Automatically Extracting Grounding Tags from BF Tags

Teresa Zollo and Mark Core
Department of Computer Science
University of Rochester
Rochester, New York 14627-0226
zollo@cs.rochester.edu
mcore@cs.rochester.edu

Abstract

This paper describes how to automatically extract grounding features and segment a dialogue into discourse units, once the dialogue has been annotated with the DRI backward- and forward-looking tags. Such an approach eliminates the need for separate annotation of grounding, making dialogue annotation quicker and removing a possible source of error. A preliminary test of the mapping against a human annotator is presented.

1 Introduction

The annotation scheme (AC97) developed by the Discourse Research Initiative's Backward- and Forward-Looking Group (henceforth referred to as the BF scheme) provides a set of tags that can be applied to individual utterances in a dialogue, describing the utterance's illocutionary force. The BF scheme provides a standard top-level tag set that allows researchers to reuse corpora that have been annotated for other projects, and also allows tags to be refined by individual projects to provide detail on particular phenomena being studied.

There are a number of dialogue features that are of interest to researchers, and for which tagging schemes have been developed. One feature that we are concerned with is **grounding**, the mechanism by which dialogue participants augment their mutual beliefs. In his dissertation work (Tra94), Traum establishes a set of tags to describe grounding behavior, and then uses this taxonomy of grounding acts to describe a computational model of how dialogue participants achieve a state of mutual understanding. Traum's model describes how grounding acts can be combined to form **discourse units**, segments of a dialogue that correspond to individual contributions to the common ground. Clark and Schaefer define a contribution as the presentation of a proposition by one dialogue participant, as well as all subsequent related utterances until there is adequate evidence that the initial utterance was understood or abandoned (CS89). Discourse units are the level of granularity at which other dialogue tags, such as the problem-solving acts described in (SA97), are applied.

Annotating dialogues can be a time-consuming and error-prone undertaking. To make the annotation process easier and more reliable, care should be taken to avoid manually tagging information that can be derived from other tags or that can be automatically extracted. This paper explores how we can *automatically* annotate dialogues with grounding tags, given a corpus that has been annotated with the BF tags. Once grounding has been marked, we can automatically segment the dialogue into discourse units, using Traum's model.

In order to tag with BF tags or grounding tags, a dialog must be segmented into utterances, a problem that is discussed briefly in section 2. Section 3 gives an overview of the BF tags and grounding tags, section 4 discusses the mapping from BF tags to grounding tags, and section 5 presents a comparison of the automatic mapping to a human annotator.

2 Segmenting dialogues into utterances

Dialogues need to be segmented into utterances before annotation with the BF tags. Unfortunately, there is no widely accepted criteria for identifying utterances. Traum's approach to utterance segmentation is to segment utterances based on the presence of prosodic evidence such as pauses and boundary tones, and on changes of speaker. The benefit of this approach is that it can be done automatically given prosodic annotation. However, we have found this approach to be somewhat problematic since very often the resulting utterance units need to be combined or split when assigning the BF tags. Traum uses a special grounding tag, CONTINUE, when a prosodically-segmented utterance is not an independent grounding act, but rather part of the same grounding act as a previous utterance by the same speaker.

Another possible approach to utterance segmentation for BF tagging is to allow the annotator to segment the dialogue and label it for BF tags at the same time. The problem with this approach is that different annotators may segment the same dialogue differently, making it difficult to compare annotations. One way of dealing with this problem is to have subsequent annotators use the first annotator's segmentation. A drawback of this solution is that the first annotator's segmentation may influence subsequent BF labeling. Despite this drawback, we are assuming the second approach in order to avoid the need to split or join utterances, and therefore do not need Traum's CONTINUE tag.

3 Overview of Tag Sets

Table 1 shows the illocutionary act features included in the BF tagging scheme, along with the tags for each feature. Actions performed during the grounding process are shown in Table 2.[1] In Traum's annotation scheme for grounding, the tags are not mutually exclusive.

Feature	**Tags**
Communicative Status	
Self-Talk	YES | NO | MAYBE
Unintelligible	YES | NO | MAYBE
Abandoned	YES | NO | MAYBE
Information Level	
Info-level	COMMUNICATION-MGMT | TASK | TASK-MGMT
Forward Communicative Functions	
Statement	NONE | ASSERT | REASSERT
Influence-on-listener	NONE | OPEN-OPTION | ACTION-DIRECTIVE
Influence-on-speaker	NONE | OFFER | COMMIT
Info-request	NONE | INFO-REQUEST | CHECK
Conventional	YES | NO
Other-forward-function	YES | NO
Backward Communicative Functions	
Agreement	NONE | ACCEPT | ACCEPT-PART | MAYBE | HOLD | REJECT-PART | REJECT | WH-ANSWER
Understanding	NONE | ACKNOWLEDGE | SIGNAL-NON-UNDERSTANDING | CORRECT-MISSPEAKING | SU-REPEAT-REPHRASE | SU-COMPLETION
Response-to	⟨*any prior utt number*⟩ | NONE

Table 1: BF Features and Tags

The BF scheme has four main layers: communicative status, information level, forward communicative function, and backward communicative function. Communicative status is used to label utterances that cannot be understood, are broken off, or are not directed at other conversational participants. Information level is used to differentiate between utterances discussing the topic at hand (TASK and TASK-MGMT) and utterances whose sole purpose is to manage the conversation (COMMUNICATION-MANAGEMENT). COMMUNICATION-MANAGEMENT utterances can be simple acknowledgments (*okay*) or explicit comments on the communication process (*I didn't hear that*). Forward communicative functions are aspects of an utterance that directly address future actions. Requests and suggestions are included in INFLUENCE-ON-LISTENER and INFO-REQUEST; Commitments and offers are included in INFLUENCE-ON-SPEAKER; and statements about the world are included in STATEMENT. OTHER-FORWARD-FUNCTION identifies utterances that have a turn-taking function but no other forward communicative function. The second utterance below is an example of OTHER-FORWARD-FUNCTION:

```
utt1 u: and that would be the fastest
utt2    okay okay um
utt3    we're done
```

Backward communicative functions include comments on the content of previous utterances (AGREEMENT) as well as utterances that signal whether previous material was understood or not (UNDERSTANDING). Examples of UNDERSTANDING include SIGNAL-NON-UNDERSTANDING as well as various types of showing understanding: simple ACKNOWLEDGMENTs, acknowledgment through repetition/paraphrase (SU-REPEAT-REPHRASE), acknowledgment through correction (CORRECT-MISSPEAKING), and acknowledgment through elaboration/completion (SU-COMPLETION).

The grounding acts of Traum are INITIATE,

[1] The CONTINUE act is merely an artifact Traum's approach to utterance segmentation, and we omit it from further discussion.

Grounding Act	Description
INITIATE	the initial presentation of a proposition
REPAIR	a modification to the content or presentation of the current proposition under consideration
REQUEST-REPAIR	a request that the other participant perform a REPAIR
ACKNOWLEDGE	evidence that a previous utterance has been understood
REQUEST-ACKNOWLEDGE	a request that the other participant perform an ACKNOWLEDGE
CANCEL	an abandonment of the proposition under consideration

Table 2: Traum's Grounding Acts

REPAIR, REQUEST-REPAIR, ACKNOWLEDGE, REQUEST-ACKNOWLEDGE, and CANCEL. Dialogue participants use these actions to form discourse units as they converse. INITIATEs start discourse units. A discourse unit is terminated either through an ACKNOWLEDGE, in which case the discourse unit is considered grounded, or through a CANCEL, in which case the discourse unit is not grounded. Acknowledgments may be either explicit or implicit. Explicit acknowledgments can be requested by performing a REQUEST-ACKNOWLEDGE, such as *Did you get that?*. Once an initial presentation is made, either participant may make a REPAIR, or enter into a repair subdialogue by performing a REQUEST-REPAIR.

4 Mapping from BF tags to grounding tags

In general, any utterance tagged as having a forward communicative function in the BF scheme initiates a new discourse unit and should be given an INITIATE grounding tag. Exceptions are utterances that *only* perform a turn-taking act. These are tagged as OTHER-FORWARD-FUNCTION in the BF scheme, but have no content that requires acknowledgment. Utterances that have both a turn-taking function and some other forward communicative function, such as *Give me a second.* (tagged as an ACTION-DIRECTIVE and OTHER-FORWARD-FUNCTION at the COMMUNICATION-MANAGEMENT level) *do* have content that can be acknowledged and should be tagged as INITIATE. Another exception found frequently in dialogues from collaborative task-oriented domains are utterances that are tagged as COMMIT because they ACCEPT an ACTION-DIRECTIVE. Utterances 2 and 4 in the following dialogue excerpt are examples of COMMITs that are not INITIATEs.

```
utt1 u: pick up two tankers in Corning
utt2 s: okay
utt3 u: then on the way back to Elmira
        pick up another tanker
utt4 s: okay
```

The BF tag SU-COMPLETION is interesting since an utterance having this tag should be INITIATE and ACKNOWLEDGE in Traum's scheme, despite the fact that completions are not labeled with forward communicative functions. The completion has an implicit forward communicative function which is taken as the same as the utterance (by another speaker) that it is completing.

Repairs are attempts to fix an utterance through *correction* or *clarification*. Corrections reject an utterance and offer a replacement. Clarifications provide additional information about an utterance. Because of the level of granularity at which the BF tags are applied, self-repairs made mid-utterance are not included.

An utterance B, should be given a REPAIR grounding tag with respect to utterance A, if B is a response to A and any of the following patterns of BF tags are seen:

1. Utterance B is tagged as SU-CORRECT-MISSPEAKING.

2. Utterance B is tagged with COMMUNICATION-MANAGEMENT and either REJECT or REJECT-PART, and a forward communicative function. In this case, the dialogue participant is making an *unsolicited* repair of their previous utterances.

3. Utterance A has the tag SIGNAL-NON-UNDERSTANDING and utterance B has a forward communicative function and does not have REJECT or REJECT-PART tags. In this case, the dialogue participant is making an *solicited* repair.

All utterances having a SIGNAL-NON-UNDERSTANDING BF tag receive a REQUEST-REPAIR grounding tag.

An utterance is given a REQUEST-ACKNOWLEDGE grounding tag when it has either of the following patterns of BF tags:

1. The utterance is tagged as CHECK. These are check-questions, also known as tag-questions, and include examples such as *we will take the top route right?*.

2. The utterance is tagged as both COMMUNICATION-MANAGEMENT and INFO-REQUEST, and is not tagged as SIGNAL-NON-UNDERSTANDING. Examples of utterances of this type are *Did you get that?* and *Are you listening?*

Utterances that are tagged as ABANDONED in the BF scheme will be tagged as CANCEL in Traum's grounding scheme. Sometimes a dialogue participant CANCELs an open discourse unit by saying something like *Forget it* or *Never mind* in response to a repair initiation, such as *What did you say?* In the BF scheme, these CANCELs appear as REJECTs at the COMMUNICATION-MANAGEMENT level, responding to SIGNAL-NON-UNDERSTANDINGs.

In the BF scheme, acknowledgments are utterances that *explicitly* indicate that a previous utterance was understood. In Traum's scheme, acknowledgments can either explicitly or *implicitly* signal understanding. Explicit acknowledgments occur when a dialogue participant repeats, paraphrases, or completes what was said or when they use an acknowledgment term such as *okay*. Implicit acknowledgments occur when a dialogue participant continues the dialogue in a way that is consistent with what has been said previously in the dialogue.

An utterance B, should be tagged as an ACKNOWLEDGE to utterance A in Traum's scheme under any of the following conditions:

1. Utterance B is tagged as SU-ACKNOWLEDGE in the BF scheme, with the Response-to field set to A. These utterances are examples of acknowledgment terms such as *okay*.

2. Utterance B is tagged as SU-REPEAT-REPHRASE or SU-COMPLETION in the BF scheme, with the Response-to field set to A. These utterances are examples of explicit acknowledgments by paraphrase, repetition, or completion.

3. Utterance B is tagged with an agreement tag with the Response-to field set to A, and the combination of BF tags has not already been determined to indicate CANCEL or REPAIR. These utterances implicitly acknowledge A by indicating agreement with the propositional content of A.

4. Utterance B is tagged as either WH-ANS, ASSERT or REASSERT, with the Response-to field set to A, and A was tagged as INFO-REQUEST. Such utterances implicitly show acknowledgment of a previous utterance by answering a question posed in the previous utterance.

Problems arise when an interlocutor implicitly acknowledges an initiator's presentation either by continued attention or by initiating a new contribution that is consistent with and relevant to the previous presentation. The following dialogue segment is an example of such an exchange:

```
utt1 u: our task is to get two tankers
        of orange juice to Corning by
        7 am
utt2 s: the orange warehouse is in
        Corning
```

The reason that this case is somewhat problematic to our scheme is that it is not clear that utterance 2 should be tagged as an ACCEPT of utterance 1 in the BF scheme, and if the BF annotators fail to tag utterance 2 as an ACCEPT, it will not be identified as an ACKNOWLEDGE. (In the BF scheme, the Understanding feature is only tagged when an explicit acknowledgement or signal of non-understanding is made.)

5 Evaluation

In order to determine whether the mapping we propose here results in accurate grounding annotation, we wrote a Perl script to perform the mapping on SGML-format files containing dialogues annotated with the BF tags. We used the script on a set of four TRAINS-93 dialogues containing a total of 325 utterances, that had been previously tagged with BF tags (HA95; CA97).

The procedure for tagging the dialogues with BF tags was to have an annotator segment and annotate the dialogue, pass the segmented (but untagged) dialogue to a second annotator to tag independently, and finally for the two annotators to meet and produce a reconciled version of the tagged dialogue.

To evaluate the quality of the tags that were output by the script, we had a human annotator tag the

Category	Number of Occurrences	Number of Disagreements
INIT	367	33
ACK	332	44
NO-TAG	41	21
REQACK	24	16
REPAIR	9	5
CANCEL	8	2
REQREP	2	0

Table 3: "Partial Credit" Analysis

Category	PA	PE	kappa	Sig Level
INIT	0.8985	0.5084	0.7935	0.000005
ACK	0.8646	0.5002	0.7291	0.000005
NO-TAG	0.9354	0.8818	0.4533	0.005
REQACK	0.9508	0.9289	0.3078	0.1
REPAIR	0.9846	0.9727	0.4366	0.1
CANCEL	0.9939	0.9757	0.7469	0.025
REQREP	1	0.9939	1	0.1

Table 4: "Partial Credit" Scores

same four TRAINS-93 dialogues with grounding acts. Our grounding annotator is a computational linguist familiar with the concept of grounding but with no prior knowledge of Traum's coding scheme, the BF coding scheme, or the mapping scheme we were using. Before performing the annotation task, the annotator read Traum's descriptions of the grounding tags, tagged a preliminary dialogue (found in Traum's dissertation), and compared the tags he assigned to those assigned by Traum.

Tables 3 through 6 show the similarity of the human annotator's grounding tags to those automatically derived. The analysis is split into two parts to deal with the ability of annotators to give an utterance multiple labels. Tables 3 and 4 show a per tag analysis. If both the annotators (the human and the Perl script) gave a tag such as INIT to an utterance (in addition to possibly other tags) then it is counted as agreement with respect to the INIT tag. Table 3 shows the number of times a tag appeared and the number of times there was disagreement.

Table 4 shows PA (percent agreement), PE (percent expected agreement), and kappa for each tag. PA is simply the total agreement (either on the presence or absence of a tag in an utterance) divided by the total number of utterances. If N=number of utterances, TotalInit = number of utterances tagged as INIT and TotalNone = number of utterances not tagged as INIT, then $PE = (TotalInit/2N)^2 + (TotalNone/2N)^2$. In this case, there are 2N data points, the two sets of dialogs by the two annotators. Kappa is defined as $K = \frac{(PA-PE)}{(1-PE)}$. See (Car96; SC88) for more details on these measures and the significance levels listed.

Category	Number of Occurrences	Disagree on
INIT	242	40
ACK	225	35
INIT+ACK	101	39
NO-TAG	41	21
INIT+REQACK	18	12
CANCEL	8	2
REPAIR	4	4
INIT+REPAIR+REQACK	2	2
INIT+REQACK+ACK	2	2
REPAIR+ACK	2	2
INIT+REPAIR+ACK	1	1
INIT+REQREP	1	1
REQACK	1	1
REQACK+ACK	1	1
REQREP	1	1

Table 5: "All-or-nothing" Analysis

Table 5 presents the various combinations of grounding tags seen in the corpus. Disagreement is counted whenever two utterances do not have the same exact set of tags. Since the groups of tags are mutually exclusive, we can calculate PA, PE, and kappa over all the tag groups. If agree = utterances where annotators assigned the same set of tags, then $PA = agree/N$. If C_j is the number of times a set of tags such as CANCEL or INIT+ACK was assigned, then $PE = \sum_{j=1}^{15}(C_j/2N)^2$. The definition of kappa remains the same. Given these definitions, PA = 0.7477, PE = 0.2876, and kappa = 0.6458. To help determine where the disagreements occurred, a simple measure of PA was applied to the tag sets, if agreeonTag = cases where annotators agreed on a certain tag and NTag = occurences of tag, then in table 6, $PA = agreeonTag/Ntag$.

The kappa of the "All-or-nothing" analysis is somewhat low compared with the 0.67 standard for tentative conclusions and the 0.8 standard for reliable results as reported in (Car96). The "partial credit" analysis is more favorable as the kappas for

Category	PA
INIT	0.8347
ACK	0.8444
INIT+ACK	0.6139
NO-TAG	0.4878
INIT+REQACK	0.3333
CANCEL	0.75
REPAIR	0
INIT+REPAIR+REQACK	0
INIT+REQACK+ACK	0
REPAIR+ACK	0
INIT+REPAIR+ACK	0
INIT+REQREP	0
REQACK	0
REQACK+ACK	0
REQREP	0

Table 6: "All-or-nothing" Scores

INIT and ACK are close to the 0.8 standard. The grounding tags are somewhat independent; an *init* always starts a new discourse unit whether or not it also acknowledges a previous discourse unit. Thus, the partial credit analysis is likely to be closer to the actual reliability we want to measure. The remaining "partial credit" kappas have low significance levels indicating that more examples are needed to calculate these measures.

Another limitation of this study was that technical papers were used for annotator training rather than an annotation manual designed to explain how tags apply in different situations. This was especially problematic when several tags seemed to apply at once. The BF tags themselves were not perfect as explained in (CA97). Kappas for these annotations varied from the lowest at 0.15 to 0.77 for the highest.

Given these limitations, the results of this experiment are promising. An annotation manual needs to be developed for labeling grounding and more dialogs need to be labeled. When these sources of confusion are addressed, analysis of remaining differences will reveal any minor changes necessary to the mapping.

6 Conclusion

We have presented an automatic mapping from DRI backward- and forward-looking tags to grounding features and discourse units. Our approach assumes simultaneous segmentation into utterance units and annotation of BF tags, which eliminates the need to split or join utterances. The mapping is still being tested but preliminary comparison with a human annotator was promising. Automatic derivation of grounding tags will eliminate the need for separate annotation of grounding, making dialogue annotation quicker and removing a possible source of error.

7 Acknowledgments

This work was supported in part by DARPA grant 5-28838 and by National Science Foundation grants IRI-9503312 and 5-28789.

References

James Allen and Mark Core. *DAMSL: Dialog Act Markup In Several Layers*, 1997. Draft version, available at http://www.cs.rochester.edu/research/trains/annotation/.

Mark Core and James Allen. Annotating dialogs with the damsl annotation scheme. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, 1997.

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

Herbert Clark and Edward Schaefer. Contributing to discourse. *Cognitive Science*, 13, 1989.

Peter Heeman and James Allen. The trains 93 dialogues. Technical report, University of Rochester, 1995.

Teresa Sikorski and James Allen. A scheme for annotating problem solving actions in dialogue. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, 1997.

S. Siegel and N. J. Castellan. *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill, second edition, 1988.

David Traum. *A Computational Theory of Grounding in Natural Language Conversation.* PhD thesis, University of Rochester, 1994.