History-based visual mining of semi-structured audio and text[*]

Matt-Mouley Bouamrane[†] Saturnino Luz[†]
[†]*Department of Computer Science*
Trinity College, University of Dublin
Dublin 2, Ireland
{bouamrane,luzs}@cs.tcd.ie

Masood Masoodian[‡]
[‡]*Department of Computer Science*
The University of Waikato
Hamilton, New Zealand
m.masoodian@cs.waikato.ac.nz

Abstract

Accessing specific or salient parts of multimedia recordings remains a challenge as there is no obvious way of structuring and representing a mix of space-based and time-based media. A number of approaches have been proposed which usually involve translating the continuous component of the multimedia recording into a space-based representation, such as text from audio through automatic speech recognition and images from video (keyframes). In this paper, we present a novel technique which defines retrieval units *in terms of a log of actions performed on space-based artefacts, and exploits timing properties and extended concurrency to construct a visual presentation of text and speech data. This technique can be easily adapted to any mix of space-based artefacts and continuous media.*

1 Introduction

Visual data mining of multimodal meeting data is a relatively new field in which modality translation has emerged as the dominant paradigm [6, 7]. In the case study presented in this paper, which is based on mining collaboratively written texts coupled with audio recordings, we propose an alternative approach where the recording of actions (or interaction) history on space-based artefacts (segments of text) is used in order to uncover not only non-sequential temporal links with the continuous medium (audio), but also links to other non-contiguous space-based artefacts. In this scenario, the timing properties of the various data units are not used as the underlying structure of our data representation but rather as a means of linking these various data units. We subsequently used the inherent structure of text as the basis of a meeting mining tool for exploring multimedia recordings which graphically represents meetings in a tree structure. Extended concurrency of the editing operations performed on each paragraph of a shared text document and the speech exchanges between participants is used as the basis of our retrieval units. The results presented here can be adapted and extended to any mix of space-based data and continuous media by defining and capturing a set of actions (according to the nature of the data) performed on the space-based data.

2 Multimedia Data Description

The data collected for each meeting consist of a text document written collaboratively, individual audio files for each participant, and XML-encoded interaction metadata. A collaborative writing environment was specifically designed to support and capture editing, gesturing and audio interaction among remote participants [2]. When writing text, co-authors naturally structure their documents into semantic units. Our aim is to keep track of these semantic units from their creation, and follow their evolution during the co-writing process. We assume that the appropriate granularity for text units for this purpose is the paragraph. Whenever an editing or gesturing operation is performed on a paragraph, our system generates timestamps containing information on the agent who performed the operation, the type of action (*Insert, Delete, Paste, Cut, Point* etc) the start and end time of the action, and the exact content of the editing operation. An XML representation of the document attaches these timestamps to the paragraphs for which they were generated. If the document is structurally modified, the system ensures that these paragraph timestamps are handled accordingly. A detailed description of this timestamping model can be found in [3].

3 Retrieval Model and Units

The final text document produced during the collaborative editing task is used as the basis of our data represen-

[*]This research was supported by an Enterprise Ireland Basic Research Grant

Bouamrane, M.-M., Luz, S., and Masoodian, M. History based visual mining of semi-structured audio and text. In *Proceedings of the 12th International Multi-media Modelling Conference, MMM06* (Beijing, China, Jan. 2006), IEEE Press, pp. 360–363.

tation. We believe the textual outcome is a natural and intuitive starting point for visual data mining. Furthermore, one of the challenges in representing orthogonal media such as text and audio stems from the fact that there is no obvious way of representing these in a structured manner. Co-writers naturally use text segmentation (chapters, paragraphs, sentences, etc.) to structure their ideas in semantic units. The most interesting aspect of the paragraph-based editing history recording lies in using the timing of editing operations performed on each paragraph in order to provide temporal entry points to an otherwise sequential audio file.

Using text as a starting point, we create a data structure called MeetingTree. The basic retrieval unit of a MeetingTree is determined through *extended concurrency* (explained in detail in section 3.1) of editing and gesturing actions performed on each paragraph of the text document and participants speech exchanges. This is illustrated in Figure 1 which represents the retrieval unit, or temporal neighbourhood, formally defined in [5] of one paragraph of our meetings.

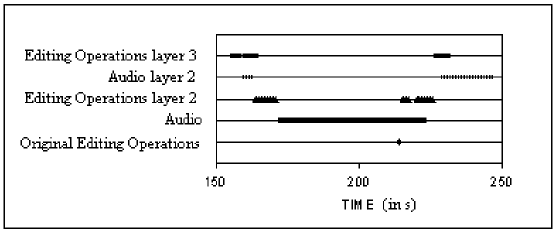


Figure 1. Retrieval Unit, or Temporal Neighbourhood for a paragraph

The bottom line shows all operations performed on one particular paragraph. In this case, it is a single atomic operation (gesturing). The line above this shows a concurrent audio segment of participants speech exchanges. Layer 2 shows all editing operations performed on *paragraphs* other than the original one within the duration of the audio segment. These operations are linked to a different set of audio segments (audio layer 2). Finally, editing operations layer 3 represent all editing operations *not included* in the previous editing layers which happened within the duration of audio layer 2. The retrieval algorithm stops when no more concurrent editing operations or audio segments can be found.

3.1 Audio Segmentation and Reduction

One of the challenges in multimedia retrieval resides in organising the data in a meaningful graphical representation of the information, an issue which is usually compounded by the large data sets available. In the model described above, simply representing the retrieval units without any pre-processing has the major inconvenience of creating tree representations with thousands of nodes. This is mainly due to the large number of speech exchanges occurring during the meeting, typically in the hundreds, even for relatively short meetings (20 to 45 minutes). In what follows, we discuss the strategies we implemented for efficiently mapping speech exchanges in our prototype and what results this mapping has had on the information mining task. We wish to stress that all subsequent considerations on audio segment reduction and merging only applies to the displaying of audio nodes information on the user interface. The audio file itself is never truncated and can always be accessed either sequentially or randomly.

The initial segmentation of audio is done through silence detection. Due to the nature of the audio recording, initially done in RTP packet format, the issue of speaker identification is trivial. The speaker is simply inferred from the RTP source identifier. We are henceforth able to build a binary audio profile for each participant, using a granularity of one second, representing speech or silence intervals. These easily define temporal speech segments. Once the meeting is recorded, these segments are of known length and can be totally ordered. Let s_1, e_1, s_2 and e_2 be respectively the start and end time of two audio segments Au_1 and Au_2. We use a subset of Allen's [1] temporal interval relations as follows: Au_1 is *before* Au_2 if $e_1 < s_2$. Au_1 is *during* Au_2 if $s_1 \geq s_2$ and $e_1 \leq e_2$ (equal, during, starts or finishes). Au_1 *overlaps* Au_2 if $s_1 \leq s_2 \leq e_1 \leq e_2$ (meets, overlaps).

The criteria for audio segment merging discussed below aims to meet the following requirements: to reduce the number of audio segments mapped for compact graphical representation while following the natural structure of discourse for meaningful listening. Audio segments with the *during* relation refer to sections of the audio recording when several participants are speaking at the same time. As people generally tend not to speak at the same time in remote settings, we have found that occurrences of these relations within our meetings corpus are rare in comparison with the other two *before* and *overlaps* relations[2]. These speech segments are generally of very short duration and almost always consist of false starts, acknowledgements or comments made while another participant is speaking (e.g "yes","right", "ok"). It can also be argued that in remote collaborative meetings, participants feel a greater need to acknowledge each other's presence with verbal utterances as a way of maintaining awareness [4] in remote collaborative meetings. Taken out of context these individual utterances would appear to be meaningless. Therefore, in order to prevent them from being displayed the system merges all audio segments related by the *during* relation. In other words, the audio mapping reduction means that if a participant makes a comment while another participant is speaking, these concurrent speech exchanges are formally regarded as a single segment. The identities of the individual speakers taking part on these merged audio segments is nevertheless preserved.

As pointed out above, the number of audio intervals with the *during* relation is relatively small. In order to more efficiently map speech exchanges in the tree representation of the meeting, a second step of audio intervals merging with the *overlaps* relation is performed. In accordance with the natural structure of discourse, it is reasonable to assume that audio segments in close time proximity may be relevant to one another. This can be viewed as the question-answer pair paradigm [7] where adjacent speech exchanges are more informative jointly than when considered on their own.

In remote meetings, participants tend not to start speaking immediately one after another, usually waiting a few seconds in order to make sure that the other person has finished making their points. Therefore, a strict interpretation of the *overlaps* interval relation would yield poor results. In order to take into account such natural short speech pauses, we have introduced a tolerance value of a few seconds for the *overlaps* relation. This means that two audio segments separated by a silence whose duration is less than the tolerance value are considered to be *overlapping*. This tolerance value can be adjusted by the user as it might yield different results for different meeting tasks or participants. Figure 2 shows the result of such audio merging for a chunk of audio segments of one meeting, using a tolerance merging value of 3s, which has shown good overall results in our prototype testing. As there would usually be a short delay between participants speaking and subsequently typing, considerations previously expressed regarding the *overlap* of audio segments may also be extended in order to find links between audio and editing operations. In other words, if an audio segment ends within less than the *overlap tolerance* value of an editing operation it is regarded as *overlapping*. The merging and linking of audio intervals and editing operations described above is important for the data mining task in that they allow for clustering of audio segments which are not necessarily sequential into retrieval units and relate text segments that are not necessarily contiguous.

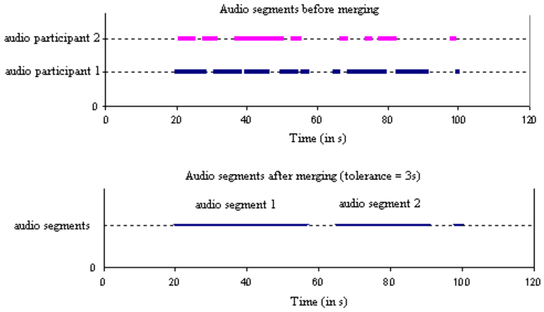


Figure 2. Audio segmentation before and after merging using a tolerance value of 3s

4 Implications for Browsing

MeetingTrees can be straightforwardly mapped to hierarchical user interface components. A tree component could be used to display each paragraph's retrieval unit as a branch of audio nodes and editing actions, as shown in Figures 4 and 5. We claim that this form of visual representation highlights segment relationships that would be undetectable in linear browsing modes. Consider, for example, the retrieval units illustrated in Figure 3. The retrieval unit inferred from paragraph 4 links three non sequential audio segments Au_1, Au_2 and Au_3. It also links paragraph 4 to editing operations performed to four other non-contiguous text segments. As the time line suggests, these paragraphs were modified in an arbitrary order, showing that paragraph 5, physically the closest to paragraph 4, was modified last, after a number of other considerations were discussed and taken into account. The text content of these paragraphs is as follows:

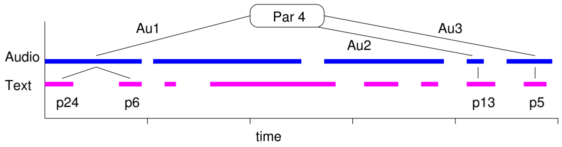


Figure 3. Non-linear linking

(par4) budget of 3000 from the student union
(par5) maybe charge people more?

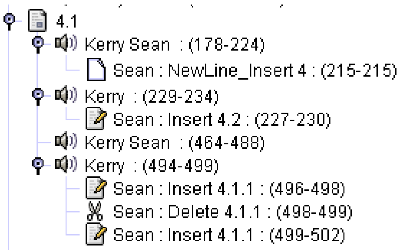
Before the participants came to the conclusion that they needed to charge people more, in paragraph 5, they first made changes to a number of paragraphs, in this order:

(par24) bus hire 1500 for 60 people
(par6) travel 1500
(par13) 4400 for students 2700 for staff in single room

The subjects participants discussed verbally while these paragraphs were modified are the following:

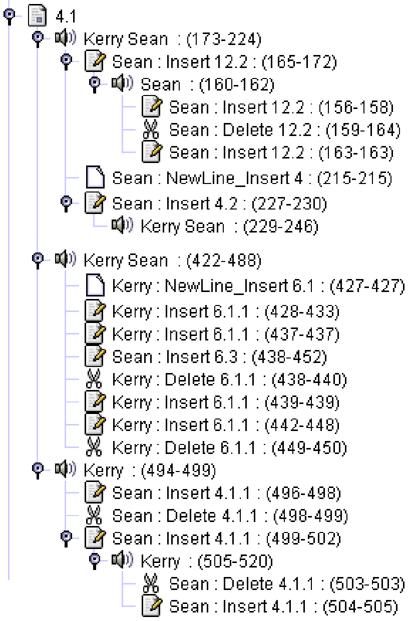
(Au1) (par24,6) travel arrangements, cost of hiring a bus, existence of a budget
(Au2) (par13) hotel expenses
(Au3) (par5) need to charge people more

In other words, only when the participants realised that the cost of travel and the hotel would exceed their initial budget (paragraph 4) did they decide to charge for the trip (paragraph 5). Adjusting the value of the *overlap* tolerance affects the shape and depths of retrieval units. A tolerance value of 3 seconds was used in the example above. Figure 4 shows a paragraph retrieval unit when the *overlap* tolerance



```
4.1
  Kerry Sean  : (178-224)
    Sean : NewLine_Insert 4 : (215-215)
  Kerry  : (229-234)
    Sean : Insert 4.2 : (227-230)
  Kerry Sean  : (464-488)
  Kerry  : (494-499)
    Sean : Insert 4.1.1 : (496-498)
    Sean : Delete 4.1.1 : (498-499)
    Sean : Insert 4.1.1 : (499-502)
```

Figure 4. A paragraph's retrieval unit with the overlap tolerance set to 0



```
4.1
  Kerry Sean  : (173-224)
    Sean : Insert 12.2 : (165-172)
      Sean  : (160-162)
        Sean : Insert 12.2 : (156-158)
        Sean : Delete 12.2 : (159-164)
        Sean : Insert 12.2 : (163-163)
    Sean : NewLine_Insert 4 : (215-215)
    Sean : Insert 4.2 : (227-230)
      Kerry Sean  : (229-246)
  Kerry Sean  : (422-488)
    Kerry : NewLine_Insert 6.1 : (427-427)
    Kerry : Insert 6.1.1 : (428-433)
    Kerry : Insert 6.1.1 : (437-437)
    Sean : Insert 6.3 : (438-452)
    Kerry : Delete 6.1.1 : (438-440)
    Kerry : Insert 6.1.1 : (439-439)
    Kerry : Insert 6.1.1 : (442-448)
    Kerry : Delete 6.1.1 : (449-450)
  Kerry  : (494-499)
    Sean : Insert 4.1.1 : (496-498)
    Sean : Delete 4.1.1 : (498-499)
    Sean : Insert 4.1.1 : (499-502)
      Kerry  : (505-520)
        Sean : Delete 4.1.1 : (503-503)
        Sean : Insert 4.1.1 : (504-505)
```

Figure 5. Same unit with tolerance set to 3s

value is set to 0 and Figure 5 shows the same branch when the value is set to 3 seconds. Not surprisingly, branches of the retrieval unit will grow with a larger *overlap* tolerance value as more links are uncovered. In the example above, a tolerance value set to 0 would not have uncovered links to paragraphs 24 (travel arrangements) and 13 (hotel booking), giving no clues as why the participants suddenly decided to charge people more. The difficulty here lies in adjusting the tolerance value so that relevant links can be made (tolerance value above a certain threshold) without increasing it to the extent where non-related information is being included in the retrieval unit (tolerance value too high). Ideally, the tolerance value should be roughly the same length as the participants *short* speech pauses so that two subsequent audio exchanges will be linked. We are currently investigating strategies for automatically setting tolerance values.

Analysis of the meeting corpus according to the model described above revealed two basic modes of collaboration: *tight collaboration*, where participants work within the text document on semantically related text segments, performing tasks that relate to or depend upon one another within the same time frame, and *loose collaboration*, where participants delegate each other unrelated tasks on which they work independently until these tasks are completed. The ability to distinguish between patterns of cooperation is important with respect to the choice of browsing strategy to be adopted.

5 Conclusion

The model presented in this paper used interaction history on text segments and extended concurrency with speech segments to generate information retrieval units. These units reveal how individual paragraphs relate to various (possibly non-sequential) audio segments as well as other (possibly non-contiguous) paragraphs. Time is no longer regarded as the single structuring factor for information presentation and retrieval. Instead, properties of individual (space- and time-based) data units are used to uncover information patterns among sets of data units. Future work will involve a thorough analysis of the cooperation modes discussed in the last section. Sets of features will be defined in order to allow the system to segment and classify recordings according to those cooperation modes.

References

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 11(26):832–843, Nov 1983.

[2] M.-M. Bouamrane, D. King, S. Luz, and M. Masoodian. A framework for collaborative writing with recording and post-meeting retrieval capabilities. IEEE Distributed Systems Online, 2004.

[3] M.-M. Bouamrane, S. Luz, M. Masoodian, and D. King. Supporting remote collaboration through structured activity logging. In *Proceedings of the 4th International Conference on Grid and Cooperative Computing (GCC 2005), LNCS 3795*, pages 1096–1107, Beijing, 2005. Springer-Verlag.

[4] P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *CSCW '92*, pages 107–114. ACM Press, 1992.

[5] S. Luz and M. Masoodian. A model for meeting content storage and retrieval. In Y.-P. P. Chen, editor, *MMM 2005*, pages 392–398, Melbourne, 2005. IEEE Computer Society.

[6] S. Tucker and S. Whittaker. Accessing multimodal meeting data: Systems, problems and possibilities. In *Proceedings Of MLMI04, LNCS 3361*, pages 1–11. Springer-Verlag, 2005.

[7] A. Waibel, M. Brett, F. Metze, K. Ries, T. Schaaf, T. Schultz, H. Soltau, H. Yu, and K. Zechner. Advances in automatic meeting record creation and access. In *Procs. of ICASSP'01*, pages 597–600. IEEE Computer Society, 2001.