Recent Advances in Speech Recognition System for IBM DARPA Communicator

Yuqing Gao, Hakan Erdoğan, Yongxin Li, Vaibhava Goel and Michael Picheny

IBM Thomas J. Watson Research Center
PO Box 218 Yorktown Heights, NY 10598
{yuqing, erdogan, vgoel, yongxin, picheny}@us.ibm.com

Abstract

In this paper, we present methods to improve speech recognition performance of the IBM DARPA Communicator system. Our efforts for acoustic modeling include training a domain specific yet broad acoustic model, speaker clustering and speaker adaptation using feature space transforms. For language modeling, we achieved improvements by using compound words, carefully designed LM classes and adjusting the within class probabilities, using NLU state information to enhance the language model and building a language model with embedded grammar objects. Our efforts produced a relative error rate reduction of 34.6% on the test set that consists of 1173 utterances that IBM received during the NIST evaluation of the DARPA Communicator systems in June 2000. We also tested our decoding on the data from some other sites to further demonstrate the robustness of the system improvements.

1. Introduction

The DARPA Communicator project describes a hub-and-spoke architecture for the design and development of natural language understanding systems. The system combines speech recognition, natural language understanding, dialog management, database access, language generation and speech synthesis to perform the desired task, which at present can be described as an automated travel agent that helps callers make airline reservations. The basic architecture of IBM's DARPA Communicator system is described in [1].

In this paper, we report our progress on the speech recognition portion of the IBM communicator system. Our work can be divided into two major areas, acoustic modeling and language modeling.

In the acoustic model area, we (1) used a new in-domain speech database and combined this with some general purpose data and re-trained the system; (2) used speaker clustering algorithms to further reduce the recognition error rate; (3) also used feature space MLLR transformations to conduct unsupervised speaker adaptation. These techniques have generated more than 30% relative improvement in reducing recognition error rate. Other new techniques, such as MMIE training and SAT training are also being experimented. No sizable improvement has been seen yet so far.

In the language model area, we have achieved improvements by (1) using compound words, carefully designing LM classes and adjusting the within class probabilities; (2) using NLU state information to enhance the language model; (3) and building a language model with embedded grammar objects. The new techniques generated about 15% error reduction.

Overall improvements are over 34.6% relative even when some new algorithms, such as unsupervised speaker adaptation, have not yet been included in a combined.

2. Experimental Setup

Three sources of data are served as test sets. Our main test set is composed of a subset of the calls that the IBM Communicator system received during the NIST evaluation in June 2000 (1173 utterances out of 1262 received). We refer it as NSTVAL. Data sets comprised of calls received by other sites which also participated in the same evaluation are also used in some test conditions. They are referred as DC1, DC2, DC3, ..., etc. to maintain anonimity. The third test data source is the two test sets referred as SMALL and LARGE in [1]. We use them only for baseline system comparison purpose.

3. Acoustic Modeling

The previous acoustic model (SYS0) reported in [1] was trained from an in-house general purpose telephony speech database which has about 600 hours of speech. This data includes personal names, zip codes, business names as seen in the yellow pages, street addresses, credit card, telephone numbers, and mutual fund names, etc. It was collected from over 20,000 speakers. There are obviously some phonetic context mismatches between the general purpose training data and the air travel domain application. A new database has been collected within the domain of airline travel reservation, which has has about 380 hours of speech and was collected from over 4000 speakers over various telephone lines.

3.1. Baseline Acoustic Model Training

First, we experimented with the best way to combine the new domain dependent data with the old general purpose speech corpus to minimize the speech recognition error rate.

The recognition error rates produced by the system (SYS1) trained from the in-domain data are much lower than the error rates of SYS0 (See Table 1) based on previous experiments. However, we had expected more error reduction from the in-domain data. We found that the in-domain data is not rich enough in terms of acoustic context variety as it only covers about 1000 words. More precisely, the 1000 most frequently used words in the speech corpus covered 99% of the text. This lead us to believe that the decision tree trained from this new data did not cover a rich variety of phonetic contexts, some of context cases, for example, city names which do not appear in the training corpus, may not be well represented by the training data and the resulting decision tree.

In order to enlarge the phonetic context coverage of the decision tree, some portion of general-purpose speech data was selected to augment the new data. Although ideally one should pick sentences from generic corpus which introduce new words or contexts, we only picked a random 250 hours of speech. One indication of that the phonetic context has been enriched by including a subset of the general purpose database is that the number context-dependent phones is increased to 3000 from 2000 when the same spliting threshold is used in building decision trees in both cases. The new system trained from the combined speech corpus is called SYS2.

	SYS0	SYS1	SYS2
LARGE	18.0%	15.0%	14.7%
SMALL	20.3%	17.0%	16.2%
NSTVAL	23.7%	21.0%	19.0%

Table 1: Comparison of baseline acoustic models

Table 1 showed recognition error rates obtained from 3 systems. These systems all have about 40,000 Gaussians. It can be seen from 1 that the system (SYS2) trained from the combined speech corpus produces a consistent and significant error reduction across 3 test sets over SYS0 and SYS1. So SYS2 is used as the baseline acoustic model throughout.

The error rates in table 1 may be a little different than the ones reported in [1] since a different homonym filter in scoring was used in their results.

3.2. MMIE and SAT Training

Systems SYS0, SYS1 and SYS2 are all trained using maximum likelihood algorithm, known as Baum-Welsh re-estimation algorithm. An attempt to improve the baseline system performance is to use the MMIE criterion in estimating the Gaussian parameters as it has shown significant improvement for other tasks, such as Switchboard [9]. The MMIE statistics only for a subset of the combined speech corpus (about half in size) are obtained by the time this paper is written because of the heavy computation cost. The Gaussian parameters re-estimated using MMIE criterion on this half data does not generate sizable improvement yet. Experiments on the complete training data set are on-going.

We are also experimenting SAT (Speaker Adaptive Training) during speaker-independent acoustic model training in order to reduce the inter-speaker variation. A modified version of [2] is implemented as applying a feature space transformation to each speaker to reduce the speaker-specific variation in the speech signal. The experiments in this area are still going on and no sizable improvement has been seen yet.

4. Speaker Clustering

From our experience on large vocabulary speech recognition for wide-band speech, we know that speaker clustering can help to reduce the covariance of the Gaussian models and leads to significant recognition error reduction [5]. Another important experience is that when 2 clusters are used, the automatic speaker clustering algorithm produces a separation of speakers according to their gender. The cluster selection algorithm used during testing also selects the cluster for the testing speaker very accurately matching with the speaker's gender. For example, we have found that for a wide-band test database of 25 speakers, the cluster selection result completed and correctly matches the speaker's gender. More importantly, the speech recognition error rate is always minimized when the right gender model is used for recognition.

However, the same algorithms did not produce the same speaker clustering and cluster selection results when apply to the DARPA Communicator data. When 2 clusters are used, the automatic speaker clustering algorithm does not produce a gender separation for speakers in the training database. Moreover, the automatic cluster selection algorithm does not always select a cluster for a speaker to minimize the recognition error rate. Instead, one of the cluster models always produces the same or less (for some test sets) amount of recognition errors than the baseline speaker-independent model for all the speakers. This phenomenon lead us to believe that there exist some other unknown conditions in our telephony database which are stronger than the gender differences among speakers that are being captured by the speaker clustering algorithm. We have already checked thoroughly that these conditions do not correspond to differences among landline, cellular phone, handset or speaker phone, etc. We are still experimenting to try to understand what are these effects.

Table 2 comparied the results between speaker independent system SYS2 and a two-clusters system. The superscripts on the test set names indicate which language model is used, as we have had more than one language models. The superscript 1 represents old LM, and 2 means new LM as described in Section 6. The column CLS2.1 and CLS2.2 represent decoded error rates when each of two cluster models are used to decode all the speakers' data. The "auto" column under CLS2 represents error rates when an automatically selected cluster model is used for test each speaker. The "cheat" column under CLS2 showed the error rate if a cluster is manually selected according to the lower error rate for each speaker. We are not satisfied by the automatic cluster selection results, because it is not always able to select a cluster that produces the lowest recognition error for a speaker. We have experimented with several different methods and different features for cluster selection.

The "rover" column in Table 2 showed the results when 3 decoded scripts generated by SYS2, CLS2.1 and CLS2.2 are post-processed by a voting program known as ROVER provided by NIST [3]. ROVER has enhanced speaker clustering significantly and has consistently produced much lower error rates than any automatic cluster selection algorithms and in some cases, it is also better than "cheating" results. Table 3 showed similar results for 4-cluster experiments. We are trying to implement multiple decoders to generate multiple decoding scripts for ROVER, for a real-time Communicator implementation.

In both Table 2 3, the feature vectors used for speaker clustering are MFCCs, with no CMN applied, as we know that cepstral mean normalization (CMN) reduces inter-speaker variability, in addition to eliminating unknown channel transfer functions [6]. We attempt to maintain all variabilities present. After speakers in training database being decided which cluster belong to, normal LDA transformed features are used to trained cluster models.

test set	SYS2	CLS2.1	CLS2.2	CLS2		
				auto	cheat	rover
NSTVAL^1	19.0	18.5	20.1	18.1	17.2	16.5
NSTVAL^2	17.9	17.6	20.0	17.2	17.0	16.1
DC1^2	26.8	27.1	28.4	-	-	24.5
DC2^2	12.0	12.6	12.4	-	-	10.7
DC3^2	28.2	27.4	29.8	-	-	27.1
DC4^2	17.0	16.5	17.0	-	-	15.5

Table 2: Word error rates in percentages for speaker clustering experiments: 2 Clusters, MFCCs without CMN

test set	SYS2	CLS4		
		auto	cheat	rover
NSTVAL^1	19.0%	18.3%	15.8%	16.3%
NSTVAL^2	17.9%	17.4%	15.5%	15.7%

Table 3: Speaker Clustering Experiments: 4 Clusters, MFCCs without CMN

In order to understand the unknown effects which exist in the telephony training data, we have been experimenting using different feature vectors to separate speakers in the training database. Such features include MFCCs after CMN (called CMN hereafter), MFCCs after CMN plus delta MFCCs (called DELTA), and MFCCs after CMN and after applying LDA transformation (named LDA).

In an effort to produce gender dependent speaker clustering, we investigated Vocal Tract Length (VTL) Normalization algorithm [11]. The frequency warping factor generated from the VTL algorithm can determine the speaker's gender with about 90% accuracy for randomly selected 100 speakers from the training database for listening experiment.

Unfortunately, none of these different features for separating speakers resultes speaker clustering system which produces very different recognition error rate when an automatically selected cluster is used for decoding, although some have much lower "cheat" error rate than others. Table 4 shows the 2-cluster systems where training speakers are separated using with different features.

test set	SYS2	MFCC	CMN	DELTA	LDA	VTL
NSTVAL[2]	17.9	17.2	17.4	17.7	17.6	17.4

Table 4: Word error rates for speaker clustering experiments: 2 Clusters, Various Features

5. Unsupervised Adaptation - Feature Space MLLR

MLLR [7], as a speaker adaptation algorithm is well known as one of the key techniques to reduce the recognition error rate. Normally in MLLR, a linear transform, which maximizes the likelihood of the acoustic data associated with an utterance with respect to a word hypothesis, is applied to Gaussian means and/or covariances. However, in real-time, on-line telephony applications, such as DARPA Communicator, adaptation on means and/or covariances may require too much overhead on computation and therefore resulting a delayed system response to a caller's query. We use feature space MLLR [10] which is a dual to the constrained MLLR [4]. The feature space MLLR transformation is applied to feature vectors themselves [10].

The word hypothesis used in feature space MLLR is the decoded script, which include recognition errors, thus the speaker adaptation is unsupervised.

We experimented three different ways of adaptation. In the first experiment referred as FMLLR0 in Table 5, one transformation is estimated for each speaker using all the test utterances from the same speaker and the assocaited decoded scripts, and then is applied to all utterances of the same speaker for re-decoding. The transformation is fixed. This is an ideal condition for recognition performance, but obviously impractical for real-time applications. The second experiment (FMLLR1) is to re-estimate the speaker's feature space transform every time a new utterance is decoded and then to re-decode the same utterance after the updated transformation is applied to the utterance. In the third experiment (FMLLR2), the speaker's transformation is also re-estimated every time an utterance is decoded, but the updated transformation is only applied to the next utterance received. No re-decoding of the same utterance is involved. The third senerio is very close to most real-time application conditions where the system response time is an important issue.

Table 5 showed progress made by the unsupervised adaptation. Even when no re-decoding is invloved, we still see substantial error reduction. It is useful and will be used with speaker cluster algorithm together to achieve more improvement.

	SYS2	FMLLR0	FMLLR1	FMLLR2
NSTVAL[1]	19.0%	15.8%	16.4%	18.1%

Table 5: Feature Space MLLR adaptation

6. Language Modeling

We improved the language model for DARPA communicator using various techniques as detailed below.

6.1. Compound Words and LM classes

Automatically generated compound words did not improve recognition rate as reported previously [1]. The most important information for the system is city names, dates and times. Confusibility between city names is a big source of error. Most of the time, when a caller calls the system, they say the city and state together one after the other, such as "LOS ANGELES CALIFORNIA". For common cities, the LM training data covers them and the LM score for the state following the city name is high. But, the training data is limited such that only a small portion of city names appear in it. To increase coverage of the language model, cities and states can be grouped into LM classes, say class [city] and class [state]. However, this causes a problem for word sequences in the form [city] [state], since due to lumping all cities and all states together in LM classes, any state can follow any city name with equal (or close) probability, causing recognition errors such as "SIOUX_CITY OHIO" instead of "SIOUX_CITY IOWA". To fix this problem, we formed compound words of the form "city_state" and put them in an LM class [city_state]. We did the same process for cities and airports and formed a [city_airport] class which contained word sequences like "NEW_YORK_LA_GUARDIA" and such.

Furthermore, we experimented with more compound words and LM classes that are appropriate for this system. Currently, we have 42 LM classes in our system with uniform or non-uniform within class probabilities. However, the biggest improvement was observed due to the [city_state] compound words and classes. The reduction in error rate is observed when we interpolate the compound word LM with a no-compound-word trigram LM with equal weighting between the two.

6.2. NLU State Feedback for Language Modeling

The dialog manager in the system can provide feedback to the recognizer about the state of the dialog. This information is useful, because it can be used to help in modeling the language of the user's response. For example, if the dialog manager is asking about a target destination, it is strongly expected that a city name will be in the response. Similarly, if a question about the departure time is being presented, the response most likely will contain time information. To use this information, we built language models specific to each state and interpolated these language models with the base LM to obtain an LM for each state. This is done since we do not have enough data for each state to train a separate LM and the answers do not have to comply with the questions all the time since this is an open dialog system. NLU state feedback for language modeling improves recognition rate about 10% relative.

It should be noted that, since our training data for each state was limited, it is better to use a bigram language model for each state in order not to overtrain the system. However, if enough data can be obtained for each state, a trigram LM can be trained.

6.3. Embedded grammar objects

The IBM DARPA communicator is a free-form dialog system, so direct application of grammars is not appropriate. However, within the utterances, there are well structured portions such as the parts that are describing the date, time and places. The IBM telephony engine has the recently added capability of using embedded grammar objects within the language models [8]. The portions of speech that matches a grammar are tokenized with a special token and the LM is trained with the new tokens in place. During decoding, if a word is hypothesized that occurs in the first position for the grammar, then the grammar is hypothesized and that path is scored using the grammar. This scoring is incorporated into our stack decoder. We formed embedded grammars corresponding to dates, times, cities and "city state" pairs, airports and airlines. Using embedded grammars reduced the error rate in some test sets, but did not help in all. Embedded grammars are also interpolated with a trigram language model to observe benefits.

6.4. Experimental Results

Most of the language model improvement came from compound words and design of LM classes. We conducted many tests with different ways of designing LM classes. There was also the decision of choosing whether to use uniform within class LM weights or assign non-uniform weights according to unigram counts or a combination of both. For example, we used population and airport volume for the cities to assign non-uniform weights within the city class.

The baseline language model ("old LM") was trained using 90K sentences from the travel domain and is the one used during the NIST evaluation in June 2000.

We added 10K more sentences to the LM training data, redesigned the classes and added new classes and compound words as described in Section 6.1. A few such LM's were designed. The best one ("new LM") is chosen among all.

Table 6 showes the overall progress on acoustic model and language model. The old AM and the new AM are SYS0 and SYS2, respectively, as defined in Section 3.1. The test set NSTVAL is used.

Acoustic Model	Language Model	WER
old AM	old LM	23.70%
old AM	new LM	20.63%
new AM	old LM	19.03%
new AM	new LM	17.79%

Table 6: Comparison between old and new, AM and LM. Overall a big reduction is shown.

The effect of the NLU state feedback and embedded grammars are shown in table 7. Here, we show the effects on 3 test sets. Again, the acoustic model used is SYS2.

It is shown that the NLU state feedback reduces the error rate by about 5-10%. For other sites, the NLU state is estimated from their system prompts (for testing purposes) using some rules since the real feedback information was not available. For the embedded grammar LM result in column three, no NLU state feedback is used. Embedded grammar increased error rate for IBM received calls, but reduced error rate for the other two sites. This could be due to the different nature of calls that were received by different sites. Some sites have a more directed dialogue system that encourages more structured responses which might explain why embedded grammars helped more for the other sites.

Language Model	NSTVAL	DC4	DC3
new LM	17.79%	19.70%	27.74%
NLU state feedback	16.91%	19.15%	27.45%
embedded grammars	19.19%	17.23%	27.23%

Table 7: Comparison showing LM with NLU state feedback and embedded grammars. Same acoustic model SYS2 is used throughout.

7. Conclusions

Our innovative work in acoustic and language modeling reduced IBM DARPA Communicator system speech recognition error rate a substantial amount.

Most of the reduction is obtained by carefully retraining acoustic and language models using domain specific data and cleverly increasing the LM coverage using classes. Speaker clustering is shown to improve the error rate significantly when combined with ROVER post-processing scheme. The implementation of feature space adaptation and speaker clustering are still pending.

8. Acknowledgements

The authors thank Peder Olsen, Karthik Visweswariah, Harry Printz, George Saon, Geoffrey Zweig, Etienne Marcheret, Andrezej Sakrajda, Mike Monkowski and Burn Lewis for their helps in making all the experiments and implementations possible during a short period of time. We also thank Xiaoqiang Luo, Adwait Ratnaparhki, Wei-Jing Zhu, Todd Ward and Salim Roukos for contributions to the DARPA Communicator system that served as our experimental testbed.

9. References

[1] A Aaron, et al, "Speech Recognition for DARPA Communicator", ICASSP2001.

[2] T Anastasakos, et al, "A Compact Model for Speaker Adaptive Training", ICSLP'96.

[3] J Fiscus, "A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)", Technocal Report, National Institute of Standards and Technology, 1997.

[4] M Gales, "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition". Technical Report CUED/F-INFENG, Cambridge University Engineering Department, 1997.

[5] Y Gao, et al, "Speaker Adaptation Based on Pre-Clustering Training Speakers", Proceedings of EuroSpeech'97.

[6] R Haeb-Umbach, "Investigations on Inter-Speaker Variability in the Feature Space", Proceedings of the ICASSP, pp397-400, 1999.

[7] C Leggetter, et al, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models", Computer Speech and Language, Vol 9, pp 171-185, 1995.

[8] M Monkowski, "Embedded Grammar Objects within the Language Models", IBM Technical Report, US patent pending, 2001.

[9] D.Povey, P.C.Woodland, "Frame Discrimination Training of HMMs for Large Vocabulary Speech Recognition", Proceedings of the ICASSP, pp333-336, 1999.

[10] G Saon, et al, "Linear Feature Space Projections for Speaker Adaptation", ICASSP2001.

[11] S Wegmann, et al, "Speaker Normalization on Conversational Telephone System", Proceedings of ICASSP, 1996.