Accessing a Large Multimodal Corpus Using an Automatic Content Linking Device

Citation for published version:
Popescu-Belis, A, Carletta, J, Kilgour, J & Poller, P 2009, Accessing a Large Multimodal Corpus Using an Automatic Content Linking Device. in Multimodal Corpora - From Models of Natural Interaction to Systems and Applications. Lecture Notes in Computer Science, vol. 5509, Springer Berlin Heidelberg, pp. 189-206. DOI: 10.1007/978-3-642-04793-0_12

Digital Object Identifier (DOI):
10.1007/978-3-642-04793-0_12

Link:
Link to publication record in Edinburgh Research Explorer

Published In:
Multimodal Corpora - From Models of Natural Interaction to Systems and Applications

Accessing a Large Multimodal Corpus using an Automatic Content Linking Device

Andrei Popescu-Belis[1], Jean Carletta[2], Jonathan Kilgour[2], and Peter Poller[3]

[1] Idiap Research Institute
Rue Marconi 19, P.O. Box 592, 1920 Martigny, Switzerland
andrei.popescu-belis@idiap.ch
[2] Human Communication Research Centre, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, Scotland, UK
J.Carletta@ed.ac.uk, jonathan@inf.ed.ac.uk
[3] DFKI GmbH, Stuhlsatzenhausweg 3
D-66123 Saarbruecken, Germany
peter.poller@dfki.de

Abstract. As multimodal data becomes easier to record and store, the question arises as to what practical use can be made of archived corpora, and in particular what tools allowing efficient access to it can be built. We use the AMI Meeting Corpus as a case study to build an automatic content linking device, i.e. a system for real-time data retrieval. The corpus provides not only the data repository, but is used also to simulate ongoing meetings for development and testing of the device. The main features of the corpus are briefly described, followed by an outline of data preparation steps prior to indexing, and of the methods for building queries from ongoing meeting discussions, retrieving elements from the corpus and accessing the results. A series of user studies based on prototypes of the content linking device have confirmed the relevance of the concept, and methods for task-based evaluation are under development.

Key words: multimodal corpora, meeting corpora, speech-based retrieval, meeting assistants, meeting browsers.

1 Introduction

Researchers often think of multimodal corpora as simply a research tool – something which yields samples of information about human behaviour that will allow component technologies, such as gesture recognizers, to be built. However, technology makes it currently affordable to record data from relatively complex sensors, such as cameras or microphone arrays, in a variety of settings. Rich historical archives related to organizations, events, projects or people can thus easily be set up and continuously extended, in the form of multimodal corpora. This opportunity is currently being explored in many different ways, from attempts to capture all of the events in an individual's life to analyses of activities in public spaces. In such uses, the multimodal corpus is both a research tool – for developing the automatic analyses that underpin the ability to make use of large amounts of recorded material – and a platform for developing, demonstrating and refining the new application concepts that these automatic analyses enable.

One commercially important application of this general approach is on meeting archives. The AMIDA Automatic Content Linking Device (ACLD) performs real-time searches over a multimodal meeting archive using automatic speech recognition, thus providing assistance with past information during an ongoing meeting, at a minimal attentional cost for the participants. Throughout the process of developing the ACLD from the initial concept to a standalone demonstration, we have been completely reliant on the AMI Meeting Corpus. Taking this process as a case study, we explain how we leveraged this multimodal recorded data set in order to create an application for online use, without the need for staging frequent live meetings to test the application.

We begin with an explanation of the content linking concept (Section 2), followed by a description of the AMI Meeting Corpus (Section 3) and a discussion of the uses to which it has been put during the development of our technology demonstration (Section 4). The four main stages of the content linking approach – corpus preparation and indexing, query preparation, retrieval, and access to multimodal data – are described respectively in Sections 5.1 through 5.4. Initial results from user studies, based on reactions from focus groups (Section 6.1) and on feedback from commercial partners on pilot versions of the ACLD (Section 6.2), are encouraging but do not preclude more systematic evaluation in the future (Section 6.3). The plans for future development of the ACLD are finally outlined in Section 7.

2 The Content Linking Concept

2.1 Scenario of Use and Inferred User Needs

Imagine for a moment that your company is moving into a new office building. The meeting rooms in the building are instrumented with cameras, microphones, and other devices to record what happens in them. On the day that you move in, the rooms are turned on, recording everything that happens in them, unless someone presses a privacy switch to disable the system temporarily. Over the course of a year, a large meeting archive is built up. As well as allowing meeting playback, it contains information about when meetings happened, who attended them, what slides were shown at what times, and what documents were presented or consulted during a meeting, or were produced as a result of a meeting. What use would the company wish to make of such an archive, and what tools would employees need for that use?

Although this scenario may sound far-fetched, many companies already make recordings of their most important meetings for their archives, and can see a use for recording all meetings, as long as the benefits to the company's employees are high enough for them to leave the system on. The problem comes with sifting through the archive to find the right bits. Although there are many times at which one might wish to search the archive, this problem is especially severe during meetings themselves. Meetings take a great deal of staff time, which is expensive. Often people have a feeling during meetings that they need some piece of information, but they can't lay their hands on it, at least not during the meeting itself. And yet, producing the right piece of information at the right time can change the course of a meeting, and sometimes, even determine the future of a company. This is the problem that motivates our design of a *content linking* application.

2.2 Outline of the Proposed Solution

Our solution to this problem is to have the room not just record the meetings that happen in it, but also "listen" to them and search quietly in the background for the most relevant documents and meeting segments from the archive, ready to be consulted whenever someone in the meeting feels the need. A system providing tailored access to potentially relevant fragments of recorded meetings or documents could be very valuable in improving group communication and individual awareness. Participants in a meeting often mention previous discussions, or documents related to them, containing facts that are relevant to their current discussion, but accessing them requires more time than participants can afford to spend during the discussion. If provided with an efficient device for searching past meetings and documents, participants only need to decide if they want to explore any further, and possibly introduce in the current discussion, the meeting fragments or documents retrieved automatically for them. The system could be used privately by every participant, or its results could be shown to the entire group on a separate projection screen.

2.3 Previous Work

The real-time content linking application put forward here borrows from several other applications described in the literature, with the main notable differences being the access to multimodal processed data, in real-time, through a speech-based, autonomously functioning system. Our system is also – to our knowledge – the first one that is fully implemented in a multimodal interaction context, giving access to indexed multimedia recordings as well as websites, based on automatic speech recognition and keyword spotting.

The initial versions of the content linking concept were introduced either as *query-free search*, which was implemented in the Fixit system [1], or as *just-in-time retrieval*, implemented in the Remembrance Agent [2, 3]. Fixit is an assistant to an expert diagnostic system for the products of a specific company, such as fax machines and copiers. Fixit monitors the state of the user's interaction with the diagnostic system, in terms of positions in a belief network, and runs searches on a database of maintenance manuals to provide additional support information related to the position in the network. The results of the searches are in fact pre-computed for each node of the belief network, in order to speed up the process at run time.

The Remembrance Agent is much closer, in its design, to the AMIDA content linking approach. The agent, which is integrated to the Emacs text editor, runs searches at regular intervals (every few seconds) using a query that is based on the last typed words (the size of the set of words is configurable, e.g. from 20 to 500). Results from a repository of emails or text notes are displayed in a separate frame, and can be opened within Emacs as well.

Both terms that refer to these two systems – query-free search and just-in-time (information) retrieval – capture important aspects of the general approach: there is no need for the user to formulate explicit queries, and results are updated regularly so that the users receive them, in theory, exactly when they need them. Several other systems have pursued and extended the approach. Watson [4] monitors the user's operations in a text editor, but proposes a more complex mechanism than the Remembrance Agent for selecting terms for queries, which are directed to the Altavista web search engine. Besides automatic queries, Watson also allows users to formulate queries and disambiguates them using the terms that were selected automatically.

Another query-free search system was designed for enriching television news with articles from the Web [5]. The system annotates TV broadcast news, using queries derived from closed captioning text, with links to potentially relevant news wire from the Web. A related type of work, though from a different perspective not inspired by information retrieval, are systems for document/speech alignment [6, 7] specifically applied to meeting browsers. These systems typically perform the alignment offline, though online techniques can also draw inspiration from them [8].

Turning to speech-based systems, the creators of the Remembrance Agent have also designed Jimminy, a wearable assistant that helps users to take notes and to access information when they cannot use a standard computer keyboard [9]. Jimminy uses a number of contextual capture devices, in particular a positioning device that detects the room where the user is situated, and a device for identifying the user's interlocutors based on their badges. However, the use of speech was not implemented, and the detection of the subject of conversation was simulated by entering this topic as real-time notes. The testing of the system concerned in fact mostly the note-taking function. More recently, several speech-based search engines have been proposed, for instance a spoken interface to Google [10], or a spoken interface for a mobile device [11]. Such systems are comparatively less common than systems for searching spoken document archives [12] or for multimedia information retrieval [13].

3 The AMI Meeting Corpus

The AMI Meeting Corpus [14–16] consists of 100 hours of recordings from rooms like the one described in our opening scenario. The recordings use a range of signals synchronized to a common timeline. These include close-talking and far-field microphones, individual and room-view video cameras, and output from a slide projector and an electronic whiteboard. During the meetings, the participants also have unsynchronized pens available to them, which record what they write. The meetings were recorded in English, in three different rooms [17, p. 6–12], and include mostly non-native English speakers.

It was important for us to be able to understand what was happening during the meetings in the corpus. Real groups are messy – they have a shared understanding built up over their shared history, they sometimes are not sure what they are meant to achieve, and their participants are sometimes motivated by things that bear no relationship to the actual task. Although the AMI Corpus contains some real meetings of real groups to ensure that we do not diverge from reality too far, most of the material in the corpus is elicited experimentally to be as similar to real meetings as possible, but to ensure that we can capture everything there is to know about the groups involved.

In our elicitation technique, a group of four people come together in a one-day role-playing exercise in which they act as a team that needs to design a new kind of remote control for a TV set. At the beginning of the day, they are given separate crash courses in the specialty roles that such a team might contain, learning how to be a project manager, industrial designer, user interface designer, or marketing expert. They then engage in a series of four meetings which take them from an introduction to the problem that they need to solve, all the way through to their final design of the remote control. In between meetings, the participants do individual work towards the remote control design, receive emails from other people in "the company" that supply new information, and fill out some questionnaires that can be used *a posteriori* to assess things like how efficient the group is being or whether a leader is emerging.

During both the individual work and the team meetings, everything the team members produce is collected – emails, presentations, hand-written notes – so that it is available to the research team. The AMI Meeting Corpus is a publicly available [14] interdisciplinary resource, and the work that has been done on it contributes to organizational psychology and corpus linguistics as well as serving as material for the development of a range of multimodal processing and language recognition technologies.

4 The Many Uses of a Corpus

One way of viewing the AMI Meeting Corpus is as a collection of features, with training and test data, that allow us to develop component technologies. Since most corpus users, especially those involved in international evaluations, are concerned with improving the performance of some component or other, this is the dominant view. However, we have relied on the corpus not just during component development, but during all stages of the software lifecycle.

Our first use for the corpus was simply as inspiration for the kinds of applications we might build. With hindsight, it may seem obvious that we could build a content linking device using our technologies, and that the end users would be enthusiastic about it. However, it was not something we considered doing until we considered precisely how people actually behave in the recorded meetings.

There is no substitute for this kind of observation. In particular, potential users can usually identify things they do not need, but are less good at introspecting about what they do need. This is why advanced user studies make use of recorded data among other observation and elicitation techniques [18].

Our second use was for the capture of user requirements. Potential users are able, given a concrete enough concept, to make suggestions about what would make for useful software features, and a way of making a concept concrete for them is by showing them how the concept would work. In the very early stages, before any programming is done, two techniques are helpful. One is describing specific meeting problems, either taken from or inspired by the corpus, and having users talk around them. The other is to use materials from the corpus to create mock-ups – still images that show a series of screenshots for an interface that could potentially be built. We used both techniques to inform our development.

Our final use was during application development itself. Part of this is continuing work on user requirements by taking prototypes to focus groups in the same way as the initial mock-ups, and of course, giving demonstrations to potential vendors and future funders, using the AMI Corpus in the following way. Working on an application that applies to live meetings creates particular challenges for development and demonstration. One cannot know what will happen in a live meeting ahead of time. This makes demonstrating on a live meeting fraught with danger, especially for early system versions. For this reason, we usually demonstrate using a recorded meeting, by playing it back in a separate frame or window and feeding its signals into the system as if they were live. For this purpose, the AMI Corpus design of meetings in series helps us considerably. We use a group's last meeting (a meeting labeled with 'd', e.g. 'ES2008d') to simulate a live meeting, treating segments from the three previous meetings (labeled 'a', 'b' and 'c') and associated documents as the group's history to be linked. This method makes it possible to demonstrate our technology when no meeting is happening, and, essentially, to replicate the same behaviour over and over again, a crucial property during software development and debugging.

There is one final point worth making about development of this kind of application. Making the componentry work online, fast enough for live use, is something that the research community often considers to be an uninteresting engineering task. Online processing usually involves trading off speed against accuracy, which is unattractive when one is judged by one's results without reference to processing time. Partly, it is the possibility of end user applications like our content linking device that motivates having this engineering work done. Because we demonstrate on a recorded meeting, at least at first, none of our componentry need work online. We can simply pre-process the recorded meeting and store the results for use during the demonstration, feeding them into the system as if they were live just as we do with the signals. To find out how the technology would actually work given the constraints of on-line processing, we gradually replace our simulated components with real online ones as the researchers make them available.

5 The AMIDA Automatic Content Linking Device

In a nutshell, the Automatic Content Linking Device performs searches at regular intervals over a database of documents and meeting recordings – derived from the AMI Meeting Corpus at this stage – with a search criterion that is constructed based on the words that are recognized automatically from an ongoing discussion. This requires the following processing stages. The corresponding modules and data exchanges between them are represented in Figure 1, and are further described in Sections 5.1 through 5.4 below.

1. *Document Bank Creator and Indexer*: prepares, and updates before each meeting, a database of documents and of snippets of previous meetings in which media and annotations are aligned.
2. *Query Aggregator*: at regular and frequent intervals during a meeting (e.g. every 30 seconds or on the user's demand by clicking a button), prepares a query to this database, derived from the ongoing conversation through automatic speech recognition (ASR) or keyword spotting (KWS), possibly emphasizing certain keywords. The module then executes the query, retrieves the results, and integrates them with previous ones, so that their variation is smoothed in time.
3. *User Interface*: displays the current search results, as clickable document links, and provides through them access to past documents and meeting recordings, but also to web pages through a web search API.

These functionalities are supported by a modular architecture called *The Hub* – represented as an area connecting several modules in Figure 1 – which allows communication through a subscription-based client/server protocol [19]. The Hub allows the connection of heterogeneous software modules, which may operate locally or remotely, and ensures that data exchange is extremely fast – a requirement for real-time processing. Data circulating through the Hub is formatted as timed triples – i.e. tuples of (time, object, attribute, value) – and is stored in a relational database, which is able to deal with large-scale, real-time annotations of audio and video recordings. 'Producers' of annotations send triples to the Hub, which are received automatically by the 'consumers' that subscribed to the respective types, defined by pattern matching over object names. Consumers can also query directly the Hub's database for past annotations and metadata about meetings.

An annotation-playback function that resubmits existing annotations to the Hub, without duplication, is required by the ACLD for simulating online processing and is currently under implementation. Until this becomes available, real time is simulated using data streamers that produce data from the files containing AMI Corpus annotations in the NITE XML Toolkit (NXT) format [20].

Complementing the Hub, the HMI Media Server broadcasts audio and video that was captured in an instrumented meeting room to various media consumers, mainly the ASR, KWS, and User Interfaces. The media sources are the various files stored on the Multimodal Media File server [14].

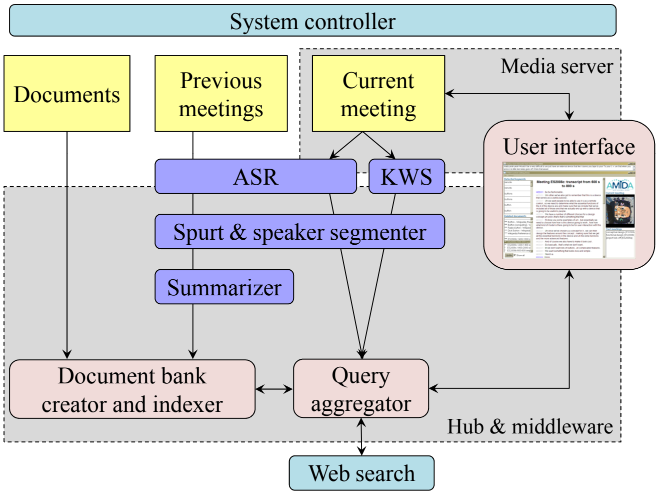


Fig. 1. Main components of the AMIDA ACLD. The two areas labeled as "Hub & middleware" and "Media server" represent the connection between all modules covering the respective areas. The Hub ensures real-time annotation exchange, while the Media server enables playback of audio and video from files or capture devices. ASR stands for automatic speech recognition, and KWS for keyword spotting.

5.1 Data Preparation: Pre-processing of Past Meetings

The *Document Bank Creator* (DBC) is run offline before a meeting, to create or update the repository of documents and pseudo-documents that will be searched during the meeting. This is a preparation task, which generates text versions of documents for indexing, by extracting text from heterogeneous file formats (essentially HTML and MS Office formats). HTML versions of each document are also generated for quick visualization in a web browser. The DBC then inserts into the Hub's database the document metadata – including URIs of source files – which identifies the list of documents that are available for each meeting, and lists their main properties: title, date, associated meeting, type, and authorship information when available. In a future scenario, the DBC could automatically construct the set of documents that are potentially relevant to a meeting, based on the project to which the meeting is related. The DBC could also be connected to an organizational content management system, which many large institutions set up as part of their policies regarding document sharing.

The document set includes snippets of previous meetings, as well as documents such as reports, memos, slides, emails, and so on. The snippets are currently one minute long fragments prepared from the ASR transcript [21] of the past meetings, which can be obtained with acceptable quality (for human readers) a couple of hours after the meeting (note that real-time ASR, used by the Query Aggregator, has a significantly lower accuracy). The trade-off in choosing the length of these snippets is between the need to have enough content words in each snippet, and the need for precision when snippets are opened in a meeting browser: if they are too large, then the user cannot easily find the region of interest. The segmentation into snippets currently avoids cutting through a speech segment. The use of topic segments as snippets is under study, the main limitation being the large granularity of thematic episodes.

The text version of the files is indexed using the Apache Lucene open-source software, creating indexes for each meeting of the AMI Meeting Corpus (in the present demonstration version). Indexing optimizes word-based search over large document sets. Here, all words are used as keywords, and the index is optimized using word stemmers and the TF*IDF weighing scheme.

5.2 Sensing the User's Information Needs

The retrieval of relevant fragments from past meetings, or of related documents, requires input about the current topic of conversation of meeting participants, in an ongoing or replayed meeting. Although many capture devices could be used to provide such input, we believe that the most informative cues lie in the words that are spoken by the participants. Therefore, our main goal at this level is to use Automatic Speech Recognition and/or Keyword Spotting modules to construct queries over the meeting/document database.

Real-time large vocabulary speech recognition is still a challenging objective for the ASR community, but within the AMIDA project such a system has recently become available [22]. One of its main features is the trade-off between speed and accuracy, which allows it to run in real-time (with a slight but constant delay only) even on standard workstations. The AMI Corpus is also accompanied, for internal use, by sample results from the offline ASR system developed within the AMIDA project [21]. Three ways of running the ACLD are thus mainly distinguished here by the origin of the ASR output:

1. Real-time ASR from signals captured in a smart meeting room (here, the audio setting has a strong influence on the recognition accuracy).
2. Real-time ASR over recorded signals (for demonstration purposes).
3. Simulating real-time ASR by producing and sending to the Hub the results of higher-quality offline ASR (for development and demonstration purposes).

The words from the ASR are filtered for stopwords, so that only content words are used for search. Our list of stopwords has about 80 words, including the most common function words, interjections and discourse markers. Furthermore, we believe that existing knowledge about the important terminology of a project can be used to increase the impact of specific words on search. A list of pre-specified keywords can thus be defined, and if any of them are detected in the ASR from the meeting, then they can be specifically marked in the query so that their importance is increased when doing the search (see next section). A specific list was defined by the user-study group for the AMI Meeting Corpus, and at present it contains words or expressions such as 'cost', 'energy', 'component', 'case', 'chip', 'interface', 'button', 'L_C_D', 'material', 'latex', 'wood', 'titanium', and so on, for a total of about 30 words. This list can be modified online by the users of the ACLD, by adding or removing words. If no list of keywords is available, then all words from the query (except for the stopwords) simply receive equal weight.

The list of keywords can also be used directly by a real-time keyword spotting system, which identifies their occurrence in speech without performing ASR on the entire signal. We experimented with a system available within AMIDA [23], which allows to update the list of keywords during execution, i.e. during a meeting. The results are encouraging, especially for long keywords (short ones tend to be over-recognized), and current work is aimed at combining the KWS results with those from the real-time ASR, in order to increase the precision and the coverage (e.g. for out-of-vocabulary words) of speech recognition.

The functions described in this sub-section are performed by the *Query Aggregator* (QA) module. The QA periodically obtains from the Hub the words or keywords that were recognized, and processes them in batches corresponding to time frames of fixed size, e.g every 20-30 seconds – this can be modified when running the application. This typical duration is a compromise between the need to gather enough words for search, and the need to refresh the search results reasonably often. Instead of the fixed time frame, information about audio segmentation into spurts or utterances could be used for a more natural segmentation of the ASR input. Additionally, the QA can also launch a query when a user demands it by pressing a button in the interface.

5.3 Retrieval of Documents and Snippets

The QA uses the query words obtained from the ASR and/or the KWS to formulate a query which is addressed to an information retrieval engine (Apache Lucene), over the index of documents created by the DBC's indexer. If any of the pre-specified keywords are detected in the speech, then they are specifically marked in the query, and their importance is increased when doing the search. Using this keyword boosting mechanism (specific to the Lucene engine), the weight of keywords is currently set at five times the weight of the other, non-boosted, words. The results returned by the Lucene engine are the meeting fragments and documents that most closely match the query – in information retrieval terms – in the respective time frame.

The QA then returns to the Hub the results, specifically, as a list of structured records such as: (meeting, time, keyword, relevance, document type, URL). To improve the informativeness of the result displayed in the user interface, it is useful to include in this record the keywords that were matched, i.e. the ones that helped to retrieve the specific document, as well as a relevance score produced by the search engine. This retrieval task has thus a similar goal as speech/document alignment [7, 6], except that alignment is viewed here as the construction of sets of relevant documents for each meeting segment, and not only as finding the document that the segment "is about". The retrieval techniques that are employed here are therefore quite different too, as speech/document alignment relies on precise matching between a referring expression and one of the elements of a document.

To avoid inconsistent results from one time frame to another, due to the fact that word choice varies considerably in such small samples, and therefore search results vary as well, a persistence (smoothing) mechanism was defined, partly inspired by the notion of perceptual salience of entities: the relevance of the meeting fragments and documents amounts to a form of conceptual salience that evolves in time.

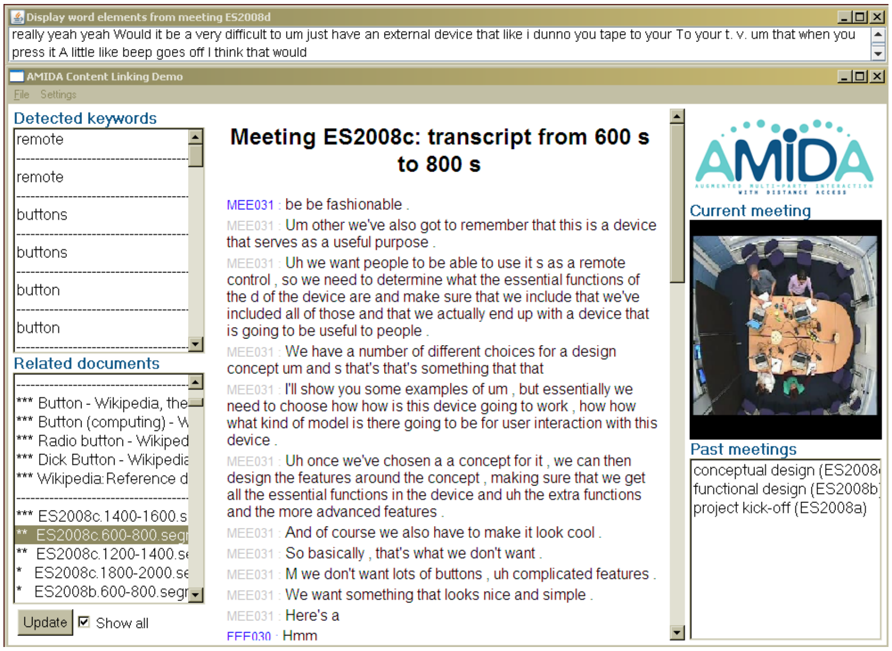
The persistence mechanism adjusts the current relevance scores for each document returned by the search engine, considering also the documents from the previous time frame and their own adjusted relevance scores. If t_n denotes the current time frame and t_{n-1} the previous one, and if $r(t_n, d_k)$ is the raw relevance of document d_k computed by the search engine after a query at time t_n, then the *adjusted relevance* $r'(t_n, d_k)$ computed using the persistence (smoothing) mechanism, is defined as follows (we note $r(t_n, d_k)$ as r_n for clarity reasons): $r'_n = \alpha * r_n + (1 - \alpha) * r'_{n-1}$ – with a typical value of $\alpha = 0.8$ being used. The larger the α factor, the more persistent the documents tend to be in the display interface, where they are sorted by relevance. Additionally, a filtering mechanism deletes the least relevant of the documents which are sent to the Hub and to the interface, using absolute and relative thresholding, as described in [24, p. 280].

Some user groups encouraged an extension of the Query Aggregator to provide web search as a complement to the existing document search. This was implemented using the Google API, which, like other web search engines but unlike Lucene, does not allow keyword boosting in queries and limits query size (32 words for Google). Therefore, the web query is built in a slightly different way: if any keywords from the pre-specified list are detected during the latest time interval, only these keywords are appended to build the query; if no keywords are detected, then all detected words are used. The web domain which is searched can be specified and changed during the meeting by the users (for the moment, it is set to http://en.wikipedia.org).

5.4 Accessing the Meeting Corpus and Documents

A number of concepts for the *User Interfaces* (UI) are currently being developed and tested: user-friendliness is one of the main criteria orienting our design. Two examples are shown here: a full-screen UI (Figure 2) that displays simultaneously, in several frames, all the information related to content linking, and a widget UI (Figure 3), which minimizes the use of the screen's real estate through the use of tabs, and the opening of documents and meeting snippets in separate viewers.

Figure 2 shows a snapshot of the full-screen UI over meeting ES2008d. On the top left, the list of recognized keywords, referring to important concepts for the group's activity, reassures the user about the search terms being used, as they were recognized from the audio. Every 30 seconds or on demand (by pressing the "update" button), a newly recognized keyword set is added in the top left frame; this also triggers and immediate query in the QA, and the update of results (documents/snippets, and websites). The bottom left frame, which scrolls in the


Display word elements from meeting ES2008d
AMIDA Content Linking Demo
File
Settings
AMIDA
AUGMENTED MULTI-PARTY INTERACTION WITH DISTANCE ACCESS
Update
☑ Show all

really yeah yeah Would it be a very difficult to um just have an external device that like i dunno you tape to your To your t. v. um that when you press it A little like beep goes off I think that would

Detected keywords

remote
remote
buttons
buttons
button
button

Related documents

*** Button - Wikipedia, the
*** Button (computing) - W
*** Radio button - Wikiped
*** Dick Button - Wikipedia
*** Wikipedia:Reference d
*** ES2008c.1400-1600.s
** ES2008c.600-800.segr
** ES2008c.1200-1400.se
* ES2008c.1800-2000.se
* ES2008b.600-800.segr

Meeting ES2008c: transcript from 600 s to 800 s

MEE031 : be be fashionable .

MEE031 : Um other we've also got to remember that this is a device that serves as a useful purpose .

MEE031 : Uh we want people to be able to use it s as a remote control , so we need to determine what the essential functions of the d of the device are and make sure that we include that we've included all of those and that we actually end up with a device that is going to be useful to people .

MEE031 : We have a number of different choices for a design concept um and s that's that's something that that

MEE031 : I'll show you some examples of um , but essentially we need to choose how how is this device going to work , how how what kind of model is there going to be for user interaction with this device .

MEE031 : Uh once we've chosen a a concept for it , we can then design the features around the concept , making sure that we get all the essential functions in the device and uh the extra functions and the more advanced features .

MEE031 : And of course we also have to make it look cool .

MEE031 : So basically , that's what we don't want .

MEE031 : M we don't want lots of buttons , uh complicated features .

MEE031 : We want something that looks nice and simple .

MEE031 : Here's a

FEE030 : Hmm

Current meeting

Past meetings

conceptual design (ES2008
functional design (ES2008b
project kick-off (ES2008a)

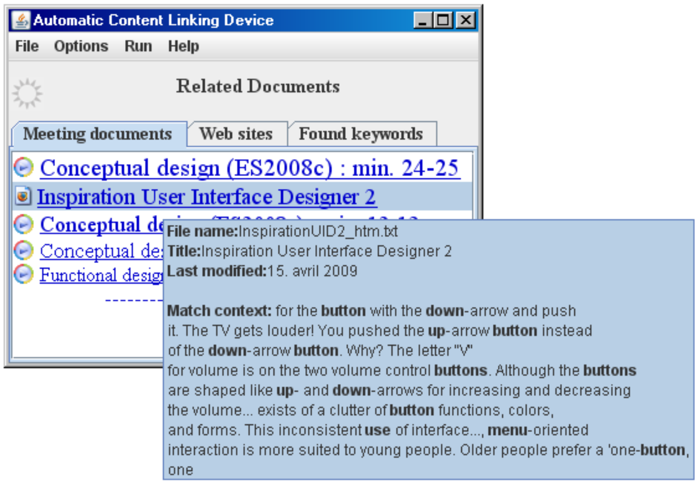
Fig. 2. Snapshot of the ACLD's full-screen UI, showing a fragment of a past meeting being retrieved among the list of relevant documents at the bottom left corner, and being opened in the central frame. The window above the ACLD displays the words as they are recognized by the ASR, or are streamed into the Hub from offline ASR.

same way as the keywords, shows the five most relevant document names for that time in the meeting, as well the five most relevant web pages, ordered by relevance. The frame can also display all past results as well, appended to the current one, if the user wishes so ("show all" checkbox at the bottom left).

At the bottom right of the full-screen UI is a static display showing the three meetings in the history, giving access to their contents, metadata and summaries. Above that, the interface displays a room-view video of the ongoing meeting, with the audio in the case of past meetings. The UI displays in the centre frame a text or HTML version of the selected document, or opens the selected web page.

Figure 3 shows a snapshot of the widget UI, taken at some time during replay of meeting ES2008d. This version of the UI is a deliberately simplified window frame reduced to show exclusively content that is actually delivered by the ACLD at runtime, split into three tabs, which contain respectively:

1. Labels of the relevant documents and past meeting snippets found in the meeting index, preceded by an appropriate icon corresponding to the document type, to support faster identification by the user.
2. Relevant web links found within the pre-specified web domain.



File name: InspirationUID2_htm.txt
Title: Inspiration User Interface Designer 2
Last modified:15. avril 2009

Match context: for the **button** with the **down**-arrow and push
it. The TV gets louder! You pushed the **up**-arrow **button** instead
of the **down**-arrow **button**. Why? The letter "V"
for volume is on the two volume control **buttons**. Although the **buttons**
are shaped like **up**- and **down**-arrows for increasing and decreasing
the volume... exists of a clutter of **button** functions, colors,
and forms. This inconsistent **use** of interface..., **menu**-oriented
interaction is more suited to young people. Older people prefer a 'one-**button**,
one

Fig. 3. Snapshot of AMIDA ACLD's widget UI, showing the tab with the list of relevant documents at a given moment, with explicit document and snippet labels. Hovering over a label displays the metadata associated with the document, as well as excerpts where the keywords were found. Tabs displaying detected keywords and retrieved websites are also available.

3. Keywords recognized in the respective system turn, or alternatively all recognized words with highlighted keywords.

The rank of a label in the document result list, as well as its font size, indicate its relevance within the query result; for web search results, the only indicator of relevance is the rank. Hovering over a result link (document, meeting snippet, web link) provides metadata about it in a pop-up window, including most importantly the *match context* shown in Figure 3. The match contexts which shows excerpts of the document that match keywords and words detected from speech, with surrounding words. Clicking on a link opens the respective document using an appropriate viewing program – respectively, a native editor, a meeting browser (JFerret, a successor to Ferret [25]), or a web browser.

6 Evaluation

Several approaches were considered for the evaluation of the ACLD application, but all of them require a significance investment of resources, in terms of human subjects or of ground truth data, which amounts to human annotators as well. If reference-based and task-based evaluation methods are distinguished, then the first two of the following approaches are reference-based, and the last two are task-based:

1. Construct ground truth data by asking human annotators to associate to each meeting segment the documents they believe are relevant, from a pool of limited size; then, check whether the ACLD actually finds those documents.
2. Ask human judges to assess the relevance of each document returned by the ACLD for a given meeting recording.
3. Test the ACLD *in use* on the participants to an ongoing meeting, by measuring for instance how often and for how long they consult the proposed documents.
4. Assess user-satisfaction with the ACLD (e.g. using questionnaires or interviews) after using the application in a meeting, or after seeing a demonstration of the concept.

The first two approaches provide "objective" measures of performance, in terms of precision and recall, but they evaluate the performance of the retrieval engine (Lucene and the persistence model) rather than the entire application, which includes a larger set of components. Therefore, the task-based or user-centric evaluation appears to be more suitable for evaluating the entire application, by assessing its utility for a specific use. Of course, many factors beyond the concept and the technology contribute to overall utility of an application, such as the user-friendliness of the UI, the IT literacy of the users, and so on.

In the two following sub-sections, we summarize initial evaluation results obtained using the fourth approach from the list above, while in the last sub-section we outline the setting that will be used for the third approach, evaluation in use.

6.1 Feedback from Focus Groups

The ACLD concept was shown to two focus groups of about eight persons each, one concerned with the military domain, and the other with meetings in large organizations [26].

The groups found the general concept of content linking useful for groups that have a large information-base from which to draw material, but suggested that content be linked only on demand, not continually. The content linking concept was seen as adding value also, or sometimes mostly, for the individual meeting participant, by helping them retrieve information that has been forgotten and even catch up with parts of a meeting missed through inattention or non-attendance. Access to personally held information was seen as useful, but so was sharing personal linked content with the group, as long as it would be possible to separate private and common spaces. The capacity to add one's own keywords to the system was seen as crucial, although it appeared clearly that keywords alone could not be used exclusively to find the right information. Other search aids that could enrich the ACLD should focus on information about the people involved in meetings, and other meeting metadata such as location and attendance. One group thought that it would be useful to look back at the documents or presentations used during previous meetings; documents should be identified by author, version number, date, and time. To avoid distracting the user, the ACLD needs to work as much as possible autonomously in the background.

6.2 User Feedback to Prototype Demonstrations

We have demonstrated the ACLD to potential industrial partners, namely, about forty representatives of companies that are active in the field of meeting technology. Typically, a series of sessions, lasting 30 minutes each, started with a presentation of the ACLD and continued with a discussion, during which notes were taken. The participants found that both online and offline application scenarios seemed promising, as well as both individual and group uses. The ACLD received very positive verbal evaluation, as well as useful feedback and suggestions for future work [27, p. 18-21]. Several companies are negotiating with us to demonstrate the technology on their own meetings.

6.3 Evaluation in Use: the TBE Setup

We have selected the same Task-Based Evaluation (TBE) method that we have previously used to evaluate meeting browsers [28, 29, p. 8–18 and 31–35] for the ACLD meeting assistant as well. The protocol makes use of the AMI Meeting Corpus, and requires a group of four subjects to carry out a remote control design task that was started by a previous group recorded in the corpus, from which three recorded meetings (‘a’, ‘b’ and ‘c’ from a series) and the related documents are available. In the original TBE campaign, we tested various groups that either only had access to the documents that came out of the previous meetings, or could use one of three types of meeting browser. The evaluation measures were the efficiency of the groups in solving the task, their satisfaction with the tools, but also the analysis of tool usage in itself. These experiments provide a “baseline” to which groups carrying out a similar task, but using now the ACLD to access previous recordings and documents, can be compared, so that the benefits of the ACLD can be assessed. We are currently at the stage of pilot testing this form of evaluation, and adapting the ACLD software to the testing conditions.

7 Conclusion and Future Work

The ACLD is currently a research prototype that demonstrates how access to a very large corpus can be made easier in a context where automatic, query-free retrieval can be used to facilitate access to past meetings and documents from ongoing meetings. The complexity of the demonstrator, and the challenges of real-time processing, were gradually solved by using offline annotations from the AMI Meeting Corpus. Therefore, at present, the ACLD offers a sound platform for experimenting with access techniques to multimodal corpora, to which new data processing modules can be added, followed by tests of their effectiveness in improving retrieval or visualisation of corpus data.

The first implementation of the ACLD allowed us to demonstrate the concept and to collect feedback that tells us what is most important for turning it into a real-world application. For instance, potential users have made clear that the graphical layout of the interface is important, leading us to experiment with the full-screen and widget UIs. A general goal is to reduce the number of mouse clicks required to access the content of documents. Color-coding the document types and displaying their relations to the (key)words recognized from the audio might also improve user experience, according to the feedback obtained.

Potential users suggested some additional functionalities that we are considering. For instance, keeping a record of the documents that were consulted during a meeting might help users who want to go back to them after the meeting, and is of course essential for evaluation. Retrieval could be improved by including a relevance feedback mechanism for the returned documents, by representing keywords in a structured manner, e.g. using tag clouds, and by using word sense disambiguation to improve the precision of the retrieval.

The ACLD could also be useful, in the future, in a series of meetings to understand what topics re-occur, when a discussion is being repeated, and also which topics never get discussed because they are forgotten. Tag clouds for topics can help to support this understanding. This kind of functionality would take the concept from information presentation towards intervention in the current group discussion, and could make the ACLD part of a broader-scope meeting assistant.

Modern recording techniques open out many possibilities for new technologies that exploit archived multimodal data. Our Automatic Content Linking Device is just one possible example, but one that has excited the interest of companies that work in the area of meeting support or are potential consumers of such technology. We relied on a recorded corpus throughout the process of creating it – as inspiration, to capture user requirements, and as a platform for development, demonstration, and evaluation. The same process could be applied for a wide range of novel applications that exploit the same kind of underlying resource.

Acknowledgments

This work was supported by the European Union's IST Programme, through the AMIDA Integrated Project FP6-0033812, Augmented Multiparty Interaction with Distance Access, and by the Swiss National Science Foundation, through the IM2 National Center of Competence in Research. The authors would like to thank here their colleagues from the ACLD and Hub development teams: Erik Boertjes, Sandro Castronovo, Michal Fapso, Mike Flynn, Alexandre Nanchen, Theresa Wilson, Joost de Wit, and Majid Yazdani, as well as Danil Korchagin and Mike Lincoln for help with real-time ASR.

References

1. Hart, P.E., Graham, J.: Query-free information retrieval. IEEE Expert: Intelligent Systems and Their Applications **12**(5) (1997) 32–37

2. Rhodes, B.J., Starner, T.: The remembrance agent: A continuously running information retrieval system. In: PAAM 1996 (1st International Conference on Practical Applications of Intelligent Agents and Multi-Agent Technology), London (1996) 486–495
3. Rhodes, B.J., Maes, P.: Just-in-time information retrieval agents. IBM Systems Journal **39**(3-4) (2000) 685–704
4. Budzik, J., Hammond, K.J.: User interactions with everyday applications as context for just-in-time information access. In: IUI 2000 (5th International Conference on Intelligent User Interfaces), New Orleans, LA (2000)
5. Henziker, M., Chang, B.W., Milch, B., Brin, S.: Query-free news search. World Wide Web: Internet and Web Information Systems **8** (2005) 101–126
6. Popescu-Belis, A., Lalanne, D.: Reference resolution over a restricted domain: References to documents. In: ACL 2004 Workshop on Reference Resolution and its Applications, Barcelona (2004) 71–78
7. Mekhaldi, D., Lalanne, D., Ingold, R.: From searching to browsing through multimodal documents linking. In: ICDAR 2005 (8th International Conference on Document Analysis and Recognition), Seoul (2005) 924–928
8. Nijholt, A., Rienks, R., Zwiers, J., Reidsma, D.: Online and off-line visualization of meeting information and meeting support. The Visual Computer **22**(12) (2006) 965–976
9. Rhodes, B.J.: The wearable remembrance agent: A system for augmented memory. Personal Technologies: Special Issue on Wearable Computing **1** (1997) 218–224
10. Franz, A., Milch, B.: Searching the Web by voice. In: Coling 2002 (19th International Conference on Computational Linguistics), Taipei (2002) 11–15
11. Chang, E., Seide, F., Meng, H.M., Chen, Z., Shi, Y., Li, Y.C.: A system for spoken query information retrieval on mobile devices. IEEE Transactions on Speech and Audio Processing **10**(8) (2002) 531–541
12. Garofolo, J.S., Auzanne, C.G.P., Voorhees, E.M.: The TREC spoken document retrieval track: A success story. In: RIAO 2000 (6th International Conference on Computer-Assisted Information Retrieval), Paris (2000) 1–20
13. Lew, M., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) **2**(1) (2006) 1–19
14. AMI Consortium: The AMI Meeting Corpus, http://corpus.amiproject.org (accessed 18 November 2008)
15. Carletta, J.: Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus. Language Resources and Evaluation Journal **41**(2) (2007) 181–190
16. Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., Lathoud, G., Lincoln, M., Lisowska, A., McCowan, I., Post, W., Reidsma, D., Wellner, P.: The AMI Meeting Corpus: A pre-announcement. In Renals, S., Bengio, S., eds.: Machine Learning for Multimodal Interaction II. LNCS 3869. Springer-Verlag, Berlin/Heidelberg (2006) 28–39
17. AMI Consortium: The AMI multimodal meeting database – infrastructure, data and management. Deliverable 2.2, AMI (Augmented Multi-party Interaction) Integrated Project FP6506811 (August 2005)
18. Whittaker, S., Tucker, S., Swampillai, K., Laban, R.: Design and evaluation of systems to support interaction capture and retrieval. Personal and Ubiquitous Computing **12**(3) (2008) 197–221

19. AMI Consortium: Commercial component definition. Deliverable 7.2, AMIDA (Augmented Multi-party Interaction with Distance Access) Integrated Project IST033812 (November 2007)
20. Carletta, J., Evert, S., Heid, U., Kilgour, J.: The NITE XML Toolkit: Data model and query language. Language Resources and Evaluation **39**(4) (2005) 313–334
21. Hain, T., Burget, L., Dines, J., Garau, G., Karafiat, M., Lincoln, M., Vepa, J., Wan, V.: The AMI system for the transcription of speech in meetings. In: ICASSP 2007 (32nd International Conference on Acoustics, Speech, and Signal Processing), Honolulu (2007) 357–360
22. Garner, P.N., Dines, J., Hain, T., El Hannani, A., Karafiat, M., Korchagin, D., Lincoln, M., Wan, V., Zhang, L.: Real-time asr from meetings. Technical report (2009)
23. Szoke, I., Schwarz, P., Matejka, P., Burget, L., Karafiat, M., Fapso, M., Cernocky, J.: Comparison of keyword spotting approaches for informal continuous speech. In: Eurospeech 2005 (9th European Conference on Speech Communication and Technology), Lisbon (2005) 633–636
24. Popescu-Belis, A., Boertjes, E., Kilgour, J., Poller, P., Castronovo, S., Wilson, T., Jaimes, A., Carletta, J.: The AMIDA automatic content linking device: Just-in-time document retrieval in meetings. In Popescu-Belis, A., Stiefelhagen, R., eds.: Machine Learning for Multimodal Interaction V (Proceedings of MLMI 2008, Utrecht, 8-10 September 2008). LNCS 5237. Springer-Verlag, Berlin/Heidelberg (2008) 272–283
25. Wellner, P., Flynn, M., Guillemot, M.: Browsing recorded meetings with Ferret. In Bengio, S., Bourlard, H., eds.: Machine Learning for Multimodal Interaction I. LNCS 3361. Springer-Verlag, Berlin/Heidelberg (2004) 12–21
26. AMI Consortium: HCI evaluation of prototype applications. Deliverable 6.3, AMIDA (Augmented Multi-party Interaction with Distance Access) Integrated Project IST033812 (October 2008)
27. AMI Consortium: AMIDA proof-of-concept system architecture. Deliverable 6.7, AMIDA (Augmented Multi-party Interaction with Distance Access) Integrated Project IST033812 (March 2008)
28. Post, W.M., Elling, E., Cremers, A.H.M., Kraaij, W.: Experimental comparison of multimodal meeting browsers. In: HCII 2007 (12th International Conference on Human-Computer Interaction), Beijing (2007)
29. AMI Consortium: Meeting browser evaluation. Deliverable 6.4, AMI (Augmented Multi-party Interaction) Integrated Project FP6506811 (December 2006)