Natural Language Queries on Natural Language Data: a Database of Meeting Dialogues

Susan Armstrong[1], Alexander Clark[1], Giovanni Coray[2], Maria Georgescul[1], Vincenzo Pallotta[2], Andrei Popescu-Belis[1], David Portabella[3], Martin Rajman[3], Marianne Starlander[1]

[1]ISSCO/TIM/ETI
University of Geneva
Boulevard du Pont-d'Arve, 40
CH-1211 Geneva 4 – Switzerland
{Susan.Armstrong, Alex.Clark, Andrei.Popescu-Belis}@issco.unige.ch
{Maria.Georgescul, Marianne.Starlander}@eti.unige.ch

[2]Computer Science Theory Laboratory (LITH)
Swiss Federal Institute of Technology (EPFL)
EPFL-DI-LITH, IN (Ecublens)
CH-1015 Lausanne – Switzerland
{Giovanni.Coray, Vincenzo.Pallotta}@epfl.ch

[3]Artificial Intelligence Laboratory (LIA)
Swiss Federal Institute of Technology (EPFL)
EPFL-DI-LIA, IN (Ecublens)
CH-1015 Lausanne – Switzerland
{David.Portabella, Martin.Rajman}@epfl.ch

Abstract: This paper describes an integrated system that enables the storage and retrieval of meeting transcripts (e.g. staff meetings). The system gives users who have not attended a meeting, or who want to review a particular point, enhanced access to an annotated version of the recorded data. This paper describes the various stages in the processing, storage and query of the data. First, we put forward the idea of shallow dialogue processing, in order to extract significant features of the meeting transcriptions for storage in a database, whose structure is briefly outlined. Low-level access to the database is provided as a Web service, which can be connected to several interfaces. A description of how multimodal input can be used with VoiceXML is also provided, thus offering an easy solution for voice and web based access to the dialogue data. The paper ends with considerations about the available data and its use in the current version of the system.

Keywords: dialogue processing, structured dialogue database, natural language queries, multimodal queries, meeting records, speech retrieval.

1 Introduction

Interaction through meetings is among the richest human communication activities. Recording meetings implies the storage and the structuring of a large set of heterogeneous information scattered over time and media. The raw data format from the various recording devices is not directly usable neither for the creation of indexes, nor for content-based access to the relevant parts of the meeting recording.

Annotating a meeting recording entails the production of the content's meta-descriptions. Assuming as a starting point a collection of time-stamped audio-video sequences, the transcription of the dialogues is of primary importance. The transcription can be done manually by transcribers or automatically by a speech recognition system. The quality of the transcription, in terms of word-error rates, is a crucial issue on which all of the subsequent processing depends. It is clear that a manual transcription provides us with an extremely reliable source of information, whereas a speech recognition engine will not perform at its best in a noisy situation such as a meeting is likely to be. Furthermore, several additional layers of annotation are necessary in order to capture the relevant features used to answer a given user query. Depending on the type of the query, we might need to extract features at different levels of abstraction and granularity, a point that will be examined further.

The *application scenario* we envisage for meeting recording, understanding, storage and retrieval is the following: suppose someone has not attended a group meeting, but needs information about "what happened" at the meeting. In this situation the user might like to ask queries about the meeting participants, about the issues that were discussed and the decisions that were made. Answers to these queries can be of different types. A list of participants can be an answer to the former question, whereas a written summary or an excerpt of the crucial audio-video recorded sequence can be the answer to the latter. The user might also be interested in accessing the documents related to the meeting, such as the agenda, reports, presentation handouts, etc.

In this paper we are mainly concerned with the higher-level annotations of meeting dialogues based on existing transcriptions. We assume for the time being the existence of a transcription, since we aim at designing tools for the extraction of semantic content from the linguistic surface expression. However, we are aware of the fact that additional information arises from several communication modalities, which we need to integrate into the analysis tools we are currently designing. The overall goal of the project is to achieve a multimodal processing architecture where the linguistic dialogue analysis is one of the components.

How to retrieve the relevant information from the meeting recordings is itself an important issue in the design of the overall architecture. On the one hand, and depending on the type of meetings, we need to specify the user requirements in terms of the relevant information that must be extracted from the transcription. On the other hand, the users might not be immediately aware of how the meeting recordings are structured, and they may be unable to formulate a direct query to the retrieval system. This last issue excludes the simple solution based on a search engine and forces us to consider a dialogue interface for query formulation and refinement. In this paper we propose a web based architecture for multimodal access to the meeting recordings. This allows us to structure the navigation through the meeting content as a dialogue and to combine several dialogue modalities, including speech.

This paper is organized in the following order, from meeting recording to meeting retrieval. Section 2 describes the data capture devices and the input data for our research. Sections 3 and 4 explain how the meeting recordings are processed, i.e. which information about the discourse/dialogue structure can be reliably extracted. The storage of the processed data in a database is described in section 5, while section 6 proposes a robust architecture for natural language and multi-modal query of the meetings database.

2 Source data and initial formatting

The source data of the application is the meeting recording itself, which can be produced using a variety of settings. Within the (IM)2 project[1], which our application is a part of, two meeting recording rooms have been set up. Each of them has *individual microphones* for each participant, *far-field microphones*, as well as *individual cameras* and *overview cameras*, including cameras or intelligent boards for meeting material (slides, documents, drawings).

Other partners of the (IM)2 project are working on speech recognition and facial/gesture recognition. Therefore, for our research, we consider that *separate transcriptions of each speaker's words* are available. To account for the other modalities, we will model the data using a simple {($event_i$, $timestamp_j$), …} file. Events in other modalities (such as pointing, expression of emotions, etc.) are of potential use for dialog understanding, and will be considered in a further stage of the project.

[1] Cf. the website for the Interactive Multimodal Information Management project: *http://www.im2.ch*

The minimal formatting we hypothesize is, for each speaker/channel, a sequence of words (including exclamations, corrections, false starts, etc.) with corresponding timestamps. This mimics the output of a perfectly accurate Automatic Speech Recognition (ASR) device; manual transcription is used for development and testing of our tools. The processing formats and their input in the database are described below. However, in order to display and analyze the input data, we use a tabular representation, with one column per speaker, and one line per second. This and other representations are obtained automatically from the initial XML data using XSLT stylesheets and formatting scripts.

At the time of writing, a 'Media File Server' is being set up to host the (voluminous) recording data. Though meeting recording has already begun, the availability of *transcribed* data is limited. For the work reported here, we use English and French data from other sources. The English data was kindly provided by the ICSI Meeting Recorder project[2], and the French data was taken from a television program recording, that has been transcribed and annotated at ISSCO. The resulting data will finally be stored in an (IM)2 database through the Media File Server.

3 Shallow dialogue processing

In order to generate and store an enhanced version of the meeting recording enriched with additional layers of annotation, one must analyze or process the input dialog with Natural Language Processing (NLP) tools.

However, the current state of the art in NLP does not permit a completely automatic analysis of spontaneous unrestricted dialogs, especially concerning the semantic content of the utterances, the dialog structure, and the beliefs-desires-intentions of the participants. This is not only the case for transcriptions provided by an ASR system (which, given the acoustic environment, will have a word error rate of over 25%) but even for manually transcribed speech. Nonetheless, as the NLP methods are more accurate on the better quality manual transcriptions, these will be used in the initial phases of our application.

We use a "shallow" approach, which avoids much of the computational complexity and ambiguity that is problematic in NLP. By "shallow" we mean an approach that uses only a fixed number of levels of description, and avoids the recursive representations characteristic of traditional linguistic analyses. This has two significant advantages in this situation: first, the representation of the linguistic descriptions and thus the associated queries are greatly simplified; secondly, it is computationally feasible with state-of-the-art statistical NLP techniques to automatically derive these structures with reasonable accuracy, even when dealing with ASR output (in this case the hypotheses are delivered in the form of a word lattice).

---

[2] Cf. *http://www.icsi.berkeley.edu/Speech/mr/mtgrcdr.html*

The shallow dialogue processing relies on fairly standard initial 'low-level' processing such as part-of-speech tagging [van Halteren 1999] and shallow parsing [Li and Roth 2001] (identification of non-recursive syntactic constituents), which includes preliminary steps of named entity recognition.

The shallow dialogue processing itself aims at the following interrelated subtasks (see also Figure 1):

1. dialogue segmentation into individual utterances [Stolcke and Shriberg 1996] and into episodes that focus on a coherent topic [Choi 2000];

2. dialogue tagging with discourse acts and adjacency pairs detection [Stolcke, Ries et al. 2000] – each utterance is assigned a tag that identifies its discourse function, then pairs of utterances such as question/answer, invitation/accept, offer/decline are linked;

3.a. named entity detection, such as defined in the Entity Detection and Tracking campaign [Automatic Content Extraction 2000], i.e. detection of organizations, persons, facilities or buildings, geographical-social-political entities, and locations;

3.b. coreference resolution between entities, as in the MUC-6,7 campaigns, discussed also in [van Deemter and Kibble 2000]. Robust approaches exist also for this task, based on machine learning [Soon, Ng et al. 2001].

The integration of the various modules dealing with these phenomena can be performed either sequentially, in a pipeline, or using a more sophisticated blackboard architecture. Indeed, these phenomena are not completely independent features of discourse, but demonstrate close dependencies (e.g. "coreference relations are sparser *between* episodes than *within* episodes"). Therefore, a robust organization of the three modules could be defined as follows, using an XML-based annotation model for the data: (1) execute each module once; (2) loop through the modules 1-2-3, with the following condition: if something has changed in the annotation since the previous execution of a module, then execute the module again, but a module *cannot delete* annotation, it *can only add* some; (3) stop when no module is able to add any further annotation.

These tasks, considered separately, are in general well understood and have been extensively studied in the context of traditional corpora or text databases, though there are active research areas associated with them. The modules that perform these tasks in our application, mostly based on existing software developed by the participants, must still be integrated in the overall application. However, the particular data we are modeling here present unusual problems: for instance, multi-party discourse leads to particular difficulties in relation to turn-taking, particularly with respect to meetings. This area has not been studied computationally in any depth.

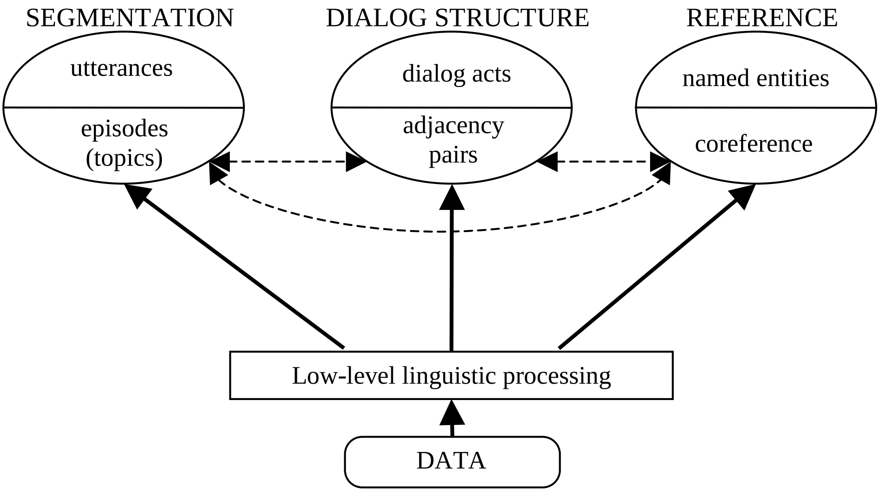


Figure 1. Overview of the shallow dialogue processing modules.

4 Towards deeper understanding of dialogue phenomena

In addition to shallow processing, which will be mostly automated, we also study the use of a deeper representation produced manually, that will offer the user more possibilities to identify the desired segments of the meeting. Depending on the type of meeting, it is useful to consider a taxonomy of dialogue situations such as that of [Walton and Krabbe 1995] shown in Table 1.

Type of Dialogue	**Initial Situation**	**ParticipantGoal**	**Goal of Dialogue**
Persuasion	Conflict of opinions	Persuade other party	Resolve or clarify issue
Inquiry	Need to have proof	Find and verify evidence	Prove (disprove) hypothesis
Negotiation	Conflict of interests	Get what you most want	Reasonable settlement
Information seeking	One party lacks information	Acquire or give information	Exchange information
Deliberation	Dilemma or practical choice	Co-ordinate goals or action	Decide best course of action
Eristic	Personal conflict	Verbally hit out at opponent	Reveal deeper basis of conflict

Table 1. Classification of dialogue situations [Walton and Krabbe 1995].

This classification is particularly useful to formally define the protocols of meetings where a decision-making process is taking place. The main rationale of having a formal description of such decision processes is to provide answers to complex questions from the user. These questions are mainly those pertaining to the outcome of the discussion in terms of the arguments invoked, questions raised, and consensus achieved on the discussed issues, for instance:

1. Let me see the moment when _XX_ and _YY_ strongly disagree.
2. How did we arrive at the conclusion that _XX_?
3. What were the objection to the proposal _ZZ_?
4. What was the position of _XX_ on subject _ZZ_?
5. What open questions remained?
6. Which criteria were adopted to take the decision _D1_?
7. Which arguments did the members who disagreed on the decision _D1_invoke?
8. Give me for each topic the list of people that said something related to it. Then give me their main point (agreement / disagreement with _XX_).
9. For each topic, give me the conclusion or the decision that was made.

In order to answer these types of questions we need to further annotate meeting recordings with appropriate meta-descriptions. In other words, we need to annotate the parts of the meeting where the decision process takes place with a suitable "argumentative structure". This information will serve as an additional index for retrieving the relevant information about the discussion outcomes.

A simple model of argumentative structure is the Issue Based Information Systems (IBIS), proposed by [Kunz and Rittel 1970] and adopted as a foundational theory in some *computer-supported collaborative argumentation* (CSCA) systems such as Zeno [Gordon and Karacapilidis 1999], HERMES [Karacapilidis and Papadias 2001], Questmap [Conklin, Selvin et al. 2001], and Compendium [Selvin 2001]. The main goal of IBIS is to arrive at a rational agreement on topic issues by following a pre-defined argumentation protocol. We adopt this model as a starting point for the description of the argumentative structure of decision meetings. The model captures and highlights the essential lines of a discussion in terms of what issues have been discussed and what alternatives have been proposed and accepted by the participants. The discussion is thus represented by a graph where each node is linked to the related segments in the meeting recordings. In order to generate this structure automatically we need to extract argumentative acts from the transcription. Argumentative acts are interpreted as update actions to the current argumentative structure.

One main difference between CSCA and meeting recordings is that, in the former case, the argumentative structure is used to constrain the interaction whereas, in the latter, we aim at deriving the corresponding structure (if any) by observing a recorded interaction. It is apparent that the second problem is harder since one has to infer the causal relationships of dialogue events without knowing the participants' intentions and goals. Modeling the dynamics of argumentation also means dealing with multimodal knowledge about the dialogue events, since some argumentation acts can be stated in other modalities (e.g. agreement by silence or by applause, disagreement by laughing).

The automatic construction of argumentative structures from meetings is a long-term goal, while manual construction seems to be more reasonable in the short-term. Nevertheless, there are some important steps which can be performed automatically. In an adaptive annotation tool [Ballim, Fatemi et al., 2000], topic segmentation can be used to propose annotation of issues, and the presence of a high number of propose-accept or propose-reject adjacency pairs may signal a segment which contains an argumentation act. We suggest that not all the types of meetings will require an argumentation analysis and not every part of the meeting will contain arguments. Indeed, the automatic detection of candidate segments where an argumentation is likely to be found could be of great help to the human annotator.

Another important advantage of having an additional layer of annotation is the improvement of precision during retrieval. We believe that it will beneficial to the end user if the argumentative structure is included in the dialogue model. Both queries and results would be better formulated and presented if the system can rely on a well defined structure of meeting events.

5 Database: contents and access

The database (henceforth, DB) is the central element of our system, to be exploited by means of a multimodal interface that combines natural language with visual browsing: graphics, menus, forms or even other modalities. The DB stores the information obtained through dialogue analysis by the processing modules and gives access to data for various interfaces. In the following section, we will describe what is stored in the DB, how it is input in the DB and how it is accessed.

The database model is relational and is represented in Figure 2. As we can see, in order to pinpoint the shallow dialog processing aspects, the database contains information like:

- the transcriptions of each utterance and the corresponding dialog acts, i.e. what the speaker intends to communicate ("UTTERANCES" table)
- topic segmentation ("TOPICS" table)
- the named entities ("NAMED ENTITIES" table)
- information about the speaker ("PERSONS" table)

- timing information to correlate with other modalities of the recorded stream and also to correlate information contained in different tables from the database (e.g. the timestamps attributes from the “TOPICS” and “UTTERANCES” tables give the relations between utterance transcriptions and topic segmentation)

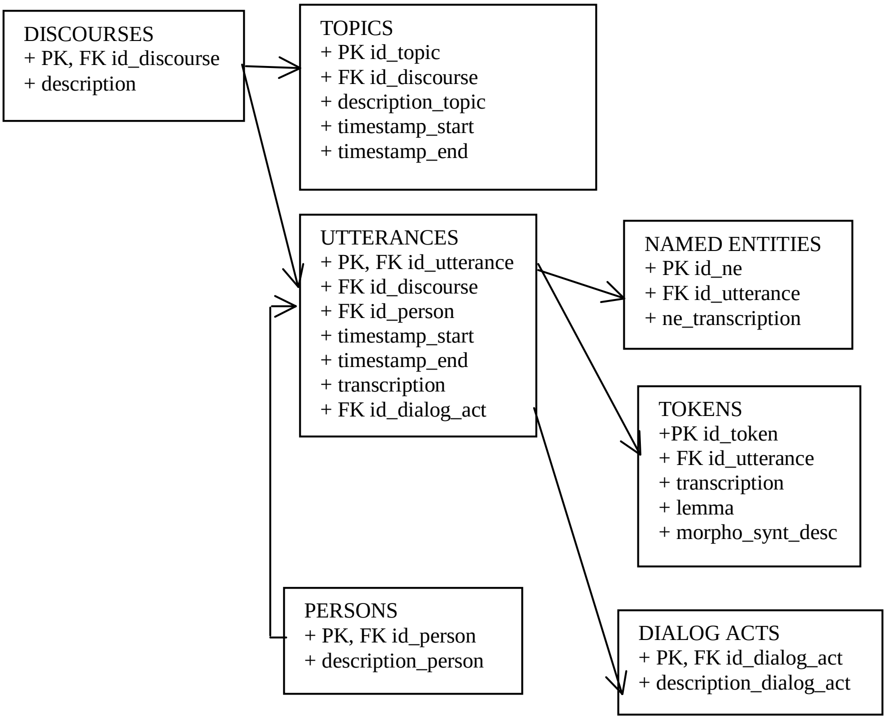
In addition to this information, we need detailed annotations at the token level and indicators of grammatical features (not elaborated in the schema). So the “TOKENS” table contains information like the orthographic form of every token as it appears in the corpus, grammatical annotations such as the base form, and morpho-syntactic information.

The initial shallow dialog annotations are stored in XML format and then in a relational database. Thus we can benefit from both the maturity of relational database engines and the productivity of XML technologies. To convert the initial annotations stored in XML format to relational database tables, we used standard XSL transformations.

For general applicability, the user queries should not rely on a specific schema or a too specific type of data modeling. In order to achieve this independence, we implemented a Web service access to the database that contains the shallow dialog processing analysis. The client application for the Web service produces a database-independent, canonical representation of a query entered into the multimodal interface. The query has been previously processed by the VoiceXML component (see Section 6). The Web service is dedicated to identify database elements that are necessary to satisfy the client queries (represented using an internal meaning representation). Also the Web service component generates a database query enabling the retrieval and aggregation of data from the database to satisfy the initial user query.

As the internal meaning representation format for the client-server communication, we currently use both SOAP (Simple Object Access Protocol) and MRML[3] (Multimedia Retrieval Markup Language). MRML has been successfully used for content-based image retrieval [Müller, Müller et al. 2000] and we use an extension of this protocol in the framework of natural language queries on natural language data. MRML is an XML-based protocol, so we can benefit from the XML advantages, including XML debugging advantages, i.e., the XML message can be easily understood by a developer who can verify that a query entered into the multimodal interface is properly interpreted to enable the correct retrieval and aggregation of the data from the database. Moreover, MRML offers the possibility to integrate a user “relevance feedback” mechanism in order to refine queries automatically. Given all the queries the user has submitted and the user’s ranking of query results, these are viewed as an interest profile for the user.

[3] Cf. *http://www.mrml.net/*



Legend:

→ represents one-to-many relation

PK represents primary key constraint

FK represents foreign key constraint

Figure 2. Overview of the database model.

6 Multimodal access using VoiceXML

Data annotation and storage of the multimodal data is not an end in itself, but rather the means for intelligent and efficient retrieval. Our concern here is to offer a natural user-interface by accepting a range of input modalities for the queries. Natural input implies natural language, in this case spoken input augmented with screen interaction by pointing. The latter modality is also useful for disambiguating queries (as discussed below).

Developing new services, as in the context described here, requires new multimodal input capabilities to facilitate the user interaction. VoiceXML can be extended to use some other input devices just like DTMF – in this case, a more general pointing device to identify parts of a screen[4]. In the context of a meeting, for instance, a pointer device could be used to ask the name of a given participant in a picture or video. In another context such as the SmartHome application (also under development) a pointer device could be used to select a device (e.g. a light), while verbally asking to switch it on. Like DTMF, the device sends some input to the interpreter. Filtered with the help of a grammar, this information can be parsed to produce a semantic result.

Pointing is a natural means of human communication and can also be used by the system to help disambiguate or contextualize the query. If the user says "Switch on this light", it may not be clear what "this" refers to, if numerous lights are available. Let us assume that the following grammar (in Java Speech Grammar Format) is used:

```
public <main> =
[<politeness><command>[<politeness>]<object><politeness>];

<command> = switch on | switch off {off};

<object> = entrance light {entrance}

     | bedroom light {bedroom}

     | dining room light {dining_room}

     | this light {this}

<politeness> = please;
```

In this case, the interpreter needs to contextualize the word "this" in the command, and therefore can ask a passive position tracking device to which light the user is pointing to. In the case of the meeting retrieval application, the same kind of "contextualization" can be used to identify a document or a participant (for instance when a list of possibilities is displayed).

To implement this functionality we have designed an extended input component for the Elvira VoiceXML interpreter[5]. This component is based on Web services (see Figure 3). It is able to take the input from the speech recognizer, the keyboard and also from a set of other specified input devices. When there is an input from any of these, the data is just converted to some predefined string format and passed to the interpreter. The interpreter then uses the active grammars to parse the input and produce a semantic result.

---

[4] Voice XML was primarily designed for telephone applications, in particular speech input. It soon became apparent that another important input device was DTMF keypresses, which were subsequently incorporated into the specifications.

[5] The work on the Elvira interpreter is carried out in collaboration with the Speech and Dialogue group at the Faculty of Informatics, Masaryk University Brno, Czech Republic.

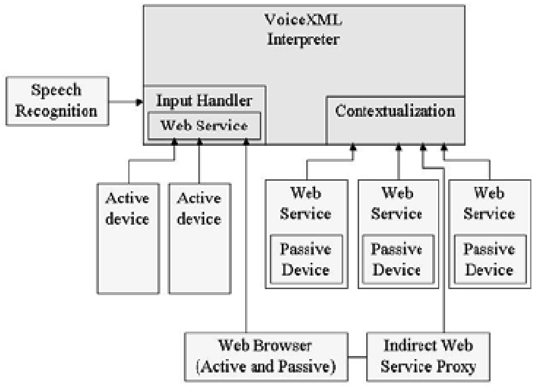


Figure 3. Overview of VoiceXML interface.

Note that this solution is adequate for "active" devices, i.e. devices that are proactively triggered by the user. The information extracted is semantically equivalent to a user query in the form of a sentence (typed or spoken). For the passive devices, i.e. devices that need to be called in order to provide information about their state (such as a position tracking device), additional functionality must be integrated in the dialogue manager. Again, a solution based on Web services has been implemented by extending this component to accommodate passive input devices.

As already mentioned, passive devices can be very useful to contextualize ambiguous or incomplete queries. When such a query is identified, the dialogue manager calls the relevant Web service asking for additional information and immediately receives the response, which can then be used for contextualization. In such a framework, a Web browser can be viewed as both an active device, calling the VoiceXML Web service and a passive one, implementing a Web service that the VoiceXML interpreter can call at any time.

For the (IM)2 project, this approach provides a framework for voice and Web based access to the dialogue data, and also a way to implement Wizard-of-Oz functionalities that are important for the prototyping of applications. Future work will collect data using these devices.

7 Conclusion

The application described in this paper combines the processing, storage and retrieval of natural language dialogues that occur in meetings. The need for robustness is pervasive throughout the application, due to the central role of natural language, which constitutes both the data and the means to access it. Moreover, since natural language dialogue appears as data and in queries, further research in this direction should study the possibility to design robust dialogue models applicable at the same time to human dialogue understanding and to human-computer dialogue management.

8 Acknowledgements

This work is part of a Swiss National Science Foundation project, (IM)2, i.e. Interactive Multimodal Information Management (*http://www.im2.ch*). More specifically, the present work belongs to the Multimodal Dialogue Management module (*http://www.issco.unige.ch/projects/im2/mdm*). The authors are grateful to the ICSI Meeting Recorder project for access to their data.

9 References

[Automatic Content Extraction 2000] Automatic Content Extraction. Entity Detection and Tracking (EDT): ACE Pilot Study Task Definition, National Institute of Standards and Technology, 2000.

[Ballim, Fatemi et al. 2000] Ballim, A.; N. Fatemi; Ghorbel, H.; Pallotta, V.: A knowledge-based approach to semi-automatic annotation of multimedia documents via user adaption. First EAGLES/ISLE Workshop on Meta-Descriptions and Annotation Schemas for Multimodal/Multimedia Language Resources (LREC 2000 Pre-Conference Workshop), Athens, Greece, 2000.

[Choi 2000] Choi, F.: Advances in domain independent linear text segmentation. The North American Chapter of the Association for Computational Linguistics (NAACL), Seattle, USA, 2000.

[Conklin, Selvin et al. 2001] Conklin, J.; Selvin, A.; Buckingham Shum, S.; Sierhuis, M.: Facilitated Hypertext for Collective Sensemaking: 15 years on from gIBIS, Knowledge Media Insitute, Open University, 2001.

[Gordon and Karacapilidis 1999] Gordon, T.; Karacapilidis N.: The Zeno Argumentation Framework. German Artificial Intelligence **3**: 20-29, 1999.

[Karacapilidis and Papadias 2001] Karacapilidis, N.; Papadias D.: Computer Supported Argumentation and Collaborative Decision Making: The HERMES system. Information Systems 26, no 4: 259-277, 1999.

[Kunz and Rittel 1970] Kunz, W.; Rittel H. W. J.: Issues as elements of information systems, Universität Stuttgart, Institut für Grundlagen der Planung. 1970.

[Li and Roth 2001] Li, X.; Roth D.: Exploring Evidence for Shallow Parsing. Proceedings of Workshop on Computational Natural Language Learning, 2001, Toulouse, France.

[Müller, Müller et al. 2000] Müller, W.; Müller H.; Marchand-Maillet, S.; Pun, T.; McG. Squire, D.; Pecenovic, Z.; Giess C.; Arjen P. de Vries : MRML: A Communication Protocol for Content-Based Image Retrieval. International Conference on Visual Information Systems, Lyon, France, 2000.

[Selvin 2001] Selvin, A.: Compendium: Making Meetings into Knowledge Events. Proceedings of Knowledge Technologies 2001 March 4-7, Austin TX, 2001.

[Soon, Ng et al. 2001] Soon, Wee Meng; Ng, Hwee Tou; Lim, Daniel Chung Yong: A Machine Learning Approach to Coreference Resolution of Noun Phrases. Computational Linguistics 27(4): 521-544, 2001.

[Stolcke, Ries et al. 2000] Stolcke, A.; K. Ries; Coccaro, N.; Bates, R.; Jurafsky, D.; Taylor, P.; Martin, R.; Meteer, M.; Van Ess-Dykema, C.: Dialogue Act Modeling for Automatic Tagging and Recognition of Conversational Speech. Computational Linguistics 26: 339-371, 2000

[Stolcke and Shriberg 1996] Stolcke, A.; Shriberg, E.: Automatic Linguistic Segmentation of Conversational Speech. The 4th International conference on Spoken Language Processing (ICSLP-96), Philadelphia, 1996.

[van Halteren 1999] van Halteren, H.: Syntactic Word-class tagging. Edited by Hamilton, Heidi Ehernberger. Kluwer Academic Publishers, 1999.

[van Deemter and Kibble 2000] van Deemter, K.; Kibble, R.: On Coreferring: Coreference in MUC and Related Annotation Schemes. Computational Linguistics 26(4): 629-637, 2000.

[Walton and Krabbe 1995] Walton, D. N.; Krabbe, E. C. W.: Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning. State University of New York Press, Albany, NY, 1995