



UNIVERSITY
OF TRENTO

DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

38050 Povo – Trento (Italy), Via Sommarive 14
<http://www.disi.unitn.it>

AN OPEN-DOMAIN DIALOG ACT TAXONOMY

Silvia Quarteroni, Giuseppe Riccardi, Sebastian Varges and Arianna Bisazza

August 2008

Technical Report # DISI-08-032

AN OPEN-DOMAIN DIALOG ACT TAXONOMY

Silvia Quarteroni, Giuseppe Riccardi, Sebastian Vargas and Arianna Bisazza
silviaq@disi.unitn.it, riccardi@dit.unitn.it, vargas@dit.unitn.it, bisazza@disi.unitn.it

Dipartimento di Ingegneria e Scienza dell'Informazione

Università di Trento

Via Sommarive 14 - Povo (TN), Italy

ABSTRACT

This document defines the taxonomy of dialog acts that are necessary to encode domain-independent dialog moves in the context of a task-oriented, open-domain dialog.

Such taxonomy is formulated to satisfy two complementary requirements: on the one hand, domain independence, i.e. the power to cover all the range of possible interactions in any type of conversation (particularly conversation oriented to the performance of tasks). On the other hand, the ability to instantiate a concrete set of tasks as defined by a specific knowledge base (such as an ontology of domain concepts and actions) and within a particular language.

For the modeling of dialog acts, inspiration is taken from several well-known dialog annotation schemes, such as DAMSL (Core & Allen, 1997), TRAINS (Traum, 1996) and VERBMOBIL (Alexandersson et al., 1997).

INTRODUCTION

In the context of the EU FP6 project ADAMACH (ADaptive meaning MACHines, EU contract no. 022593), we are designing an adaptive multi-modal spoken dialog system. One of the fundamental steps in such design is the formulation of a taxonomy of dialog acts which constitute the basic units of interaction of the dialog manager.

In order to guide the creation of the ADAMACH dialog act taxonomy, the following requirements have been formulated:

- The dialog acts must cover the range of basic dialog moves that can take place in open-domain, task-oriented dialog;
- They must be sufficient to cover the range of conversational situations taking place in both human-human and human-computer dialog;
- They must be domain independent, i.e. valid for any type of task-oriented dialog;
- They must be used as the dialog moves encoded by the Dialog Move Engine and to annotate dialogs during training. This means they must have an appropriate level of specificity but must also guarantee that annotation and data-driven classification accuracy will not suffer;

Two additional requirements for the future of the ADAMACH project are:

- It must be possible to extend and complete the initial set of speech acts with domain-dependent speech acts;
- There must be speech acts accounting for multi-modal dialog events.

RELATED WORK

In line with most modern theories of conversational analysis (Sinclair & Coulthard, 1975), human-computer dialog is represented in this document as a set of exchanges composed by speech acts (also called dialog acts).

Based on speech act theory, several initiatives have produced taxonomies representing the utterances occurring in task-oriented dialog systems. These include:

- The DAMSL generic annotation scheme (Core & Allen, 1997), based on three types of dialog functions:

1. *forward* communicative functions, proposing something to the interlocutor (“directive”, “offer”, “commit”, . . .),
2. *backward* communicative functions, relating to previous utterances (“accept”, “acknowledge”, “answer”, “re-phrase”, . . .),
3. *utterance-level* features (“task management”, “conventional form”, . . .);

An adaptation of the DAMSL scheme is the SWBD-DAMSL scheme used in the annotation of the SWITCHBOARD human-human spontaneous (i.e. task-free) telephone conversation corpus (Stolcke *et al.*, 1998);

- The HCRC annotation scheme (Kowtko & Isard, 1993; Anderson & Bader, 1991).

This was designed for a cooperative application, where the goal was that an instruction giver (having a path on his/her map) would help an instruction follower (having the map only) to reconstruct a path on a given map.

The scheme is based on 12 main dialog moves, such as “instruct”, “clarify”, “query”, “acknowledge”, “reply”, and “check”.

- The LINDA/LINLIN scheme (Dahlbaeck & Jonsson, 1998), developed for annotating information retrieval dialogs. The taxonomy involves:

1. *initiative* moves such as “question” and “update”;
2. *response* moves such as “answer”;
3. *dialog management* moves, such as “greeting”, “farewell” and “discourse continuation”.

- The VERBMOBIL scheme (Alexandersson *et al.*, 1997) was developed for the translation of spontaneous speech-to-face dialogs. The annotation scheme contains 45 different illocutionary acts grouped in three main sets:

1. the acts aiming at *dialog control* (“greet”, “bye”, “thank”, . . .),
2. the acts aiming at *task management* (“init”, “defer”, “close”),
3. the acts aiming at *task promotion* (“request”, “suggest”, “inform”, “feedback”, . . .).

- The TRAINS conversation act typology (Traum, 1996) distinguishes between four types:

1. *turn-taking* acts (“take-turn”, “release-turn”, . . .),
2. *grounding* acts (“ack”, “repair”, . . .),
3. *core speech* acts (“inform”, “request”, . . .),
4. *argumentation* acts (“clarify”, . . .).

While the level of granularity and the range of dialog moves of most of the above schemes was determined by the application of the dialog system, as pointed out in Larsson (1998) there are three main groups of generic dialog moves, namely¹:

1. *Core* moves (TRAINS): these are the moves representing the core actions performed of the dialog, e.g. requesting and providing information, or executing a task. They include initiatives (i.e. forward-looking acts) and responses (backward-looking acts);
2. *Conventional* (DAMSL) or *discourse management* (LINLIN) moves: these are the moves that “glue” the dialog together and delimit specific phases of it, such as opening, continuation, closing, and apologizing;
3. *Feedback* (VERBMOBIL) or *grounding* (TRAINS) moves: these are used to elicit and provide feedback in order to establish or restore a common ground in the conversation.

¹ See also http://www.ling.gu.se/~sl/sdime/sdime_type.html

Another notable dialog act tagging scheme is the DATE (Walker and Passonneau, 2001) scheme for spoken dialog systems, used for the DARPA COMMUNICATOR project. Here, the speech act tagset includes 10 acts, most of which are also present in the taxonomies analyzed by Larsson (1998): *request-info*, *present-info*, *offer*, *acknowledgement*, *status-report*, *explicit-confirm*, *implicit-confirm*, *instruction*, *apology* and *openings/closings*.

On the grounds of the above-mentioned previous work, we propose in the following section a dialog act taxonomy for the ADAMACH project. Notice that we adopt the term “dialog act” instead of “speech act” in order to model a potentially multi-modal dialog setting where acts are achieved not only via speech but also via other channels such as GUI interactions.

ADAMACH DIALOG ACT TAXONOMY

The ADAMACH dialog act taxonomy is summarized in Table 3.1. Here, three sub-tables distinguish between core dialog acts, conventional ones and feedback dialog acts. For each tag in the table, a brief description is reported as well as one or more examples taken from the LUNA corpus of human-human conversation collected by CSI Piemonte (in Italian).

Such dialogs have been collected for the EU project LUNA (<http://www.ist-luna.eu>, contract no. 33549), with the aim of providing a multi-layer spoken dialog annotation (see Raymond et al., 2007).

Indeed, a subset of the LUNA corpus has been annotated using the ADAMACH taxonomy in order to highlight interesting dialogic phenomena and relations between the dialog act annotation level and other semantic levels of dialog interpretation. The latter levels include ontology and predicate-argument annotation; for details, please refer to (Bisazza et al, 2008).

Tag	Description	Example
<i>Core dialog acts</i>		
Info-request	The speaker wants the addressee to provide him with some information	cosa è la procedura quella nuova allora mi lasci solo per favore il censimento del computer già verificato il collegamento del cavo dell'alimentazione del monitor <i>(according to prosody and context can also be tagged as Inform)</i>
Action-request	The speaker wants the addressee to perform an action	provi collegare un'altra presa
Yes-answer	Affirmative answer	si si certamente si ho provato
No-answer	Negative answer	no no non ci sono
Answer	Other kinds of answer	non lo sappiamo quattro tre due cinque cinque sei tre
Offer	The speaker offers or commits himself to perform an action	ti lascio il numero di remedy te lo segnalo urgente attivo subito
ReportOnAction	The speaker notifies that an action is being or has been performed	te lo segnalo urgente
Inform	The speaker provides the addressee with some information which wasn't explicitly required (by means of an Info-request)	la richiesta è la numero uno zero zero comunque è abbastanza urgente
<i>Conventional dialog acts</i>		
Greet	Conversation opening	salve salve
Quit	Conversation closing	buona giornata ciao ciao
Apology	Apology	mi scusi non saprei dire

Thank	Thanking and optional down-playing	grazie di niente si figuri
Feedback/Turn management dialog acts		
ClarificationRequest	The speaker wants to make sure that he understood the previous utterance by repeating/rephrase it and asking for confirmation, or by asking the addressee to repeat/reformulate it.	allora non non compare proprio la voce e mi hai detto che sei chesta marco giusto
Ack	The speaker express his agreement with the previous utterance, or simply provides feedback by showing that he understood what the addressee said	si okay perfetto
Filler	Utterance whose main goal is to manage conversational time (i.e. a speaker’s taking time while keeping the turn)	si allora un attimino solo allora allora
Non interpretable/Non classifiable		
Other	Default tag for non-interpretable and non-classifiable utterances	spero che comunque non riesce a fare nulla col senza monitor per cui pensavo fosse una cosa che si potesse fare subito

Table 3.1. ADAMACH dialog act taxonomy

As the ADAMACH tag set has been designed to be domain-independent and generic, domain-specific information must be encoded at a different level.

This is why we propose the definition of domain-dependent arguments (or parameters) to represent this type of information. In Section 7, we present possible implementations of such arguments and explain how these can render dialog-manager specific tasks.

Sections 3.1—3.4 discuss the dialog acts in Table 3.1. in more detail.

3.1. Core dialog acts

The core dialog acts are represented by **Info-request**, **Action-request**, **Yes-answer**, **No-answer**, **Answer**, **Offer**, **Inform** and **ReportOnAction**.

The main acts used for requests are **Info-request** and **Action-request**, representing a request for information and to execute an action, respectively.

The acts used as responses to **request** acts include **Yes-answer**, **No-answer** and **Answer**.

Symmetrically, **ReportOnAction** is used to notify that an **Action-request** is being carried on/has been terminated.

Moreover, **Offer** represents both a proposal to perform a generic action, such as conversation follow-up proposal (e.g. “Is there any other way I can help you?”), and an offer to carry on a specific action. The latter is possibly an alternative action to the one requested, amending for instance not being able to perform the action requested by the user. (“Can I give you the email of the person you are looking for instead of the phone number?”). In this case, the **Offer** act can be followed by an answer act (e.g. **Yes-answer**).

Finally, the **Inform** dialog act represents the fact that an item of information is provided to the interlocutor. This act type is used to annotate utterances where useful information for building a common ground is given. Examples of **Inform** acts could be: “I have a problem with one of your devices” in a customer service application, or “I don’t want to fly via London” in a travel planning application, or “By the way, remember that there are strikes on the French railways next Friday”.

3.2. Conventional dialog acts

Amongst conventional/discourse management dialog acts, we find the **Greet** and **Quit** acts that signal the opening and closing of the conversation, as well as **Thank** following conventional models of conversation.

An additional conventional dialog acts is **Apology**, useful to replace answer and **ReportOnAction** acts in case the speaker is unable to provide information or execute an action.

3.3. Feedback dialog acts

Amongst feedback dialog acts, **ClarificationRequest** is the act representing a request for clarification or confirmation. The clarification being sought can be:

1. a clarification about a preceding utterance (using expressions such as “Sorry, I don’t understand what you just said. Can you please reformulate?”).
2. a clarification about the value of a specific object, demanding explicit or implicit confirmation about the value of such object (e.g. “You want to subscribe to Calculus, confirm?” versus “Subscribing you to Calculus then.”) or
3. a choice between objects (“Do you want to subscribe to Calculus A or Calculus B?”).

A common feedback move to several taxonomies in the literature is **Ack**, with which the previous utterance pronounced by the interlocutor is acknowledged (e.g. “All right”, “OK”, etc.).

Finally, the **Filler** dialog act represents utterances that have a purely accessory function, “gluing” the conversation together without providing any information or content. An example of **Filler** utterance would be “So...” or “Let’s see”.

3.4 Non interpretable/Non classifiable moves

The **Other** act is used to annotate a dialog move that cannot be interpreted or classified according to any of the previous items in the tag-set.

DISCUSSION

The three-way partition of the ADAMACH dialog act taxonomy into core, conventional and feedback dialog acts is modeled on the partition proposed by Larsson (1998) to summarize a number of well-known state-of-the-art taxonomies.

The directionality of dialog acts, represented in the DAMSL taxonomy by the forward-looking vs backward-looking definition, is not used as a primary criterion to classify acts in ADAMACH as there are cases where directionality does not seem to apply; however, Tables 1-3 report the forward/backward directionality of acts when applicable as an additional information.

Similarly to the HCRC taxonomy and in contrast with larger taxonomies e.g. DAMSL and VERBMOBIL, the ADAMACH taxonomy is structured around a dozen dialog acts; however, the specification of arguments such as request type allows a broad coverage of dialog situations.

IMPLEMENTATION

The current dialog annotation scheme is meant to be applied during two phases of dialog manager development:

1. The generation of the following utterance by the dialog manager, which must be fully specified by the above dialog act definitions. This means that it may be necessary for a dialog act to be defined by several arguments (e.g. the possible types of questions and answers recognized by the system, such as yes/no questions, *wh*-questions, etc.).
2. The manual annotation phase carried out once a corpus of dialogs is available, for the purpose of evaluation or learning. This requires the acts to be easily annotated, hence the annotation to be as “flat” as possible, relieving annotators from the burden of selecting many different arguments for a given act.
3. The automatic annotation phase performed by the dialog system itself to evaluate its own performance without human supervision.

FUTURE WORK AND EXTENSIONS

The current dialog act taxonomy must not be taken as definitive: indeed, there are a number of aspects according to which it can be further specified and completed.

6.1 Domain-dependent moves

First, the domain independent dialog acts formalized in the current taxonomy may be completed by domain-dependent ones in order to be better suited to the individual applications developed within the project.

One way to create domain-dependent moves could be the introduction of new speech acts specific to a given application, such as **call(person)** in the context of an automatic directory helpdesk.

This solution has the advantage of being “flat” and therefore potentially easy to implement, however it appears conceptually inappropriate considering the domain-independence requirements behind the design of the ADAMACH taxonomy.

An alternative approach would be the introduction of **domain-dependent arguments** (i.e. parameters) for domain-independent moves. An example of this solution is e.g. **Info-request**(*examMark(studentID,examName)*), where **Info-request** is a domain-independent dialog act and *examMark(studentID,examName)* is an argument specific to the university helpdesk application. This approach has the advantage of maintaining the domain independence of the speech acts, leaving the domain-level dependence to the arguments. Previous work along the same lines is e.g. (Gupta et al., 2006).

The obvious way to implement this is by representing the domain-dependent actions and concepts in the form of domain ontologies.

For instance, the university helpdesk application would have a concept ontology made of the concepts *Student* and *Exam*, having attributes such as *studentID* and *examName*, respectively. Such concepts would be used to perform tasks such as *examMark(studentID,examName)*, which in turn would be specified by actions such as *askStudentID()* or *lookUpExamInfo(studentID,examName)*, defined in an action ontology.

In the following paragraphs, we show how some of the dialog acts in Table 3.1 can be parameterized by task-specific arguments.

Core dialog acts

Info-request and **Action-request**, representing a request for information and to execute an action, respectively, can be parameterized by the information/action that is being requested.

The acts used as responses to **request** acts, i.e. **Yes-answer**, **No-answer** and **Answer**, can be parameterized by the answer **a** returned by the interlocutor. The nature of **a** depends on the application for which the system is intended. For instance, in a QA application, **Info-request** might be parameterized by **q**: “When was Shakespeare born?”, and the **a** argument of its corresponding **Answer** act might be: “In 1564”. In a university helpdesk application, **q** might be: *examMark(studentID,examName)* - e.g. *examMark(“123456”, “Calculus”)* - and **a** might be: “B”.

Symmetrically, **ReportOnAction** is used to notify that an **Action-request** is being carried on/has been terminated. Hence, this act can be parameterized by the action **a** in question. Most likely, in a human-computer dialog system, the user is the action- requesting interlocutor and the system is the action-executing one. This dialog act may not be needed and implemented in a purely information seeking application such as a QA system; in a university helpdesk application, **a** could be

subscribe(studentID,examName) - e.g. *subscribe*("123456", "Calculus") - or *call(person)* - e.g. *call*("John Smith").

Moreover, **Offer** represents both a proposal to perform a generic action, such as conversation follow-up proposal (e.g. "Is there any other way I can help you?"), and an offer to carry on a specific action, that becomes the dialog act parameter (**a**). In the former case, a "dummy" action *_action* can be used in place of **a**. The latter is possibly an alternative action to the one requested, amending for instance not being able to perform the action requested by the user. ("Can I give you the email of the person you are looking for instead of the phone number?"). In this case, the **Offer** act can be followed by an answer act (e.g. **Yes-answer**).

Finally, the **Inform** dialog act can be parameterized by the item of information **u** provided to the interlocutor.

Feedback dialog acts

Amongst feedback dialog acts **ClarificationRequest** can be specified by an argument **t** representing the type of clarification being sought. The latter can be:

1. *utterance*: a clarification about a preceding utterance (using expressions such as "Sorry, I don't understand what you just said. Can you please reformulate?").
2. *value*: a clarification about the value of a specific object, demanding explicit or implicit confirmation about the value of such object (e.g. "You want to subscribe to Calculus, confirm?" versus "Subscribing you to Calculus then.") or
3. *choice*: a choice between objects ("Do you want to subscribe to Calculus A or Calculus B?").

6.2 Multimodality

Another aspect which deserves investigation is the fact that no specific dialog acts are currently defined for multi-modal interaction: the current approach is to implement the same dialog acts as for spoken dialog for the purpose of "visual" or multi-modal dialog.

However, it may be necessary in the future to extend such set of moves using modality-specific dialog acts that do not apply to e.g. speech.

An approach towards the solution of this issue is the specification of a modality argument to extend the current specification of dialog acts. For instance, the **Answer** dialog act in a multi-modal setting would be parameterized by an (additional) argument **m** for modality (taking values such as *mouse-click*, *speech*, *text*, etc.) to specify in which format the answer has been provided.

6.3 Task-information level annotation

Along with the tag set in Table 3.1, we propose for future development an experimental level of annotation, relating to the task at hand, called task-information level.

The task-information level is a **domain-independent** level of abstraction that represents the function of a group of dialog acts within a task-oriented dialog system. Annotating a dialog in terms of task information provides a segmentation that is potentially useful to distinguish the former's task-addressing portions from conversational parts. In this, the task information level is inspired by the information-level in the DAMSL multi-layer annotation scheme:

"The Information-Level annotation provides an abstract characterization of the utterance. In task-oriented dialogs, we can roughly divide utterances into those that address the task in some way, those that address the communication process (Communication-management), and those that do not fall neatly into either category (Other-level). In addition, we can subdivide the first category into utterances that advance the task (Task) and those that discuss the problem solving or process or experimental scenario (Task-management)." (Core & Allen, 1997)

Keeping in mind that this level of annotation is meant to be domain-independent, we define the task information level tags summarized in Table 6.1.

Tag	Description	Example
<i>Task-information level</i>		
Communication	Utterances only addressing the communication management process, not conveying any useful information to the task solving process	ciao ciao buona giornata
		va bene
Task-instantiation	Speakers try to define the problem, hence instantiating the task to be performed cooperatively during the dialog	e ciao Simone sono una collega del di Corso Tazzoli
Task-resolution-information	Speakers collaborate to address the task at hand by exchanging information	mi dai anche il censimento della tua macchina per cortesia
Task-resolution-action	Speakers collaborate to address the task at hand by performing some action	è sì infatti il io avevo già provato a riavviarlo
Task-management	Speakers discuss about the way the problem may be solved in the future	perchè non è detto eventualmente si apre una chiamata all hardware e si manda un tecnico facendosi sostituire la tastiera

Table 6.1. ADAMACH dialog act taxonomy: task-information level

Hence, each utterance is annotated not only with the dialog act it carries out, but also with either of the following tags:

- **Communication:** these are utterances that only address the communication management process and do not convey any useful information to the task solving process;
- **Task-instantiation:** caller and/or operator try to define the problem, hence instantiating the task to be performed cooperatively during the dialog.
Includes dialog-opening offers of help uttered by the operator (e.g. “How may I help you?”);
- **Task-resolution:** caller and/or operator collaborate to solve the current problem/address the task at hand. This is further specified into two categories:
 - **Task-resolution-information**, when the purpose of the corresponding dialog acts is the exchange of information (e.g. providing information about an exam, obtaining a delivery address, etc.);
 - **Task-resolution-action**, when the purpose of the corresponding dialog acts is the performing of actions (e.g. subscribing to an exam, turning off a device, etc.)
- **Task-management:** caller and operator discuss about the way the problem may be solved in the future, e.g. by assessing a future call by the operator or evaluating the user’s satisfaction about the service supplied by the system.

Furthermore, during task-information annotation, we propose to add a binary attribute **position** having values **B** or **I** (recalling the I-O-B convention) to signal whether the current tag is the beginning resp. the continuation of the corresponding task-information. For instance, **Task-instantiation-B** signals that the current utterance is the beginning point of a task-instantiation, while all of the following utterances in charge of instantiating the same task will be labeled as **Task-instantiation-I**. If this generic frame proves to be applicable to a various range of problem solving dialogs, this annotation level may represent a junction point between the generic dialog acts and the domain dependent task level annotation, where requests and completions of specific tasks to a given application are annotated.

However, a caveat is that “without knowing exactly which activities occur while doing the task, it can be hard, though, to know which utterances advance the task” (Core & Allen, 1997).

REFERENCES

- Alexandersson, J., Reithinger, N., & Maier, E. 1997. Insights into the dialogue processing of VERBMOBIL. Tech. rept. 191. Saarbrücken, Germany.
- Bunt, H. 2006. Dimensions in Dialogue Act Annotation.
- Core, M. G., & Allen, J. F. 1997. Coding Dialogs with the DAMSL Annotation Scheme. In: Proceedings of the 1997 AAAI Fall Symposium on Communicative Actions in Humans and Machines.
- Dahlbaeck, N., & Jonsson, A. 1998. A coding manual for the Linköping dialogue model.
- Larsson, S. 2002. Issue-based dialogue management. Ph.D. thesis, Göteborg, Sweden.
- Larsson, S., & Traum, D. R. 2000. Information state and dialogue management in the TRINDI dialogue move engine toolkit. *Journal of Natural Language Engineering*, 6(3-4), 323–340.
- Larsson, S., Ljunglöf, P., Cooper, R., Engdahl, E., & Ericsson, S. 2000. GoDiS—An Accommodating Dialogue System. Pages 7–10 of: Sidner, C. (ed.), *ANLP/NAACL Workshop on Conversational Systems*. Somerset, New Jersey: ACL.
- Larsson, Staffan. 1998. Using a Type Hierarchy to Characterize Reliability of Coding Schemas for Dialogue Moves. Unpublished document.
- Sinclair, J. M., & Coulthard, R. M. 1975. *Towards an Analysis of Discourse: The English Used by Teachers and Pupils*. Oxford University Press.
- Traum, D. 1996. Dialogue Management in Conversational agency: The TRAINS-93 dialogue manager. Pages 1–11 of: *Proceedings of the Twente Workshop on Language Technology: Natural Language Systems (TWLT 11)*.
- Stolcke, A, Shriberg, E, Bates, R, Coccaro, N., Jurafsky, D., Martin, R. et al. 1998. Dialog Act Modeling for Conversational Speech.
- Walker, M. and Passonneau, R. 2001. DATE: A Dialogue Act Tagging Scheme for Evaluation of Spoken Dialogue Systems. *Proceedings of HLT'01*.
- Gupta, N.; Tur, G.; Hakkani-Tur, D.; Bangalore, S.; Riccardi, G.; Gilbert, M. 2006. The AT&T spoken language understanding system, *Audio, Speech, and Language Processing, IEEE Transactions on* [see also *Speech and Audio Processing, IEEE Transactions on*], vol.14, no.1, pp. 213-222.
- Varges, S., Riccardi, G. and Quarteroni, S. Persistent Information State in a Data-Centric Architecture. *Proceedings of SigDial'08, Columbus, OH, USA, June 2008*.
- Raymond, C., Riccardi, G., Rodriguez, K. and Wisniewska, J. The LUNA Corpus: an Annotation Scheme for a Multi-domain Multi-lingual Dialogue Corpus. In *Proceedings of Decalog, Rovereto, Italy, 2007*.
- Bisazza, A., Dinarelli, M., Quarteroni, S., Tonelli, S., Moschitti, A. and Riccardi, G., Semantic annotations for conversational speech: from speech transcriptions to predicate-argument structures. Submitted to: *SLT'08, Goa, India, 2008*.

ANNEX A: SWBD-DAMSL AND ADAMACH TAG SETS COMPARED

We here present a parallelism between the well-known SWITCHBOARD-DAMSL dialog act taxonomy and the taxonomy defined in this document for the ADAMACH project (Table A.1). The level of granularity of the former taxonomies is noticeably finer than in the ADAMACH taxonomy. Our choice is motivated by two main considerations: first, we want to alleviate the burden (and low precision) of annotators by reducing the number of tags. Moreover, we rely on the presence of tag arguments for more specific (and indeed also task-specific) dialog act annotation.

SWBD-DAMSL	ADAMACH
<i>Communicative-Status</i>	
Uninterpretable	Other
Non-verbal	
Abandoned	Other
Self-talk	Other
3rd-party-talk	Other
<i>Information-level</i>	
Task	Task-instantiation, Task-resolution-action, Task-resolution-information
Task-management	Task-management
Communication-management	Communication
Other	NOT CURRENTLY MARKED
<i>Forward-Communicative-Function</i>	
Statement	
Assert	Inform, ReportOnAction
Reassert	Inform
Statement-non-opinion	Inform
Statement-opinion	Inform
Influencing-addressee-future-action	
Open-option	Inform
Directive	
Info-request	Info-request
Yes-No-question	Info-request, Action-request, Offer, clarificationRequest
Wh-Question	Info-request, Action-request, Offer, clarificationRequest
Open-Question	Info-request, Action-request, Offer, clarificationRequest
Or-Question	Info-request, Action-request, Offer, clarificationRequest
Or-Clause	Info-request, Action-request, Offer, clarificationRequest

Declarative-Question	Info-request, Action-request, Offer, clarificationRequest
Tag-Question	Info-request, Action-request, Offer, clarificationRequest
Action-directive	Action-request
Committing-speaker-future-action	
Offer	Offer
Commit	Offer
Other-forward-function	
Conventional-opening	Greet
Conventional-closing	Quit
Explicit-performative	NOT CURRENTLY ANNOTATED
Exclamation	
Other-forward-function	
Thanking	Thanking
You're-Welcome	Thanking
Apology	Apology

Table A.1. Parallelism between the SWBD-DAMSL dialog act tagset and the ADAMACH tagset. The tags that only appear in the SWBD tagset and not in the DAMSL one are reported in boldface.

ANNEX B. ANNOTATION MANUAL

This annex provides guidelines to annotate speech corpora using the ADAMACH dialog act tag set.

ANNOTATION UNIT

Dialog act annotation is performed on speech transcriptions that have been segmented into turns. In human-machine dialogs, turn boundaries are easily detectable as turn management is fully determined by the dialog system and each system turn is normally followed by a user turn (even if the latter can optionally be empty).

This is not the case for human-human dialogs, where turn management patterns are far more complex. Even if silences are a good cue to find turn boundaries, this kind of dialog needs to be segmented by human labelers. When referring to the units resulting from manual transcription of human-human dialogs, the term *segment* is preferable to *turn*.

Turns/Segments (speech units deriving from the transcription of the audio file) and *utterances* (semantic units corresponding to one dialogue act) do not necessarily correspond. That is why turns often include several dialog acts. This problem arises much more frequently in human-human dialogs than in human-machine ones.

Therefore the task of the labeler also includes segmenting the dialog transcription into utterances. An utterance is a semantic unit aiming at achieving one dialog act. If more than one tag seems suitable to represent an utterance, the labeler should choose the tag corresponding to its main function. In case of doubt among several tags, priority should be given to tags belonging to the core dialog acts set. For instance in the following example, the utterance corresponding to turn 100001 is tagged as **Inform** although it also has a conventional function (**Greet**).

The labeler is nevertheless allowed to split the turn into several utterances or to merge several turns into one utterance when this is needed. See for example turn 100002, which has been split into three **Inform** dialog acts, each aiming at giving a different piece of information to the addressee.

TurnID Spk	Turn	Utterances	Dialog act
100001 operator	Eldes buongiorno sono Simone	Eldes buongiorno sono Simone	Inform
100002 caller	e ciao Simone sono una collega del di Corso Tazzoli e ho un problema con la tastiera del mio computer che improvvisamente si è messa a fare a fare quello che vuole praticamente io schiaccio un tasto ma in realtà il comando non corrisponde a quello che	e ciao Simone sono una collega del di Corso Tazzoli	Inform
		e ho un problema con la tastiera del mio computer che improvvisamente si è messa a fare a fare quello che vuole	Inform
		praticamente io schiaccio un tasto ma in realtà il comando non corrisponde a quello che	Inform

ANNOTATION GUIDELINES

Difference between ack and yes-answer. Typical examples of ambiguity are: “yes”, “definitely”, “of course”². The choice mainly depends on whether the utterance is preceded by a yes-no question or not. Notice that a few utterances can occur between a question and its answer.

² si, certo, naturalmente

Here's a simple test on the utterance to be tagged: if an utterance can be paraphrased by "*I see*" ("*Capisco*") then it should be tagged as **ack**. Otherwise it should be tagged as **yes-answer**.

(1) e hai già provato comunque a cambiare magari la presa <- info-request
 sì (*Cannot* be replaced by "I see") <- yes-answer
(2) allora numero della chiamata è uno zero tre <- inform
 sì (*Can* be replaced by "I see") <- ack

Difference between inform and answer. The two tags basically apply to the same kind of utterances having the form of statements. Again the difference depends on the preceding utterance(s). If there is an adjacency pair having the current utterance as its second part, then the current utterance should be tagged as **answer**.

How to detect an info-request. If a question's main goal is to obtain some information it should be tagged as an **info-request**, even if it is formulated as a request to perform an action (verbs like *give, say, tell...*).

mi dica il suo numero di matricola senza interruzioni, dicendo un numero alla volta. <- info-request
allora mi lasci solo per favore il censimento del computer <- info-request

Politeness often causes the speaker to formulate an action-request as an info-request. The choice is left to the labeler. For instance the following utterance has both functions: it is aimed to know if a certain action has already been performed (info-request) and also to invite the addressee to perform it (action-request) if need be.

e hai già provato comunque a cambiare magari la presa <-info/action-request

In a more general way, an utterance does not need to be interrogative to be tagged as a request. Consider the following statement:

allora avrei bisogno solo di un recapito telefonico perchè giro una segnalazione per l'assistenza così faccio verificare la procedura <- info-request

Offer. The labeler should be aware of the real meaning of this tag, as attested by the literature: Committing-speaker-future-action (Core & Allen, 1997, p.7) or Commissive function (Bunt, H. 2006). An utterance addressing to the addressee a suggestion to perform an action *is not* an offer, and should be tagged as action-request. Compare the following \:

Vuole gentilmente aiutarci a migliorare la qualita' del corso rispondendo a tre domande? <- action-request
ti lascio il numero di remedy <- offer
come la posso aiutare? <- offer

Difference between offer and reportOnAction. Proposals to perform an action can sometimes be tagged in both ways. The difference should be made by considering the status of the action mentioned³: if the action is being or has just been performed by the speaker the utterance is a reportOnAction. If it hasn't been performed yet, then the speaker is committing himself to perform that action. The utterance is therefore an offer.

(1) Sto verificando la sua matricola. <-reportOnAction
(The speaker is checking the ID number *while uttering this*)
(2) ti lascio il numero di remedy <- offer
(The speaker is going to give the number *right after this utterance*)
(3a) okay allora le metto anche l'urgenza nella chiamata <-reportOnAction

³ Notice that no difference is made whether the utterance's commitment is conditional on the listener's agreement or not. Compare to DAMSL's distinction between **offer** and **commit**.

(The speaker is signaling the call as urgent *while uttering this*)

```
(3b) okay allora le metto anche l'urgenza nella chiamata ? <-offer
```

(The speaker *will* signal the call as urgent if the addressee agrees)

Concerning (3a) and (3b) if the transcription doesn't include punctuation marks, then the only way to know is to listen to the audio file and check if the prosody is declarative or interrogative.

Difference between inform and request. Without listening to the audio files, disagreement among labelers may be high since there is no question mark in the transcription. Context should help.

Difference between ack and filler. Backchannel utterances mainly have the form of acknowledgements and should therefore be tagged as **ack**, whereas **filler** signals an utterance whose main goal is to manage conversational time (a speaker takes time while keeping the turn).

```
(1) Sì <- ack
    allora un attimino solo <- filler
    censimento è duemilasei barra è quello <- clarificationRequest
(2) quali sono i tempi è una sempre una domanda molto difficile è una risposta più
    che altro molto difficile da dare <- filler
```

THE ANNOTATION TOOL

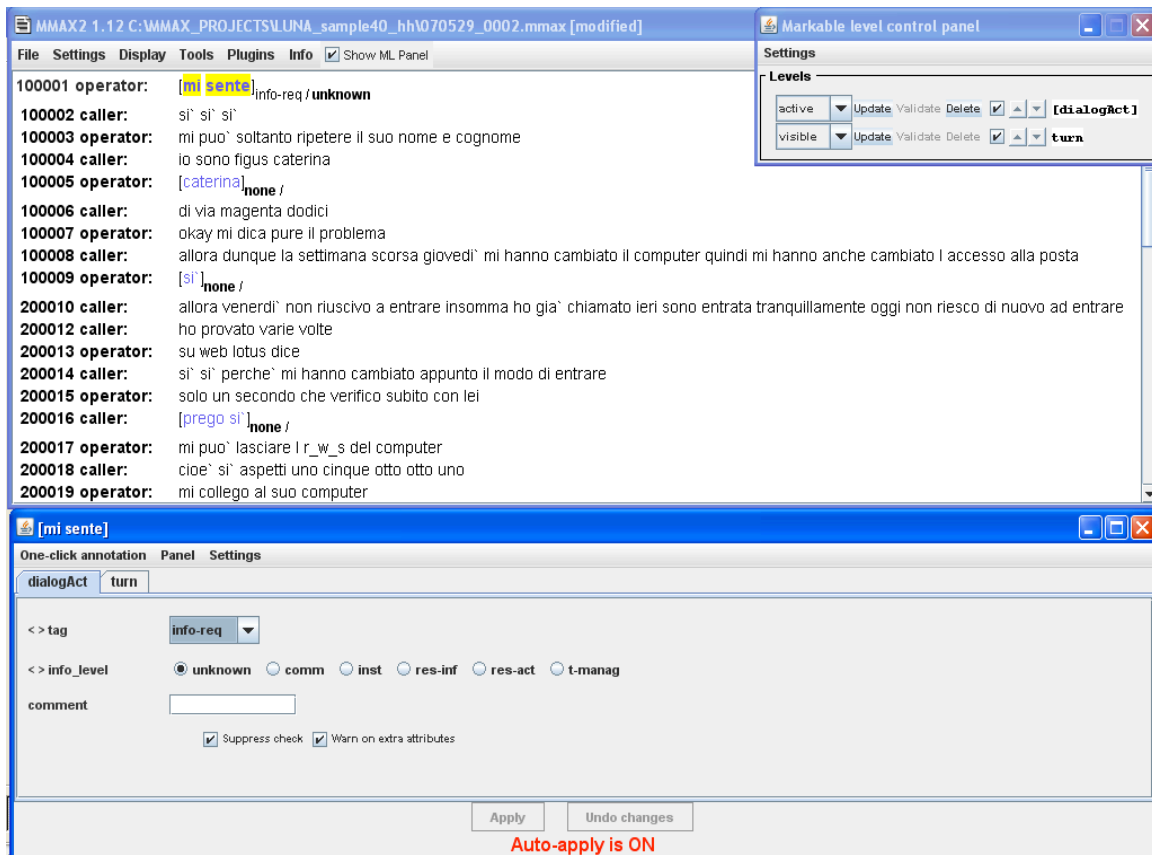
We propose the tool MMAX (<http://mmax.eml-research.de>) for dialog acts annotation of the LUNA corpus. This tool is highly customizable and respects the principle of *stand-off annotation*, i.e. file-level separation of base-data from the annotation itself (Ide & Greg, 1996). We can thus envisage using the same tool for different levels of annotation, which dramatically decreases data annotating and processing time cost. Other positive effects of *stand-off annotation* are that base data remain untouched during annotation thus allowing for simultaneous browsing and annotating on several linguistic levels by different users (Müller & Strube 2006).

Annotation unit. In MMAX representation, tokens are the basic language units to which all annotation layers must refer. Turn segmentation itself is seen as an annotation layer, with speaker and time stamps being represented as turn attributes. As a result dialog act segmentation is totally independent from turn segmentation and one dialog act can correspond to several turns (not necessarily contiguous) or to a subpart of a turn. Segmentation in MMAX is rendered by mouse drag selection of a sequence of tokens and creation of a “markable” (i.e. an annotation unit) on one of the active annotation levels (only *dialogAct* is concerned in our case). We suggest to first segment all of the dialog in dialog acts and to classify them later.

The MMAX project directory. The MMAX project archive contains all the XML and XSL configuration files needed to annotate a dialog according to ADAMACH annotation scheme through the MMAX interface (subdirectories: Schemes/, Styles/ and Customizations/). Project files (dialogID.mmax) are needed to load a dialog on the interface. Tokenized data to be annotated are stored in the Basedata/ subdirectory, and annotation output in the Markables/ one, which also contains turn segmentation.

How to get started. After launching MMAX, load the project file corresponding to the dialog you are to annotate. Make turn level *visible* on the *Markable level control panel* in order to prevent modification of turn segmentation. Enable the Auto-apply button on the Settings menu of the *Attribute window*. The interface is now ready for annotating.

Display features. Main display features include blue coloring of tokens belonging to a dialog act annotation unit (useful to visualize the text already segmented) and bold rendering of annotated utterances (useful to know which utterances still need to be tagged). In order to update the displaying of tags and dialog act segmentation, you should frequently check the button *Reapply current stylesheet* on the *Display* menu of main window.



In order to speed up the annotation process turns composed of up to two tokens are also dialog acts by default. Most frequently occurring short utterances like “okay” and “grazie” have been automatically tagged.

REFERENCES

- Core, M. G., & Allen, J. F. 1997. *Coding Dialogs with the DAMSL Annotation Scheme*. In: Proceedings of the 1997 AAAI Fall Symposium on Communicative Actions in Humans and Machines.
- Bunt, H. 2006. *Dimensions in Dialogue Act Annotation*.
- Müller, C. & Strube, M. 2006. Multi-level annotation of linguistic data with MMAX2. In: Sabine Braun, Kurt Kohn, and Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods*. Peter Lang: Frankfurt a.M., Germany.
- Ide, N., and Priest-Dorman, G. 1996. The corpus encoding standard.

ANNEX C. ANNOTATED LUNA DIALOG SAMPLES

Tables C.1 and C.2 present sample annotations of two LUNA human-human dialogs.

As mentioned above, such dialogs have been collected for the EU project LUNA with the aim of providing a multi-layer annotation, including syntactic (parse trees) and semantic levels of annotations (among which the dialog act level).

Notice that the original dialog transcription file (.trs) has been post-processed to resolve speaker overlap.

SegmentID, speaker	Segment Transcription	Dialog act
100001 operator	Eldes buongiorno sono Simone	info
100002 caller	e ciao Simone sono una collega del di Corso Tazzoli	info
	e ho un problema con la tastiera del mio computer che improvvisamente si è messa a fare a fare quello che vuole	info
	praticamente io schiaccio un tasto ma in realtà il comando non corrisponde a quello che	info
100003 operator	quello che hai quello che	oth
100004 caller	digitato digito sì	info
100005 operator	okay il tuo cognome	info-req
100006 caller	Bessone Susanna	answer
100007 operator	allora solo un attimo che recupero i tuoi dati	repaction
	allora Bessone Susanna okay	clarif
	mi dai anche il censimento della tua macchina per cortesia	info-req
100008 operator	allora duomilasei okay	clarif
100009 caller	sì allora diciotto zero tre C duomilasei diciotto	answer
	ho già una remedy aperta tra l'altro per un altro problema	info
200010 operator	per un altro discorso	clarif
200011 caller	sì sì sì	y-ans
200012 operator	allora	fil
200014 operator	e quindi il PC tu digiti qualunque tipo di cosa e lui a schermo ti ripropone delle altre lettere dico bene	clarif
200015 operator	okay	ack
200016 caller	sì e anche altri comandi proprio cioè si attivano altre	answer
200017 caller	altre funzioni proprio	
200018 operator	altre funzioni okay	ack
	allora dovresti farmi la cortesia di spegnere la macchina	act-req
200019 caller	sì	ack
200020 operator	e provare a scollegare e ricollegare la tastiera	act-req
200021 caller	sì	ack
200022 operator	perchè se un Mitas è un problema nostro	info
200023 caller	aspetta è dunque non sono tanto comoda allora aspetta che c'ho le vitine nella tastiera	repaction
200024 operator	fai pure con comodo	oth
200025 caller	no sto aspetta è	fil
200026 operator	no ma stai svitando delle viti	info-req
200027 caller	no no no	n-ans
	sto scollegando sul monitor	repaction
200028 caller	è sì sì aspetta	fil
200030 operator	infatti no stavo per dirti ho capito viti tastiera no	oth
	dovrebbe essere come P_S due o U_S_B tastiera del Mitas	info
200031 caller	no adesso lo trova e	info
200032 operator	è U_S_B se non sbaglio	clarif
200033 caller	no no no no	n-ans
200034 operator	o P_S due ancora	info-req
200035 caller	P_S due aspetta è okay allora	ack
	ho riattaccato	repaction
200036 caller	sì	ack
200037 operator	ha riattaccato okay	ack

	prova a riavviare un attimo il PC	act-req
	intanto te lo dovrebbe da già dare in in fase di inserimento di user name e password questo problema dico bene	info-req
200038 operator	okay	ack
200039 caller	è sì infatti il io avevo già provato a riavviarlo	info
200041 caller	quando ha iniziato a a fare le bizzze però e però poi non mi faceva neanche più inserire la password quindi	info
200042 operator	in teoria riavviandolo dopo aver scollegato e ricollegato la tastiera dovrebbe dovrebbe risistemarsi la cosa	info
200043 caller	va bene	ack
200044 operator	dovrebbe	oth
200045 caller	va bene	ack
200046 operator	perchè non è detto eventualmente si apre una chiamata all hardware e si manda un tecnico facendosi sostituire la tastiera	offer
200047 caller	la tastiera però lunedì entro lunedì	info-req
200048 operator	assolutamente sì sì sì	y-ans
	cioè oggi di oggi no	oth
200049 operator	perchè	fil
200050 operator	okay	ack
200051 caller	no no adesso andrei a casa però tanto oggi tra la le la luce poi tra i problemi di rete e questo proprio	oth
200052 operator	fantastico	ack
200053 caller	stata abbastanza inutile la giornata sta andando è	oth
200054 operator	okay sta caricando Windows okay	info-req
200055 operator	okay	ack
200056 caller	è sì sì allora control al canc dunque il la tastierina dei numerini me lo prende ma me lo prendeva anche prima okay	info
200057 operator	funziona anche il resto	info-req
200058 caller	sì è andato	y-ans
200059 caller	grazie mille	thank
200060 operator	okay direi che siamo a posto niente	quit
200061 caller	ciao ciao ciao ciao	quit
200062 operator	ciao ciao buona giornata	quit

TABLE C.1: Annotated dialog from the LUNA corpus (Dialog ID: 070300_0003)

SegmentID, speaker	Segment Transcription	Dialog act
100001 operator	signora mi dice solo il suo nome per cortesia	info-req
100002 caller	sì Rosolino Monica	answer
100003 operator	allora Rosolino ha mica il numero di inventario della della stampante del settore informatica	info-req
100004 operator	sì	ack
100005 caller	allora io ne ho segnati due è il cento dodici sei otto nove	answer
100006 operator	allora cento dodici cento nove dovrebbe andar bene	ack
100007 operator	aspetti che verifichiamo	repaction
100008 caller	perchè se no ho quella del settore informatico non so qual è	info
100009 caller	sei sette cinque uno	answer
200010 caller	cinque uno	
200011 caller	con la diretta	info
200012 operator	ce l ha allora mi dia quello del settore informatico	info-req
	sette sette cinque uno è una brother H_L mille cento settanta allora mi ha detto che errore le segnala	info-req

200013 caller	errore E quarantanove si inceppa la carta	answer
200014 operator	allora	fil
200015 caller	adesso ho tolto tutti i fogli però mi dà comunque errore quarantanove e non riesco proprio più a lavorare	info
200016 caller	sì	ack
200017 operator	allora sul display glielo segnala questo errore	info
200018 caller	è rimasta in rosso non non va proprio più	answer
200019 operator	allora ha provato anche a spegnerla riaccenderla	info-req
200020 caller	sì sì da ieri	y-ans
200021 operator	da ieri che la situazione	oth
200022 caller	sì ogni tanto dà anche errore E cinquanta però quando è venuto il tecnico mi aveva spiegato come come sbloccarlo questo qua invece non so	info
200023 caller	sì	ack
200024 operator	quindi inceppa proprio i fogli all interno	clarif
	allora mi lascia solo un recapito telefonico	info-req
200025 caller	allora quattro tre due cinque due quattro tre	answer
200026 operator	ok l ufficio in cui si trova la stampante è al terzo piano	info-req
200027 caller	sì ufficio trecento dodici	y-ans
200028 caller	sì	ack
200029 operator	trecentododici	ack
	allora le lascio il numero della chiamata	offer
200030 caller	sì	ack
200031 operator	vedo di mandargli l assistenza hardware	offer
200032 caller	va bene	ack
200033 operator	allora uno zero quattro	info
200034 caller	sì	ack
200035 operator	zero quattro tre sette	info
200036 caller	zero quattro tre sette zero quattro uno zero quattro zero quattro tre sette	clarif
200037 operator	Perfetto	ack
200038 operator	Ok	ack
200039 caller	va bene la	quit
200040 caller	ringrazio arrivederci salve	
200041 operator	grazie a lei buona giornata salve	quit

TABLE C.2: Annotated dialog from the LUNA corpus (Dialog ID: 070400_0005)

ANNEX D. ANNOTATED HUMAN-COMPUTER (WIZARD OF OZ) DIALOG SAMPLE FROM THE LUNA CORPUS

Here is a sample annotation of a LUNA human-computer dialog. Turns uttered by the system contain punctuation marks used as the input of the TTS module.

TurnID, Speaker	Turn content	Dialog Act
001 operator	help desk buongiorno !	Greet
	sono paola !	info
	in cosa posso esserti utile ?	info-req
002 caller	buongiorno guardi abbiamo un problema molto grave perche' stamattina tutta la nostra sede qua` di corso unione e' fuori rete	answer
003 operator	mi dici il tuo nome e cognome ?	info-req
004 caller	si mi chiamo mario giallo	answer
005 operator	per quale ente lavori ?	info-req
006 caller	c_s_i piemonte	answer
007 operator	mi dici il tuo numero di telefono ?	info-req
008 caller	zero undici tre quattro cinque ottantotto ottantotto	answer
009 operator	i colleghi della tua sede , ma di uffici diversi , hanno lo stesso problema ?	info-req
010 caller	si si tutti i	y-ans
011 operator	ora ti comunico il numero di ticket assegnato a questa richiesta .	Offer
	il tuo numero di ticket e` : 1338229 .	Info
	te lo sei segnato ?	Info-req
012 caller	di tutti gli uffici no cioe` c' e` tutta la sede fuori rete	oth
013 operator	grazie ! sarai contattato nel piu` breve tempo possibile dal tecnico dell' help desk .	thank
	buona giornata !	quit
014 caller	si grazie me lo sono segnato	y-ans
016 caller	arrivederci	quit

Table D.1 Annotated dialog from the LUNA corpus (Dialog ID: 018)

ANNEX E: ANNOTATED HUMAN-COMPUTER DIALOG SAMPLES FROM THE ADAMACH HELPDESK CORPUS

In this annex, we report the dialog act annotation performed according to the proposed dialog act tag set on two sample human-computer dialogs from the ADAMACH Helpdesk corpus (Tables E.1 and E.2). The corpus has been collected during the evaluation of the first prototype of an adaptive student helpdesk application (see Varges, Riccardi and Quarteroni 2008 for more details).

TURN	Transcription	Dialog act
S1	Questo e', l'help desk, dell'universita' di Trento!	Inform
	La telefonata potrebbe essere registrata per migliorare il servizio offerto.	Inform
	Il trattamento dei dati raccolti e' effettuato nel rispetto della normativa sulla pra'ivasi	Inform
	Buongiorno Sono Paola,	Greet
	come la posso aiutare?	Offer
U1	sapere il risultato di fondamenti di informatica	Info-request
S2	OK, mi dica, il suo numero di matricola senza interruzioni, dicendo un numero alla volta.	Info-request
U2	125064	Answer
S3	Sto verificando la sua matricola.	ReportOnAction
S4	Mi conferma che lei e' AVELLA STEFANIA matricola 125064	ClarificationRequest
U4	sì confermo	Yes-answer
S5	Sto leggendo il voto dell'esame dall'archivio.	ReportOnAction
S6	La prova e' molto positiva, il suo voto e' 29	Answer
S7	Vuole gentilmente aiutarci a migliorare la qualita' del corso rispondendo a tre domande?	Offer
U7	No	No-answer
S8	Scusi non ho capito vuole rispondere ad un breve questionario?	ClarificationRequest
U8	No	No-answer

Table E.1. Annotated dialog from the ADAMACH TrialNov07 corpus.
Dialog date: 20071122, time: 15:43:42, VoxNauta ID: 127.0.0.1-4044-1195742621-834

TURN	Transcription	Dialog act
S1	Benvenuto;	Greet
	Questo è, l'help desk, dell'università di Trento!	Inform
	La telefonata potrebbe essere registrata per migliorare il servizio offerto.	Inform
	Il trattamento dei dati raccolti è effettuato nel rispetto della normativa sulla privacy	Inform
	Buongiorno Sono Paola,	Greet
	come la posso aiutare?	Offer
U1	iscrizioni esami	Action-request
S2	Mi dica, a quale esame intende iscriversi?	Info-request
U2	fondamenti informatica	Answer
S3	Mi dica, il suo numero di matricola senza interruzioni, dicendo un numero alla volta	Info-request
U3	130871	Answer
S4	Il numero di matricola che Lei ha detto 1 3 0 8 7 1 , conferma ?	ClarificationRequest
U4	no	No-answer
S5	Dica se è corretto, oppure no!	ClarificationRequest
U5	No	No-answer
S6	Mi dica di nuovo, il suo numero di matricola	Info-request
U6	130871	Answer
S7	Il numero di matricola che Lei ha detto 1 3 0 8 7 1 , conferma ?	RequestClarification
U7	sì	Yes-answer
S8	Vuole iscriversi per l'esame fondamenti_di_informatica, il suo numero di matricola è 1 3 0 8 7 1 ed il suo nome MATTEO FRISINGHELLI, corretto?	ClarificationRequest
U8	Sì	Yes-answer
S9	Grazie, per essersi iscritto.	ReportOnAction
	Posso aiutarla in un altro modo?	Offer
U9	giorno dell'esame	Info-request
S10	Grazie e Arrivederci!	Thank Quit

Table E.2. Annotated dialog from the ADAMACH Trial III corpus.
Dialog date: 20080121, time: 15:57:40, VoxNauta ID: 1200927461

