Fall 2004 Rich Transcription (RT-04F) Evaluation Plan

Table of Contents

1 INTRODUCTION

The goal of this document is to define the evaluation tasks, performance measures, and test corpora to support the 2004 Rich Transcription (RT-04F) fall evaluation. Rich Transcription (RT) is broadly defined to be a fusion of speech-to-text (STT)[1] technology and **m**eta**d**ata **e**xtraction (MDE) technologies, which will provide the basis for the generation of more usable transcriptions of human-human speech for both humans and machines. The RT-04F evaluations will support DARPA's Effective, Affordable, Reusable Speech-to-text (EARS) program.[2] In addition to EARS contractors, these evaluations are open to all interested volunteers. This evaluation will support nine tasks:

Speech-to-text (STT) tasks

- Unlimited time STT
- Less than or equal to twenty times realtime STT
- Less than or equal to ten times realtime STT
- Less than or equal to one times realtime STT

Metadata Data Extraction (MDE)

Structural Metadata Extraction tasks

- Edit Word Detection
- Filler Word Detection
- Interruption Point (IP) Detection
- SU Boundary Detection

Diarization Metadata Extraction task

- Who spoke when

The RT-04F STT evaluations will be on English, Mandarin, and Arabic data while the RT-04F Metadata evaluations will be limited to English language only.

1.1 PRIMARY VS. CONTRASTIVE SYSTEMS

Primary systems: For each task they participate in, participants must submit output from exactly one *primary* system.[3] Participants must run their primary system on the speech-input condition (see section 8) and may also run it on other conditions[4] specified in section 8. Only the primary systems will be compared across sites.

Contrastive systems: Participants may submit output from additional *contrastive* systems, for tasks on which they have submitted output from a primary system. But participants must also run each of their contrastive systems on the required conditions[5]. These contrastive system submissions will be used for intra-site comparisons only.

Additional required condition for EARS STT Contractors: For the STT tasks, EARS contractors must make a primary submission on the Eval-04 data set and a corresponding submission from the *same system* on the progress test set.

[1] formerly known as automatic speech recognition (ASR)

[2] The EARS research effort is dedicated to developing powerful new speech transcription technology that provides substantially richer and more accurate transcripts than are currently possible. The research focus is on natural, unconstrained speech from broadcasts and telephone conversations in a number of languages. The program objective is to create core enabling technology suitable for a wide range of advanced applications.

[3] That submission is to be designated as primary — see the description of the SYSID string in section 9.3.1.

[4] Those submissions will still be *primary*.

[5] That submission will still be *contrastive* not *primary*.

Participants who are not EARS contractors will not run the progress test set at all.

1.2 CHANGES FROM RT-03

This section briefly lists the differences between the RT-03S and RT-03F evaluations and RT-04.

1.2.1 CHANGES FROM RT-03S

The RT-04F STT metrics and evaluation procedures will have no significant changes from the STT evaluation in RT-03S.

- Submissions for the speaker diarization evaluation (who spoke when) will be in RTTM format, rather than MDTM.
- The STT accuracy targets for EARS contractors are now based on processing time of ten times realtime for Broadcast news and twenty times realtime for conversational telephone speech.[6] The sponsor expects EARS contractors to submit STT systems with those processing speeds.
- The data and data sources will be new.
- The scoring for speaker diarization will use a 0.25 second collar, rather than no collar.
- The "silence smoothing" value is now 0.5 seconds (twice the collar value), rather than 0.3 seconds.

1.2.2 CHANGES FROM RT-03F

- There will be only one official set of metrics (the ones defined in this evaluation plan) and one official data format (RTTM). NIST will provide scoring software implementing this evaluation plan. The "BBN Rich Transcription Framework" is no longer included in this evaluation plan.
- Subtypes of filler words and SU's will be evaluated.
- There will be no Speaker Attributed STT (SASTT) task and no integrated 04rt task.
- The UEM files are substantially different and now focus on defining the data to be processed. Exclusions of small regions (e.g., around speaker-attributed non-lex sounds) are now done by the scoring software controlled by command-line options rather than via UEM files.
- In the RTTM specification (Appendix A), we have renamed **propername** to **propernoun** and renamed **lip-smack** to **lipsmack**, in order to correspond to actual practice and to actual reference data.
- The data and data sources will be new.

2 BACKGROUND

While the traditional STT evaluations have provided a mechanism for evaluating word accuracy, it is clear that words alone are insufficient to formulate a transcription of speech that is maximally useful. A verbatim transcription of the speech stream into a string of lexical tokens yields a transcript that is often difficult to understand. This is because spoken language is much more than just a string of lexical tokens. It contains information about the speaker, prosodic cues to the speaker's intent, and much more. Spoken language also contains disfluencies, which speakers correct and which textual renderings could delete. All of this makes the task of rendering spoken language into text a great challenge, especially with less-than-perfect automatic speech recognition (ASR) performance.

Beginning in the early 1980's, evaluation of ASR stabilized on the current performance measure of word error rate (WER). This measure scores ASR performance using a case-less lexicalized form of ASR output known as the Standard Normalized Orthographic Representation (SNOR) format.[7] The WER is defined as the sum of all ASR output token errors divided by the number of scoreable tokens in a reference transcription of the test data. There are three types of errors: tokens that are missed (deletion errors), inserted (insertion errors), and incorrectly recognized (substitution errors).[8]

Transcripts with the sorts of metadata called for by the RT-04F evaluations will be easier for humans to read and can be processed downstream in more useful ways by computers. Although downstream processing is important, there is not yet clear agreement on what that processing will be. The RT-04F metadata metrics focus on metadata that make transcripts more readable and more understandable to the reader.

The EARS program and the RT evaluation series seek to develop technology that transforms spoken language into a form that is maximally informative. This requires new approaches to acoustical modeling and insightful models of disfluencies, dialogue, and other relevant speaker behaviors. As the EARS program has an overarching goal of making large improvements in STT accuracy, it is expected that the metadata extraction aspects of the program will also advance that goal.

2.1 THE NATURE OF DISFLUENCIES (IN BRIEF)

Spoken disfluencies are portions of speech in which a speaker's utterance is not complete and fluent. There are two kinds of disfluencies: edit disfluencies and filler disfluencies. This section describes the two kinds of disfluencies.

[6] There is no required STT processing speed task for participants who are not EARS contractors.

[7] Since some languages' written forms are not word-based, this concept has been extended to cover lexemes — a representation of a written unit of meaning within a language. Thus, this document frequently refers to lexemes, lexical tokens, or tokens rather than words. For English, these terms may be treated more or less equivalently.

[8] Underlying the tabulation of errors is a requirement to align the tokens in the system output transcript with the tokens in the reference transcript. Traditionally, this has been done using a dynamic programming algorithm that searches for an alignment that minimizes the WER.

Edit disfluencies are speech that the speaker corrects, repeats, or abandons. As this suggests, edit disfluencies have internal structure, with up to three parts.

Although the full three-part form of the common structure to edit disfluencies is not always present, it occurs in some edit disfluencies and is described in this paragraph. The full form begins with the speaker's fluent initial attempt at an utterance followed by a prosodic transition from fluent to non-fluent speech. The initial attempt is known as the *reparandum* and is followed by an *interruption point*. Next in the full form comes an *editing phase* (sometimes called the *editing phrase*) consisting of *fillers* (words that act as pause fillers, discourse markers, or explicit editing terms). The three-part version of an edit disfluency ends with a *repair* (which we will call a *correction*) — a repetition or corrected version of the reparandum.

We score three types of edit disfluencies: revisions, repetitions, and restarts (plus complex disfluencies, which are some combination of one or more of the other three types). The three types are defined as follows.

An edit disfluency in which the *correction* is a corrected version of the *reparandum* is an edit disfluency of type *revision*.

An edit disfluency in which the correction repeats the reparandum is classified as an edit disfluency of type *repetition.* [9]

Some edit disfluencies do not have the full three-part form. Any type of edit disfluency may have an empty editing phase (no editing phase). An edit disfluency of type *restart* has a reparandum but no related correction (the speaker simply abandons what he or she was saying in the reparandum and launches into a restructured version).

In every edit disfluency there is at least one interruption point, at the [right-hand] end of the reparandum.

Edit disfluencies nesting inside other edit disfluencies (or linking in an SU) create *complex edit disfluencies*. This is quite common; the complications that this creates will, however, be glossed over in RT-04F and in the reference data annotation.

Distinguished from edit disfluencies, a filler disfluency (or simply "filler") consists of an interruption point followed by one or more filler words. The interruption point is thus at the beginning of the filler disfluency. There are four subtypes of fillers defined by the Simple Metadata Annotation Specification:

- filled pauses,
- discourse markers,
- explicit editing terms (in the editing phase of an edit disfluency), and
- asides or parentheticals[10] (which are not evaluated).

Disfluencies may occur in succession, and disfluencies of any type may nest inside disfluencies of any type.

2.2 THE RT-04F MODEL OF DISFLUENCIES

Because the metadata annotation of the reference data is an expensive (labor-intensive) process, the EARS community has simplified the RT-04F model of spoken disfluencies from the model that the previous section explains.

RT-04F will not include any treatment of the *correction* portion of edit disfluencies — in fact, the [right-hand] end of the *correction* will not even be marked in the annotation of the reference data.

Although edit disfluencies are often complex (nested or linked), the EARS program has decided to address only the top-most level of these *complex edit disfluencies* at this time, and prohibit annotation as nested edit disfluencies. If the original reparandum has multiple adjacent (serial or linked) disfluencies[11], then the annotated AG file (the reference data format actually produced by the Linguistic Data Consortium) will indicate multiple deletable regions, with an IP at the [right-hand] end of each.[12] In these and other cases of complex edit disfluencies, the complex disfluency will be annotated as a series of simple adjacent disfluencies, rather than as one disfluency with multiple interruption points.

The Simple Metadata Annotation Specification[13] more fully discusses and explains the RT-04F model of disfluencies. The disfluency task that systems are to perform in RT-04F is to identify the regions that are annotated (following the Simple Metadata Annotation Specification) as deletable.

[9] There are some subtleties (related to contractions and fragments) about what counts as a repetition—see the Simple Metadata Annotation Specification for details.

[10] Asides and parentheticals are treated as one subtype in the Simple Metadata Annotation Specification and are not evaluated in the RT-04F evaluation.

[11] For example, "**Yeah** but [the * the big * the b- * the big] * **um** the betrayal or whatever she called it." In this example, fillers are shown boldfaced, the reparandum is in square brackets, the correction is underlined, and Interruption Points (IP's) are shown with asterisks. Although this example has multiple interruption points in the reparandum, the annotation tool *cannot* output a reparandum with multiple Interruption Points (internal IP's), and will instead generate a series of simple adjacent disfluencies.

[12] Similar treatment (as multiple deletable regions) occurs with, for example, a repetition nested inside a restart. For example, "[That is better than * than **um** expecting]* **well** we should have higher expectations than that." But, as in the example in the preceding footnote, the annotation tool cannot output a reparandum with multiple IP's.

[13] http://macears.ll.mit.edu/macears_docs/data/SimpleMDE_V*x.y*.pdf — where *x* and *y* indicate the version.

The two disfluency types, edits and fillers, are independent[14] speech events, although they have similar structure. We have, therefore, divided their detection into separate tasks. In RT-04F (as in RT-03) the editing phase of an edit disfluency is treated as a filler disfluency in its own right—thus, the deletable region of a simple edit disfluency (the *reparandum*) may be followed by a filler (the *editing phase*) that is also deletable and is evaluated separately.

2.3 DEFINITION OF "DELETABLE REGIONS"

As was the case in RT-03, we intend the metadata extraction research in RT-04F to support the creation of transcripts with disfluencies "cleaned up" and with capitalization and punctuation associated with the sentence-like units. The cleanup will include deleting parts of disfluencies. The *deletable region*[15] of a simple edit disfluency is the time taken by the reparandum. The entire time taken by a filler disfluency is deletable.[16] The deletable region of an edit disfluency may include some fillers that are annotated as part of the reparandum. Evaluation will focus on the systems' ability to identify the deletable regions of time for disfluencies.[17]

The reader should keep in mind that *"deletable region"* is not meant to imply a structural part of a single disfluency, but rather a stretch of time during which a sequence of words is uttered.

3 THE RT-04F SPEECH-TO-TEXT (STT) TASKS

We will evaluate speech-to-text systems separately from other submissions. There are four STT processing speed tasks:

- Unlimited time (sttul)
- Less than or equal to twenty times realtime (stt20x),
- Less than or equal to ten times realtime (stt10x), and
- Less than or equal to one times realtime (stt1x).

Participants can build systems for any of the listed processing speeds. However, EARS contractors are required to meet the error rate goals based on processing speed. The RT-04F error rate targets are based on stt10x broadcast news systems and stt20x conversational telephone speech systems. The sponsor expects EARS STT contractors to submit systems with these STT processing speeds.

[14] Fillers and edits are not totally independent, since the editing phase of an edit (if present) is a filler. But fillers also occur by themselves.

[15] Reflecting the "clean up" orientation, the EARS RT-03 metadata model introduced the neologism "*DEPOD*", defined as the DEletable Part Of a Disfluency. This is what we are now calling the *deletable region* for edit disfluencies.

[16] Because the entire time of the filler is deletable, we (and the Simple Metadata Annotation Specification) do not call it the *deletable region* of a filler.

[17] Note that filler disfluencies and the deletable region of edit disfluencies could be defined as words rather than regions of time. The RT-04F submissions are, however, in terms of time.

3.1 DEFINITION OF STT PROCESSING SPEED TASKS

The EARS STT researchers have defined the four processing speed tasks as ratio of the wall-clock Total Processing Time (TPT) to the duration of the recorded audio input. The Total Processing Time is the total amount of time used to process the data, summed over all CPUs used, including I/O and all operations performed after first accessing the test data. The details and the official definition are in Appendix B.

The TPT does not include echo cancellation time. Participants will not necessarily pipeline their echo cancellation to run at the same time as their other processing, so to keep the playing field level, the EARS STT participants have decided that echo cancellation time does not count as part of the TPT. This also makes the STT processing speed calculation for RT-04F more comparable to evaluations in previous years that were run on data that already had echo cancellation.

The system description for each STT submission should include processing time information, calculated as described in Appendix B.

3.1.1 ECHO CANCELLATION

The CTS evaluation test material is distributed without echo cancellation, so that systems may use the echo cancellation algorithm of their choice. The algorithm that NIST has traditionally used in preparing training and test material in the past is implemented in the echo cancellation software available from the Mississippi State archive:

http://www.isip.msstate.edu/projects/speech/software/legacy/fir_echo_canceller/index.html

3.2 SCOREABLE STT TOKENS

We will use the existing STT scoring conventions unchanged (in particular, the same conventions as in RT-03S). RT-04F will score lexical tokens and will not score non-lexical speaker sounds (cough, sneeze, breath, lipsmack, and laugh), or non-speech sounds (such as door slams and so forth).

The RT-04F STT evaluation will include data sets in English, Arabic, and Mandarin.

3.2.1 TOKEN STRING FORMATTING

A single standardized spelling is required for scoreable lexemes, and the STT system must output this spelling in order to be scored as correct.[18] Homophones must be spelled correctly according to the given context in order to be

[18] Token spelling is determined by NIST by first consulting an authoritative reference — e.g., the American Heritage Dictionary (AHD) for English. Lacking an authoritative reference, the internet is searched to find the most common representation. If no single form is dominant, then two or more forms will be permitted via an orthographic map file. As in previous years, a transcription filter and orthographic map file will be used on both the reference and hypothesis transcripts to apply rules for mapping common alternate representations to a single scoreable form.

considered correct. System are to generate all tokens according to Standard Normal Orthographic Representation (SNOR) rules:

Whitespace-separated lexical tokens (for languages that use whitespace-defined words)

Case insensitive alphabetic text (usually in all upper case)

Spelled letters are represented with the letter followed by a period (e.g., "a. b. c.")

No non-alphabetic characters (except apostrophes for contractions and possessives and hyphens for hyphenated words and fragments).

Note that in scoring, hyphenated words will be divided into their constituent parts. Thus, for scoring, a hyphen within a token will be treated as a token separator. A hyphen at either end of a token string indicates the missing part of a spoken fragment.

3.3 STT EVALUATION FRAMEWORK

The STT tasks are similar to previous ASR "Hub-4" and "Hub-5" evaluations, but with additions to support the classification of output tokens and (optionally) speaker assignment. The existing scoring conventions will be used unchanged from RT-03S.

The STT performance measure is essentially the same as the traditional NIST ASR WER measure using the NIST SCLITE software. The primary metric for the RT-04F STT evaluation will (as in RT-03S) be calculated over non-overlapping speech (i.e., omitting regions with multiple reference speakers in the same channel speaking simultaneously). [19]

3.3.1 SYSTEM OUTPUT GENERATION

The system output will be a CTM[20] file (see section 9.2.2). A CTM file is token-based and is to include the following information for each recognized token: the name of the source file, the channel processed, the beginning time of the recognized token, the duration of the recognized token, the string representation of the recognized token, a confidence probability, a token type, and a speaker identifier. The speaker information is optional, but is included to support STT/MDE fusion experiments. If a system does not generate speaker information, then the system should use *unknown* as the speaker for lexical token types. The system should use *null* as the speaker identifier for tokens of non-lex type. See section 9.2.2 for specific formatting requirements. The following describes each possible system output (CTM) token type[21]:

lex - a lexical token.

frag - a lexical fragment.

Note: In the token string, the system may use an optional hyphen to indicate the missing (unspoken) part of the token, but the system must also use the frag CTM token type.

fp - a filled pause.

un-lex - an uncertain lexical token. This type tag is normally used in the reference only.

for-lex - a "foreign" lexical token. This type tag is normally used in the reference only.

non-lex - a non-lexical acoustic phenomenon (breath-noise, door-bang, etc.)[22].

misc - other annotations not covered in above.[23]

Of the token types listed above, we will strip all CTM tokens with types other than lex from the system output prior to STT scoring. Therefore only tokens tagged as type lex in the system output will be aligned and scored, and all others (because stripped out) may be regarded as optional.[24] Although scoring will not penalize (or reward) systems for outputting those optional types, we encourage their output to support metadata experiments.

3.3.2 REFERENCE TOKEN PROCESSING

We will generate a Segment Time Marked (STM) scoring reference from the human reference transcripts.[25] Contraction expansions are annotated in the human reference: the annotator will choose (and the STM file will contain) the single most likely expansion for each contraction. Non-scoreable regions (such as untranscribed areas) are explicitly tagged in the STM file for exclusion from scoring (there will be no scoring UEM file for the STT evaluation). We will score the tokens of the various STM token types[21] in the STM reference as follows:

lex – STM tokens of type lex are not specially tagged in the reference. As such, we will align and score

[19] Note that anticipated upcoming domains in future evaluations, such as STT transcription of meetings, will include processing of overlapping speech.

[20] The CTM file format is one of the immediate predecessors of the RTTM file format. The CTM and RTTM file formats *differ*.

[21] The STM and CTM formats have the same set of token types. Note that in the RTTM format, some of what are token types in CTM and STM format data are instead subtypes of the RTTM *lexeme* type.

[22] RTTM (the reference data for the MDE evaluations) divides this category into non-speech (non-vocal noises) and non-lex (vocal noises). See Appendix A.

[23] A system may give this tag to any token which is to be excluded from scoring — including tokens for which the more specific CTM types exist. But where possible, sites are encouraged to use the supported more specific CTM types to enhance the usefulness of the data for MDE experiments.

[24] A CTM token of type lex, but with orthography that is a known filled pause, will be converted to a generic fp token but will not be stripped from the system output. Thus, such CTM tokens will be scored (as correct, or a substitution, or an insertion).

[25] See ftp://jaguar.ncsl.nist.gov/current_docs/sctk/doc/infmts.htm

them. As mentioned in the preceding section, system output CTM lex tokens are the only ones that we will not strip out and thus are the only system output tokens that we will align and score.

fp – STM tokens of this pause-filler type are tagged as optionally deletable[26] in the reference. As the first step of scoring them, in the reference we will substitute a generic internal fp token in place of these tokens (their orthography will be ignored). Reference fp tokens contribute to the WER denominator.

frag – STM tokens of type frag are tagged in the reference both as optionally deletable and as fragments. They contribute to the WER denominator. Note: In addition, if a system output token of type lex aligns with a frag in the reference, we will count that system output token as correct if the reference frag token string is a substring of the system output token string.[27]

un-lex, for-lex – Tokens of these types are tagged as optionally deletable in the reference. They contribute to the WER denominator.

non-lex and **misc** – These token types are removed from the reference and do *not* contribute to the WER denominator.

3.3.3 GLM PROCESSING

Prior to scoring, we will use a global map file (GLM) to transform both the reference and system output token strings via a set of rules specified in a global map (GLM) file. We do so to ensure that we score as correct hypothesis tokens that do not differ semantically from corresponding reference tokens.

The GLM rules expand contractions in the system output to all possible expanded forms, which may generate several alternative token strings in the system output.

The GLM rules may also split a token string into two or more strings. For example, compound words are split into their constituents.

After GLM filtering, hyphens in both the system output and reference are transformed into token separators.

[26] "Optionally deletable" means that a system may omit the token without penalty, but if the system does output the token then it is supposed to be scored as correct or incorrect (note, however, that in RT-04F these tokens are being stripped from the system output before scoring—thus even if the system outputs them they will *not* get scored as correct or incorrect in RT-04F). "Optionally deletable" tokens do contribute to the count of reference tokens (the WER denominator) whether or not the system outputs them.

[27] But not the other way round. A complete word in the reference will never align to a frag in the system output because all frag's in the system output get stripped out before alignment occurs.

3.3.4 SCORING

Once the pre-processing is complete, we align the system and reference tokens, using a token-mediated alignment optimized for minimum word error rate (WER). The scoreable lexical token sequences from the reference and system output are aligned (using Dynamic Programming) to minimize the Edit Distance[28] between the two token sequences (edit distance is usually called the Levenshtein Distance, after the paper[29] by V. I. Levenshtein that appears to have introduced the idea).

The reference STM file will mark regions of overlapping speech as well as non-transcribed stories/sections as excluded. These will not be scored.

3.4 STT EVALUATION METRICS

An overall STT error score will be computed as the average number of token recognition errors per reference token:

$$Error_{STT} = \left(N_{Del} + N_{Ins} + N_{Subst}\right) / N_{Ref}$$

where

N_{Del} = the number of unmapped reference tokens,

N_{Ins} = the number of unmapped STT output tokens,

N_{Subst} = the number of mapped STT output tokens with non-matching reference spelling per the token rules above, and

N_{Ref} = the number of reference tokens[30]

As an additional optional performance measure, the confidence of a system in its transcription output will be evaluated. In order to do this, the system must attach a measure of confidence to each of its scoreable output tokens. This confidence measure represents the system's estimate of the probability that the output token is correct and must have a value between 0 and 1 inclusive. The performance of this confidence measure will be evaluated using the same normalized cross entropy score that NIST has been using in previous ASR evaluations.[31]

[28] Edit Distance is the minimum number of edits (insertions, deletions, and substitutions) necessary to convert one string into another. The three kinds of edits are simply counted (in effect, equally weighted). Each edit counts as an error in the WER.

[29] V. I. Levenshtein: "*Binary Codes Capable of Correcting Deletions, Insertions and Reversals*", in Soviet Physics Doklady, Vol. 10, Nr. 8, Feb. 1966, pp. 707 – 710.

[30] N_{Ref} includes all scoreable reference tokens (including optionally deletable tokens) and counts the maximum number of tokens (e.g., the expanded version of contractions). Note that N_{Ref} considers only the reference transcript and is not affected by tokens in the system output transcript, regardless of their type.

[31] For a tutorial introduction to normalized cross-entropy as well as the ideas behind normalized cross-entropy and the information-theoretic idea of entropy, see:
http://www.nist.gov/speech/tests/rt/rt2004/fall/docs/NCE.ps

3.4.1 CONDITIONED SUB-SCORING:

STT WER performance statistics will be tabulated for the following conditions:

Language – Performance will be measured separately for English, Chinese (Mandarin), and Arabic language data.

Source – Performance will be measured separately for broadcast news sources and for telephone conversations.

CPU processing time – See section 3.1 and Appendix B for processing time options, calculation, and requirements.

Speaking conditions – Performance will be measured on only non-overlapping speech (primary metric for EARS)[32]

4 THE RT-04F STRUCTURAL-METADATA TASKS

RT-04F features a variety of tasks related to metadata. There are two broad classes of metadata tasks: structural metadata and diarization. This section (section 4) of the document deals with the structural-metadata extraction tasks. The following section (section 5) deals with the "who spoke when" diarization metadata extraction task. The metadata tasks described in these two sections are classified as shown in the following outline list.

Metadata extraction (MDE)

Structural metadata extraction tasks

- Edit Word Detection (EWD)
- Filler Word Detection (FWD)
- Interruption Point Detection (IPD)
- SU Boundary Detection (SUBD)

Diarization metadata extraction task

- Who spoke when

The EWD and FWD tasks require the system to specify regions of time, but their primary metrics are word-based, and these metrics are computed by determining which reference words are covered[33] by the regions of time that the systems have specified.

The IPD and SUBD tasks require the systems to specify points of time, and the SUBD task also requires the system to identify the type of each SU. The primary metrics for these two tasks are detection based (detection includes getting the type correct for SU's).

Although it is not a structural metadata extraction task, we mention here that the "who spoke when" task requires the system to identify regions of time and can be performed without the system generating (or submitting) any STT output at all. The primary metric for that task is time-based.

In contrast, the RT-04F structural metadata extraction tasks (the metadata tasks other than "who spoke when") all require the systems to have (or generate as STT output) a list of the words in the speech signal (including their times).

The system output for all the RT-04F metadata tasks will be submitted in RTTM format.

4.1 STRUCTURAL-METADATA FRAMEWORK

As the primary required evaluation condition, systems get only a digital audio signal as input. Some of the tasks are defined in terms of detection of "extent", i.e., the system must detect and output one or more spans of time indicating the locations and durations of particular metadata events. Other tasks require the detection of "points", i.e., the system must detect and output events that occur at a particular instant in time. A system may implement any combination of the tasks.

For RT-04, the NIST metadata extraction framework defines four structural-metadata detection tasks (for Edit Words, Filler Words, Interruption Points, and SU Boundaries) and one diarization task (who spoke when).

Except for the "who spoke when" diarization task, each system will provide STT output in the form of a sequence of tokens with their start times and durations. The start time and duration for each such token are required for the MDE scoring process.[34] After token alignment and before scoring, the start times and durations of the tokens in the sequence of system STT tokens are transformed or warped to the corresponding reference token times and durations, and those warped times are then used to warp the times of associated metadata. This process is detailed in Appendix C.

All reference data will be distributed as RTTM files, the corresponding UEM files, and the relevant GLM file. All submissions of system output for MDE scoring (including for who spoke when) shall be in RTTM format, and no other data format will be accepted.

Two UEM-formatted files (see section 6) will be used. The metadata *scoring* UEM file will exclude non-transcribed regions (commercials are typically among the non-transcribed material). The metadata *input* UEM file will identify the entire broadcast to be processed. This input UEM file will *not* exclude commercials.

---

or
http://www.nist.gov/speech/tests/rt/rt2004/fall/docs/NCE.pdf

[32] The overlapping speech is not present in the reference data, so it cannot be scored.

[33] A word is covered by a region of time if the mid-point time of the word falls within the region of time.

[34] For diagnostic purposes, performance will also be reported without applying this STT-based alignment.

The metadata objects are represented in RTTM files by dedicated Filler, Edit, IP, SU, and Speaker MDE record types (see Appendix A). A system may, optionally, attach a measure of confidence to each of these records. This confidence measure represents the system's estimate of the probability that the MDE record represents a correctly detected metadata object of that type.[35] This confidence measure will not, however, be evaluated.

4.1.1 Regions ignored by Metadata Scoring Software

The md-eval scoring software implements two types of exclusion regions: called no-eval regions and no-score regions. The two types are distinct; neither is a subset of the other.

no-eval: The no-eval zones determine what events are evaluated. Before alignment, md-eval discards all reference metadata events whose midpoint times fall within a no-eval region. The alignment process then maps system metadata events to reference metadata events but may be unable to find a mapping for some system metadata events. After alignment, md-eval discards each *unmapped* system metadata event whose midpoint falls in a no-eval region.

Prototypically, regions of time that are excluded by the scoring UEM files (see section 6.3) are no-eval regions. Commercials in broadcast news are an example.

In addition to regions that the scoring UEM file excludes as no-eval regions, regions of time covered by RTTM *noscore* records in the reference RTTM file will be no-eval zones.

In the default (official) scoring, regions of time covered by *no_rt_metadata* records in the reference RTTM file will be no-eval zones for structural metadata scoring but will be eval'd time for speaker diarization scoring.

Regions of overlapping speech are those with multiple speakers in the same channel. Command-line options in the scoring software will determine whether the scoring software will treat regions of overlapping speech as eval or no-eval regions, but by default they will be no-eval regions.

no-score: The no-score zones determine what errors and ref events are counted. After alignment, md-eval does not accumulate scoring statistics in no-score regions. Prototypically, speaker diarization has a no-score region around each non-lex vocal noise (such as a sneeze, cough, breath, laugh, or lipsmack).

Regions of time covered by RTTM *noscore* records in the reference RTTM file will be no-score regions (as well as no-eval regions).

In the default (official) scoring, structural metadata (but not speaker diarization) will have no-score regions for the time covered by *lexeme* records of subtype *un-lex* and for the time covered by SU's of subtype *unannotated*.

The no-eval regions are intended to be large sections of audio with well-defined boundaries and are assumed to be speech data that is not of interest. Exclusions of small little pieces of time should occur via no-score regions. A table summarizing the exclusion regions that are in effect will appear at the beginning of the output from the scoring software.

4.1.2 Summary of evaluation procedure

The evaluation procedure consists of four stages. First (after removing all reference events from the no-eval regions) the scoreable lexical token[36] sequences from the reference and system output are aligned (using Dynamic Programming) to minimize the edit distance between two token sequences. This alignment is fixed for the remainder of the evaluation procedure. In the second stage, the timestamps on the system output tokens are warped to match the timestamps on the reference tokens to which they have been aligned. In the third stage, (using the alignment from the first stage, and the warped times from the second stage) the system and reference metadata tokens are aligned with each other (using Dynamic Programming) to maximize the number of lexical tokens covered jointly by the reference and system metadata regions. In the fourth stage, an error rate is calculated for each of the RT-04F metadata tasks; during this fourth stage, no system metadata events are accumulated in the no-score regions.

All metrics are defined over the alignment produced in the first stage of the evaluation procedure. This common alignment operates on the set of scoreable tokens as defined in Section 4.2. When aligning the reference and system output token sequences,

- lexeme tokens of any of the scoreable lexeme subtypes are allowed to align to any other type if the orthography matches,
- lexeme and foreign-lexeme tokens are considered matched if their un-cased orthographic representations are the same, and
- when a system token and a reference token are both of type *filled pause* or both of type *fragment*, they are matched based on their type only, without regard to their orthography.

The end result of the first three stages is that the common alignment is governed principally by the token orthography and type, but metadata (expressed as token subtypes or

[35] The confidence measure represents the confidence in metadata object identification, not confidence in STT transcription.

[36] Section 4.2 specifies the scoreable structural-metadata tokens and explains which tokens can match or align.

attributes) also exerts an influence upon the alignment whenever the orthographies differ between the tokens being compared. In other words, metadata is not permitted to dislodge a token from an alignment that results in an orthographic or type match, but wherever the orthographies or types are mismatched, the alignment is optimized jointly for all the metadata.

Although it does not enter into any of the metadata metrics, for calibration purposes we also compute a Word Error Rate[37] using the alignment from the first stage of the evaluation procedure.

4.2 SCOREABLE STRUCTURAL-METADATA TOKENS

The structural-metadata systems that we are evaluating produce sequences of tokens to represent acoustic events in the speech signal. In scoring, we use these token sequences for two purposes. First, we use them to align the system output with the reference. Second, we then use them to measure the accuracy of the system output against the reference.

Systems will submit RTTM format data[38] for all the RT-04F metadata tasks. Note that in the RTTM format (see Appendix A), some of what are token *types* in CTM and STM format data are instead *subtypes* of the RTTM **lexeme** type.

In RTTM format submissions, we align and score tokens of type lexeme. But we treat lexemes of *frag* and *fp* subtypes differently than lexemes of other subtypes. During the alignment and scoring process, *frag* and *fp* lexeme tokens are considered to match if the type and subtype match, even when the orthographies of the tokens do not match. But in the case of all other subtypes of lexeme tokens (*lex*, *for-lex*, *alpha*, *acronym*, *interjection*, *propernoun*, and *other*), identical, un-cased orthography matches[39] between the reference and system outputs will constitute a correct match.

Tokens of type lexeme with subtype un-lex can occur in reference token sequences but will never fall within an evaluable region of the evaluation data.

un-lex – a representation of a word whose identity is not clear to the human transcriber, or words infected with or affected by laughter.

4.3 STRUCTURAL MDE EVALUATION METRICS

We define separate performance measures for each of the EARS MDE tasks. For each task, we accumulate the total number of errors over all of the files and channels then divide by the total reference count for that task (to normalize) yielding one average for the system on that task.

[37] corresponding to that computed by SCLite, except based on different alignments and on a different notion of what tokens are scoreable

[38] Thus, if a system's STT output is in CTM format, the system must convert that data to RTTM before submitting it for the MDE evaluations.

[39] identical after GLM processing as described in section 3.3.3

4.3.1 CONDITIONED SUB-SCORING

We tabulate structural MDE performance statistics separately for each of the four combinations of data source (broadcast news sources or telephone conversation) and input condition (speech-plus-reference or speech-only). The input conditions are described in section 8.

4.3.2 EDIT WORD DETECTION

The Edit Word Detection (EWD) task is to detect regions of the input signal containing the words in *deletable regions* of edit disfluencies, as they are defined in the Simple Metadata Annotation Specification. For the RT-04F evaluation, the detection task requires the system to specify the start time and duration of the deletable regions. The scoring will be in terms of the reference words covered by these regions of time.

There is no reward or penalty for splitting a single detected region into two or more contiguous regions having identical overall extent. Nor is there any reward or penalty for combining two or more contiguous detected regions into a single detected region of identical extent.

An edit disfluency may have fillers that occur within its deletable region. For the purposes of the Edit Word Detection task, systems should detect the regions containing such filler tokens, as part of this task.

For the RT-04F evaluation, automatic identification of edit disfluency subtype will *not* be part of the Edit Word Detection task metric.

The primary metric is as follows.

$$Error_{EditWordDetection} = \frac{\left(\begin{array}{l}\#\text{ of deletable ref edit tokens that are} \\ \quad\text{not covered by deletable regions of sys edits} \\ +\,\#\text{ of ref tokens that are not deleteable, yet are} \\ \quad\text{covered by deletable regions of sys edits}\end{array}\right)}{\#\text{ of deletable ref edit tokens}}$$

In addition, the software will output each of the three components of the metric (the denominator and the two terms of the numerator).

The preceding formula refers to deletable ref edit tokens, which means tokens that are covered by the deletable regions of edit disfluencies. A token is "covered" by a deletable region if the midpoint (i.e., the average of beg time and end time) of the token falls within that deletable region's time interval.

4.3.3 FILLER WORD DETECTION

The Filler Word Detection (FWD) task is to detect regions of the input signal containing fillers and to correctly identify the subtypes of fillers. Fillers are defined in the Simple Metadata Annotation Specification. This detection task requires the system to specify the start and duration of all regions of the

input signal containing fillers and to specify the subtype of each, (filled pause, discourse marker, or explicit editing term).

In the metric for FWD, there is no reward or penalty for splitting a single detected region into two or more contiguous regions having identical overall extent and the same filler type. Nor is there any reward or penalty for combining two or more contiguous detected regions (of matching type) into a single detected region of identical extent.

Filler tokens may occur within the reparandum (as well as editing phase) of an edit disfluency, and for the purposes of the Filler Word Detection task, those inside the reparandum are to be detected as part of this task. Thus, each such filler token should be detected in the Edit Word Detection task and also in the Filler Word Detection task.

Section 2 of the Simple Metadata Annotation Specification defines four subtypes of fillers: filled pauses, discourse markers, explicit editing terms, and asides/parentheticals. The metric for Filler Word Detection will ignore fillers of subtype "aside/parenthetical".[40]

The primary metric is as follows.

$$Error_{TypedFillerWordDetection} = \frac{\left(\begin{array}{l}\#\text{ of ref filler tokens that are} \\ \quad \text{not covered by sys fillers} \\ +\#\text{ of ref filler tokens that are} \\ \quad \text{covered by sys fillers of a different subtype} \\ +\#\text{ of non-filler ref tokens that are} \\ \quad \text{covered by sys fillers}\end{array}\right)}{\#\text{ of ref tokens in the ref fillers}}$$

In addition, the software will output each of the four components of the metric (the denominator and the three terms of the numerator).

A token is "covered" by a filler if the midpoint (i.e., the average of beg time and end time) of the token falls within the filler's time interval.

4.3.4 IP DETECTION

The Interruption Point Detection (IPD) task is to produce the locations in time where interruption points occur. Interruption points are discussed in the Simple Metadata Annotation Specification (see footnote 5 and section 3.2 of that document). For the RT-04F evaluation, the detection task requires the system to specify the location in time of each interruption point. A complex edit disfluency will have multiple interruption points and (in the RT-04F data) will normally be represented as a series of simple edit disfluencies.

An interruption point (IP) occurs at the [right-hand] end of the deletable region of an edit disfluency (the reparandum in the case of a simple edit), which may be followed by a filler (e.g., a non-empty editing phase will be a filler and will be separately marked as a filler). Because an IP occurs at the beginning of a filler, the deletable region of an edit followed by a filler suggests two contiguous IPs (one for the end of the deletable region and one for the beginning of the filler). In this situation, systems should output either one or two IPs according to the following rule.

When a filler follows the deletable region of an edit within the same SU (i.e., not separated by an incomplete or complete SU boundary) and when there are no intervening RTTM tokens of type "lexeme" (see Appendix A) between the deletable region of the edit and the filler, a single, shared IP should be output. The location of such a shared IP should be specified as the time of the end of the deletable region of the edit. This sharing is independent of the gap in time between the end of the deletable region of the edit and the beginning of the filler. If these conditions are not met, two IPs should be emitted.

For the RT-04F evaluation, automatic identification of IP subtype is *not* part of the IP detection task.

The overall IP error rate will be simply the average number of missed IP detections and falsely detected IPs per reference IP:

$$Error_{IP} = \frac{\left(\#\text{ of missed IP's} + \#\text{ of false alarm IP's}\right)}{\#\text{ of ref IP's}}$$

In addition, the software will output each of the three components of the metric (the denominator and the two terms of the numerator).

4.3.5 SU BOUNDARY DETECTION

The SU Boundary Detection task is to detect the SU endpoint[41], and the SU subtype, for each SU whose midpoint time is in eval'd time. The definition of an SU[42] is provided in the Simple Metadata Annotation Specification. For the RT-04F evaluation, this task requires the system to specify the start time of the SU and its duration (from which the scoring software calculates its endpoint time). The system must also identify the SU's subtype (statement, question, backchannel, or incomplete).

The official scoring of SU boundary detection is based on the last scoreable word in each reference SU. During word alignment, the reference words are aligned to the system words. Then the metadata (including SU's) are aligned. If the reference SU successfully aligns then we can speak of a reference SU and of the corresponding system SU. In scoring,

[40] "Discourse Responses" are neither represented as, nor evaluated as, a separate independent data type or subtype.

[41] The md-eval scoring software reports the metrics of interest under a heading that reads "SU (exact) end detection statistics — in terms of reference words".

[42] SUs have been variously called "slash units", "sentence units", "sentence-like units", "semantic units" and "structural units".

the exact SU boundary is scored as detected if the following three conditions all hold.

1) The reference SU is aligned to a corresponding system SU.
2) The reference SU contains at least one reference word that is not in a no-score region (e.g., some reference RTTM *lexeme* record that is not of subtype un-lex).
3) The midpoint time of the last such reference word in the reference SU is covered by the time taken by the corresponding system SU.

The primary overall SU error score will be computed as the average number of missed SU endpoint detections, falsely detected SU endpoints, and correctly detected SU endpoints that are incorrectly classified as the wrong SU subtype, per reference SU:

$$Error_{\text{typed SU Detection}} = \frac{\left(\begin{array}{l}\#\text{ of missed SU end points} \\ +\ \#\text{ of false alarm SU end points} \\ +\ \#\text{ of detected SU end points with subtype incorrect}\end{array}\right)}{\#\text{ of ref SU end points}}$$

In addition, the software will output each of the four components of the metric (the denominator and the three terms of the numerator). The denominator counts no reference SU (and the numerator counts no error) unless the reference SU contains at least one reference word that is not in a no-score region.

5 DIARIZATION – "WHO SPOKE WHEN" MDE

Diarization is the process of annotating an input audio channel with information that attributes (possibly overlapping) temporal regions of signal energy to their specific sources. These sources can include particular speakers, music, background noise sources, and other signal source/channel characteristics.

A transcript where the speakers are labeled, so that the reader can tell who spoke when, is more readily interpreted. The RT-04F diarization MDE task will be performed on Broadcast News[43] datasets only, and non-lex regions will be excluded from scoring by the scoring software rather than via a UEM file.

As in RT-03S, diarization in RT-04F will be limited to just the speaker segmentation "who spoke when" task, including speaker type (gender) classification. For the "who spoke when" task, small pauses in a speaker's speech, of less than 0.5 seconds, are not considered to be segmentation breaks. Material containing no pauses of 0.5 seconds or more should be bridged into a single continuous segment. Although somewhat arbitrary, the cutoff value of 0.5 seconds has been determined to be a good approximation of the minimum duration for a pause in speech resulting in an utterance boundary. The 0.5 second value is twice the 0.25 second scoring collar (discussed in section 5.1), and those two values are intended to be complementary. Systems should consider vocal noise (laugh, cough, sneeze, breath, lipsmack) to be silence in constructing segment boundaries. Systems are to identify the speaker type: adult_male, adult_female, child, or unknown. Systems must apply the same speaker-type label to all segments attributed to a particular speaker[44].

Although many systems perform the diarization task without transcribing the text, note that a system may make use of the output of its STT word/token recognizer (or any other form of automatic signal processing) in performing this task. The approach used should be clearly documented in the task system description.

5.1 "WHO SPOKE WHEN" DIARIZATION SCORING

In order to measure performance, we will compute an optimum one-to-one mapping of reference speaker IDs to system output speaker IDs. The measure of optimality will be the aggregation, over all reference speakers, of time that is jointly attributed to both the reference speaker and the (corresponding) system output speaker to which that reference speaker is mapped. We will always compute this mapping over all speech, including regions of overlap.[45] Mapping is subject to the following restrictions:

- Each reference speaker will map to at most one system output speaker, and each system output speaker will map to at most one reference speaker. If the system performance is perfect, this mapping will be one-to-one.
- Mapping of speakers will be computed separately for each speech data file.

Although the speaker mapping will take regions of overlapping speech into account, we compute the primary metric over only the non-overlapping speech.

A time collar of 0.25 seconds will be employed to forgive timing errors in the reference (timing errors in the forced-alignment).

We express speaker detection performance in terms of the miss, false alarm, and speaker-error rates that result from the mapping.

[43] Distinguishing the speakers in Conversational Telephone Speech (CTS) data amounts to speech activity detection (each speaker is on a separate channel) and is therefore not of separate interest as a "who spoke when" diarization research task. "Who spoke when" diarization will not be evaluated on CTS datasets.

[44] No sex change in mid conversation.

[45] By "overlap" we mean regions where more than one reference speaker is speaking on the same audio channel.

As the overall time-based **primary metric** for speaker segmentation diarization error, we compute the fraction of speaker time that is not attributed correctly to a speaker:

$$Error_{SpkrSeg} = \frac{\sum_{\substack{\text{all}\\ \text{segs}}} \{\text{dur}(seg) \cdot (\max(N_{\text{Ref}}(seg), N_{\text{Sys}}(seg)) - N_{\text{Correct}}(seg))\}}{\sum_{\substack{\text{all}\\ \text{segs}}} \{\text{dur}(seg) \cdot N_{\text{Ref}}(seg)\}}$$

where the speech data file is divided into contiguous segments at all speaker change points[46] and where, for each such segment, *seg*:

$\text{dur}(seg)$ = the duration of *seg*,

$N_{\text{Ref}}(seg)$ = the # of reference speakers speaking in *seg*,

$N_{\text{Sys}}(seg)$ = the # of system speakers speaking in *seg*,

$N_{\text{Correct}}(seg)$ = the # of reference speakers speaking in *seg* for whom their matching (mapped) system speakers are also speaking in *seg.*

The numerator of the overall diarization error score represents speaker diarization error time, and the score can be decomposed into speaker time that is attributed to the wrong speaker, missed speaker time, and false alarm speaker time.

Speaker time that is attributed to the wrong speaker (called speaker error time) is scored as the sum of the following over all segments:

$\text{dur}(seg) * \{\min(N_{\text{Ref}}(seg), N_{\text{Sys}}(seg)) - N_{\text{Correct}}(seg)\}$.

Missed speaker time is scored as the sum of the following over only segments where more reference speakers than system speakers are speaking:

$\text{dur}(seg) * (N_{\text{Ref}}(seg) - N_{\text{Sys}}(seg))$.

False alarm speaker time is scored as the sum of the following over only segments where more system speakers than reference speakers are speaking:

$\text{dur}(seg) * (N_{\text{Sys}}(seg) - N_{\text{Ref}}(seg))$.

No segment is counted as both miss time and false-alarm time.

By using word counts instead of time, we also calculate and report word-based counterparts to the time-based speaker diarization error score and each of its three parts (speaker error time, missed speaker time, false-alarm speaker time). These word-based versions count the number of reference words covered by the segment (a word is covered by a segment if the word's midpoint time falls in the segment). (Midpoint time is the start time of the word plus half its duration—or the average of its start time and end time.)

In areas of overlap (segments where more than one reference speaker is speaking), note that the duration of the segment is attributed to all the reference speakers who are speaking in the segment, thus counting the time more than once. But since the reference data tells us which speaker actually spoke each reference word, we can (and do) attribute each word to its actual speaker, and in areas of overlap this means the words are not counted more than once.

A system may, optionally, attach a measure of confidence to each of its output speaker segments. This confidence measure represents the system's estimate of the probability that the speaker of this segment is correctly assigned.[47] This confidence measure will not, however, be evaluated.

Using this optimal mapping of reference speaker IDs to system speaker IDs, the scoring software will also compute and report the accuracy of recognition of speaker type (adult_male, adult_female, child, or unknown). There will be two versions of this speaker-attribute-mapping information: one over just successfully detected speakers (i.e., for mapped speakers) and the other (separately) over all system output speakers. The primary metrics, however, for the speaker-type diarization task are described in the next section.

5.2 SPEAKER TYPE (GENDER) DIARIZATION SCORING

The diarization "who spoke when" scoring program can be run in a mode that uses the speaker type (adult_male, adult_female, child, or unknown) as the speaker ID. In this mode, the program will bypass the algorithm to compute an optimum mapping of reference speakers to system output speakers, as the correct mapping (e.g., adult_female to adult_female) is known *a-priori.* As a result, more of the time and words are likely to be mapped than when the mapping was based on speaker IDs. The output in this mode will include the same time-based and word-based metrics described above, but will also include confusion matrices for the speaker types.[48] Using the speaker type as the speaker ID, the primary metric for speaker type diarization is calculated in the same way as indicated above for speaker segmentation diarization.

[46] A "speaker change point" occurs each time any reference speaker or system speaker starts speaking or stops speaking. Thus, the set of currently-speaking reference speakers and/or system speakers does not change during any segment.

[47] The confidence measure represents the confidence in speaker assignment only. It should exclude consideration of the correctness of other attributes such as speaker type and segment times.

[48] These speaker type confusion matrices are always generated by the program, both for speaker segmentation scoring and speaker type scoring. However, they will differ for segmentation and type scoring since they are based on different mappings.

5.3 CONDITIONED SUB-SCORING

We will tabulate MDE Who Spoke When Diarization segmentation statistics separately by Speaker ID and by Speaker Type (gender).

6 EVALUATION UN-PARTITIONED EVALUATIONS MAPS (UEM)

Un-partitioned evaluation maps (UEMs) are the mechanism the evaluation infrastructure uses to specify time regions within an audio recording. An *input* UEM file will be provided for all tasks (including STT), to indicate what audio data is to be processed by the systems. A *scoring* UEM file will be used to specify the time regions to be scored for all the RT-04F MDE tasks. No scoring UEM files will be used in scoring the STT tasks (the STM files will be used to score the STT tasks, and will exclude regions of overlapping speech as well as non-transcribed stories/sections).

6.1 UEM FILE STRUCTURE

The UEM file format is a concatenation of time mark records for a segment of audio in a speech waveform. The records are separated with a newline. Each record must have a file id, channel identifier [1 | 2], begin time, and end time. Each record follows this BNF format:

```
UEM :== <F><SP><C><SP><BT><SP><ET>
```

where,

<SP> indicates a space (" ").

<F> indicates the file id, consisting of the path, filename, and extension of the waveform to be processed.

<C> indicates the waveform channel, which can have a value of "1" or "2".

<BT> indicates the beginning time of the segment measured in seconds from the beginning of the file, which is time 0.

<ET> indicates the ending time of the segment measured in seconds from the beginning of the file, which is time 0.

For example:

```
audio/dev/english/cts/sw_47620.sph 1 0 291.34
audio/dev/english/cts/sw_47621.sph 1 0 301.98
...
```

6.2 SYSTEM INPUT UEM FILES

A UEM file is provided with the evaluation data to define the regions of the audio that the system must process. The boundaries specified by the UEM file will include the beginning and end of a conversation or broadcast-news show.

6.3 METADATA SCORING UEM FILES

As part of the reference data, we provide an MDE scoring UEM file that defines the scoreable regions of the audio file. In addition to the boundaries specified by the system input UEM, the MDE scoring UEM excludes extended regions of non-transcribed speech. These extended untranscribed regions in the Broadcast News data for RT-04F will include commercial breaks, reporter chit-chat outside the context of a story, station identifications, promotions for upcoming broadcasts, public-service announcements, and long musical interludes.

The boundaries (in the reference file) defined by the UEM file apply to all objects in that file. No reference word, speaker-turn, segment, or forced-aligned token will cross them in the reference file (and similarly, none of these objects in the system output should cross a boundary defined by the System Input UEM file).[49] Re-running a forced-alignment process or running an alternative forced-alignment will not affect the UEM files.[50]

No metadata will be scored in any area excluded by the scoring UEM file or after the end of scored time specified by the UEM file. For example, an SU with a duration of 10.0 seconds that ends 0.001 seconds after the end of scored time will not be scored even though the vast majority of its time is scored time.

7 CORPORA RESOURCES

The "development data" (also known as dev data) and the "training data" for RT-04F are listed in Appendix D. The LDC has also released some data that was used for a study of annotation consistency.

8 EVALUATION CONDITIONS

There are many different conditions under which system performance may be evaluated. This section identifies those conditions for which we will compute performance and identifies which of them are the required evaluation conditions.

The following list of evaluation conditions apply to all RT-04F Evaluation tasks.

[49] Boundaries that can be crossed by some object will be generated within the scoring software. Examples of such objects include regions of overlapping speakers, uncertain lexemes (un-lex), and regions surrounding non-lexeme or non-speech tokens. Further, regions that pertain to only part of the signal on a channel (for example, only one speaker) will also be handled by the scoring software rather than the UEM files.

[50] The forced alignment is, in fact, done on a segment at a time, with the segment boundary times as inputs to the forced alignment. The times in the UEM files are also segment times. Thus, the forced alignment has no opportunity to affect the UEM files.

Data set:

Eval 04F (current data set)

Progress (EARS STT contractors only)

Language:

English, (MDE tasks will be English-only in RT-04F)

Mandarin, and

Arabic

Domain:

Broadcast News (BN), and

Conversational Telephone Speech (CTS)

CTS data sets are *not* used for diarization

(STT and structural MDE participants may build systems to address either or both of these domains, and may build a separate system for each of the two domains.)

Input:

Speech-only input: Any desired fully-automatic signal processing approaches may be employed—for structural MDE, this may include the use of a site developed STT system. This is the required evaluation condition for Input for all RT-04F tasks.

Speech plus the reference transcriptions: The function of this evaluation condition (which applies to structural MDE tasks only) is to serve as a perfect-STT control condition. It is an optional contrast evaluation condition. The system inputs will be RTTM formatted files derived from the reference RTTM files and placed in the 'input' directory (described in section 9.2.1 below) of the evaluation corpus. The derived RTTM files will contain only *lexeme* RTTM records — with the speaker's identity expunged, (replaced by <NA>), and with the lexeme subtypes *'alpha'*, *'acronym'*, *'interjection'*, *'propernoun*', and *'other'* mapped into the *lex* subtype.

All participants must agree to completely process all of the data for at least one task and must complete a required condition for that task. This means that, at a minimum, the speech-input-only processing condition must be implemented.

9 PARTICIPATION INSTRUCTIONS

Participation is encouraged for all those who are interested in one or more of the RT-04F tasks. Participants have the freedom to implement systems for either or both domains, Broadcast News or Conversational Telephone Speech. Note the details in section 1.1 about required submissions.

As a condition of participation, all sites must agree to make their submissions (system output, system description, and ancillary files) available for experimental use by other research sites. Further, submission of system output to NIST constitutes permission on the part of the site for NIST to publish scores and analyses for that data including explicit identification of the submitting site and system.

9.1 PROCESSING RULES

9.1.1 RULES THAT APPLY TO ALL EVALUATIONS

All developed systems must be fully automatic requiring no manual intervention to influence the system's decision-making infrastructure when generating the system output. Manual intervention is allowed to shepherd system processes but not to change any parameter settings or processing steps in response to knowledge or intuition gained from processing the evaluation data.[51]

The only exemption from the automatic processing restriction is for the structural MDE reference text condition. Participants who use the reference text condition can manually add pronunciations to their dictionaries to enable forced alignment of the out-of-vocabulary items. Participants cannot use the lexical knowledge gained from the reference+speech-input system to modify their speech-input only system.

Systems will be provided with recorded SPHERE formatted waveform files and a UEM file specifying the speech files and regions within them to be processed. Each conversational telephone speech test waveform will be provided in 2-channel files, and both channels must be processed. Broadcast news speech test data will be presented in single channel files, one per broadcast.

While entire broadcast and conversation files will be distributed, only the material specified in the UEM test index file for the experiment to be run is to be processed. Material outside of the times specified in the UEM test index file is not to be used in any way (e.g., for adaptation).

9.1.2 ADDITIONAL RULES FOR PROCESSING BROADCAST NEWS

News-oriented material (audio, textual, etc.) generated after the beginning of the current test epoch (beginning December 1, 2003) or material (other than the RT03 eval data) from the preceding test epoch (February 2001) **may not be used in any way for system development or training. The RT03 eval data is to be used as test data only.**

Broadcast news material must be processed in the chronological order of the date/time of the original broadcast. Although automatic adaptation may be performed using previously-processed material, systems may not "look ahead" in time at later recordings. Hence, processing must be complete on a particular broadcast news test file before

[51] For example, after processing one file and before processing the next file, shepherding does not include doing anything to exploit knowledge gained *by the researchers* as a result of processing that file.

moving on to the next file.[52] Any form of within-file adaptation is permitted, however, and systems may look backwards in time at previously-processed files. The show identity and original broadcast date are allowable side information that systems may use. Therefore, systems may make use of show-dependent models.

9.1.3 ADDITIONAL RULES FOR PROCESSING CONVERSATIONAL TELEPHONE SPEECH

Conversational telephone speech may be processed in any order and any form of automatic within-conversation and cross-conversation adaptation may be employed. No side information is provided for telephone conversations (e.g., corpus collection name, recording time, etc.). No manual or automatic segmentation will be provided, although systems may make use of segmentation-system outputs donated from other sites.

9.1.4 ADDITIONAL RULES FOR PERFORMING THE STT TASK

EARS contractors (and only EARS contractors) will process the Progress Test Set. The same system must be used to process both the Progress and Current Test sets.

Please note that to ensure the integrity of the Progress Test Set, special rules governing the use (and disposal) of this data must be strictly observed. These are specified in a document to be published at the EARS evaluation website at http://ears.ll.mit.edu/.

Note that all of the constraints specified for the English STT tests regarding training, adaptation, and processing also apply to the Non-English STT tests.

9.2 DATA FORMATS

9.2.1 AUDIO DATA AND OTHER CORRESPONDING INPUTS

For practicality, the recorded waveform files to be processed will be distributed on CD-ROM and the corresponding indices, annotations, and transcripts will be made available via the Web or FTP using an identical directory structure. After the evaluation, system outputs will be released in this structure as well.

Directory	Description
indices/	input UEM files, specifying the files and times to be processed for particular experiments
scripts/	scripts to produce reference data
audio/	Audio files
input/<EXP-ID>/	ancillary data including reference annotations for various experiments — must be used in accordance with instructions for that experiment
output/<EXP-ID>/	system output submissions — will be made available as received for integration tests
reference/	reference transcripts and annotations for post-evaluation scoring and analyses
reference/concatenated/	concatenated eval and dev data — created using scripts in the scripts directory

Note: EXP-ID specifies a unique identifier for each experiment and is defined in section 9.3.1.

For clarity, the "audio/" and "reference/" directories are subdivided into `<DATA>/<LANG>/<TYPE>` subdirectories:

where,

`<DATA>` is either [dev04f | eval04f]

`<LANG>` is one of [english | mandarin | arabic]

`<TYPE>` is either [bnews | cts]

The "indices/" directory contains a set of UEM test index files specifying the waveform data to be evaluated for each EXP-ID condition supported in this evaluation as described in 9.3.1 and these files are named <EXP-ID>.uem with the special site code "expt". Separate UEM files, defined in section 6, will be provided for each experiment for each supported <DATA>, <LANG>, and <TYPE>. Corresponding ancillary data for some control conditions is given in the "input/" directory under subdirectories with the same EXP-ID.

9.2.2 STT OUTPUT FORMAT

The RT-04F STT output format will be the CTM format (.ctm filename extension), as in RT-03S. Each output file is to begin with two special comment lines specifying the experiment run and inputs used. These lines must appear at the beginning of the file and are to be formatted as follows:

The first line may be an optional special comment specifying the experiment ID as defined in section 9.3.1 (EXP-ID) and is of the form:

```
;; EXP-ID: <EXP-ID>
```

For example,

```
;;EXP-ID: bbn_04f_stt10x_eval03_eng_cts_spch_p-ab_1
```

If present, this optional special comment line must begin with two semicolons "`;;`". Note that for purposes of scoring, all lines beginning with two semicolons are considered comments and are ignored. Blank lines are also ignored.

[52] This applies to *all* tasks.

The header comments are followed by a list of CTM records. See the list below for the specific supported token types.

The CTM file format is a concatenation of time mark records for each output token in each channel of a waveform. The records are separated with a newline. Each field in a record is delimited with whitespace. Therefore, field values may not include whitespace characters. Each record follows the following BNF format:

```
CTM-RECORD :==
   <SOURCE><SP><CHANNEL><SP><BEG-TIME>
   <SP><DURATION><SP><TOKEN><SP><CONF>
   <SP><TYPE><SP><SPEAKER><NEWLINE>
```

where

<SP> is whitespace.

<SOURCE> is the waveform basename (no pathnames or extensions should be included).

<CHANNEL> is the waveform channel: "1", "2", etc. This value will always be "1" for single-channel files.

<BEG-TIME> is the beginning time of the token. This time is a floating point number, expressed in seconds, measured from the start time of the file. [53]

<DURATION> is the duration of the token. This time is a floating point number, expressed in seconds. [53]

<TOKEN> is the orthographic representation of the recognized word/lexeme or acoustic phenomena. For English, this is represented as a string of ASCII characters, but a token in the context of a non-English test might be represented in Unicode or some other special character set. Token strings are case insensitive and may contain only upper or lowercase alphabetic characters, hyphens (-), and apostrophes (’). No special characters are to be included in this field to indicate the type of token. Rather, the “TYPE” field is to be used to indicate the token type. Note however that a hyphen may be used for fragments to indicate the missing/unspoken portion of the fragment. However, the “frag” TYPE must still be used.

<CONF> is the confidence score, a floating point number between 0 (no confidence) and 1 (certainty). A value of “NA” is used (in CTM format data) when no confidence is computed and in the reference data. [54]

<TYPE> is the token type. The legal values of <TYPE> are “lex”, “frag”, “fp”, “un-lex”, “for-lex”, “non-lex”, “misc”, or “noscore”. See Section 3 for details on generation and scoring rules for each of these types.

lex is a lexical token.

frag is a lexical fragment. Note: A (optional) hyphen may also be used in the token string to indicate the missing (unspoken) part of the token, but the frag TYPE must also be used.

fp is a filled pause.

un-lex is an uncertain lexical token normally used in the reference only.

for-lex is a “foreign” lexical token normally used in the reference only.

non-lex is a non-lexical acoustic phenomenon (breath-noise, door-bang, etc.)

misc is other annotations not covered above.[55]

noscore is a special tag used only in reference files for scoring, to indicate tokens that should not be aligned or scored.

<SPEAKER> is a string identifier for the speaker who uttered the token. This should be “null” for non-lex tokens and “unknown” when the speaker has not been determined.

Included below is an example of STT system output:

```
7654  1  11.34  0.2  YES  0.763  lex  1
7654  1  12.00  0.34  YOU  0.384  lex  1
7654  1  13.30  0.5  C-  0.806  frag  1
7654  1  17.50  0.2  AS  0.537  lex  1
:
7654  2  1.34  0.2  I  0.763  lex  2
7654  2  2.00  0.34  CAN  0.384  lex  2
7654 2 3.40 0.5 ADD 0.806 lex 2

7654 2 3.70 .2 door-bang 0 non-lex null
7654  2  7.00  0.2  AS  0.537  lex  2
:
```

[53] A required time accuracy for BEG-TIME and DURATION is not defined, but these times must provide sufficient resolution for the evaluation software to align tags with the proper token in the reference when time-alignment-based scoring is used. This alignment can be problematic in the case of quickly-articulated adjoining words. Therefore, systems should produce time tags with as much resolution as is reasonably possible. For context, note that the word with the shortest duration in the RT-03 MDE development test set was 15 ms.

[54] STT systems are required to compute a confidence for each scoreable token output for this evaluation. The “NA” value may be used only for non-scoreable tokens.

[55] Any token which is to be excluded from scoring may be given this tag — including those for which specified types exist. However, where possible, sites are encouraged to use the supported types to enhance the usefulness of the data for MDE experiments.

9.2.3 MDE OUTPUT FORMAT

The RT-04F data format, both for the reference data and for the system submissions, will be RTTM (with .rttm filename extension). See Appendix A for a description of the RTTM format. Each RTTM file corresponds to a single source file in the test.

9.2.4 SYSTEM DESCRIPTION

For each test run (for each unique EXP-ID), a description of the system (algorithms, data, configuration) used to produce the system output must be provided along with your system output. If multiple system runs are submitted for a particular experiment with different systems/configurations, explicitly designate one run as the primary system and the others as contrastive systems in the system description (as well as in the SYSID string in the submission filename). The system descriptions must correspond to the instructions in section 1.1. The system description information is to be provided in a file named:

<EXP-ID>.txt

(where EXP-ID is defined in Section 9.3.1)

and this file is to be placed in the "output" directory alongside the similarly-named directories containing your system output. The system description file is to be formatted as follows:

1. EXP-ID = <EXP-ID>
2. Primary: yes | no
3. System Description:

   [*brief technical description of your system; if a contrastive test, contrast with primary system description*]
4. Training:

   [*list of resources used for training; for STT, be sure to address acoustic and LM training, and lexicon*]
5. References:

   [*any pertinent references*]

9.3 SUBMISSION INSTRUCTIONS

9.3.1 SUBMISSION EXPERIMENT CODES

As was mentioned above, the output of each submitted experiment must be identified by the following code:

```
EXP-ID ::=
   <SITE>_<YEAR>_<TASK>_<DATA>_<LANG>_
   <TYPE>_<COND >_<SYSID>_<RUN>
```

where,

`SITE` ::= expt | bbn | bbnplus | cu | elisa | clips | sri | sriplus | ibm | mitll | ms | pan | ...

(The special SITE code "expt" is used in the EXP-ID-based filename of the UEM test index files under the "indices/" directory to list the test material for a particular experiment and in the EXP-ID-based subdirectory name under the "input/" directory to indicate ancillary data to be used in certain control condition experiments.)

`YEAR` ::= `04f`

`TASK` ::= `ewd | fwd | ipd | subd | spkr | sttul | stt20x | stt10x | stt1x | sttulmb | stt10xmb | stt1xmb`

where the tasks for the RT-04F Rich Transcription Evaluation are:

`ewd` = edit word detection

`fwd` = filler word detection

`ipd` = IP detection

`subd` = SU boundary detection

`spkr` = diarization (who spoke when)

`sttul` = STT with unlimited processing time

`stt20x` = STT running in less than or equal to 20 X realtime

`stt10x` = STT running in less than or equal to 10 X realtime

`stt1x` = STT running in less than or equal to 1 X realtime

`sttulmb` = STT with unlimited processing time, using a mothballed system

`stt10xmb` = STT running in less than or equal to 10 X realtime, using a mothballed system

`stt1xmb` = STT running in less than or equal to 1 X realtime, using a mothballed system

`DATA` ::= eval04f | prog

`LANG` ::= eng | man | arab

RT-04F STT will include all three languages: Arabic, English, and Mandarin.

RT-04F MDE includes only English (eng) material.

`TYPE` ::= bnews | cts

`CONDITION` ::= `spch | ref`

where,

`spch` = audio input only

`ref` = audio input + reference transcript

The "spch" (speech) condition is the primary condition of interest. The "ref" (reference) condition is provided as a control for perfect speech recognition and includes

both the speech and reference transcript as input.[56] The MDE tasks for this condition may make use of only the LEXEME entries in the supplied RTTM as defined in Section 8 "Evaluation Conditions".

`SYSID` ::= site-named string designating the system used

The SYSID string must be present. It is to begin with p- for a primary system or with c- for any contrastive systems. For example, this string could be *p-wonderful* or *c-amazing*.

This field is intended to differentiate among contrastive systems for the same condition. Therefore, a different SYSID should be created for systems where any manual changes were made to a particular system.

`RUN` ::= `1..n` (with values greater than 1 indicating multiple runs of the same experiment/system)

An incremental run number ***must*** be used for multiple submissions of any particular experiment with an identical configuration (due to a bug or runtime problem.) This should *not* be used to indicate contrastive systems; instead, a different SYSID should be used. However, please note that ***only*** the first run will be considered "official" and be scored by NIST unless special arrangements are made with NIST.

Please also note: submissions that reuse identical experiment IDs/run numbers from previous submissions will be automatically rejected.

Examples:

bbn_04f_ip_eval04f_eng_cts_ref_c-superreco1_1

sri_04f_spkr_eval04f_eng_bnews_spch_p-speakerid2_1

9.3.2 SUBMISSION DIRECTORY STRUCTURE

All system output submissions must be formatted according to the following directory structure:

output/`<SYSTEM-DESCRIPTION-FILES>`

output/`<EXP-ID>/<OUTPUT-FILES>`

where,

`<SYSTEM-DESCRIPTION-FILES>` one per `<EXP-ID>` as specified in 9.2.4

`<EXP-ID>` is as defined in Section 9.3.1

`<OUTPUT-FILES>` are as in sections 9.2.2, section 9.2.3, and section 9.2.4.

Note: one output file must be generated for EACH input audio file, as specified in the input UEM file for the experiment being run. (Input UEM files will be in the `indices` directory. See section 9.2.1.)

[56] Reference-condition submissions are extremely useful for data analysis, so participants are encouraged to submit them.

The output files are to be named so as to be identical to the input file basenames with the appropriate .ctm or .rttm filetype extension. For example, an STT output file for the speech waveform file sw_47620.sph must be named sw_47620.ctm and an MDE output file must be named sw_47620.rttm.

When generated, these output files are to be placed under the appropriately-named EXP-ID directory on your system identifying the experiment run.

9.3.3 SUBMISSION PACKAGING AND UPLOADING

To prepare your submission, first create the previously-described file/directory structure. This structure may contain the output of multiple experiments, although you are free to submit one experiment at a time if you like. The following instructions assume that you are using the UNIX operating system. If you do not have access to UNIX utilities or ftp, please contact NIST to make alternate arrangements.

First change directory to the parent directory of your "output/" directory. Next, type the following command:

tar -cvf - ./output | gzip > <SITE>_<SUB-NUM>.tgz

where,

`<SITE>` is the ID for your site as given in section 9.3.1

`<SUB-NUM>` is an integer 1 – n, where 1 identifies your first submission, 2 your second, and so forth.

This command creates a single tar file containing all of your results. Next, ftp to jaguar.ncsl.nist.gov giving the username 'anonymous' and your e-mail address as the password. After you are logged in, issue the following set of commands, (the prompt will be 'ftp>'):

ftp> `cd incoming`
ftp> `binary`
ftp> `put <SITE>_<SUB-NUM>.tgz`
ftp> `quit`

You have now submitted your recognition results to NIST. Note that because the "incoming" ftp directory (where you just ftp'd your submission) is write- protected, you will not be able to overwrite any existing file by the same name (you will get an error message if you try) and you will not be able to list the incoming directory (i.e., with the "ls" or "dir" commands). So, pay attention to whether you get any error messages from the ftp process when you execute the ftp commands stated above.

The last thing you need to do is send an e-mail message to Audrey Le at audrey.le@nist.gov to notify NIST of your submission. The following information should be included in your email:

- The name of your submission file
- A listing of each of your submitted experiment IDs

Example

```
Submission: bbnplus_1 <NL>
Experiments: <NL>
```

```
bbnplus_04f_subd_eval04_eng_cts_spch_
c-superreco1_1<NL>
bbnplus_04f_ipd_eval04_eng_cts_spch_c
-superreco2_1 <NL>
```

Please submit your files in time for us to deal with any transmission/formatting problems that might occur — well before the due date if possible.

Note that **submissions received after the stated due dates *for any reason* will be marked late.**

10 SCHEDULE

16-Aug-2004 NIST releases Current Test (CT) data set. All languages. (Note: the CT data set is also known as the RT-04F test set)

10-Sep-2004 Sites submit CT English STT system outputs, by 8 a.m. Eastern Daylight Time

10-Sep-2004 NIST releases Progress Test (PT) data set to EARS contractors

13-Sep-2004 NIST releases CT English STT system outputs and MDE reference transcripts

21-Sep-2004 Sites submit PT STT system outputs, by 5 p.m. Eastern Daylight Time (EARS contractors only)

21-Sep-2004 NIST releases CT English STT results

24-Sep-2004 Sites submit Arabic and/or Mandarin CT STT system outputs, by 8 a.m. Eastern Daylight Time

24-Sep-2004 NIST releases PT STT results

1-Oct-2004 Sites submit CT MDE system outputs, by 8 a.m. Eastern Daylight Time

1-Oct-2004 NIST releases Arabic and Mandarin CT STT results

8-Oct-2004 NIST releases CT MDE results

15-Oct-2004 Sites submit CT English Mothballed RT03 system outputs, by 8 a.m. Eastern Daylight Time (optional for both EARS contractors and RT participants)

19-Oct-2004 NIST releases CT mothballed results

To Be Determined (possibly 1-Nov-2004) Slides and papers for RT-04 and EARS notebook due

7,8,9,10-Nov-2004 RT-04F Workshop (begins with evening meal on the 7th, ends late afternoon on the 10th)

10, 11-Nov-2004 EARS PI meeting (begins with evening meal on the 10th, ends late afternoon on the 11th)

Please note that the stated dates are hard deadlines. All late submissions will be marked as such, and given the tight schedule, severely late submissions may not be scored at all prior to the workshops.

11 WORKSHOPS

To be determined.

Information regarding workshop logistics and registration will be posted at a later date in email.

Appendix A: RTTM File Format Specification

We have renamed **propername** to **propernoun** and renamed **lip-smack** to **lipsmack**, to correspond to actual practice and actual reference data. There are four general object categories to be represented. They are STT objects, MDE objects, source (speaker) objects, and structural objects.[57] Each of these general categories may be represented by one or more types and subtypes, as shown in table 1.

Table 1 Rich Text object types and subtypes

Type	Subtypes
Structural types:	
SEGMENT	**eval**, or (none)
NOSCORE	(none)
NO_RT_METADATA	(none)
STT types:	
LEXEME	**lex**, **fp**, **frag**, **un-lex**[58], **for-lex**, **alpha**[59], **acronym**[59], **interjection**[59], **propernoun**[59], and **other**
NON-LEX	**laugh**, **breath**, **lipsmack**, **cough**, **sneeze**, and **other**
NON-SPEECH	**noise**, **music**, and **other**
MDE types:	
FILLER	**filled_pause**, **discourse_marker, explicit_editing_term**, and **other**
EDIT	**repetition**, **restart**, **revision**, **simple**, **complex**, and **other**
IP	**edit**, **filler**, **edit&filler**, and **other**
SU	**statement**, **backchannel**, **question**, **incomplete**, **unannotated**, and **other**
CB	**coordinating**, **clausal**, and **other**
A/P	(none)
SPEAKER	(none)
Source information:	
SPKR-INFO	**adult_male**, **adult_female**, **child**, and **unknown**

The STT, MDE and Source information objects are potential research targets. And, except for the static speaker information object [**SPKR-INFO**], each object exhibits a temporal extent with a beginning time and a duration. (The duration of interruption points [**IP**] and clausal boundaries [**CB**] is zero by definition.)

These objects are represented individually, one object per record, using a flat record format with object attributes stored in white-space separated fields. The format is shown in table 2.

[57] Structural objects provide a modicum of temporal organization in the annotation and identify non-evaluable regions.

[58] Un-lex tags lexemes whose identity is uncertain and is also used to tag words that are infected with or affected by laughter.

[59] This subtype is an optional addition to the previous set of lexeme subtypes which is provided to supplement the interpretation of some lexemes. In the STT evaluations, these are treated the same as the lex subtype.

Table 2 Object record format for EARS objects

Field 1	2	3	4	5	6	7	8	9
type	file	chnl	tbeg	tdur	ortho	stype	name	conf

where

`file` is the waveform file base name (i.e., without path names or extensions).

`chnl` is the waveform channel (e.g., "**1**" or "**2**").

`tbeg` is the beginning time of the object, in seconds, measured from the start time of the file.[60] If there is no beginning time, use `tbeg` = "**<NA>**".

`tdur` is the duration of the object, in seconds.[60] If there is no duration applicable, use `tdur` = "**<NA>**".

`stype` is the subtype of the object. If there is no subtype, use `stype` = "**<NA>**".

`ortho` is the orthographic rendering (spelling) of the object for STT object types. If there is no orthographic representation, use `ortho` = "**<NA>**".

`name` is the name of the speaker. `name` must uniquely specify the speaker within the scope of the `file`. If `name` is not applicable or if no claim is being made as to the identity of the speaker, use `name` = "**<NA>**".

`conf` is the confidence (probability) that the object information is correct. If `conf` is not available, use `conf` = "**<NA>**".

This format, when specialized for the various object types, results in the different field patterns shown in table 3.

Table 3 Format specialization for specific object types

Field 1	2	3	4	5	6	7	8	9
Type	*File*	*chnl*	*tbeg*	*tdur*	*ortho*	*stype*	*name*	*conf*
SEGMENT	File	chnl	tbeg	tdur	**<NA>**	**eval** or **<NA>**	name or **<NA>**	conf or **<NA>**
NOSCORE	File	chnl	tbeg	tdur	**<NA>**	**<NA>**	**<NA>**	**<NA>**
NO_RT_METADATA	File	chnl	tbeg	tdur	**<NA>**	**<NA>**	**<NA>**	**<NA>**
LEXEME **NON-LEX**	File	chnl	tbeg	tdur	ortho or **<NA>**	stype	name	conf or **<NA>**
NON-SPEECH	File	chnl	tbeg	tdur	**<NA>**	stype	**<NA>**	conf or **<NA>**
FILLER **EDIT** **SU**	File	chnl	tbeg	Tdur	**<NA>**	stype	name	conf or **<NA>**
IP **CB**	File	chnl	tbeg	**<NA>**	**<NA>**	stype	name	conf or **<NA>**
A/P **SPEAKER**	File	chnl	tbeg	Tdur	**<NA>**	**<NA>**	name	conf or **<NA>**
SPKR-INFO	File	chnl	**<NA>**	**<NA>**	**<NA>**	stype	name	conf or **<NA>**

[60] If `tbeg` and `tdur` are "fake" times that serve only to synchronize events in time and that do not represent actual times, then these times should be tagged with a trailing asterisk (e.g., `tbeg` = **12.34*** rather than **12.34**).

Appendix B: Processing Time Calculation for System Descriptions

Total Processing Time (TPT):
The time to be reported is the total amount of time used to process the data, summed over all CPUs used, including I/O and all operations performed after first accessing the test data.

TPT should be measured using the time or date command. If the time command is used, the real time must be used, and its accuracy should be tested prior to use. Runtime should be measured as the sum of the elapsed time for all CPUs used. A sequence of jobs can be broken up with uncounted time intervals between processes, but in all cases I/O time must be included. The system can be in any state prior to beginning to process the test data, and operations performed ***before*** looking at the test data do not need to be counted.

CTS Echo Cancellation:
To keep the playing field level, you need not count echo cancellation in your realtime calculation. If you run it during recognition processing, the official realtime calculation you report should be (your total processing time minus your echo cancellation processing time) divided by the duration of the test data recording.

Source Signal Duration (SSD):
In order to calculate the realtime factor, the duration of the source signal recording must be determined. The source signal duration (SSD) is the actual recording time for the audio used in the experiment as specified in the experiment's UEM files. This time is channel-independent and should be calculated across all channels for multi-channel recordings.

Speed Factor (SF) Computation:
The speed factor (SF) (also known as "X" and "times-realtime") is calculated as follows:

```
SF = TPT/SSD
```

For example, a 1-hour news broadcast processed in 10 hours would have a SF of 10 (regardless of whether the broadcast is stereo or monaural). And a 5-minute telephone conversation processed in 50 minutes would also have an SF of 10 (regardless of whether the signal is a 4-wire/2-channel signal or a 2-wire/1-channel signal).

Reporting Your Processing Speed Information:
Although we encourage you to break out your processing time components into as much detail as you like, you should minimally report the above information in the system description for each of your submitted experiments in the form:

```
TPT = <FLOAT>
SSD = <FLOAT>
SF  = <FLOAT>
```

Appendix C: The Approach to Alignment and Scoring for STT and MDE

The alignment for both MDE and STT uses the same basic approach: it aligns system output tokens and reference tokens, so that the system output may be scored against the reference: System output tokens that the alignment has mapped to reference tokens are scored as correct if they match as explained in section 3.3.2 and otherwise as substitution errors, while unmapped system output tokens are scored as false alarms ("insertions") and unmapped reference tokens are scored as misses ("deletions").

The alignment is done so as to optimize some ***score***, subject to some ***constraints***.

STT alignment:

Traditionally, for STT scoring, the alignment is done so as to minimize the WER, with the constraints being:

- Token sequencing must be preserved.
- System token times must fall within the enclosing reference segment time interval.
- No more than one system token may be mapped to a reference token.
- No more than one reference token may be mapped to a system token.

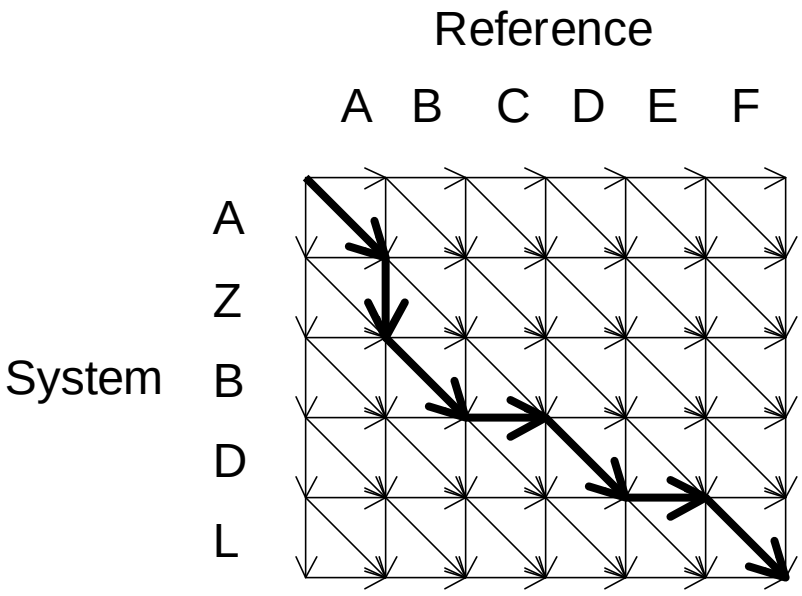

Figure 1. Illustration of alignment task for STT alignment

Taking the spelling and the start/end times into account, the alignment uses a dynamic-programming algorithm to find the alignment of reference words to system words that minimizes the STT error rate. In Figure 1, above, the square matrix represents the entire possible set of alignments, and the heavy line path is the chosen alignment.

The heavy line path in figure 1 can be interpreted as follows.

A diagonal move represents a mapping of a reference word and a system word to each other. For example, the diagonal move in the upper-left square (the first move in the path) represents reference word A being mapped to system word A, and the diagonal move in the lower-right square (the last move in the path) represents reference word F being mapped to system word L.

A vertical move represents an un-mapped system word—an insertion error in the system output. For example, the vertical second move in the path shows that Z in the system output is mapped to nothing in the reference.

Similarly, a horizontal move in the path represents a deletion of a reference word. For example, reference word C was deleted and does not map to anything in the system output.

MDE alignment:

For MDE scoring, the basic approach to alignment is identical to that for STT: find the alignment that optimizes some ***score***, subject to some ***constraints***. We can illustrate this with exactly the same figure 1, but with A B C D etc. representing metadata tokens rather than STT words.

For MDE alignment, however, in contrast to STT alignment, optimization is ***not*** done so as to optimize the MDE score (although that could easily be done). Instead, the optimization maximizes the overlap between system and reference metadata tokens, with the constraints being:

- Token sequencing must be preserved.
- No more than one system token may be mapped to a reference token.
- No more than one reference token may be mapped to a system token.

The MDE tokens are then scored by computing and tallying the discrepancies between system output token start/end times and reference token start/end times. (The discrepancies are computed in terms of reference tokens rather than time.)

Time warping:

In order for the alignment process to generate the lowest possible error rates, the system times (of metadata) must match-up to the reference times. We want this to occur, and it is part of the "official" method of scoring. So in order to do this, the MDE system must also output an auxiliary STT token transcript (which thus requires that the MDE system also perform STT), because the metadata times are tied to (and, in effect, come from) the word times. The scoring program then warps the system times to harmonize with the reference times, as follows. The first step is to perform an STT alignment. From this STT token alignment a piecewise linear continuous transformation is derived so as to map the start/end times for all mapped system output STT tokens to the start/end times of their corresponding reference word tokens (in effect, warping the system word times to match the reference word times). This transformation is then used to modify the system MDE token times (to harmonize with the reference times) prior to MDE alignment. The MDE alignment process then proceeds normally, as before, except that the modified system metadata times are used instead of the original times supplied with the MDE output.

Appendix D: The EARS Development and Training Data for 2004

Development Data

BN and CTS multi-lingual audio data, taken from the DevTest speech data, was delivered 31-3-2004

STT Development Data

Language	DataType	Source	Epoch	Amount	Annotation	Delivery
English	BN	Site-defined devset (TDT-4)	Jan 2001	180 min.	n/a	LDC will not distribute
English	BN	RT-03 eval data (TDT-4)	Feb 2001	180 min.	done in 2003	no redistribution by LDC
English	BN	EARS 2003	Nov 2003	180 min	careful transcription	
Mandarin	BN	RT-03 eval data (TDT-4)	n/a	60 min	done in 2003	no redistribution
Mandarin	BN	EARS 2003	Nov 2003	30 min	careful transcription	LDC2004E19 on 2-4-2004, Version 1.1 on 16-4-2004
Arabic	BN	RT-03 eval data (TDT-4)	n/a	60 min	complete, corrected, careful transcription	LDC2004E19 on 2-4-2004, Version 1.1 on 16-402004
English	CTS	RT-03 eval data (fisher)	n/a	180 min	done in 2003	no redistribution
English	CTS	new fisher English calls	n/a	180 min	careful transcription	LDC2004E19 on 2-4-2004, Version 1.1 on 16-4-2004
Mandarin	CTS	RT-03 eval data (CallFriend)	n/a	60 min	done in 2003	no redistribution
Mandarin	CTS	HKUST	n/a	120 min	transcription	5-2004
Arabic	CTS	CallHome Egyptian	n/a	60 min	done in 2003	no redistribution
Arabic	CTS	new fisher Levantine calls	n/a	120 min	*quick* transcription	LDC2004E19 on 2-4-2004, Version 1.1 on 16-4-2004
Arabic	CTS	new Fisher Levantine calls (same calls as preceding row)	n/a	120 min	careful transcription	LDC2004E19 on 2-4-2004, Version 1.1 on 16-4-2004

MDE Development Data

Language	DataType	Source	Epoch	Amount	Annotation	Delivery
English	BN	RT-03 eval data (TDT-4)	Feb 2001	180 min.	MDE Version 6.2	LDC2004E16 on 2-4-2004, Version 1.1 on 14-5-2004
English	BN	new EARS 2003 collection	Nov 2003	180 min	MDE Version 6.2	delivered 21-5-2004
English	CTS	RT-03 eval data (fisher)	n/a	180 min	MDE Version 6.2	LDC2004E16 on 2-4-2004, Version 1.1 on 14-5-2004
English	CTS	new fisher English calls	n/a	180 min	MDE Version 6.2	delivered 21-5-2004

Training Data

STT Training Data

Lang.	DataType	Source	Epoch	Amount	Annotation	Delivery
English	BN	new EARS collection	Mar–Nov 2003	≈ 7080 hrs audio, CCAP	none	incremental deliveries 10/2003 – 1/2004
English	BN	TDT-4 collection	Mar–July 2001	≈ 530 hrs audio, CCAP	none	incremental deliveries 10/2003 – 1/2004
English	BN	TDT-4 collection / new EARS collection	Mar–July 2001	≈ 340 hrs audio, no CCAP	none	shipped 19-5-2004
English	BN	TDT-4 collection	Dec 2000 – Jan 2001	250 hrs	quick transcription	94 hrs shipped 19-1-2004, other 156 hrs on 9-2-2004
Mandarin	BN	new EARS collection	Mar–Jul 2001 and Mar–Nov 2003	1600+ hrs. audio	none	shipped 19-5-2004
Arabic	BN	new EARS collection	Mar–Jul 2001 and Mar–Nov 2003	2300+ hrs. audio	none	shipped 19-5-2004
English	CTS	English fisher	n/a	200 hrs	quick transcription	140 hrs shipped 11/2003, 40 hrs shipped 2/2004, the rest shipped 2-3-2004
English	CTS	English fisher	n/a	1720 hrs	quick transcription	shipped incrementally through 2-3-2004
English	CTS	English fisher	n/a	1920 hrs of audio only	none	shipped incrementally through 2-3-2004
Mandarin	CTS	new Mandarin collection	n/a	200 hrs	transcription	incremental deliveries through 15-8-2004
Arabic	CTS	Levantine fisher	n/a	18 hrs	quick transcription	delivered 1-4-2004
Arabic	CTS	Levantine fisher	n/a	50 hrs	quick transcription	30-6-2004

MDE Training Data

Lang.	DataType	Source	Epoch	Amount	Annotation	Delivery
English	BN	Hub 4	1998	up to 20 hrs	MDE Version 6.2	Incremental deliveries: version 1.0 on 4-6-2004 version 1.1 on 9-7-2004
English	CTS	Switchboard	n/a	up to 40 hrs	MDE Version 6.2	Incremental deliveries: version 1.0 on 4-6-2004 version 1.1 on 9-7-2004

Evaluation Data

The LDC is scheduled to deliver all the evaluation data on September 1, 2004

STT Eval Data

Language	DataType	Source	Epoch	Amount
English	BN	EARS 2003 collection	December 2003	180 minutes
Mandarin	BN	EARS 2003 collection	?	60 minutes
Arabic	BN	EARS 2003 collection	December 2003	60 minutes
English	CTS	English fisher	n/a	180 minutes
Mandarin	CTS	HKUST collection	n/a	60 minutes
Arabic	CTS	Levantine fisher	n/a	60 minutes

MDE Eval Data (annotated to the MDE V 6.2 spec.)

Language	DataType	Source	Epoch	Amount
English	BN	EARS 2003 collection (same as STT)	December 2003	180 minutes
English	CTS	English fisher (same as STT)	n/a	180 minutes