ISCA Archive
http://www.isca-speech.org/archive

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

TASK AND DOMAIN SPECIFIC MODELLING IN THE CARNEGIE MELLON COMMUNICATOR SYSTEM

Alexander I. Rudnicky, Christina Bennett, Alan W Black, Ananlada Chotomongcol, Kevin Lenzo, Alice Oh, Rita Singh

School of Computer Science, Carnegie Mellon University, Pittsburgh PA 15213 USA

{air,cbennett,awb,ananlada,lenzo,oh,rsingh}@cs.cmu.edu

ABSTRACT

The Carnegie Mellon Communicator is a telephone-based dialog system that supports planning in a travel domain. The implementation of such a system requires two complimentary components, an architecture capable of managing interaction and the task, as well as a knowledge base that captures the speech, language and task characteristics specific to the domain. Given a suitable architecture, the principal effort in development in taken up in the acquisition and processing of a domain knowledge base. This paper describes a variety of techniques we have applied to modeling in acoustic, language, task, generation and synthesis components of the system.

1. INTRODUCTION

System development involves a great deal of knowledge engineering, which is both time-consuming and requires a variety of experts to participate in the process. Therefore methods that seek to minimize this resource, for example through training based on domain-specific corpora are preferred. Effective use of corpora, however, requires developing techniques that intelligently make use of (typically) limited domain-specific resources.

2. THE CMU COMMUNICATOR

The Carnegie Mellon Communicator [8] is a telephone-based dialog system that supports planning in a travel domain. Currently the task is captured in an approximately 2700-word language based on corpora derived from human-human, wizard of oz and human-computer interaction. Domain information is obtained in real-time from sources on the Web. The system understands about 500 destinations worldwide, chosen on the basis of passenger statistics, with a concentration on North America. The system is available for public use, see http://www.speech.cs.cmu.edu/Communicator for details.

The system uses the Sphinx II decoder in a real-time mode, with post-recognition barge-in; state-specific language models are used to improve recognition [10]. The top hypothesis produced by the decoder is processed by the Phoenix parser using a domain-specific semantic grammar (based on ATIS [2] but extended to cover Communicator-specific language). The resulting parse is evaluated for coherence then passed to the AGENDA dialog manager [9]. Coherence is evaluated using goodness of the parse (features such as coverage and fragmentation) as well as word-level decoder confidence; inputs deemed incoherent. The system monitors the frequency and pattern of rejection and uses this information to modify its strategy for interaction. The parse result is treated as a set of concepts that individual *handlers* on the agenda consume. Once matched, the concepts are either used directly (i.e., to set a target value) or are first transformed through a call to a domain agent. Currently the system uses three major domain agents, a travel backend, a date-time module and a user profile module. The transform result is either stored in a product structure (for this domain, an itinerary) or an immediate action taken (for example, notifying the user of an error).

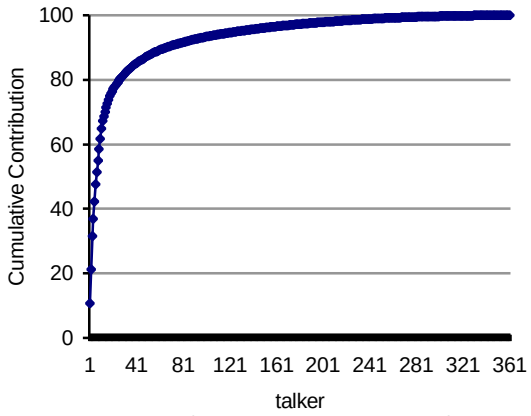

Figure 1 Percent of training corpus account for by individual talkers.

3. ACOUSTIC MODELLING

It is our belief that optimal recognition performance can be obtained most readily using domain-specific data. Therefore our acoustic modeling efforts concentrate on using Communicator-specific data as the core of the training corpus. There are unfortunately two difficulties with this approach. Most of the early data captured for training will be from a relatively small pool of developers; at the same time the rate of data acquisition will be slow. Figure 1 shows the distribution of data across speakers for the CMU Communicator (through the Fall of 1999). Note that although several hundred speakers are represented in the corpus, seven speakers contribute about half the data. Figure 2 shows corpus growth over time. In August the Communicator was publicized on the Web and made available for public use, increasing variability.

All models are 5-state semi-continuous HMMs, though the number of states differs as noted below. Model performance was evaluated using two different test sets, from June 1999 (1759 utterances) and from October 1999 (3459 utterances).

The June set contains predominantly (though not exclusively) developer speech, while the October test set contains a greater proportion of public speech, as well as more challenging data (e.g., from cell phones). For current purposes the two sets are best thought of as "easy" and "difficult".

Model 1: For training this model, we used all transcribed data collected between April 1998 and January 2000, excluding the

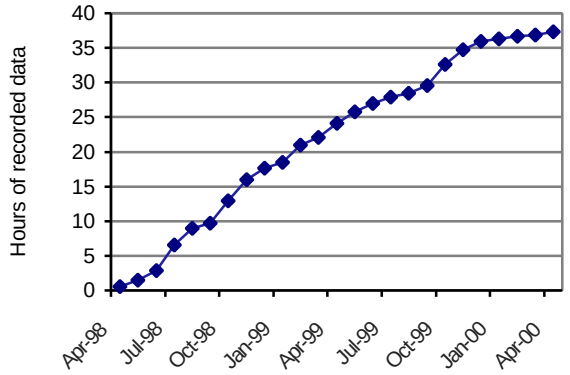

Figure 2 Available acoustic data, over time. Note that, the usable acoustic data is typically 10-15% less than the total available (due to noise-only utterances).

data collected during June 1999 and October 1999. Table 1 shows performance for a 4000 tied-state model. We observed that the triphone coverage of the corpus was rather sparse. For example, while there were 129516 possible triphones in the domain (computed from the dictionary used for recognition), only 15783 were present in the training data. To compensate for this, we identified all triphones (e.g., from city names), and created a list of these that appeared under-sampled in the training corpus. This in turn was used to design a set of 500 sentences densely sampling these triphones and recordings made, totaling 3141 utterances. The addition of such focused supplementary data resulted in a relative reduction in word error rates of less than 5% on the above test sets. While this is a modest improvement we believe that this technique can produce significantly better generality (as a function of corpus size) than unfocussed collection. We can compensate for lack of data and for overfitting by smoothing the state distributions of the models with uniform distributions. This produces a slight improvement in error rate for both test sets (**Model 1b**).

Model 2: We investigated state tying using a rule-based tying procedure prior to building decision trees. Decision trees were built using data recorded up to October 1999. The training data was observed to contain a large number of triphones with very poor representation. Since the data for these triphones was scarce, distributions learnt for these triphones, and decision trees built based on these distributions, were likely to be poorly estimated. In order to compensate for the under-representation of the triphones at this stage, states of triphones that represented similar transitions between phones were tied together in a preliminary state-tying step. The distributions of the various states of the triphones that were learnt in this manner were then used to build the final decision trees. Since this tends to merge the identity of triphones with similar transitions, the entropy of adjacent states was also considered during the decision tree building process to maintain triphone identities. Following this, HMMs with 4000 tied states were trained using all available data. Model 2 represents a significant improvement in accuracy.

	Test Set	
Model	June 1999	Oct 1999
1	13.5	21.0
1b	13.3	20.4
2	11.7	15.3
3	13.1	15.4
4	13.0	15.0

Table 1. WER obtained using different acoustic models

Model 3: It was observed that the performance of Model 1 on tracking test sets had steadily deteriorated in the Fall of 1999, due in part to a noisier signal. To avoid using such data to distribute parameters for training, decision trees were built using the ATIS corpus and pruned using the available Communicator data. Acoustic models with 5000 tied states were trained using these trees, and the same data as used by Model 2. This however did not appear to improve performance.

Model 4: We trained models 6000 state models in a standard fashion, using all Communicator data recorded through April 2000. Performance was better than Model 3 (likely due to simply more data) with better improvement for clean data over noisy data. This result contrasts with a 17.2% error rate observed for models trained on the much larger Switchboard I corpus and adapted to the Communicator domain (June 99 test set).

4. MODELING LANGUAGE

The basis for the Communicator language is the ATIS language developed previously for a similar domain (airline schedule information retrieval). Initial grammar coverage was quite low, however examination of utterances collected early on in the project yielded useful improvements. A corpus of 21890 transcriptions (from June 1998 through February 1999) was used for this purpose. This was reduced to 5162 sentences through preprocessing (essentially replacement of tokens by class type) then analyzed in order of frequency. For the June 1999 test set, the coverage error for an initial grammar (essentially ATIS with minimal additions to reflect new Communicator functionality) was 13.7% (10.4% for completely in-domain utterances). This was improved to 6.5% coverage error (3.3% for in-domain). The structure of a semantic grammar is such that, once a concept hierarchy is created, addition to language variants is a simple process. Although this process can be automated, we have not as yet done so.

It has been our experience that once the core of the language for a domain is identified, it remains stable as long as the definition of the domain is not significantly altered. This is due in part to the inherent stability of certain sub-languages such as that for dates and times as well as an apparent independence

between sub-domains comprising the full domain. Thus the addition of a hotel component to the domain does not materially impact the existing language for air travel. The significance of this observation is that it implies that the language for a domain can be incrementally extended without the concomitant need for restructuring the entire grammar as its complexity increases. It further raises the possibility that language components may be reused from application to application, provided that the sub-domains in question are substantially the same.

In our work we make a distinction between a *core* language and a *variable* component that includes domain-specific entities. In the case of Communicator, destination names and the names of users registering for the service. The latter components can be easily modified to accommodate evolution and do not appear to impact the core language. This is accommodated in language modeling through the use of a class language model. The Communicator core language contains a total of 1141 words; it uses a total of 20 classes, of which ten are open classes (e.g., city names or airlines) and ten are closed. Of the latter three are deemed closed with respect to the domain (e.g., holidays, ordinal numbers). There is a total of 1573 words in the class component of the language model.

A key issue in managing the knowledge base as a whole is coordinating modifications that impact different components of the base. For example, the introduction of a new destination (e.g., Tripoli) requires changes to the database (airport code, etc.), the dictionary, the "city" class in the language model and the definition of the corresponding city concept in the grammar. We have experimented with various techniques for automating this process. Nevertheless human intervention is required at two points: the identification of alternative renderings of a particular identifier (e.g., JFK as well as KENNEDY for an airport name) and the choice of a pronunciation. While it is possible to automatically generate pronunciations as well as variants, human review is always necessary to ensure accuracy. Moreover contact with informants (e.g., the named person, or someone knowledgeable about a particular region) is unavoidable, since many variants are culturally defined rather than generated by rule.

5. LANGUAGE GENERATION

To date, most of the research in natural language generation (NLG) has focused on generating text using template-based or rule-based (linguistic) techniques. In order to overcome some of the difficulties in using the current NLG technologies for spoken dialogue systems, we investigated a corpus-based approach to NLG [7]. The corpus used consisted of 39 dialogs between a professional travel agent and her clients, a total of 970 utterances and 12852 words. Utterances were classified into 29 categories, corresponding to major speech acts; within each utterance concepts were tagged as belonging to one of 24 classes. For each utterance category, a 5-gram language model was built then used for generation (the NLG component was provided with a speech act and a set of concepts to transmit to the user). About half of all utterances in the Communicator are generated using this stochastic technique (the remainder involve set phrases specific to the system). Several evaluations were performed, showing that stochastic generation produces output equivalent to, and in some cases judged better than, handcrafted templates. The advantage of stochastic generation derives from two factors: it takes advantage of the practiced language of a domain expert (rather than the intuition of the developer) and it restates the problem in terms of classification and labeling, which do not require the level of expertise customarily needed for building a rule-based generation system.

6. LIMITED-DOMAIN SYNTHESIS

The quality of speech output in a dialog system is important to a user's perception of the system. That is, it must both be appropriate sounding, and also work fast enough so that user does not think something is wrong. In earlier versions of the CMU Communicator we used a general-purpose commercial speech synthesis system. More recently we have begun to experiment with synthesis tailored specifically to this domain.

Unit selection synthesis, where appropriate sub-word units are selected from general speech databases, for example AT&T's NextGen [3], can produce very high quality synthesis. But the effort in building such general synthesizers is considerable. However it has been noted that unit selection synthesis is often better when the utterance to be synthesized is closer to the domain of the database. To take advantage of this observation, and with the intention of removing the bad selection examples for which unit selection synthesizers are, unfortunately also famed for, we used a limited domain synthesis technique where we record data specific in domain and ensure that the utterances to be synthesized are very close to that domain. The result offers very high quality synthesis that sounds almost human. The techniques used for this are more fully described in [4]. Important to this technique is not just the resulting high quality synthesis but that we also developed the tools, documentation and techniques to allow such high quality voices to be built for other systems reliably in a very short time (much less than a month). This voice was built in under a week.

The stages involved in building such limited domain synthesizers are as follows. First we constructed a set of sentences for recording which adequately covered the desired domain. For Communicator we analyzed the utterances from logs of the most recent three months and sorted them by frequency. We selected the most common phrases (around 100) that are effectively simple unchanging prompts e.g. "Welcome to the CMU Communicator", and "I'm sorry, I didn't understand that". We then took the basic templates used by the language generation system and filled them out with the most frequent cities, airlines and ensured we had full coverage (at least one instance) for numbers, dates, times and other closed classes in the system.

To provide coverage of over 400 cities and many airline names, we added around 500 sentences to our initial 100 "fixed" forms. Giving a prompt list of just over 600 utterances. These were then recorded in the style of a helpful agent. The

recordings were autolabelled using a simple alignment technique between the naturally spoken utterances and synthesized equivalents. The utterances were then used to build a unit selection synthesizer using the algorithm first described in [[5]]. This technique takes all units of the same type and calculates an acoustic distance between and then using CART techniques recursively partitions the instances by questions about phonetic and prosodic context to produce clusters indexed by decision trees.

In the original algorithm unit types are simple phones, but in this limited domain synthesizer we constrain this further by defining types as phones plus the word that phone came from. This apparently severe constraint allows the system to produce near perfect synthesis as the phones it selects always come from an instance of the word that is to be synthesized. This technique however is not just word concatenative synthesis. It is often, in fact common, that selections for a single word come different instances of that word joined at appropriate parts of the speech. The common prompts are invariably rendered from the original full prompts, thus preserving the original quality exactly. Other utterances with variable parts such as flights times, cities etc. are also rendered with comparable quality by selecting appropriate units from different utterances in the database. The stochastic language generation process described earlier is not a problem for this technique, reformulating similar sentence forms is dealt with adequately.

Of course although we have coupled the synthesis closely to the generation it is still possible that some words are generated which do not appear in the recorded database. Rather than falling back to more general unit types for selection, which could easily produce very poor quality synthesis (and which cannot be detected automatically) we use a standard diphone synthesizer (from the same voice as the limited domain speaker). After initially using the diphone synthesizer for the out of vocabulary word alone, it became obvious that the change in voice quality made that word very difficult to understand so if any word in a phrase is found to be out of vocabulary the whole phrase is synthesized with the backup diphone synthesizer this makes it easier to understand, even if it does reduce the overall quality. Over a period of three weeks the system synthesized 18,276 phrases, 459 of which (2.5%) contained out of vocabulary words (71 different words). These were all less frequent (or forgotten) places names.

This work was done within the Festival Speech Synthesis Systems [6]. And the code, documentation and tools for building such voices and the data used in this particular voice are available without restriction from http://festvox.org/ldom/.

7. DISCUSSION

The Carnegie Mellon Communicator system has provided a framework for experimenting with domain-specific, corpus-driven knowledge base configuration at different levels of a spoken dialog system. Corpus-driven techniques provide two benefits: they generally produce higher quality performance than general techniques, and they simplify the process of knowledge base development by reducing the level of expertise required since given the availability of basic algorithms that generate and make use of domain-specific models, human participation can be limited to corpus preparation which in turn can be codified in standard procedures.

8. ACKNOWLEDGEMENTS

We would like to thank Maxine Eskenazi and Karin Gregory for their work in managing corpus collection and transcription, as well as lexicon maintenance. We would like to thank Ricky Houghton for sharing some of his recognition results.

This research was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

9. REFERENCES

[1] R. Singh, B. Raj, and R. M. Stern, Domain adduced state tying for cross domain acoustic modelling, *Proc. Eurospeech*, 1999.

[2] Ward, W. and Issar, S. Recent improvements in the CMU spoken language understanding system. In *Proceedings of the ARPA Human Language Technology Workshop*, March 1994, 213-216.

[3] Beutnagel, M.,Conkie, A., Schroeter, J., Stylianou, Y. and Syrdal, A., The AT&T Next-Gen TTS system, *Joint Meeting of ASA, EAA, and DAGA*, Berlin, Germany, 18-24, 1999.

[4] Black, A. and Lenzo, K., Limited domain synthesis, *ICSLP2000*, Beijing, China, 2000, this volume.

[5] Black, A. and Taylor, P., Automatically clustering similar units for unit selection in speech synthesis, *Proceedings of Eurospeech*, 1999, Rhodes, Greece, 1997, 2, 601-604.

[6] Black, A. and Taylor, P. and Caley, R., The Festival Speech Synthesis System, http://www.cstr.ed.ac.uk/projects/festival.html, 1998.

[7] Oh, A. H. and Rudnicky, A. Stochastic language generation for spoken dialogue systems. *ANLP/NAACL Workshop on Conversational Systems*, May 2000, pp. 27-32.

[8] Rudnicky, A., Thayer, E., Constantinides, P., Tchou, C., Shern, R., Lenzo, K., Xu W., Oh, A. Creating natural dialogs in the Carnegie Mellon Communicator system. *Proc. of Eurospeech*, 1999, 4, 1531-1534.

[9] Xu, W. and Rudnicky, A. Task-based dialog management using an agenda. *ANLP/NAACL Workshop on Conversational Systems*, May 2000, pp. 42-47.

[10] Xu, W. and Rudnicky, A. Language Modeling for Dialog System, *ICSLP*, 2000, Beijing, China, this volume.