

```
1 {F oh, } {D well, } I guess I'll get over it
1 {F oh, } {D well } that's okay, {F um, }
1 {F oh, } you're not
1 {F oh, } that's okay.
```

6.2.2.6.3 bc "Correct-misspeaking-by-other-speaker"

These aren't very common in this genre but they can be amusing:

```
sd          B.182 utt1:  My other son is just as happy as a bed bug. /
bc          A.183 utt1:  A clam. /
```

Sometimes (but not always) the speaker acknowledges the error afterwards.

```
sd          B.38 utt2:  {C and } I suppose they all have the balloons. /
bc          A.39 utt1:  The air bags, /
b           A.39 utt2:  yeah. /

b^m        B.40 utt1:  The air bags, /
b           B.40 utt2:  yeah. /
```

6.3 Answers

SWBD-DAMSL treats answers quite differently than DAMSL. First, where DAMSL has no subtyping of answers, SWBD-DAMSL answers are divided into 4 classes. Second, in order to speed up coding, we code the unmarked situation with a null label:

Answers-to-(pragmatic)-yes-no-questions

```
affirmative answers
  ny          affirmative answers that are "yes" or a variant
  na          affirmative answers that are not "yes" or a variant
  ny/sd^e     affirmative answers that are "yes" and then an expansion

negative answers
  nn          negative answers that are "no" or a variant
  ng          negative answers that are not "no" or a variant
  nn/sd^e     negative answers that are "no" and then an expansion

other answers
  no          none-of-the-above (maybe, i don't know, etc)
  nd          dispreferred response (well...)
  ^h         hold
```

Answers-to-non-yes-no-questions

The immediate response to a non-yes-no question, (qw, qo, etc)

is *assumed* to be the answer unless it is marked with
'^h' hold-before-answering.

6.3.1 Yes and No answers ny and nn

"ny" is only "yes", "yeah", "yep", "uh-huh", and such other variations on "yes".

We mark ny even if there's a filled pause or discourse marker along with the "yes". These are all ny's, counts from the first 18 conversations:

```
17 yeah
 5 yes
 5 uh-huh
 3 {F uh}, yeah
 2 {F oh}, yeah,
 1 {F oh}, yes
 1 {D well}, yes
 1 yes {F uh,}
 1 yes, actually
 1 yeah, I do
 1 yep
```

nn is "no" and variations: Counts are from about 942 nn's from the first 755 conversations:

```
709 no (75%) 49 uh no (5%) 45 huh-uh (5%) 22 well no (2%) 19 oh no (2%) 16 um no (2%) 11 uh-huh
(1%) 9 no uh (1%) 5 nope 3 uh actually no 2 yes 2 yeah 2 so no 2 probably not 2 but uh no 2 but no 2
actually no
```

*Coder's Heuristic

ny doesn't include "he is" or "he does". A.49.1 is *not* an ny, it's an na:

```
qy      B.48 utt1: Is that the only pet that you have? /
na      A.49 utt1: It is, /
```

If the answer begins with "yes" and then *in the same slash-unit* expands on the yes, ^e can be added (i.e. ny^e) to mark a yes/no answer that has the expansion in the same slash unit:

```
qy      A.1 utt1: Okay, {F um, } Chuck, do you have any pets # there at you
ny^e    B.2 utt1: # Yeah, I do. # /
```

6.3.2 na [a for 'affirmative']

An affirmative answer to a preceding y/n question that does not contain 'yes' or variations.

qy B.16 utt2: do you have kids? /

na A.17 utt1: I have three. /

Another example:

qy A.67 utt1: {C And } do [they, + they] just paper train it or some thin

na B.68 utt1: I guess. /

6.3.3 ng [g for neGative]

For negative answers to a preceding y/n question that does not contain 'no' or a variation.

qy A.18 utt2: did you happen to see last night the special on
Channel Two with James Galway? /

ng B.19 utt1: We don't get Channel Two. /

6.3.4. no [o for 'other' answer]

For responses to y/n questions that are neither affirmative responses ("yes" or "Indeed I do") nor negative responses ("no" or "I don't think so"). The most common case is "I don't know:

qy A.15 utt2: Do you think the jury should have a dollar figure
for losing an arm, a dollar figure for losing different body parts? /

no B.16 utt1: I don't know. /

6.3.5 ^e

The first statements by the same speaker after a yes or no response *to a question* have an sd^e, sv^e. These mark statements which are 'expansions' of the yes/no answer.

nn B.56 utt2: no. /

sd^e B.56 utt3: [I, + I] live alone in an apartment, /

We chose to mark *only* the first utterance after the yes/no answer, even though it will be often the case that utterances after the first one are also "expansions" of the yes/no.

^e can also be added to ny (i.e. ny^e) to mark a yes/no answer that has the expansion in the same slash unit:

```
qy          A.1 utt1:  Okay, {F um, }  Chuck, do you have any pets # there at you
ny^e       B.2 utt1:  # Yeah, I do. # /
```

^e is only added to the first utterance which contains the elaboration:

```
ny  B2.utt1. Yeah, /
sd^e B2.utt2. we do have the death penalty here./
sd  B2.utt3. It's not exercised very often,/
sd  B2.utt4. {C but} we do have it./
```

*Coder's Heuristic

We are **not** marking expansions after answers that do not consist of 'yes' (ny) or 'no' (nn).

```
qy          A.22 utt2:  Do you ride a lot of rallies or a lot of those
                around there? /
ng          B.23 utt1:  Not so much. /
sd          B.23 utt2:  {F Uh, } I guess mostly I bike on my own. /
```

6.3.6 nd "aNswer Dispreferred"

"Dispreferred" responses are marked 'nd'. These are pre-answer sequences of two specific types: answering negatively to a question that presupposes an affirmative answer or responding negatively to a question that presupposes an affirmative) often start with a hedge. This pre-answer sequence is marked "nd" (aNswer Dispreferred). Yes-no questions generally presuppose an affirmative answer as do tag questions with a negative tag:

```
You like Clinton, don't you?
Yes, I do.
```

Formal tag questions with an affirmative tag, on the other hand, presuppose a negative response:

```
Question:          You don't have a problem with that, do you?
Preferred Response: No.
                  I don't.
```

Where these patterns are contradicted by speakers, we may expect dispreferred markers, 'nd' as in the following examples:

```
qy          A.63 utt1:  {F Um, } you kind of think it's something else then? /
nd          B.64 utt1:  {D Well, } that's what the environmentalists were
                claiming in this article. /
```

qy B.100 utt1: Do you and your husband like to work in the yard?/
nd A.101 utt1: {F Oh, } {D well, } we like it once in a while
but not as often as we have to do it . /

(*Question: do we have examples responding to tag questions with 'nd' in our database?)

If the dispreferred pre-answer sequence is transcribed in the same slash unit as the 'no' answer, it is not coded. Rather, the answer itself is coded 'nn' as shown in this example:

bf B.66 utt1: Okay. /
B.66 utt2: {D So, } [[you, + you were out of s-,] + you went
to school] for awhile and quit. Then went back. /
nn A.67 utt1: {D Well, } no. /

Here B starts a "dispreferred response" sequence, (Well...) but then A changes the question allowing B to answer "yes".

qy A.5 utt1: {D Well, } it should be used as a deterrent do you think? /
nd B.6 utt1: {D Well, } -/
qrr* A.7 utt1: {C Or } should it be used, {F uh, } /
+* A.7 utt2: [a-, +] to prevent further, # {F uh, } crime? #
sy B.8 utt1: # Yes, /

Some additional examples are shown below:

b A.1 utt1: Okay, {F uh, } /
qw^t A.1 utt2: could you tell me what you think contributes most to,
{F uh, } air pollution? /
nd B.2 utt1: {D Well, } it's hard to say. /

sd A.203 utt1: A lot of people say it doesn't matter where they live if
they have a nice house /
% A.203 utt2: [{C and, } +
nd B.204 utt1: {D Well, }
sd A.205 utt1: {C but }] I disagree with that, /
% A.205 utt2: I. -/
aa B.206 utt1: I do too, /

qy B.100 utt1: Do you and your husband like to work in the yard?/
nd A.101 utt1: {F Oh, } {D well, } we like it once in a while
but not as often as we have to do it . /
b B.102 utt1: Yeah. /

If the pre-answer sequence is transcribed in the same slash unit as the 'no' answer, it is not coded. Rather, the answer itself is coded 'nn' as shown in this example:

```
bf      B.66 utt1: Okay. /
        B.66 utt2: {D So, } [ [ you, + you were out of s-, ] + you went
        to school ] for awhile and quit. Then went back. /
nn      A.67 utt1: {D Well, } no. /
```

In the following example, B.84 utt1 is NOT an 'nd' because it is not followed by a dispreferred answer:

```
b       B.82 utt1: Yeah, /
ba      B.82 utt2: I know. /
sd      B.82 utt3: [ [ They, + they, ] + they're ] just spoiled rotten, /
%       B.82 utt4: [ {C but, } + {F uh, }
x       A.83 utt1: .
b       B.84 utt1: {C but, } ] no, /
sd      B.84 utt2: [ I, + {F uh, } {F uh, } we ] love to eat out, /
```

6.3.7 hold before answering ^h

This code started out as the DAMSL "Hold", but it drifted a bit; it now covers two kinds of phenomena that we would probably rather have separated out. The two are "true holds" (i.e. putting off the answer to a question), and "floor-holding holds" ("let's see", "what else now").

Type 1: If a question is not directly answered, but the response is nonetheless responsive in some way, it may be marked ^h (Hold).

If the response is itself a question, the question type is coded, followed by the ^h code, as shown in the following example (this is the standard DAMSL Hold):

```
qw      B.6 utt1: {C And } what did you graduate in? /
sd      A.7 utt1: I ju-, - /
qw^h    A.7 utt2: in what major or what year? /
b       B.8 utt1: Yeah, /
sd      B.8 utt2: major . /
```

While a ^h may be applied to questions as noted above, it may also be used as the complete marker for a slash-unit, as in the following examples:

```
qy^d    A.9 utt1: {D Well, } like what? /
^h      B.10 utt1: {D Well, } let's see.
```

qw A.1 utt2: could you tell me what you think contributes most to, {F
 uh, } air pollution? /

^h B.2 utt1: {D Well, } it's hard to say. /

sd B.2 utt2: {E I mean, } while it's certainly the case that
 things like automobiles and factories, {F uh, } pollute a lot, {F uh, }

qw A.1 utt1: Do you ever think that there's a crime that's just so heinous

^h B.2 utt1: That's a good question. /

The second use of "^h" is to mark things like "let's see" even if they don't directly follow a question (as in utt3 below):

qw B.5 utt1: {D Well, } {D now, } {D so } if you were going to have a dinne

^h @A.6 utt1: {F Um, } let's see, /

sd @A.6 utt2: {F uh, } I like seafood. /

^h @A.6 utt3: {F Uh, } let's see,

7. Other

7.1 quoted material ^q and (^q)

^q and **(^q)** are used to mark an utterance that has a direct quotation in it. (we code this because we suspect this may effect pitch and other prosodic features of the utterance).

If the quoted material is embedded in an utterance, the matrix utterance will be coded and the **^q** code will be enclosed in parentheses. (so **sd(^q)** means a statement with a quotation in it, while **^q** means the entire slash-unit is a quote).

The illocutionary force of the utterance in which the quoted material is embedded will be coded, **not** the illocutionary force of the quoted material, as shown in the example below.

sd(^q) B.32 utt1: {C And } when the kids have kids come, {D you know, } s
 he's always saying, {D you know, } why do they have to be here, /

^q B.32 utt2: why can't they send them home, /

^q B.32 utt3: it's too noisy /

sv B.90 utt3: I think that's one of those things when we get to heaven
 we're going to ask God . /

ba A.91 utt1: I know. /

^q B.92 utt1: Why did you do it that way . /

7.2 hedge (h)

A hedge (h) is used to diminish the confidence or certainty with which the speaker makes a statement or answers a question. We code hedges only when they are in a single slash unit of their own (although of course there will be hedges in other utterances as well). Hedges may occur before the statement they diminish as well as after the statement. (The hedges we have been coding look very little like the sentence-internal hedges discussed in the semantics literature (Lakoff 1972, Kay 1987)).

The most common example of a single-slash-unit hedge seems to be "I don't know." Here are some examples:

Hedge before statement:

```
br      A.19 utt1: # The accuracy? # /
h       A.19 utt2: I don't know. /
h       A.19 utt3: I don't know. /
sd      A.19 utt4: {C But, } I know there are a lot of things that can
          influence them /
sv      A.19 utt5: {C and } I think that a person deserves a second chance
          with it or something because most things will stay in your system for a lo
```

Hedge after statement:

```
sv      A.103 utt1: {C so, } [ I think, + I think ] that has helped a
          little bit, /
h       A.103 utt2: I don't know. /
```

```
+       A.25 utt1: -- {D you know. } Then you could lose out on a job when
          really you didn't do anything. /
b       B.26 utt1: Yeah. /
h       A.27 utt1: # {C So } I don't know. # /
sv      B.28 utt1: # {C And } [ I, + I'm ] not # so sure they are that needed. /
```

```
nm      A.58 utt1: # {E I mean, } # no , /
sv      A.58 utt2: you probably know, /
h       A.58 utt3: I don't know. /
```

Hedges other than "I don't know" will, again, only be coded 'h' if they are contained in a single slash unit.

```
sd      A.45 utt1: I have no interest in that, /
sd      A.45 utt2: [ I, + I ] don't have interest of losing my ears, /
h       A.45 utt3: let's just put it that way,
```


b A.41 utt1: Yeah, /
h A.41 utt2: I guess. /

Uncoded hedge (due to slash unit segmentation):

b A.57 utt1: Yeah, /
sv A.57 utt2: I think maybe they'd need to be more knowledgeable though
than just your average Joe off the street --

b B.58 utt1: uh-huh. /

+ A.59 utt1: -- for something like that because of the cultural
differences.

b B.60 utt1: Right. /

+ A.61 utt1: Things like that. /

The h code will NOT be used if the speaker uses "I don't know" to answer a question.
In such a case, there is no hedge, as in the following:

qw^t B.80 utt1: How long is this going to go on, do you know? /

no^t A.81 utt1: I don't know. /

unless the speaker goes on to indicate knowledge as in the following, where "I don't
know" is a hedge:

qy A.35 utt4: however the question is is that making the difference. /

h B.36 utt1: {F Oh, } [I, + I] don't know. /

sv B.36 utt2: {C But } we have a lot of welfare programs /

% B.36 utt3: {C and } -- -/

7.3 How to use "+"

'+' is used to mark DAMSL's "Segment". SWBD-DAMSL has '+' because of our inability
to alter the slash-unit segmentation of SWBD.

Coder's Heuristics

The following is a *wrong* use of +. Don't use a + if the same speaker finished
their previous slash unit with a slash.

sd B.12 utt1: {D Well, } it's, {F uh, } {D you know } they're just,
{ F uh, } aggressive by nature -- /

b A.13 utt1: Uh-huh. /

+ B.14 utt1: -- {C and, } {F uh, } he's been neutered and declawed, # /

7.4 Double labels

Where two labels may apply, ';' is used to separate the two labels. The preferred label appears first, followed by the semi-colon, followed by the 'second-choice' code.

```
sv;sd      B.12 utt2: {C so, } {D you know, } really when you look at it,
            they have full coverage, /
```

We currently use only the first code for interlabeler reliability.

7.5 Transcription errors

Transcription changes are flagged by marking the affected utterance with "*", unless they are slash segmentation errors, in which case they are marked as "@". In general, we only mark transcription errors that directly affect the utterance coding.

7.5.1 Missing slash: one slash unit

If a single slash unit contains too much material (i.e., it should be broken down into more than one slash unit), the utterance is coded o@, as shown in the following examples:

```
o@      A.25 utt7:  Am I a pacifist, physical pacifist, I'm a Christian, /
o@      A.25 utt11: {C but } I'm really not.  You know what I'm saying? /
```

7.5.2 slash units extend over utterances in error

When the slash unit extends over more than one utterance in error, and the first utterance can be coded in spite of the slash unit error, we code the first utterance 'code@' and the second utterance '+@' as shown in the two examples below:

```
aa@      A.27 utt1:  I tell you,
sv        B.28 utt1:  Boats are kind of expensive to maintain. /
+@       A.29 utt1:  {F Oh, } they are, /
```

```
sd@      A.201 utt1: [ We had, + we had ] two Siamese cats.
b        B.202 utt1:  Uh-huh. /
+@       A.203 utt1:  Different times. /
```

If the error extends over more than one slash unit, and no appropriate code is available due to the slash unit error, code the first utterance 'o@' and the second utterance '+@' as in the following example:

```
o@          B.42 utt1:  {D Well, } the,
%          A.43 utt1:  {D So, } /
+@          B.44 utt1:  other issue [ [ is, + is, ] + is ] [ how do you
allow, + {F uh, } [ how, + how ] do you allow ] injustice. Just
like [ the, + the ] policeman [ in, + in ] Los Angeles -
```

At the option of the coder, comments may be inserted regarding the coder's understanding of a correct code in light of the anticipated slash unit correction flagged by the '@' code.

7.5.3 Transcription errors in text

When transcription errors affect text only, the utterance is marked with * and a comment is inserted after the utterance, as in the following:

```
sv*          A.49 utt6: {C and } I know, like now ((  )) in China he did all
these terrible things, / *[['Mao' not 'now']]
```

```
+*          B.82 utt1: -- is ] if you look in the old test meat, and in the
numbers of places that, {F uh, } the Lord went out and just
simply struck down, / *[[old test meat = Old Testament]]
```

8. Bibliography

- Allen, James and Mark Core. 1997. Draft of DAMSL: Dialog Act Markup in Several Layers. March 21, 1997
- Bard, E., Sotillo, C., Anderson, A., and Taylor, M. (1995). The DCIEM map task corpus: Spontaneous dialogues under sleep deprivation and drug treatment. In Proc. ESCA-NATO Tutorial and Workshop on Speech under Stress, Lisbon.
- Carletta, Jean. 1996. Assessing Agreement on Classification Tasks: The Kappa Statistic. Computational Linguistics 22, 249-254.
- Carletta, Jean, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Snwd and Anne H. Anderson. 1997. The Reliability of A Dialogue Structure Coding Scheme. Computational Linguistics 23.1 13-32.
- Godfrey, J., Holliman, E., and McDaniel, J. (1992). SWITCHBOARD: Telephone speech corpus for research and development. Proc. ICASSP, pp. 517-520, San Francisco: IEEE.

- Grosz, Barbara. 1978. Discourse analysis. In D. Walker, editor, "Understanding Spoken Language". 235-268. NY, NY: Elsevier, North-Holland.
- Jefferson, Gail. 1984. Notes on a systematic deployment of the acknowledgement tokens 'yeah' and 'mm hm'. Papers in Linguistics 17.197-216
- Kay, Paul. 1987. Linguistic competence and folk theories of language: Two English h in "Cultural Models in Language and Thought", edited by Dorothy Holland and Naomi Quinn, 67-77. Cambridge: Cambridge University Press
- Lakoff, George. 1972. Hedges: a study in meaning criteria and the logic of fuzzy c CLS 8, 183--228.
- Meteer, Marie and Ann Taylor. 1995. Dysfluency Annotation Stylebook for the Switchboard Corpus
- Weber, Elizabeth G. 1993. Varieties of Questions in English Conversation. Amsterdam: John Benjamins.
- Sacks, Harvey, Emanuel A. Schegloff, and Gail Jefferson. 1974 A simplest Systematics for the organization of turn-taking for conversation. Language 50.4, 696-735.
- Schegloff, Emanuel. 1968 'Sequencing in conversational openings' American Anthropologist 70: 1075-1095
- Schegloff, Emanuel A. 1982. Discourse as an interactional achievement: Some uses of 'uh huh' and other things that come between sentences. In Analyzing Discourse: Text and Talk, edited by Deborah Tannen. Washington, D.C.: Georgetown University Press.
- Schegloff, Emanuel. & H. Sacks. 1973. Opening up closings. Semiotica, VIII: 289-327
- Schiffrin, Deborah. 1987. Discourse Markers. Cambridge: Cambridge University Press.
- Yngve, Victor H. 1970. On getting a word in edgewise. Proceedings of Chicago Linguistics Society 6, 567-577.