

Towards an automatic content linking device: online document retrieval and display during meetings

Andrei Popescu-Belis
Idiap Research Institute

IM2 Summer Institute
Riederalp, September 2, 2008

FNSNF

FONDS NATIONAL SUISSE
SCHWEIZERISCHER NATIONALFONDS
FONDO NAZIONALE SVIZZERO
SWISS NATIONAL SCIENCE FOUNDATION



Outline of the talk

- What is “Content Linking”?
- Components of the Automatic Content Linking Device (ACLD), focus on the Query Aggregator
- Architecture using the Hub
- Demo on meeting ES2008d
- Perspectives

Automatic Content Linking Device

- User requirements
 - participants in a meeting often mention documents containing facts that are currently discussed
 - but they do not have the time to search for the facts during the discussion flow

Content linking

- **what**: relate ongoing discussion to potentially relevant “documents” (in a large sense)
- **how**: perform real-time searches in a database of documents based on the words that are pronounced during a discussion

Automatic Content Linking Device

- Application scenarios
 - Just-in-time retrieval
 - meeting participants are given suggestions about relevant “documents”
 - they can ignore them or start consulting the documents
 - Document/speech alignment for meeting browsers
 - recordings of previous meetings are augmented with related documents

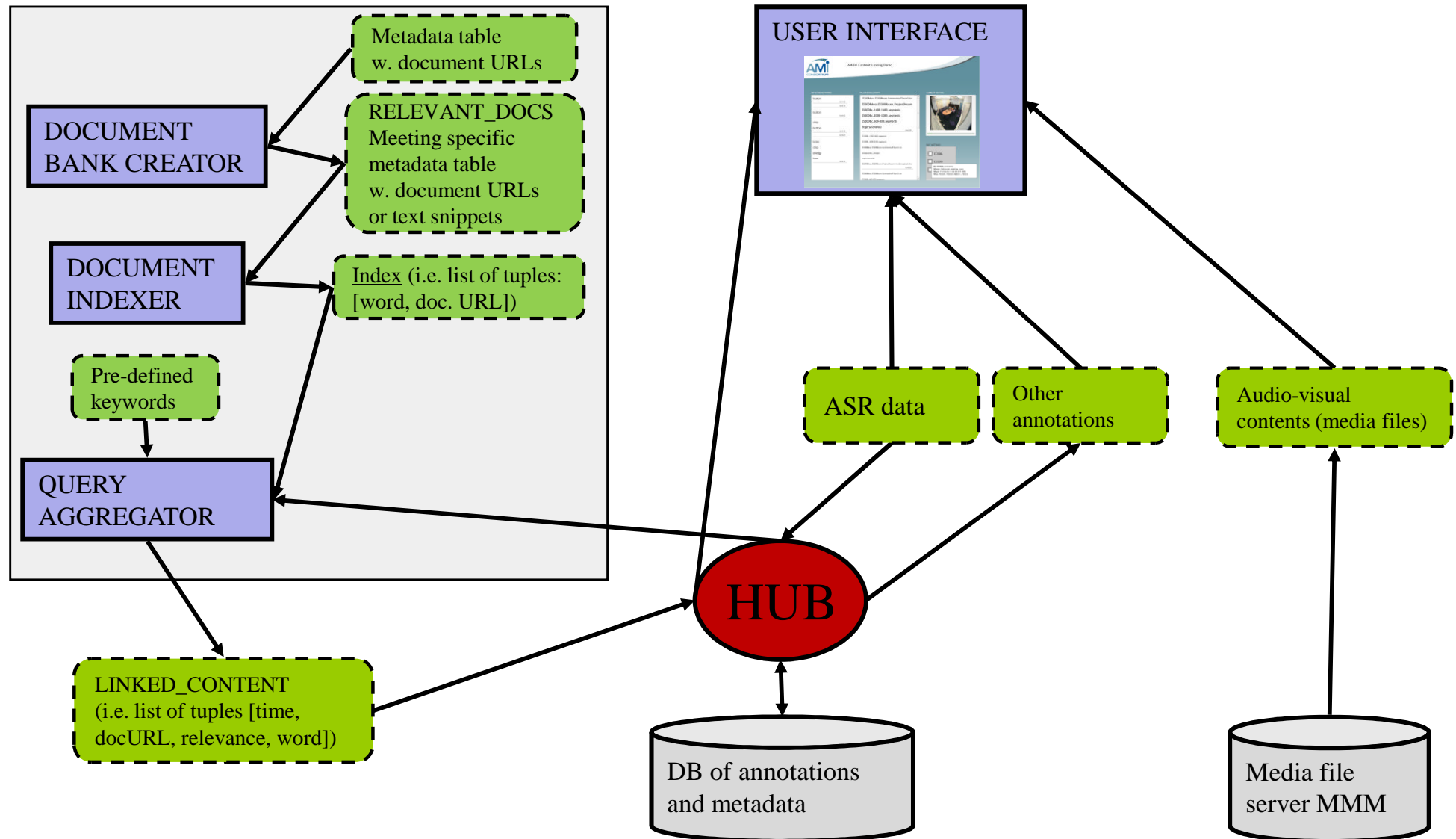
Components

- **Document Bank Creator**
 - gathers documents for a given series of meetings
 - documents = reports, emails, slides, minutes, etc.
 - pseudo-docs = fragments of previous meetings from a series
- **Document Indexer**
 - creates an index $\{(\text{word}_m, \text{doc}_k), \dots\}$ using Apache Lucene
- **Query Aggregator**
 - run searches using ASR, aggregate results with previous ones
- **“The Hub”**
 - subscription-based data exchange architecture
- **User Interface** (*from non-IM2 partner in AMI*)
 - display results, quick access to HTML and source of documents

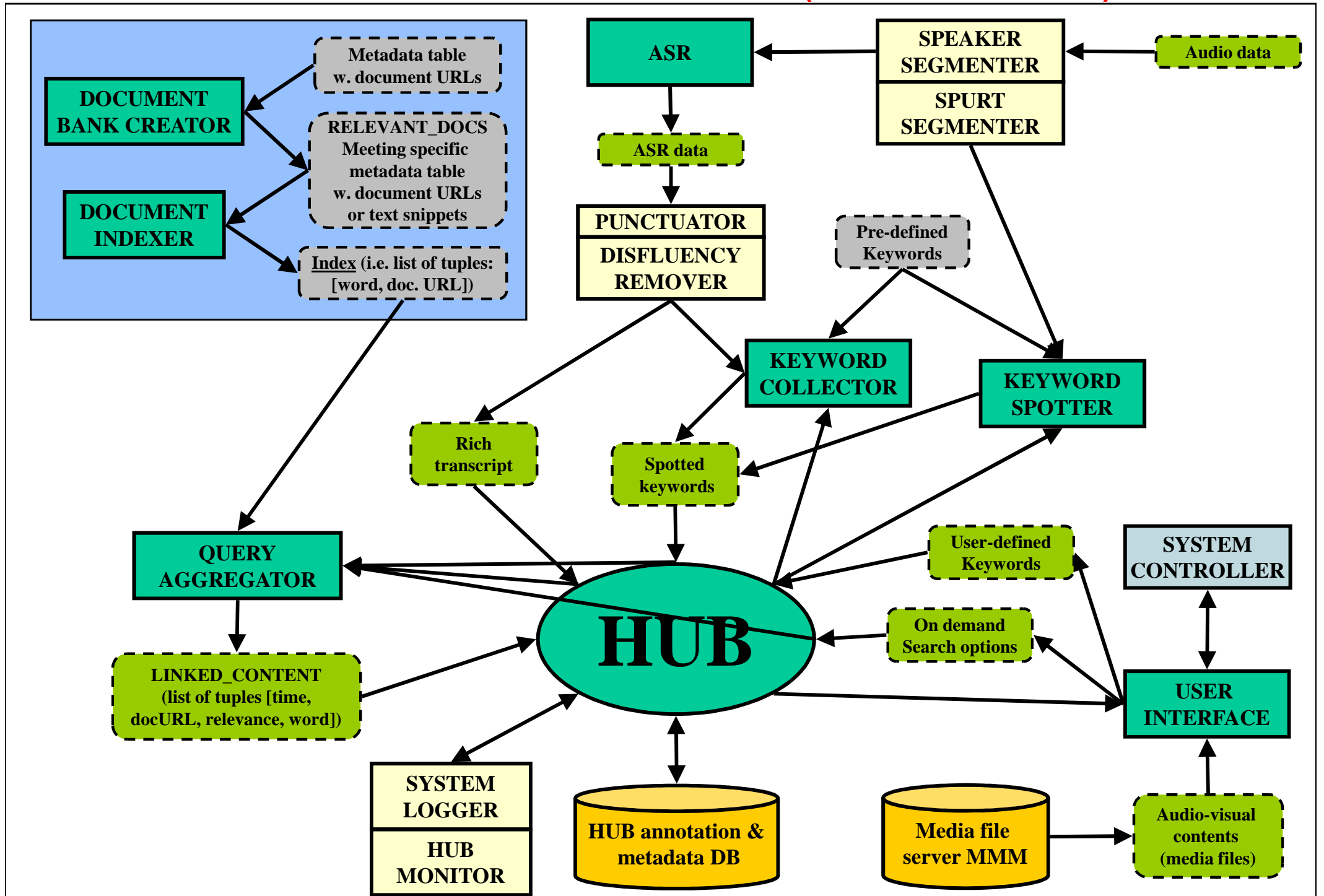
Query Aggregator

- **Query: list of search words**
 - recognized automatically from the discussion using real-time ASR (*ongoing*)
 - or recognized using a keyword spotting module
- **Keywords: optional but useful**
 - receive more important in the query
 - pre-defined list for a project or user
 - updatable during the meeting (*ongoing work UI*)
- **Results: every ~30 seconds**
 - query sent to document index using Apache Lucene
 - returns list of document names + relevance scores
- **Persistence model**
 - avoids variation in document list due to word variation in speech samples
 - adjusted relevance $R'(t_n) = \alpha \cdot R(t_n) + (1 - \alpha) \cdot R'(t_{n-1})$ (α is the persistence)
 - cutoff low-relevance documents

Current architecture of the ACLD



Architecture: version 2 (under work)



Demo

- Meeting ES1008d is the fourth meeting in a series of remote-control design meetings
 - meeting ES1008d is ongoing
 - ASR results are streamed via the Hub to the Query Aggregator
- Interface displays in real-time (refreshed)
 - keywords that were recognized in speech
 - most relevant documents (names), with font size codes for relevance
 - access to documents, summaries, etc.

Perspectives (1): evaluation

1. Construct **ground truth data** (to optimize or evaluate automatically the ACLD any time) through two experiments
 - A. Subjects associate to each meeting segment the relevant docs
 - **challenge**: demonstrate acceptable inter-coder agreement
 - **solution**: present only subsets of docs (more subjects needed)
 - B. Subjects watch part of a meeting and judge the relevance of each document returned by the ACLD
 - **limits**: does not measure silence, only noise

2. **Evaluation in use** on participants to a meeting
 - how often they consult the docs found by the ACLD + questionnaire
 - **challenges**: cost / non repeatable, difficult to generalize results
 - **alternative**: “focus group” study with demo only + questionnaire

Perspectives (2)

- Ongoing development work
 - Document repository
 - add websites
 - include documents from larger sets
 - use private vs. public
 - Query Aggregator
 - adjust 30-sec sampling with speech segments
 - search on demand
 - add/remove keywords to/from an initial list

Perspectives (3)

- Improve graphical layout of the user interface
 - keyword representation using tag clouds
 - relate clearly the documents to the recognized keywords
 - improve access to documents & overall user experience
- Later: redesign the interface using JFerret
- Reference

Popescu-Belis A., Boertjes E., Kilgour J., Poller P., Castronovo S., Wilson T., Jaimes A. & Carletta J. (2008) - "The AMIDA Automatic Content Linking Device: Just-in-Time Document Retrieval in Meetings". *Machine Learning for Multimodal Interaction V (Proc. of MLMI 2008, Utrecht, 8-10 Sep 2008)*, LNCS 5237, Springer, p.272-283.