# Improving Dialogue Annotation Reliability

**Teresa Sikorski**
Department of Computer Science
University of Rochester
Rochester, New York 14627-0226
sikorski@cs.rochester.edu

**Abstract**

Data collection and annotation represent the most significant portion of the effort involved in building statistical models of dialogue. We must be able to annotate reliably if there is any hope of building predictive statistical models. This paper describes the annotation process being used to tag the TRAINS-93 corpus with dialogue acts, and measures we have taken to improve inter-annotator reliability.

## Introduction

The TRAINS-93 corpus is a set of over 90 dialogues in a cargo shipping domain. In each of these dialogues, two participants cooperate to build a plan to perform a given task under some specified constraints. (Heeman & Allen 1995). The corpus was compiled at the University of Rochester in an experiment where one of the dialogue participants emulated a computer system and the other participant acted as the human user.

At the University of Rochester, we are currently annotating the TRAINS-93 corpus with tags that describe the dialogue acts. Our motivation in tagging the corpus with dialogue acts is threefold:

- To facilitate the study of the structure of task-oriented dialogue.
- To produce a coded corpus that can be used by machine learning algorithms to model dialogue.
- To be used as a standard by which dialogue system performance can be evaluated.

## The DAMSL Annotation Scheme

The first step in the annotation process was to decide on an appropriate set of tags. DAMSL (**D**ialogue **A**ct **M**arkup in **S**everal **L**ayers) is a dialogue annotation scheme that provides tags for several characteristics of utterances that indicate their role in the dialogue (Allen & Core Draft 1997). Table 1 gives a summary of

| **Characteristic** | **Tags** |
|---|---|
| **Communicative Status** | |
| **Self-Talk** | [Yes \| No \| Maybe] |
| **Unintelligible** | [Yes \| No \| Maybe] |
| **Abandoned** | [Yes \| No \| Maybe] |
| **Information Level** | |
| **Info-level** | [Communication-management \| Task \| Task-management \| Other] |
| **Forward Communicative Functions** | |
| **Statement** | [None \| Assert \| Reassert \| Other-statement] |
| **Influence-on-listener** | [None \| Open-option \| Action-directive] |
| **Influence-on-speaker** | [None \| Offer \| Commit] |
| **Info-request** | [Yes \| No] |
| **Conventional** | [Yes \| No] |
| **Exclamation** | [Yes \| No] |
| **Explicit-performative** | [Yes \| No] |
| **Other-forward-function** | [Yes \| No] |
| **Backward Communicative Functions** | |
| **Agreement** | [None \| Accept \| Accept-part \| Reject \| Reject-part \| Maybe \| Hold] |
| **Understanding** | [None \| Acknowledge \| Signal-non-understanding \| Correct-misspeaking \| SU-Repeat-rephrase \| SU-Completion] |
| **Answer** | [Yes \| No] |
| **Response-to** | [⟨*any prior utterance number*⟩ \| None] |

Table 1: DAMSL Annotation Scheme

the utterance characteristics identified within DAMSL and the tags provided for each characteristic. Note that when utterances are given a "Yes" tag for any of the Communicative Status characteristics, the expectation is that all other characteristics will be tagged as "No" or "None" since the utterances do not actually contribute to the dialogue.

## The Annotation Process

Before being tagged, each dialogue is stored in a separate file and is preprocessed to partition it into numbered utterance units with the speaker of each utterance indicated.

Our procedure for training annotators is to have them read the general DAMSL manual and then read a set of TRAINS-specific annotating instructions. All annotators tag the same dialogue independently in separate files using a dialogue annotation tool developed at the University of Rochester. When the annotators have finished tagging the training dialogue, a meeting is held to review the results and to rectify misunderstandings. This process is repeated with different dialogues until the annotators have achieved an acceptable level of proficiency.

After training is complete, each dialogue is assigned to two annotators who make a first pass at the annotations independently. When both annotators have completed a dialogue, a tool is run to compare the pair of tagged dialogues. The comparison tool generates files for each of the utterance characteristics, itemizing the differences in the tags assigned to each utterance, and giving the reliability scores. The annotators meet to discuss the differences, and together produce a reconciled version of the annotated dialogue. The reconciled version of the annotated dialogue is then used as the correctly tagged dialogue.

The reliability of a dialogue tagging scheme such as DAMSL is generally evaluated with the kappa statistic, which measures agreement between annotators, while accounting for chance agreement (Carletta 1996). The current rule of thumb applied is that kappa scores must be above 0.67 for tentative conclusions to be drawn, with above 0.8 considered reliable.

Unfortunately, the inter-annotator reliability scores from our early attempts to tag the TRAINS-93 corpus using the DAMSL scheme did not meet even the "tentative conclusion" threshold for many of the most important features in which we are interested. Reliability scores for Influence-on-speaker, Agreement, Info-Level, and Other-forward-function were particularly low.

Three steps we have taken so far to increase inter-annotator reliability to an acceptable level are:

- The DAMSL manual was revised in an attempt to make crisper distinctions between tags.
- An annotation manual specific to the TRAINS-93 corpus was written that explicitly sets out the "correct" tags for common and problematic phenomena within the TRAINS dialogues (Sikorski 1998).
- A feature was added to the annotation tool that causes warnings to be issued when unlikely tags are assigned.

## Domain-specific Annotation Instructions

The DAMSL manual is fairly general, describing a tagging scheme appropriate for dialogues collected in various domains. The TRAINS-specific instructions are comparatively fine-grained and address specific linguistic phenomena that occur frequently in the TRAINS-93 domain.

Many times, convincing arguments can be made to justify different sets of tags on the same utterance. Consider the following utterance:

```
you need to design a plan for
shipping a boxcar of oranges to
Bath by eight a.m. today
```

In virtually all of the TRAINS-93 dialogues, the task's goal is explicitly stated. When the goal is stated as an imperative as above, annotators are tempted to tag the utterance as an Action-directive, and a reasonable argument can be given for doing so. However, since the speaker is addressing *what* needs to be done rather than instantiating a plan specifying *how* to do it, we have decided to treat goal statements simply as Inform acts, at the Task level. This was one of many such policies adopted during early reconciliation processes that have been documented in the TRAINS-specific annotation manual. Domain-specific annotation manuals allow policies to be documented about how problematic cases particular to a domain should be handled.

Of the fifty-thousand words in the TRAINS-93 corpus, the word *okay* appears over 2000 times. Furthermore, other words that can serve the same functions as *okay*, such as *right, alright*, and *yes*, combine to account for seven percent of the words in the entire corpus. Since these terms have a number of possible functions, they are especially difficult to annotate reliably. For example, *okay* can serve as an answer, an acknowledgment, an acceptance, or different turn-taking acts (which are tagged under the Other-forward-function characteristic in the DAMSL scheme). Unfortunately, mistagging one of these "*okay*-like" terms can cause errors in multiple dimensions. For example, if an acknowledgment/acceptance is mistaken for a turn-taking action, the Info-level, Agreement, Understanding, and Other-forward-function characteristics will all be tagged incorrectly. The TRAINS-specific annotation manual attempts to minimize such errors by providing fairly rigid criteria for determining how to tag a particular occurrence of one of these terms.

## Rochester's Dialogue Annotation Tool

As with most tasks, having the right tools is crucial for annotating dialogues. Figure 1 shows the graphical interface provided by Rochester's dialogue annotation tool.

Users of the annotation tool are able to scroll through an entire dialogue, select individual utterances, and apply tags to the utterances by clicking on the appropriate buttons. The annotation tool also provides the capability of playing audio recordings of single utterances or a consecutive set of utterances. The ability to listen to utterances when annotating dialogue acts is important because many times speakers use intonation to indicate the intended dialogue act. For example, consider the following utterance:

```
only four boxes with cargo can go at a
time
```

The surface form of the above utterance suggests an Inform act. However, when spoken with rising intonation, the intended act is likely to be an Info-request. We have found several examples in the TRAINS corpus where the the intended act is ambiguous without the intonational information.

Another important feature of the annotation tool is that it issues warning messages via a pop-up window when unlikely combinations of tags are assigned to an utterance or pairs of utterances. In the current tool, the rules that govern the warning messages were handcrafted. Specifically, the tool issues the following warnings, when appropriate:

- A response to an Info-request should probably be an Answer.
- An utterance tagged as Acknowledge that does not also show agreement should probably be tagged as Communication-management.
- Responses to Action-directives should probably not have the Agreement characteristic tagged as None.
- Answers should probably not have the Statement characteristic tagged as None.
- The Response-to field for utterances that are Answers, or show Understanding or Agreement, should be set to a prior utterance number.

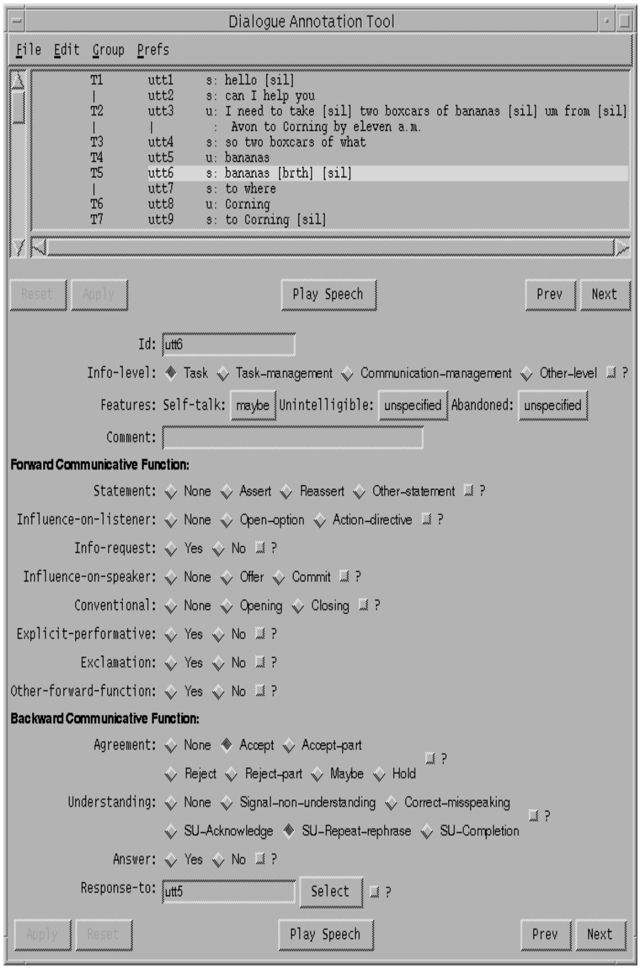


Figure 1: Rochester's Dialogue Annotation Tool

## Applying Machine Learning To Annotation

Now that we have tagged a subset of the TRAINS corpus, machine learning algorithms can discover correlations that can help us predict how a given utterance should be tagged. These predictions can be of use in one of two ways to simplify the annotation process and increase annotation reliability when tagging the remainder of the corpus:

- Preprocess dialogue files so that annotators start off with a hypothetical set of tags for the utterances

| Word | Tag | A Priori Prob of the Tag | Prob (Tag \|Word) |
|---|---|---|---|
| **okay** | Accept<br>Acknowledge | 0.32<br>0.27 | 0.62<br>0.72 |
| **right** | Accept<br>Acknowledge | 0.32<br>0.27 | 0.72<br>0.53 |
| **you** | Info-request<br>Offer | 0.15<br>0.05 | 0.60<br>0.41 |
| **can** | Info-request<br>Offer | 0.15<br>0.05 | 0.48<br>0.39 |
| **have** | Accept | 0.32 | 0.00 |
| **take** | Accept | 0.32 | 0.00 |
| **engine** | Accept | 0.32 | 0.00 |
| **boxcars** | Accept | 0.32 | 0.00 |
| **then** | Accept | 0.32 | 0.00 |
| **of** | Accept | 0.32 | 0.00 |

Table 2: Predictive Words from the TRAINS Corpus

in the dialogue. The annotators need only make changes to tag assignments with which they disagree.

- Modify the dialogue annotation tool to issue warnings when unlikely tags (based on learned correlations) are assigned, or when likely tags are not.

Even a simple analysis of the tagged subset for TRAINS using single words as predictors can identify unlikely utterance tags. Table 2 contains some results from such an analysis. After tagging and reconciling a set of fifteen TRAINS-93 dialogues, the set was analyzed to determine the distribution of tags. For example, of the approximately 1100 utterances tagged, 32% of the utterances were tagged as Accepts. We also looked at the correlation between words appearing in utterances and the tags that were assigned to the utterance. In our simple analysis, each time a word was encountered in the tagged subset, it voted for the tags that had been applied to the utterance in which it appeared. In drawing our conclusions, we considered only high-frequency words: those that appeared at least fifty times in the tagged subset. This analysis provided us with evidence that can be used in our dialogue annotation tool when an unlikely tag assignment is made. For example, although 32% of the utterances were Accepts, the analysis shows that it is highly unlikely that utterances containing the words *have, take, engine, boxcars, then* or *of* should be tagged as Accepts. On the other hand, although only 5% of the utterances are Offers, the presence of the words *can* and *you* provide evidence that the Offer tag should be considered.

## Future Work

Our future work includes applying machine learning techniques to discover patterns based on more complex features of the tagged dialogues, e.g.:

- n-grams of the words in an utterance
- position of words in an utterance
- n-grams of tags applied to previous utterances

An evaluation of the predictive power of the correlations drawn would need to be performed, and the annotation tool revised to make use of the learned correlations that are found to be especially predictive.

## Acknowledgments

This work was supported in part by National Science Foundation grants IRI-9528998 and IRI-9623665. Thanks to Mark Core for his work on the DAMSL manual and various annotation support tools and to George Ferguson for developing the dialogue annotation tool.

## References

Allen, J., and Core, M. Draft 1997. DAMSL: Dialog annotation markup in several layers. Available from http://www.cs.rochester.edu:80/ research/trains/annotation.

Carletta, J. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics* 22(2).

Heeman, P., and Allen, J. 1995. The TRAINS-93 dialogues. TRAINS Technical Note 94-2, Department of Computer Science, University of Rochester.

Sikorski, T. 1998. Applying the DAMSL tagset in the TRAINS-93 domain. TRAINS Technical Note 98-1, Department of Computer Science, University of Rochester.