EXPERIMENTS IN AUTOMATIC MEETING TRANSCRIPTION USING JRTK

Hua Yu, Cortis Clark, Robert Malkin, Alex Waibel

Interactive Systems Laboratories
Carnegie Mellon University, Pittsburgh, PA, USA
Email: {hyu,cortis,malkin,ahw}@cs.cmu.edu

ABSTRACT

In this paper we describe our early exploration of automatic recognition of conversational speech in meetings for use in automatic summarizers and browsers to produce meeting minutes effectively and rapidly. To achieve optimal performance we started from two different baseline English recognizers adapted to meeting conditions and tested resulting performance. The data was found to be highly disfluent (conversational human to human speech), noisy (due to lapel microphones and environment), and overlapped with background noise, resulting in error rates comparable so far to those on the CallHome conversational database (40-50% WER). A meeting browser is presented that allows the user to search and skim through highlights from a meeting efficiently despite the recognition errors.

1. INTRODUCTION

Meetings, seminars, lectures and discussions represent verbal forms of information exchange that frequently need to be retrieved and reviewed later on. Human-produced minutes typically provide a means for such retrieval, but are costly to produce and tend to be distorted by the personal bias of the minute taker or reporter. To allow for rapid access to the main points and positions in human conversational discussions and presentations we are developing a meeting browser which records, transcribes and compiles highlights from a meeting or discussion into a condensed summary. The early experiments described here report on the particular problem of recognizing conversational speech in meetings and on the user interface of a meeting browser for later presentation.

We have recorded discussions of three or more participants. To minimize interference with normal styles of speech, we have ruled out the use of close talking microphones and recorded meetings with lapel microphones on two or more speakers. The resulting speech was found to be highly disfluent, similar to spoken telephone conversations as in the Switchboard and CallHome databases, and include many rare words and/or unusual language. The signal quality is further degraded by crosstalk between speakers and reverberation and echo due to the use of the omnidirectional lapel microphones.

2. MEETING TRANSCRIPTION EXPERIMENTS

Different from any other speech recognition task, our particular goal in this task is to improve the performance of existing recognizers on the meeting data, with NO additional training data. As it's not obvious which existing recognizer to start from, we tried a dictation system (Wall Street Journal system *WSJ*) and a spontaneous speech system (English Spontaneous Scheduling Task *ESST*). We first introduce the test data in Section 2.1, then describe in detail our experiments in Section 2.2 and Section 2.3 contains the results and discussion.

2.1. Testing Data

The test data is collected in an internal group meeting. 3 lapel microphones were given to 3 of the 10 participants. The meeting was approximately 1 hour in length, giving us 3 hours of speech to test. The 3 speakers consist of 2 females (referred to as flsl and fdmg), and 1 male speaker (referred to as maxl). The advantage of using lapel microphone is that the speaker can wear it in a pocket, not as intrusive as close-talk microphone. The disadvantage is of course, degraded sound quality. Since it's not uni-directional, there's significant channel mixing. There's also a lot of crosstalk, laughter, electric humming, paper scratching noise, etc. in the recording.

2.2. System Specification

Our system is built upon Janus Recognition Toolkit (*JRTk*), which is summarized in [1]. Incorporated into our continuous HMM system are techniques like linear discriminant analysis (*LDA*) for feature space dimension reduction, vocal tract length normalization (*VTLN*) for speaker normalization, cepstral mean normalization (*CMN*) for channel normalization, and wide-context phone modeling (Polyphone modeling). The speech feature used in ESST system is 24-order plp coefficient, in WSJ it's 48-order mel-spectrum. The exact configuration of the 2 systems are tailored to its respective task, summarized in Table 1:

Feature	*System*	
	ESST	WSJ
speech style	spontaneous	read
train data	26.5 hrs	83 hrs
front end	PLP (24)	mel-spectra (48)
# codebooks	1500	3000
# distributions	7000	3000
WER	20%	9%

Table 1: Distinguishing system features.

Note: WER (Word Error Rate) is measured with respect to its respective test set. For WSJ it's the official Nov.1994 evaluation set.

2.2.1. Language Modeling

The meeting task we're facing actually has no clear definition of the task domain, we knew nothing about the topic of a meeting in advance. Another way of saying is that we're facing a unrestricted domain, but meeting data clearly has its own characteristics: false start, broken sentence due to interruption, etc. So we built language models from several generic English corpus: Broadcast News (BN), Switchboard (SWB), and our own English Spontaneous Scheduling Task corpus (ESST), tested on the meeting transcripts. The results are shown in Table 2.

We find the SWB LM produces the best perplexity, which is reasonable because ESST doesn't have much coverage out of the scheduling domain, while BN can't handle spontaneous speech well, especially noise words, which constitute 16% of the meeting data. The unusually high perplexity for the BN model is caused by backing off to uniform distribution when predicting those noise words.

We could as well a build interpolated language model, but the lack of data was so severe that we couldn't even afford a reasonable sized cross-validation set to obtain a good interpolated model. So we decided to go with the SWB language model for the following experiments. As a first stage, we elected to use a closed-vocabulary system, that is, every word in the transcripts is included in the vocabulary.

Corpus	*Size (words)*	*Perplexity*	*Conversational*
BN	2.7MW	915.2	No
SWB	2.9MW	171.2	Yes
ESST	300KW	246.1	Yes

Table 2: Language Model Perplexities on Meeting Data

2.2.2. Acoustic modeling

As noted above, our task in acoustic modeling is to match existing acoustic models with testing data. Since both of the existing models were trained in a clean environment with close-talking microphone, while our data were recorded using non-close-talking microphones in a real world environment, there's significant mismatch between training and testing scenario. On the signal side, we used vocal tract length normalization (VTLN) for speaker normalization, cepstral mean subtraction (CMN) for channel normalization. On the model side, we used MLLR adaptation to move the model to fit data.

1. *MLLR* MLLR[4] has recently achieved much popularity as a dependable technique. In our system, we used a regression tree to define regression classes, the tree is constructed based on the criterion of acoustic similarity. Depending on the amount of adaptation data, the tree is pruned so that each resulting leaf node has enough adaptation data. For each leaf node we calculate a linear transformation to maximize the likelihood of adaptation data. In effect the number of transformations is automatically determined.

2. *Iterative batch-mode unsupervised adaptation* The quality of adaptation depends directly on the quality of hypothesis on which the alignment is based, i.e. the fewer errors there are in the hypothesis, the better we can adapt to the target speaker. Thus we can iterate the recognition - adaptation run several times, hoping that with better acoustic models given by the 1st adaptation pass, we can get better hypothesis, which leads again to better adaptation, better acoustic models, better hypothesis ... In our experiments we found significant gain in the first and second iteration, and the performance asymptotes quickly.

3. *Adaptation with confidence measures* In addition to using better hypothesis, another way to improve adaptation is to do adaptation only on the "better" portion of speech, i.e. the portion we feel confident the hypothesis is correct. We used a simple lattice-rescoring based confidence measure to "supervise" the adaptation. The confidence measure of a word is based on its acoustic stability. When rescoring the lattice with different language model weights and insertion penalty combination, we watch how consistently a word emerges in the top-1 hypothesis. If it always appears in the top-1 hypothesis, we have strong reason to believe it's acoustically stable, i.e. it's strongly backed up by the acoustic model, thus a good candidate to adapt on. This results in an overall 1-2% (absolute) gain in word accuracy over adaptation without confidence measure.

4. *Guided adaptation* To see how far away we still are from perfect adaptation, we conducted supervised adaptation with transcripts, as can be seen from the Table 5, there's still a lot of room for possible further improvements.

5. *Other experiments* As noted in [3], there're several variations to the basic cepstral mean normalization (CMN)

Speaker	*Adaptation Iterations*			
	0	*1*	*2*	*Adaptation Gain*
maxl	37.0	48.2	51.3	22%
fdmg	40.7	46.9	49.7	15%
flsl	20.8	32.3	33.3	16%
Total	32.6	42.5	44.8	18%

Table 3: Word Accuracy with ESST Acoustic Model

Speaker	*Adaptation Iterations*			
	0	*1*	*2*	*Adaptation Gain*
maxl	48.3	54.7	54.8	12%
fdmg	51.6	56.2	55.1	9%
flsl	36.2	40.5	40.4	7%
Total	45.2	50.4	50.1	9%

Table 4: Word Accuracy with WSJ Acoustic Model

method. In this experiment we find "global CMN"(cepstral mean of a set of utterances) is better than per utterance based CMN. The reason might be that global cepstral mean is more robust an estimate than per utterance based mean, since utterances are rather short in our data.

2.3. Results & Analysis

As expected, MLLR gives us considerable improvements for both the ESST system (Table 3) and the WSJ system (Table 4). It's also interesting to note how adaptation brought different initial acoustic models to a comparable level of performance on the testing data. Each model was developed to its best with regard to its own test scenario, but this is not necessarily the best criterion if we'll want use it in a different target domain. We have the feeling that future speech recognizers might have a compact generic acoustic model plus a powerful adaptation module that allows it to be used universally.

It's somewhat to our surprise that WSJ system had outperformed the ESST system, which we had felt to be more close a matched condition (it's spontaneous speech vs. read speech) at the first place. Upon closer examination, though, it turns out that the WSJ system has one very important strength that ESST system doesn't have: much more training data (83 hours vs. 26.5 hours), thus better polyphone coverage (in WSJ a minimum of 1000 training samples is required for each distribution). In fact we found words like "Japanese", "Recognition", "Analysis", which are not well represented in ESST domain, had great difficulty to show up in the hypothesis. They were often hypothesized as a sequence of short words that sounded very similarly.

Speaker	*Supervised Adaptation*
maxl	62.1
fdmg	61.0
flsl	48.3
Total	57.2

Table 5: ESST word accuracy with Supervised Adaptation

Also, the word accuracy varies widely among different speakers. We attribute this to personal speech styles, in our case speaker flsl is more spontaneous (fast) than others.

3. THE MEETING BROWSER INTERFACE

As noted in Section 1 above, we also require an interface with which to view and browse transcribed meetings. The interface we have created for this task is our Meeting Browser system, pictured in Figure 1.

The Meeting Browser interface displays meeting transcriptions, time-aligned to the corresponding sound files. The user can select all or a portion of these sound files for playback; text highlighting occurs in sync with the sound playback.

The Meeting Browser is built around information streams. Transcribed meeting text is just one such stream; the interface can accept streams from virtually any source which produces text output. These streams are fully editable and searchable, allowing humans to annotate and correct recognizer output as well as add new streams manually.

Since ultimately, the usefulness of a meeting transcription system is bounded by the usability of the interface, we feel that the flexibility present in the Meeting Browser is extremely important in user acceptance of the meeting recording and transcription process.

4. CONCLUSIONS AND FUTURE WORK

We have described our preliminary experiments in automatic meeting transcription as well as the interface we have designed for viewing and browsing transcripts. Early transcription experiments have been highly encouraging, and have helped us to identify the problems peculiar to meeting transcription.

In both experiments, we found that MLLR helps a lot in adapting a recognizer to new data, which is important to future application of speech recognizers in real world, because we can't afford building a recognizer for every domain. Though there's still some distance to go before we can reach the performance of matched condition system.

We find segmentation to be an important issue here. Currently, we're using a simple energy-based endpoint detection method. This method turns out to have produced overfragmented data, leading to recognition errors at the beginning and end of utterances. We feel we can define this issue

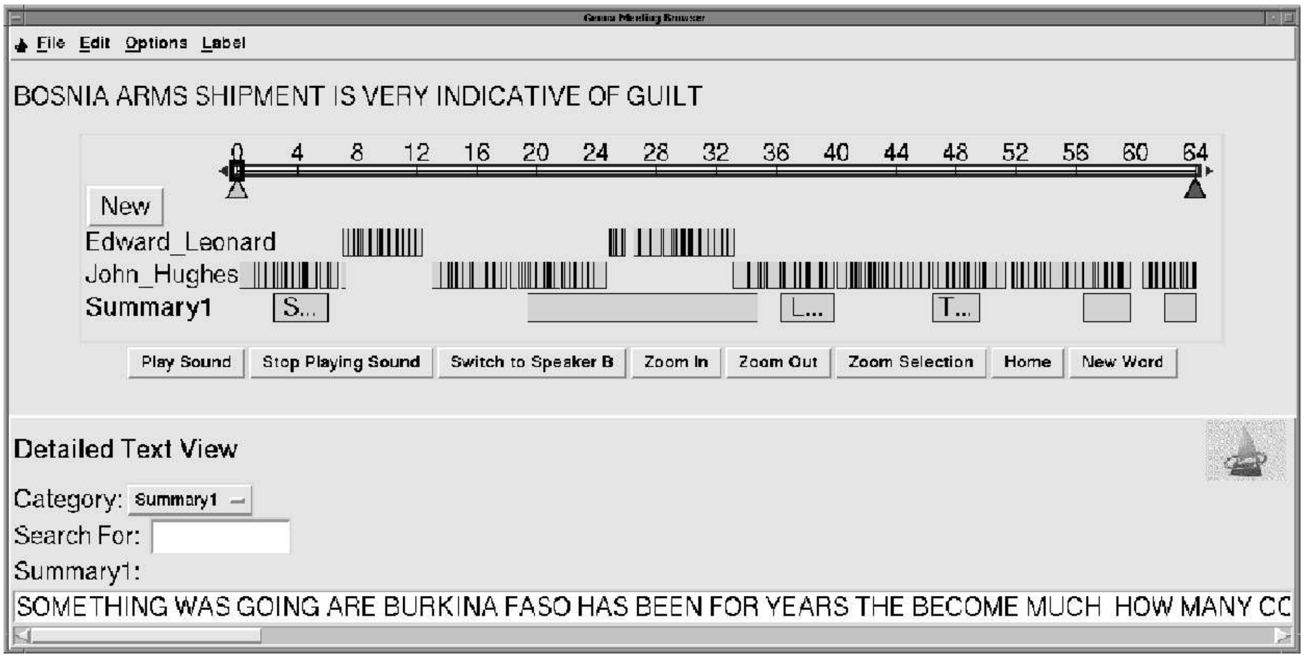

Figure 1: The Meeting Browser Interface

as part of the general problem of understanding the acoustic environment, whether it's noisy or clean, whether it's speech or not, whether there's crosstalk, etc.

Future work in meeting transcription will incorporate new methods to deal with these problems, as well as an expansion from meeting transcription to general meeting tracking and summarization, hopefully without the need for lapel microphones. We plan to combine the many sources of information present in a meeting setting, including speaker localization and channel separation using microphone arrays; face and gaze tracking to model who is speaking to whom; lip reading to aid speech recognition; and automatic summarization procedures, in order to produce an accurate summary of the events of a meeting with minimal human effort or supervision.

All of this information can be included in the streams passed to the Meeting Browser interface. This interface is being extended in numerous ways to increase usability and user acceptance, including security features to restrict access to portions of some streams and incorporating multimodal repair facilities [5] into the interface. We are also exploring ways to produce and include information describing the topical and discourse structure of a meeting, as well as multimedia presentations of such structures.

5. ACKNOWLEDGEMENTS

This research is sponsored by the Defense Advanced Research Projects Agency under the Genoa project, subcontracted through the ISX Corporation under Contract No. P097047. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA, ISX or any other party.

6. REFERENCES

[1] Finke, Michael, et.al. "The JanusRTk Switchboard/Callhome 1997 Evaluation System". *Proceedings of the LVCSR Hub5-e Workshop*, Baltimore, USA, 1997.

[2] Zeppenfeld, Torsten, et. al. "Recognition of Conversational Telephone Speech Using the Janus Speech Engine". *IEEE International Conference on Acoustics, Speech, and Singal Processing*, Germany, 1997.

[3] Westphal, Martin. "The Use of Cepstral Means in Conversational Speech Recognition". *Proceedings of Eurospeech Conference*, Greece, 1997.

[4] Zhan, Puming. "Speaker Normalization and Speaker Adaptation - a Combination for Conversational Speech Recognition". *Proceedings of Eurospeech Conference*, Greece, 1997.

[5] Suhm, Bernhard, et. al. "Interactive Recovery from Speech Recognition Errors in Speech User Interfaces". *Internation Conference on Spoken Language Processing*, Philadelphia, USA, 1996.