An Exploration of Large Vocabulary Tools for Small Vocabulary Phonetic Recognition

Tara N. Sainath, Bhuvana Ramabhadran and Michael Picheny

IBM T.J. Watson Research Center
1101 Kitchawan Road, Yorktown Heights, NY 10598, U.S.A.
{tnsainat, bhuvana, picheny}@us.ibm.com

Abstract—**While research in large vocabulary continuous speech recognition (LVCSR) has sparked the development of many state of the art research ideas, research in this domain suffers from two main drawbacks. First, because of the large number of parameters and poorly labeled transcriptions, gaining insight into further improvements based on error analysis is very difficult. Second, LVCSR systems often take a significantly longer time to train and test new research ideas compared to small vocabulary tasks. A small vocabulary task like TIMIT provides a phonetically rich and hand-labeled corpus and offers a good test bed to study algorithmic improvements. However, oftentimes research ideas explored for small vocabulary tasks do not always provide gains on LVCSR systems. In this paper, we address these issues by taking the standard "recipe" used in typical LVCSR systems and applying it to the TIMIT phonetic recognition corpus, which provides a standard benchmark to compare methods. We find that at the speaker-independent (SI) level, our results offer comparable performance to other SI HMM systems. By taking advantage of speaker adaptation and discriminative training techniques commonly used in LVCSR systems, we achieve an error rate of 20%, the best results reported on the TIMIT task to date, moving us closer to the human reported phonetic recognition error rate of 15%. We propose the use of this system as the baseline for future research and believe that it will serve as a good framework to explore ideas that will carry over to LVCSR systems.**

I. Introduction

Speech recognition research in the past few years has focused heavily on large vocabulary continuous speech recognition (LVCSR). Large vocabulary corpora are attractive as they provide a testbed for which to tackle many real-world problems such as noisy speech, accented speech, large vocabulary modeling and real-time decoding. Research in LVCSR, particularly in the areas of discriminative training and speaker adaptation, have resulted in a significant improvement in performance and an increase in usage of speech recognition systems. However, LVCSR research suffers from two major drawbacks. First, because of the large number of parameters and poorly labeled transcriptions, gaining insight into further improvements based on error analysis is very difficult. Second, model training typically requires many hours compared to small vocabulary, providing challenges for testing new ideas.

However, many ideas which have shown good gains on small vocabulary tasks do not necessarily translate to gains in LVCSR. For example, as we will demonstrate in this work, Mel-Scale Cepstral Coefficients (MFCCs) have been shown to offer better performance on certain small vocabulary corpora compared to Perceptual Linear Predictive (PLP) features. However, once discriminative and speaker adaptive training methods are incorporated, the performance using the two different feature sets are the same. In addition, conditional random fields [1] have also shown promising results for phonetic recognition, but require a some-what structured dataset, and thus currently have not been heavily explored in LVCSR.

Furthermore, models which have shown success on small vocabulary tasks can be computationally expensive in large vocabulary systems. For example, while recurrent neural nets (RNNs) have demonstrated good performance on both on small and large vocabulary tasks [2], these methods are computationally expensive to train acoustic models, and thus have not been heavily pursued for LVCSR. In addition, while Hidden Trajectory Models (HTMs) have also shown promising results for small vocabulary phonetic recognition, the computationally expensive decoding process [3] again provides for a computational challenge when applied to large vocabulary.

In this paper, we introduce a framework to address the problem of training and error analysis in LVCSR systems, as well as gains not carrying through to from small scale to large scale tasks. Specifically, we explore applying a complete LVCSR system to a small vocabulary corpus. Most LVCSR systems, including our IBM Recognizer [4], utilize a specific "recipe" during acoustic model training. First a set of speaker independent (SI) models are built. Next, a set of speaker adapted (SA) models are built for each speaker or speaker cluster. Finally, a discriminative training step is employed to produce a set of discriminative features and models for further error rate reduction. This recipe has shown considerable gains on conversational speech [4] and broadcast news [5] tasks.

Our experiments are conducted on the TIMIT corpus [6]. Our motivation of applying our LVCSR system to TIMIT is threefold. First, it provides a fair benchmark for comparing the performance of our LVCSR recipe to other state of the art results on this phonetic recognition task. Second, having time-aligned phonetic transcriptions allows for a detailed error analysis and suggests areas for future improvements in LVCSR research. Third, exploring this LVCSR recipe on TIMIT provides a framework for testing new LVCSR ideas. Specifically, if others can quickly apply new ideas to TIMIT using this LVCSR recipe, and are able to achieve improvements on top of speaker adaptation and discriminative training, then we believe similar gains will be seen on a large scale task. For example, gains first seen using discriminative training on small vocabulary [7] have translated into huge gains for LVCSR [8]. In addition, improvements seen on TIMIT using neural nets [2] have also been successfully applied to LVCSR systems [9].

Our phonetic recognition experiments reveal that at the SI level, we are able to achieve a phonetic error rate (PER) of 25.6%, which compares to one of the best SI-Hidden Markov Model (HMM) results reported in the literature ([10]). Next, we find that utilizing discriminative training, the results are significantly better than the performance of other discriminative training systems on the TIMIT task [11]. Incorporating speaker adaptation allows us to achieve an error rate of **20.0%**. To our knowledge, we believe that utilizing our full system offers the best results on the TIMIT task to date. A spectrogram reading experiment in [12] reported a human level error rate of reading phonemes of approximately 15.0%. Our error rate of 20.0% illustrates the benefits of an LVCSR recipe for speech recognition, pushing speech research closer towards the ultimate goal of reaching human-level performance. A further error analysis indicates that most of the errors are due to confusions between phonemes within the same manner class, suggesting areas for future possible LVCSR research.

The rest of this paper is organized as follows. In Section II, an overview of the IBM LVCSR System used for experiments in this paper is provided. Section III outlines the experiments performed, while Section IV analyzes the results. A discussion of implications for LVCSR research of applying this LVCSR recipe on small vocabulary tasks is presented in Section V. Finally, Section VI summarizes the main contributions of the paper and discusses future work.

II. System Architecture

The LVCSR recognizer at IBM operates in a series of steps, as indicated in Figure 1. First, feature vectors are extracted from the speech signal. Next, a set of speaker independent (SI) sub-word unit models are trained. Then, using the set of SI models, a set of speaker adapted (SA) features and models are learned. Finally, feature and model space discriminative training is applied on top of the SA system. Below each component of the process is described in more detail.

A. Front-end Processing

A speech utterance is first chunked into 20ms frames, with a frame-shift of 5 ms. Each frame can either be represented by 13 dimensional MFCCs or 19 dimensional PLP features. Features are then mean and variance normalized on a per utterance basis. Then, at each frame, a series of consecutive frames surrounding this frame are joined together and a Linear Discriminative Analysis (LDA) transform is applied to project the feature vector down to 40 dimensions.

B. Speaker Independent Acoustic Modeling

In SI modeling, each sub-word unit is a phoneme, represented by a 3 state left-to-right Hidden Markov Model (HMM) with no skip states. The output distribution of each state is modeled by a Gaussian Mixture Model (GMM). First a series of a set of context-independent (CI) models are trained using information from the transcription. Maximum likelihood (ML) estimation is used to train parameters of the HMM. The training of CI models produces a set of state-level alignments of the speech against their corresponding transcripts.

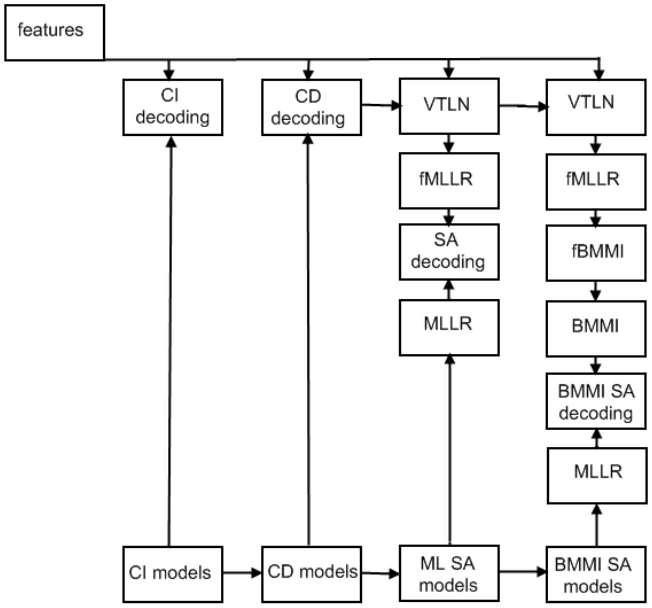

Fig. 1. Block Diagram of Various Stages in the IBM LVCSR System

The CI models are then used for bootstrapping the training of a set of more complex triphone context-dependent (CD) models, which can capture more acoustic variability. These CD models are also modeled by a 3-state left-to-right HMM with no skip states. As with most speech recognizers, due to data availability issues in modeling all possible CD triphones, a clustering procedure is employed to share data across various CD models. First, a set of united CD HMMs are estimated for each possible triphone combination. Next a top-down decision tree is grown for each phone, and states belonging to the same phone are tied together. The questions used to generate a top-down decision tree are the standard questions used in HTK for TIMIT [13]. However, due to data limitations in a small vocabulary task such a TIMIT, we explore using a global decision tree to tie together phones. Specifically, a data-driven set of 13 broad phonetic classes (BPCs) is specified, and phones in the same BPC are tied together. After the clustering learns a set of CD states, a set of GMMs is trained for each state. The number of GMMs for each CD state is not fixed, but is rather a function of the number of frames assigned to that state. First, a set of low complexity GMMs are used to model each state, and the Gaussian components are split and grown in subsequent training iterations.

C. Speaker Adapted Acoustic Modeling

After a set of SI models are designed, they are used to bootstrap the training of a set of SA models. In SA modeling, first vocal tract length normalization (VTLN) is applied, followed by a feature/model space adaptation. Both steps are discussed below in more detail.

1) Vocal Tract Length Normalization: The length of a speaker's vocal tract is often a large factor in speaker variability. VTLN is a popular technique used to reduce this variability. In this procedure, a scaling factor is learned for each speaker that warps the speech from this speaker into a canonical speaker with an average vocal tract length. The warp is applied to the given set of acoustic features before they are LDA transformed. After the warp, features are again spliced together at each frame and an LDA transform is applied to produce a set of 40 dimensional "warped" features.

2) Feature/Model Space Adaptation: After VTLN, the "warped" features are adapted for each speaker using a popular feature adaptation method known as feature space Maximum Likelihood Linear Regression (fMLLR) [14]. Next, using the adapted fMLLR features, the set of CD models are adapted to each speaker using a technique known as Maximum Likelihood Linear Regression (MLLR) [14]. For MLLR, an eight-level binary regression tree is used, which is built by successively splitting the nodes in a top-down manner using a soft K-means algorithm.

D. Discriminative Training

Finally, the SA ML models are used to used to bootstrap the training of a set of discriminatively trained features and models. A variety of criterions can be used for discriminative training, including Minimum Phone Error (MPE) [8], Maximum Mutual Information (MMI) and Boosted Maximum Mutual Information (BMMI) [8]. In this paper, we explore using the BMMI criterion to design a set of discriminatively trained features. Then using these new fBMMI features, a second discriminative step using the BMMI criterion is applied to produce a set of discriminatively trained models. Finally, MLLR transforms are applied to the discriminatively trained models. Please note that the discriminative training step can occur Section II-B after a set of CD ML models are trained. However, discriminative training is usually done after SA models are built (i.e., Section II-C), as we have observed this to provide larger gains. To fairly compare our results on TIMIT to other results, in this work we will explore discriminative training after both the SI CD and SA CD stages.

III. Experiments

The experiments in this paper are conducted on TIMIT [6], a continuous speech recognition corpus recorded and transcribed by Texas Instruments (TI) and the Massachusetts Institute of Technology (MIT), respectively. It contains over 6,300 phonetically rich utterances read by 630 speakers. The sentences from the corpus are divided into three sets. The standard NIST training set consists of 3,696 sentences, used to train various models used by the recognizer. The development set is composed of 400 utterances and is used to train various tuning parameters in the LVCSR system. The full test set includes 944 utterances, while the core test set is a subset of the full test set containing 192 utterances.

In accordance with standard experimentation on TIMIT [15], the 61 phonetic labels are collapsed into a set of 48 for acoustic model training, ignoring the glottal stop [q]. A set of CI HMMs are trained using information from the phonetic transcription. The output distribution of each CI state is a 32-component diagonalized-covariance GMM. The CI models are then used for bootstrapping the training of a set of triphone CD HMMs. Due to the small vocabulary nature of the task, a global-tree clustering algorithm described in Section II-B is used, which allows for both states and phones to be tied together. Totally the CD system has 2,400 states and 15,000 Gaussian components, which was chosen to optimize performance on the development set. The number of training iterations for each stage of the LVCSR process was also chosen to optimize performance on the dev set and avoid overtraining. A trigram language model is used for all experiments.

For testing purposes, the standard practice is to collapse the 48 trained labels into a smaller set of 39 labels [15]. A variety of experiments are conducted to compare different stages in our LVCSR process to results reported in the literature. More specifically, we first compare the performance of the IBM CI system, followed by an investigation of the behavior of the IBM CD system. Third, we analyze the benefits to discriminative training. Fourth, we explore the results when feature and speaker adaptation are performed. All phonetic error rates (PERs) are reported on the TIMIT core test set.

IV. Results

In this section, we present our results on TIMIT for various stages in the LVCSR framework.

A. Speaker Independent System

1) Context Independent System: Since many CD systems are designed from bootstrapping CI models, we first explore the behavior of our CI models. Table I compares the results of various CI systems reported in the literature. As we can see, the IBM system has the lowest PER of all systems. We believe one major explanation for the improved performance over other techniques is the use of robust features which are mean and variance normalized and then LDA transformed. The benefit of having good performance from CI models will be discussed in more detail in the next section.

TABLE I
Comparison of CI Systems on TIMIT Core Test Set

System	PER (%)
3-state CI HMM [13]	38.3
CI Segment-Based System[16]	35.9
7-state CI HMM [15]	35.9
IBM, 3-state CI HMM (this paper)	**27.7**

2) Context Dependent System - Maximum Likelihood Trained: In order to train a set of CD models, models can either be bootstrapped using CI models, or trained using phonetic transcriptions. Table II compares the results on CD models for the two initial model procedures. The table indicates that bootstrapping from CI models provides a slight improvement over using phonetic transcriptions. The benefit to bootstrapping from CI models is that the sequence of states within a particular phone is determined from a force alignment using the CI models, as opposed to having state alignments across a particulary phone evenly split when underlying phonetic transcripts are used. While subsequent iterations of training CD models remove this constraint on state alignments when using phonetic transcriptions, the poor initial state alignment provided from using phonetic transcriptions leads to a slightly higher PER after multiple training iterations compared to when CD models are bootstrapped from CI models, as shown by Table II.

TABLE II
IBM CD Systems on TIMIT Core Test Set for Different CI Boosting Methods

System	PER (%)
IBM CD HMMs - trained from transcripts	25.9
IBM CD HMMs - bootstrapped from CI	**25.6**

Most LVCSR systems utilize a top-down decision tree to tie states together. However, because of data limitations on a small-scale task like TIMIT, we utilize a clustering technique, discussed in Section II-B, where both states and phones are grouped together at the phone class level. Table III compares the results of the two clustering techniques. Tying both states and phones, allowing better data sharing, offers a 0.7% improvement over tying just states.

TABLE III
IBM CD Systems on TIMIT Core Test Set for Different Clustering Techniques

System	PER (%)
IBM CD HMMs - state tying	26.1
IBM CD HMMs - state+phone tying	**25.6**

To further analyze the performance of our CD system, Table IV compares the results of various CD systems reported in the literature. For fair comparison, please note that none of these systems are discriminatively trained. The IBM CD system offers a PER of **25.6%**, which performs better than the HMM systems listed in [13] and [17], and comparable to [10]. Further improvements in PER were achieved in [18], though the total acoustic model scoring was produced by combining scores produced from separate HMM and HTM systems. In addition, the results in [19] are achieved using a combination of feature sets, rather than one feature as is done in the HMM systems. Typically the use of specialized features has not been shown to provide gains in LVCSR systems once additional techniques such as speaker adaptation and discriminative training are applied, as discussed in more detail in Section V. Therefore, it would be interesting to see if the specialized features in [19] are able to provide gains on top of the LVCSR recipe.

TABLE IV
Comparison of CD ML Trained Systems on TIMIT Core Test Set

System	PER (%)
Triphone Discrete HMMs [15]	33.9
CD Segment-Based Model [16]	30.5
Triphone Continuous HMMs [17]	26.6
Generalized Triphone HMMs [13]	26.3
Recurrent Neural Nets [2]	26.1
Bayesian Triphone [10]	25.6
IBM CD HMMs (this paper)	**25.6**
Monophone HTMs [18]	24.8
CD Segment-Based Model, Heterogeneous Measurements [19]	24.4

3) Context Dependent System - Discriminatively Trained: Discriminatively trained acoustic models have been shown to significantly improve error rates compared to ML trained models, as these discriminative models have more power to better differentiate between confusable sounds, such as “ma” and “na”. In this work, we use a large margin discriminative training approach using the Boosted Maximum Mutual Information (BMMI) criterion [8]. We apply discriminative training first to the feature space and then the model space, as past research has indicated that method allows for significant improvements in word error rate [8]. Please note that we also explored the MPE criterion, though the objective function appeared sensitive to the phone accuracy counts for phonetic recognition and therefore little gain was found. Similar results were also observed in [20].

Table V compares the results of variously discriminatively trained systems on the TIMIT Core test set. The feature and model space discriminative training are indicated as fBMMI and BMMI respectively. Since it is somewhat difficult to compare error rates for different discriminative training methods, as the baseline ML error rates are different, we have also provided the relative improvement provided by discriminative training over ML for each method. Note that our discriminative training methods provide a large relative improvement in PER ML trained models and the best results of all discriminatively trained methods from an absolute perspective.

TABLE V
Comparison of CD Discriminatively Trained Systems on TIMIT Core Test Set

System	PER (%)	Rel. PER Red. from ML
MMI Training [7]	28.2	4.2
Large-Margin Training [11]	28.2	13.8
P-MCE [20]	27.0	6.5
IBM fBMMI (this paper)	22.7	11.3
IBM fBMMI+BMMI (this paper)	**22.7**	11.3

B. Speaker Adaptive System

In most LVCSR systems, after a set of SI CD models are built, a set of SA models are trained. In this process, first VTLN is applied to reduce speaker variability. Next, the features are transformed for each speaker using fMLLR. This series of steps is known as speaker adaptive training (SAT). Next, feature and model space discriminative training is applied on top of the SAT system, as typical LVCSR systems have found better gains applying discriminative training after SAT rather than before. Finally, the discriminatively trained models are further adapted using MLLR. Table VI shows the error rates for each stage of this process. The SAT stage provides a 2.9% decrease in PER. Note that while the PER increases from the VTL stage, we found better performance using an fMLLR+VTL system (22.7% PER) rather than just applying fMLLR (23.5% PER). Discriminative training allows for a further 2.3% reduction in PER, and applying MLLR on top of this provides a PER of **20.0%**.

We would particularly like to comment on the error rate of 20.0%, which to our knowledge is the lowest reported error rate on the TIMIT task to date. In [12], human level error rate of reading phonemes from speech spectrograms was measured at approximately a 15.0%. We believe that our error rate of 20% illustrates the benefits of an LVCSR recipe for speech recognition, pushing speech research closer towards the ultimate goal of reaching human-level performance.

TABLE VI
PER FOR VARIOUS LVCSR STAGES ON TIMIT CORE TEST SET

System	PER (%)
SI System	25.6
+VTL	26.2
+fMLLR	22.7
+fBMMI	20.4
+BMMI	20.4
+MLLR	**20.0**

C. Error Analysis

The benefit of using TIMIT for LVCSR research is that it allows for a detailed error analysis, which we present in this section. First, we analyze the error rates for each stage of the LVCSR recipe. Figure 2 shows the breakdown of error rates (log-scale) for each stage of the process within 3 BPCs, namely vowels/semivowels, stops and closures/silence. Here the error rate was calculated by counting the number of insertions, deletions and substitutions that occur for all phonemes within a particular BPC. A few points can be observed from the figure. First, notice that the PER within the stop class decreases more than 18% from the CI to BMMI stage. Due to their short durational nature, stops are one of the more difficult phonemes to model with an HMM. Most of the reduction in error rate at each LVCSR stage is due to improved modeling of stop consonants. Second, the PER for the silence/closure class remains relatively constant for all LVCSR stages. Similar trends were also observed for nasals, strong fricatives and weak fricatives. Third, notice that the PER within the vowel class increases approximately 10% from the CI to CD stage. A closer analysis revealed that most error rates were due to short-duration monothongs (i.e. [ah], [eh], [ih]). This implies that when bootstrapping from CI models at the CD level rather than using phonetic transcriptions as is done at the CI level, determining the boundaries of these short-duration phonemes during training can be difficult, leading to poor modeling. This suggests that duration modeling may improve results in the future.

Second, we analyze the substitution errors, which constitutes the majority of the error rate, for the best performing system at 20% PER. Figure 3 shows a confusion matrix of substitution errors for each phoneme, with phonemes within the same manner class also indicated. We find that approximately 80% of confusions occur within the same manner class, as was similarly observed in [19]. A high number of confusions exists because linguistic knowledge when recognizing a sequence of phonemes as belonging to a word was not used in our system, but was available in the experiment in [12].

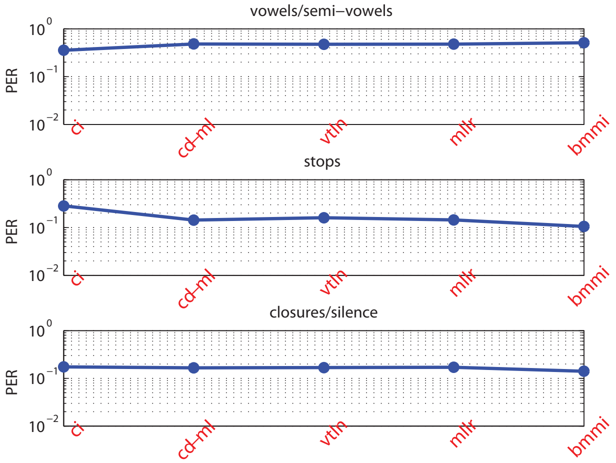

Fig. 2. Error Rates (log-scale) within BPCs for various LVCSR Stages

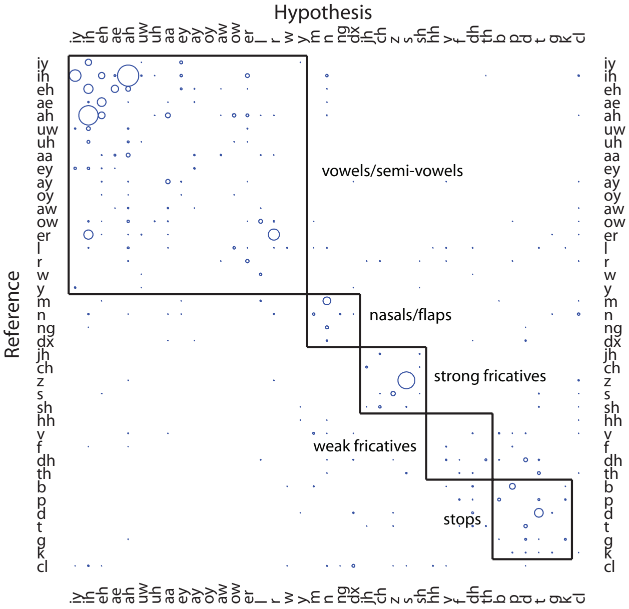

Fig. 3. Confusion Matrix of substitution errors with radii linearly proportional to the error. The largest bubble represents 5.1% of the total error.

V. IMPLICATIONS FOR RESEARCH IN LVCSR

One benefit of using TIMIT for LVCSR research is that it offers a phonetically rich and hand-labeled corpus, allowing for a detailed error analysis. Because a reliable phonetic transcript is often not available for LVCSR systems and the number of parameters is very large, it is often hard to pinpoint exact causes of errors. However, using the LVCSR recipe for TIMIT allows for error analysis and potential improvements for various aspects of LVCSR systems, for example improving model topology and pronunciation generation.

Another benefit of TIMIT is that its provides a good benchmark to quickly test solutions, which is often a problem in most large vocabulary tasks. For example, parallelizing on over 30 machines[1], training more than 10 hours of TIMIT training data for any stage in the LVCSR process took less than one hour. In addition, parallelizing on over 30 machines, decoding on the core test set took less than 5 minutes.

Oftentimes gains seen at the CD stage for different modeling techniques do not always carry through once the SA and discriminative training stages are applied, as these provide most of the gains in LVCSR systems. For example, we have observed that gains from various feature representations seen at the CD stage on TIMIT do not consistently hold through all stages. Table VII compares the results on the TIMIT for various LVCSR stages using both MFCC and PLP features. Notice that at the CD level, MFCC features provide better performance than PLP features. However, at the SA stage, the performance using both feature sets is the same. We believe that if new research ideas, such as new acoustic features, produce improvements on TIMIT using the entire LVCSR recipe, than these ideas are able to withstand the discriminative training and speaker adaptation stages, and will hopefully result in gains on a large vocabulary task as well. This is one of the major benefits for using TIMIT for LVCSR research.

TABLE VII
COMPARISON OF FEATURE CHOICES ON TIMIT CORE TEST SET

Stage	MFCC-PER	PLP-PER
CI	27.7	27.6
CD-ML	25.6	26.3
VTLN	26.2	24.1
fMLLR	**22.7**	**22.7**

VI. CONCLUSIONS

In this paper, we presented a framework for quickly testing ideas for LVCSR systems on a small scale task. Specifically, we analyzed the phonetic recognition performance of our IBM LVCSR system on the TIMIT corpus. We showed that at the speaker-independent level, our results were comparable to the best previously published SI HMM results. In addition, utilizing speaker adaptation and discriminating training provided an error rate of 20%, the best results on the TIMIT task to date, and moving us closer to human phonetic recognition performance. We hope that this presented LVCSR "recipe" for small scale tasks will provide a new framework for LVCSR research. Specifically, if ideas can quickly be tested on TIMIT corpus and gains are found using this recipe, these ideas can subsequently be applied to an LVCSR system.

In the future, we would like to expand this work in a number of directions. First, we are interested in seeing if new features, such as articulatory features, will provide gains on the TIMIT corpus using this LVCSR framework, and can thus subsequently be applied to LVCSR systems. Secondly, many LVCSR system lack the ability to process speech outwards from reliable regions, something which humans take advantage of when processing speech. This method of processing speech from reliable to unreliable regions, known as island-driven search, is an area we are very interested in exploring in the future. In addition, our analysis of within class confusions hints that vowel performance in LVCSR can be improved by merging together highly confusable vowels into one class (i.e. [ih] and [ah]), which can aid in pronunciation generation by reducing the number of pronunciations. In addition, due to the high error rates within short and long vowels, voicing and duration modeling in LVCSR might also improve error rates.

VII. ACKNOWLEDGEMENTS

The authors would like to thank Hagen Soltau, George Saon, Brian Kingsbury and Stanley Chen for their contributions towards the IBM toolkit and recognizer utilized in this paper.

REFERENCES

[1] E. Fosler-Lussier and J. Morris, "Crandem Systems: Contitional Random Field Acoustic Models for Hidden Markov Models," in *Proc. ICASSP*, 2008.

[2] A. Robinson, "An Application of Reccurent Nets to Phone Probability Estimation," *IEEE Transactions on Neural Networks*, vol. 5, pp. 298–305, 1994.

[3] D. Yu, L. Deng, and A. Acero, "Speaker-Adaptive Learning of Resonance Targets in a Hidden Trajectory Model of Speech Coarticulation," *Computer Speech and Language*, vol. 21, pp. 72–87, 2007.

[4] H. Soltau, B. Kingsbury, L. Mangu, D. Povey, G. Saon, and G. Zweig, "The IBM 2004 Conversational Telephony System for Rich Transcription," in *Proc. ICASSP*, 2005.

[5] H. Soltau, G. Saon, B. Kingsbury, J. Kuo, L. Mangu, D. Povey, and G. Zweig, "The IBM 2006 Gale Arabic ASR System," in *Proc. ICASSP*, 2007.

[6] L. Lamel, R. Kassel, and S. Seneff, "Speech Database Development: Design and Analysis of the Acoustic-Phonetic Corpus," in *Proc. of the DARPA Speech Recognition Workshop*, 1986.

[7] S. Kapadia, V. Valtchev, and S. J. Young, "MMI Training for Continuous Phoneme Recognition on the TIMIT Database," in *Proc.ICASSP*, 1993.

[8] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for Model and Feature Space Discriminative Training," in *Proc. ICASSP*, 2008.

[9] B. Kingsbury, "Lattice-Based Optimization of Sequence Classification Criteria for Neural-Network Acoustic Modeling," in *Proc. ICASSP*, 2009.

[10] J. Ming and F. J. Smith, "Improved Phone Recognition using Bayesian Triphone Models," in *Proc. ICASSP*, 1998.

[11] F. Sha, "Comparison of Large Margin Training to Other Discriminative Training Methods for Phonetic Recognition by Hidden Markov Models," in *Proc. ICASSP*, 2007.

[12] V. Zue and R. Cole, "Experiments on Spectrogam Reading," in *Proc. ICASSP*, 1979.

[13] S. J. Young, "The General Use of Tying in Phoneme-Based HMM Speech Recognizers," in *Proc. ICASSP*, 1992.

[14] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition," *Computer Speech & Language*, vol. 12, pp. 75–98, 1998.

[15] K. F. Lee and H. W. Hon, "Speaker-independent Phone Recognition Using Hidden Markov Models," *IEEE Transacations on Acoustics, Speech and Signal Processing*, vol. 37, pp. 1641–1648, 1989.

[16] J. Glass, J. Chang, and M. McCandless, "A Probabilistic Framework for Feature-Based Speech Recognition," in *Proc. ICSLP*, 1996.

[17] L. Lamel and J. Gauvain, "High Performance Speaker-Independent Phone Recognition using CDHMM," in *Proc. Eurospeech*, 1993.

[18] L. Deng and D. Yu, "Use of Differential Cepstra as Acoustic Features in Hidden Trajectory Modeling for Phonetic Recognition," in *Proc. ICASSP*, 2007.

[19] A. Halberstat and J. Glass, "Heterogeneous Measurements and Multiple Classifiers for Speech Recognition," in *Proc. ICSLP*, 1998.

[20] Q. Fu, X. He, and L. Deng, "Phone-Discriminating Minimum Classification Error (P-MCE) Training for Phonetic Recognition," in *Proc. Interspeech*, 2007.

[1] Note that the machines used for training and testing were Intel Core Blade Servers, with 3GHz Dual Core Processors.