Annotation Scheme for Authored Dialogues
Version 1.1

Svetlana Stoyanchev and Paul Piwek
Centre for Research in Computing, The Open University, Milton Keynes, UK
{s.stoyanchev, p.piwek} @open.ac.uk

Contents

1 Introduction 3

2 Related work 5

3 Dialogue Annotation and Translation Guide 8

3.1 Definitions (alphabetically ordered) 8

3.2 Using the CODA Annotation Tool 9

3.2.1 Creating New Annotations 10

3.2.2 Editing Annotations 11

3.3 Key and Decorative 12

3.3.1 Primary and Secondary Tags 13

3.4 Dialogue Act Annotation Procedure 13

3.5 Choosing a Dialogue Act 14

3.5.1 Explain 16

3.5.2 Initiating Dialogue Acts 17

3.5.3 Responsive Dialogue Acts 18

3.5.4 Transition Tags 21

3.6 Example 22

4 Discourse Annotation 23

4.1 Definitions 23

4.2 Existing Annotated Corpora 23

4.3 Annotation Procedure 24

4.3.1 Segmentation 24

4.3.2 RST Annotation 25

4.3.3 Deciding between Confusable Relations 27

4.3.4 Scoping 29

4.4 Modifications to the Annotation Guidelines 30

4.4.1 Splitting Monologue into Paragraphs 30

4.4.2 Structure 30
4.4.3 Nuclearity 30

5 File Formats **33**
5.1 Dialogue Act-Annotated Dialogue (FileName.xml) 33
5.2 Mapped Monologue XML (FileName.Monologue.xml) 34
5.3 RST-parsed Monologue (FileName.RSTParsedMonologue.rs3) . 35
5.4 M2D Mapping (FileName.AlignedMonologueDialogue.xml) . . . 35

Chapter 1

Introduction

The annotation we describe here has been developed in the context of the CODA project[1]. The goal of this project is to develop tools and technologies for automatically generating dialogue from monologue. The idea is to create a corpus of aligned monologue and dialogue and automatically derive rules from this corpus for converting monologue into dialogue. Thus a first step towards the overall goal is to create a parallel corpus of monologues and dialogues that express the same information. To create the corpus, we start from authored (i.e., fictive) dialogues and want to:

1. segment the dialogue into dialogue acts (e.g, questions, answers, explanations),
2. label segments with one or more dialogue act types,
3. align the dialogue with a monologue that expresses the same information. To do this we group the segments from the dialogue, such as Question/Answer pair or an explanation and convert it into a monologue, and
4. annotate the monologue with discourse relations (such as explanation, enablement, cause, etc.).

Much of the previous research on dialogue modelling focused on spoken task-oriented dialogue. In this work we look at a different kind of dialogues – authored dialogue with an expository purpose, such as Mark Twain's "What is Man" [10]. Such dialogues are used by the author to convey information or present an argument to the readers. Consequently, we are interested in capturing how authored

[1] See [14] and computing.open.ac.uk/coda.

dialogues present information, make arguments, and which dialogue acts are used by characters in such dialogues.

The form and purpose of authored dialogues differs from naturally-occurring spoken dialogues. For example, if an authored dialogue contains a clarification, its purpose usually is to clarify or emphasize information for the reader, rather than the interlocutors (see [11]). Also, according to our preliminary corpus analysis, most of the clarifications in authored dialogues are on the level of consideration (level 4 of Clark's theory [6]).

We develop a new annotation scheme for a parallel corpus of authored dialogues and the corresponding monologues. In our annotation scheme we are interested to capture how authors present arguments to the readers through the interaction between the dialogue characters. In the following sections, we first review previously developed dialogue annotation schemes and describe our new scheme, which is designed with the purpose of annotating and translating authored dialogues.

Chapter 2

Related work

Researchers have proposed several theories for analysing dialogue. Searle [13] proposed a taxonomy of Illocutionary Acts which classifies an utterance as either assertive, directive, commissive, expressive, or declarative. More recently, Carletta et al. [3] proposed a scheme that captures deep dialogue structure. Dialogues are analysed on three levels: 1) dialogue moves, 2) dialogue games, and 3) dialogue transactions. The scheme has been used to annotate the Maptask [1] corpus of human-human task-oriented communication.

Bunt [2] proposed Dynamic Interpretation Theory (DIT). The main goal of DIT is to provide a computational model of communicative agents. This scheme emphasises that the agents in dialogue aim at the achievement of a communicative goal and each utterance is a contribution towards this goal. The scheme describes dialogue control management including social obligations management, interaction management, and feedback, as well as task-oriented communicative functions.

Core and Allen [7] designed a generic scheme for dialogue analysis (DAMSL). The scheme allows assignment of multiple dialogue acts to a single utterance. The authors point out that an utterance in a dialogue can play multiple roles and have hidden meanings: "an utterance may simultaneously accept information and acknowledge that the information was understood as well as answer a question". Task-oriented human-human (Trains corpus) as well as non-task-oriented human-human dialogues (Switchboard corpus) were annotated with a variation of the DAMSL scheme.

We find a lot of similarities between the previously proposed schemes. For example, the dialogue move level annotations of Carletta et al.'s scheme closely correspond to the Forward and Backward communicative functions of DAMSL scheme and task-oriented communicative functions of DIT. Table 2 shows the correspondence between Carletta's, DAMSL, and our tag set.

DAMSL	Carletta	CODA
forward; statement; assert	init; statement (explain)	Init-Explain
forward; statement; reassert		
forward; other-statement;		
forward; directive; info-requets	init; question; Wh	Init-Factoid-InfoReq, Init-Complex-InfoReq
	init; question; YN	Init-YNQ-InfoReq
forward; directive; action-directive	init; command; instruct	Init-Instruct
forward; commit speaker future action		
forward; performative		
backward; agreement; accept		Resp-Agree
backward; agreement; accept-part		Resp-Agree
backward; agreement; accept-maybe		Resp-Acknowledge-Neutral
backward; agreement; reject		Resp-Contradict
backward; agreement; reject-part		Resp-Contradict
backward; agreement; hold		
backward; understanding; signal-nonunderstanding	resp-communicate-ack	Init-Clarify-Request
backward; understanding; signal understanding; acknowledge		Resp-Acknowledge-Neutral
backward; understanding; signal understanding; repeat-rephrase		
backward; understanding; signal understanding; completion		
backward; understanding; correct misspeaking		
backward; answer	resp-info-yes resp-info-no resp-info-wh resp-info-clarify	Resp-Answer-Yes Resp-Answer-No Resp-Factoid Resp-Explain
backward; information-relation		

Table 2.1: Comparison of Dialogue Act tags from DAMSL, Carletta et al., and CODA annotation schemes. DAMSL and Carletta et al. allow multilevel tagging. ';' separates tags on different levels.

Most of the previous work focused on task-oriented dialogues. The previously proposed dialogue schemes were developed in the context of projects aiming at conversational agents for human-computer interaction or interpretation of human-human communication. In our work, we deal with fictive expository dialogues whose purpose is to present an idea/information to the reader. While previous schemes modelled communication between two agents, we are modelling the flow of information between two agents and its effect on the reader. In particular, we are interested in determining how dialogue acts map to rhetorical structures in monologue.

Expository dialogues are written by a single author. Since they are meant to resemble real dialogues, we borrow most of the dialogue act tags from DAMSL and Carletta et al., adapting them to our purpose. Even though the characters in an authored dialogue do not have a real human-human dialogue, we do occasionally find clarification dialogues in our corpus. These are typically included by the authors to emphasise a particular point (see [11]). We do not include DAMSL's communication level tags in our schema because these clarifications are infrequent in our corpus. Though we do not have multiple dialogue act levels, we do allow multiple annotations of single dialogue act, as proposed by DAMSL. For those cases where an utterance is both a response and an initiation, we allow multiple tags: the annotator can assign both a primary and (if required) a secondary tag to an utterance. However, we find that in our corpus, if an utterance carries multiple functions, it can generally be split into segments that each carry a different function. For example, *yes, the mind ...* can be split into a positive answer *yes* and an explanation *the mind ...* which is an explanation dialogue act.

Chapter 3

Dialogue Annotation and Translation Guide

3.1 Definitions (alphabetically ordered)

- **Decorative** information, as opposed to key information, is included by the dialogue author to liven up the dialogue and create natural transitions between segments (see definition below) of key information (see also the definition below).

- "**Dialogue act**" is synonymous to "**move**"

- **Informational Unit (IU)** A set of segments which together translate to a single *snippet*. An IU should satisfy the following constraints:

  – An IU should not contain a proper subset of segments which could also translate to a snippet. In other words, IUs should be as small as possible.

  – Every segment in a dialogue should belong to exactly one IU. Because each IU corresponds with exactly one snippet, every segment in a dialogue maps to exactly one snippet.

  – If two segments cannot be translated into a monologue snippet independently of each other, they should belong to the same IU.

  Examples of IUs are:

  – An init dialogue act followed directly by a response dialogue act.

  – A single dialogue act, typically a key *explain* act.

The following shows a schematic example of a dialogue that has been partitioned into two IUs. The two question-answer pairs each constitute an IU.

1. A: [Question (init)
2. B: Answer (response)]$_1$
3. [Question (init)
4. A: Answer (response)]$_2$

- **Key** information in an expository dialogue contributes to the main information/ideas that the author is trying to get across; it is directly relevant to the main purpose of the dialogue. It should be present in the corresponding monologue. Generally, key information contains concept words.

- **Monologue Sentence** Sentences in monologue should be marked by punctuation. Sentence boundaries do not need to correspond to snippet boundaries.

- **Segment**: substring of a turn that is either labelled key or decorative information. A segment of key information typically consist of a dialogue act (or two dialogue acts in case the segment expresses at the same time a forward and a backward looking dialogue act). Each substring of a dialogue should be part of exactly one segment. If two consecutive segments of the same speaker have the same dialogue act and map to the same monologue snippet, they should be merged.

- **Snippet**: a monologue translation of a set of segments which together form an *informational unit*. Snippet boundaries should coincide with elementary discourse unit boundaries (EDUs) of the rhetorical structure of the text – each snippet boundary should be an EDU boundary, *but* not each EDU boundary has to be a snippet boundary.

- **Turn** : everything dialogue participant *A* says before dialogue participant *B* takes over.

Figure 3.1 shows a schematic example of aligned monologue snippets and dialogue segments.

3.2 Using the CODA Annotation Tool

We have created a dedicated tool for annotating authored dialogue and translating it into aligned monologue. This section describes how the tool is used to create annotated dialogue and aligned monologue.

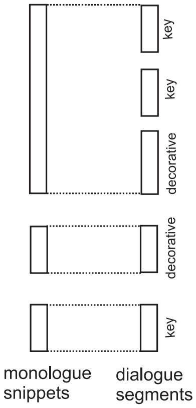

Figure 3.1: Schematic example of aligned monologue snippets and dialogue segments. Note that the first snippet aligns with three segments, which together constitute an Informational Unit (IU)

3.2.1 Creating New Annotations

1. Open a dialogue file: Dialogue | open

2. Tag several turns

   (a) Click on a cell in Tagged Turn column.

   (b) Highlight a segment (part of a Turn).

   (c) Select a Type of the segment(Refer to Section 3.3).

   (d) Select a Dialogue Act of the segment (Refer to Section 3.4).

   (e) Click *add* to add a segment to the segments that have been identified and tagged so far.

   (f) Repeat above steps until the turn has been fully segmented and click *save* to save the annotation of this turn.

3. Write a monologue snippet:

   (a) Identify a set of segments that together form an Information Unit (IU).

   (b) Enter the ids of these segments into the “Mapping index” text box (e.g., “1 4”; ids are separated by a space).

   (c) Click *Insert_segments*: the mapped segments are copied into the snippet editing window.

(d) Edit snippet's text in the snippet editing window to create a fluent monologue text.

(e) Click Add to insert a monologue snippet.

4. Save dialogue annotations: File | save dialogue

5. Save monologue annotations: File | save monologue

3.2.2 Editing Annotations

Changing dialogue act

To change a dialogue act of a segment or to add a secondary dialogue act tag use one of the methods:

1. Edit the XML file directly in a text editor.

2. Open the dialogue in the D2MTool. Click on the segment that needs to get a tag change. Change the tags, click add/change button. The change should appear in the segment display window. Click save.

Adding a segment

To add a segment, the D2MTool should be used (to preserve the mapping of monologue snippets to the dialogue segments):

1. Open the dialogue in the D2MTool,

2. Open corresponding monologue in the D2MTool,

3. Click on the turn that needs further segmentation

4. Remove or add segments

5. Click save to save the turn segmentation. The new segment is not mapped to any of the dialogue snippets. All the previous mappings are preserved (segments following the new segment get their ids incremented)

6. To map the new segment to a new monologue, select a monologue snippet before the new snippet, follow the steps described above to add a new snippet. Alternatively, edit XML in a text editor.

Modifying monologue

Edit XML directly or use D2M Tool: select the monologue snippet, click on view/edit, edit the text, click save.

3.3 Key and Decorative

In a dialogue, the interlocutors exchange information with each other. Information that is directly relevant to the main purpose of the dialogue is classified as *key information*. For instance, in a dialogue which consists of a discussion about some topic (say, whether holes exist as material objects), information that presents either side of the argument is key information. Only key information is translated into monologue.

Decorative segments perform what Bunt calls dialogue control functions. This includes social talk (phatic communication) and meta-communication. Decorative segments are not translated into monologue.

Most utterances in an authored dialogue present *key information*. They are about the topic of the dialogue and their meaning needs to be preserved in the monologue. They can be copied verbatim to the monologue or paraphrased. There are, however, also utterances which do not contain key information; in terms of Dynamic Interpretation Theory (DIT) [2] they express 'dialogue control functions'. From the point of view of an author of fictional dialogue they are often used to create a certain effect in the audience: e.g., the creation of a mood, attracting attention of the reader, or embellishing a dialogue. Such utterance express *decorative information*. Examples of decorative utterances are utterances for managing turn taking such as *Wait!* or *Just a moment*. Decorative segments also include exchanges which concern the *social context* of the dialogue. For example, decorative social dialogue may be found at the beginning or end of a fictional dialogue where characters establish acquaintance or say farewell. Decorative utterances can also occur in the body of a dialogue (as opposed to the beginning and end).

To test if a segment is key or decorative, answer the question: Can the information of the segment be omitted, without affecting the interpretation of the key information? If yes, the segment is decorative,otherwise it is key.

A decorative segment is underlined in the following example:

A: *For instance ?*
B: *Well, then, for instance.*
B: *Take the case in the book here ...*

Another example of a decorative statement is when one speaker praises the other:

Good job; Well done, etc.

3.3.1 Primary and Secondary Tags

In the DAMSL annotation scheme [7] the authors note that that an utterance in a dialogue may carry multiple communicative functions. An utterance can be an answer to a question, and at the same time it may be initiating a new idea. The authors differentiate between possibly co-existent *Forward* and *Backward* communicative functions of an utterance. Our annotation scheme allows multiple dialogue acts, similarly to DAMSL. However, to simplify the annotator's job we limit the allowed tag values.

The annotators of authored dialogues are required to annotate primary function and have an option of annotating secondary function. The values available for the secondary function of a segment depend on the value chosen for the primary function of this segment. For example, if a primary function of a segment is *Explain*, the only available secondary functions are *Response-Agree* and *Response-Contradict*. Alternatively, we could have required annotators to choose one value from *Forward* and/or *Backward* function which would yield the same end-result. Our choice was motivated by the goal of reducing the mental strain on annotators.

3.4 Dialogue Act Annotation Procedure

Segment an utterance such that each *key* segment has a single primary dialogue act. Then assign primary and secondary (when applicable) dialogue acts to the segment. If two consecutive segments in a single turn have the same primary and secondary dialogue act, merge them.

1. If the primary purpose of a segment is to initiate an Informational Unit (IU)[1], assign the most appropriate *init* primary tag. Usually but not necessarily:
   - An answer/response to the segment follows in the next turn.
   - This is the last segment of a turn.
2. If the primary purpose of a segment is to respond to a previous utterance, assign the most appropriate *response* primary tag. Usually but not necessarily:
   - The next speaker's utterance is not a response to the current utterance.

[1] For our purposes, an IU generally is a init-response adjacency pair.

3. If the primary tag of the segment is *Explain, Init-YNQ, Init-ComplexQ*, and this segment also serves as a response (agreement or contradiction) to the previous utterance, assign the secondary tag *Resp-Agree* or *Resp-Contradict*.

4. If the primary tag assigned is *Resp-Agree, Resp-Contradict, Explain*, and if the utterance is followed by a response or an answer, assign a secondary tag of *Explain*.

3.5 Choosing a Dialogue Act

Figures 3.2 and 3.3 illustrate the taxonomy for Initiating and Responsive dialogue acts.

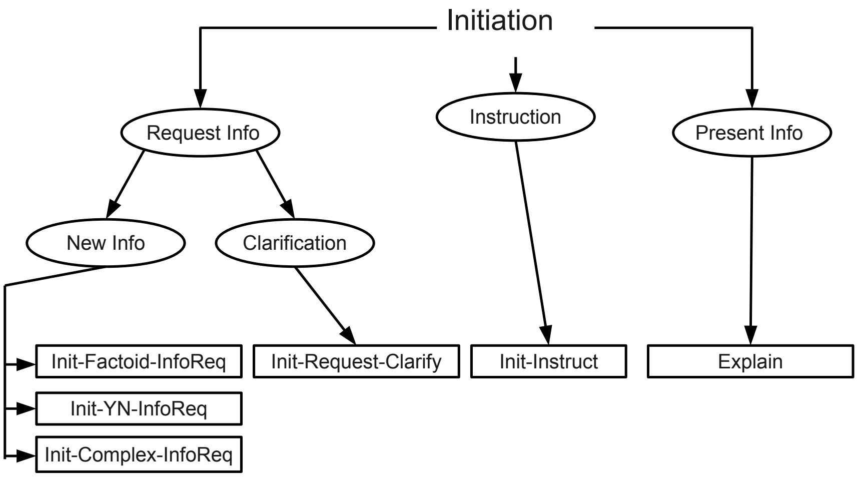

Figure 3.2: Taxonomy of initiating dialogue acts.

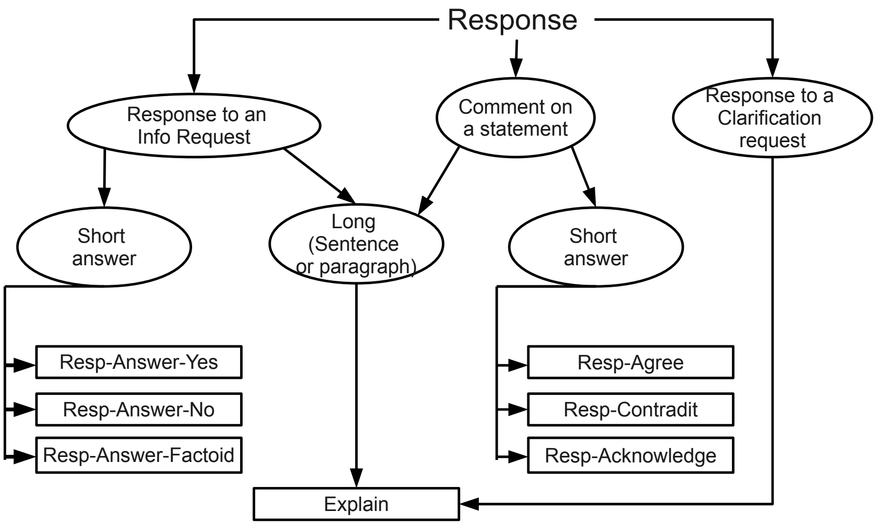

Figure 3.3: Taxonomy of responding dialogue acts.

3.5.1 Explain

Explain can be either initiation or response. In our dialogue annotation scheme we do not differentiate between *Init-Explain* and *Resp-Explain*.[2] An *Explain* dialogue act states key information that also should be presented in the corresponding monologue. The act may describe an opinion, for example:

> *The mind is independent of the man.*

It may also be an example given by one of the characters, such as:

> *For example, the algorithm may be written in a high level language that allows, say, multiplying integer matrices in one step.*

Explain can be a long on-going explanation:

> *The others offer your a hundred bribes to be good, thus conceding that the Master inside of you must be conciliated and contented first, and that you will do nothing at FIRST HAND but for his sake; then they turn square around and require you to do good for OTHER'S sake CHIEFLY; and to do your duty for duty's SAKE, chiefly; ...*

Assign this act if all of the following are true:

- This segment is a statement, a set of statements (it may contain rhetorical questions as well).
- The primary purpose of the segment is to initiate a new interaction.
- The segment contains key information: if this segment is not present in the monologue, some important information is lost.

Below, the underlined segment gets a primary tag of *Explain* because it explains a new idea. It gets a secondary tag of *Response-Agree* because it supports the yes answer to the previous question, so it also serves a *backward looking* function:

A: *And that his mind works automatically and is*
independent of his control – carries on thought on its own hook?
B: *Yes* . It is diligently at work , unceasingly

[2] Because of the low inter-annotator agreement

at work, during every waking moment. If it needed the man's help
it would wait for him to give it work when he wakes in the morning.

A: *Maybe it does.*

3.5.2 Initiating Dialogue Acts

Initiating dialogue acts correspond to DAMSL's *forward* looking communicative function and to *Initiation* moves of Carletta et al.

Init-Factoid-InfoReq

Assign this tags if all of the following are true:

- The segment is a question or an imperative such as "Tell me ..." that requests information (who, when, where, what, how many, etc.)
- It can not be answered with yes/no
- The most natural answer to the question is a simple factoid answer (and not elaboration or explanation), such as a name or a number. Although sometimes the character may elaborate on factoid questions, hence do not use the actual answer to judge.
- The segment is not why or how question.
- If it is a *what* question, it does not ask about the complex issues, such as cause, effect, etc.

A possible secondary dialogue act is *Init-Request-Clarify*. Assign this secondary dialogue act when the factoid question is asked with the intention of clarifying the previous utterance.

Init-Complex-InfoReq

Assign this tag when all of the following are true

- The segment is a question.
- The segment is not a factoid question.
- The segment is not a Yes/No question.
- The segment is not a rhetorical question. [3]

[3] Rhetorical questions are annotated with *explain*

Init-YN-InfoReq

Assign this tag when all of the following are true

- The segment is a question
- The segment can be answered with yes or no.

Possible secondary dialogue acts are *Response-Agree*, *Response-Contradict*.

Init-Clarify-Request

A clarification request always plays the dual role of an information request as well as a response to previous utterance. A speaker asks the other speaker to clarify, for example:

How do you mean?

Explain

Assign this tag when the segment is either a statement or a question that illicits clarification or additional information.[4]
Possible secondary dialogue act is *Init-FactoidQ* or *Init-YNQ*.

3.5.3 Responsive Dialogue Acts

Responsive dialogue acts correspond to DAMSL's *backward* looking communicative function and to *Response* moves of Carletta et al.

Response-Answer-Yes

A positive answer to a yes/no question:

Yes!

Oh yes, there is.

it is true.

[4] If a segment asks for a clarification and carries information, if possible, it should be split into two segments.

Perhaps so.

Assign this statement when:

- Previous dialogue act is *Init-YNQ*
- The segment can be replaced by 'Yes' without the loss of information

This act is likely to be followed by *Explain* dialogue act:

A: *Oh yes, there is. It is candidly stated, this time. That has not been done before.*
B: *Well, yes, that is a difference, it is true.*

Possible secondary dialogue act is *Explain*. Consider an example

A: *Is it raining?*
B: *Cats and dogs*

B's response is a positive answer to a yes/no question as well as an explanation. It should have the primary tag *Resp-Answer-Yes*, and *Explain* as a secondary tag.

Response-Answer-No

A negative answer to a yes/no question. Assign this tag when:

- Previous dialogue act is *Init-YNQ*
- The segment can be replaced with "no" without loss of the information

Possible secondary dialogue acts is *Explain*. Consider an example

A: *Is it raining?*
B: *It is snowing*

B's response is a negative answer to a yes/no question as well as an explanation. It should be marked as *Resp-Answer-No* as a primary tag and *Explain* as a secondary tag.

Response-Agree

A statement that shows an agreement with the previous statement, for example

You are right. It is true.

Assign this tag when:

- Previous dialogue act is *Explain*
- The segment indicates agreement or support to the previous utterance

There is an overlap with *Response-Yes* dialogue act. The main difference is that *Response-Yes* is preceded by a yes/no question, while *Resp-Agree* is preceded by a statement.

A possible secondary tag is *Explain*. In the following example, B both agrees with A and initiates an explanation (although it is questionable which tag should be primary and whicha secondary):

A: The air is warm
B: It is warm and humid.

Response-Contradict

A statement that shows a disagreement or contradiction with the previous statement, for example

I don't know where you got it from
This can not be true

Assign this tag when:

- Previous dialogue act is *Explain*
- The segment indicates disagreement or contradiction with previous utterance

Possible secondary tag is *Explain*.

Response-Acknowledge-Neutral

Example:

Maybe it does.

Assign this tag when:

- Previous dialogue act is *Explain*
- The segment indicates acknowledgement of the previous statement without clear agreement or contradiction

Possible secondary tag is *Explain*.

Response-Factoid

Assign this tag when:

- Previous dialogue act is *Init-FactoidQ*
- The segment is a simple factoid answer to the question.
- The segment contains key information

3.5.4 Transition Tags

Transition tags are the segments that link together the presentation of an idea into a coherent dialogue. Transitions are by definition decorative. They are part of dialogue but not the equivalent monologue. If a transition segment is removed, no information is lost. So, transitions are special types of decorative segments. We identify two types: Pause and Praise.

Trans-Pause

Segments that indicate a pause

At any rate; Anyway; Well; So; etc.

We are interested in capturing these to be able to generate the dialogue from a monologue.

Other

If none of the tags applies, a segment is annotated as *other*.

3.6 Example

Consider an example:

A: *At any rate , he can make it stick to a subject if he wants to*

B: *Not if it finds another that suits it better. As a rule it will*
listen to neither a dull speaker nor a bright one. It refuses all persuasion.

Segmentation and annotations of the above utterances:

1. 'At any rate': **decorative**
2. 'he can make it stick to a subject if he wants to': **key, Explain**
3. 'Not if it find another that suits it better': **key, Resp-Contradict**
4. 'As a rule it will listen to neither a dull speaker nor a bright one. It refuses all persuasion.': **key, explain**

The first segment is not converted into a monologue, it is a decorative statement used by the speaker A in order to emphasise his level of commitment, bring attention to the utterance. Segments 2, 3, and 4 are *key* as they carry information that will be present in the corresponding monologue.
Monologue sentence paraphrase:

> *He can make it stick to a subject, but not if it finds another that suits it better.*

Monologue snippets and their mapping:

1. he can make it stick to a subject **segment 1, 2**

2. but not if it finds another that suits it better **segment 3**

3. As a rule it will listen to neither a dull speaker nor a bright one . It refuses all persuasion **segment 4**.

Chapter 4

Discourse Annotation

4.1 Definitions

- Elementary Discourse Unit (EDU)[1]. A segment of a monologue that participates in a discourse relation with another segment and contains no internal discourse relations.
- Multiple Discourse Unit (MDU). A sequence of multiple EDUs that map verbatim into a single dialogue segment.

4.2 Existing Annotated Corpora

Most large scale discourse annotation efforts have focussed on news articles, typically of the sort found in the Wall Street Journal (Carlson et al. [5], Wolf and Gibson [15], Penn Discourse Treebank [12]). Whereas most of this work has assumed that the structure of annotations is tree-shaped, Wolf and Gibson have departed from this assumption, allowing crossing dependencies. The number of relations that is used varies widely: e.g., 78 were used by Carlson et al. (grouped into 16 more general types of relation), three levels were used for the PDTB (4 relation at the most general level, 16 at the level below that, and 29 on the most specific level), and the HILDA discourse parser [8] uses 18 relations (16 from Marcu's general relations + Same-Unit and Textual-Organisation). Note that the work on the PDTB differs from the other efforts by focussing on disambiguating/interpreting the meaning of connectives, rather than uncovering (possibly implicit/inferred) relations in a text.

[1] As defined in the RST annotation scheme of Daniel Marcu and collaborators [5].

4.3 Annotation Procedure

In this section we describe the process for annotating Rhetorical structure on the manually written monologues. For the M2D translation project we perform *partial annotations*. For the purpose of the M2D translation some of the relations inside the monologue are irrelevant: relations inside the snippets that are mapped 1-to-1 to a dialogue segment and copied verbatim will not be present in the M2D rules. We call these snippets multiple discourse units (MDU) and do not split them further. Identifying MDUs allows us to save time on annotation while annotating all discourse relations that contribute to the M2D transformation rules. The relations inside an MDU are internal segment relations and will not be part of translation.

4.3.1 Segmentation

Steps for segmenting monologue into EDU and MDUs:

1. Initially segment monologue using monologue snippets.
2. For each snippet, if it contains multiple EDUs and is not an MDU, segment it further: create an EDU boundary where Marcu's segmentation rule applies.

For example, the following four segments of a dialogue

1. A: key:Init-YN-InfoReq:none: Did the man possess it who gave the old woman his last shilling and trudged home in the storm?
2. B: key:Resp-Explain:none: He had the choice between succouring the old woman and leaving her to suffer.
3. key:Init-YN-InfoReq:none: Isn't it so?
4. A: key:Resp-Answer-Yes:none: Yes,

are translated into a single monologue snippet:

> *The man who gave the old woman his last shilling and trudged home in the storm had the choice between succouring the old woman and leaving her to suffer.*

As the monologue snippet contains EDUs and is not an MDU, during manual annotation it is split as indicated by the "|":

> *The man | who gave the old woman his last shilling and trudged home in the storm | had the choice between succouring the old woman and leaving her to suffer.*

The goal of manual segmentation is to 1) add segment boundaries inside snippets in order to define relations between them and 2) fix the text of the monologue snippet 3) fix the existing boundaries of the snippets.

To add a new boundary, simply add it.

During the RST annotation the annotator has an opportunity to improve manual translation of dialogue to monologue by finding mistakes, inconsistencies, and style of the monologue. If you see a mistake or something that may be expressed in a better way, fix it here.

Fixing boundaries is necessary in rare cases when the snippet boundary mistakenly crosses an EDU boundary remove the boundary and if necessary add it in a correct location.[2] In the following example, the snippet boundary crosses an EDU:

> *I would not use those words – Free Will – but others:* | *Free Choice*

To fix the segmentation, remove the boundary before *Free Choice* and add a boundary before *but* for a contradiction relation between the two segments:

> *I would not use those words – Free Will –*| *but others : Free Choice*

4.3.2 RST Annotation

In our annotation scheme we were guided by the annotation scheme defined by Carlson et al. [5]. We use the grouped list of RST tags as the basis for the annotation scheme. We choose to expand the most frequent relations in our data (Evaluation, Elaboration, Explanation, and Topic-comment) into their sub-relations. For these relations the annotator has an option of selecting a fine-grained tag or high level tag. This decision was inspired by the method of Penn Treebank annotations where annotators chose granularity of their labels. An additional advantage to using fine-grained tags for the above relations is that it will force annotators to think more thoroughly and disambiguate between explanation, evaluation, and elaboration, the three most common and most confusable relations in our corpus, according to our initial observation. We did not expand the rest of the relation tags because 1) they are not as frequent 2) the semantics of the sub-tags is closer to each other. Some of the tags were merged or renamed as indicated in the footnotes. We found in several trials, that the new names are a better representation of the meaning of the tag. The following tags are used in the discourse annotation of CODA corpus:

1. Explanation (S explains N)

[2] This hardly ever happens and arises from incorrect segmentation during dialogue-to-monologue translation, as the annotators were instructed to create snippets matching EDU boundaries.

- Evidence (Evidence is satellite)[3]
- Reason (Reason is satellite)

2. Enablement (S enables N)

3. Cause (S causes N)[4]

4. Evaluation (S evaluates N)

   - Subjective[5]
   - Inferred[6]
   - Comment (Comment is satellite)

5. Attribution (S is the person, organization, etc. to whom N is attributed)

6. Condition-Hypothetical (S is condition for N)

7. ContrastMono[7] (Although/despite of S, N)

8. Contrast[8] (multinuclear)

9. Comparison (multinuclear)

10. Summary (S summarizes N)[9]

11. Manner-means (S is means or manner of achieving N)

12. Topic-Comment

    - problem-solution (Solution is satellite)
    - Statement-response (Response is satellite)
    - Question-Answer (Answer is satellite)
    - Rhetorical Question (Question is satellite)

13. Background (Background info is satellite)

[3] Merged Carlson et al. [5]'s Evidence and Explanation-Argumentative.

[4] This order was changed form Marcu's definition for consistency and ease of annotation purposes.

[5] Corresponds to Marcu's Assessment and Interpretation.

[6] Corresponds to Marcu's Conclusion.

[7] Corresponds to Marcu's concession and antithesis.

[8] Corresponds to Marcu's multinuclear Contrast.

[9] In contrast with Evaluation-Inferred, the Nucleus of a summary has a larger scope. Because of the limit on the annotation level in our corpus, the nucleus of the summary most often is a DUMMY node.

14. Temporal (when S, N)

15. Elaboration

    - Additional (Additional information is satellite)
    - General-Specific (specific information is satellite)
    - Example (Example is satellite)
    - Object-attribute (Attribute is satellite)
    - Definition (Attribute is satellite)

16. Same-unit (multinuclear)

In the RST annotation manual, Marcu et al. [4] note that "In some cases, more than one relation may hold between two textual segments. For example, a causal and a temporal relation may hold between two segments simultaneously". The authors of the RST annotation manual define the order of applying the relations according to their specificity. More specific relations are applied first. Relations in the list above are ordered from more to less specific and should be applied in this order.

4.3.3 Deciding between Confusable Relations

From the initial annotations we identified a highly confusable set of relations: Elaboration, Explanation, and Evaluation. Table 4.1 is designed to guide the annotator for making a decision between the tags.

Elaboration-Additional is a relation that applies in most cases because every sentence presents some new information. It often coincides with evaluation-inferred. To disambiguate the two relations an annotator should apply the following test:

A relation *Evaluation-Inferred(A, B)* holds between the statements A and B if given A, one is in the position to answer *is B true?* In other words, annotate *evaluation-inferred* rule if B can be inferred from A. On the other hand, if one can not answer *is B true?* given A, then the relation may be *Elaboration-Additional(A, B)* or *Evaluation-Subjective(A, B)*. Annotators use table 4.1 to disambiguate.

Examples

Example 1:

St1 The man who gave the old woman his last shilling and trudged home in the storm had the choice between succouring the old woman and leaving her to suffer.

Rule-name	Rule
Elaboration vs. Explanation vs. Evaluation	
Explanation-Reason	Sat. provides a logical explanation for the situation presented in the nucleus.
Explanation-Evidence	Sat. provides a factual explanation for the situation presented in the nucleus.
Evaluation-Subjective	Sat. is subjective, presenting the personal opinion of the writer or of a third party; or an assessment of the relationship on a scale of good to bad.
Evaluation-Inferred	Sat. can be inferred from Nucleus.
Evaluation-Comment	Sat. presents a comment statement.
Elaboration-Additional	Sat. gives additional information or detail about the situation presented in the nucleus.
Elaboration-Definition	Sat. provides specific information to help define a very general concept introduced in the nucleus.
Elaboration-Obj-Attribute	Sat. describes an object mentioned in the nucleus.
Elaboration-Example	Sat. provides an example to illustrate a statement in the nucleus.
Elaboration-General-Specific	Nucleus is a general statement and sat. provides specific information.

Table 4.1: Reference rules for differentiating relations.

St2 There was a choice to be made, between bodily comfort on the one hand and the comfort
of the spirit on the other. The body made a strong appeal, of course –
the body would be quite sure to do that; the spirit made a counter appeal.
A choice had to be made between the two appeals, and was made.
St3 One could say that the man determined free choice, and that in doing it he exercised Free Will.

At a first glance, the relation between St1 and St3 can be evaluation or elaboration. To determine which one it is, try to apply the rules from Table 4.1. The rules that apply are:

- *Elaboration-Additional* (St3 adds information to St1)
- *Evaluation-Subjective* (St3 presents personal opinion on St1)

The evaluation relation has a higher level of specificity than Elaboration as it appears earlier in the list of rules (see Section 4.3.2), therefore the annotator should mark this relation as *Evaluation-Subjective*. Note that if an annotator was confident that relation is evaluation but was not confident of the type of the evaluation, s/he can mark it as *Evaluation*.

The relation between St1 and St2 is *Explanation-Evidence* because St2 provides factual information about St1.

Example 2:
St1 Then if a man KNOWS which of two things is right he is absolutely
BOUND to do that thing.
St2 His temperament and training will decide what he shall do, and he will do it;
he cannot help himself, he has no authority over the mater.

Rules apply to Relation(N=St1, Sat=St2):

- *Elaboration-Additional* (St2 adds information to St1)
- *Explanation-Evidence* (St2 provides factual information about St1)

Explanation has a higher level of specificity than Elaboration, therefore the relation is *Explanation-Evidence*.

4.3.4 Scoping

To determine scope of rule application, see Marcu and Carlson's annotation manual [4] (page 33): section 4.2 on "Higher Level Organization of Rhetorical Relations".

4.4 Modifications to the Annotation Guidelines

This section defines CODA modifications to the Marcu's RST annotation guidelines.

4.4.1 Splitting Monologue into Paragraphs

In the CODA corpus we create a mapping between sequences of dialogue segments and monologue paragraphs. We define a paragraph as a meaningful and coherent presentation of an idea that is understandable as a stand-alone segment of text. We confirm the quality of the paragraph splitting during a manual validation procedure. In the manual validation procedure, the non-annotator reads through the parallel dialogue and monologue and identifies incoherent segments.

By splitting the monologue into paragraphs we limit the *RST relation depth*, or the distance from the root node to its most distant child. Limiting the RST relation depth is important because we found that the inter-annotator agreement on RST annotations is reasonably high for shallow relations (closer to the leaf nodes) while it is very low for deep relations (closer to the root). By avoiding to annotate deep relations we are improving reliability of our RST annotations.

4.4.2 Structure

We observe that in some cases, we may have a good idea about the span of one side of a rule but not the other. For example, an annotator may be confident of a span for a satellite of a summary or evaluation relation but less confident of a span fore a nucleus (text is being summarized or evaluated) of this relation. We may choose to avoid creating a relation, but then we lose some information about structure. To address this, we create a DUMMY EDU and connect the satellite to it. See illustration on Figure 4.1 where *Explanation-reason* relation is annotated with a DUMMY node as a nucleus and RST structure as a satellite.[10]

4.4.3 Nuclearity

Each RST relation is either multinuclear or mononuclear. In Carlson et al.'s annotation scheme the role of nucleus plays a dual role, in some cases it is semantic, or attributed to the role of the discourse constituent (e.g. evaluator vs. evaluated) while in other cases it is used to indicate which of the constituents is "more important". In CODA's RST annotations the role of nucleus is always driven by semantics. For example, in a cause relation *satellite* always causes *nucleus* and

[10] A DUMMY node may play the role of either a nucleus or a satellite.

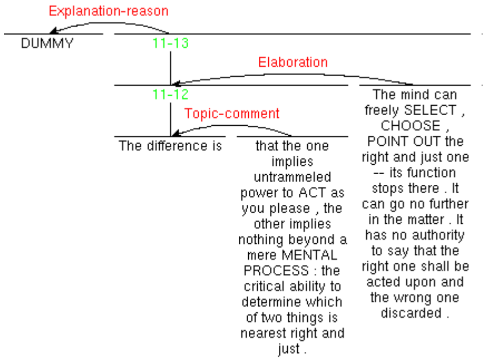
Explanation-reason
DUMMY
11-13
Elaboration
11-12
The mind can freely SELECT , CHOOSE , POINT OUT the right and just one -- its function stops there . It can go no further in the matter . It has no authority to say that the right one shall be acted upon and the wrong one discarded .
Topic-comment
The difference is
that the one implies untrammeled power to ACT as you please , the other implies nothing beyond a mere MENTAL PROCESS : the critical ability to determine which of two things is nearest right and just .

Figure 4.1: Example of RST rule with DUMMY EDU.

in evaluation relation *satellite* always evaluates *nucleus*. By attributing nuclearity to semantics, we simplify the annotation at the cost of not annotating the "most important" constituent in a relation.

Chapter 5

File Formats

This chapter describes the file formats produced for the CODA corpus including three formats for the files produced during manual annotation: *dialogue act-annotated dialogue, mapped monologue, RST-annotated monologue*; and a format for the files derived automatically from the annotations: *M2D mapping* format.

5.1 Dialogue Act[1]-Annotated Dialogue (FileName.Dialogue.xml)

Dialogue with annotated segments is an output from the D2MAnnotationTool described in Section 3.2. A dialogue is represented as a sequence of dialogue turns. Each turn contains a *SPEAKER* attribute which indicates the name of the character that performs this turn and a *SPEECH* attribute which contains the text of the turn. A turn has a *SEGMENTS* element child. *SEGMENTS* contain one or more *SEGMENT* elements. Together the segments should span the whole turn. The attributes for each *SEGMENT* are: a unique identifier for the segment in the dialogue *SEGMENT_ID*, and *SEGMENT_START* and *SEGMENT_END* to indicate first and last character of the segment in the *SPEECH* of the turn. *SEGMENT_MOVE, SEGMENT_SECONDARY_MOVE*, and *SEGMENT_TYPE* were assigned by the annotator. *SEGMENT_MOVE* and *SEGMENT_SECONDARY_MOVE* record the primary and (optional) secondary dialogue act of the segment. The *SEGMENT_TYPE* tag allows us to distinguish between key and decorative segments. For convenience, *SEGMENT* also includes *SPEAKER* and text of the segment (although this can be derived from the turn and start and end tags).

```
<TURN SPEAKER="Y.M." SPEECH="No -- in ores . ">
<SEGMENTS>
```

[1] Dialogue Act and Move are used interchangeably

```
   <SEGMENT SEGMENT_END="5"
         SEGMENT_ID="5"
         SEGMENT_MOVE="Resp-Answer-No"
         SEGMENT_SECONDARY_MOVE="none"
         SEGMENT_SPEAKER="Y.M."
         SEGMENT_START="0"
         SEGMENT_TYPE="key">
         No --
   </SEGMENT>
   <SEGMENT SEGMENT_END="15"
       SEGMENT_ID="6"
       SEGMENT_MOVE="Resp-Factoid"
       SEGMENT_SECONDARY_MOVE="none"
       SEGMENT_SPEAKER="Y.M."
       SEGMENT_START="6"
       SEGMENT_TYPE="key">in ores .
   </SEGMENT>
<COMMENT/>
</SEGMENTS>
</TURN>
```

5.2 Mapped Monologue XML (FileName.Monologue.xml)

This file format describes mapping of monologue snippets to dialogue segments. The files in this format are generated by the D2MAnnotationTool described in Section 3.2.

A mapped monologue is a list of *SNIPPET* elements with a *TEXT* attribute that contains the text of the monologue snippet. *MAPPINGLIST* associated with each *SNIPPET* contains indexes of the *SEGMENT_ID*s of the corresponding annotated dialogue file.

```
<SNIPPET TEXT="These are found in the rocks not
               in a pure state but in ores . ">
    <MAPPINGLIST>
        <MAP>2</MAP>
        <MAP>3</MAP>
        <MAP>4</MAP>
        <MAP>5</MAP>
        <MAP>6</MAP>
    </MAPPINGLIST>
```

```
</SNIPPET>
```

5.3 RST-parsed Monologue (FileName.RSTParsedMonologue.rs3)

CODA annotators use the RST-Tool [9] to produce RST-parsed monologue.

5.4 M2D Mapping (FileName.AlignedMonologueDialogue.xml)

The files of the M2D Mapping format are automatically derived from the files in the above three formats. The AlignedMonologueDialogue file is a collection of *M2D_PARAGRAPH2SEGMENTS*. Each *M2D_PARAGRAPH2SEGMENTS* element encodes a mapping between a monologue paragraph *MONOLOG*, a sequence of dialogue acts *VERBALIZED_DIALOGSTRUCT*, and *MONOLOGSTRUCT* element. The *COMMENT* attribute indicates the file name and snippet range from which this mapping was derived. *STARTSNIP* and *ENDSNIP* indicate a range of snippet sequence numbers in the monologue for reference purposes.

A *Segment* element under the *MONOLOG* parent indicates an elementary discourse unit. The attributes *id*, *parent*, and *relname* are derived from the RST-annotated monologue. They indicate RST relation between the segments. The *Order number* attribute is a segment sequence number within the paragraph, starting with 0. The *Snippet_id* corresponds to the snippet index (this will be mapped from the dialogue). We introduce snippet_id because the annotators mapped dialogue segments to monologue snippets.

A *VERBALIZED_DIALOGSTRUCT* element encodes dialogue sequence mapped to the monologue. Similarly to the annotated dialogue format, the dialogue is represented as a sequence of *SEGMENT*s with mostly the same attributes. SEGMENT_ID in this structure is a sequence number within the *VERBALIZED_DIALOGSTRUCT*, starting with 0. *SNIPPET_ID* is the link from the segment to the monologue *Segment*s with identical *Snippet_id*s.

The *MONOLOGSTRUCT* element describes the monologue structure: *PARENT* indicates the RST relation of the top element in the structure and *RELATIONROLE* is either *nucleus* or *satelite*. *PARENT* and *RELATIONROLE* are empty when the monologue structure does not have external RST links annotated. The *TREE* attribute describes the RST relation structure between the monologue segments in a human-readable format. The indexes in *TREE* attribute correspond to *OrderNumber*s of the *Segment*s.

```
<M2D_PARAGRAPH2SEGMENTS COMMENT="0-1(Twain-part1_1)"
                  ENDSNIP="1" STARTSNIP="0">
```

```
<MONOLOG>
   <Segment id="0" orderNumber="0" parent="-1"
   relname="" snippet="0">
       A steam - engine is made of materials such as iron ,
       steel , brass , white - metal , and so on .
   </Segment>
   <Segment id="1" orderNumber="1" parent="0"
   relname="elaboration-obj-attribute" snippet="1">
       These are found in the rocks not in a pure state  but in ores .
   </Segment>
</MONOLOG>
<VERBALIZED_DIALOGSTRUCT>
  <SEGMENT SEGMENT_END="59" SEGMENT_ID="0"
   SEGMENT_MOVE="Init-Factoid-InfoReq"
   SEGMENT_SECONDARY_MOVE="none" SEGMENT_SPEAKER="O.M."
   SEGMENT_START="0" SEGMENT_TYPE="key"
   SNIPPET_ID="0">
      What are the materials of which a steam - engine is made ?
  </SEGMENT>
  <SEGMENT SEGMENT_END="51" SEGMENT_ID="1" SEGMENT_MOVE="Resp-Factoid"
  SEGMENT_SECONDARY_MOVE="none" SEGMENT_SPEAKER="Y.M."
  SEGMENT_START="0" SEGMENT_TYPE="key"
  SNIPPET_ID="0">
      Iron , steel , brass , white - metal , and so on .
  </SEGMENT>
  <SEGMENT SEGMENT_END="24" SEGMENT_ID="2"
  SEGMENT_MOVE="Init-Factoid-InfoReq"
  SEGMENT_SECONDARY_MOVE="none" SEGMENT_SPEAKER="O.M."
  SEGMENT_START="0" SEGMENT_TYPE="key"
  SNIPPET_ID="1">
      Where are these found ? </SEGMENT>
  <SEGMENT SEGMENT_END="15" SEGMENT_ID="3"
  SEGMENT_MOVE="Resp-Factoid" SEGMENT_SECONDARY_MOVE="none"
  SEGMENT_SPEAKER="Y.M." SEGMENT_START="0"
  SEGMENT_TYPE="key" SNIPPET_ID="1">
      In the rocks .
  </SEGMENT>
  <SEGMENT SEGMENT_END="18" SEGMENT_ID="4"
  SEGMENT_MOVE="Init-YN-InfoReq"   SEGMENT_SECONDARY_MOVE="none"
  SEGMENT_SPEAKER="O.M."  SEGMENT_START="0" SEGMENT_TYPE="key"
```

```
    SNIPPET_ID="1">
        In a pure state ?
    </SEGMENT>
    <SEGMENT SEGMENT_END="5" SEGMENT_ID="5" SEGMENT_MOVE="Resp-Answer-No"
    SEGMENT_SECONDARY_MOVE="none"
    SEGMENT_SPEAKER="Y.M." SEGMENT_START="0"
    SEGMENT_TYPE="key" SNIPPET_ID="1">
        No --
    </SEGMENT>
    <SEGMENT SEGMENT_END="15" SEGMENT_ID="6"
    SEGMENT_MOVE="Resp-Factoid" SEGMENT_SECONDARY_MOVE="none"
    SEGMENT_SPEAKER="Y.M." SEGMENT_START="6"
    SEGMENT_TYPE="key" SNIPPET_ID="1">
        in ores .
    </SEGMENT>
  </VERBALIZED_DIALOGSTRUCT>
  <MONOLOGSTRUCT PARENT="" RELATIONROLE=""
   TREE="elaboration-obj-attribute(0N , 1 );"/>
</M2D_PARAGRAPH2SEGMENTS>
```

***

Bibliography

[1] A. Anderson et al. The HCRC map task corpus. *Language and Speech*, 34:351–366, 1991.

[2] H. Bunt. Dialogue pragmatics and context specification. In H. Bunt and W. Black, editors, *Abduction, Belief and Context in Dialogue: Studies in Computational Pragmatics*, volume 1 of *Natural Language Processing*, pages 81–150. John Benjamins, 2000.

[3] J. Carletta, A. Isard, and J. C. Kowtko. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23:13–31, 1997.

[4] L. Carlson and D. Marcu. Discourse Tagging Reference Manual. Technical report, ISI-TR-545, 2002.

[5] L. Carlson, D. Marcu, and M. E. Okurowski. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*, Aalborg, Denmark, 2001.

[6] H. Clark. *Using Language.* Cambridge University Press, New York, 1996.

[7] M. Core and J. Allen. Coding Dialogs with the DAMSL Annotation Scheme. In *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machine*, 1997.

[8] D. duVerle and H. Prendinger. A novel discourse parser based on support vector machines. In *Procs of ACL*, Singapore, 2009.

[9] M. O'Donnell. RSTTool 2.4 – A Markup Tool for Rhetorical Structure Theory. In *Procs of International Natural Language Generation Conference (INLG 2000)*, Mitzpe Ramon, Israel, 2000.

[10] P. Piwek. Presenting arguments as fictive dialogue. In *8th Workshop on Computational Models of Natural Argument (CMNA08; in conjunction with ECAI 2008)*, Greece, 2008.

[11] P. Piwek and K. van Deemter. Generating under Global Constraints: the Case of Scripted Dialogue. *Journal of Research on Language and Computation*, 5(2):237–263, 2007.

[12] R. Prasad et al. Penn Discourse Treebank 2.0. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, 2008.

[13] J. R. Searle. *A Taxonomy of Illocutionary Acts*. Minnesota Studies in the Philosophy of Science. University of Minnesota Press, language, mind, and knowledge edition, 1975.

[14] S. Stoyanchev and P. Piwek. Constructing the CODA corpus. In *Procs of LREC 2010*, Malta, 2010.

[15] F. Wolf and E. Gibson. Representing discourse coherence: A corpus-based study. *Computational Linguistics*, 2(31):249–288, 2005.