Multi-layer Dialogue Annotation for Automated Multilingual Customer Service

Hilda Hardy[1], **Kirk Baker**[2], **Laurence Devillers**[3], **Lori Lamel**[3], **Sophie Rosset**[3], **Tomek Strzalkowski**[1], **Cristian Ursu**[4] and **Nick Webb**[4]

[1]ILS Institute
University at Albany, SUNY
Albany, NY 12222
{hardyh,tomek}
@cs.albany.edu

[2]Dept. of Computer Science
Duke University
Durham, NC 27708
kbaker3@email.
unc.edu

[3]LIMSI-CNRS
BP 133
91403 Orsay cedex, France
{devil,lamel,rosset}
@limsi.fr

[4]Dept. of Computer Science
University of Sheffield
Sheffield S1 4DP UK
{c.ursu,n.webb}
@dcs.shef.ac.uk

Abstract

For the AMITIÉS multilingual human-computer dialogue project [1], we have developed new methods for the manual annotation of spoken dialogue transcriptions from European financial call centers on multiple levels. We have modified the DAMSL schema [2] to create a dialogue act taxonomy appropriate to the functions of call center dialogues. We use a domain-independent framework populated with domain-specific lists to capture the semantics of spoken dialogues. Our new flexible, platform-independent Java annotation tool, called XDMLTool, takes plain-text dialogue files as input, and yields annotated files in the widely used XML format. To date, XDMLTool has been used to annotate several hundred call-center dialogues in France, the UK and the US. We present definitions of each tag as well as examples in English and French. These annotation methods are developed for an experimental system that automates financial call centers in Europe. The multi-level annotation scheme has been used to develop a prototype triaging application for financial services.

1 Introduction

The field of natural language human-computer dialogue is closely linked to the speech (or dialogue) acts theory, which postulates that speakers' utterances carry embedded communication devices that manipulate the beliefs of the listeners. For example, Bunt [3] developed theories of dialogue acts within what he called context change theory. These dialogue acts refer to the "functional units" used by a speaker to change a context. Bunt's theory makes a general distinction between utterances that accomplish a part of the desired transfer of factual information (known as task-oriented dialogue acts) and those that serve explicitly to control the dialogue (known as dialogue control acts).

To date there have been two major approaches to implementation of dialogue systems: grammar-based and statistical. In the grammar-based approach, which is prevalent in commercial systems (such as Nuance's various telephony products) as well as in practically oriented research prototypes (e.g., systems developed under the DARPA Communicator program), a complete dialogue grammar or transition graph is designed to guide the conversation and predict user responses, which is suitable for closed domains only. In the statistical approach, a dialogue transition graph is derived from a large body of recorded and annotated conversations. The best known dialogue annotation system, Dialogue Act Markup in Several Layers (DAMSL), developed by the Multiparty Discourse Group [2], is a system of functional dialogue acts and can conceivably be trained given a sufficiently large dialogue corpus and the right selection of features over which machine learning is done.

Nonetheless, DAMSL structure, even if learned, captures only the functional layer of the dialogue, whereas we are also interested in the semantic

layer, that is, the information exchange and information building effects of a conversation. In order to properly understand a dialogue, both semantic (information content) and functional (speaker intentions) layers need to be considered. For instance, in call center dialogues much of the semantic layer can be represented as *transactions*: buying, selling, reporting problems, verifying status, etc. For example, *ChangeAddress* is a frequent transaction in services call centers. Furthermore, transactions have attributes, such as *AccountNumber, CustomerName, Address,* etc., whose values differentiate one transaction of the same type from another. A transaction can be initiated by either party of the dialogue, but it cannot be properly executed until its attributes are understood. Thus, a typical information exchange dialogue consists of possibly overlapping segments of attribute negotiation and transaction execution. One of the major advantages of transaction semantics is that it can support mixed initiative dialogue (a key property of team interaction) as well as its data-driven dynamics.

2 Design and Use of XDMLTool

XDMLTool (eXtensible Dialogue Markup Language Tool) was designed to be an efficient, flexible software tool for annotating transcribed dialogues according to semantic, functional, and stylistic characteristics. Written in Java, XDMLTool is easy to install and use on Solaris, Linux, and Windows platforms. The expected input format is plain text files, one file per dialogue, with turns labeled according to who is speaking. In our case these files are created with the Transcriber

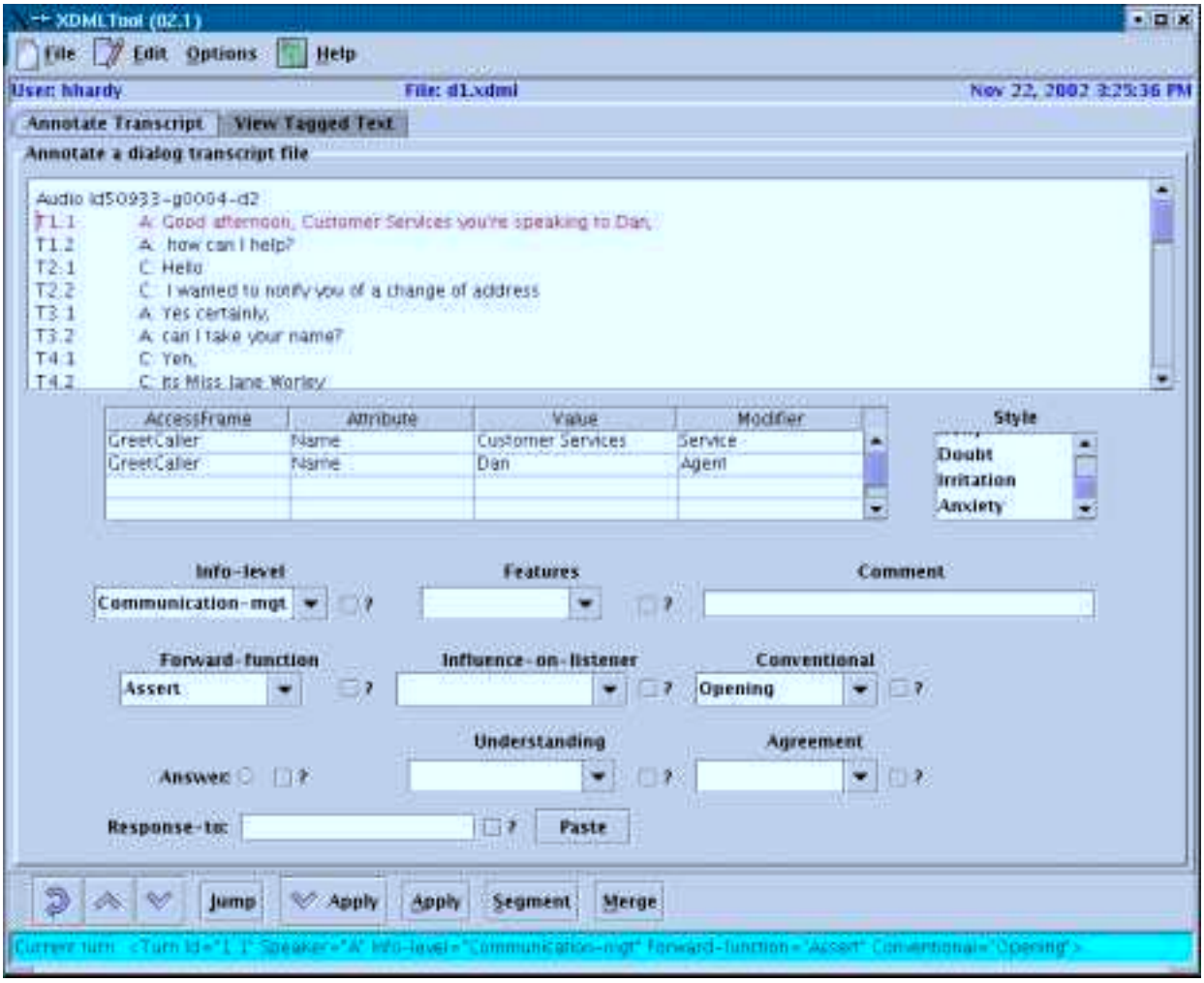

Figure 1: XDMLTool User Interface, "Annotate Transcript" panel

tool [4]. After the user has made annotation selections, XDMLTool stores the file in the widely-used XML format. This format was chosen for its easy automatic processing capabilities.

The first version of XDMLTool was posted for AMITIÉS partners in France, the UK, and the US in October 2001. Frequent updates were made based on users' suggestions during the following year. To date, several hundred call-center dialogues in English and French have been annotated.

XDMLTool was designed with a windows interface familiar to those who work with current office software, so that annotators can use it easily after a brief introduction. Several timesaving features have been incorporated, such as automatic tag selections, rapid navigation capabilities, and user-defined multi-label tagging. The tool works well with single-pass or multi-pass approaches to annotation. XDMLTool provides a visual organization important for meaningful, at-a-glance review of annotation selections (See Figure 1). A related tool for querying annotated files, QXDMLTool, has been developed by Consortium partners at the University of Sheffield. Both tools, as well as the AMITIÉS annotation manual, may be downloaded from the AMITIÉS website [1].

3 Functional (Dialogic) Annotation

The functional or dialogic aspect of an utterance has to do with its role or purpose in the interchange. Statements, questions, answers, and expressions of thanks are examples of such functions, or dialogue acts. To annotate this layer for the dialogues in the AMITIÉS corpus, we have found that in general, the DAMSL tags work well [2]. With this level of annotation, both the categories and the lists remain largely independent of the domain. However, we have made some adjustments in the tag set in order to reflect more accurately the features found in the AMITIÉS corpus.

Our taxonomy follows the general DAMSL categories Information-Level, Communicative Status (Features), and Forward- and Backward-Looking Functions. This way we can capture broad topical distinctions, unusual occurrences in conversations, and ways in which a particular utterance relates to previous or subsequent parts of the dialogue. Our tag set for the categories Information Level and Communicative Status (Features) is shown in Figure 2. The category *Information Level*

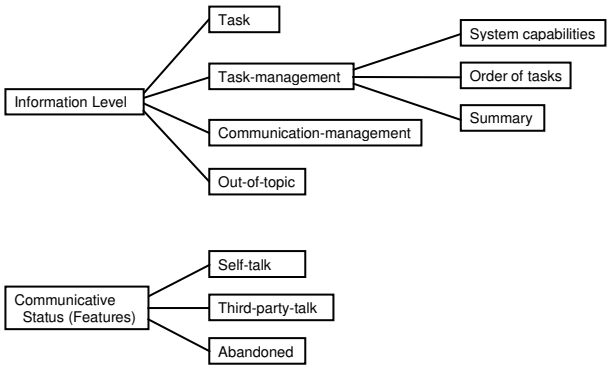

Figure 2: Hierarchy of annotation labels for Information Level and Communicative Status

allows us to make an abstract description of the content of each utterance. We are concerned here with the broad topic of a particular turn or portion of a turn. Is the speaker participating in an exchange of information that will accomplish a task? Or is he stepping above the task, so to speak, and talking about the process needed to achieve some goal or complete the task (Task-management)? Perhaps his words serve to initiate, maintain or close the conversation (Communication-management). Or perhaps he is digressing from the subject of the dialogue (Out-of-topic).

Most turns in call-center dialogues make progress toward accomplishing a customer-service task, such as making a payment, verifying a customer's identity, or giving the customer information about his account balance. Questions and answers, directions, suggestions, explanations, commitments to perform some action connected with the task, and agents' offers of help all fall into this category. For convenience, XDMLTool labels every turn (or turn segment) *Task* by default. An utterance concerned with "doing the task", then, receives the Information Level label Task.

If an agent or customer is not making direct progress toward completing the task, but instead is talking about the task or the *process* of doing the task, we use the label *Task-management*. It is as if the person were standing outside or above a "task arena" and looking in. The person makes statements such as "The task cannot be completed right now", "Now we can start this task", "First we'll do this; then we'll work toward that goal", "Ok, that job is done". We have found it useful to subdivide Task-management into three categories: System capabilities, Order of tasks, and Summary.

The *System capabilities* category of Task-management means that the speaker is addressing problems that the computer system or the service center can and can't solve; this is the "competence domain" for customer service.

C: Erm I'd like to make a payment please on my account
A: Certainly Mrs Smith **but I'm sorry at the moment we are face, facing a technical problem** (18mar02-421.trs.txt)

A: alors euh un petit instant
A: je ne peux pas vous commander de chéquier monsieur c'est bloqué (am7-d52.txt)
A: hold on a second please
{A: I can't order a checkbook because the account is frozen}

For *Order of tasks,* the speaker talks about the order in which tasks will be completed, or indicates that a task will be started. Typically these utterances take one of the following forms: "We'll do this before this", "Before I can do this task, I need to ask you some questions", "Now we'll do this".

A: (pause) Right the next thing we're doing now is waiting for your bank to confirm for you (26mar02-689.trs.txt)

A: alors je vais vous demander tout d'abord votre numéro de compte s'il vous plaît (0pe.txt)
{A: ok, first I am going to ask you for your account number please}

Occasionally the agent will make a statement that serves to summarize, wrap up, or recapitulate the task that has been accomplished in the dialogue, or to indicate that the task has been completed. We reserve the *Summary* tag for summary or completion statements that refer to the entire task, such as closing an account or changing an address, not parts of the task.

A: Right, one account closed as of today (26mar02-686.trs.txt)

A: No problem now we've put a stop on there for you (26mar02-693.trs.txt)

A: [b-] ben je vais le mettre au 20 [-b] y a aucun problème donc à partir donc du mois de septembre vos prélèvements se feront à la date du 20 (40.800-47.240) (0pb.txt)
{A: I will put it on the 20th there's no problem so starting in September your payments will be made on the 20th}

A third category under Information-level serves to describe utterances that deal with social obligations [3] such as greetings, introductions, apologies, expressions of gratitude, farewells, as well as marks which maintain the conversation. Conventional phrases such as "hello" and "goodbye", as well as backchannel words or non-words such as "uh-huh", "yes", and "ok" are examples of *Communication-management*. Ordinary sentences and phrases used to signal misunderstanding or to manage delays in the conversation should also be labeled Communication-management. One useful test suggested by Allen and Core [2] is to remove the utterance in question from the dialogue. The conversation might be less fluent but would still have the same content relative to the task and how it is solved.

Another Information-level category has to do with brief or extensive digressions from the task in the conversation. *Out-of-topic* includes jokes, non-sequiturs, small talk, and any comments or remarks that have no direct bearing on accomplishing the task.

A: Thank you, you're speaking to Paul, good afternoon.
C: Uh hmm
A: Right so it's Paul and Pauline
C: (laughs) (26mar02-687.trs.txt)

C: j'aurais voulu acheter un petit téléviseur pour noël (am5-d40.txt)
{C: I wanted to buy a television for Christmas}

Whereas every utterance in a dialogue can be assigned an Information Level tag, *Communicative Status* tags are intended to be used only in exceptional circumstances. We use three labels under this category: Self-talk, Third-party-talk, and Abandoned. Interruptions and unintelligible utterances are best annotated using a transcription tool while the user is listening to the dialogue.

A *Self-talk* utterance indicates that the speaker does not intend the other person to respond to or otherwise use what he is saying. The speaker's apparent intentions are the key, rather than whether or not the other person actually responds to or uses the utterance.

A: It's a whole year or more
C: Gosh I must have paid other things by cash then
A: Hmm
C: I can't believe that
A: There you go
A: Right

C: Ok
A: We shall now do your address (26mar02-690.trs.txt)

An utterance labelled *Third-party-talk* is one in which the speaker is addressing someone other than the second party to the conversation. Typically a customer will ask a question of a family member; or an agent will speak to a customer who is with him, while the agent is holding a telephone conversation with another agent.

A speaker abandons an utterance occasionally, when he makes an error or changes his mind about what to say. The *Abandoned* label should be used only if the utterance has no effect on the progress of the dialogue; that is, the utterance could be removed from the conversation without changing its content. An utterance can be marked *Abandoned* whether or not the speaker was interrupted, as long as the speaker actually leaves his thought and does not return to it. We do not use the label Abandoned for cases in which the speaker corrects himself during the turn.

A: But are? Are all these?
C: I don't think
A: Sorry, err err you're coming through to change of address
A: Is that right?
C: No no I've lost my New Look card (26mar02-701.trs.txt)

An utterance having a *Forward-Looking Function* anticipates the future in some way, having an effect on what is answered, discussed or done next. The speaker may be making a statement, asking a question, or committing himself to some course of action. He may be suggesting or directing the other person to do something. Forward-looking functions can be distinguished from backward-looking functions in that backward functions are primarily responses to something that was said, and forward functions typically elicit a response. Some functions, such as the various kinds of statements, as well as the Expression function, have either a forward or a backward orientation, depending on the context. Sometimes an utterance can be tagged with labels from both the forward and the backward categories. For example, the backward function Answer is also labeled Assert. If more than one tag (forward, backward, or both) is applicable for an utterance, the annotator should select all appropriate tags. Our hierarchy of Forward-Looking Functions is illustrated in Figure 3.

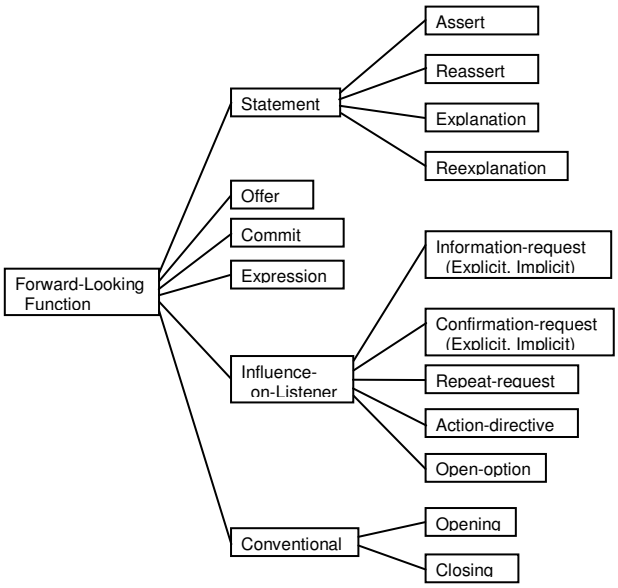

Figure 3: Hierarchy of annotation labels for Forward-Looking Function

In general, a sentence or phrase that is a *Statement* can be said to be true or false. A statement makes a claim about the world, and tries to change the beliefs of the listener. We use four tags under the category Statement: Assert, Reassert, Explanation, and Reexplanation.

Assertions and *Reassertions* are ordinary statements, distinguished by whether or not the speaker has already made the claim in an earlier part of the dialogue. Here we include, first, any yes/no answer to a question. Second, Assertions and Reassertions can take the form of statements that communicate some specific details. The attributes and possibly values contained in these statements are also annotated at the semantic level. Third, Assertions or Reassertions may take the form of a recapitulation, reformulation, or summary. These are also tagged "Task-management-summary" (see above).

In contrast to Assertions, which are simple statements, *Explanations* are reasons people give for their answers or for the questions they ask, or elaborations about topics such as customer service policies. Either the agent or the customer may use these. Explanations and Reexplanations, similar to Assertions and Reassertions, are distinguished by whether the statement has been made previously in the dialogue.

A: We can't in, no we can't increase the limit if you are already above it. Because obviously, well, it's just something we don't offer I'm afraid

C: Alright okay thanks anyway (22mar02-509.trs.txt)

C: et sinon euh je voudrais savoir est-ce que je peux recevoir un chéquier
C: parce que j'en ai un mais ça date de il date d'il y a longtemps et euh donc si c'est possible d'en recevoir un ou pas (am18-d27.txt)
{C: I wanted to know if it's possible to receive a check-book
C: because I've got one but it's an old one and I was wondering if it was possible to receive a new one}

Offers are implicit or explicit questions that, if answered in the affirmative or with some positive information, mean that the speaker will perform some action for the listener.

A: Is there any other accounts that I need to deal with for you?
C: Erm actually erm do you also deal with erm accounts for River Island?
A: Yes (26mar02-686.trs.txt)

C: ben j'en ai pas pour l'instant
C: mais je peux vous donner mon numéro de portable (am10-d25.txt)
{C: I don't have any for the moment
C: but I can give you my cell phone number}

The *Commit* tag is used for utterances in which the speaker obligates himself to perform a future action, phrased in such a way that the commitment is not contingent on the listener's agreement. A Commit may also be the response to an Action-directive.

A: et ben il s'est pas débloqué le chéquier ne s'est pas débloqué
A: donc je vais faire le nécessaire (am-d12.txt)
{A: if it is not unfrozen, the checking account is not unfrozen
A: I'll see to it}

The *Expression* tag encompasses conventional phrases such as "Thank you", "I apologize", and "Sorry", exclamations, short words used to hold or grab the turn, such as "Right" or "Okay", and other expressive phrases. These may also be tagged Backchannel, Accept or Non-understanding, depending on the context.

In the *Influence-on-listener* group of tags, the speaker is asking the listener a question, directing him or her to do something, or suggesting some course of action the listener may take.

A request for information, whether it is spoken in the interrogative form (Explicit), as in the following two examples, or the imperative or declarative form (Implicit), is tagged *Information-request.* We exclude from this category questions that call for a yes/no answer.

A: Ok, can you just confirm your name and address please? (05apr02-45.trs.txt)

C: c'est pour euh je voudrais savoir le solde de mon compte (am0-d45.txt)
{C: it is for euh I would like to know my account balance}

A *Confirmation-request* is an utterance that calls for the listener either to confirm or to deny the request or the question; in other words, it calls for a simple acceptance or rejection: a yes/no answer. In this category we also make a distinction between Explicit (illustrated below) and Implicit.

A: And this is your Principles card isn't it?
C: Yeah (26mar02-684.trs.txt)

The label *Repeat-request* is used to mark any request, whether it is for information or confirmation, that has been made earlier in the dialogue.

If the speaker directs the listener to perform some action, we label the utterance *Action-directive.* If the directive is done to manage some delay in the conversation, for example, "Please wait" or "Bear with me", then we also use the Information Level label Communication-management. Agents and customers in call-center dialogues rarely phrase action-directives in a "Do this" manner. Instead they make a polite request or a statement of a problem that must be solved or a task that needs to be done.

C: Oh hi **I want to change my address please**
A: Ok, **just bear with me a second** (05apr02-45.trs.txt)

C: alors vous pouvez me verser 1000 francs (am7-d55.txt)
{C: so you can transfer 1000 francs on my account}

If a speaker suggests a course of action but puts no obligation on the listener, we use the tag *Open-option*. The difference between Open-option and Offer has to do with who will perform the action. The Offer (see above) means that the speaker proposes to do something for the listener, as in "I can do this for you", or "What can I do for you?" The Open-option, on the other hand, suggests that the listener or some other person perform the action. This type of utterance takes the form "You can do this", or "This [option or course of action] can be done."

A: Yeah you surely can. **You can pay over the phone using a debit card** (15mar02-399.trs.txt)

A: vous souhaitez rembourser par anticipation (am3-d29.txt)
{A: you want to pay in advance}

The *Opening* tag indicates that the speaker is beginning the interaction by using a conventional social phrase to greet the listener, or by replying to such a greeting with a conventional phrase. Often the speaker, if he or she is a customer service representative, will identify the service name and/or the agent name as part of the greeting. These details are marked at the semantic level.

The *Closing* label is used for turns in which the speaker utters a conventional social phrase or expression to finish or wrap up the conversation. If the annotator selects the Opening or Closing tag, for convenience XDMLTool automatically assigns the Information Level tag Communication-management.

Utterances in the *Backward-Looking Function* category respond in some way to one or more previous turns in the dialogue. An answer to a question is one example of a common backward-looking function. If the speaker signals some level of understanding or not understanding what the previous speaker has said, we use one of the five tags in the Understanding sub-category. If the speaker signals some level of agreeing or disagreeing with the previous speaker's question (or some degree of accepting or rejecting the previous speaker's proposal), then we select a tag in the Agreement sub-category. Note that most, if not all, acceptances and rejections are also answers. Our set of Backward-Looking Function tags is shown in Figure 4.

The Response-to field on XDMLTool's user interface provides a place to annotate the antecedent, or the utterance to which the current turn is responding. Because the most common antecedent is the previous turn, XDMLTool automatically fills in the number of the previous turn or turn segment when the user selects any Backward Function tag. This number may be overridden manually if the antecedent occurs earlier in the dialogue, or if more than one utterance forms the antecedent.

An *Answer* is a response to an Information-request or Confirmation-request. An answer by definition will always be an assertion, as it provides information or confirms a previous supposition, and it makes a claim about the world. XDMLTool automatically tags an utterance Assert if the user chooses Answer.

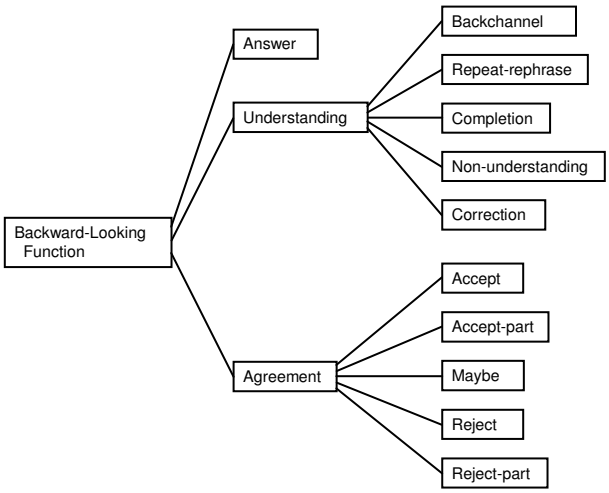

Figure 4: Hierarchy of annotation labels for Backward-Looking Function

An *Understanding* response to an utterance reveals whether and in what way the speaker heard and understood what the other speaker was saying. This aspect indicates nothing about whether the speaker accepts or rejects what was heard. Because a speaker may be indicating understanding and agreement at the same time, choices in both areas can be appropriate. We use five tags in the Understanding category: Backchannel, Repeat-rephrase, Completion, Non-understanding and Correction. Because all these types of utterances can be categorized at the Information Level of Communication-management, XDMLTool makes that choice automatically. The user should change the Information-level label to "Task" if he or she determines that the utterance also has a Task role; for example, Accept or Commit.

A *Backchannel* response is typically a short phrase such as "okay", "yes" or "uh-huh", indicating that the speaker heard and understood the previous utterance, but did not necessarily accept what he heard. A Backchannel utterance may be paraphrased "I heard you", "I understand what you said", "I heard that; please go on", or "That's clear; you can continue". This type of response may or may not interrupt the previous speaker, or it may occur while the other speaker is still speaking. If the utterance also counts as an acceptance or a commitment, it should be annotated at both levels.

The annotator should look carefully at the context in order to determine the appropriate tags. The capability of listening to the audio signal can potentially help the annotator to distinguish backchannel and agreement.

The *Repeat-rephrase* label is used for utterances that repeat or paraphrase the previous speaker's words, to show that those words were understood but not necessarily accepted. If the speaker repeats or paraphrases some words with uncertainty, or a rising inflection (indicated in the transcription by a question mark), then we use the tag Non-understanding (see below).

A: Have you got erm? Have you got a card? It's got like a picture of a shop on the front
C: Picture of a shop, let's see if I can find any (26mar02-704.trs.txt)

C : on a le droit à un débit de combien
A : à combien vous avez le droit (am-d6.txt)
{C : how much am I allowed to spend
A : how much you can use}

The *Completion* tag is used for utterances that indicate understanding by continuing or finishing the other person's sentence or phrase.

A: And then you say it's E R
C: H A double L (22mar02-518.trs.txt)

C: voilà je viens de recevoir mon dernier relevé là et on me demande de vous contacter pour recevoir un
A: un chéquier en euros (am0-d41.tx)
{C: yes I've just received my bank statement and there's a note joined with it that tells I have to call to receive a
A: a checkbook in euros}

If the speaker has not understood or has partially understood something he has just heard, we use the *Non-understanding* tag. Utterances of this type can usually be paraphrased "What did you say?" "What did you mean?" or "Is this what you said?" Many of these examples can also be labeled Explicit Confirmation-request.

A: Uh huh the name was Noble
C: Sorry?
A: N O B L E
C: N O B L E? I can't hear properly, on this phone it's on a very very low sound on it thank you very much (06mar02-112.trs.txt)

A: vous ne connaissez pas les détails des timbres fiscaux
C: comment ça (am-d17.txt)
{A: you do not know the payment details
C: what}

The *Correction* tag is used to indicate that the speaker has understood what the other person has said, but wants to correct a perceived error in the other person's utterance. We reserve the label Correction for cases in which the speaker corrects the other person, not for cases of self-correction.

A: Is it Midonhall Road?
C: Milderhall M I L D E R (22mar02-518.trs.txt)

The set of tags in the *Agreement* category indicate whether the speaker accepts a proposal, offer or request, or confirms the truth of a statement or confirmation-request. We use five labels in this category: Accept, Accept-part, Maybe, Reject, and Reject-part.

We mark an utterance with *Accept* or *Accept-part* if the speaker accepts all or part of the other speaker's proposal or request; or if the information or claim conveyed in an Assert is accepted or confirmed.

C: Okay well I can't do that unless the address is changed can I?
A: Yeah
A: Unfortunately no (22mar02-615.trs.txt)

C : d'après le tableau que j'ai c'est ça
A : tout à fait (am-d13.txt)
{C: if you refer to the document I have that's it
A: exactly}

We use the *Maybe* tag when the speaker is uncertain of an answer, or says "I'll have to think about it", "I'm not sure". We also use this tag when the person cannot answer the question or address the proposal or offer because of lack of knowledge: "I don't know".

We use Reject or Reject-part when the speaker disagrees, rejects a proposal or offer, says he will not comply, or says that all or part of the claim or the information conveyed by the other speaker is incorrect.

A: Morning customer services, could I take the account number?
C: I can't, err unfortunately I'm blind (26mar02-692.trs.txt)

C: mais qu'est-ce que j'en fais de ce relevé je pensais vous le retourner
A: non juste détruisez-le (am4-d8.txt)
{C: do I have to send you back the bank statement
A: no you can destroy it}

XDMLTool provides ComboBoxes on its user interface for annotators to select Functional (Dialogic) tags conveniently (See Figure 1). The *Comment* text field allows annotators to record further information, questions, or unusual aspects of an utterance. Each category provides an Uncertainty CheckBox, designated by a question mark (?), so that the annotator may record his uncertainty or ambivalence regarding the appropriateness of his tag selection.

The Style ComboBox has been used to annotate emotion behaviors such as Anxiety, Irritation . . ., observed during the conversation. Some experiments using the multi-level annotations such as dialogic and emotion tags carried out with the XDMLTool are reported in [5].

4 Semantic Annotation

Most transactions or *AccessFrames* are associated with details such as names, addresses and account numbers that may be organized into key-value pairs. We use the abstract categories *Attribute* and *Value* to hold these semantic details. The additional category *Modifier* (a descriptor intended to accompany "Attribute") allows us to shorten the attribute list.

The abstract nature of these categories means that this annotation scheme for the semantics of a dialogue is highly flexible and adaptable to dialogues in other domains. Without changing the top-level headings, we may substitute new lists of frames, attributes and modifiers as needed. Similarly, we can add entries to reflect new topics that are encountered in a large corpus, or refine or delete existing entries, without altering the top-level schema.

We have found it helpful to begin the annotation process with a set of labels developed during a preliminary mark-up. For large numbers of dialogues, it is necessary to allow the annotator to add to the lists and revise them as he or she comes across new information in the data. Different annotators will naturally create different names for the same frame or attribute; for example, ChangeAddress and UpdateAddress; VerifyId and VerifyCallerID, Card and NameofCard. Frequent checking among annotators is required to merge such overlapping labels and to ensure that the annotators are following the same conventions. Our goal has been to unify the lists as far as possible, while still leaving room for new labels necessitated by the data.

Figure 5 illustrates typical frames, along with their attributes and modifiers, encountered in call-center dialogues.

FRAME	ATTRIBUTES	MODIFIERS
GreetCaller	Name	Service Agent Customer
VerifyId	Name AcctNo PostCode Address BirthDate PhoneNo	First, Full, Last Old, Current Old, Current Home, Work
ChangeAddress	HouseNumber PostCode Address PhoneNo	New New New New
MakePayment	PaymtMethod AcctBalance AcctNo Amount	

Figure 5: Sample transaction frames, attributes and modifiers.

To annotate semantic information with XDMLTool, the user makes entries for a particular turn or turn segment in a semantic table on the user interface (Figure 1). One row is used for each distinct piece of information in the turn. Recommended choices for AccessFrame, Attribute and Modifier appear in ComboBoxes on the table. If necessary, the user may enter a new label by typing it into the appropriate ComboBox. For the Value column, text from the displayed dialogue may be copied into a Value cell.

At LIMSI a finer semantic annotation is being developed (a set of 60 concepts, 10 markers and 3 modes [+/-/?] have been defined). The goal is to use the manually labeled dialogs to bootstrap a statistically based semantic annotator [6].

5 Segmenting Dialogue Turns

XDMLTool allows the annotator to split a transcribed dialogue turn into separate segments or utterances, for more accurate annotation. The annotator may also reverse this operation and merge segments as needed. The AMITIÉS team recom-

mends that segmentation be done along functional/dialogic lines only. That is to say, if different parts of a turn consist of separate dialogic acts, meaning that they have different roles or purposes in the conversation, the turn should be split. Segmentation along semantic lines is not necessary, because many semantic details may be captured in XDMLTool's semantic table.

An example of proper segmentation is the following turn, where the greeting (Conventional → Opening) must be separated from the Influence-on-Listener → Action-directive role (actually, two Action-directives, which may be marked as separate AccessFrames on the semantic table):

C: Hi there | erm I've lost my card and also I've changed address (22mar02-508.trs.txt)

Sometimes a turn must be separated into more than two segments, if it consists of more than two discernible dialogic acts:

A: And who do you bank with please?
C: Erm the Bank of Scotland in Town
A: Yeah ok fine |
Erm I'll er take your new address in a minute, issue another, | well do you want a new card? (22mar02-508.trs.txt)

6 Building an Empirical Dialogue System

The dialogic and semantic tags for the AMITIÉS corpus were selected in such a way as to model an automatic dialogue system as close to human agents as possible. The AMITIÉS team has built components for a prototype triaging application for financial services. This demonstrator system is a multilingual, mixed-initiative application based on scenarios taken from the corpus of transcribed dialogues. It carries out simple conversations to identify the customer from a sample database, and to identify the task, or the reason the customer is calling, from a subset of tasks handled by the call centers. The dialogue manager was built to be language-independent, so that it can handle calls in French, German, and English, and even switch languages during a conversation. Once the annotation phase of the project is completed, we plan to build a dialogue transition graph derived from the annotated conversations, and extend the demonstrator so that it can complete specified tasks based on the graph. The dialogue system is intended to be deployed and evaluated in real operating call centers, and tested with real customers. Measures of success are customer satisfaction, efficiency and accuracy in performing the task. We will quantify characteristics such as the length of call, number of turns, and accuracy of system responses.

Acknowledgments

The authors from LIMSI would like to thank Hélène Bonneau-Maynard, Isabelle Wilhem and Celine Le Meur for helping to define the dialogic annotations and Isabel for writing the French annotation manual. The Albany authors would like to thank Habiba Ibrahim and Janet Dymond for their suggestions and their work on annotations. This paper is based on work supported in part by the European Commission under the 5th Framework IST/HLT Programme, and by the U.S. Defense Advanced Research Projects Agency.

References

[1] http://www.dcs.shef.ac.uk/nlp/amities/

[2] Allen, J. and Core, M. Draft of DAMSL: Dialog Act Markup in Several Layers. 1997. http://www.cs.rochester.edu/research/cisd/resources/damsl/

[3] Bunt, H. C. Dynamic Interpretation and Dialogue Theory. In M. Taylor, D. Bouwhuis, F. Neel, eds., The Structure of Multimodal Dialogue, volume 2. John Benjamins Publishing Company, Amsterdam, 1997.

[4] Barras, C., E. Geoffrois, Z. Wu, M. Liberman. Transcriber: development and use of a tool for assisting speech corpora production, Speech Communication, 33(1-2), pp. 5-22, January 2001.

[5] Devillers, L., S. Rosset, H. Maynard, L. Lamel. Annotations for Dynamic Diagnoses of the Dialog State, Proceedings of LREC, Las Palmas, 2002.

[6] Lefevre, F. and H. Bonneau-Maynard. Issues in the Development of a Stochastic Speech Understanding System, ICSLP 2002, Denver, September 2002.