MULTIMODAL MEETING ANALYSIS BY SEGMENTATION AND CLASSIFICATION OF MEETING EVENTS BASED ON A HIGHER LEVEL SEMANTIC APPROACH

Stephan Reiter, Sascha Schreiber and Gerhard Rigoll

Institute for Human-Machine Communication
Technische Universität München (TUM)
Arcisstr. 21, 80290 Munich, Germany
email: {reiter, schreiber, rigoll}@ei.tum.de

ABSTRACT

This paper encompasses the analysis of meetings for a segmentation into sub-genres. Therefore an approach on a higher semantic level has been chosen. The algorithms make use of the results of specialized recognizers like a speaker turn detector and a gesture recognizer. Basically, the goal of this investigation was to answer the question, how well meeting analysis is possible if only the results of these recognizers are available. After introducing shortly the basics of these recognizers two slightly different methods for the segmentation are presented. The results show the potential of the used methods to find the segment boundaries and to categorize the detected segments into sub-genres (also called meeting events or group actions). Based on this segmentation further analysis regarding topic detection and content extraction can be accomplished.

1. INTRODUCTION

In everyday life of organizations meetings are an important part. Usually meeting minutes are taken in order to preserve the most important issues for those who were not able to attend the meeting. Nowadays much effort is being put into the generation of systems to automatically record, transcribe and summarize meetings, in order to enable persons who were not able to attend the meeting to get an overwiew of the topics that were discussed as well as the decisions that were made. With such an automatically generated meeting protocol it should be possible to obtain the relevant information about the meeting without the need to watch the whole video or listen to the entire recording. A number of groups are concerned with developing a meeting recorder or a meeting browser system. In the meeting project at ICSI [1], for example, the main goal is to produce a transcript of the speech. At CMU the intention is to develop a meeting browser, which includes challenging tasks like speech transcription and summarization [2] and the multimodal tracking of people throughout the meeting [3], [4]. Microsoft is developing a distributed meeting system that provides features like teleconferencing and recording of meetings [5]. In the European research project M4, in which this work is integrated, the main concern is the construction of a demonstration system to enable structuring, browsing and querying of an archive of automatically analyzed meetings. At one of the partner sites of the M4 project the human interaction is modeled by using a dynamic approach [6].

Due to the complex information flow of visual, acoustic and other information sources in meetings (e.g. from documents or projectors) the segmentation of a meeting in appropriate sections represents a very challenging pattern recognition task, which is currently investigated by only a few research teams.

In this paper we present a method to divide a meeting into meeting events like discussion, monologue, note-taking, white-board activities and presentations, using two different segmentation techniques, among others with dynamic programming.

The paper is organized as follows. Section 2 describes the meeting data. In Section 3 the low level algorithms are briefly presented. Section 4 then discusses the classification methods. Finally, the two segmentation approaches are presented in Section 5.

2. MEETING DATA

For our experiments with meeting segmentation and meeting event recognition special scripted meetings were recorded in the IDIAP Smart Meeting Room. This is a $8.2\,\mathrm{m} \times 3.6\,\mathrm{m} \times 2.4\,\mathrm{m}$ rectangular room containing a $4.8\,\mathrm{m} \times 1.2\,\mathrm{m}$ rectangular meeting table. The room is equipped with fully synchronized multichannel audio and video recording facilities. Each participant has a close-talk lapel microphone attached to his clothes. Additionally a microphone array on top of the table was used. Three closed-circuit television video cameras provide PAL quality video signals that are recorded onto separate digital video tape recorders. For full details of the hardware setup see [7].

The recorded meetings consist of a set of predefined meeting events in a specific order. The appearing meeting events were

- Monologue (one participant speaks continuously without interruption)
- Discussion (all participants engage in a discussion)
- Note-taking (all participants write notes)
- White-board (one participant at front of room talks and makes notes on the white board)
- Presentation (one participant at front of room makes a presentation using the projector screen)

A total of 53 scripted meetings with two disjoint sets of meeting participants were recorded. Each meeting has a length of about five minutes. The complete recording task is specified in [6].

The basic idea in this work is to take advantage of the results of single specialized recognizers. Currently available are the speaker turn detection and a gesture recognizer. The results of these recognizers are used to derive the higher semantic items, the sub-genres of the meeting, which we call meeting events.

0-7803-8874-7/05/$20.00 ©2005 IEEE

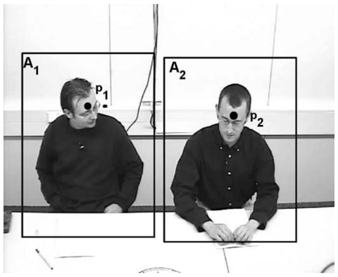

Fig. 1. Video frame marked with action regions and center of the head provided by the tracking algorithm

3. FEATURE EXTRACTION

This section illustrates in short the low level algorithms that provide the single actions of each meeting participant like speaker turns and various individual actions.

3.1. Speaker turn detection

The results of the speaker turn detection have been taken over from another partner in this international project. A generic, short-term clustering algorithm is used that can track multiple objects for a low computational cost. In [8] the three-step algorithm consisting in frame-level analysis, short-term analysis and long-term analysis is presented in detail.

3.2. Gesture recognition

Basically actions can be defined as movements in a certain surrounding of any person. In order to recognize actions one approach will be to extract features representing the motion in those surrounding areas. In [9], global motion features have turned out as suitable features, which can be computed as described by the following procedure. At first the difference image $I'_d(t)$ is built by subtracting image $I(t-2)$ from the actual image $I(t)$, followed by a threshold operation to reduce the noise. The resulting difference image I_d represents the motion in the whole image. Since we are only interested in the motion occuring in the nearer surrounding of the person, so called action regions A_i have to be defined as depicted in Fig. 1, consisting of a fixed sized rectangle. This subregion is always located relative to the position $\mathbf{p}$ of the person's head. For that reason a tracking algorithm has to be run in order to obtain the center of the head. Considering the difference image in the action region we are given a discription for the motions the respective participant of the meeting is performing. Features characterizing this subregion can be extracted by calculating the center of mass $\mathbf{m}(t) = [m_x(t), m_y(t)]$, the change of the center $\Delta\mathbf{m}(t) = [\Delta m_x(t), \Delta m_y(t)]$, the variance of motion $\sigma(t) = [\sigma_x(t), \sigma_y(t)]$ and the intensity of motion $i(t)$. This results in a 7-dimensional feature vector stream for each of the participants. To classify the actions the beginning and the ending has to be found and therefore the feature stream has to be segmented temporally. This step is actually assisted manually, but we also intend to deploy a Bayesian Information Criterion framework based on the approach presented in [10] to detect automatically boundaries for the actions in the feature stream vector. Finally these segments are fed to a HMM based recognizer which has been trained on roughly 1000 gestures consisting of *writing*, *pointing*, *standing up*, *sitting down*, *nodding* and *shaking head*. In Table 1 the recognition results are shown for a continuous HMM with 6 states and 4 mixtures.

	writing	pointing	standing up	sitting down	nodding	shaking head	Recognition rate [%]
writing	471	19	0	0	42	18	85.64
pointing	0	68	1	0	3	0	94.44
standing up	1	1	9	0	0	0	81.82
sitting down	0	0	2	7	0	0	77.78
nodding	8	7	7	0	225	43	77.59
shaking head	3	0	2	0	22	16	37.21

Table 1. Confusion matrix of single person action recognition (rows represent the truth, columns the recognizer output)

4. CLASSIFICATION OF MEETING EVENTS

The results of the recognizers described above can now be used to classify a temporal segment of a meeting into a meeting event as mentioned in Section 2. Thus, using this information, a static feature vector can be derived that contains the relative percentage of the various individual actions. We use for example the length of writing of a single person with respect to the whole considered time window. The same procedure applies for the remaining features like talking, nodding and so on.

For the classification of the meeting events we chose the following classifiers:

- a simple hybrid Bayesian Network (BN) consisting of a discrete node as parent with five states (one for each meeting event) and nine continuous nodes directly connected to the parent node, representing the nine dimensions of the feature vector,
- Gaussian Mixture Models (GMM) with various numbers of Gaussians depending on the number of training material,
- a Neural Net with Multilayer Perceptrons (MLP) with 3 layers,
- a Radial Basis Network (RBN)with maximum 10 neurons,
- Support Vector Machines (SVM) with RBF-Kernel.

Each of the classifiers has been trained with the meeting events of the 30 training meetings. For evaluation purposes the remaining 23 meetings were used. For the recognition task alone, where the segment boundaries are given, the MLP performs best and achieves a recognition rate of 95.90%. Two classifiers (RBN and SVM) yield a quite good result with 95.08% whereas the GMMs seem not to be able to adapt well enough and achieve a recognition rate of 70.61%. The Bayesian Network is somewhere in between with 93.44%. One cause of this difference may be the relatively small amount of training material available.

5. SEGMENTATION OF MEETING EVENTS

While segmentation of individual actions has been done manually as outlined in Section 3.2, an attempt has been made to automatically perform the segmentation of the meeting data into meeting events.

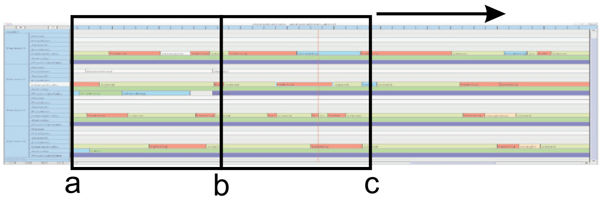

Fig. 2. Two connected windows are shifted over the time scale to produce potential boundaries.

5.1. Integrated approach

The integrated approach combines the detection of the boundaries and classification of the segments in one step. The strategy is similar to that one used in the BIC-Algorithm [10] and is illustrated in Figure 2. Two connected windows with variable length are shifted over the time scale. Thereby the inner border is shifted from the left to the right in steps of one second and in each window the feature vector is classified. If there is a different result in the two windows, the inner border is considered a boundary of a meeting event. If no boundary is detected in the actual window, the whole window is enlarged and the inner border is again shifted from left to the right. This procedure can be described by the following algorithm (a is the left border, b is the inner border, c is the right border of the window, L is the minimum length of a meeting event, $K(a,b)$ is the classification result of the interval $[a,b]$):

```
(1) initialize interval $[a,c]$:
    $a = 1$; $b = a + L$; $c = a + 3L$;
(2) if $K(a,b) \neq K(b,c)$ then
       save b as boundary
       $a = c$; $b = a + L$; $c = a + 3L$;
    else
       $b = b + 1$;
(3) if $(c - b) < L$
       $c = c + 1$; $b = a + L$;
       goto (2)
    else
       goto (2)
```

This algorithm is run until the right border c has reached the end of the video file.

5.2. Dynamic programming approach

Here the segmentation task is performed in two steps. At first, potential segment boundaries are searched; in the second step from all these possible boundaries those are chosen that give the highest overall score.

First the possible boundaries have to be found. Again two connected windows are shifted over the time scale as shown in Figure 2. This time the length of the windows remains fixed at 10 seconds each. Inside these two windows the feature vector is calculated and classified. If the results differ a potential segment boundary is assumed. In the same step a clustering of all found boundaries is performed. As long as the classification result $K(a,b)$ in the left window remains equal, the new assumed boundary is appended to the existing cluster $G\{i\}$. Otherwise a new cluster $G\{i+1\}$ is created. After that all clusters that contain less than three possible boundaries are discarded so that only important boundaries remain. Now we have a collection of arrays $G\{i\},\ i = 1,\ldots,N$, where N is the number of clusters, consisting in the potential boundaries.

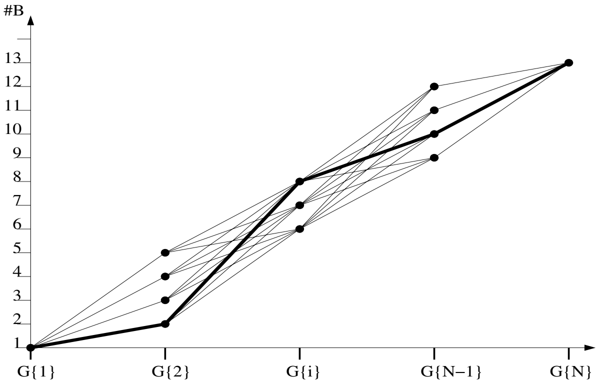

Fig. 3. Finding the optimal boundaries: the path with the highest overall score is found through backtracking. The abscissa denotes the clusters of potential boundaries, the ordinate the number of the boundary.

Having found all boundaries that come into question, in each cluster $G\{i\}$ the in some sense 'best' boundary has to be chosen. This is accomplished via Dynamic Programming (DP). This approach assumes that the meeting events are mutually independent. So each boundary of a meeting event can be found if only the direct predecessor is known. The first and the last boundary are known a priori (beginning and end of the meeting), so the task is to choose the remaining inner boundaries that give the highest overall score. The score of a meeting event is calculated as the pseudo-probability that the classifier returns for the examined interval. This could be for example the normalized probability of the GMM or the normalized output of the neural net. As additional constraint only those boundaries could be chosen that ensure a minimum length of a meeting event of 15 seconds.

In Figure 3 the procedure for finding the optimal segment boundaries is illustrated. For each boundary $x \in G\{i\}$ the score $s_x(y)$ to each boundary $y \in G\{i-1\}, \quad i = 2,\ldots,N$ is calculated. Then the maximum score $s_{max}$for each x is chosen.

$$s_{x,max} = \max\ s_x(y); \tag{1}$$

The sum of this score and the overall score until $i-1$ is calculated and saved in a score-matrix $SG\{i\}$ together with the predecessor y.

$$SG\{i\} = \begin{bmatrix} \vdots & \vdots & \vdots \\ x & s_{x,max} + SG\{i-1\}_{y,2} & y \\ \vdots & \vdots & \vdots \end{bmatrix}; \tag{2}$$

This is done for all clusters $G\{i\}$. Afterwards the best path through all score matrices is found through backtracking. Starting with the last score matrix $SG\{N\}$, which contains only one boundary, and following the indices in the third column those boundaries are chosen that produce the best overall score. In a completing step two segments that contain the same meeting event are merged.

This approach has the advantage of being computationally much less expensive, since there are much less segments to test due to the fixed length of the sliding windows.

Classifier	Insertion	Deletion	Accuracy	Error
BN	0.1474	0.0622	7.9316	0.3903
GMM	0.2475	0.0233	10.8718	0.4140
MLP	0.0861	0.0167	6.3326	0.3244
RBF	0.0689	0.0300	5.6654	0.3164
SVM	0.1779	0.0083	9.0838	0.3576

Table 2. Segmentation results using the integrated approach (BN: Bayesian Network, GMM: Gaussian Mixture Models, MLP: Multilayer Perceptron Network, RBF: Radial Basis Network, SVM: Support Vector Machines). The columns denote the insertion rate, the deletion rate, the accuracy in seconds and the classification error rate (see text).

Classifier	Insertion	Deletion	Accuracy	Error
BN	0.1650	0.0467	6.6667	0.3664
GMM	0.2971	0.0250	33.2812	0.4911
MLP	0.1871	0.0317	16.0696	0.3896
RBF	0.1738	0.0083	16.0127	0.3969

Table 3. Segmentation results using Dynamic Programming.

5.3. Segmentation results

From the 53 available meetings, mentioned in Section 2, 30 were chosen for the training of the classifiers, the remaining 23 were used for evaluation purposes.

The results of the segmentation are shown in Table 2 and Table 3 respectively (BN: Bayesian Network, GMM: Gaussian Mixture Models, MLP: Multilayer Perceptron Network, RBF: Radial Basis Network, SVM: Support Vector Machines). Each row denotes the classifier that was used. The columns show the insertion rate (number of insertions in respect to all meeting events), the deletion rate (number of deletions in respect to all meeting events), the accuracy (mean absolute error) of the found segment boundaries in seconds and the recognition error rate. In all columns lower numbers denote better results.

As can be seen from the tables, the results are quite variable and heavily depend on the used classifier. With the integrated approach (cf. Table 2) the best outcome is achieved by the radial basis network. Here the insertion rate is the lowest. The detected segment boundaries match pretty well with a deviation of only about five seconds to the original defined boundaries.

The results of the segmentation with dynamic programming were in general slightly worse. Due to the impossibility to get a score from the SVMs, these were not used here. Remarkable is the difference of ten seconds in the accuracy of the found boundaries between the Bayesian Network and the Neural Networks. The Bayesian Networks miss the given boundaries by 6.6 seconds on average. The neural network approaches make a greater mistake and produce a deviation of approx. 16 seconds.

6. CONCLUSIONS

In this work we presented a higher level approach for the automatic analysis of meetings and the segmentation of meeting events. Based on results of speaker turn detection and individual gesture recognition the proposed segmentation techniques provide a quite encouraging outcome. The recognition error could be decreased if more data were available. But even with the existing data segmentation results can be produced that can be used well for a subsequent recognition step.

The meeting, structured in this way, can then help to extract the most important events, or can assist further processing of the data, e.g. automatic topic detection and summarization algorithms.

7. ACKNOWLEDGMENTS

This work was partly supported by the EU 5th FWP IST project M4 (IST-2001-34485) and the EU 6th FWP IST Integrated Project AMI (FP6-506811, pub. AMI-11). We also thank IDIAP for providing the results of the speaker turn detection.

8. REFERENCES

[1] Nelson Morgan, Don Baron, Jane Edwards, Dan Ellis, David Gelbart, Adam Janin, Thilo Pfau, Elizabeth Shriberg, and Andreas Stolcke, “The meeting project at icsi,” in *Proceedings of the Human Language Technology Conference*, San Diego, CA, March 2001.

[2] Klaus Zechner, “Automatic generation of concise summaries of spoken dialogues in unrestricted domains,” in *Proceedings of the 24th ACM-SIGIR International Conference on Research and Development in Information Retrieval*, New Orleans, LA, September 2001.

[3] Michael Bett, Ralph Gross, Hua Yu, Xiaojin Zhu, Yue Pan, Jie Yang, and Alex Waibel, “Multimodal meeting tracker,” in *Proceedings of RIAO2000*, Paris, France, April 2000.

[4] Rainer Stiefelhagen, “Tracking focus of attention in meetings,” in *IEEE International Conference on Multimodal Interfaces*, Pittsburgh, PA, USA, October 14–16 2002.

[5] Ross Cutler, Yong Rui, Anoop Gupta, JJ Cadiz, Ivan Tashev, Li wei He, Alex Colburn, Zhengyou Zhang, Zicheng Liu, and Steve Silverberg, “Distributed meetings: A meeting capture and broadcasting system broadcasting system,” in *Proceedings of ACM Multimedia Conference*, 2002.

[6] I. McCowan, S. Bengio, D. Gatica-Perez, G. Lathoud, F. Monay, D. Moore, P. Wellner, and H. Bourlard, “Modeling human interaction in meetings,” in *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003.

[7] D. Moore, “The idiap smart meeting room,” IDIAP-COM 07, IDIAP, 2002.

[8] Guillaume Lathoud, Iain A. McCowan, and Jean-Marc Odobez, “Unsupervised Location-Based Segmentation of Multi-Party Speech,” in *Proceedings of the 2004 ICASSP-NIST Meeting Recognition Workshop*, Montreal, Canada, May 2004.

[9] Martin Zobl, Frank Wallhoff, and Gerhard Rigoll, “Action recognition in meeting scenarios using global motion features,” in *Proceedings of the Fourth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS)*, J. Ferryman, Ed., 2003.

[10] Alain Tritschler and Ramesh A. Gopinath, “Improved speaker segmentation and segments clustering using the bayesian information criterion,” in *Proceedings of EUROSPEECH*, 1999, pp. 679–682.