6

SYSTEM EVALUATION

CONTENTS

OBJECTIVE OF THE CHAPTER

The objective of this chapter is to evaluate the proposed system performance in terms of accuracy and proposed some mechanism to increase the performance of the base line system. The confidence building measure is done by applying multiple decoders and voice activity detection. Further, the issues in adapting the ASR system with small and unseen speaker data are highlighted. A simple grid search based cluster model interpolation for model mean and/or mixture-weight adaptation is explored which provides about 11% relative improvement in baseline performance.

6.1 CONFIDENCE BUILDING MEASURES

Confidence measures [69] are mainly used to detect incorrect words, ambiguity of words to provide an estimation of the probability of correctness. It helps to go with a proper flow in the dialogue manager reduce to probability errors that may happen during information retrieval process. There are three different levels of confidence measure in spoken dialogue system [70]:

a. Word Level gives the idea about the accuracy of each recognized word by using Language Model (LM) features.
b. Utterance Level targets the detection of out of domain utterances by using acoustic, LM and parsing features.
c. Concept Level focuses on parts of phrases which is meaningful to the task. Decoder, LM and parsing features are used to tag the concepts with the confidence measures.

We consider decoder and LM features for word level confidence measures. Decoder features has the following parameters [70-71]:

a. Normalized Score: It is defined as the total acoustic score of the word divided by the number of frames that it spans.
b. Count in the Nbest: It is the percentage of times that the word appears in the 100-best hypotheses in similar position.

c. Lattice Density: It is the number of alternative paths to the word considered in the word-graph generated in the second pass of the recognizer.

d. Phone Perplexity: It is the average number of phones searched along the frames where the recognized word has been active in the decoding process.

Language Model features has the following parameters:

a. Language Model Back-Off Behavior: back-off behavior of an N-gram language model along a 5 word context.

b. Language Model Score; the log-probability for each word in a sequence as computed from a back-off language model along a 5 word context.

Following mechanisms are adopted to avoid a wrong path in the call-flow of IVR module:

a. The feeble and no-responses are detected using a VAD and the system then prompts to repeat the query.

b. The final hypothesis is generated by polling the output of acoustic models

c. At each stage of the call-flow, the user is asked for confirmation by uttering Nongwo / Nonga(yes/no).

6.2 PROPOSED VAD ALGORITHM

A simple voice activity detector which works on short term energy is used to detect whether the user has uttered a word or not. The average energy is computed for three different regions in the speech file. Where, three different regions being classified as initial silence, middle speech and the final silence regions. Since, the application works on isolated word recognizer the initial and final 0.9 seconds of speech data is being considered as silence regions. The silence energy is considered to be from either the initial or end regions of the speech based on lesser average energy criteria. Once the average energy of the silence (Silence Energy) is computed, this is used to find whether there is reliable speech activity in the speech based on the following relationship

```
If (Average Energy < (4 * Silence Energy))
        Voice Activity = yes
else
        Voice Activity = no
```

i.e., if the average energy of the speech region is at least 4 times greater than the average energy of the silence region then it is assumed that there is voice activity in the speech file else it is assumed there is no voice activity in the speech file. The speech region is detected by using the endpoint detection algorithm. The endpoint detection algorithm works on the basis of lower ($\mathbf{T_l}$) and upper threshold ($\mathbf{T_u}$) values of the evidence. The $\mathbf{T_l}$and $\mathbf{T_u}$are computed using the maximum value of the average energy ($\mathbf{E_{max}}$) and the minimum value of the average energy ($\mathbf{E_{min}}$) of the speech signal. The following relations are used to obtain $\mathbf{T_l}$ and $\mathbf{T_u}$:

$$\mathbf{I_1 = 0.03 * (E_{max} - E_{min}) + E_{min}}$$

$$\mathbf{I_2 = 4 * E_{min}}$$

$$\mathbf{T_l = min(I_1, I_2)}$$

$$\mathbf{T_u = 2 * T_l}$$

The starting points for searching the speech regions start from the extreme endpoints on either side. These points are labeled as the beginning and end points of the speech region. The end points are moved towards the center from either side towards the center unless the energy falls below $\mathbf{T_l}$ before it rises above $\mathbf{T_u}$. These points obtained finally are determined as the endpoints of the speech region. If there is a voice activity detected then the speech file is further subjected to the speech decoder for recognition of word.

6.3 PROPOSED HYPOTHESIS FOR MULTIPLE DECODERS

We propose a confidence scoring technique based on multiple ASR decoders. The key idea is that one may want to build more than one ASR decoders, where each decoder tries to capture complementary facts approximately the speech data. This could be attempted through training multiple decoders using different training datasets, or different features such as Mel-frequency cepstral coefficients and linear prediction cepstral coefficients [72][73]. Given these multiple decoders, if a majority of these decoders agree on a hypothesis, i.e., their recognized output is same, and then dialogue manager could choose to avoid an explicit confirmation from the user. Consider a set of decoders {d1, d2} ε D. For a given acoustic signal, let the corresponding hypothesis of D be {h1, h2} ε H. Let Ci is the contextual information for the dialogue state i. The following are possible cases and the corresponding

actions incorporated into proposed system using multiple decoders and contextual information:

Case 1: The hypotheses h1 and h2 are same and are present in the contextual information:

h1 = h2 and h1 ε Ci

Action: The recognition output is most likely to be correct and the system would jump to subsequent dialogue states.

Case 2: h1 = h2 and h1, h2 ε Ci

Action: The recognition output is most likely to be correct, but input is of no help as it is not present in the contextual information. The system would prompt the user, saying that no such information is available pertaining to that given input and would ask the user to provide some other query.

Case 3: h1! = h2 and h1/h2 ε Ci

Action: In such cases, system would try to consider the hypothesis that is present in the contextual information and discard the other. To make sure that the recognition is correct, the system would ask for an explicit confirmation from the user.

Case 4: h1! = h2 and h1, h2 ε Ci

Action: Proposed system-recognition might have occurred and the system would prompt the user to provide the information again. The above hypothesis can be easily extended for (n> 2) decoders.

6.4 ROLE OF CONTEXTUAL INFORMATION

In the current version of proposed system, the scheme of multiple decoders is implemented using two decoders. Each decoder differs from the other in its acoustic models[73]. The first decoder uses the acoustic models *AM-1* which is built on Bodo data. The second decoder uses the acoustic models referred to as *AM-2*. These acoustic models are built on a speech corpus consisting of isolated Bodo words. On a test data set of 5718 utterances from 20 speakers, the word level accuracy of *AM-1* and *AM-2* is shown in **Table 6.1**.

Table 6.1: Recognition score using the built acoustic models

Acoustic Models	Speech Data	Accuracy
AM-1	Bodo	81.24
AM-2	Bodo+ Continuous Speech	76.86

The proposed system could choose a specific affirmation simplest when more than one decoder is not in contract. **Table 6.2** shows the number of times *AM-1* and *AM-2* agree/disagree on their hypotheses. This evaluation was done on test data set. When the decoders are not in agreement, then the contextual information (expected set of concepts for the given state of a dialogue) plays a role in accepting/ rejecting the hypothesis. For the cases of (~A, B) and (A,~B) in **Table 6.2**, if one of the hypothesis is present in the contextual information, then it could be asserted through an explicit confirmation from the user. **Table 6.2** shows an example dialogue for Case 3 in the proposed system.

Table 6.2: Comparing AM-1 and AM-2 on the test data set

	B	~B
A	4092 (71.56%)	541 (9.46%)
~A	303 (5.29%)	782 (13.67%)

Here A indicates AM-1 being correct in its hypothesis, while B indicates AM2 being correct. ~A indicates AM-1 being incorrect. ~B indicates AM-2 being incorrect in its hypothesis

6.5 GRID SEARCH BASED CONSTRAINED CLUSTER MODEL INTERPOLATION

Adaptation techniques intend to reduce the acoustic mismatch between the speaker-independent (SI) acoustic models and the test data. These methods have become an integral part of the state-of-the-art automatic speech recognition (ASR) systems. Maximum a-posteriori (MAP) [75] and maximum likelihood linear regression (MLLR) [76] criteria form the basis for most of the conventional

adaptation techniques. These techniques require a considerably large amount of adaptation data and hence become largely ineffective when available adaptation data is small (≤10 s). A number of fast/rapid adaptations approaches have been proposed over the past decade to address this problem. These techniques generally use bases model parameter interpolation to derive the model parameters for the test speaker/utterance [77-79]. The interpolation weights are either estimated as a global parameter or a number of Gaussians are tied depending on some criterion (like regression classes) and one set of weights is estimated for each class. Since, only the interpolation weights are estimated in bases interpolation based approaches, even a small amount of adaptation data is sufficient for the estimation of these interpolation weights. As reported in [80-81], further improvements in system performance can be achieved by a dynamic selection of bases for each test speaker/utterance instead of keeping them fixed[74].

The reported fast adaptation techniques are generally evaluated in batch mode of operation where the transform parameters (interpolation weights) are estimated under the supervision of true transcription. These transform parameters are then used during the testing with speaker specific information assumed to be known to the system. There are some applications where such knowledge cannot be presumed and the adaptation data available is that test utterance only, i.e., online mode of adaptation as in case of spoken dialogue systems [82-85]. In our proposed work, we intend to target the issues of adaptation for an on-line recognition task where not only the available adaptation data is very small (≤ 2 s) but also no prior knowledge about the test speaker is presumed by the system. Consequently, an accurate representation of the test speaker in terms of an adapted model is difficult to achieve. At the same time, adaptation is required to be performed efficiently so that it does not add much latency to the overall system. The proposed adaptation approaches, though not explicitly optimal are yet found to be effective for the aforementioned unseen speaker and small data adaptation task in the context of SQ systems.

The speaker independent (SI) ASR systems have to deal with both the intra- and inter-speaker variability in contrast to the speaker dependent (SD) systems which have to deal only with the former. The intra-speaker variability refers to the inherent variations in a speaker and is also aided by the variations due to the changes in the health conditions and emotional state of the speaker. The inter-speaker variability is mainly caused due the gender differences (male and female speakers have different

fundamental frequencies due to differences in their vocal tract lengths), the speaking styles and accent differences among native and non-native speakers and the speaking rates differences among speakers. Consequently, the performance of the SI system is reported to be 2 to 3 times inferior compared to a speaker dependent system [86]. In addition to these, both the speaker independent and SD systems have to deal with the channel variability (speech collected over telephone channel and over microphones have different properties). There are techniques which address the channel distortions [87-88] beyond applying CMN. But those are still too complex to be applicable in real-time systems [74, 88].

The SD systems, though quite effective, are infeasible to be built for each speaker as it requires a large amount of data per speaker. Hence, speaker adaptation techniques have been developed which intend to modify the parameters of the SI system to better suit a particular speaker with as low data as possible from that speaker. Generally, a certain amount of speaker-specific developmental (adaptation) data, for which the true transcription is available, is used for transform parameter estimation. Once the transform parameters are generated for a speaker, the same is used to decode the test data belonging to that speaker. Unfortunately, this requires a significant amount of data as well as the a-priori knowledge about the test speaker. In context of the SQ system discussed in this work, both the conventional and the rapid adaptation approaches are found to be ineffective. This is so because neither a sufficient amount of adaptation data is available nor the test speaker is one among the training set speakers. In these cases we can hope to adapt the ASR system to the test data based on broad acoustic similarity only and not to any speaker specific space. In this regard, we explored rapid adaptation approaches that assume that the adapted model parameters lie in a low dimensional space defined by the linear interpolation of a set of acoustic models. These acoustic models can be defined either by creating a speaker adapted (SA) model for each of the speakers in the training set [77, 80, 81] or by clustering the speakers using some similarity criteria [79]. Employing SA models is reported to be very effective as it provides a greater acoustic/linguistic diversity by capturing the intra-speaker variability. Such an approach, at the same time, results in an increased complexity due to the interpolation of a large number of bases (SA models). In case of SQ systems, this can be reduced by clustering the speakers in the training set into a small number of clusters. This leads to a loss in the finer acoustic details (the intra-speaker variability) due to the pooling of data from a large number of

speakers. Deriving such averaged models is the price paid to avoid the increased latency incurred in case of the former approaches. In this work, for acoustic clustering, each speaker in the training set is first represented as a super vector derived by concatenating the Gaussian mean parameters of its respective SA model [88]. In this work, MAP adaptation of the SI monophone models is performed to create the speaker specific supervectors. This ensures that almost all the Gaussians get adapted using the available speaker specific data and hence the derived super vector uniquely represents a speaker [87].

These super vectors are then grouped into a desired number of clusters using vector-quantization (VQ). Pooling the speech data corresponding to all the speakers assigned to a particular cluster, a cluster model is then created using MAP adaptation of the SI model (triphone HMM). An added advantage of acoustic clustering is that it reduces the memory requirements for storing the candidate models. Using SA models corresponding to each of the speakers also hampers the system portability due to large memory requirements [84]. Once the set of acoustic cluster models are created, on-line adaptation can be performed in the following two ways:

a. For each test utterance, the most appropriate acoustic (cluster) model is chosen using Viterbi-alignment based maximum likelihood (ML) search. The test utterance is then re-decoded using the cluster model that has the highest likelihood with respect to that utterance.

b. Like [57], instead of re-decoding using the highest likelihood cluster model, the adapted model parameters can be obtained by the linear interpolation of the model parameters of the top K most likely acoustic models.

The fast adaptation techniques discussed in [76,77] attempt to find an acoustic space which is more closer to the test utterance through a dynamic selection of vectors from a set of bases rather than using fixed basis vectors. Once the preferred basis vectors are found, these are linearly interpolated in chosen parameter space to estimate the adapted HMM model. A set of optimal weights for interpolation are estimated by maximum likelihood estimation iteratively. These interpolation weights may be initialized either equilikely or randomly. These interpolation weights are usually applied to all Gaussians in the model. Among the different parameters of the model which could be interpolated for adaptation, the mean and/or the mixture-weight

vector are most successful ones [76, 77]. Unlike the mean vector interpolation, the mixture-weight vector interpolation requires that weights should be non-negative as well as sum to one. It is well known that the estimated weights should turn out to be such that more emphasis is given to those basis models which have a higher likelihood for the adaptation data. This motivated us to explore the derivation of interpolation weights in proportion to the likelihoods of the adaptation data with respect to a set of basis models. The test data is force-aligned with a set of basis models and K-top likelihood basis models are selected. The likelihood scores of these selected models are then normalized to form a set of interpolation weights for mean and/or mixture-weight vectors of the basis models. The detailed procedure is outlined in the following proposed algorithm.

Algorithm: Estimation of interpolation weights wi and adapted model parameters Lemda.

Assume: N basis models for adaptation

Step1: Obtain hypothesis for test data using first-pass decoding with SI model

Step2: Force-align test data under the constraint of first pass hypothesis against each of the bass model and obtain the likelihood score array {Li}, i=1,…….N

Step3: Sort the array {Li} in descending order and select the top K acoustically close basis models (k<N)

Step4: For K base models, estimate the interpolation weights Wi with 0< Wi <1 and

$$\sum_{i=1}^{K} \mathrm{Wi} = 1$$

$$\mathrm{Wi}=\frac{Li-\min\{Li}{\sum_{i=1}^{K}[\mathrm{Li}-\min\{\mathrm{Li}\}=1]}$$

Step5: Derive adapted model as λ_i $=\sum_{i=1}^{K} \mathrm{Wi}$ λ_i

mean super vector by men-only MAP adaptation on 8 mixtures monophone-HMM models. This super vector is then clustered using LBG algorithm into desired number of clusters. Using the data belonging to the speakers in each of the clusters, the corresponding basis (cluster) models are then derived by MAP adapting the mean and/or mixture-weight parameters of the triphone level SI model. We have

experimented with different number of cluster models and noted that best performance is obtained with 8 cluster models. The performance of the proposed algorithm when applied to mean vector, mixture-weight vector, and both mean and mixture-weight vectors interpolation is given in **Table 6.3**.It is to note that the proposed approach not only out performs the baseline ASR system (commodity name recognition) but also the simple maximum-likelihood cluster decoding in all.

Table 6.3: Performances of the proposed adaptation approach with 4 top likelihood clusters derived out of 8 acoustic clusters employed for interpolating different model parameters.

Type of Adaptation	WER (in %)
Max likelihood cluster search	15.00
Cluster mean interpolation	14.80
Cluster mix-with interpolation	14.8
Cluster mean and mix with interpolation	14.30
Baseline ASR System	16.00

SUMMARY

The work presented in this chapter made an attempt to make a hypothesis along with VAD module for increasing the recognition accuracy. Proposed hypothesis helps the user to proper understating of the call flow and finding there result in efficient manner. Apart from this, the issues in adapting an entity recognition system with small adaptation data and that too in unseen speaker case are discussed. A simple cluster model interpolation technique for model mean and/or mixture-weight adaptation is proposed. The proposed approach results in a performance improvement by 11% relative over the baseline ASR system. In future, we would like to explore the use of the interpolation weights obtained using the proposed approach as the prior in existing ML weight estimation approaches.