Annotation Guidelines

The following contains the full guidelines for manual annotation of candidate corrective feedback utterance pairs. Clarification questions were also marked, as these constitute another parental reaction to a child's grammatical mistake which might indicate the presence of an error to the child. They can thus be interesting to investigate later on.

1 Main Procedure

1. First choose $COF, $CLQ or $NOT:

$COF If the preceding adult utterance contained corrective feedback i.e. the child utterance contained an error and the adult utterance contains the corrected form of exactly the erroneous expression, possibly embedded in a longer sentence.

$CLQ If the preceding adult utterance contained a clarification question i.e. the child utterance is picked up and part of it is rephrased as a wh- or disjunctive question.

$NOT If neither of the two above applies.

   Note that while both $COF and $CLQ *imply* that the child utterance contained an error, neither of these markers are explicitly related to the child error.
   Also, despite errors often being acceptable in spoken language (such as subject and verb omission if these are clear from the context) this does count as an error if it gets corrected.
   In some transcriptions errors are corrected using the *wrong form [: corrected form]* notation. Unfortunately programs running on CHAT look only at the form in the brackets, in this case the corrected form. Thus in this case the error needs to be ignored.

2. Next, code possible errors in the child utterance.

   - In case **$COF** or **$CLQ** were marked before, code only those errors which were corrected / led to the question, respectively. These two markers expect a following explanation of the child error. For clarification questions, sometimes no error is visible (for example if pronunciation is unclear, but the transcription is correct). Then $ERR = 0 should be chosen.

- In case several errors were picked up by corrective feedback in the same utterance, mark all errors separately.
- In case several errors were picked up in the same utterance, one by corrective feedback and the other by a clarification question, mark both, making sure to mention the corresponding errors right after the **$COF** and **$CLQ** labels.
- In case **$NOT** was marked in the previous step errors can but do not need to be coded.

To code errors, chose

(a) The linguistic level of the error, according to the most specific item available in the following list:

General	**Specific**
Syntax	Subject Main Verb Object Other
Noun morphology	Possesive -'s Regular plural -s Irregular plural Other
Verb morphology	3rd person singular -s Regular past -ed Irregular past Other
Unbound morphology	Determiner Preposition Auxiliary verb Present progressive auxiliary Other
Other	-
0	-

Pick this location according to what should correctly be there (e.g. the irregular past tense form of the verb).

(b) Then chose the type of error, according to what the child did wrong, from the following list:

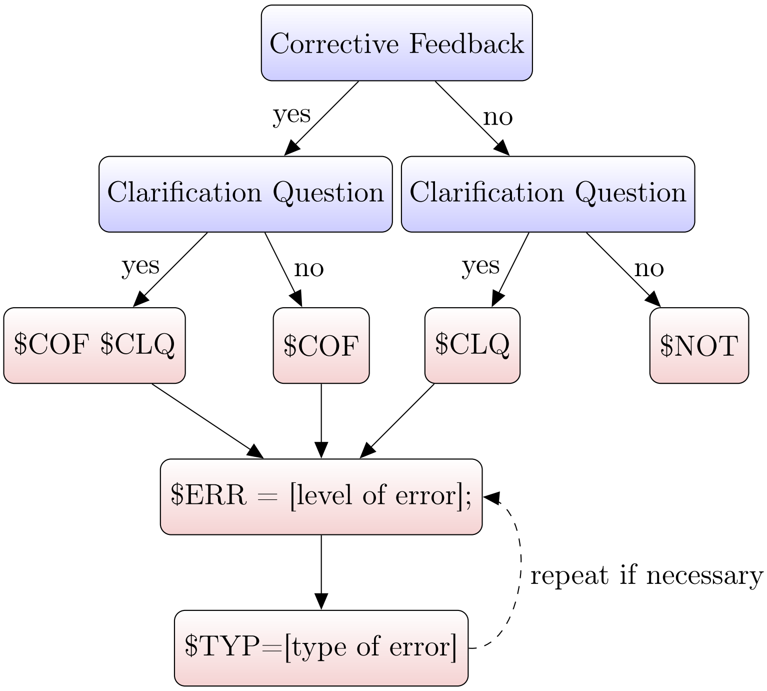
- Omission
- Addition
- Substitution

- Other

  Omission and *addition* should be clear, chose *substitution* if the erroneous form was substituted for another form, possibly from another linguistic level (e.g. regular past tense construction).

Be sure to match both the location and the type to the actual error. Thus, for example, an error of word choice in the object position is not a syntactic error and must thus be classified as *other – substitution* and not as *syntax:object – substitution.*

The following shows the decision tree used for the annotation. Blue nodes represent decisions, red nodes the corresponding annotation.



2 Additional Notes

Annotation was done using the provided coder mode in CLAN. To do this, the following steps were taken:

1. Choose one of the files from the "data" folder for annotation
2. Open it in CLAN
3. Make sure you are in editor mode and CHAT mode, i.e. the line at the bottom of the window reads *date[E/CHAT]*linenumber*
   if this is not the case change using the "Mode" dropdown menu

4. Click at any position in the line containing the first utterance which to annotate

5. Enter coder mode by holding **esc-e**

6. Chose codescof.cut as the codes file

7. At the very bottom of the file the possible tier which to use (%cof) is depicted

8. Press enter

   The annotation is organised hierarchically, thus first the tier is chosen (only one possible option here), then the comment, afterwards the entry for the comment (only applies to ERR comments)

9. The possible comments are listed

10. choose one of
    $COF (preceding exchange contains corrective feedback)
    $CLQ (preceding exchange contains a clarification question)
    $NOT (preceding exchange contains neither of the two)
    using the arrow keys

11. Press enter

    Now there are two possible ways to continue. If the child utterance contained an error follow the next steps, else skip them and go immediately to 14. If the exchange contains corrective feedback list only those errors which get corrected.

12. Choose $ERR using the arrow keys, press enter

13. Choose the linguistic level of the error (see above), press enter

14. Choose the type of the error, press enter

15. Repeat for all errors (keep in mind the restrictions applying to COF and CLQ instances)

    To end annotating one utterance and go to the next one:

16. If the current level of the annotation hierarchy is not the top one (i.e. not the list of comments starting with $) hold **esc-c** to go up. This should not occur, as it implies incomplete annotation.

17. Hold **ctrl-t** twice to get to the next adult utterance

18. To correct mistakes, exit coder mode, remove the complete tier containing the erroneous annotation manually, and re-enter coder mode to fill in the correct annotation

Important Remark: In the CHAT format used in the transcripts the notation *0word [*]* means the word was **not** said, but the transcriber thought it belonged into the sentence and thus put it in the transcript for better understanding.