Multistream Dynamic Bayesian Network for Meeting Segmentation

Alfred Dielmann and Steve Renals*

Centre for Speech Technology Research
University of Edinburgh,
Edinburgh EH8 9LW, UK
Email:{a.dielmann,s.renals}@ed.ac.uk

Abstract. This paper investigates the automatic analysis and segmentation of meetings. A meeting is analysed in terms of individual behaviours and group interactions, in order to decompose each meeting in a sequence of relevant phases, named meeting actions. Three feature families are extracted from multimodal recordings: prosody from individual lapel microphone signals, speaker activity from microphone array data and lexical features from textual transcripts. A statistical approach is then used to relate low-level features with a set of abstract categories. In order to provide a flexible and powerful framework, we have employed a dynamic Bayesian network based model, characterized by multiple stream processing and flexible state duration modelling. Experimental results demonstrate the strength of this system, providing a meeting action error rate of 9%.

1 Introduction

Group meetings are part of many professional activities. Meetings are not only useful to plan work or to solve problems, but also to share knowledge between people and to promote good interpersonal relations. Since a large amount of information is generated during a meeting, automated systems to preserve and access meeting contents could prove to be invaluable [1].

Meetings may be successfully recorded using multiple cameras, microphones and other specialised multimodal recording equipment. However, without additional processing, the semantic content and the meeting structure remains locked into a number of distinct multimodal data streams. We are interested in the development of models able to discover meeting structure automatically through the analysis of such multimodal data. Our current work is mainly focused on the automatic segmentation of meetings into a set of actions or phases (*meeting actions*). Following Mc Cowan et al [2], we have defined a meeting as a sequence of basic group social actions, such as monologue, discussion and presentation.

Multiparty meetings are a good example of an interactive situation in which participants show both an individual behaviour and a joint group behaviour. We are interested in the automatic recognition of meeting actions which involve the whole group and are independent from who is attending the meeting. Thus we need to identify the set of

* Supported by EU IST project M4 (IST–2001–34485).

clues in both individual and group behaviours, and to highlight repetitive patterns in the communicative process. These may then be integrated into the abstract concept of meeting actions.

In this work we have been mainly concerned with multichannel audio data streams, from which we extracted a variety of features relating to prosody, speaker activity and lexical content. In section 2 we outline the IDIAP M4 Meetings Corpus used in this work. In section 3 we outline the feature sets that we have used to characterize multiparty meeting interactions, and we present some introductory results, achieved using these feature families and a simple hidden Markov model. We have modelled these data streams using multistream dynamic Bayesian networks (DBNs). We review DBNs and their graphical formalism in section 4, introducing the basic multistream DBN model, describing an extended and enhanced version and outlining how inference is performed. In section 5 we describe a set of meeting segmentation experiments on the IDIAP corpus, using these DBN models.

2 M4 Meetings Data Collection

Our experiments have been performed using the M4 corpus of 53 short meetings, recorded at IDIAP[1] using an instrumented meeting room [2]. These meetings all involved four participants, and were recorded using four lapel microphones (one for each participant) and an eight element circular microphone array placed on the table between the participants. In addition to the audio, video data was captured using three fixed cameras, and all recording tracks were time-synchronized. The recording conditions were realistic and without any constraint over factors such as noise, reverberation, cross-talk and visual occlusion.

The four participants in each meeting were chosen randomly from two independent sets of 8 people. Each meeting had a duration of about five minutes, resulting in a corpus of about four hours of multichannel audio/video recordings. For each meeting the sequence (and approximate timing) of meeting actions was defined in advance, with the meeting actions drawn from a dictionary containing the following: monologue (one for each participant), discussion, note taking, presentation, presentation at the white-board, consensus and disagreement. The dictionary of meeting actions was exhaustive and the individual actions were mutually exclusive, hence each meeting could be described by a sequence of non-overlapping group actions. On average, discussion and monologue were the most frequent actions, and also had the longest average duration. The mean number of actions per meeting was five.

These meetings are scripted at the level of the sequence of meeting actions, and are somewhat naïve from a social psychology viewpoint. However the acoustic and visual behaviours are natural and spontaneous, and this corpus provides a good resource for experiments to model higher level behaviours in terms of lower level signals.

[1] This corpus is publicly available from http://mmm.idiap.ch/

3 Features

The human communicative process behind a meeting is usually spread over a wide set of modalities, such as speech, gesture, handwriting and body movement. Not all modalities carry the same importance: for example, speech may be regarded as the most informative one. For this reason, we have based our initial efforts on speech and audio modalities, in particular features based on prosody, speaker activity and the lexical transcription. We are currently investigating the incorporation of streams based on video features to the models described in this paper.

3.1 Prosody

Prosodic features were extracted from the four audio channels associated with individual lapel microphones. We computed three feature streams:

- Baseline pitch: based on a rough intonation contour estimate, obtained using the ESPS pitch extraction algorithm, then denoised with a histogram filter and a median filter, and stylised with using a piecewise linear interpolation [3];
- Rate of speech: an estimate of the syllabic rate of speech using the multiple rate (MRATE) estimator [4]
- Energy: root mean square value of the signal energy.

These acoustic features appeared as four feature sets (one per channel) with three features each, or as a 12-dimensional feature vector. In order to cope with the high level of cross-talk between audio channels, each feature set was forced to zero if the corresponding speaker was not active. Individual speaker activities were evaluated using a speaker localization process applied to the eight-channel microphone array. The whole prosodic feature set highlights the currently active speakers, and may indicate the level of engagement in the conversation for each participant.

3.2 Speaker activity features

Microphone arrays can be used to simulate steerable directional microphones, enabling the estimation of sound source directions (*localization*) and the algorithmic steering of the array to improve sensitivity in a given direction (*beamforming*). In the M4 data collection, meeting participants tend to occupy only a restricted set of spatial regions i (their seats $i = 1,...,4$, a presentation space $i = 5$, and the whiteboard area $i = 6$). We predefine these spatial regions and collect sound source activations from each region, to give an estimate of speaker activity in that region [5]. For example a high sound activity $L_3(t)$, from the region around seat 3, means that the participant number 3 is probably speaking. Information about speaker activities taking was extracted, building up a 216-dimensional feature vector, whose elements corresponded to the 6^3 possible products of "sound activities" $L_i(t)$ evaluated at the 6 most probable speaker locations, during the most recent three frames [6]:

$$S_{ijk}(t) = L_i(t) \cdot L_j(t-1) \cdot L_k(t-2) \quad \forall i,j,k \in [1,6]$$

A speaker activity feature vector at time t thus gives a local sample of the speaker interaction pattern in the meeting at around time t.

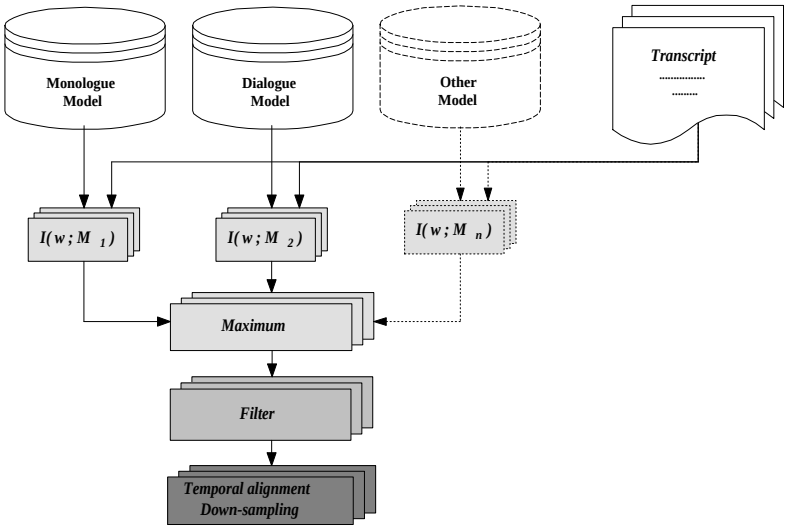

Fig. 1. Overview of the lexical feature generation process

3.3 Lexical features

In addition to the paralinguistic features outlined above, we also used a set of lexical features extracted from the word-level transcription. A transcript is available for each speaker, resulting in a sequence of words. In these initial experiments, we have used human generated transcriptions, however work in progress is employing speech recognition output.[2]

In this work we have used lexical features to discriminate between monologues and discussion. Our approach (outlined in figure 1) is based on unigram language models with a multinomial distribution over words used to model the monologue class M_1 and the discussion or dialogue class M_2; this approach could be extended easily to the other meeting actions. The sequence of words (from the transcript under test) is compared with each model M_k, and each word w is classified as a member of the class $\tilde{k}$ which provides the highest mutual information $I(w;M_k)$:

$$\tilde{k}(w) = \arg\max_{k \in K} \{I(w;M_k)\}$$

The sequence of symbols $\tilde{k}$ is very noisy, and the true classification output is hidden by a cloud of mis-classified words. To address this drawback we compute a smoothed version of $\tilde{k}$, that uses only the most frequent symbols. This is achieved by computing the relative symbol frequencies of $\tilde{k}$ across a sliding window of 24 words, and taking only symbols with higher frequency. After filtering in this way, these lexical features may

[2] The M4 corpus is a challenging speech recognition task, due to its conversational nature, the high percentage of non-native accents, and the degraded acoustic quality arising from the fact that head-mounted microphones were not used.

Feature	Accuracy
Prosodic features	50.0
Speaker activity features	65.4
Lexical features	58.3
All 3 feature groups	**70.5**

Table 1. Accuracy (%) of a simple HMM based meeting action recognizer using only one feature set at a time, or all 3 sets together. Higher values means better performances.

be used to label monologues and discussions, taken from unseen hand labelled transcriptions, with an accuracy of 93.6% (correct classified words). The resulting symbol sequence is then translated from the discrete word level temporal scale, into the same frame rate used for the prosodic and speaker activity features.

3.4 Some experimental results about features

Each feature class has its own temporal scale and its individual sampling frequency. In order to share a common sampling frequency, all three features groups were downsampled to a sampling frequency of 2Hz.

To compare the different feature families, we used a baseline hidden Markov model (HMM) approach to segment the meetings into sequences of meeting actions. Each meeting action was modelled using an 11-state HMM, and we experimented with observations consisting of each one of the feature streams, and all features combined. We used the 30 meeting training set for these experiments, using a leave-one-out cross validation procedure. The results are shown in table 1. Considering the models trained on each the three feature streams independently, it is clear that the speaker activity features result in the most accurate classifier (65% of actions correctly recognized), with the prosodic features resulting in a model with only 50% of actions correctly classified. The lexical features, which offer 93% correct classification between monologue and discussion result in an HMM with an overall accuracy of 58.3% when all actions are considered (monologues and discussions cover about the 60% of the meeting corpus). When all these features are merged into a single feature vector, the number of correctly recognized actions rises to 70.5%, indicating that the different feature families supply non-redundant information that the HMM can exploit.

4 Dynamic Bayesian Networks

Bayesian Networks (BNs) are directed acyclic graphical models, in which the network topology represents statistical relationships among variables [7]. In the BN graphical formalism, nodes represent random variables, and arcs represent conditional dependencies. Thus directed arcs between nodes depict the influence from each variable to the others, and the lack of direct and indirect connections represents a conditional independence relationship between variables. The generalization of BNs to dynamic processes are usually referred as Dynamic Bayesian Networks (DBNs) [8, 9]. In a DBN the time

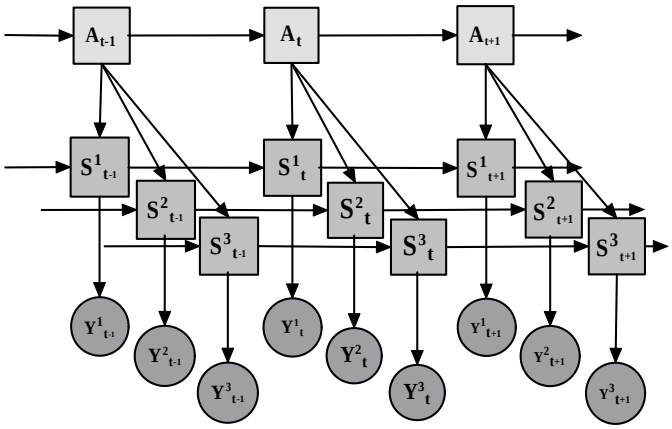

Fig. 2. Multistream DBN model ; square nodes represent discrete hidden variables and circles must be intend as continuous observations

flow is discretized, and a static BN is assigned to each temporal slice. Variables of different time-slices are connected through directed arcs, which explicitly represent the time flow in terms of conditional dependences. DBNs are a powerful mathematical formalism, able to group together a large variety of statistical models such as HMMs, hierarchical HMMs, input-output HMMs, factorial HMMs, and Kalman filters [10].

4.1 Multi stream DBN model

The DBN formalism allows the construction and development of a variety of models, starting from a simple HMM and extending to more sophisticated models, with richer hidden state. Among the many advantages provided by the adoption of a DBN formalism, one benefit is the unequalled flexibility in the model internal state factorization. With a small effort, DBNs are able to factorize the internal hidden state, organizing it in a set of interconnected and specialised hidden variables.

Our model (figure 2) exploits this principle in two ways: decomposing meeting actions into smaller logical units, and modelling the three feature streams independently. We assume that a meeting action can be decomposed into a sequence of small units: meeting subactions. In accordance with this assumption the state space is decomposed into two levels of resolution: meeting actions (nodes A) and meeting subactions (nodes S^F). Note that the decomposition of meeting actions into meeting subactions is done automatically through the training process. These synthetic subactions do not necessarily have a clear human interpretation.

Feature sets derived from different modalities are usually governed by different laws, have different characteristic time-scales and highlight different aspects of the communicative process. Starting from this hypothesis we further subdivided the model state space according to the nature of features that are processed, modelling each feature stream independently—a multistream approach. The resulting model has an independent substate node S^F for each feature class F (prosodic features, speaker activities,

and lexical features), and integrates the information carried by each feature stream at a 'higher level of the model structure (arcs between A and S^F,$F = [1,3]$). The joint distribution for a sequence of T temporal slices is:

$$P(A_{1:T}, S^1_{1:T}, S^2_{1:T}, S^3_{1:T}, Y^1_{1:T}, Y^2_{1:T}, Y^3_{1:T}) = \\ P(A_1) \cdot \prod_{F=1}^{3} \left\{ P(S^F_1 \mid A_1) \cdot P(Y^F_1 \mid S^F_1) \right\} \cdot \prod_{t=2}^{T} \{ P(A_t \mid A_{t-1}) \cdot \\ \cdot \prod_{F=1}^{3} \left\{ P(S^F_t \mid S^F_{t-1}, A_t) \cdot P(Y^F_t \mid S^F_t) \right\} \} \quad (1)$$

Each substate node S^F,$F = [1,3]$ follows an independent Markov chain, but the substate transition matrix and an initial state distribution are functions of the action variable state $A_t = k$:

$$\tilde{A}^F_k(i,j) = P(S^F_t = j \mid S^F_{t-1} = i, A_t = k) \quad (2)$$

$$\tilde{\pi}^F_k(j) = P(S^F_1 = j \mid A_1 = k) \quad (3)$$

where $\tilde{A}^F_k(i,j)$ is an element of the transition matrix for subaction S^F_t given that the meeting action variable ($A_t = k$) is in state k, and $\tilde{\pi}^F_k(j)$ is the initial substate distribution for the stream F, given k as initial action ($A_1 = k$). The discrete substates S^F generate the continuous observation vectors Y^F through mixtures of Gaussians.

The sequence of action nodes A form a Markov chain with subaction nodes S^F, $F = 1,2,3$ as parents. Hence A generates three hidden subaction sequences S^1 , S^2 , S^3 through $\tilde{A}^1_k(i,j)$, $\tilde{A}^2_k(i,j)$ and $\tilde{A}^3_k(i,j)$ respectively. Like any ordinary Markov chain A has an associated transition matrix $P(A_t = j \mid A_{t-1} = i) = A(i,j)$ and an initial state probability vector $P(A_1 = i) = \pi(i)$. A has a cardinality of 8, since there is a dictionary of 8 meeting actions. The cardinalities of the subaction nodes are part of parameter set, and for all our experiments we chose the following values:

$$\left|S^1\right| = 6 \ , \ \left|S^2\right| = 6 \ , \ \left|S^3\right| = 2 \quad (4)$$

Considering only one feature stream, our model apparently looks like a hierarchical HMM, but here A is free to change independently of the state of S^F: there is no feedback from S^F to A enabling state transitions of A only when S^F is in a terminal state [11, 12].

4.2 Counter Structure

In an HMM, the probability of remaining in a given state decreases following an inverse exponential distribution [13]. This distribution is not well-matched to the behaviour of meeting action durations. Rather than adopting ad hoc solutions, such as action transition penalties, we preferred to improve the flexibility of state duration modelling, by enhancing the existing model with a counter structure (figure 3). This additional structure is composed of a Markov chain of counter nodes C, and a set of binary enabler variables E. The counter variable C, being ideally incremented during each action transition, attempts to model the expected number of recognized actions. The binary enabler

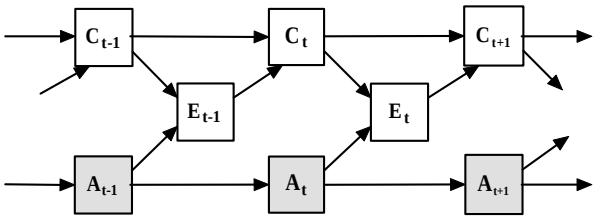

Fig. 3. Counter structure

variable E ($E_t = 1$ only if $A_t \neq A_{t-1}$ and therefore $C_t = C_{t-1} + 1$) forms an interface between the action variables A and counter nodes C, thus reducing the model's dimension. The joint distribution for the "counter structure" alone, computed for a sequence of T temporal slices is:

$$P(C_{1:T}, E_{1:T}, A_{1:T}) = P(C_1) \cdot P(E_1) \cdot P(A_1) \cdot \cdot \prod_{t=2}^{T} \{P(C_t \mid C_{t-1}, E_{t-1}) \cdot P(E_t \mid C_t, A_t) \cdot P(A_t \mid A_{t-1})\} \quad (5)$$

The whole joint distribution of a multi stream model enhanced with a counter structure (figures 2 and 3 combined) is given by the product of (1) with

$$P(C_1) \cdot P(E_1) \cdot \prod_{t=2}^{T} \{P(C_t \mid C_{t-1}, E_{t-1}) \cdot P(E_t \mid C_t, A_t)\}$$

Note that now action variables A generate, not only a sequence of subaction nodes S^j, but also a sequence of hidden counter nodes C.

The state transition probability for the counter variable C is given by:

$$P(C_t = j \mid C_{t-1} = i, E_{t-1} = f) = \begin{cases} j = i+1 \text{ if } f = 1 \\ j = i \quad \text{if } f = 0 \end{cases} \quad (6)$$

this means that C can be incremented only if the enabler variable E was high ($E_{t-1} = 1$) during the previous temporal slice $t-1$. $D_{j,k}(f)$ represents the state transition probability for the enabler variable E_t given that the action variable A is in state k and the counter C in state j:

$$P(E_t = f \mid C_t = j, A_t = k) = D_{j,k}(f) \quad (7)$$

If $A_t = k$ is the j^{th} recognised "meeting action", the probability to start evaluating the $(j+1)^{th}$ action and therefore to activate E ($E_t = 0$,$E_{t+1} = 1$), is modelled by $D_{j,k}(f)$. Initial state probabilities of C and E are respectively set to: $P(C_1 = 0) = 1$ and $P(E_1 = 0) = 1$, stating that *action* transitions are not allowed at the initial frame $t = 0$.

4.3 Inference

Statistical inference, which may be regarded as the estimation of a conditional distribution, is an essential step both for testing purposes (making predictions based on the model) and for model training (learning about the parameters of the model).

As mentioned in section 4, each directed acyclic graph (DAG) associated with a BN encodes a set of conditional independence properties. Joint probability distributions can be factorised, exploiting conditional independence statements provided by the graph. For example, consider a chain of four variables X,B,A,Y in which: X is the only parent of B, B is the only parent of A and A is the only parent of Y. The computation of $P(Y \mid X) = \sum_{A,B} P(Y,A,B \mid X)$ can be simplified by taking into account conditional independences:

$$P(Y,A,B \mid X) = P(Y \mid A)P(A \mid B)P(B \mid X) \ .$$

Therefore $P(Y \mid X)$ could be evaluated with less operations through:

$$P(Y \mid X) = \sum_{A} P(Y \mid A) \sum_{B} P(A \mid B)P(B \mid X) \ .$$

An efficient factorization of this type is at the core of every inference algorithm for graphical models, such as the junction-tree (JT) algorithm [14, 7].

Some graph manipulations [7, 15, 8] are needed to transform the original DAG into an equivalent form, suitable for the exact inference JT algorithm. The first step called *moralization* converts the DAG into an undirected graph, removing the directions of arcs and joining unconnected parents. The second step *triangulation* ensures the decomposability of the graph, by numbering each node A in decreasing order and adding arcs between all the pairs of A's neighbours characterized by a lower ordering number. A joint distribution that can be factorized on the original graph can also be factored on the larger triangulated graph. The final step is the construction of a junction tree from the triangulated (hence, decomposable) graph. The junction tree is a tree of cliques made with nodes from the original graph, satisfying some properties [15] such as the *running intersection property* and the *immediate resolution property*. The JT algorithm provides exact inference, exploiting as efficiently as possible the conditional independence contained into the original graph. Frequently, exact inference is not a feasible approach, due to model complexity and practical temporal constraints. In such cases approximate inference approaches, such as Monte Carlo sampling and variational techniques have been successfully applied [8].

There are several software packages that possess the functionalities required to work with graphical models. In this work we have used the Graphical Models ToolKit (GMTK) [16].

5 Experiments

Experimental evaluations were performed on 30 fully transcribed meetings, part of the corpus described in section 2. Performances were evaluated using a leave-one-out cross-validation procedure, in which the system was trained using 29 meetings and tested on the remaining one, iterating this procedure 30 times. The annotation of meeting actions is rather subjective, and their boundaries must be considered to be approximate. We adopted the Action Error Rate (AER) metric that privileges the recognition of the correct action sequence, rather than the precise temporal alignment of recognised symbols. Like the Word Error Rate metric used in speech recognition, the AER is obtained

Model	Corr.	Sub.	Del.	Ins.	AER
(A) HMM	64.1	14.7	21.2	21.2	57.1
(B) HMM	70.5	10.3	19.2	14.7	44.2
(A) multistream	84.6	9.0	6.4	1.3	16.7
(B) multistream	91.7	4.5	3.8	2.6	10.9
(A) multistream + counter	86.5	6.4	7.1	1.3	14.7
(B) multistream + counter	92.9	5.1	1.9	1.9	9.0

Table 2. Performances (%) for: a simple HMM, our multistream approach, and the multistream model enhanced with a "counter structure"; using: (A) prosody and speaker activities or (B) prosody, speaker activities and lexical feature

by summing the insertion, deletion and substitution errors when aligned against the reference sequence. Table 2 shows some experimental results achieved using six experimental configurations. These configurations are obtained evaluating three models: a hidden Markov model, a basic multistream approach and the counter enhanced variant. Two features sets were used. Feature set (A) contains prosodic features and speaker activities. Feature set (B) extends (A) with the addiction of lexical features. Therefore multistream based models applied to the feature set (A) are evaluated using a double-stream model with only subactions S^1 and S^2. Multistream models associated with (B) have an additional Markov chain composed by substates S^3 and observable binary lexical features Y^3. During both the experiments with a simple HMM, all features families have been merged in advance into a single feature vector (*early integration*). As anticipated in section 3.4 a simple HMM has poor performances, independently from the feature set used: 57% AER using two feature families and an improvement to 44% using lexical features as well. The adoption of a multistream based approach reduces the AER to less than 20%, and another small improvement is granted by the counter structure. Independently of the feature set, a counter structure leads to a better insertions/deletions ratio, enabling the model to fit better the experimental data and to have a further improvement in AER. Enhancing the feature set with lexical features, improves the percentage of correctly recognized actions by about 6%, independently of the adopted model (HMM, multistream, counter enhanced multistream). Therefore we reached our best results (9% AER and 93% correct) using the most elaborate DBN model with the most comprehensive feature set.

6 Conclusion

In this paper, we have presented a framework for automatic segmentation of meetings into a sequence of meeting actions. These actions or phases are result of the participants' interactions and involve multiple modalities. Starting from individual and environmental audio recordings, some relevant acoustic features were extracted. A set of prosodic features was evaluated over individual lapel microphone signals, and the dynamics of speaker activity were highlighted using a microphone array based sound direction estimation process. A lexically based monologue/discussion discriminator was developed

using textual transcriptions. All these three features streams were individually tested, and then integrated using a specialized DBN model. This model included the individual processing of different feature families (multistream approach), and a simple mechanism to improve action duration modelling (counter structure). The resulting system, tested with the M4 meeting corpus, attained an accuracy of 92% correct, with an action error rate of 9%. The chosen DBN framework seems to be a flexible and promising approach to the meeting segmentation and structuring task. Further multimodal features will be soon integrated into this scalable model. Ongoing work concerns some video related features, and an extended lexical meeting actions discriminator is being investigated.

References

1. R. Kazman, R. Al Halimi, W. Hunt, and M. Mantei. Four paradigms for indexing video conferences. *IEEE Multimedia*, 3(1), 1996.
2. I. McCowan, S. Bengio, D. Gatica-Perez, G. Lathoud, F. Monay, D. Moore, P. Wellner, and H. Bourlard. Modelling human interaction in meetings. *Proc. IEEE ICASSP*, 2003.
3. K. Sonmez, E. Shriberg, L. Heck, and M. Weintraub. Modelling dynamic prosodic variation for speaker verifi cation. *Proc. ICSLP*, pages 3189–3192, 1998.
4. N. Morgan and E. Fosler-Lussier. Combining multiple estimators of speaking rate. *Proc. IEEE ICASSP*, pages 729–732, 1998.
5. I. McCowan, D. Gatica-Perez, S. Bengio, and G. Lathoud. Automatic analysis of multimodal group actions in meetings. *IDIAP RR 03-27*, May 2003. Submitted to IEEE Transactions of Pattern Analysis and Machine Intelligence.
6. A. Dielmann and S. Renals. Dynamic Bayesian networks for meeting structuring. *Proc. IEEE ICASSP*, pages 629–632, 2004.
7. R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, 1999.
8. K. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, UC Berkeley, 2002.
9. J. Bilmes. Graphical models and automatic speech recognition. *Mathematical Foundations of Speech and Language Processing*, 2003.
10. P. Smyth, D. Heckerman, and M. I. Jordan. Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9(2):227–269, 1997.
11. S. Fine, Y. Singer, and N. Tishby. The hierarchical hidden Markov model: Analysis and applications. *Machine Learning*, 32(1):41–62, 1998.
12. A. Divakaran L. Xie, S.-F. Chang and H. Sun. Unsupervised discovery of multilevel statistical video structures using hierarchical hidden Markov models. *Proc. IEEE ICME, Baltimore*, 2003.
13. L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 2(77):257–286, 1989.
14. F. V. Jensen. *An Introduction to Bayesian Networks*. Springer, 1996.
15. G. Zweig. *Speech Recognition with Dynamic Bayesian Networks*. PhD thesis, UC Berkeley, 1998.
16. J. Bilmes and G. Zweig. The Graphical Model ToolKit: an open source software system for speech and time-series processing. *Proc. IEEE ICASSP*, Jun. 2002.