The Ambient Spotlight: Queryless Desktop Search From Meeting Speech

Jonathan Kilgour
School of Informatics
University of Edinburgh
10 Crichton Street, Edinburgh
jonathan@inf.ed.ac.uk

Jean Carletta
School of Informatics
University of Edinburgh
10 Crichton Street, Edinburgh
jeanc@inf.ed.ac.uk

Steve Renals
School of Informatics
University of Edinburgh
10 Crichton Street, Edinburgh
srenals@inf.ed.ac.uk

ABSTRACT

It has recently become possible to record any small meeting using a laptop equipped with a plug-and-play USB microphone array. We show the potential for such recordings in a personal aid that allows project managers to record their meetings and, when reviewing them afterwards through a standard calendar interface, to find relevant documents on their computer. This interface is intended to supplement or replace the textual searches that managers typically perform. The prototype, which relies on meeting speech recognition and topic segmentation, formulates and runs desktop search queries in order to present its results.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

speech, calendar, meeting, retrieval

1. INTRODUCTION

Now that microphone array technology is becoming cheaper and more portable, it is feasible to develop practical personal applications. In this demonstration paper we describe a prototype for one such application, *the Ambient Spotlight*, that is designed as an aid for project managers. Project managers spend most of their time in meetings, preparing for meetings (which often means reviewing what happened previously), or acting on them. Documents come to them in all kinds of ways - email, intranets, version control repositories, memory sticks - and just getting these in some semblance of order so they can be found reliably can be a major headache. One coping strategy is to make sure that everything is at least somewhere on one machine, typically a laptop, and use, for instance, Spotlight or Google Desktop to search it — but it typically takes managers several attempts to formulate the right textual search string.

The idea behind the Ambient Spotlight is help project managers find the documents that are relevant to a past meeting without having to formulate queries explicitly. The Ambient Spotlight uses information from a calendar application as a natural source of structure for the working life of the user, and the recorded speech of these meetings as its source of information about what happens in each meeting. Audio is passed through speech recognition software and the resulting transcript is segmented into 20 second chunks, with each chunk being analysed to produce a query for documents on the user's laptop. In this context, documents can be anything containing text, from emails to slide presentations to PDF documents. These results are too low-level to present directly so they are aggregated to provide the most popular results over an entire meeting, or, more usefully, over automatically derived topic segments from the meeting.

Our prototype uses Google Calendar and Spotlight for the calendar and desktop search, respectively, although there are other off-the-shelf technologies that we could have employed. Most of the rest of our demonstration adapts components previously developed by the AMI and AMIDA European projects [8]. We briefly describe each and what we needed to do to use them in this way, as well as the Ambient Spotlight's end user interface. We then describe its operation, give some examples and discuss what we have learned from the prototype.

2. COMPONENT TECHNOLOGIES

The component technologies that we have used include meeting recording; speech recognition; turning words into queries; finding useful topic boundaries; and aggregating results.

2.1 Meeting Recording

Serious meeting speech recognition attempts began in around 2000, but relied on wearable devices like lapel or headset microphones. More recently, speakers have been liberated from wiring by the development of special-purpose instrumented meeting rooms [8], [9]. Although the speech recognition that can be achieved under these arrangements is technically suitable for many applications, access to the rooms limits their deployment in the same way that has previously been observed for videoconferencing suites. Moreover, many meetings are fairly impromptu, and take place in offices or over coffee. For any one person, many of the useful interactions that take place in working meetings could not conceivably be scheduled in a shared meeting facility. In the past, this has made personal meeting speech applications particularly

SSCS10 2010 Florence, Italy
Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...$10.00.

difficult. New portable microphone arrays, like room-based arrays, can compare the signals over the set of individual microphones to help assign speech to speakers. They can also use beamforming to boost the relevant parts of a signal and improve recognition results. Unlike room-based arrays, they can be deployed without specialist support and in less formal settings.

The particular microphone array we are using for our prototype is the Microcone^{TM} from Dev-Audio [1], a portable USB recording device featuring built-in beamforming software. It is shown in Figure 1.

2.2 Speech Recognition

The AMI Consortium produced three different automatic speech recognition (ASR) systems for meeting speech [3] that we can use. Ordered by the increasing time it takes them to run, and correspondingly their increasing accuracy, they are:

- Realtime speech recognition running on the manager's laptop
- WebASR - a web-based speech recognition service [4]
- Full ASR running *in the cloud*

The three levels of ASR are not strictly alternatives: we can imagine the document results for a meeting changing over time as the different ASR processes complete.

Results from ASR are fed into *The Hub* [7], which allows time-aligned annotations of signals to be transferred between software modules in real-time, and also stores all annotations so they are available for future query.

2.3 Turning Speech Results into Queries

Speech results are read from the Hub as they arrive, and for each 20 second period, they are turned into textual queries over the documents on the manager's machine. The process of deriving queries from ASR output is adapted from [7], but instead of relying on a stoplist and a set of keywords, we filter the ASR output through a TFIDF (term frequency inverse document frequency) threshold process to remove the more common words. The only keywords we use are the meeting names and descriptions obtained from the manager's calendar using the Google API.

The tool we use to query the documents on the machine is called *mdfind*, which is essentially the command-line equivalent of the Mac's *Spotlight* tool. Alternative indexing and searching approaches could include *Google Desktop* or *Lucene*. The advantage of using a native tool like *mdfind* over *Lucene* is that its index is already present and integrated over the whole content of the machine: relevant email messages and elements of web-browsing history are just as likely to be search results as are PDF documents. *Google Desktop* similarly creates indexes automatically but its API is not available on the Mac platform on which we are prototyping. Our software makes it easy to swap in different index and search approaches.

Our current approach is to run queries where the search terms are *and*-ed together. For some 20 second portions of the meeting, the threshold function may allow too many words through, with the result that no documents are returned. Currently there is no check for that. It would of course be possible to alter the threshold to filter out more words in this case, or to apply a less restrictive form of search.

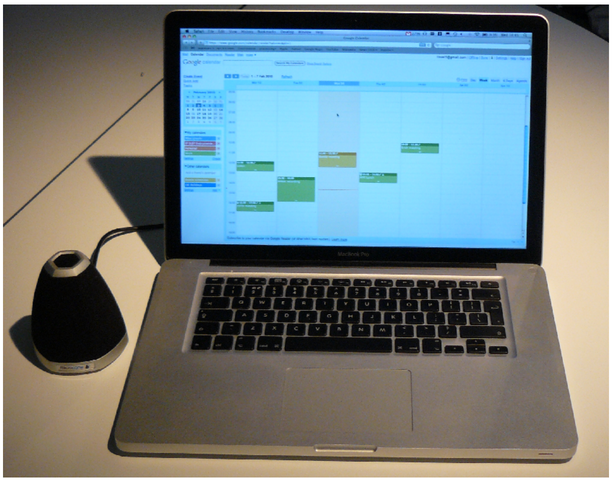


Figure 1: Microcone with laptop

2.4 Generating Topic Segments

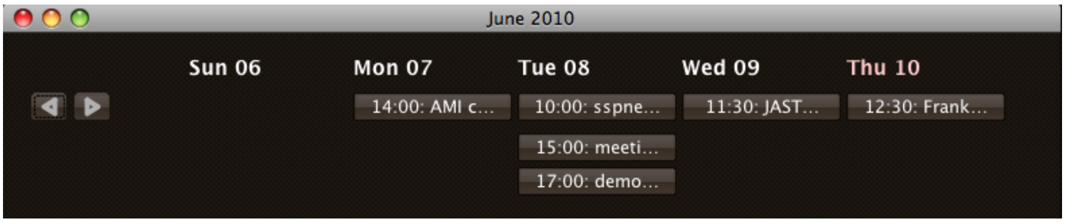
The first prototype of the Ambient Spotlight showed the most common document results aggregated over the entire meeting. This seemed to provide fairly relevant results, but they were rather too general in nature and some better documents that appeared in a few return lists were being "averaged out". In order to reduce this effect we decided to aggregate at some less coarse level. For this, topic segments is the natural choice, since different documents may be relevant for different topic segments anyway. Moreover, meeting participants naturally think of meetings as being divided into topics, making it useful to display them in the end user interface.

To obtain a topic segmentation for recorded meetings, we use a process based on based on [5] that segments a meeting and derives labels for topics using only lexical features. An example of topics generated in this way is shown in Figure 3. Adding topic segmentation to the processing stream does indeed seem to improve the set of returned documents.

3. INTERFACE

The main component of the Ambient Spotlight as far as the user is concerned is the calendar display, shown in Figure 2. This is a simple representation of the manager's Google Calendar and as such provides a familiar interface for the manager to review his meetings. The Archivus system [6] represented meetings as books in a library and while that may be an effective metaphor for certain applications, in the personal space a calendar interface seems much more natural.

The initial view of the calendar shows the current date furthest to the right with older meetings to the left. This reinforces the idea that this is a tool for reviewing past meetings. Users can skip forward and back in the calendar a day or a week at a time, and mouse over each meeting to see the full name and any description that exists. If a future or current meeting is clicked that has no associated ASR in the Hub, users can then choose to start recording using *Microcone Recorder* - software that comes with the Micro-


June 2010

Sun 06

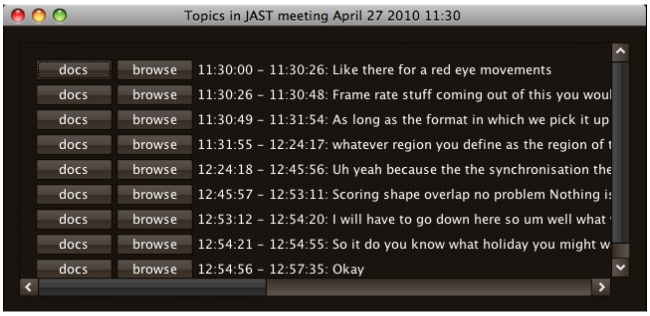
Mon 07
14:00: AMI c...

Tue 08
10:00: sspne...
15:00: meeti...
17:00: demo...

Wed 09
11:30: JAST...

Thu 10
12:30: Frank...

Figure 2: The Ambient Spotlight Calendar Display


Topics in JAST meeting April 27 2010 11:30

docs browse 11:30:00 - 11:30:26: Like there for a red eye movements
docs browse 11:30:26 - 11:30:48: Frame rate stuff coming out of this you woul
docs browse 11:30:49 - 11:31:54: As long as the format in which we pick it up
docs browse 11:31:55 - 12:24:17: whatever region you define as the region of t
docs browse 12:24:18 - 12:45:56: Uh yeah because the the synchronisation the
docs browse 12:45:57 - 12:53:11: Scoring shape overlap no problem Nothing i:
docs browse 12:53:12 - 12:54:20: I will have to go down here so um well what
docs browse 12:54:21 - 12:54:55: So it do you know what holiday you might w
docs browse 12:54:56 - 12:57:35: Okay

Figure 3: The Topic Display

cone. Starting to record is then a very simple process: optionally listing the meeting participants (from a known list or by adding new names) and clicking the record button. Once recording has started there is an unobtrusive display on the screen to indicate activity and to allow the user to stop recording. Recording could instead be managed automatically, but the possible methods for doing this have some disadvantages. The system could record whenever the laptop is on with the MicroCone attached, sending signals for speech recognition in arbitrary chunks, but this could bloat the amount of disk space and compute power needed by processing speech that is irrelevant for document retrieval. It also risks capturing material that is more more personal than the manager intended. Alternatively, managers could indicate whether a meeting is to be recorded when they add it to their calendar, and the system could record automatically using the declared meeting start and end times. In this arrangement, the system would fail to record the ends of meetings that run longer than expected. Moreover, if most recording is automatic, the manager may then forget to record important impromptu meetings.

Clicking on a meeting name that does have associated ASR and topic segmentation brings up the topic display as shown in Figure 3. This displays the output of the topic segmentation process with the timings and label of each topic to the right of a pair of buttons.

Clicking on the *Browse* button pops up a meeting browser for the meeting and makes it jump to the appropriate point. Our meeting browser is a useful outcome of our automatic meeting processing in its own right. It allows users to play back their recorded audio in sync with the generated ASR, and also to browse through the meeting by topic segments. Further than that it displays the result of another automatic process: extractive summarisation, allowing users to excise utterances deemed less important by moving a sliding scale.

Clicking on the *Docs* button pops up a display of the most returned documents form that period of the meeting. This is done by simply counting all the document results that occurred during the topic and finding the most-returned.

This then allows us to pop up relevant documents of whatever kind have been returned. When the user clicks on a particular document, it will open in its native application so that everything will look familiar. For example, if the document is an email the mail program is started and the relevant email displayed so the user can send a response, or a reminder to the rest of the group.

4. OPERATION

Behind the scenes, the Ambient Spotlight queries the user's Google Calendar to populate the calendar display using the Google Data APIs [2]. Moving backward and forward through the calendar may cause further queries to the Google Calendar.

Each meeting is assigned a unique identity which is recorded as an attribute of the meeting within the Google Calendar appointment itself. Any meetings that have not been assigned such an ID will be given one on startup, again using the Google Data API. These IDs are also used in the Hub database to identify annotations on the meetings and are unique within a user's calendar.

When a meeting is clicked in the calendar window (Figure 2), a query is sent to the Hub to determine if there are topics and ASR available for the meeting. The first time one of the *docs* buttons for a meeting is pressed, all *LinkedContent* elements in the Hub are retrieved for the whole meeting, and they are recorded for processing. Then the only task for successive *docs* clicks is to calculate the most-returned documents for the relevant time period.

For the three different approaches to speech recognition described above, the process to start recognition is only thusfar automated for running live ASR on the laptop. This is simply a case of starting up another Java process using Ant. Speech recognition using WebASR is in fact fairly tightly integrated into the Microcone Recorder software, but currently the process of passing recorded audio to that process, and indeed to ASR in the cloud, is manual. When these processes return ASR a simple script needs to be run to process the output and send it to the Hub. This should be quite easy to automate.

Higher level processes to derive and run *Spotlight* queries on the user's laptop; to create topic segments and put them into the Hub; and to create a browser for each meeting also currently require a small degree of manual intervention.

5. DISCUSSION

In order to test the prototype informally, we cloned a laptop belonging the second author. Among other duties, she takes an active part in multiple research projects, both technically and as a scientific manager. Therefore she has many documents on her machine, covering a wide variety of subjects. The machine clone includes all emails along with all the standard documents and directories present on the machine. Our test uses a set of six recorded meetings, choosing them from a spread of projects so that we can more easily judge the relevance of the query results. Five of them have been recorded using the Microcone, and one, for comparison, using an instrumented meeting room.

For example, consider the following 20 second extract from a meeting about eyetracking research.

> ..events. there's Mm-hmm. fixations and blinks now at the moment as Right. well. Um and, yeah, and oh yeah and see uh looks at looks at objects and Mm-hmm. looks whatever else. So Okay. But I um think the s thing to do is um for you to go away and think about it this way with the different tracks for the different objects and the um..

Once this input has been passed through the TFIDF process we are left with the five word query

> fixation blink right object track

This query produces about 30 document results, most of which are papers on eyetracking written either by the manager or by others. Some of the results from this 20 second portion of the meeting were returned sufficiently often to be retained in the list of top documents for the topic. In some sense, this is a “top line” example; this meeting is from the meeting room, the processing uses the highest quality ASR, and the speaker did use some real content words during the 20 seconds. However, in general, it does seem that there is a reasonably good distinction between projects even when using Microcone output and real-time recognition running on the laptop itself. That is to say that documents from the relevant project, and related ones, are prominent in the result sets. Results are better when descriptive meeting names are entered in the Google Calendar so that they can be used as search keywords.

Currently, we are manipulating a number of variables in order to determine what will give us the best results on our cloned example machine, where our subjective judgment of what is best depends on our knowledge of the owner of the cloned machine and of the meetings she attends. Our manipulations include comparing results using the three different ASR engines; averaging results over different time-periods; using different indexing and search techniques; and adapting to situations where there are too many or too few results.

As with other personal applications, formal evaluation is tricky. Although we could devise some standardized task for experimental subjects and compare their performance using our interface to what they do using the best available alternatives, this would not necessarily tell us whether our concept works for real users. Managers vary greatly in their level of self-organisation and in how they review and act upon meetings, and so one way of populating the laptop and one standardized task would only tell us how the interface suits a small and difficult to identify subset of the target user community. In this situation, it is more appropriate to field test the application with a range of target users. We have not yet conducted field tests of this sort.

The Ambient Spotlight shows that by combining newly available hardware and software technologies, we can begin to develop personal applications that utilize recognised speech from meetings. The fact that we can link relevant documents from a set of example meetings in the demonstrator, even with imperfect ASR, points toward applications that don't display recognised transcripts to users at all, instead using them to derive a set of higher-level features that could provide direct assistance to users in an ambient fashion.

6. ACKNOWLEDGMENTS

This work was supported by a University of Edinburgh IDEA Lab POPP project.

7. REFERENCES

[1] Dev-Audio. Dev-audio intelligent audio devices, http://www.dev-audio.com/, accessed 14 July 2010.

[2] Google. Google calendar apis and tools, http://code.google.com/apis/calendar/, accessed 08 June 2010.

[3] T. Hain, L. Burget, M. Karafiat, J. Dines, D. van Leeuwen, G. Garau, M. Lincoln, and V. Wan. The 2007 ami(da) system for meeting transcription. In *Rich Transcription 2006 Spring (RT07s) Meeting Recognition Evaluation*, Baltimore, USA, May 2007.

[4] T. Hain, A. E. Hannani, S. Wrigley, and V. Wan. Automatic speech recognition for scientific purposes - webasr. In *Interspeech 2008*, Brisbane, Australia, September 2008.

[5] P. Hsueh and J. Moore. Automatic topic segmentation and lablelling in multiparty dialogue. In *First IEEE/ACM workshop on Spoken Language Technology (SLT 2006)*, Aruba, 2006.

[6] A. Lisowska, M. Rajman, and T. Bui. *ARCHIVUS: A System for Accessing the Content of Recorded Multimodal Meetings*, pages 291–304. Springer, 2005.

[7] A. Popescu-Belis, J. Carletta, J. Kilgour, and P. Poller. Accessing a large multimodal corpus using an automatic content linking device. In *Multimodal corpora: from models of natural interaction to systems and applications*, pages 189–206. Springer-Verlag, Berlin, Heidelberg, 2009.

[8] S. Renals, T. Hain, and H. Bourlard. Interpretation of multiparty meetings: The AMI and AMIDA projects. In *IEEE Workshop on Hands-Free Speech Communication and Microphone Arrays, 2008. HSCMA 2008*, pages 115–118, 2008.

[9] A. Waibel and R. Stiefelhagen. *Computers in the Human Interaction Loop.* Springer, 2009.