Reference Data Cookbook
for the "Who Spoke When" Diarization Task

(Version 2.4 March 17, 2003)

Introduction

This document specifies the process by which NIST will create the reference data used for "who spoke when" diarization. This document is the requirements specification for the programs to convert the .aif data from the LDC into .ctm .mdtm and .uem files. The .uem file discussed in this document is a "scoring .uem file".

Main tasks to be performed

The LDC will provide a force-aligned initial version of the data, as an .aif file.

The .ctm file

Produce a .ctm file with the tokens. The "speaker ID" and "speaker type" in the .ctm file must exactly match the corresponding speaker ID and speaker type in the .mdtm file.

The .uem and .mdtm files

Produce a .uem file that specifies the region(s) that *are* to be scored in the audio file. We will not score any regions that were not transcribed—for example, in the case of broadcast news or similar material, the LDC does not transcribe commercials, reporter chit-chat outside the context of a story, station identifications, public-service announcements, promotions for upcoming broadcasts, and long musical interludes. Start with a .uem file that excludes these.

In step two below, we will modify the .uem file to exclude noscore regions around tokens of type *vocalNoise*.

FIRST STEP (TENTATIVE SEGMENT BOUNDARIES FOR THE .MDTM FILE):

Silence of at least 0.3 seconds will force a segment boundary. For purposes of this calculation, tokens of type *nonvocalNoise* and those of type *vocalNoise* will count as silence. Generate this list of segment boundaries as an .mdtm file. The start time for each segment will be the start time of the first token in it that is not of type *vocalNoise*. The end time of the segment will be the end time of the last token in it that is not of type *vocalNoise*. The start and end times will not be padded (no collars).

The .mdtm file will have two comment lines at the top, and the comment line that lists the input files (normally the second of the two comment lines) should include the name of the .ctm file.

SECOND STEP (NO-SCORE ZONES IN THE .UEM FILE):
Then, for the 5 speaker-created vocal noises (breath, cough, laugh, lip-smack, sneeze), define a "no-score" zone that extends from the vocal noise to the edge of the closest segment boundary or word (by *any* speaker), whichever is closer, in both directions. Forced alignment time marks will be used for words. The segment boundary time marks are from the first step.

The result will be like this:

```
                    <----NO-SCORE ZONE---->
              [word]<-------noise--------->[word]
              [word]<-------noise--------->|segment boundary
segment boundary|<-------noise--------->[word]
segment boundary|<-------noise--------->|segment boundary
```

Process the .aif tokens as follows

Any token in the .aif file with tokenContent type of *vocalNoise* should be translated to token type *non-lex* in the .ctm file.

Any token in the .aif file with tokenContent type of *nonvocalNoise* should be translated to token type *non-lex* in the .ctm file, and the speaker field in the .ctm file should be *null*.

Any token in the .aif file with tokenContent type of empty string (“”), *mispronounced*, *acronym*, *spokenLetter*, *interjection* or *properName* should be translated to token type *lex* in the .ctm file.

Any token in the .aif file with tokenContent type of *misspelled* should be translated to token type *noscore* in the .ctm file. We hope misspelled will not occur in the reference data, and we may try to fix the data if it does. It would be a good idea to have the data conversion program throw an exception if this token type occurs (so we can find and fix the problem in the data).

Any token in the .aif file with tokenContent type of *foreign* should be translated to token type *for-lex* in the .ctm file.

Any token in the .aif file with tokenContent type of *preFragment* should be translated to token type *frag* in the .ctm file.

Any token in the .aif file with tokenContent type of *postFragment* should be translated to token type *frag* in the .ctm file.

Any token in the .aif file with tokenContent type of *idiosyncratic* should be translated to token type *un-lex* in the .ctm file.

Any token in the .aif file with tokenContent type of *filledPause* should be translated to token type *fp* in the .ctm file.

Any tokens in the .aif file that are children of an annotation of type *questionableTranscription* should be translated to token type *noscore* in the .ctm file.

Any tokens in the .aif file that are children of an annotation of type *unclear* should be translated to token type *un-lex* in the .ctm file.

Any tokens in the .aif file that span the range of annotations of type *noscore* should be translated to token type *noscore* in the .ctm file.

Note about the treatment of overlaps

Although the presence of overlap is explicitly marked in .aif files, overlap will be implicit in the .mdtm and .ctm files — as multiple words or segments with overlapping times.