Semantic and Dialogic Annotation for Automated Multilingual Customer Service

Hilda Hardy[1], *Kirk Baker*[2], *Hélène Bonneau-Maynard*[3],
Laurence Devillers[3], *Sophie Rosset*[3], *and Tomek Strzalkowski*[1]

[1]Institute for Informatics, Logics and Security Studies
University at Albany, State University of New York, Albany, NY 12222
[2]Dept. of Computer Science, Duke University, Durham, NC 27708
[3]LIMSI-CNRS, BP 133, 91403 Orsay cedex, France
hardyh,tomek@cs.albany.edu; kbaker3@email.unc.edu; helene.maynard,devil,rosset@limsi.fr

Abstract

One central goal of the AMITIÉS multilingual human-computer dialogue project is to create a dialogue management system capable of engaging the user in human-like conversation in a specific domain. To that end, we have developed new methods for the manual annotation of spoken dialogue transcriptions from European financial call centers. We have modified the DAMSL dialogic schema to create a dialogue act taxonomy appropriate for customer services. To capture the semantics, we use a domain-independent framework populated with domain-specific lists. We have designed a new flexible, platform-independent annotation tool, XDMLTool, and annotated several hundred dialogues in French and English. Inter-annotator agreement was moderate. We are using these data to design our dialogue system, and we hope that they will help us to derive appropriate dialogue strategies for novel situations.

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 developed theories of dialogue acts within what he called context change theory [1]. 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 (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 [2], as well as in practically oriented research prototypes: e.g., systems developed under the DARPA Communicator program [3], a complete dialogue transition graph is designed to guide the conversation and predict user responses. This is suitable for closed domains only. In the statistical approach, a transition graph is derived from a large body of annotated conversations. The best known dialogue annotation system, Dialogue Act Markup in Several Layers (DAMSL), is a system of functional dialogue acts and can conceivably be trained given a sufficiently large corpus and the right selection of features over which machine learning is done [4].

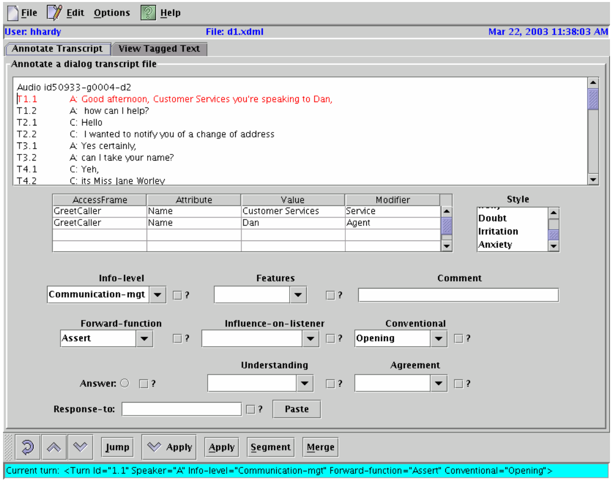
Nonetheless, DAMSL structure, even if learned, captures only the functional layer of the dialogue, whereas we are also interested in the semantic layer, that is, the information exchange and information building effects of a conversation. In order to properly understand a dialogue, both semantic and functional layers need to be considered. For instance, in call center dialogues much of the semantic layer can be represented as *transactions:* making payments, reporting problems, verifying status, etc. Furthermore, transactions have attributes, such as *AcctNo, Name, 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 real call center 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 Java software tool for annotating transcribed dialogues according to semantic, functional and stylistic characteristics. The expected input format is plain text dialogue files with turns labeled by speaker (in our case created with the Transcriber tool) [5]. Annotated dialogues are stored in XML format.

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 (Figure 1). Several timesaving features have been incorporated, such as automatic default tag selections, rapid navigation capabilities, and user-defined multi-label tagging. A related tool for querying annotated files, QXDMLTool, has been developed at the University of Sheffield. Both tools, as well as the AMITIÉS annotation manual, may be downloaded from the AMITIÉS website (http://www.dcs.shef.ac.uk/nlp/amities).



User: hhardy File: d1.xdml Mar 22, 2003 11:38:03 AM

Audio id50933-g0004-d2
T1.1 A: Good afternoon, Customer Services you're speaking to Dan,
T1.2 A: how can I help?
T2.1 C: Hello
T2.2 C: I wanted to notify you of a change of address
T3.1 A: Yes certainly,
T3.2 A: can I take your name?
T4.1 C: Yeh,
T4.2 C: its Miss Jane Worley

AccessFrame	Attribute	Value	Modifier
GreetCaller	Name	Customer Services	Service
GreetCaller	Name	Dan	Agent

Style: Doubt, Irritation, Anxiety

Info-level: Communication-mgt □ ?
Features: □ ?
Forward-function: Assert □ ?
Influence-on-listener: □ ?
Conventional: Opening □ ?
Understanding: □ ?
Agreement: □ ?
Response-to: □ ?

Current turn: <Turn Id="1.1" Speaker="A" Info-level="Communication-mgt" Forward-function="Assert" Conventional="Opening">

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

3. 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. 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. With a large corpus, it is necessary to allow the annotator to revise the lists according to new information. Frequent checking among annotators is required for consistency. Our goal is to unify the lists as far as possible, while still leaving room for new labels necessitated by the data. Table 1 illustrates typical frames, along with their attributes and modifiers, encountered in call-center dialogues.

We try to limit the number of AccessFrames to the most common tasks initiated by the agent (GreetCaller, VerifyId, TransferCall) and the customer (ChangeAddress, Make Payment, CloseAccount, etc.). We add new frames if topics arise that receive significant attention in the dialogue, or if they have distinct attributes and values.

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). Recommended choices for AccessFrame, Attribute and Modifier appear in ComboBoxes on the table. If necessary, the user may type in new labels. For Value, text from the displayed dialogue may be copied into a table cell.

For example, the following question and answer, part of a *VerifyId* AccessFrame, would be labeled with the Attributes *Name* (Modifier *Full*) and *PostCode*. The Values *John Smith* and *AB1 1CD* would be tagged for the answer. (A denotes agent; C denotes customer.)

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	

Table 1: Sample transaction frames, attributes and modifiers.

A: Your full name and postcode please?
C: Yes it's err John Smith AB1 1CD

A typical dialogue in the AMITIÉS corpus progresses with this general pattern of frames:

GreetCaller → VerifyId → Customer topic(s) → Closure

There may be more than one task discussed during a dialogue, though the majority of cases deal with only one.

Analyzing the semantic information collected from the annotated dialogues allows us to make observations useful for designing the dialogue system; for example, what attributes are most commonly discussed for the each topic, how many turns and how long can we expect the turns to be for a particular topic, and what is the general semantic progression for each topic.

4. 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 AMITIÉS corpus, we have found that, in general, the DAMSL tags work well [4]. For dialogic annotation, both the categories and the lists remain largely independent of the domain. However, we have made some adjustments to the tag set in order to reflect more accurately the features found in the corpus [6].

Our taxonomy follows the DAMSL categories Information-Level, Communicative Status, 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. Categories for Information-Level include Task, Task-management (System capabilities, Order of tasks, and Summary), Communication-management, and Out-of-topic. Communicative Status (features) are Self-talk, Third-party-talk, and Abandoned.

An utterance having a *Forward-Looking Function* (Figure 2) 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, committing himself to some course of action, or directing the other person to do something. Forward-looking functions typically elicit a response, in contrast to backward functions, which are primarily responses. 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. An utterance can be tagged with labels from both the forward and backward categories. For example, the backward function Answer is also labeled Assert.

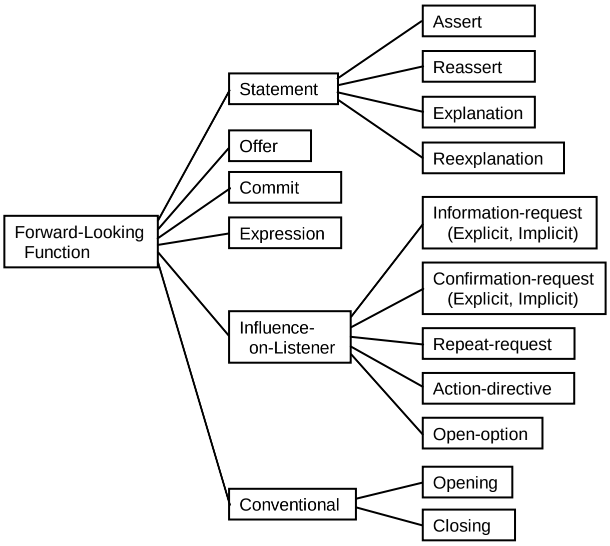


Figure 2: Hierarchy of annotation labels for Forward-Looking Function.

Utterances having a *Backward-Looking Function* (Figure 3) respond in some way to one or more previous turns in the dialogue, as, for example, an answer to a question. If the speaker signals some level of understanding or not understanding what the previous speaker has said, we use an Understanding tag. 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 an Agreement tag. Note that most, if not all, acceptances and rejections are also answers.

The Style ComboBox has been used to annotate emotion behaviors such as Anxiety, Irritation etc., observed during the conversation. Some experiments using the multi-level annotations such as dialogic and emotion tags carried out with the XDMLTool are reported by Devillers et al. [7].

5. Segmenting Dialogue Turns

XDMLTool allows the annotator to split a dialogue turn into separate segments or utterances (or merge them together), for more accurate annotation. The AMITIÉS team recommends that segmentation be done along dialogic lines only. That is to say, if different parts of a turn have different roles or purposes in the conversation, the turn should be split.

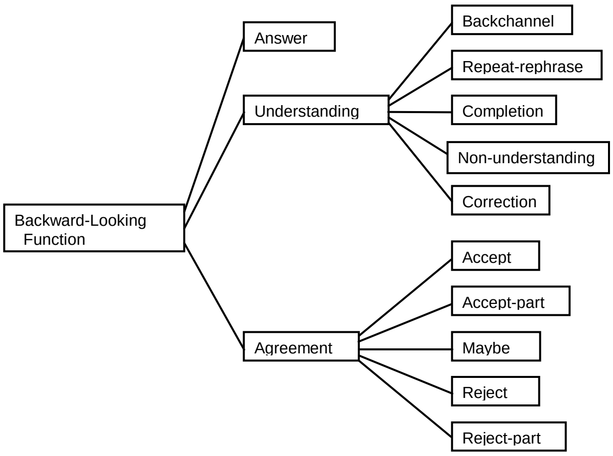


Figure 3: Hierarchy of annotation labels for Backward-Looking Function.

An example of proper segmentation is the following turn, where the greeting (Opening) must be separated from the 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

Sometimes a turn must be separated into more than two segments, if it consists of more than two discernible dialogic acts. Here we have Backchannel, Commit, and Confirmation-request-explicit in the third turn:

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?

The following portion of an annotated dialogue illustrates attribute negotiation and transition from one task to another (VerifyId to ChangeAddress):

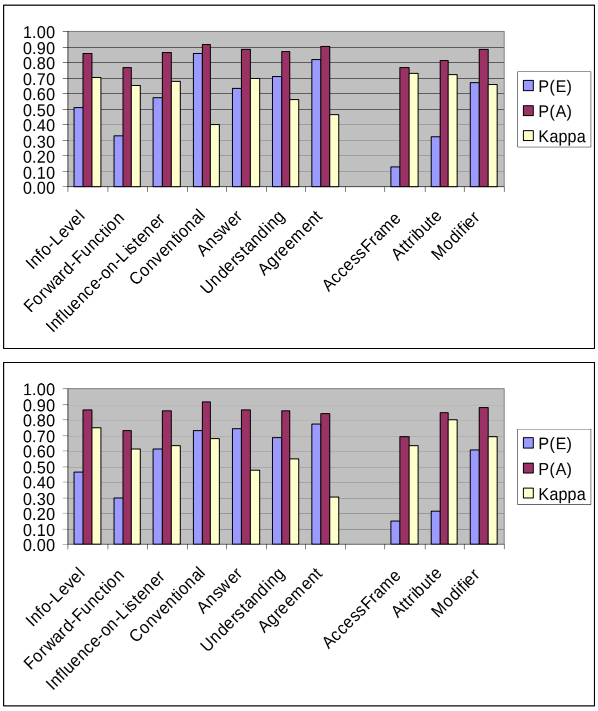
```
<Turn Id="9.2" Speaker="A" Info-level="Task" Influence-on-
  listener="Info-request-explicit">
  <SemanticUnit Id="9.2.1" AccessFrame="VerifyId"
    Att="BirthDate" />
  A: would you care? Can you confirm your date of
  birth?
</Turn>
<Turn Id="10.1" Speaker="C" Info-level="Task" Forward-
  function="Assert" Answer="true" Response-to="T9.2">
  <SemanticUnit Id="10.1.1" AccessFrame="VerifyId"
    Att="BirthDate" Val="11-11-11" />
  C: 11-11-11
</Turn>
<Turn Id="11.1" Speaker="A" Info-level="Communication-
  mgt" Forward-function="Expression">
  <SemanticUnit Id="11.1.1" AccessFrame="VerifyId" />
  A: Thank you,
</Turn>
<Turn Id="11.2" Speaker="A" Info-level="Task" Influence-on-
  listener="Info-request-explicit">
  <SemanticUnit Id="11.2.1" AccessFrame="Change
    Address" Att="HouseNumber" Modifier="New" />
  A: and your new house number?
</Turn>
<Turn Id="12.1" Speaker="C" Info-level="Task" Forward-
  function="Assert" Answer="true" Response-to="T11.2">
  <SemanticUnit Id="12.1.1" AccessFrame="Change
    Address" Att="HouseNumber" Val="10"
    Modifier="New" />
  C: 10
</Turn>
```

6. Inter-Annotator Agreement

345 English dialogues and 60 French dialogues from our corpus were annotated by two annotators. Inter-annotator agreement was measured according to the kappa statistic [8]:

$$K=\frac{P(A)-P(E)}{1-P(E)}, \tag{1}$$

where *P(A)* is the proportion of times that the annotators agree and *P(E)* is the proportion of times that we would expect the annotators to agree by chance. If there is complete agreement among the raters, then K = 1; whereas if there is no agreement other than would be expected to occur by chance, then K = 0. Kappa values were found for the most part to be moderate in both the dialogic and the semantic categories. Agreement for our new semantic categories was comparable to the highest rates of agreement for the dialogic categories in both languages, despite differences in lists and numbers of tags (Figures 4 and 5).



Figures 4 and 5: Inter-annotator agreement for English (top) and French dialogues.

The Conventional category in the English dialogues was found to have the lowest Kappa value (0.4) but the highest rate of actual agreement (P(A) = 0.9139; French P(A) = 0.9130). We may note that proportions of tags play a significant role in these calculations. In the French corpus, the Opening and Closing tags were selected 15% of the time; whereas these tags were chosen only 7% of the time in the English corpus, yielding a higher rate of expected chance agreement (P(E) = 0.857) and a corresponding lower Kappa value.

Concerning the Answer and Agreement categories, the differences between the two French annotations is mainly due to different strategies of accounting for implicit information. For the Answer class, one of the French annotators made this selection only for explicit information requests, while the other one used the Answer tag for both implicit and explicit information requests. In the English data, implicit info-requests comprise a smaller percentage of all info-requests (1.4%, 61/4211) than in the French data (10%, 48/479).

Concerning the Agreement class, the difference between the two French annotations is due most of the time to the Backchannel tag (typically a short phrase such as “yes” or “uh-huh” meaning “I heard you”). If the phrase is also an acceptance, it should be annotated at both levels. During the French annotation, the annotators listen to the signal. One annotator used an Agreement tag in case of acceptance while the other one used Answer to count also as an acceptance.

In order to improve inter-annotator agreement, new rules will be added to the Dialogic Annotations Manual.

7. Conclusion

The new annotation scheme described here reflects our approach to dialogue design. By separating the functional and semantic levels, but annotating them together, we can more easily apply our tools and methods to new corpora and new domains. We can also study complementary functional and semantic structures, and exploit patterns discovered to improve components of our automatic system. We hope the data will help us to automatically derive appropriate strategies for novel interactive situations.

8. Acknowledgements

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.

9. References

[1] Bunt, H. C., “Dynamic Interpretation and Dialogue Theory”, in Taylor, M., Bouwhuis, D., Neel, F., eds., *The Structure of Multimodal Dialogue,* Vol. 2, John Benjamins Publishing Company, Amsterdam, 1997.

[2] Nuance, http://www.nuance.com.

[3] DARPA, http://www.darpa.mil/iao/Communicator.htm.

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

[5] Barras, C., et al., “Transcriber: development and use of a tool for assisting speech corpora production”, *Speech Communication,* 33(1-2): 5-22, 2001.

[6] Hardy, H., et al., “Multi-layer dialogue annotation for automated multilingual customer service”, in *Proceedings of the ISLE Workshop on Dialogue Tagging,* Edinburgh, Scotland, 2002.

[7] Devillers, L., et al., “Annotations for Dynamic Diagnosis of the Dialog State”, in *Proceedings of LREC,* Las Palmas, 2002.

[8] Carletta, J., “Assessing agreement on classification tasks: the kappa statistic”, *Computational Linguistics,* 22 (2):249-254, 1996.