SPEECH RECOGNITION FOR DARPA COMMUNICATOR

A Aaron, S Chen, P Cohen, S Dharanipragada, E Eide, M Franz, J-M Leroux, X Luo
B Maison, L Mangu, T Mathes, M Novak, P Olsen, M Picheny, H Printz, B Ramabhadran
A Sakrajda, G Saon, B Tydlitat, K Visweswariah, D Yuk

IBM Watson Research Center
PO Box 218
Yorktown Heights, NY 10598
`printz@us.ibm.com`

ABSTRACT

We report the results of investigations in acoustic modeling, language modeling and decoding techniques, for DARPA Communicator, a speaker-independent, telephone-based dialog system. By a combination of methods, including enlarging the acoustic model, augmenting the recognizer vocabulary, conditioning the language model upon dialog state, and applying a post-processing decoding method, we lowered the overall word error rate from 21.9% to 15.0%, a gain of 6.9% absolute and 31.5% relative.

1. INTRODUCTION

In this paper we report on experiments with the speech recognition module of a DARPA Communicator system. The aim of the Communicator project is to construct a computer system that plays the role of a travel agent speaking by phone with a customer. Ideally, this system will function just as a human would: conversing with the user to determine the outline of the desired itinerary, querying airline data bases to establish flight availability, reporting suitable flights to the user, answering questions to resolve uncertainties or misunderstandings, and finally booking the trip. More information about this task can be found in [1, 2, 3].

The system described here was not the one fielded by IBM in evaluations organized by the National Institute of Standards and Technology (NIST). Our objective was to serve as a prototyping platform for ideas under consideration for the fielded system.

We begin this paper by describing our experimental setup. Then we discuss our modifications to the acoustic portion of our system, notably the enlargement of our acoustic model, and the inclusion of more training data. Next we describe changes to the language model. We give results for an improvement in decoding technology, implemented here as a post-processing step on lattices generated by our familiar stack decoder architecture. Finally we apply some tuning and performance tweaks. By applying together all the techniques discussed in this paper, we succeeded in achieving significant reductions in the word error rate (WER). We close with some speculation on why these methods worked.

2. ARCHITECTURE AND EXPERIMENTAL SETUP

The architecture of IBM's DARPA Communicator system appears in Figure 1. It is similar to the one described in [4], though modified to be Galaxy-II compliant [5].

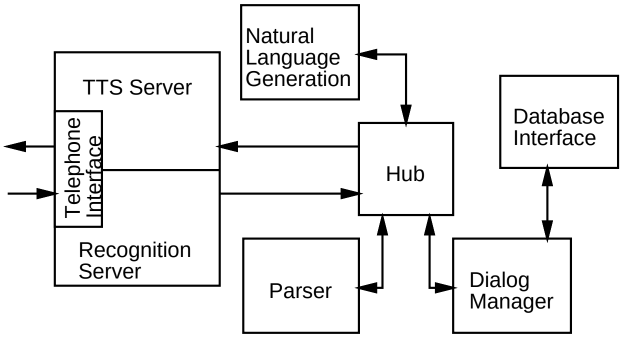

Fig. 1: DARPA Communicator Architecture

This paper concerns only the speech recognition module. But it is important to note that the dialog manager maintains a notion of current dialog state, and supplies this state to the recognition module on each conversation turn. This permits us to condition the language model (used in decoding the next user utterance) upon this state.

We used two test sets in our experiments. Both were collected from travel-domain human-computer interactions conducted via telephone. However, they were recorded under different conditions, and so we have treated them separately. (However, the performance figures in the abstract aggregate these two sets.) Since they differ by size as well, we will refer to them simply as SMALL and LARGE throughout. We made no attempt to identify or adapt to individual speakers or speaker clusters.

Test Set	Size	OOV by Voc	
	Sents / Words	Small	Big
SMALL	686 / 2932	1.59%	1.31%
LARGE	2082 / 8256	1.41%	1.08%

Table 1: Test Set Characteristics

3. ACOUSTIC MODELING

We experimented with four different acoustic models. All of them model 24-dimensional mel cepstra, and all were estimated from a common body of 600 hours of telephone-bandwidth (8 KHz) training data. However, as detailed below one model made use of an additional 20 hours of acoustic data for adaptation. The performance of these models is summarized in Figure 2; these results are for language model LM1, discussed below.

Our baseline model, denoted 40K, contained some 40,000 distinct Gaussians, with phone context determined by a decision tree conditioned on the five adjacent phones to the left and to the right. Although it was trained using telephone bandwidth data, this decision tree was built from wideband dictation data.

We felt that this mismatch of the data used to determine the tree structure and that used to train the system might be a source of errors, and also that 40,000 Gaussians might not adequately capture the large population of speakers and channel conditions present in our training set. For this reason, our first project was to rebuild the decision tree on 8 KHz data, and then use this tree to train a model containing 70,000 prototypes, and denoted 70K. This yielded a significant improvement on the small test set, and a barely measurable degradation on the large test set. We proceeded tobuild a third model in exactly the same way, but containing 280,000 prototypes, denoted 280K. This model achieved significant gains on both test sets.

Our final experiment was to try to take advantage of an additional 20 hours of telephone-bandwidth acoustic data, all of it comprising utterances within the domain of airline travel reservations. We felt that simply adjoining this data to our 600 hour training set would have little or no effect. Instead we treated it as if it were a body of adaptation data for a single speaker (in fact it contains a large number of speakers), and used it as such when applying the MLLR speaker adaptation technique [6]. Here the base model adapted by MLLR was a 40,000 prototype system built using the 8 KHz decision tree described above. The resulting model, named 40K MLLR, yielded a further improvement on the small test set, but the worst performance of all on the large test set.

4. LANGUAGE MODELING

We employed a variety of techniques to enhance our system's language model, including augmentation of vocabulary and training data, conditioning upon dialog state, and the introduction of compound words. Some but not all of these methods yielded accuracy improvements, as we detail below.

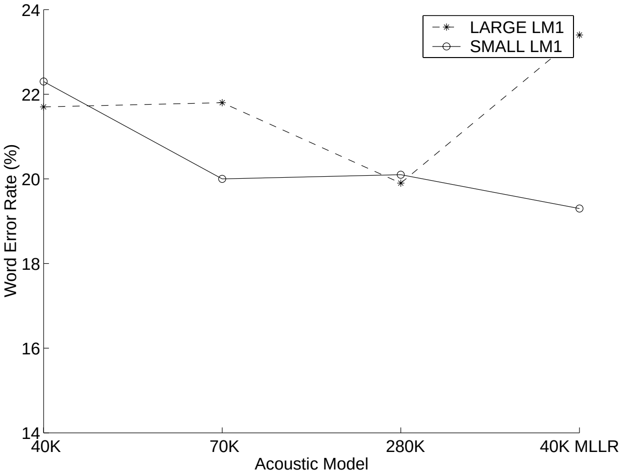

Fig. 2: Acoustic Model Performance.

The starting point for our experiments was a conventional linearly interpolated trigram language model [7, Chapter 4], built on a vocabulary of 2987 words comprising 4439 distinct baseforms, and trained on a corpus of 89,585 sentences comprising 646,012 words, plus another 300K words of synthetic data, to cover sparsely-represented place names. The training corpus was tokenized by a natural language understanding classer as described in [8]. We will refer to this model as LM1; decoding results appear in Figure 2

This vocabulary had out-of-vocabulary (OOV) rates of 1.59% and 1.41% on the SMALL and LARGE data sets respectively. Our next step was to enlarge the vocabulary, through the addition of selected airlines, airports and geographic locations. This yielded a vocabulary of 18866 words comprising 24913 baseforms, and reduced the OOV rates to 1.31% and 1.08% respectively.

Because the large number of proper names had very sparse representation in our training data (most did not appear at all), and we felt synthetic data was an inefficient means of dealing with this problem, we elected to build a class-based model, factoring the nominal language model probability as $p(w \mid h) = p(w \mid c\, h)p(c \mid h)$. Here w is the predicted word, c is the word's class, and h is a bigram of history. We then estimated $p(c \mid h)$ as a conventional ngram model, placing regular words into singleton classes. But we placed all place names into one of six classes. For each class set $p(w \mid c) = 0.25/|c| + 0.25 p_{\text{unigram}}(w) + 0.5 p_{\text{population/volume}}(w)$, where $|c|$ was the class size, p_{unigram} was a raw unigram model, and $p_{\text{population/volume}}$ estimated its probability from population or flight volume information. This gave us our second language model, LM2.

Prior experience had demonstrated to us the value of adapting the language model to the domain of discourse. Moreover the system's language generation module maintains a notion of the state of discourse, which corresponds to the nature of the utterance that is generated and ultimately played back to the user. The complete list of dialog states appears in Table 2 below.

State	Meaning
NEWCALL	initial state
TIMES	departure or arrival time
DATES	departure or arrival date
PLACES	departure or arrival city or airport
YN	yes / no question
LIST	list of choices
NONE	unknown or uncommitted state
DONE	itinerary complete

Table 2: System States and Putative Meanings

By conditioning the language model upon this state, we could achieve some degree of dependence upon the response likely to be triggered by the system's output. However, we were loathe to narrowly restrict the user utterances that would receive a non-zero language model probability, lest the user say something contrary to our expectations. Therefore our approach was to build a state-dependent language model $p_s(w \mid h)$ for each distinct state s, and interpolate each model with our baseline $p_{\mathrm{LM2}}(w|h)$, yielding a final model $p_{s\mathrm{LM2}} = \lambda_s p_s + \bar{\lambda}_s p_{\mathrm{LM2}}$ for each state. We then decode with interpolated model $p_{s\mathrm{LM2}}$ after the system has issued an output in a given state s.

To train the family of models $\{p_s(w|h)\}$ we used a corpus of 11,272 user utterances, comprising 52,217 words, collected from IBM-internal experiments. Each utterance was labeled with the system state that provoked the user's response. The interpolation weights λ_s were varied for each state s, but as we found this had little effect upon performance, we set $\lambda_s = 0.5$ uniformly. We will refer the the resulting family of state-dependent, linearly interpolated models as sLM2. Figure 3 compares decoding results with models LM1, LM2 and sLM2.

A final round of experiments revolved around the use of compound words. We had noticed early on that the word pair *to fly*, was frequently confused with the pair *a flight*. Although this confusion is harmless, there being little difference in meaning between *I'd like to fly to Paris* and *I'd like a flight to Paris*, our experience with voicemail transcription [9] led us to believe there might be some advantage in treating some word pairs as a single unit.

We tried three figures of merit for selecting word pairs $x\ y$ to treat as compound words, as follows

$$m_1(x,y) = \log\left(p(x,y)/p(x)p(y)\right)$$
$$m_2(x,y) = h(p(x,y)) - p(x)h(p(y|x)) - p(y)h(p(x|y))$$
$$m_3(x,y) = \log\left(p(x,y)/\sqrt{p(x)p(y)}\right)$$

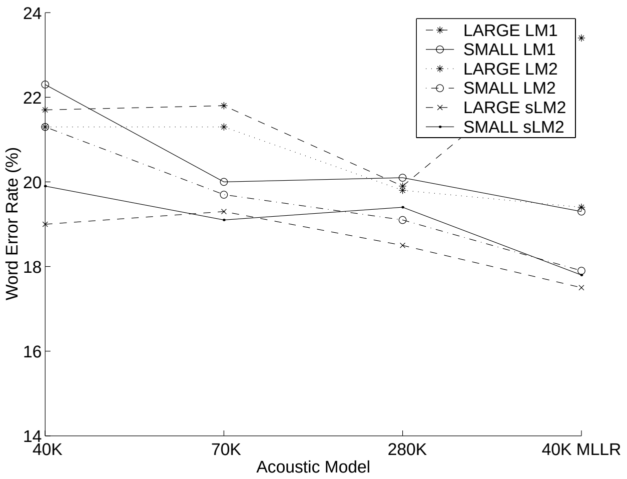

Fig. 3: Language Model: LM1, LM2 and sLM2.

$m_1(x,y)$ is the mutual information between words x and y; not to be confused with the average mutual information between random variables X and Y [7, Section 7.7]. $m_2(x,y)$ can be shown to be the perplexity gain (reduction in perplexity) that obtains for an empirical unigram model by treating the pair $x\ y$ as a single word throughout the training corpus. Here h is the binomial entropy function, $h(\alpha) = -\alpha\log\alpha - (1-\alpha)\log(1-\alpha)$. $m_3(x,y)$ has no simple interpretation, but it was found to be effective in prior studies of selection of compound words [9].

We selected the 50 most beneficial word pairs according to each of these criteria, then built standard linearly-interpolated trigram language models, where the selected word pairs were counted as single words throughout the training corpus. We then interpolated the resulting models with the standard language model, experimenting with weights of $\lambda = 0.5$ and 0.75 for the compound word model. Unfortunately, these experiments yielded either small WER gains or no gains at all.

5. DECODING TECHNOLOGY

We also experimented with a post-processing technique known as consensus decoding. The consensus decoding scheme processes word lattices, determined by a familiar stack decoder. (Formally the stack decoder produces a tree of hypotheses; branches of the tree are extended and merged to yield a lattice.) The consensus scheme performs further surgery upon the lattice supplied as input, until its topology is series of branchpoints connected by parallel arcs; at each branchpoint the word with the maximal posterior probability is selected for output. More detail on this method can be found in reference [10].

We found that including consensus decoding further reduced the error rate significantly. Figure 4 shows the effect of applying consensus decoding to lattices generated with language model sLM2. We refer to the consensus results obtained this way as sLM2+C.

6. ADDITIONAL TECHNIQUES

We experimented with two additional techniques. First, we added explicit acoustic models to cover mumble phonemes, which we expected to appear frequently in spontaneous speech. Second, we expanded the scope of the search that takes place in the initial phase of acoustic modeling, so that approximately twice as many hmm output densities would be examined for each frame of speech. Results for these techniques, labeled sLM2+C+MX, appear in Figure 4.

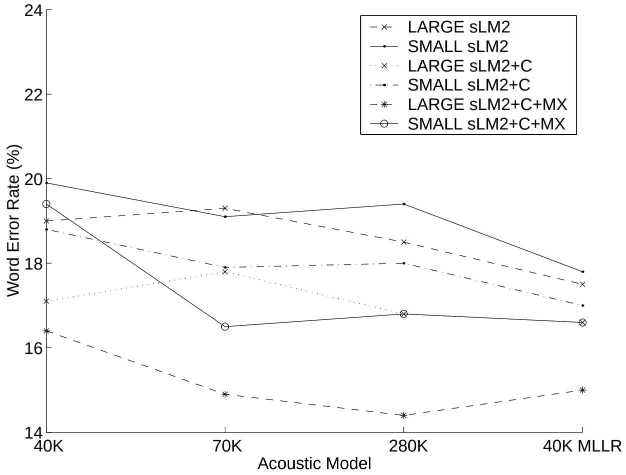

Fig. 4: Consensus Decoding, Mumble Model and Expanded Search.

7. SUMMARY

We have achieved a substantial reduction in word error rate on test utterances for a telephone-based dialog system. We believe that most of these gains are reflections of two special characteristics of dialog systems. First, recognition is performed within the context of a dialog, in which the user responds to known prompts or queries. Second, responses are likely to contain the usual flora and fauna of hesitations, mumbles and other disfluencies found in spontaneous speech. Thus techniques like consensus decoding (which stitches together disparate branches of the stack decoder's hypothesis tree) and explicit acoustic models of disfluency are likely to be useful.

8. ACKNOWLEDGEMENTS

We thank Scott Axelrod, Abe Ittycheriah, Kishore Papineni, Adwait Ratnaparhki, Salim Roukos and Todd Ward for contributions to the DARPA Communicator system that served as our experimental testbed.

9. REFERENCES

[1] E. Levin, S. Narayanan, R. Pieraccini, K. Biatov, E Bocchieri, G. Di Fabbrizio, W. Eckert, S. Lee, A. Pokrovsky, M. Rahim, P. Ruscitti, and M. Walker, "The AT&T–DARPA Communicator Mixed-Initiative Spoken Dialog System," in *Proceedings of ICSLP 2000*, Beijing, PRC, 2000, vol. II, pp. 122–125.

[2] Bryan Pellom, Wayne Ward, and Sameer Pradhan, "The CU Communicator: An Architecture for Dialogue Systems," in *Proceedings of ICSLP 2000*, Beijing, PRC, 2000, vol. II, pp. 723–726.

[3] Alexander I. Rudnicky, Christina Bennett, Alan W. Black, Ananlada Chotomongcol, Kevin Lenzo, Alice Oh, and Rita Singh, "Task and Domain Specific Modelling in the Carnegie Mellon Communicator System," in *Proceedings of ICSLP 2000*, Beijing, PRC, 2000, vol. II, pp. 130–133.

[4] K. Davies et al, "The IBM Conversational Telephony System for Financial Applications," in *Proceedings of EuroSpeech 1999*, 1999, vol. I, pp. 275–278.

[5] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, "Galaxy-II: A Reference Architecture of Conversational System Development," in *Proceedings of ICSLP 1998*, Sydney, Australia, 1998, vol. 3, pp. 931–934.

[6] C. Leggetter and P. Woodland, "Maximum likelikhood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, , no. 9, pp. 171–185, 1995.

[7] Frederick Jelinek, *Statistical Methods for Speech Recognition*, The MIT Press, Cambridge, MA, 1997.

[8] Xiaoqiang Luo and Martin Franz, "Semantic Tokenization of Verbalized Numbers in Language Modeling," in *Proceedings of ICSLP 2000*, Beijing, China, 2000, vol. I, pp. 158–161.

[9] G. Saon and M. Padmanabhan, "Data-driven approach to designing compound words for continuous speech recognition," in *Automatic Speech Recognition and Understanding*, Colorado, 1999.

[10] L Mangu, E Brill, and A Stolcke, "Finding Consensus in Speech Recognition: Word Error Minimization and Other Applications of Confusion Networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, 2000.