Lecture and Presentation Tracking in an Intelligent Meeting Room

Ivica Rogina, Thomas Schaaf
Interactive Systems Labs
Universitaet Karlsruhe
E-mail: {rogina|tschaaf}@ira.uka.de

Abstract

Archiving, indexing, and later browsing through stored presentations and lectures is a task that can be observed with a growing frequency. We have investigated the special problems and advantages of lectures and propose the design and adaptation of a speech recognizer towards a lecture such that the recognition accuracy can be significantly improved by prior analysis of the presented documents using a special class-based language model. We define a tracking accuracy measure which measures how well a system can automatically align recognized words with parts of a presentation and show that by prior exploitation of the presented documents, the tracking accuracy can be improved. The system described in this paper is part of an intelligent meeting room developed in the European-Union-sponsored project FAME (Facilitating Agent for Multicultural Exchange).

1. Introduction

While a decade ago, presentations that were using more than simple black and white overhead foils were rare, today, the standard way of presenting a lecture or a talk is by using powerful tools that help in designing a multimedia presentation which itself is presented in a room equipped with many supporting audio and video devices.

With both, the contents of lectures or presentation as well as the equipment in lecture halls and meeting rooms becoming more and more complex, the task of reducing the workload for the people giving presentation and the people listening is becoming more important. Archiving, indexing, and later browsing through stored presentations is another task that can be observed with a growing frequency. To address these problems, we have investigated the special problems and advantages of lectures. In this paper we propose the design and adaptation of a speech recognizer towards a lecture such that the recognition accuracy can be significantly improved by analyzing the presented documents and adapting the recognizer's vocabulary and special class-based language model accordingly. We use a tracking accuracy measure which measures how well a system can automatically align recognized words with parts of a presentation and show that by prior exploiting of the presented documents the tracking accuracy can be improved. The system described in this paper is part of an intelligent meeting room developed in the European-Union-sponsored project FAME (Facilitating Agent for Multicultural Exchange).

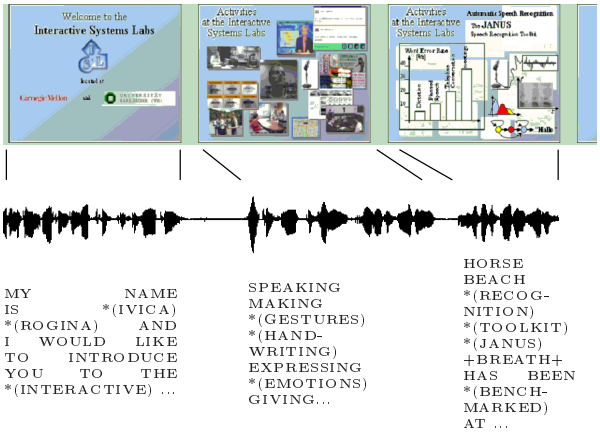


Figure 1: Synchronizing slides with recognized text

2. The FAME Room

Rather than an intelligent living room, the FAME room is more of an intelligent meeting room. In addition to the tasks performed by our previously presented meeting tracker systems [1][2][3][4], the FAME project foresees activities of the room during a meeting or lecture, namely to act as an information butler in the background. Meetings and lectures should be held as usual, only in cases where the participants explicitly or implicitly require additional information, the information butler becomes active.

The most important differences for the system between tracking a meeting and tracking a lecture are the quality of the speech acoustics, the availability of prepared documents (the presentation documents), and the speaking style. We can expect much more planned speech in a lecture than in a meeting.

The speech acoustics are easier for the recognizer for two reasons: first, the speech is more planned than in a discussion, and secondly, it's more reasonable to expect a lecture speaker to wear a head-mounted microphone or at least a lapel microphone. Although we have experienced rather significant differences in the audio quality of speech recorded with lapel microphones due to effects like acoustic shadow made by the speaker's chin or like rubbing the microphone with parts of the body or the clothing, the recognition accuracy is still better than with distant speaking table-mounted microphones. Although experiments [10] have shown, that using microphone arrays can improve the recognition of distantly recorded speech, in the FAME room, we will first focus on using microphone arrays only for localization of sound sources.

For later indexing and browsing of lectures and for transmitting the lecture via a video conference to audience distributed in different rooms, the lecturer is monitored by an automatic intelligent camera. The room is equipped with one or more cameras that are tracking the lecturer and the audience. The system can automatically decide which image is transmitted. For browsing purposes, in addition to the meeting browser [3], the lecture assistant will also synchronize the given talk with the presented slides (and, if available, with presented audio and video documents).

The two major tasks in assisting the lecturer consist of controlling the audio and video devices in the room and controlling the presentation. The former means turning on and off devices, dimming the lights, setting volumes of speakers etc., the latter means automatically selecting and displaying slides and optionally audio or video documents. Both services can be performed implicitly by the system "guessing" what is currently needed, or by having the speaker give explicit commands.

3. The Lecture Assistant

The tasks of the lecture assistant addressed in this paper are

- analysis of the presented documents
- related information retrieval from the internet
- adaptation of the vocabulary and language model and pronunciation lexicon
- speech recognition and lecture tracking

We will now describe these tasks in greater detail (see Fig. 2).

3.1 Analysis of the Presentation

The analysis of the presentation documents has two goals. One is to extract the important content words, the other is to retrieve an ASCII representation of the document which can be used to correlate it with the recognizer output such that the system always knows which part of the presentation the speaker is talking about.

The extracted words are compared with the recognizer's background vocabulary. A $tfidf$-value is computed for every content word. All out-of-vocabulary words (OOV) and the words with higher-than-average $tfidf$-values are considered to be important.

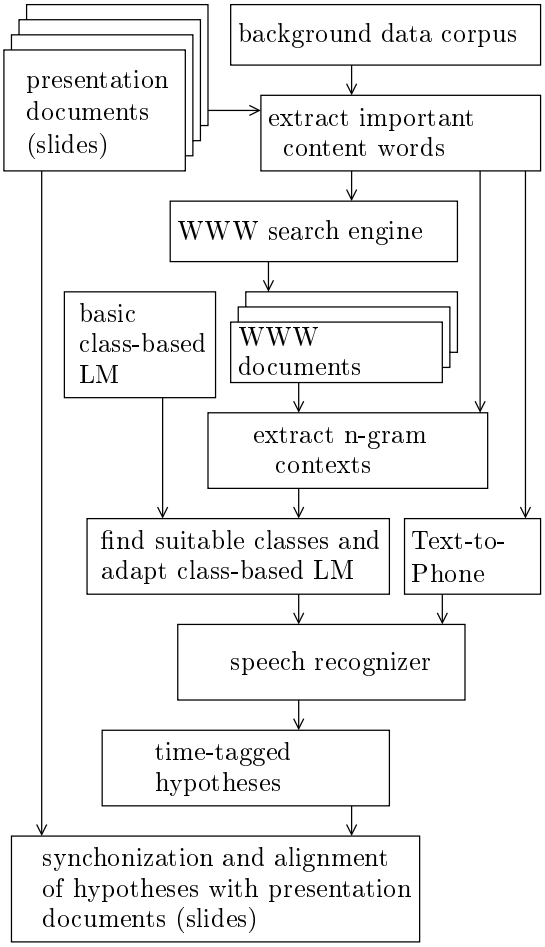


Figure 2: Overview of the lecture assistant

3.2 Document Retrieval from the Internet

Similar to the system described in [5], every important content word from the presentation documents is used for a simple singe-word query in a standard internet search engine, the top n (we used n=100 in our experiments) documents delivered by the search engine are analyzed, all quintuples of words where the middle word is an important content word are extracted and stored.

In addition, one search engine query is made with all important words with the expectation that the retrieved documents, especially the most highly ranked ones, have more relevance.

3.3 Adapting the Class-Based Language Model

The primary goal of the language model design was to allow easy adding of new words. Therefore a class-based design was chosen. The basis was a trigram model on a 40k vocabulary trained with the HUB-4 (broadcast news) training corpus. This language model did not contain any classes. To define classes that would be suitable to accept new OOV words, 20k more words from the HUB-4 vocabulary were taken. All sentences from the corpus which contained one of the additional 20k words were fed into a Kneser-Ney [9] bigram clustering algorithm. The result of the clustering process was a set of 2500 classes – representing the OOV-words from the point of view of the base 40k-language model.

The 10k most frequent words were removed from these classes. Then, only the set Z of classes that consisted of a sufficient number of words (minimum of $N = 50$) were kept (72 classes), all other classes were ignored and not used any further. The removing of the top 10k words was necessary to assure that these were not ignored together with the unused classes.

The resulting language model thus consisted of trigrams on the 40k most frequent words from HUB-4 plus 72 "OOV-classes". The model performed equally well, in terms of recognition accuracy, as a corresponding plain trigram.

Some of the 72 classes were built off words whose relation to each other is difficult to see, some classes could be clearly identified, (see example in figure 3).

In the class M30, which is displayed in Fig. 3, there are 56 non-OOV words plus one word named OOV_{30} representing all OOV-words (min. 50 words) assigned to M30 during the clustering process.

Now, the model was prepared to accept new OOV words by adding them into an appropriate class as follows.

```
CLASS:Z30 = { OOV30 ASTHMA TUBERCULOSIS
DIABETES POLIO PNEUMONIA DIARRHEA
CHOLERA RAPHAEL ALCOHOLISM HEPATITIS
MALARIA OBESITY MEASLES DEHYDRATION
SCHIZOPHRENIA INGENUITY NAUSEA ADVIL
MALNUTRITION ALLERGIES VALIUM UNTREATED
MELANOMA HERPES ACETAMINOPHEN DYSENTERY
ULCER SYPHILIS OSTEOPOROSIS COLDS
LONGEVITY VOMITING PEROXIDE FLASHBACKS
LIGGETT MENINGITIS ALS DIZZINESS
TREMORS INFLUENZA SOYBEANS INDIGESTION
DIPHTHERIA INSOMNIA NUMBNESS BULIMIA
DEMENTIA LUPUS SIRHAN MENOPAUSAL
AFFECTIONS PIMPLES MAIM MARKIE
CHLAMYDIA POLYGAMY }
```

Figure 3: An example language model class

Let v be a word that has to be added to the recognizer. Let $\Phi(w)$ be the index of the class to which the non-OOV word w belongs. For each class $c \in Z$, we define $\Phi_c^v(w)$ as:

$$\Phi_c^v(w) = \begin{cases} \Phi(w) & w \in \mathbf{V} \\ c & w = v \\ \Phi(\texttt{UNK}) & \text{else} \end{cases} \quad (1)$$

$$p(w|\Phi_c^v(w)) = \begin{cases} p(\text{OOV}_c|c) \cdot p(v|\text{OOV}_c) & w = v \\ \frac{\#(w)}{\#(\text{instances in } c)} & w \in \mathbf{V} \end{cases} \quad (2)$$

where

$$p(\text{OOV}_c|c) = \frac{\#(\text{OOV instances in } c)}{\#(\text{all instances in } c)} \quad (3)$$

and

$$p(v|\text{OOV}) := \frac{1}{\#(\text{OOV tokens in } c)} \quad (4)$$

is the approximation of the probability of observing the word v among the OOV words in c, assuming an equal distribution of all tokens.

$$\hat{C}_v = \text{argmax}_{c \in Z} \prod_j p(w_j|\Phi_c^v(w_j)) \cdot p(\Phi_c^v(w_j)|H)$$
$$\text{where } H = \Phi_c^v(w_1), \ldots \Phi_c^v(w_{j-1}) \quad (5)$$

Here, w_j denotes the j-th word of a text that was concatenated from all retrieved documents from the internet that contain the word v. While in eq. 5, H is generally the entire history of word w_j, we restricted the actual history to trigrams just like in the recognizer's language model.

Actually, it is not necessary to compute $\hat{C}_v$ over all retrieved text but we get the exact same result if we regard only the word v together with a sufficiently wide context. For training trigrams, it is enough to use a context of ± 2 words.

In our experiments, we computed not only the most suitable class as in eq. 5 but also computed the second and the third best class and added every content word to the top three classes. This approach was chosen because in such a rather small number of 72 classes, one single class can not cover all semantic properties of a word. Adding a word to all classes of the system did not improve the performance any further.

When new words were added to their classes, they were given an extraordinary high within class probability compared to the other OOV words in that class. This approach is justified by the expectance that the important words from the presentation documents are almost certainly to be spoken by the lecturer. The appearance of these words is much more likely than would be estimated from the retrieved documents.

What remains to be done when adding new words to the system, is to find an entry for the pronunciation lexicon. In our system, we look the word up in a large background lexicon. If it is not found in the lexicon a text-to-speech system [7] is used to provide a phone sequence.

3.4 Recognition and Tracking

A tracking system for a presentation or a lecture uses the same basic technology that most systems would use which have to monitor people in action and relate their actions (esp. their spoken utterances) with corresponding documents or parts of documents. A good lecture tracker can be used as a basis for a good meeting tracker in

Lecture	1	2	3	4	5
WER [%]	58.5	37.9	33.5	43.7	31.0

Table 1: baseline error rates on five lectures

Lecture	3	4	5
baseline	33.5%	43.7%	31.0%
system 1	28.1%	39.7%	29.8%
system 2	26.8%	37.8%	27.6%

Table 2: improvements by LM-adaptation

which several people talk about and work with a set of possibly shared documents.

The tasks performed by the lecture tracker during the lecture consist of recognizing the lecturer's speech, possibly switching slides and displaying documents (WWW, video, audio) when appropriate, and, in hopefully rare cases, interacting with the lecturer to resolve problems. While interaction is a problem that is part of the FAME project, we have not addressed it in this paper.

After the lecture, the recorded audio and the presented documents have to be aligned, indexed and stored, such that it will be possible to retrieve the documents and the audio recording of the speaker for later browsing.

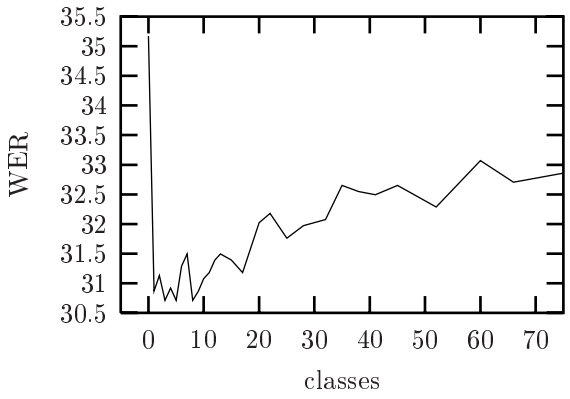
4. Experiments

We have conducted experiments on five self-recorded lectures. The speech recognizer used in our experiments is based on the JANUS speech recognition toolkit [6]. It was trained on the HUB-4 broadcast news training corpus and was used in other systems like [1][8]. The baseline word error rates for the five test lectures are shown in table 1. Lectures 1 and 2 were recorded with lapel microphones, lectures 3, 4, and 5 were recorded with a head mounted microphone.

We trained two systems, system 1 used only OOV words for adapting the language model and the vocabulary, system 2 also used important content words and treated them as if they were OOV. After adapting the language model, the word error rates were as shown in table 2.

The OOV-rate of the test set lectures is approximately 5%. So the adding of OOV words found in the presented documents to the recognizer's vocabulary is not sufficient to explain the significant gain in word accuracy. The other major contribution to the improvement comes from increasing the unigram probabilities for the important content words.

In our system, we treat every content word in the analyzed documents like an OOV-word. Even if they are already in the vocabulary, they are added to the three top-scoring classes (see eq. 5) as if they were OOV. In such an approach, the new words appear more than once in the vocabulary and can be regarded as homographs (two different words with the same spelling). It makes sense, if we consider that these words can very likely be used in completely different contexts with completely different meanings as in the text that was used to train the basic background language model. Using many more than three classes resulted in a increase of the word error rate as the following graph (averaged over lectures 3,4, and 5) shows:



In the synchronization experiments, every hypothesis word was automatically assigned a slide of the presentation and compared to the actual slide that was displayed at the corresponding time. To find a temporal alignment of the slides, we use a standard dynamic time warping algorithm which, for now, assumes that the slides are presented in a linear order in only one direction. Every word of the recognizer's hypothesized output is assigned a slide corresponding to the resulting DTW path. Local distances in the DTW-matrix, e.g. the distance between a hypothesis-word w_t and a slide s_i) are computed by comparing the content words of s_i with the context $w_{t-n} \dots w_t \dots w_{t+n}$ around the questioned word w_t. The inverse likelihood

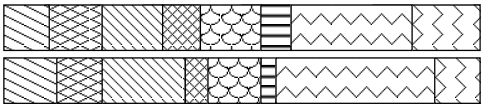


Figure 4: A typical alignment for a presentation of 8 slides (top = actual, bottom = predicted)

Lecture	3	4	5
baseline	34.7%	31.8%	32.1%
adapted	28.3%	31.0%	27.8%

Table 3: Improvements in tracking error rates

of a word from the context appearing on the slide is taken as the distance. We found that the DTW-path is rather stable and choosing $n = 3$ is sufficient. Increasing n did not result in higher tracking accuracies.

A zero knowledge approach, that would assign every slide a time slot of the same size would produce in our lectures a tracking error of 50-60%. The Tracking error rate improved as shown in table 3. Here we found that the tracking accuracy is only loosely correlated with recognition accuracy.

Of course, the tracking accuracy highly depends on the amount of information found on the slides. Presentations containing only headlines and images are much harder to track than presentations with lots of text.

5. Conclusion and Further Plans

We have shown, that it is possible to improve the speech recognizer's word accuracy significantly if we can used data from documents that a speaker plans to present during a lecture. We have defined and evaluated the tracking accuracy and have shown that this too can profit from prior exploitation of the presentation documents.

In the future, we plan to also spot explicit commands to switch slides or refer to specific slides either by naming them or their number or by addressing their contents. Also from human experience, we can expect some improvements in the tracking accuracy when we also take prosody into account, because many speakers lower their voice or make pauses in their talks when finishing a slide and switching to the next slide.

Acknowledgements

Part of this work was carried out within the FAME project and has been funded by the European Union as IST project No. IST-2000-28323.

6. References

[1] Alex Waibel, Michael Bett, Michael Finke, Rainer Stiefelhagen: "*Meeting Browser: Tracking and Summarizing Meetings*", Proceedings of the Human Technology Conference, San Diego 2001

[2] Tanja Schultz, Alex Waibel, Michael Bett, Florian Metze, Yue Pan, Klaus Ries, Thomas Schaaf, Hagen Soltau, Martin Westphal, Hua Yu, and Klaus Zechner: "*The ISL Meeting Room System*", Proceedings of the Workshop on Hands-Free Speech Communication (HSC-2001), Kyoto Japan, April 2001.

[3] Alex Waibel, Michael Bett, Florian Metze, Klaus Ries, Thomas Schaaf, Tanja Schultz, Hagen Soltau, Hua Yu, Klaus Zechner: "*Advances in Automatic Meeting Record Creation and Access*", ICASSP 2001, Salt Lake City

[4] Ralph Gross, Michael Bett, Hua Yu, Xiaojin Zhu, Yue Pan, Jie Yang, Alex Waibel: "*Towards a Multimodal Meeting Record*", ICASSP 2001, Salt Lake City

[5] Thomas Kemp and Alex Waibel: "*Reducing the OOV Rate in Broadcast News Speech Recognition*", Proceedings of the ICSLP 98,Sydney, Australia, 30th November-4th December 1998.

[6] Ivica Rogina and Alex Waibel: "*The JANUS Recognizer*", ARPA Workshop on Spoken Language Technology, 1995

[7] William M. Fischer: "*A Statistical Text-to-Phone Function Using N-Grams and Rules*", Proceedings of the ICASSP 99, Phoenix, Arizona, March 1999.

[8] Thomas Schaaf: "*Detection of OOV Words Using Generalized Word Models and a Semantic Class Language Model*", Proceedings of the Eurospeech 2001, Aalborg, September 2001

[9] Reinhard Kneser and Hermann Ney: "*Improved Clustering Techniques for Class-Based Statistical Language Modelling*", Proceedings of the Eurospeech 1993, pages 973–976

[10] Michael L. Seltzer, Bhiksha Raj, and Richard M. Stern: "*Speech Recognizer-Based Microphone Array Processing for Robust Hands-Free Speech Recognition* ",Proceedings of the ICASSP 2002