The Ambient Spotlight: Queryless Desktop Search from Meeting Speech

Citation for published version:

Digital Object Identifier (DOI):
10.1145/1878101.1878112

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the 2010 International Workshop on Searching Spontaneous Conversational Speech

General rights
Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.
The Ambient Spotlight: Queryless Desktop Search From Meeting Speech

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.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SSCS10 2010 Florence, Italy
Copyright 20XX ACM X-XXXX-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\textsuperscript{TM} from Dev-Audio \cite{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 \cite{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 \cite{4}
- Full ASR running in the cloud

The three levels of ASR are not strict 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 \cite{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 \cite{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 name 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 over the documents on the manager’s machine. The process of deriving queries from 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.

The initial view of the calendar shows the current date further 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-
Start recording 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 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 tech-
nically and as a scientific manager. Therefore she has many
documents on her machine, covering a wide variety of sub-
jects. The machine clone includes all emails along with all
the standard documents and directories present on the ma-
chine. 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 Mn-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 Mn-hmm. looks whatever else. So
Okay. But I um think the thing to do is um find
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 pro-
cess 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 man-
ger 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 re-
sult 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 manip-
ulations 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 be-
gin 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 demon-
strator, even with imperfect ASR, points toward applica-
tions 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://code.google.com/apis/calendar/, accessed 08
June 2010.
ami(da) system for meeting transcription. In Rich
Transcription 2006 Spring (RT07s) Meeting
Automatic speech recognition for scientific purposes -
webasr. In Interspeech 2008, Brisbane, Australia,
September 2008.
and labelling in multiparty dialogue. In First
IEEE/ACM workshop on Spoken Language Technology
System for Accessing the Content of Recorded
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,
[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.