International Journal of Soft Computing 2 (1): 88-95, 2007

Speech Recognition Standard Procedures, Error Recognition and Repair Strategies

[1]E. Ramaraj and [2]E. Chandra
[1]Department of Computer Science and Engineering, Alagappa University, karaikudi
[2]Department of Computer Applications, DJ Academy for Managerial Excellence, Coimbatore

Abstract: The accuracy of input speech signal is very important as speech signal is now considered as an input source for several business domains.. This study attempts to propose a usability standard procedure for speech recognition. It also proposed an algorithm to investigate the cause for errors in recognition by including several related research works in this area and suggest an error recognition and repair procedures. The research suggests that it is practically possible to predict the misrecognized utterances with a high degree of accuracy from an utterance's sound file, the language model being employed and recognizer outputs such as confidence, In addition, there is empirical data upon which to base successful repair strategies in relation to these misrecognitions.

Key words: Recoginition, procedures, startegies, errors

INTRODUCTION

As spoken language systems are being used more often for a wide range of applications, there is an increasing demand for standards to test and compare performance and usability. Most usability studies do not evaluate the system beyond speech recognition accuracy, adequate language modeling and appropriate semantic representations for efficient interaction with back-end knowledge sources[1-3]. Significant progress has been made towards identifying standards to achieve this goal. A wide variety of measures have been used, including measures like task success. Other metrics evaluate user satisfaction in conjunction with task success.

The research study propose a usability standard based on three factors

- Accepting the speech signal in an optimal way.
- Assessing the Speech recognition task success.
- Assessing the user satisfaction in conjunction with the task success.

The research proposed a procedure for accepting the speech signal based on Input Signal processing[4] which identifies spoken word for validating fast and slow. The research also measure task success and also user satisfaction in conjunction with task success . User satisfaction was calculated using questionnaires. Interaction between the user and the system was recorded to calculate the remaining two metrics.

The research also proposed an efficient method to identify errors in recognition and repair procedures. In real time speech recognition application typically, where the confidence level is low, systems will reject the recognition and reprompt. If the system has a hypothesis, but is unsure as to its correctness, a confirmatory question is asked. Both strategies can be very frustrating to the user if they are used repeatedly. More sophisticated systems might proceed with an implicit confirmation, as in Example.

In these study, the system also has to allow for the user's protest when recognizing the next response and to negotiate an appropriate correction of the error:

System: OK, that's flying to kolkatta?.
Where are you departing from?

User: No, I said Coimbatore!

System: Sorry, I must have misheard.
Where did you want to fly to?

However, it seems errors revealed by implicit confirmation take longer to repair than those handled by explicit confirmation[5] and in reality, very few real systems exhibit such sophistication; error handling and repair strategies adopted are generally quite simplistic and sometimes poorly designed. The focus of our study is handling misrecognitions by solving two problems:

Error recognition: to classify hypotheses as correct or not, with a very high level of accuracy. A subvector-based error concealment algorithm for speech recognition over mobile networks by[6].

Corresponding Author: E. Ramaraj, Department of Computer Science and Engineering, Alagappa University, karaikudi

Error repair: To repair such errors in a manner that does not frustrate or baffle the user.Not much research reference is available in this area.

RELATED WORK

Standards for speech recognition

Evaluating user interfaces: Every designer wants to build a high-quality interactive system that is admired by colleagues, celebrated by users, circulated widely and imitated frequently[7]. A necessary aspect of designing such an interface is evaluation. Evaluating user interfaces is becoming one of the most important aspects of software development. There are many factors that determine whether an interface is good or bad. Some of them are as follows

- Functionality
- Speed and efficiency
- Reliability, security, data integrity
- Standardization, consistency
- Usability

In evaluating an interface, the most significant criterion is the usability of the interface. Interactive applications require the end users to comprehend the interface, so that they may complete the tasks at hand. An interface will be inherently usable if all phases of the design process are user-centered. The system should be designed so that it is tailored to the needs and requirements of the user and not the other way around[1].

However, there are applications that are mission critical that do not emphasize user satisfaction. These interfaces require training and typically have minimum performance requirements. There are several methods used to evaluate the usability of an interface. Formative evaluation is one of them. It is an integral part of prototyping the system. It involves the collection of subjective, objective, quantitative and qualitative data, which is later analyzed. The results are used to fine tune the interface in as part of an iterative cycle. Graphical user interfaces have a well-defined set of methods that can be used to make an interface more user-friendly. Some of the major advances in this area are made using style guides. A systematic testing involving actual user is still one of the most reliable methodologies for developing user friendly software systems. Human short term memory is limited to seven plus or minus two items[8]. This is a very important factor in spoken language systems design. Graphical user interfaces are persistent over time. When a user is viewing a graphical user interface, s/he can page down or walk away from the interface and the data will persist on the screen.However, the same evaluation techniques cannot be applied to spoken language systems. Evaluating a spoken language system requires a different set of considerations[1]. Has developed an Input signal processing for adjusting the spoken words for accuracy Fig. 1.

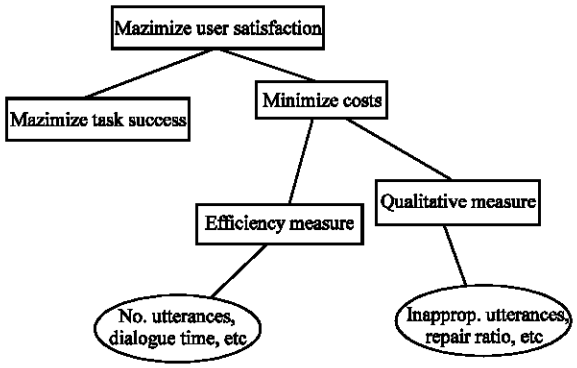


Fig. 1: Standards for speech recognition

The causes of errors: Speech engines produce hypotheses by seeking to find the best word sequence for a given speech input that maximizes the the words given some language model[8]. Speech recognition errors occur because the sounds in the utterance heard by the computer are dissimilar to its acoustic model and/or the language employed is not contained in the language model being used.combined probability of the sounds being from the proposed words and the words being

The acoustic domain: The literature shows that there are many potential sources of difference in the acoustic domain.

These include hyperarticulation, pronunciation variation, cold speech, dysarthric speech, children's speech and noise in the signal ,[L.Rabiner and B.Juang, 1993 and many more]. Shriberg and Stolcke [2001] represent a particularly good exposition of the method of automatically discovering errors using prosody; a corpus can be hand tagged for correct and false recognitions and features, such as speaking rate, intensity, pitch and durations of silence can be automatically extracted from the sound files of the associated utterances. Some machine learning technique can then be employed using the tagged data and features as a training set. Hirschberg *et al.* [2004] reports on two studies of this nature using RIPPER [Cohen, 1996]. It is normal, in these studies, to include automatic speech recognition (*ASR*) outputs such as confidence, but excluding such features they managed to achieve between 74 and 80% correct prediction that an utterance was likely to be misrecognized in two corpora.

The language domain: Work has been published based upon two useful approaches that can help catch out-of-language (*OOL*) utterances:

The two recognizer technique: By using a grammar-based recognizer, designed to capture the language the system can recognize and understand and an n-gram recognizer, designed to capture all speech, one can identify OOL utterances for a system. An example of this can be seen in[10] where it was used to significantly improve the help the system gave when users employed OOL utterances[11,12]. Has used dual recognizer technique to vary between fast and slow speech based on speech rate.

Sub-word language modeling: By designing a recognizer that is based on language units lower than words (say phonemes) one can potentially produce a speech recognizer not tied to any size of vocabulary, or form of syntax[13] used known words plus an OOL word that modeled sequences of phones[14,15] used subphonetic model for comparing the accuracy] took the approach of using an algorithm to discover sub-word units called morphs across their training set and used these units for their language modeling[13] found no material increase in misrecognition of in-language words when using the technique and correctly classified half of the OOL words[14]; confirmed these results.

The repair of errors: Much of the work on the repair of errors is highly domain specific. However, a ubiquitous resource that is always available and who knows what has been said, is the user of the system.

There is literature concerning the way users react to errors[16] noted similar strategies[17] carried out the same study as Shin *et al*., but on a commercial corpus and showed remarkable similarities in users' reactions. In the utterance following the system revealing an error, users either rephrased their last utterance, repeated it, contradicted the system, or changed their request in very similar proportions, even though the systems studied came from very different domains. All of this study supports the view that there is a consistency in user behavior across domains, that will allow the development of reliable repair strategies Table 1.

Table 1: Corpora comparison

Featrue	Pizza	Communicator
Language model	Grammar	N-gram
Acoustic model	Australian english	American english
Dialogues	2486	2334
Utterances	32728	40522
Words	54740	106562
Vocabulary size	1048	2367
Error rate	19%	36.65%

STANDARDS FOR SPEECH RECOGNITION

Evaluation is a prerequisite to designing effective and natural interfaces. The methods used to evaluate graphical user interfaces are well established and numerous. In order to design effective and natural spoken language systems, an iterative process is used. This includes cycles of design, implementation, experimentation with users and evaluation, followed by redesign and implementation of improvements, based on the results of the user tests. When conducting this iterative process, when is the speech interface good enough? What metrics determine success? Without answers to these questions, this iterative process could go on indefinitely and become very costly. Therefore, the evaluator must have some concept of when the interface is good enough to end user testing. The Speech Usability Metric (SUM)designed such that the designer can specify metrics that are relevant to the interface and specify metric goals that will determine when the interface is good enough. The SUM is stated as follows:

SPEECH USABILITY METRIC = X * (USER SATISFACTION)+Y *(ACCURACY)+Z *(TASK COMPLETION TIME) Where X+Y+Z>0 and X, Y, Z>0.

User satisfaction, accuracy and task completion time. The weights are specified by the designer with respect to each metric's importance.

User satisfaction is typically the most heavily weighted of the three metrics. Typically, user satisfaction is measured using surveys and/or interviews on a likert scale.

The acoustic domain: Most of the published work concentrates on the features in speech associated with hyper articulation. The research takes a more general position, with the idea that any mismatch between the acoustic nature of the training set and the utterance to be recognized, will cause problems. Therefore, the research looks at a wide range of features and exploring the relationships between these features and errors.

The language domain: The research needs some mechanism to identify when errors due to OOL language occur and some sort of metric relating to how far OOL an utterance is. The two recognizer technique may assist us in this. Additionally, the work wishes to capture the principals that lie behind the language employed by users when they are protesting errors introduced by misrecognition. Such utterances are often OOL and the

work suspect that the language of protest can be mapped from the language that was suitable to handle the focus of the dialogue during which the error occurred.

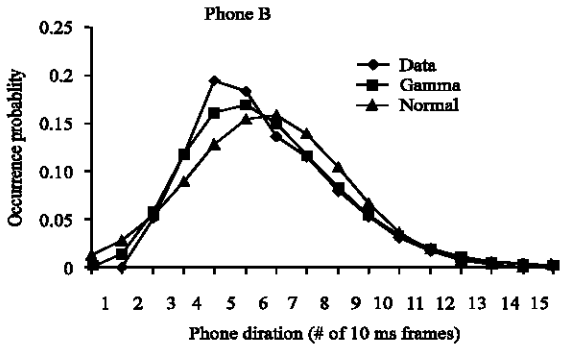
Error repair: The research suggests the need to know the clues that the system should give to the user when an error is suspected and how we should manage the dialogue, to repair the error.

INPUT SIGNAL PROCESSING MODEL

The ISP model[17] 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 ms by default, to get the length of the result. Since the research believe the largest improvement on the recognition of fast speech is in the better match with the SI phone duration and dynamic features trained from regular speaking rate speech, we use a smaller window shift in generating the cepstrum., the speed factor ? is first computed as formula outlines with the default window shift (in our case 10 ms). Then a new window shift is computed as inversely proportional to

$$s' = \frac{s}{\rho}$$

A Railway helpdesk application based on the HMM and sphinx was built and incorporated the ISP model into it. The scope includes helping the railway help line for customer frequent queries on the Train, Platform number, Departure time, Route, frequency in a week etc., The application showed positive results after incorporating the ISP Module.



Experimental setup: The following clearly depicts the experimental setup.

The application was developed for Railway Help desk using Sphinx-IV (Pure Java). The application will recognise number and some words like train number, platform, station, train name, destination and genarates result. With ISP mechanism we will trace the speech rate by identifying the time taken for the utterance and compare with the optimatal speech rate(30-35ms) which is taken as the standard speech rate after repeated experiments with the model. When the speech rate value exceeds, the application invoke optimization methods to acquire the best results out of the utterance. The following are the results.

Based on our experiment, speaking rates are defined by the length of an individual phone relative to its average duration in the SI-284 training corpus. The research has used SI-284 corpus as it best suits our research of analyzing speech rate for speaker independent scope We feel that a simple measure of the number of phones per second is not informative enough and that some knowledge or estimation of the phones that were uttered will improve thee estimation of real speaking rates.

Rate-dependent acoustic modeling: In our proposed method, each word is given parallel fast-and slow-version pronunciations in the recognition lexicon. Both fast-and slow-version pronunciations are initialized from the original rate-independent versions. This has a simple replacement of rate-independent phonemes by rate-specific phonemes. For example, the original rate-independent pronunciation of WORD is /w er d/. Consequently the fast-version pronunciation is /w_f er_f d_f/ and the slow-version /w_s er_s d_s/, consisting of fast and slow phonemes, respectively. The recognizer in this case the normal forced alignment modes[8] automatically find the best pronunciations that maximize the likelihood score during the search and thus avoid the need for ROS estimation before recognition. In addition, the search algorithm, Breadth first searching[8] is allowed to select pronunciations of different rates across word boundaries and thus can cope with the problem of speech rate variation within a sentence.

Acoustic training: The experiment is based on Sphinx4 system, which uses continuous-density genomic hidden Markov models (HMMs)[7] the application is configured with only the first-pass recognizer based on gender-dependent non-crossword genonic HMMs (1730 genones with 64 Gaussians each for male, 1458 genones for female) and a bigram grammar with a 33, 275-word vocabulary. The recognition lexicon was derived from the CMU V0.4

lexicon with stress information stripped. The recognizer used a two-pass (forward pass and backward pass) Viterbi beam search algorithm; in the first pass a lexical tree was used in the grammar backoff node to speed up search. The report results from the backward pass. The features used were 9 cepstral coefficients (C1-C8 plus C0) with their first-and second-order derivatives in 10 ms time frames. The research first calculated the ROS for all the words in the training corpus based on the above-mentioned measure, sorted these words accordingly and then split them into two categories: fast and slow. The ROS threshold for splitting was selected to achieve equal amounts of training data for the fast and the slow speech. The training transcriptions were labeled accordingly. In this way, the research was able to train the fast and slow models simultaneously. The research used genonic training tools to do standard MLE (Maximum Likelihood Estimation) gender-dependent training[4] and obtained rate-dependent models with 3233 genones for male and 2501 genones for female. The genone clustering for rate-dependent models used the same information loss threshold as the training of rate-independent models. The research compared the rate-dependent acoustic model with the rate-independent acoustic model (baseline system) on a development data set, which is a subset of the Sphinx4 data set, consisting of 1143 sentences from 20 speakers (9 male, 11 female). Table 2 shows the word error rate (WER) for both models.

Rate-dependent modeling brings an absolute WER reduction of 1.9%, which is statistically significant. To eliminate the possible effect of different numbers of parameters, there was an adjustment in the information loss threshold for genone clustering to obtain another rate-independent model that had a number of parameters similar to that of the rate-dependent model.

Adaptation vs standard training: In our previous study based on the ISP[8], instead of using the training method proposed here, the research trained the rate-dependent model based on Speech rate as the major influencer. However, in the current task of speech transcription the research had significantly more training data and the research use a different strategy to partition the data into two classes instead of three, yielding more training data for each rate class. In addition, the optimal models the research started with were smaller. Thus, the research was able to train the rate-dependent models robustly with standard training methods. For comparison the research tested the Bayesian adaptation approach[4] on the current training set. Similar to[4], even though the research has

Table 2: WER for ISP and without ISP

Corpus	Description	Vocabuln. size	Recogn. perplex	(%) Word error ISP	(%) Word error Without ISP
TI digits	read digits	15	15	0.76	0.8588
Alphabet	Read letters	28	28	5.5	5.77

Table 3: WER comparison between the baseline system with rate-independent model and the system with rate-dependent model on the development data set

Model	Male	Female	All
Rate-independent model	55.6	64.5	58.9
Rate-dependent model from training	51.7	61.7	58.9

used separate rate-specific models for each triphone, the research has not created separate copies of the genones, but let the fast and slow models for a given triphone share the same genone. In this way, the same number of Gaussians was used for the rate-dependent model as for the rate-independent model. Table 2 shows the results on the same development data set used in the previous section. This approach brings an advantage of 1.0% over the baseline, less than the standard training scheme. This indicates that the difference between fast and slow speech in the acoustic space is significant and that standard training might be better than the previous adaptation scheme to capture this difference. These differences might explain why the adaptation scheme did not achieve as much improvement as the standard training.

Experimenal setup:

WER			
WER of baseline system	44.3	53.3	47.3
WER of rate-dependent system	43.6	53.0	46.8

The experiment used Sphinx-IV in a windows 2000 platform with the following configurations, Viterbi algorithm based HMM and uses Flat structured viterbi search for decoding and continuous density acoustic mode and ASCII simple N-gram model and uses Breadth first search and the baseline system had been enhanced substantially. The research supply some minimal pair experiments based on different baseline systems during the development process. The baseline system in Table 3 used a wider-band front end (with 13 cepstral coefficients instead of 9) and Vocal Tract Length (VTL) normalization[8] during training. The success from introducing word-level rate dependency is still 1.9%, over a baseline that was itself improved by 5.0%.

Another major addition to the evaluation system was the introduction of multiword pronunciations. Here a multiword is a high-frequency word bigram or trigram, such as "a lot of", that is handled as a single word in the

Table 4: WER comparison between the baseline system with rate-independent model and the system with rate-dependent model from adaptation on the development set

Model	Male	Female	All
Rate-independent model	55.3	63.4	59.8
Rate-dependent model from adaptation	54.0	62.6	58.8

vocabulary. By using handcrafted phonetic pronunciations describing various kinds of pronunciation reduction phenomena for these multiwords, the work achieved better modeling of crossword coarticulation. In Sphinx4 system 1200 multiwords were introduced. Experiments showed that the multiword pronunciation modeling brought about a 4.0% absolute win on top of the improved baseline system in Table 3,[8]. The possible reasons for the diminished effectiveness of ROS modeling may lie in the following aspects. First, each multiword is given multiple parallel pronunciations reflecting both full and reduced forms. This by itself models fast and slow speech variants to some extent. words, the work fail to model the rate variation occurring within the multiwords and thus may influence the quality of the rate-dependent acoustic models. Third, due to our current implementation, the introduction of multiwords made the search much more expensive than before; rate-dependent modeling on top of the multiword dictionary made this problem even worse and may have produced a loss in performance due to search pruning. Based on the above analysis, another scheme was tested: instead of treating multiwords as ordinary words the research trained them with multiword-specific phoneme units, that is, using separate phonetic models to describe the multiwords. Similar to the original approach, trained three classes of phoneme models simultaneously: fast models for ordinary words, slow models for ordinary words and a separate set of phone models trained only on the multiword data. With this approach, the research improved the WER reduction to 0.7%, as shown in Table 5.

WER	Male	Female	All
WER of baseline system	44.3	53.3	49.3
WER of rate-dependent system		43.6	52.6 48.6

Table 6: Minimal pair comparison on the development set between the multiword-augmented baseline system and the rate-dependent system with multiword-specific phone models

Sub word modeling: 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

Table 5: Minimal pair comparison based on an improved baseline system using a wider front end and VTL normalization on the development set

WER	Male	Female	All
WER of baseline system	50.6	57.9	54.6
WER of rate-dependent system	49.2	55.6	52.7

Table 6: Minimal pair comparison based on a multiword-augmented baseline system on the development set

WER	Male	Female	All
WER of baseline system	44.3	53.3	47.3
WER of rate-dependent	43.6	53.0	46.8

Table 7: Shows Error reduction by interpolating cepstral frame

Training data	Original interpolation	
Original	16.64%	13.90%

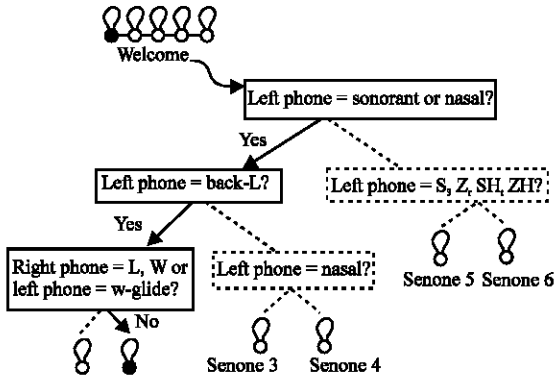


Fig 2: Sub phonetic model

ISP[17] a concept arrived by us for sphinx4 with the experiments we did for subphonetic model in this research work .

Phonetic model with ISP: 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, ρ 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

Subphonetic model with ISP: Senones were tested on a 2000-word ffice 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. In the second experiment, a third utterance (other than the

Table 8: Word error rate with monophone and senone

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

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.

Error reduction strategies: The application propose a standard learning library which can identify the real time exceptions and provides an option to update each time the recognition has gone wrong based on sentence confirmation. The application later uses this to frame new rules to assist better mechanism. The application also model user satisfaction and user confirmation for tasks and design a learning model.

CONCLUSION

The thesis had an objective to identify and reduce error during speech recognition and provide better mechanism for Improved Recognition. Much of the Utterance related problems could be solved using the ISP model suggested by the thesis which shows a significant Improvement of 85% in terms of WER. Also the practical aspects of Fast and slow speech could be reduced using the dual algorithm for recognition suggested by the thesis which also provides a better solution. Also the thesis has implemented the language unit lower than phonemes to improve the performance of speech recognition. Also the thesis suggests error recognition procedures based on learning.

REFERENCE

1. Marilyn Walker, D.H., J. Fromer, G. Di Fabbrizio and C. Mestel, 1997. EVALUATING COMETING AGENT STRATEGIES FOR A VOICE EMAIL AGENT. Proc. Eurospeech '97, Rhodes, Greece, pp: 2219-2222. [Online] available: http://www.research.att.com/~walker/elvis/elvis.html.
2. Estimation of the Short-Term Predictor Parameters of Speech Under Noisy Conditions IEEE Transactions on Audio, Speech and Language Processing, ISSN,: 1558-7916 Digital Object Identifier: 10.1109/TSA.2005.858558.
3. On the importance of phase in human speech recognition IEEE Transactions on Audio, Speech and Language Processing Accepted for future publication ISSN, pp: 1558-7916, Digital Object Identifier: 10.1109/TSA.2005.858512.
4. Ramaraj, E. and Ms.E. Chandra, 2005. Influence of Acoustics in Speech Recognition for Oriental Language published by International journal of computer processing of oriental languages, World Scientific Publishing.
5. Shin, S., L. Narayanan, Gerber, Kazemzadeh and D. Byrd, 2002. Analysis of User Behavior under Error Conditions in Spoken Dialog. In *Proc. ICSLP*, Denver, Colorado, pp: 2069-2072.
6. A SUBVECTOR-BASED ERROR CONCEALMENT ALGORITHM FOR SPEECH RECOGNITION OVER MOBILE NETWORKS *Zheng-Hua Tan, Paul Dalsgaard and Børge Lindberg* {zt, pd, bli}@kom.auc.dk SMC-Speech and Multimedia Communication, Department of Communication Technology1, Aalborg University, Denmark
7. B.S., 1997. Clarifying Search A user-Interface Framework Text Searches, D-Lib Magazine, ISSN HeyAnita.com [Online] Available: http://heyanita.com , pp: 1082-9873.
8. Lenzo, K.A. and C.M.U. Sphinx. Open Source Speech Recognition. [Online]. Available: http://www.speech.cs.cmu.edu/speech/sphinx/.2002.
9. Nanjo, H., A. Lee and T. Kawahara, 2000. Automatic diagnosis of recognition errors in large vocabulary continuous speech recognition systems. In ICSLP., 2: 1027-1030, 2000.
10. Gorrell, I. L. and M. Rayner, 2002. Adding intelligent help to mixed-initiative spoken dialogue systems. ICSLP.
11. Ramaraj, E. and E. Chandra, 2006. Performance Adjustment of Speech Rate for ASR, presented in International conference on Emerging applications of IT conducted by CSI kolkata, published in the conference proceedings.
12. Ramaraj, E. and E. Chandra, Voice Print identification-a secure speech algorithm, published by International journal of systemics, Cybernetics and Informatics(ICSCI) Pentagram Research Centre, Hydrabad.

13. Bazzi, J.R.G.,2000. Modeling out-ofvocabulary words for robust speech recognition. ICSLP, pp: 401-404.
14. Siivola, T.H.A., M. Creuts and M. Kurimo, 2003. Unlimited Vocabulary Speech Recognition Based on Morphs Discovered in an Unsupervised Manner, Geneva,. EUROSPEECH, pp: 2293-2296.
15. Ramaraj, E. and E.Chandra. Performance comparison of sub phonetic model with input signal processing, published by Journal of Computer Science,New York Science Publishing, USA , 7: 577-582 .
16. Stifelman, 1993. User Repairs of Speech Recognition Errors: An Intonational Analysis. Technical report, Speech research Group, MIT Media Laboratory.
17. Choularton, R.D., 2004. User responses to speech recognition errors: Consistency of behaviour across domains. Sydney, Australian Language Technology Workshop, ASSTA.