The ACLD: Speech-based Just-in-Time Retrieval of Meeting Transcripts, Documents and Websites

Andrei Popescu-Belis
Idiap Research Institute
Rue Marconi 19, CP 592
1920 Martigny, Switzerland
apbelis@idiap.ch

Jonathan Kilgour
HCRC, Univ. of Edinburgh
10 Crichton Street
Edinburgh EH89AB, Scotland
jonathan@inf.ed.ac.uk

Alexandre Nanchen
Idiap Research Institute
Rue Marconi 19, CP 592
1920 Martigny, Switzerland
ananchen@idiap.ch

Peter Poller
DFKI GmbH
Stuhlsatzenhausweg 3
66123 Saarbrücken, Germany
peter.poller@dfki.de

ABSTRACT

The Automatic Content Linking Device (ACLD) is a just-in-time retrieval system that monitors an ongoing conversation or a monologue and enriches it with potentially related documents, including transcripts of past meetings, from local repositories or from the Internet. The linked content is displayed in real-time to the participants in the conversation, or to users watching a recorded conversation or talk. The system can be demonstrated in both settings, using real-time automatic speech recognition (ASR) or replaying offline ASR, via a flexible user interface that displays results and provides access to the content of past meetings and documents.

Categories and Subject Descriptors: H.3.3 [Information Storage & Retrieval]: Information Search & Retrieval; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems

General Terms: Design, Human factors

Keywords: just-in-time retrieval, speech-based IR, multimedia IR

1. INTRODUCTION

Enriching a conversation with related content, such as audio-visual or text documents, is a task with multiple applications. We introduce the Automatic Content Linking Device (ACLD), a system that analyzes spoken input from one or more speakers, and retrieves linked content in real time from several repositories, such as archives of multimedia meeting recordings, document databases, and websites. The ACLD seeks to maximize the relevance of the retrieved documents, but also to ensure their presentation to users in an unobtrusive but understandable manner, through a flexible user interface. In this paper, we first describe scenarios of use (Section 2) and then outline comparable achievements (Section 3). The components of the ACLD are described in Section 4. Evaluation results from three perspectives are finally discussed (Section 5).

2. SCENARIOS OF USE

Spontaneous information retrieval, i.e. finding useful documents without the need for a user to initiate a direct search for them, is one of the ways in which the large quantity of knowledge that is available in networked environments can be efficiently put to use. Users are free to consult the suggested documents if they feel the need for such additional information, or they can ignore them otherwise.

One of the main scenarios of use of the ACLD involves people taking part in meetings. Meeting participants often mention documents containing facts that are currently discussed, but do not have the time to search for them during the conversation flow. Among these documents, audio-visual recordings of past meetings form a specific class, as they contain highly relevant information, but are seldom available for search, and have never been used for just-in-time retrieval.

The ACLD searches a database of meeting transcripts and other documents in the background, and keeps results at hand, in case participants need to refer quickly to them, which might happen at crucial moments during a meeting. The ACLD was developed on meetings from the AMI Corpus [2], with the possibility to demonstrate it over any of the 171 meetings of the corpus. By replaying live one of these meetings, the ACLD can be run even when no meeting takes place. The system can also be demonstrated live with one speaker, for instance while explaining the demo itself.

Another scenario of demonstration involves content linking over recorded courses (a Java course in the demo). The advantage of real-time content linking over a more static enrichment is that students can tune the parameters of the ACLD to suit their current needs, e.g. by introducing new keywords or new local repositories, or changing the web domain for search. The ACLD does not pre-compute any results, and does not use manual editing of the linked content.

3. JUST-IN-TIME RETRIEVAL SYSTEMS

Among the first antecedents to content linking, the Fixit query-free search system [5] and the Remembrance Agent for just-in-time retrieval [10] stand out. Fixit is an assistant to an expert diagnostic system for a given line of products, which monitors the state of a user's interaction with the diagnostic system, and runs searches on a repository of maintenance manuals to provide support information; the results of the searches are pre-computed based on the possible interaction states. The Remembrance Agent, which is closer to the ACLD, is an Emacs plugin that performs searches at regular time intervals (every few seconds) using a query that is based on the last words typed by the user (e.g. a buffer of 20–500 words). Results from repositories of emails or text notes are displayed and updated so that users receive them, ideally, exactly when they need them. The creators of the Remembrance Agent have also designed Jimminy, a wearable assistant that helps users taking notes and accessing information when they cannot use a keyboard. To that effect, Jimminy uses a number of contextual capture devices, but the use of speech was not implemented, and topic detection had to be simulated.

The Watson system [1] also 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 then directed to a web search engine. Besides automatic queries, Watson also allows users to formulate explicit ones, and disambiguates them using the selected terms. Another query-free system was designed for enriching television news with articles from the web [6], based on queries derived from closed-captioning text. Of course, many other speech-based search engines and multimedia information retrieval systems have been proposed in the past decade, and inspiration from their technology – which is not *per se* query free – can also serve to design just-in-time retrieval systems.

The FAME interactive space [8], which provides multimodal access to recordings of lectures via a table top interface, has many similarities to the ACLD. The main difference is that the information retrieval function required the use of specific (voice) commands, and was not spontaneously using the flow of conversation; the commands could only be issued by a special user called the manager, whose words were subject to ASR. Other related systems are the Speech Spotter [4] and the personal assistants using dual-purpose speech [7], which enable users to search for information using a small number of commands that are automatically identified in the user's speech flow.

4. DESCRIPTION OF THE ACLD

The ACLD comprises the following inter-connected modules (described in more detail elsewhere [9]): document database preparation; query construction; search and integration of results; and the user interface. The ACLD performs searches at regular intervals over the database of transcribed speech and documents, with a search criterion that is constructed based on the words that are recognized automatically from an ongoing discussion or monologue, and displays the results in the user interface.

4.1 Document Preparation and Indexing

The preparation of the local database of documents that will be accessible to content linking involves mainly the extraction of text, and the indexing of the documents.

Recordings of past discussions are valuable in subsequent ones, therefore the audio of past meetings is passed through the ASR module in offline mode and the resulting text is chunked into small units of fixed length called snippets. Snippets are useful units for retrieval and display (see below). Other processing tools can be applied to snippets as well, when available, such as speaker identification or topic segmentation. The resulting text is then indexed along with the other documents, using Apache Lucene.

Text is extracted from a large variety of document formats (including MS Office, PDF, and HTML), and hierarchies of directories are automatically scanned. In some scenarios, the document repository is prepared beforehand, by indicating to the system its root directory, while in others, users can add directories or individual files at will.

The ACLD can also use an external search engine operating on an external repository. In our demonstration, we connect the ACLD to the Google Web search API, restricted or not to a sub-domain, but we have also successfully applied our approach to the Google Desktop application for searching local disks.

4.2 Querying the Document Database

The retrieval of linked content is mainly based on the queries derived from the words that are uttered. The ACLD uses a real-time automatic speech recognition (ASR) system for English [3], with a word error rate that is small enough to make it applicable to our purposes (around 38% WER for a real-time factor of 1.0 on the AMI Corpus). One of the main features of the ASR is the trade-off between speed and accuracy, which allows it to adapt the processing load so as to run in real-time (with a slight delay only) even on lower performance computers. The ASR can be coupled to a microphone array to improve recognition of conversational speech. Of course, when the ASR is used to process the recordings of past meetings, it can run slower than real time to maximize accuracy of recognition (typically in 4-5 times real time, with ca. 24% WER).

The Query Aggregator (QA) gathers the words recognized by the ASR in the most recent time frame of the conversation – typically 10-30 seconds, but also on demand from the user – to construct the queries, by putting them together. Stopwords are filtered out – currently using a list of about 80 words – so that only content words are used for search.

As knowledge about the important terminology of a domain can increase the impact of specific words on search, a list of pre-specified keywords for a given project can be defined, and can also be modified afterwards while running the ACLD. For instance, for remote control design as in the AMI Corpus scenario, the keyword list includes about 30 words such as 'chip', 'button', or 'material' (regardless of singular or plural forms).

Each query is processed by Apache Lucene to search for meeting snippets and documents stored locally, or by the Google Web API to search for web sites. If any of the predefined keywords are detected in the ASR of the current conversation, then their importance is increased when doing the search by boosting them in the query to Lucene at five times the weight of regular words. For the Google queries, all other words are removed if keywords are found, because differential keyword boosting is not possible. If no keyword at all is detected during a given time frame, then all the recognized words (minus the stopwords) are used to construct the query.

The QA applies a salience-based *persistence model* to integrate results obtained for the current time frame with previous results, in order to avoid large variations from one time frame to another, due to the fact that word choice varies considerably in such small speech samples, and therefore search results vary as well. The QA estimates the salience of each document as follows. A past document not retrieved in subsequent queries sees its salience decrease in time, unless the document is retrieved again, in which case its past salience is added to the salience due to its present retrieval: $s(t_n) = \alpha * s(t_{n-1}) + s_{\text{result}}$, where α is the persistence factor ($0 \leq \alpha \leq 1$) and s_{result} is the salience of the document in the result set from the current query (possibly zero if not found). The persistence factor can be tuned depending on the curiosity of the user, knowing also that all past results are saved in the user interface so that users can return back to them any time.

The QA returns a list of URIs for documents and webpages, with relevance scores from Lucene in the case of documents, and with rankings for Google results, all accompanied by excerpts of the documents that include the words from the query that were found in the document and their immediate context.

4.3 User Interface

The main goal of the User Interface (UI) at this stage of the ACLD – still a research prototype, rather than a commercial product – is to make available all the information produced by the system in a configurable way, showing more or less information according to hypothesized user's needs. Several instances of the UI can be coupled to one instance of the ACLD, so that each user in a meeting, for instance, has their own UI displayed on their laptop. The UI has a flexible graphical layout, maximizing the accessibility but also the understandability of the results, and displaying intermediate data as well, namely recognized words and keywords. The UI can display up to five widgets, which can be enabled, disabled, and arranged at will:

1. ASR results with highlighted keywords.
2. Tag-cloud of keywords, coding for recency (bold/gray) and overall frequency (font size).
3. Names of documents and snippets found by the QA.
4. Names of web pages found by Google.
5. Names of files found by Google Desktop.

Two main modes were conceived, but other arrangements are possible too. The modes are: an informative *full-screen UI* (see Figure 1 with widgets 1–4) displaying widgets side by side; and an *unobtrusive UI* displaying only one widget at the time, the others being accessible as superposed tabs. The unobtrusive mode can be chosen, in particular, if users are bothered by suggestions during a phase of the discussion in which they do not wish to be interrupted, or if they feel that the suggestions are not relevant enough to be examined.

The document names displayed in widgets 3–5 represent links to the respective documents, which can be opened using their native application (e.g. MS Word for a .DOC document). For a meeting snippet, a meeting browser such as JFerret [11] is launched to provide access to synchronized audio-visual and slide recordings.

When hovering over document names, a pop-up window displays metadata about the document (such as the title and the creation date) along with the match context, i.e. the fragments containing the keywords or words of the query. This enables users to quickly understand why a certain document was retrieved, and to get an idea about its contents without necessarily opening it.

5. EVALUATION EXPERIMENTS

Full evaluation-in-use experiments are still to come, as they depend on the selection of a specific scenario of use, in a situated context. In the meanwhile, two types of evidence show the utility of the ACLD system: pilot experiments in a task-based scenario, and usability evaluation of the UI. Moreover, positive feedback was received from corporate viewers of our demonstration.

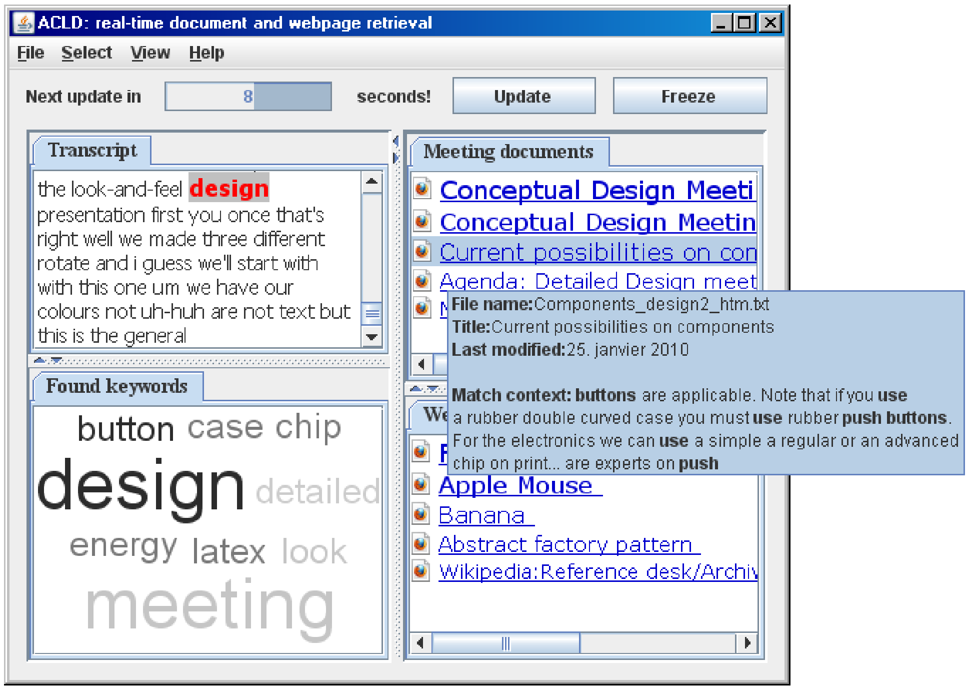
A pilot experiment was conducted with the unobtrusive version of the UI, following a task-based scenario in which four subjects had to complete the design of a remote control that was started in a series of past meetings (ES2008a-b-c from the AMI Corpus). The goal was to compare two conditions, namely *with* vs. *without* the use of the ACLD, in terms of satisfied constraints, overall efficiency and satisfaction. Two pilot runs have shown that the ACLD was consulted about five times per meeting, which is in the range of the expected utility of the system (given that the query-free results come at little or no cost), but which also implies that a large number of trials are required in order to improve the statistical significance of differences between conditions. Therefore, this experiment was not continued, but a number of informal design observations were made.

The UI was submitted to usability evaluation with ten subjects rating it on a simple usability scale. The subjects could use the ACLD over one recorded meeting to complete several operations using the UI, such as adding a keyword. The overall usability score was close to 70%, which is considered as "acceptable" by the creators of the scale. Feedback was again recorded, consisting mainly of positive comments, but also of suggestions for a simplification of the UI.

In the course of its development, the ACLD was demonstrated to about 50 potential users: industrial partners, focus groups, review panels, and so on. In one series of 30-minute sessions, each demo started with a presentation of the ACLD and continued with a discussion, during which notes were taken. The overall concept was found very useful, with positive verbal evaluation. Feedback for short and long-term changes was collected (e.g. on the importance of displaying match context, linking on demand, or offering an unobtrusive mode), thus helping to validate and improve the ACLD demonstrated.

6. CONCLUSION AND FUTURE WORK

The ACLD is, to the best of our knowledge, the first just-in-time retrieval system to use spontaneous speech and to support access to relevant multimedia documents and web pages, in a highly configurable manner. Future work aims at improving the relevance of linked content by using an innovative approach to speech/document matching using semantic distance, and by modeling the conversational context in order to ensure appropriate timing of results. On the applicative side, the generic ACLD will be applied to specific use cases. An experiment with group work in a learning en-


ACLD: real-time document and webpage retrieval
File Select View Help
Next update in 8 seconds!
Update
Freeze
Transcript
Found keywords
button case chip design detailed energy latex look meeting
Meeting documents
Conceptual Design Meeti
Conceptual Design Meetin
Current possibilities on con
Agenda: Detailed Design meet
We
Apple Mouse
Banana
Abstract factory pattern
Wikipedia:Reference desk/Archiv

the look-and-feel **design** presentation first you once that's right well we made three different rotate and i guess we'll start with with this one um we have our colours not uh-huh are not text but this is the general

File name: Components_design2_htm.txt
Title: Current possibilities on components
Last modified:25. janvier 2010

Match context: buttons are applicable. Note that if you **use** a rubber double curved case you must **use** rubber **push buttons**. For the electronics we can **use** a simple a regular or an advanced chip on print... are experts on **push**

Figure 1: Full screen mode of the UI with four widgets (counter-clockwise from top left): ASR output, keywords, websites, and document results (i.e. related documents and snippets of past meetings). Hovering over the third document displays metadata and match context.

vironment is planned, which will offer more insights into the evaluation-in-use of the ACLD.

Acknowledgments

The ACLD was supported by the EU AMIDA Integrated Project FP6-0033812, and by the Swiss IM2 NCCR. The authors would like to thank their colleagues from the ACLD development team: Erik Boertjes, Sandro Castronovo, Michal Fapso, Mike Flynn, Theresa Wilson, and Joost de Wit, as well as Phil Garner, Danil Korchagin and Mike Lincoln for help with real time ASR.

7. REFERENCES

[1] J. Budzik and K. J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *IUI 2000 (5th Intl. Conference on Intelligent User Interfaces)*, New Orleans, 2000.

[2] J. Carletta. Unleashing the killer corpus: experiences in creating the multieverything AMI Meeting Corpus. *Language Resources and Evaluation*, 41(2):181–190, 2007.

[3] P. N. Garner, J. Dines, T. Hain, A. El Hannani, M. Karafiat, D. Korchagin, M. Lincoln, V. Wan, and L. Zhang. Real-time ASR from meetings. In *Interspeech 2009 (10th Annual Conference of the International Speech Communication Association)*, pages 2119–2122, Brighton, UK, 2009.

[4] M. Goto, K. Kitayama, K. Itou, and T. Kobayashi. Speech Spotter: On-demand speech recognition in human-human conversation on the telephone or in face-to-face situations. In *ICSLP 2004 (8th International Conference on Spoken Language Processing)*, pages 1533–1536, Jeju Island, 2004.

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

[6] M. Henziker, B.-W. Chang, B. Milch, and S. Brin. Query-free news search. *World Wide Web: Internet and Web Information Systems*, 8:101–126, 2005.

[7] K. Lyons, C. Skeels, T. Starner, C. M. Snoeck, B. A. Wong, and D. Ashbrook. Augmenting conversations using dual-purpose speech. In *UIST 2004 (17th Annual ACM Symposium on User Interface Software and Technology)*, pages 237–246, Santa Fe, NM, 2004.

[8] F. Metze and al. The 'Fame' interactive space. In *Machine Learning for Multimodal Interaction II*, LNCS 3869, pages 126–137. Springer, Berlin, 2006.

[9] A. Popescu-Belis, E. Boertjes, J. Kilgour, P. Poller, S. Castronovo, T. Wilson, A. Jaimes, and J. Carletta. The AMIDA Automatic Content Linking Device: Just-in-time document retrieval in meetings. In *Machine Learning for Multimodal Interaction V*, LNCS 5237, pages 272–283. Springer, Berlin, 2008.

[10] B. J. Rhodes and P. Maes. Just-in-time information retrieval agents. *IBM Systems Journal*, 39(3-4):685–704, 2000.

[11] P. Wellner, M. Flynn, and M. Guillemot. Browsing recorded meetings with Ferret. In *Machine Learning for Multimodal Interaction I*, LNCS 3361, pages 12–21. Springer, Berlin, 2004.