REPORT IM2.MDM-07
Specification of a Shallow Dialogue Analysis Model: SDA

Andrei Popescu-Belis
ISSCO/TIM/ETI
University of Geneva

Version 1.0
July 2004

Abstract

This report defines a shallow dialog analysis model for use in the context of human dialogue processing applications. The present application concerns meeting recording, processing and retrieval. The model groups together the following elements: dialogue segmentation into utterances and episodes, detection of dialogue acts and adjacency pairs, and detection of referring expressions, including named entities, and their coreference links (in a first stage, references to documents only). Definitions, annotations guidelines, and links to tools are provided for these phenomena, as well as an instantiated XML annotation model based on boundaries, labels and links.

Chapter 1

Introduction

One of the current challenges to speech and language understanding is the situation where verbal communication is used interactively. Two main research domains deal with this issue: on one side, spoken dialogue systems (for interaction between people and computers) and on the other, human dialogue understanding systems. Research in the latter domain provides significant insights on the former, but it has also an interest of its own: proper understanding of human dialogues would enable many useful applications, among which we focus here on *automatic meeting processing*. This application would enable people who did not attend a meeting, or people who want to review a past meeting, to access in an efficient manner the essential information contained in the meeting.

The extent to which human dialogues need to be processed by computers depends strongly on the application that is foreseen—and of course on the actual technologies available. For the time being, applications must rely on shallow processing, i.e. the detection of surface elements. We propose here a *shallow dialog analysis* model (SDA) which is a result of a trade-off between robustness of extraction and relevance to dialogue understanding.

1.1 Organization of the report

This report proceeds as follows: the remaining of this chapter reviews our applicative goals, the input data and its pre-processing. In chapter 2 we define the model for shallow dialogue analysis (SDA) and principles of annotation. The components are described in the subsequent chapters:

1. segmentation into utterances—(chapter 3)

2. segmentation into topic-coherent episodes and detection of each episode's topic or keyword(s)—(chapter 4)

3. dialogue act and adjacency pair recognition—(chapter 5)

4. coreference information: recognition of referring expressions and their types, assignment of coreference links between them, with application to references to documents—(chapter 6)

Each chapter defines the respective phenomenon, provides guidelines for its annotation, describes the XML format, and suggests annotation tools and conversion procedures; each chapter ends with an annotation example based in general on the IDIAP Scripted Meeting corpus (in particular the TRN-11 dialogue).

The *SDA.XML format* functions as a pivot format, used for annotation within the IM2.MDM project. Conversion methods between the most important ones are proposed in another document, Report IM2.MDM-13. The conclusion of this report provides pointers to related work and examines possible extensions of the annotation model.

1.2 The MPR domain: meeting processing and retrieval applications

The application providing a framework to our research on dialogue comprises a database of meeting recordings and annotations, that must be made accessible to people searching for a particular piece of information connected to a meeting [1]. The retrieval system must make use of limited information extracted from the meeting recordings in order to respond to the user through a speech-based interface (a pilot study of user requirements is available as Report IM2.MDM-11). Therefore, our application must integrate a series of processing modules that pre-process recorded data and store it in a database[1].

The database is designed to store multimedia recordings of meetings: speech (mostly from individual microphones), video, meeting-related documents (including slides), etc. Focusing on the linguistic data, we suppose as a working hypothesis that separate transcriptions for every channel (i.e., every

[1]For more information about the (IM)2 project, "Interactive Multimodal Information Management", see `http://www.im2.ch`. For more information about the IM2.MDM module, "Multimodal Dialogue Management', see `http://www.issco.unige.ch/projects/im2/mdm`.

speaker) are available. While ASR systems would probably have about 25% WER in such an environment, manual annotation tools such as Transcriber[2] help produce accurate human transcriptions. The sequence of words in each channel must also be annotated with timestamps—a realistic assumption both for ASR and human transcription.

A significant corpus of meeting recordings (speech plus transcriptions, in English) is available from ICSI, Berkeley[3], and was kindly made available to us. The main example used in this report is a dialogue from the IDIAP Scripted Meeting Corpus[4], with a multimodal recording including voice (lapel microphone for each participant) and video (participants and projection screen). Other available sources are the University of Fribourg document-centric meetings [5], and a series of scenario-based meetings recorded at IDIAP by ISSCO members.

[2]Made available by LDC, USA and DGA, France at `http://www.etca.fr/CTA/gip/Projets/Transcriber/`.

[3]See `http://www.icsi.berkeley.edu/Speech/mr/`.

[4]See `http://mmm.idiap.ch`.

[5]See `http://diuf.unifr.ch/im2/`.

Chapter 2

SDA: a Shallow Dialogue Analysis Model

2.1 Overview of the SDA components

We propose in this section a model for shallow dialogue analysis (SDA) that enables robust extraction of significant information for spoken dialogues. We identified a set of promising elements from natural language processing, which we now proceed to describe. The detection of these elements requires of course variable amounts of pre-processing, illustrated in various articles and reports from IM2.MDM (see also [8]).

UT : boundaries of utterances, per channel;

DA : dialogue acts, that is, labels on UT according to a pre-defined tagset, which characterize the function of each utterance;

AP : adjacency pairs are links between UTs such as question-answer pairs; these are mostly inter-channel links, without any labels (on the links), for the moment—they are implicitly labelled by the DAs of the two utterances they connect;

EP : boundaries of thematic episodes, which signal a change of topic in the global structure of the dialogue, and are therefore expressed using the global time scale of the meeting, using a `starTime` and an `endTime`); conventionally, here, a new episode starts right at the end of the previous one, but one could also imagine some overlapping between episodes, or a hierarchical structure;

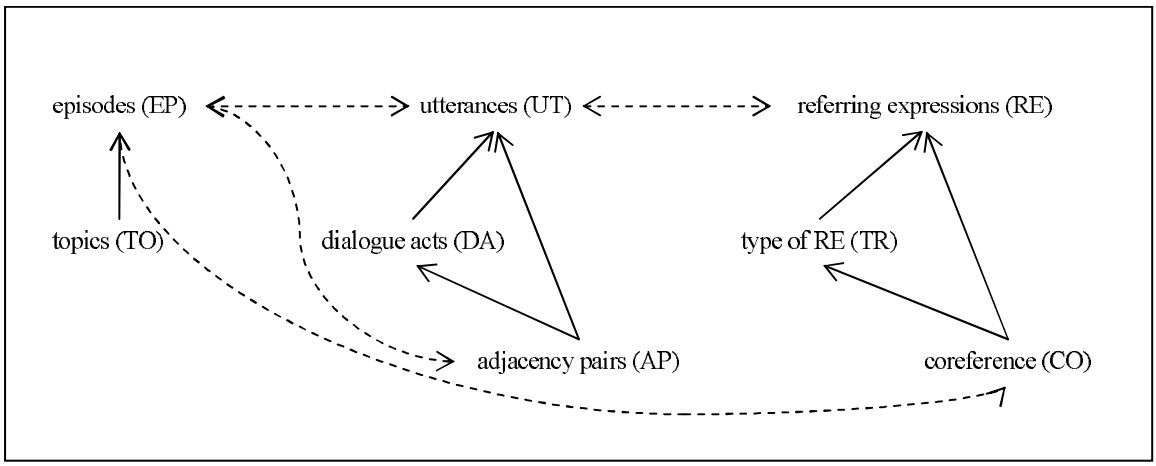

Figure 2.1: *Dependencies between components of SDA.*

TD : the topic description is a string label for each epiosde (EP); a more tractable solution is a set of keywords for each episode;

RE : referring expressions can be marked as phrases in each channel; for a start, we focus on references to documents, and markup all the REs referring to documents or document elements;

TR : the type of each RE could also be annotated, e.g. as in named-entity recognition tasks for proper names, dates, etc.; these closed-vocabulary labels on REs are not used for the moment;

CO : coreference information is a link between the REs that denote the same entity; for the moment, we wil focus on references to documents and assign to each RE the document and document element it refers to (here, using an XPath expression).

The various components of the SDA model were selected not only for their informativeness and the possibility to be detected reliably, but also because they are conceptually interrelated. Indeed, there is strong correlation between the boundaries and the links in an SDA structure, e.g., coreference relations are sparser *between* episodes than *within* episodes. These correlations are represented in Fig. 2.1, where full lines stand for strict dependency, and dashed lines for preferences.

We are at present in the process of assembling / converting / writing components for an SDA parser. The development cycle presupposes a precise

task definition, i.e. data annotation guidelines, as well as human annotated data to serve as reference for evaluation and subsequent training.

2.2 Data annotation by humans and computers

The shallow dialogue analysis model requires a concrete implementation in terms of annotations. As noted above, human annotated data is necessary to evaluate the SDA parser, and such data requires annotation conventions and tools. Also, comparison metrics between annotations are necessary for software evaluation, as well as to check inter-annotator agreement.

The phenomena discussed above give birth to three types of annotations: boundaries, labels on bounded segments, and links between bounded segments. The links could also be typed later using labels, but for the moment, these are unnecessary; indeed, adjacency pairs are implicitly typed by the DA labels of their constituents, and for the moment only references to documents are considered instead of coreference (hence a label on each RE). In addition to boundaries, labels and links, the multi-channel structure of input data must be considered. Intra-channel boundaries are straightforward to annotate (as XML elements), but those concerning all channels must make reference to the global time-scale. Labels are annotated as XML attributes, while links are annotated as separate XML elements, using the indexes of the bounded segments. Type of links could be annotated as attributes on these elements. These abstract considerations parallel the Annotation Graphs (AG) formalism[1], but since we plan to reuse existing interfaces that do not make use of AG, we did not adopt this formalism.

2.3 SDA annotations

The SDA phenomena entail the annotation elements summarized in Table 2.1 with their abbreviations. There are three types of boundaries: intra-channel ones for utterances (UT) and referring expressions (RE), and inter-channel ones for episodes (EP). There are three types of labels: dialog acts (DA), type of REs (RT), and topics (TO), the first two being defined by a closed vocabulary (fixed list of labels). There are two types of links: adjacency pairs (AP) and coreference (CO). Additional annotations include: links, possibly labelled, between episodes (for a more structured thematic structure); labels on AP (if needed); labels on CO (for non-identity coreference, such as *part-of* or *person-function* relations).

[1]See `http://agtk.sourceforge.net`.

We provide below the DTD (Document Type Definition) that defines the annotation and enables us to validate resources. Complete annotation of the SDA is a time consuming task, and for the time being only about 15′ of a dialogue were fully annotated, together with more extensive annotations of only some SDA components (see Report IM2.MDM-13). Conversion of UT, DA and AP annotation from available ICSI data (see Report IM2.MDM-09) also provides a resource which awaits EP+TO and RE+RT+AP annotation.

Code	Name	Type	Scope	Other features
UT	utterances	boundary	intra-channel	non-partitioning
DA	dialog acts	label	on UT	closed vocabulary
AP	adjacency pairs	link	between UT	mostly inter-channel, no labels
EP	episodes	boundary	global	partitioning
TO	topics	label	on EP	open vocabulary
RE	referring expressions	boundary	intra-channel	non-partitioning
RT	types of REs	label	on RE	closed vocabulary
CO	coreference links	link	between RE	intra- and inter-channel, no labels

Table 2.1: *Annotations for the various SDA elements.*

2.4 Annotation interfaces and conversion procedures

For simplicity reasons, we decided to reuse existing annotation interfaces, and to convert XML-exported data using XSLT stylesheets between interfaces (eXtensible Stylesheet Language)—see report IM2.MDM-13. The consideration of boundaries, labels and links indicated that two interfaces are sufficient: Transcriber (see note 2) is used for transcription (including revision) as well as for insertion of boundaries (UT, EP) and their labels (TO, DA). The annotation of referring information is made using MMAX[2], a parametrable interface specific to RE, RT and CO. MMAX can also be parametrated to annotate adjacency pairs AP, and these can also be annotated directly on the XML files.

Several data formats are used before the dialogs are fed into our final dialog database, depending on the transcription tool. For instance, ICSI data appears both in multi-channel Transcriber format and in TableTrans

[2] Available freely at `http://www.eml.villa-bosch.de/english/Research/NLP/Downloads`.

(AGTK) format. We prefer to use standard Transcriber format by transcribing and annotating channels separately (EP and TO must however be transformed into extra-channel annotation). MMAX uses other XML tags than Transcriber, therefore extensive use of XSLT is necessary. XSLT is also used to generate HTML tables for a user-friendly representation of the dialog information, and to generate the final tabular format that is fed into the database.

2.5 Example of raw data

This is a transcription of IDIAP's TRN-11[3] on one channel (nb. 2), i.e. one speaker, done here with Transcriber. Ideally, this is the input format that is used for SDA annotation, per channel. Note the following details:

- The XML file contains only the *transcription of the words* and precise timing information.
- The headers introduced by Transcriber are not really significant: they can easily be converted to simple XML elements that suit better our purpose, as will appear in later examples.
- Silences are marked with a special code used here by IDIAP: SKIP(OTHER) if another speaker has the turn, or GAP(OTHER) if this is a pause.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Trans SYSTEM "trans-13.dtd">
<Trans scribe="(unknown)"
       audio_filename="Lapel-2_T000010.0_T0450.369"
       version="4"
       version_date="030423">
<Episode>
<Section type="report" startTime="0" endTime="290.368">
<Turn startTime="0" endTime="290.368">
<Sync time="0"/>
SKIP(OTHER)
<Sync time="1.565"/>
uh
<Sync time="1.948"/>
so okay let me discuss about the my research plans for
<Sync time="5.934"/>
SKIP(OTHER)
```

[3] See http://mmm.idiap.ch/cgi-bin/SessionIndex?sessionDir=Scripted-Meeting-TRN-11.

```
<Sync time="6.9"/>
for the next few months
<Sync time="7.990"/>
SKIP(OTHER)
<Sync time="8.151"/>
what I'm really interested in is using computer vision and speech
<Sync time="11.637"/>
so the idea sounds funny but we could just use spectrogram
<Sync time="14.692"/>
SKIP(OTHER)
<Sync time="15.232"/>
Jitou just happened to tell me this idea
<Sync time="16.787"/>
and it sounds very [PF] impresive to me
<Sync time="19.123"/>
what we could do is just present spectrograms with varying length of windows
<Sync time="22.892"/>
and just try to pick up the
<Sync time="24.734"/>
SKIP(OTHER)
<Sync time="25.568"/>
trajectory and the contours of the high energy as they are changing with time
<Sync time="29.391"/>
[PF] so
<Sync time="30.206"/>
SKIP(OTHER)
<Sync time="31.72"/>
so
<Sync time="32.098"/>
SKIP(OTHER)
<Sync time="32.749"/>
do you think it's a good idea to-
<Sync time="34.539"/>
to to to to to
<Sync time="35.799"/>
to use computer vision for phoneme recognition
<Sync time="38.462"/>
GAP(OTHER)
<Sync time="46.23"/>
okay
<Sync time="46.56"/>
SKIP(OTHER)
<Sync time="47.125"/>
I think it's good yeah
<Sync time="48.063"/>
```

```
SKIP(OTHER)
<Sync time="48.531"/>
so second point that we should [AH] discuss is
<Sync time="51.63"/>
do you-
<Sync time="52.186"/>
do you think it's a good idea to measure the length of the windows all the time
<Sync time="56.648"/>
GAP(OTHER)
<Sync time="60.772"/>
yeah
<Sync time="61.216"/>
no
<Sync time="63.422"/>
it's complex
<Sync time="64.333"/>
SKIP(OTHER)
<Sync time="65.696"/>
so the third point is that
<Sync time="67.304"/>
do we agree with [AH] with using [AH] using power spectrum in this representation
<Sync time="72.727"/>
GAP(OTHER)
<Sync time="79.183"/>
okay I agree yeah
<Sync time="80.242"/>
I think it's fairly important yeah

[... DELETED ...]

<Sync time="257.15"/>
Jo do you think that would work
<Sync time="258.739"/>
GAP(OTHER)
<Sync time="271.726"/>
no I don't think anybody has done before but I mean
<Sync time="273.817"/>
SKIP(OTHER)
<Sync time="274.781"/>
all of you think that's potential-
<Sync time="276.93"/>
potentially good so I gonna try see
<Sync time="279.331"/>
GAP(OTHER)
<Sync time="283.235"/>
```

```
I mean I ?? in fact
<Sync time="285.923"/>
thanks
<Sync time="286.183"/>
GAP(OTHER)
<Sync time="286.782"/>
thanks
<Sync time="287.62"/>
GAP(OTHER)
</Turn>
</Section>
</Episode>
</Trans>
```

2.6 The SDA document type definition (DTD)

This is also the place to give the actual DTD, version 1.01, whic enriches 1.0 with annotations about references to documents. The main point is that transcription should be separated into channels, containing words, timing information, and segmentation into utterances, and referring expressions. The other information is annotated as standalone blocks (possibly files) at the end of the transcription. Strictly speaking, UT and RE could be annotated as external blocks, using an index on the number of words (as in MMAX).

```
<!ELEMENT dialog (channel*, episodes?, dialog-acts?, ref2doc?)>

<!ELEMENT channel (#PCDATA|Sync|utt)*  >
<!ATTLIST channel
startTime  CDATA #IMPLIED
endTime  CDATA #IMPLIED
id  CDATA #IMPLIED
speaker  CDATA #REQUIRED >

<!ELEMENT Sync EMPTY >
<!ATTLIST Sync time  CDATA #REQUIRED >

<!ELEMENT utt (#PCDATA|Sync|er)* >
<!ATTLIST utt id CDATA #IMPLIED
startTime CDATA #IMPLIED
endTime CDATA #IMPLIED >
```

```
<!ELEMENT er (#PCDATA) >
<!ATTLIST er id CDATA #IMPLIED   >

<!ELEMENT episodes (episode*) >
<!ELEMENT episode EMPTY >
<!ATTLIST episode id CDATA #IMPLIED
startTime CDATA #REQUIRED
endTime CDATA #REQUIRED
topic CDATA #IMPLIED
keywords CDATA #IMPLIED >

<!ELEMENT dialog-acts (dialog-act*) >
<!ELEMENT dialog-act EMPTY >
<!ATTLIST dialog-act utt-id CDATA #REQUIRED
label CDATA #REQUIRED
group CDATA #IMPLIED >

<!ELEMENT ref2doc (ref*) >
<!ELEMENT ref EMPTY >
<!ATTLIST ref er-id CDATA #REQUIRED
doc_file CDATA #REQUIRED
doc-id CDATA #REQUIRED >
```

More intuitively, a skeleton file is given below. It provides a fictitious example which is easier to understand than the DTD.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dialog SYSTEM "dialog-sda-101.dtd">

<dialog>

<channel startTime="0" endTime="290.368" speaker="Iain" id="0">
   <utt id="0_2" startTime="38.826" endTime="39.752">
     <Sync time="38.826"/>
     yeah
     <Sync time="39.062"/>
     ..
   </utt>
   [... DELETED ...]
</channel>

<channel startTime="0" endTime="290.368" speaker="Vivek" id="1">
  <utt id="1_2" startTime="1.565" endTime="1.948">
    <Sync time="1.565"/>
```

```
    uh
  </utt>
  <utt id="1_3" startTime="1.948" endTime="8.151">
     <Sync time="1.948"/>
     so okay let me discuss <er id="1">the first article</er>
     <Sync time="5.934"/>
     ..
     <Sync time="6.9"/>
     for the next few minutes
     <Sync time="7.990"/>
     ..
   </utt>
   [... DELETED ...]
</channel>

<channel ...DELETED... </channel>
<channel ...DELETED... </channel>
<channel ...DELETED... </channel>

<!-- ************************ -->

<episodes>
<episode startTime="0.0"
         endTime="28.0"
         topic="Vivek's monologue"
         keywords="phoneme recognition, computer vision"/>

<episode startTime="28.0"
         endTime="48.0"
         topic="Consensus"/>

<episode ...DELETED...  />
<episode ...DELETED...  />
<episode ...DELETED...  />

</episodes>

<!-- ************************ -->

<dialog-acts>

  <dialog-act utt-id="0_2"  label="sj^na"  group="positive_answer"/>
  <dialog-act ...DELETED... />
  <dialog-act ...DELETED... />
  <dialog-act utt-id="1_2"  label="fg"     group="attention"/>
```

```
  <dialog-act utt-id="1_4"  label="s"      group="statement"/>
  <dialog-act ...DELETED... />
  <dialog-act ...DELETED... />
  <dialog-act ...DELETED... />

</dialog-acts>

<!-- ************************ -->

<ref2doc>

  <ref er-id="1" doc-file="Journal-Name-Date.xml" doc-id="//MasterArticle[@ID='1']"/>
  ...

</ref2doc>

</dialog>
```

Chapter 3

Segmentation into utterances

3.1 Definition

Starting from timestamped transcribed speech, boundaries should be inserted at two key levels: individual utterances, and topic-coherent episodes (boundaries between words are already present in the transcription). An *utterance* is a coherent, contiguous series of words in a given channel, which serves a precise function in the dialogue (or sometimes more than one); or, to anticipate, carries a dialogue act [12]. An utterance can often be equated with a proposition or a sentence, but in spoken language utterances do not always correspond to well-formed or completed propositions.

Utterances are the building blocks of dialogue structure, the minimal units that are of interest in dialog retrieval. Their identification plays a significant role in the formatting of recorded dialogs, namely for capitalization and punctuation. Many cues have been used for this task, such as: lexical markers, (shallow) syntactic structure, and prosody [11].

3.2 Example: segmentation into utterances using Transcriber

Here is the transcription quoted in section 2.5 segmented this time into utterances. Note here that `Turns` were defined using Transcriber (using the *Ctrl-T* shortcut for instance), but they represent here in fact *utterances*. The timing information and the codes marking silence are of course preserved. With respect to the raw version, a `Turn` can group one or more timestamps (`Sync`).

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE Trans SYSTEM "trans-13.dtd">
<Trans scribe="(unknown)"
       audio_filename="mix"
       version="5"
       version_date="030519">
<Episode>
<Section type="report" startTime="0" endTime="290.368">
<Turn startTime="0" endTime="1.565">
<Sync time="0"/>
SKIP(OTHER)
</Turn>
<Turn startTime="1.565" endTime="1.948">
<Sync time="1.565"/>
uh
</Turn>
<Turn startTime="1.948" endTime="8.151">
<Sync time="1.948"/>
so okay let me discuss about the my research plans for
<Sync time="5.934"/>
SKIP(OTHER)
<Sync time="6.9"/>
for the next few months
<Sync time="7.990"/>
SKIP(OTHER)
</Turn>
<Turn startTime="8.151" endTime="11.637">
<Sync time="8.151"/>
what I'm really interested in is using computer vision and speech
</Turn>
<Turn startTime="11.637" endTime="15.232">
<Sync time="11.637"/>
so the idea sounds funny but we could just use spectrogram
<Sync time="14.692"/>
SKIP(OTHER)
</Turn>
<Turn startTime="15.232" endTime="16.787">
<Sync time="15.232"/>
Jitou just happened to tell me this idea
</Turn>
<Turn startTime="16.787" endTime="19.123">
<Sync time="16.787"/>
and it sounds very [PF] impresive to me
</Turn>
```

```
[... DELETED ...]

<Turn startTime="271.726" endTime="273.817">
<Sync time="271.726"/>
no I don't think anybody has done before but I mean
</Turn>
<Turn startTime="273.817" endTime="274.781">
<Sync time="273.817"/>
SKIP(OTHER)
</Turn>
<Turn startTime="274.781" endTime="277.673">
<Sync time="274.781"/>
all of you think that's potential-
<Sync time="276.93"/>
potentially good
</Turn>
<Turn startTime="277.673" endTime="279.331">
<Sync time="277.673"/>
so I gonna try see
</Turn>
<Turn startTime="279.331" endTime="283.235">
<Sync time="279.331"/>
GAP(OTHER)
</Turn>
<Turn startTime="283.235" endTime="285.923">
<Sync time="283.235"/>
I mean I ?? in fact
</Turn>
<Turn startTime="285.923" endTime="286.183">
<Sync time="285.923"/>
thanks
</Turn>
<Turn startTime="286.183" endTime="286.782">
<Sync time="286.183"/>
GAP(OTHER)
</Turn>
<Turn startTime="286.782" endTime="287.62">
<Sync time="286.782"/>
thanks
</Turn>
<Turn startTime="287.62" endTime="290.368">
<Sync time="287.62"/>
GAP(OTHER)
</Turn>
</Section>
```

```
</Episode>
</Trans>
```

3.3 Example: final SDA files segmented into utterances

We now exemplify the SDA annotation model, with specific tags (yet simple ones). This file puts together the four channels of TRN-11, simply by concatenating the four transcriptions of the four channels. This is adapted from mono-channel Transcriber format. This is the prelimnary SDA.XML format derived from "raw" input data. The time spans of the utterances are partially redundant. Silence is noted with two consective dots. All subsequent makup will be standalone (except for RE): it will be added at the end of the file, but of course still within the root `dialog` element.

Each channel was considerably shortened for presentation. This is the file: `TRN-11.sda.xml`.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dialog SYSTEM "dialog-sda-10.dtd">

<dialog>

<channel startTime="0" endTime="290.368" speaker="Iain" id="0">
      <utt id="0_2" startTime="38.826" endTime="39.752">
            <Sync time="38.826"/>
            yeah
            <Sync time="39.062"/>
            ..
      </utt>
      <utt id="0_3" startTime="39.752" endTime="40.035">
            <Sync time="39.752"/>
            yeah
      </utt>
      <utt id="0_4" startTime="40.035" endTime="41.024">
            <Sync time="40.035"/>
            I think that's nice
      </utt>

[... DELETED ...]

      <utt id="0_74" startTime="270.155" endTime="271.395">
            <Sync time="270.155"/>
            has anyone done that before
      </utt>
      <utt id="0_76" startTime="274.737" endTime="275.102">
            <Sync time="274.737"/>
            yeah
      </utt>
      <utt id="0_78" startTime="282.583" endTime="282.94">
            <Sync time="282.583"/>
            yeah
            </utt>
      <utt id="0_80" startTime="283.202" endTime="284.568">
            <Sync time="283.202"/>
            even if it doesn't work you know
      </utt>
      <utt id="0_82" startTime="286.025" endTime="286.406">
            <Sync time="286.025"/>
            okay
      </utt>
```

```
</channel>


<!-- ******************************************************** -->

<channel startTime="0" endTime="290.368" speaker="Vivek" id="1">

      <utt id="1_2" startTime="1.565" endTime="1.948">
            <Sync time="1.565"/>
            uh
      </utt>
      <utt id="1_3" startTime="1.948" endTime="8.151">
            <Sync time="1.948"/>
            so okay let me discuss about the my research plans for
            <Sync time="5.934"/>
            ..
            <Sync time="6.9"/>
            for the next few months
            <Sync time="7.990"/>
            ..
      </utt>
      <utt id="1_4" startTime="8.151" endTime="11.637">
            <Sync time="8.151"/>
            what I'm really interested in
            is using computer vision and speech
      </utt>

[... DELETED ...]

      <utt id="1_65" startTime="285.923" endTime="286.183">
            <Sync time="285.923"/>
            thanks
      </utt>
      <utt id="1_67" startTime="286.782" endTime="287.62">
            <Sync time="286.782"/>
            thanks
      </utt>
</channel>

<!-- ********************************** -->

<channel startTime="0" endTime="290.368" speaker="Jitou" id="2">

      <utt id="2_2" startTime="39.359" endTime="39.694">
            <Sync time="39.359"/>
            yeah
      </utt>
      <utt id="2_4" startTime="40.949" endTime="44.228">
            <Sync time="40.949"/>
            it's a ?
            <Sync time="41.48"/>
            good idea because-
            <Sync time="42.342"/>
            ..
            <Sync time="43.171"/>
            after all it's an image
      </utt>
      <utt id="2_6" startTime="56.604" endTime="57.206">
            <Sync time="56.604"/>
            ah no
      </utt>

[... DELETED ...]

      <utt id="2_43" startTime="284.682" endTime="285.106">
            <Sync time="284.682"/>
            yeah
      </utt>
      <utt id="2_44" startTime="285.106" endTime="285.558">
            <Sync time="285.106"/>
            good luck
      </utt>
</channel>

<!-- ******************************** -->
```

```
<channel startTime="0" endTime="290.368" speaker="Jo" id="3">

      <utt id="3_2" startTime="38.945" endTime="40.448">
            <Sync time="38.945"/>
            yeah
            <Sync time="39.198"/>
            I think that sounds good
      </utt>
      <utt id="3_4" startTime="40.808" endTime="41.419">
            <Sync time="40.808"/>
            definitely
      </utt>
      <utt id="3_6" startTime="42.735" endTime="43.043">
            <Sync time="42.735"/>
            uh-huh
      </utt>

[... DELETED ...]

      <utt id="3_58" startTime="279.217" endTime="282.994">
            <Sync time="279.217"/>
            I guess it's important though to find out
            <Sync time="281.081"/>
            whether anyone
            <Sync time="281.770"/>
            has done it before
      </utt>
      <utt id="3_60" startTime="284.848" endTime="285.245">
            <Sync time="284.848"/>
            yeah
      </utt>
      <utt id="3_62" startTime="286.411" endTime="286.818">
            <Sync time="286.411"/>
            okay
      </utt>
</channel>

</dialog>
```

Chapter 4

Topic segmentation

This chapter explains the annotation for segmenting a meeting dialogue into topic-coherent episodes and for assigning topic titles and/or keywords.

A dialogue is decomposed in thematic episodes, with a topic (short description or list of keywords) assigned to each episode. Episodes represent here a "flat" structure that cuts across channels, and therefore they require minimal theoretical assumptions about discourse structure, as opposed to more complex hierarchical structures. Studies of thematic segmentation of narrative texts [4, 6] have shown that automatic methods have 10–15% error rates. This is only a little below human performances, given that annotators themselves are quite far from perfect agreement. Assigning descriptions to each topic (e.g., a sentence or a list of keywords), is a still more arduous task, since the definition of the "correct answer" is still more problematic. The informativeness of topic extraction is essential for dialogue understanding, even if it can be more difficult to evaluate.

4.1 Examples: episodes, topic titles, keywords

A sample topic block for TRN-11 is given below. Here the topical information block is straightforwardly derived automatically from the agenda or script defined in the IDIAP Scripted Meeting recording project (except for the keywords). This block goes to the end of the SDA.XML file as it appears from the general skeleton.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE dialog SYSTEM "dialog-sda-10.dtd">

<dialog>
```

```
<channel startTime="0" endTime="290.368" speaker="Iain" id="0">
      [... DELETED ...]
</channel>

<channel startTime="0" endTime="290.368" speaker="Vivek" id="1">
      [... DELETED ...]
</channel>

<channel startTime="0" endTime="290.368" speaker="Jitou" id="2">
      [... DELETED ...]
</channel>

<channel startTime="0" endTime="290.368" speaker="Joanne" id="3">
      [... DELETED ...]
</channel>

<episodes>
  <episode startTime="0.0"
           endTime="28.0"
           topic="Monologue2"
           keywords="phoneme recognition, computer vision"/>
  <episode startTime="28.0"
           endTime="48.0"
           topic="Consensus"/>
  <episode startTime="48.0"
           endTime="68.0"
           topic="Disagreement"
           keywords="window length"/>
  <episode startTime="68.0"
           endTime="88.0"
           topic="Consensus"
           keywords="power spectrum"/>
  <episode startTime="88.0"
           endTime="108.0"
           topic="Disagreement"
           keywords="phase"/>
  <episode startTime="108.0"
           endTime="168.0"
           topic="Monologue4"
           keywords="summary"/>
  <episode startTime="168.0"
           endTime="196.0"
           topic="Note taking"/>
  <episode startTime="196.0"
```

```
            endTime="290.368"
            topic="Discussion"
            keywords="spectrogram, computer vision, phase"/>
</episodes>

</dialog>
```

Chapter 5

Dialogue acts and adjacency pairs

5.1 Definition

Much has been written about the structure of dialogue, but very little is subject to general agreement, and even less is automatically detectable by a program. For SDA, a minimum is to label utterances with dialog acts (DA), and to find utterances that are functionally related. The DA is the role of an utterance for the progression of the dialog, such as "question", "statement" or "backchannel" [12]. Several lists of DAs exist (they are compared in [5]), and though there are many correspondences between lists, it is not easy to find a common denominator. For the meeting recording domain, we modified the set proposed by ICSI [2], itself an extension of DAMSL, with twelve main DA types, and several dozen refinements. The analysis and validation of the ICSI-MR data and the definition of the abstracted tagset MALTUS are described in Report IM2.MDM-09.

DA detection [10] has been attempted using either rule-based or statistical systems. Experiments at ISSCO with machine learning systems trained on the Switchboard corpus (two-party telephone conversations, DAMSL DA tagset) reproduced current performance levels of about 60–70% success (see Report IM2.MDM-12).

Adjacency pairs (AP) are functional links between pairs of utterances such as question / answer, invite / accept, offer / decline, etc. Such links prefigure more complex dialog structures, but even at this "flat" level, their detection by a program has not been evaluated yet; even annotation guidelines are not publicly available. We will focus on links between questions

and answers, as well as orders (or invitations, proposals, suggestions) and acceptance or refusal. Such links are already marked in ICSI-MR data, but the annotation guidelines are not very clear.

5.2 Mapping of ICSI dialogue acts onto a simplified set

The ICSI dialogue acts are derived from the Switchboard-DAMSL ones (see reports IM2.MDM-05, 06 and 09 for references). We first grouped the dialogues acts into 18 groups, which are less granular and easier to use for automatic annotation. A comparison between these groups and other annotation guidelines shows that these groups match reasonably many other categories.

5.3 The abstraction of the MALTUS tagset

We defined MALTUS (Multidimensional Abstract Layered Tagset for Utterances) in order to reduce the number of possible combinations by assigning exclusiveness constraints among tags, while remaining compatible with ICSI-MR. MALTUS is more abstract than ICSI-MR, but can be refined if needed. An utterance is either marked U (undecipherable) or it has a general tag and zero or more specific tags. It can also bear a disruption mark. More formally (? means optional):

```
DA -> (U | (gen_tag (spec_tags)?)) (.D)?
   gen_tag   ->  S | Q | B | H
   spec_tags -> (RP | RN | RU)? AT? DO? PO?
```

There are about 500 possible combinations of tags. The glosses of the tags, generally inspired from ICSI-MR, are:

- U = undecipherable (unclear, noisy)
- S = statement
- Q = question
- B = backchannel
- H = hold (floor holder, floor grabber, hold)
- RP = positive answer (or positive response)
- RN = negative answer (or negative response)

Index	Code	ICSI name	ICSI class	IM2 group
1	s	Statement	statements	statement
2	sj	Subjective Statement	statements	statement
3	qr	Or Question	questions	question
4	qrr	Or Clause After Y/N Q.	questions	question
5	qw	Wh-Question	questions	question
6	qy	Y/N Question	questions	question
7	%	Indecipherable	Disruption Forms	unclear
8	%-	Interupted	Disruption Forms	unclear
9	%–	Abandoned	Disruption Forms	unclear
10	t3	3rd-Party Talk	Disruption Forms	unclear
11	x	Nonspeech	Disruption Forms	unclear
12	b	Backchannel	Short Utterances	attention
13	bk	Acknowledgement	Short Utterances	attention
14	fg	Floor Grabber	Short Utterances	attention
15	fh	Floor Holder	Short Utterances	attention
16	aa	Accept	Short Utterances	positive_answer
17	h	Hold	Short Utterances	attention
18	qh	Rhetorical Question	Short Utterances	attention
19	bh	Rh. Q. Backchannel	Short Utterances	attention
20	t1	Self Talk	Short Utterances	unclear
21	aap	Partial Accept	Responses	positive_answer
22	arp	Partial Reject	Responses	negative_answer
23	nd	Dispreferred Answer	Responses	negative_answer
23	ng	Negative Answer	Responses	negative_answer
24	ar	Reject	Responses	negative_answer
25	am	Maybe	Responses	other_answer
26	na	Affirmative Answer	Responses	positive_answer
27	no	Other	Responses	other_answer

Table 5.1: *Reduction of ICSI-MR dialogue acts to ISSCO groups (first part).*

Index	Code	ICSI name	ICSI class	IM2 group
28	co	Command	Forward F.	suggestion
29	fx	Explicit-Performative	Forward F.	other_performative
30	cc	Commitment	Forward F.	commitment
31	cs	Suggestion	Forward F.	suggestion
32	oo	Open-Option	Forward F.	suggestion
33	qo	Open-ended Question	Forward F.	question
34	tc	Topic Change	Forward F.	change_topic
35	ba	Assessment/Appreciation	Backward-Looking F.	reaction
36	bc	Correct-Misspeaking	Backward-Looking F.	correction
37	bsc	Self-Correct Misspeaking	Backward-Looking F.	correction
38	bd	Downplayer	Backward-Looking F.	reaction
39	br	Repetition Request	Backward-Looking F.	reaction AND suggestion
40	bs	Summary	Backward-Looking F.	summary
41	bu	Understanding Check	Backward-Looking F.	reaction AND suggestion
42	df	Defending/Explanation	Backward-Looking F.	reaction
43	fa	Apology	Backward-Looking F.	politeness
44	by	Sympathy	Backward-Looking F.	politeness
45	2	Collaborative Completion	Descriptive Tags	NOTHING
46	e	Elaboration	Descriptive Tags	NOTHING
47	t	About-Task	Descriptive Tags	about_topic
48	d	Declarative Question	Descriptive Tags	NOTHING
49	f	Follow Me	Descriptive Tags	attention
50	g	Tag Question	Descriptive Tags	attention
51	j	Joke	Descriptive Tags	humor
52	m	Mimic Other	Descriptive Tags	reaction
53	r	Repeat Self	Descriptive Tags	NOTHING
54	rt	Rising tone	Descriptive Tags	NOTHING
55	fe	Exclamation	Conventionalities	exclamation
56	ft	Thanks	Conventionalities	politeness
57	fw	Welcome	Conventionalities	politeness

Table 5.2: *Reduction of ICSI-MR dialogue acts to ISSCO groups (second part).*

- RU = other answer (or undecided answer or response)
- RI = restated information
- DO = command or other performative (can be refined into: command, commitment, suggestion, open-option, explicit performative)
- AT = the utterance is related to attention management (can be refined into: acknowledgement, rhetorical question backchannel, understanding check, follow me, tag question)
- PO = the utterance is related to politeness (can be refined into sympathy, apology, downplayer, "thanks", "you're welcome")
- D = the utterance has been interrupted or abandoned

These tags should thus be combined into labels, annotated on utterances as indicated above.

Chapter 6

Coreference: 'ref2doc' application

The detection of referring expressions (RE), that is, chunks of an utterance that point at or denote an object, person, place, concept, etc. is another component of SDA. Named entities (proper names, dates, times, amounts, etc.) have been a target in the MUC and ACE campaigns[1], and we propose here to follow the MUC-7 guidelines. In the future, referring acts in other modalities (e.g., pointing gestures) could also be annotated and related to linguistic REs. Coreference links between REs group those REs that denote the same entity, and here again the MUC-7 guidelines provide a reasonable definition of the targeted task, despite some problematic issues [13]. Performance levels for this task, using robust methods in the case of (relatively) unrestricted domains, reach 60–70%, with either of the existing evaluation methods [7].

Here, we focus first on a particular class of REs, namely references to meeting documents.

6.1 Annotation Model

The annotation model for the references to documents is fully integrated into the shallow dialogue analysis model and implemented in XML. The main idea is to add external annotation blocks that do not alter the master resource—here the timed meeting transcription, divided into separate

[1] Message Understanding Conferences, see `http://www.itl.nist.gov/iaui/894.02/related_projects/muc/`. Automatic Content Extraction, see `http://www.itl.nist.gov /iad/894.01/tests/ace/`.

channels. However, REs are annotated on the dialog transcription itself. A more principled solution, but more complex to implement, would be to index the master transcriptions by the number of words, then externalize the annotation of REs as well.

Here is an example of the XML annotation with ref2doc information (`er` stands for RE:

```
<dialog>
 <channel id="1">
   ...
   <er id="12">The title</er>reads...
 </channel>
 ...
 <ref2doc>
 ...
 <ref er-id="12"
      doc-file="LeMonde030404.Logic.xml"
      doc-id="//Article[@ID='3']/Author"/>
     ...
 </ref2doc>
</dialog>
```

The `ref` pointers from the REs to the document elements are grouped in a `ref2doc` block at the end of the document, using as attributes the index of the RE (`er-id`), the document filename (`doc-file`), and an XPath expression (`doc-id`) that refers to a document element from the XML document representation. This requires the availability of an XML logical structure of the document, which is constructed by tools provided by the IM2.DI project.

6.2 How to annotate `ref2doc`

There are two phases in `ref2doc` annotation: (1) annotate the REs that refer to documents (the <er> tags inserted on the transcript); (2) relate each RE to the respective document element (the <ref2doc> block at the end). Here are the main steps that should be respected for manual annotation.

- Start with an SDA.XML transcription of the dialog.
- Annotate each referring expression with <er>...</er> (it is easier if you monitor a browsable version of the transcription, such as those generated with sda-to-html-2.xsl)
- Save the annotated file as SDAER.XML

- Convert the SDAER.XML file using numeroter-ajouter.xsl If you can, insert the name of the document file (LeMondeXXXXXX.Logic.xml) in the XSL file. This will save time later. This XSLT generates the indexes for the REs (<er id="14">..</er>) as well as a template for the <ref2doc> annotation block at the end of the file. Save the result as an SDAREF.XML file.

- The ref2doc block will contain links from REs to parts of the document, as for instance: <ref er-id="39" doc-file="LeMonde061103.pdf" doc-id="//Article[@ID='10']//Title" /> In the template you generated at 4., the doc-id attribute is empty (and possibly the doc-file one), but the er-id have the numbers.

- To complete the ref2doc block, the best is to use an HTML rendering of the transcript, as obtained by processing the SDAREF.XML file with sda-to-html-2.xsl . This will highlight the REs and their index numbers.

- Display the SDAREF.XML file, the HTML file and the Logic.xml file (the document). Use the HTML and the document to determine to which part of the dcuments each RE refers, then write the corresponding xpath in the SDAREF.XML file.

- Check that all the entries of the ref2doc block are complete.

6.3 Guidelines for `ref2doc` Annotation

The annotation of the ground truth references can be done directly in the XML format described above. In a first pass, the annotator marks the REs (with `<er>...</er>` tags), if they refer to an article or to one of its parts, for instance its title or author. Table 6.1 below synthesizes the observed patterns of REs. More guidelines for annotations are provided in the list below.

The REs are then automatically indexed, and a template for the `ref2doc` block and an HTML view can be generated using XSLT. In a second pass, the annotator fills in directly the attributes of the `ref2doc` block in the template, with, for each RE (`er-id`), the name of the journal file that the RE refers to (`doc-file`), and the XPath to the respective document element (`doc-id`), using its `ID`. The following separate windows are all required for the annotation:

1. text/XML editor for the `ref2doc` block of the dialog annotation file;
2. HTML browser for the serialized HTML transcript (with REs in boldface);
3. XML browser for the document structure representation (one per document);
4. PDF viewer for the actual layout of the articles (one per document).

The following observations can also guide `ref2doc` annotation[2]:

1. REs that correspond to exact quotations of an article's title should be annotated, if they refer to the article. In this case, they could even be written with quotes in the transcript (as in 'Victor Hugo wrote *"Les Misérables"*.'). Approximate quotations often refer to entities mentioned in the documents, rather than to the document elements themselves, and should not count as REs referring to documents. In the first case, the referent should be annotated as `doc-id="//Article[ID='123']"`; only a reference to the title of an article (such as 'I did not understand its *title*.') should be annotated as `doc-id="//Article[ID='123']/Title"`.
2. Do not annotate references to articles that are not in the meeting documents (such as articles which are not on the first page, or references to pages, for the Fribourg meetings).
3. Elliptical REs such as "the next one" should be "solved" mentally to decide whether they refer to the next article (in which case they should be annotated) or to the next speaker (as in some Fribourg meetings), in which case they should not be annotated.
4. For pictures or drawings, the annotation should first make reference to the article which they are part of, such as `doc-id="//Article[ID='123']/Picture"`, the exact XPath depending on how the picture is encoded into the XML structure of the document.
5. If a RE makes reference to a subpart of an article, a subpart that is not annotated in the XML document structure, the `ref2doc` annotation

[2]Thanks to Denis Lalanne and Emmanuel Palacio for initial input on this list.

should stop at the level of the article—unless it is obvious that the subpart was forgotten from the XML structure.

6. When a RE mentions both the type of an article and its title, only one RE should be annotated, as in `<er> le point de vue` *Une apocalypse aveugle*`</er>`. Do not annotate twice, as in `<er>le point de vue <er>`*Une apocalypse aveugle*`</er></er>`.

6.4 Empirical Analysis of Occurring REs

The patterns of the annotated REs are synthesized in Table 6.1 according to the type of entity they refer to. This analysis attempts to derive regular expressions that describe the range of variation of the REs that refer to documents, but without generalizing too much. Words in capital letters represent classes of occurring words: NEWSP are newspaper names, SPEC is a specifier (one or more words, e.g., an adjective or a relative sentence), DATE and TITLE are obvious. Items in brackets are optional, and | indicates an exclusive-or. The patterns derived here could be used to recognize automatically such REs, except for two categories—anaphors and (discourse) indexicals—that must be disambiguated.

Referent	#	RE
Journal	6	(le|du) NEWSP
	2	le journal
Front page (*une*)	33	la une NEWSP
	6	la une DATE+NEWSP
	5	(la|une) une
Article	33	(l'|le premier|le dernier) article
	31	cet article
	15	[l'] article suivant
	14	un [petit] article SPEC
	11	[un] autre article [SPEC]
	7	l'article SPEC
	5	[l'article] "TITLE"
Title	10	le [grand] titre [principal]
	4	(premier|second|autre) titre
Other text elements	12	[un] autre (point|sujet| fait) [SPEC]
	10	... (rubrique|encart|enquête| page|actualité|highlight| analyse) ...
	5	(premier|dernier) point
	3	un [petit] point [SPEC]
	3	les grands points de l'actualité
	3	(le|au) point de vue [SPEC]
Graphic elements	11	... (dessin|photo|schéma| image|figure) ...
Authors	6	l'auteur
	5	le journaliste
Anaphors	27	ils
	12	il
	8	l'
	4	(le|au) dernier
	3	autre chose [SPEC]
	2	on
Indexicals	5	là
	4	ça
	4	celui-là
	2	celui-ci
	2	celui SPEC

Table 6.1: Patterns of REs that refer to documents, in French, ordered by the type of the referent (9 REs out of 322 did not follow these patterns).

Chapter 7

Related work and perspectives

Several other reports are related to this one, offering more details on specific points. For conversion procedures between various input formats, using XLST scripts, as well as for generating tabular and HTML output formats, see Report IM2.MDM-13: *Data formatting and conversion procedures: summary of results.*

For an overview of the use of these annotations into a dialogue database, accessed by a Transcript-based Query and Browsing Interface (TQB), including multimedia rendering of annotated transcriptions through hyperlinks, and installation guidelines, see Report IM2.MDM-10: *Database and TQB Demonstrator: Installation Manual.*

The shallow dialog analysis model presented in this paper groups a set of useful features from human dialogs that are recorded during meetings. The various software modules that constitute the SDA parser are under implementation; the evaluation of the results would give an idea of current performances on the joint SDA task [8]. The relevance of the extracted features should also be ascertained through user-based studies of the meeting retrieval application.

Possible extensions of the SDA model are:

1. argumentative annotation (Pallotta & Ghorbel);
2. annotation of polarity, e.g. agreement, disagreement, or neutral;
3. annotation of activity types (Bales, Levinson);
4. annotation of group tasks dimensions (McGrath).

Bibliography

[1] Armstrong, S., Clark, A., Coray, G., Georgescul, M., Pallotta, V., Popescu-Belis, A., Portabella, D., Rajman, M. and Starlander, M., "Natural Language Queries on Natural Language Data: a Database of Meeting Dialogues", Proc. of NLDB'2003, Burg/Cottbus, Germany, 2003.

[2] Bhagat, S., Dhillon, R., Carvey, H., Krupski, A., and Shriberg, L., "Labeling Guide for Dialog Act Tags in the Meeting Recorder Meetings", ICSI Report, Berkeley, CA, USA, 2002.

[3] Choi, F. Y. Y., Wiemer-Hastings, P., and Moore, J., "Latent Semantic Analysis for Text Segmentation", Proc. of EMNLP'2001, Pittsburgh, PA, USA, 2001, p. 109–117.

[4] Hearst, M. A., "TextTiling: Segmenting Text into Multi-paragraph Subtopic Passages", Comp. Ling., 23(1), 1997, p. 33–64.

[5] Klein, M., Bernsen, N. O., Davies, S., Dybkjær, L., Garrido, J., Kasch, H., Mengel, A., Pirrelli, V., Poesio, M., Quazza, S., and Soria, C., "Supported Coding Schemes", MATE Project LE4-8370, Deliv. D1.1, 1998, `http://mate.nis.sdu.dk/about/D1.1/`.

[6] Passonneau, R. J., and Litman, D. J., "Discourse Segmentation by Human and Automated Means", Comp. Ling., 23(1), 1997, p. 103–140.

[7] Popescu-Belis, A., "Evaluation-Driven Design of a Robust Reference Resolution System", Nat. Lang. Eng., 9(2), 2003, p. 1–26.

[8] Popescu-Belis A., Clark A., Georgescul M., Zufferey S. & Lalanne D., "Shallow Dialogue Processing Using Machine Learning Algorithms (or not)", in Bourlard H. & Bengio S., eds. (2004), *Multimodal Interaction and Related Machine Learning Algorithms*, LNCS Series, Springer-Verlag, Berlin, in press.

[9] Popescu-Belis A., Georgescul M., Clark A. & Armstrong S., "Building and Using a Corpus of Shallow Dialogue Annotated Meetings", in Proc. LREC'2004 (Fourth International Conference on Language Resources and Evaluation), Lisbon, Portugal, vol.IV, p.1451-1454, 2004.

[10] Stolcke, A., Ries, K., Coccaro, N., Shriberg, E., Bates, R., Jurafsky, D., Taylor, P., Martin, R., Meteer, M., and Van Ess-Dykema, C., "Dialog Act Modeling for Automatic Tagging and Recognition of Conversational Speech", Comp. Ling., 26(3), 2000, p. 339–371.

[11] Stolcke, A., and Shriberg, E., "Automatic Linguistic Segmentation of Conversational Speech", In Proc. of ICSLP-96, Philadelphia, PA, USA, 1996, p. 1005–1008.

[12] Traum, D. R., "20 Questions for Dialog Act Taxonomies", Journal of Semantics, 17(1), 2001, p. 7–30.

[13] Van Deemter, K., and Kibble, R., "On Coreferring: Coreference in MUC and Related Annotation Schemes", Comp. Ling., 26(4), 2000, p. 629–637.