models. However, since it clusters acoustic events at a relatively crude unit, an entire phoneme, it may render over generalized phonetic models. When two triphone HMMs are merged, all pairs of corresponding Markov-state information (the output distributions) are averaged. Partial sharing is not supported. One solution, which is explored in this thesis, is to analyze each subphonetic event carefully and group only at the subphonetic level in order to avoid over-generalization. We propose to extract the information associated with Markov states of phonetic models (the output distribution and/or the transition probabilities) and

locate similar ones. With this approach, it is possible for two triphones to share only some common states.

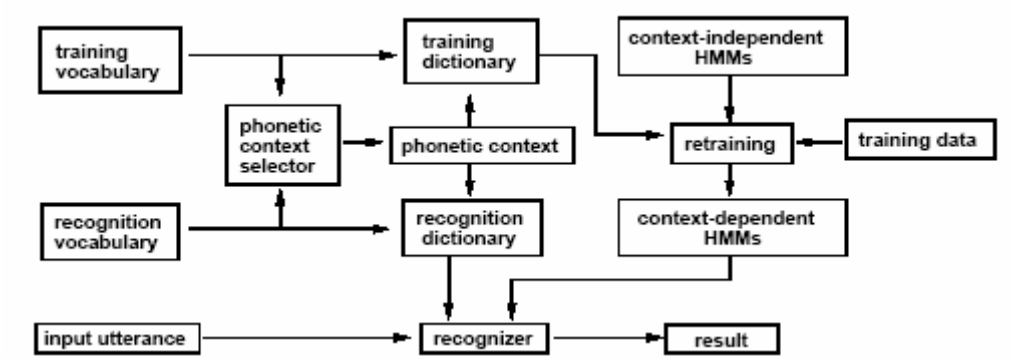


Fig. 1: Phonetic model

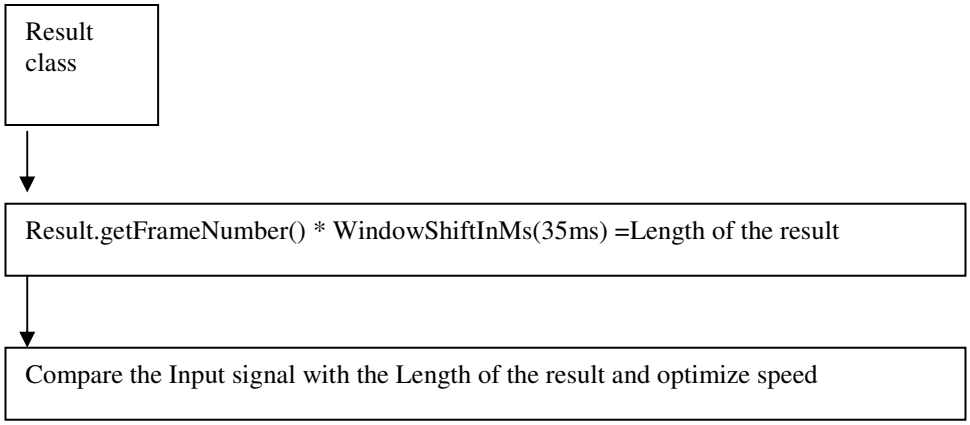


Fig. 2: ISP model

This flexibility results in accurate modeling at every state and at the same time achieves the goal of parameter reduction.

Conceptually, this searching of Markov state prototypes is similar to VQ clustering, where matches are sought among a set of prototype vectors. Implementation, output distributions and/or transition probabilities are probabilities rather than simple vectors. Therefore, their distortion measures (which make the notion of "match" precise) need to be defined differently. After clustering similar Markov states of triphone models, the original Markov states are labeled by the cluster identification, according to the clustering result. The resulting clusters are named senones, after senones, because they are state-dependent and are analogous to senones. The set of clusters becomes the contents of a "senone codebook". In SPHINX-IV, each state is identified by its associated output distribution (because many researchers share the same experience that transition probabilities play a minor role in the HMM methodology). Thus, a senone is essentially a representative of the output distributions associated with the Markov states in the same cluster. States labeled by the same cluster share the same senone (or interchangeably output distribution). The set of HMMs that share output distributions through senones are called shared-distribution models (SDMs). When senones are used, we have the luxury of using more states for each phonetic model to achieve more detailed acoustic modeling. We rely on the clustering procedure to tie redundant states. Given the sharing structure, we then train the HMM parameters so that the likelihood of generating the training data is maximized under the senone sharing constraint. The shared distribution model has proved to be significantly better than generalized triphones. For instance, on the 1000-word Resource Management speaker-independent continuous speech recognition task (using the standard bigram language model), the senone-based SDM outperformed generalized triphones by 19% in the word error rate.

Our work: One solution, which is explored in this thesis, is to analyze each subphonetic event carefully and group only at the subphonetic level in order to avoid over-generalization. We propose to extract the information associated with Markov states of phonetic models (the output distribution and/or the transition probabilities) and locate similar ones by either a bottom-up or top-down clustering procedure. With this approach, it is possible for two triphones to share only some common states. This flexibility results in accurate

modeling at every state and at the same time achieves the goal of parameter reduction.

The main goal of this thesis is to demonstrate the superiority of modeling subphonetic units over modeling phones. The subphonetic unit we investigate is the Markov state in phonetic HMMs. We compare the experiments did with phonetic model using ISP[11] a concept arrived by us for sphinx4 with the experiments we did for subphonetic model in this research work

Phonetic model with ISP: Our model considers sphinx4 with a new componet ISP. The ISP model[11] is proposed with the confidence that the input signal speed can be fine tuned for better realization by the Speech Engine (Sphinx). To explain it technically we configured the Result.getFrameNumber() function in the Result class by multipliying with the windowShiftInMs (a property of edu.cmu.sphinx. frontend.window.RaisedCosineWindower), which 10 milliseconds by default, to get the length of the result. We have also taken a standard reference to identify the silence as well as the speech based on several repeated experiments. we have also lowered the speech classifier 'threshold' property in the config files to make the input signal to be loud enough for the Sphinx engine to recognize. By using repeated experiment results we have set a standard for optimal speed for the input signal and configured the module (ISP) which identifies whether the speaker is speaking slowly or quickly and do the necessary performance modification to increase the efficiency.

In this experiment, we applied the concept of ISP only to the train data. The algorithm first estimated the phone segments in the testing utterance by running the decoder. It then used the hypothesized phone segments to find the sentence-based normalization factor, $\tilde{\rho}$Table 1 shows 16.5% error rate reduction by interpolating the cepstrum frames on *dev-fast* train data. The normalization factor ρ was determined by *AveragePeak* as defined by formula (2)The normalization factors of the utterances in *dev-fast* varied between 0.92 and 1.47

Table 1: Shows Error reduction by interpolating cepstral frame

Training data	Original	Interpolation
Original	16.64%	13.90%

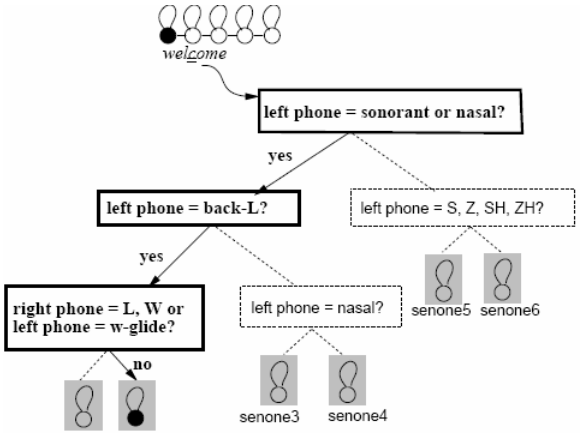


Fig. 3: Sub phonetic model

Subphonetic model with ISP: Senones were tested on a 2000-word office correspondence task in a speaker-dependent isolated-speech mode. Both the training and testing data sets each contained one utterance for each word in the vocabulary, from one speaker only. No language model was used. The word error rate using monophone models is shown in the first row of Table 2.1. In the second experiment, a third utterance (other than the training and testing utterances) was used to obtain the fenonic baseform according to the sequence of VQ codewords. The learned fenonic baseforms were trained by the training corpus which consists of one sample for each word in the vocabulary. The third experiment used another 4 utterances for each word to obtain the fenonic baseform according to (2.1). With the same training corpus, the senonic baseform learned from 4 separate utterances performed significantly better than monophone models.

Table 2: Word error rate with monophone and senone

Acoustic model	No.of utterances for Learning senonic baseforms	Word error rate
Monophone	0	5.2%
Senone	1	7.4%
Senone	4	2.2%

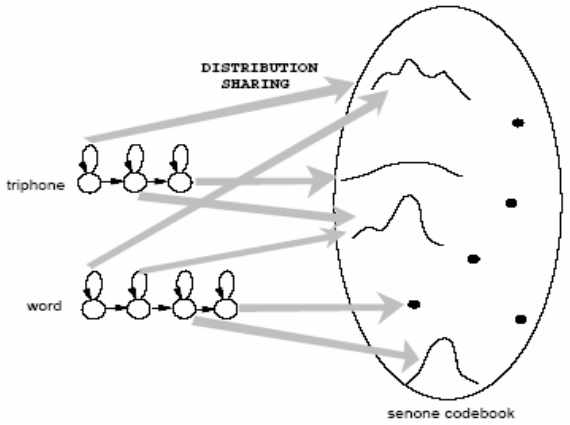


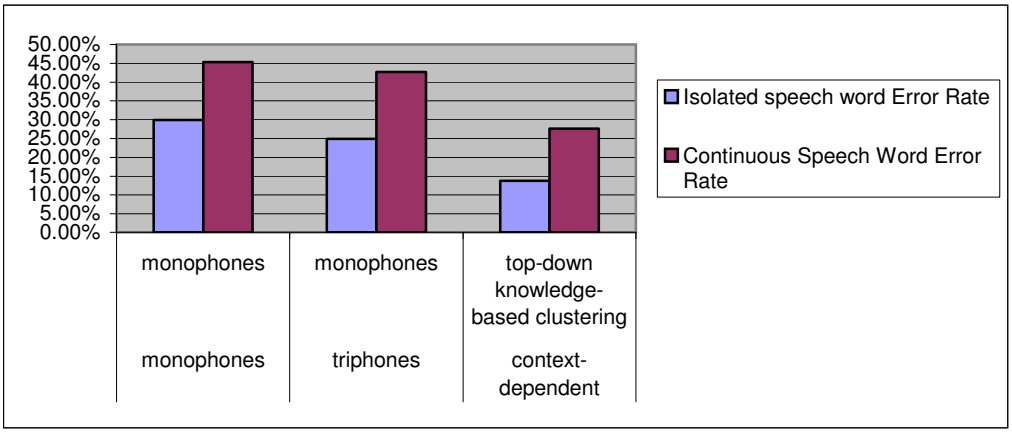
Fig. 4: Senone codebook

Given multiple utterances of a particular word, we can estimate a word HMM by the forward backward algorithm. For each state in the estimated word HMM, a most similar senone is identified to replace the estimated output distribution. This concept of state quantization is the same as the one in the hybrid approach, except that the SQ distance measure may be different. The Figure above explains the algorithm for constructing the senonic baseform for a word, given multiple utterances. The word model is allowed to be labeled by an arbitrary sequence of senones to provide the maximum freedom for automatically learned pronunciations reduced by 15% compared with the phonetic baseform.

The senonic baseform is applied to two tasks. The vocabulary for the first task contains only the 26 letters of the English alphabet.

Table 3: Comparison of results showing reduction in word error rate with the senones

Acoustic Model	Modeling of Unseen Contexts	Isolated Speech Word Error Rate	Continuous Speech Word Error Rate
Monophones	monophones	29.9%	45.3%
Triphones	monophones	24.9%	42.7%
Context-dependent Phonicles	top-down knowledge-based clustering	13.8%	27.6%



With a rich training corpus, each letter has ample training data and thus two senonic baseforms per letter are automatically learned in our experiments. With these purely acoustic-driven senonic baseforms, the word error rate is

1. Compute the average duration (number of time-frames) of the word, based on the given multiple utterance samples.
2. Build a Basis word HMM with the number of states equal to a portion of the average duration.
3. Run several iterations of the forward-backward algorithm on the word model, using the given utterance tokens.
4. Quantize each state of the estimated word model with the senone codebook.

The second task attempts to attack the problem of learning a good senonic baseform in large vocabulary tasks when there are few training tokens. Unfortunately, with few training tokens, the algorithm does not yield a highly-accurate senonic baseform, due to the high acoustic confusability implied in the vocabulary inventory. Interestingly, we find that the senonic baseform is indeed able to adapt to speakers, even with very few training samples.

COMPARISON OF RESULTS

The senone codebook has better acoustic resolution, as there are usually a few thousand senones in a system. In addition, when training examples are unavailable, phonetic baseforms can be used together with the senonic baseforms for other words without any increase in the system complexity, since both phonetic and senonic baseforms share the same set of senones. Each senonic word model, built by the algorithm described in Fig. 4 generally has more states than the traditional phone-concatenated word model and hence is capable of modeling more acoustic details. Although it usually has more states, the senonic word model never increases the number of free parameters in the system.

The senonic baseform offers 15% word error reduction over the phonetic baseform on a small vocabulary task. Applying the senonic baseform to large vocabulary tasks has not yet proven to be successful. However, it does show promise for speaker adaptation and speaker-dependent applications.

CONCLUSION

Further work, such as incorporating spelling information, should be done to improve the senonic baseform in order to take advantage of the adaptation capability. Another improvement would be to incorporate senone bigram models or other higher level models in the state quantization procedure

ACKNOWLEGMENT

Also I would like to thank the management of Sovereign Infotech India (P) Ltd., for helping me with the experimental procedures. I would like to thank Mr.P.Arthanareeswaran towards the Research support.

REFERENCES

1. http://www.hltcentral.org/page-1086.0.shtml
2. Manual of Microsoft Speech Server 2004.

3. cslu.cse.ogi.edu/HLTsurvey/ch1node7.html
4. cmusphinx.sourceforge.net/sphinx4
5. Lippmann, R.P.,(1997. Speech recognition by machines and humans. Speech Communication, 22: 1-15.
6. Pols, L.C.W., 1998b. Psycho-acoustics and speech perception. In: K. Ponting (Ed.), Computational Models of Speech Pattern Processing, Berlin: Springer Verlag, pp: 10-17.
7. Wieringen, A. van, 1995. Perceiving dynamic speechlike sounds. Psycho-acoustics and speech perception. Ph.D. Thesis, University of Amsterdam, pp: 256.
8. Richardson1, M., M. Hwang, A. Acero and X.D. Huang. Improvements on speech recognit1on for fast talkers Speech Technology Group Microsoft Research Redmond, Washington, 98052,USAhttp://research.microsoft.com/srg
9. Bakis, R., 1976. Continuous Speech Recognition via Centisecond Acoustic States. 91st Meeting of the Acoustical Society of America.
10. Martinez F., D. Tapias and J. Alvarez, 1998. Towards speech rate independence in large vocabulary continuous speech recognition. IEEE Intl. Conf. Acoustics, Speech and Signal Processing, Seattle, pp: 725-728.
11. Dr.E. Ramaraj & Chandra, E., 2005. Influence of Acoustics in Speech Recognition for Oriental Language. Accepted by Intl. J. Commun. Process. Oriental Language.