NIST Open Speech Analytic Technologies 2020 Evaluation Plan (OpenSAT20)

V1.6 updates, July 1, 2020:
Page 2, Planned Schedule updated
Page 3, Use of proprietary training or development data updated

Disclaimer: Any mention of commercial products or reference to commercial organizations is for information only; it does not imply recommendation or endorsement by NIST, nor does it imply that the products mentioned are necessarily the best available for the purpose.

1 Introduction

The National Institute of Standards and Technology (NIST) is continuing the OpenSAT (Open Speech Analytic Technologies) Series with the announcement of OpenSAT20. This series began with a pilot in 2017 followed by the first formal evaluation in 2019. OpenSAT was designed to bring together researchers developing different types of technologies to address speech analytic challenges present in some of the most difficult acoustic conditions.

The objectives of the OpenSAT20 evaluation are 1) to measure performance of the state-of-the-art speech analytics technologies, 2) to provide a forum for the speech analytics community to further test and develop multiple technologies using a common data set, and 3) to promote sharing and leveraging of knowledge.

OpenSAT20 Evaluation will continue to offer three evaluation tasks: *Speech Activity Detection (SAD)*, *Key Word Search (KWS)*, and *Automatic Speech Recognition (ASR)*; and will make use of simulated public safety communications spoken in English. Participants may choose to participate in one, two, or all three of the tasks.

The evaluation will be organized in similar manner to OpenSAT19 with the following exceptions:

1. OpenSAT20 will focus on one data domain (public safety communications).
2. New evaluation data (Test set) and OpenSAT19 evaluation data (Progress Set) will be combined
3. Live public leaderboards will display scores for the Progress set for each task.
4. System descriptions will be available to all participants from the OpenSAT20 web site.
5. For those who cannot physically attend the workshop, provisions will be made to attend remotely.

As with OpenSAT19, site registration will be required in order to participate in OpenSAT20. NIST will send updates to the general purpose OpenSAT mailing list once the registration website is open. A separate OpenSAT20 mailing list will be created for those who have registered for the OpenSAT20.

For questions about OpenSAT20 not covered in this evaluation plan, or to be added to a general purpose OpenSAT mailing list, contact opensat_poc@nist.gov.

2 Planned Schedule

OpenSAT20 Activity	**Begin Date**	**End Date**
OpenSAT20 registration	May 1	July 24
Development data released	May 1	July 31
Evaluation data released	May 1	July 31
Scores on live leaderboard for the Progress set from evaluation data	June 15	July 31
Registration for NIST workshop	TBD	TBD
Last date to upload system output to NIST for scoring		July 31
Scores for the Test set from the evaluation data made available	After July 31	By August 16
Virtual workshop at NIST	September 16	September 16

3 Data

Simulated Public Safety Communications (PSC)

The PSC data was created to support advancements in speech analytic systems for the public safety community. The data is intended to simulate a combination of characteristics found in public safety communications such as background noises typical in first responder environments and voicing characteristics like the Lombard effect, stress, or sense of urgency. The PSC data will have the following characteristics:

- Varying background noise types
- Varying background noise levels
- Speech in the presence of noise
- Speech under stress induced situations

The PSC data are simulated public safety communications spoken in English from the SAFE-T corpus that was collected by the Linguistic Data Consortium (LDC) and initially made available for the OpenSAT19 evaluation. The evaluation data for OpenSAT20 will be data extracted from unexposed portions of the SAFE-T corpus. The audio recordings in the SAFE-T corpus contain speech with potentially increased vocal effort induced by first-responder type background noise conditions and are expected to be challenging for systems to process with a high degree of accuracy. NIST intends to continue with this public-safety speech corpus in the OpenSAT series to measure year-to-year system performance progress.

3.1 Training Data

SAFE-T corpus training data will be provided by LDC that includes 131 hours of audio and "second-pass" transcription for 50 hours of it. OpenSAT participants may also use any other data that is publicly available to develop and train their systems. This includes data that may require a fee or membership dues paid to access the data, given that a sufficiently detailed description of the training data used is provided in the system description.

* Updated: OpenSAT participants may also use any other data that is publicly available and/or proprietary to train and develop their systems.

3.2 Development Data

The SAFE-T corpus development data will be provided by LDC and includes 5 hours of audio recordings with accompanying transcripts. Development data can be used for any purpose, e.g. model adaptation, hyperparameter tuning.

3.3 Evaluation Data

The SAFE-T corpus evaluation data will be provided by LDC and include audio recordings without any accompanying transcripts. The new "blind" evaluation data (Test set) and the Progress set from OpenSAT19 evaluation data will be combined into a single release without any identifiable partitioning. The Test and Progress sets are drawn from the same collection and are approximately five hours of audio recording for each set.

4 Tasks - Overview and Performance Metrics

4.1 Speech Activity Detection (SAD)

The SAD task requires systems is to automatically detect the presence of speech segments in audio recordings. A system output is scored by comparing the system produced start and end times of speech and non-speech segments in audio recordings to human annotated start and end times.

Correct, incorrect, and partially correct segments will determine error probabilities for systems and will be used to measure a system's SAD performance by calculating the Detection Cost Function (DCF) value. See Appendix I for more details.

4.2 Key Word Search (KWS)

The KWS task requires systems to automatically detect all occurrences of keywords in an audio recording. A keyword is a pre-defined single word, or two or three-word phrase, that is transcribed with the spelling convention use in the language's original orthography, in this case, English. The keyword list is prepared by algorithm randomly selecting keywords to represent high to low frequencies of repeated keywords in the data set. There are approximately 1000 Keywords in the Progress and Test sets combined with each Keyword having different frequency counts. Each instance of a detected keyword will have beginning and ending timestamps.

KWS performance is measured by the Term-Weighted Value (TWV), a function of false-positive (false-alarm) and false-negative (missed detection) rates for the keyword and is determined by comparison to the reference. An Actual Term-Weighted Value (ATWV) is calculated from the system decision threshold setting determined by the developer (see Appendix II for TWV and ATWV details).

4.3 Automatic Speech Recognition (ASR)

The ASR task requires systems to automatically produce a verbatim, case-insensitive transcript of all words spoken in an audio recording. ASR performance is measured by the word error rate (WER), calculated as the sum of errors (deletions, insertions and substitutions) divided by the total number of words from the reference (see Appendix III for more details).

5 Evaluation Rules

There is no cost to participate in the OpenSAT evaluation series. Participation is open to all who are able to comply with the evaluation rules set forth within this plan. Development data and evaluation data will be made available to registered participants. Human transcribing, speech detection, or word searching is prohibited. Participants will apply their system's analytic to the data locally and upload their system's output to the NIST server for scoring. See Appendix IV for system output submission packaging. It is required that participants agree to process the data in accordance with the following rules.

- **For KWS:**
  - **Keyword Interactions,** each keyword must be processed separately and independently during keyword detection. The system-generated detection outputs for a keyword (as derived from processing an audio recording) must not influence the detection of other keywords. To facilitate this independence, the search results for each keyword are to be output prior to performing detection on the next keyword.
- **Publication of results**

  The participants agree to follow the guidelines below governing the publication of results.

- At the conclusion of the evaluation, NIST will generate a report summarizing all of the systems' results for conditions of interest. These results/charts will contain the team names, or anonymous team names if that option is selected during registration, of the systems involved. Participants may publish, or otherwise disseminate these charts, unaltered and with appropriate reference to their system.

- Participants may publish results of the public leaderboards (including rankings) unaltered and subject to the following disclaimer that must be included, exactly as is, with the results:

  "These results are not to be construed or represented as endorsements of any participant's system or commercial product, or as official findings on the part of NIST or the U.S. Government. Participants were only required to submit their system outputs to NIST for uniform scoring and analysis. The systems themselves were not independently evaluated by NIST. The data, protocols, and metrics employed in this evaluation are chosen to support speech analytics research and should not be construed as indicating how well these systems would perform in deployed commercial applications.
  Because of the above reasons, this should not be interpreted as a product testing exercise and the results should not be used to make conclusions regarding which commercial products are best for a particular application."

- Participants will not make advertising claims about winning the evaluation or claim NIST endorsement of their system(s). The following language in the U.S. Code of Federal Regulations (15 C.F.R. § 200.113) shall be respected:

  > *NIST does not approve, recommend, or endorse any proprietary product or proprietary material. No reference shall be made to NIST, or to reports or results furnished by NIST in any advertising or sales promotion which would indicate or imply that NIST approves, recommends, or endorses any proprietary product or proprietary material, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of NIST test reports or results.*

- The report that NIST creates cannot be construed or represented as an endorsement for any participant's system or commercial product, or as official findings on the part of NIST or the U.S. Government.

6 Evaluation Protocol

All evaluation activities will be conducted over a NIST web-interface to facilitate information exchange between evaluation participants and NIST.

6.1 Signing up and Setting up an Account

Participants must sign up to perform any evaluation activities. To sign up, go to https://sat.nist.gov.

6.2 Data License Agreement and Accessing Data

Once an account is set up for a team, the team representative must complete the data license agreement and upload it. After the license agreement is confirmed by LDC, LDC will provide instructions for accessing the data.

7 System Input, Output, and Performance Metrics

7.1 System output and scoring for Evaluation Data

System output will include the scores and decisions for both evaluation data subsets (i.e., the Progress and Test subsets) in each upload for scoring. The results on the Progress subset results will be displayed on the live leaderboard to enable Teams to monitor performance progress over the course of evaluation. The Test subset results will be made available after the evaluation period ends

7.2 System scoring results for Evaluation Data

Scoring results for the Progress subset will be displayed on the live leaderboard, enabling Teams to monitor performance progress over the course of the evaluation period. Scoring results for the Test subset will be made available after the evaluation period ends

7.3 System input and Performance metrics

For SAD, see Appendix I,
For KWS, see Appendix II,
For ASR, see Appendix III.

Appendix I SAD

SAD System Input Format

1) Audio files – All audio files are currently expected to be in SPHERE format (reference SPHERE header for sample rate and encoding.
2) Test definition files – XML formatted files that define the test to be performed on the audio files. The XML schema for the SAD test definition file will be available at the OpenSAT20 website.

   Example:

```
<TestSet id=”OpenSAD” audio=”/path/to/audio/root” task=”SAD”>
    <TEST id=”SADTestDataset1”>
         <SAMPLE id=”SAD_sampleFile1” file=”set1/G/file1.sph” />
         <SAMPLE id=”SAD_sampleFile2” file=”set1/G/file2.sph” />
         ...
     </TEST>
</TestSet>
```

SAD System Output Format

System output is to be formatted as a single tab-separated ASCII text file with six columns, see Table 2.

Table 2: SAD system output

Column	Output	Description
1	File ID	Contents of the id attribute of the File tag
2	Channel	A value of 1 or 2
3	Interval start	An offset, in seconds, from the start of the audio file for the **start** of a speech/non-speech interval
4	Interval end	An offset, in seconds, from the start of the audio file for the **end** of a speech/non-speech interval
5	Type	In system output: “speech” or “non-speech” (with no quotation marks). In the reference: S, NS (for Speech, Non-Speech).
6	Confidence (*optional*)	A value in the range 0.0 through 1.0, with higher values indicating greater confidence about the presence/absence of speech

Example of four lines shown for a SAD system output file:

```
    1          2        3     4         5         6
 File ID    Channel   Start  End      Type    Confidence

20703_2020     1       0.0   4.61  non-speech     1
20703_2020     1      4.61   7.08    speech       1
20703_2020     1      7.08   7.49  non-speech     1
20703_2020     1      7.49   9.34    speech       1
```

Interval overlapping will be disallowed and will fail in validation when uploading to the NIST server. Example of overlapping:

```
    1          2        3     4         5         6
 File ID    Channel   Start  End      Type    Confidence

20703_2020     1       0.0   4.61  non-speech    0.8
20703_2020     1      4.50   7.08    speech      0.6
```

SAD System Output Evaluation

Four system output possibilities are considered:

1. True Positive (TP) - system correctly identifies start-stop times of speech segments compared to the reference (manual annotation),
2. True Negative (TN) - system correctly identifies start-stop times of non-speech segments compared to reference,
3. False Positive (FP) - system incorrectly identifies speech in a segment where the reference identifies the segment as non-speech, and
4. False Negative (FN) - system missed identification of speech in a segment where the reference identifies a segment as speech.

SAD error rates represent a measure of the amount of time that is misclassified by the system's segmentation of the test audio files. Missing, or failing to detect, actual speech is considered a more serious error than mis-identifying its start and end times.

A 0.5 s collar, a "buffer zone", at the beginning and end of each speech region will not be scored. If a segment of non-speech between collars is not 0.1 s or greater, then the collars involved are expanded to include the less-than 0.1 s non-speech. For example, no resulting non-speech segment with a duration of just 0.099 s can exist. Similarly, for a region of non-speech before a collar at the beginning of the file or a region of non-speech after a collar at the end of the file, the resulting non-speech segment must last at least 0.1 s or else the collar will expand to include it. In all other circumstances the collars will be exactly the nominal length. Figure 1 illustrates the collars expanding to include a 0.09 s non-speech segment between the collars.

Figure 1: Illustration of a < 0.1 s non-speech segment added to the collars and not including in scoring.

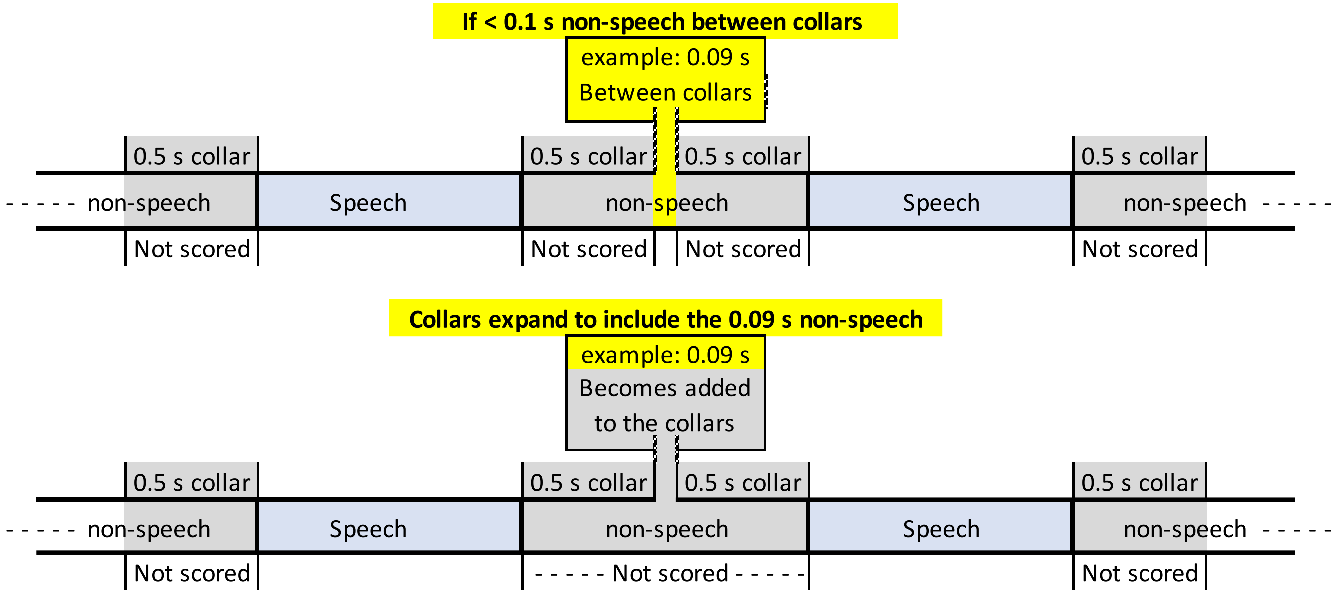
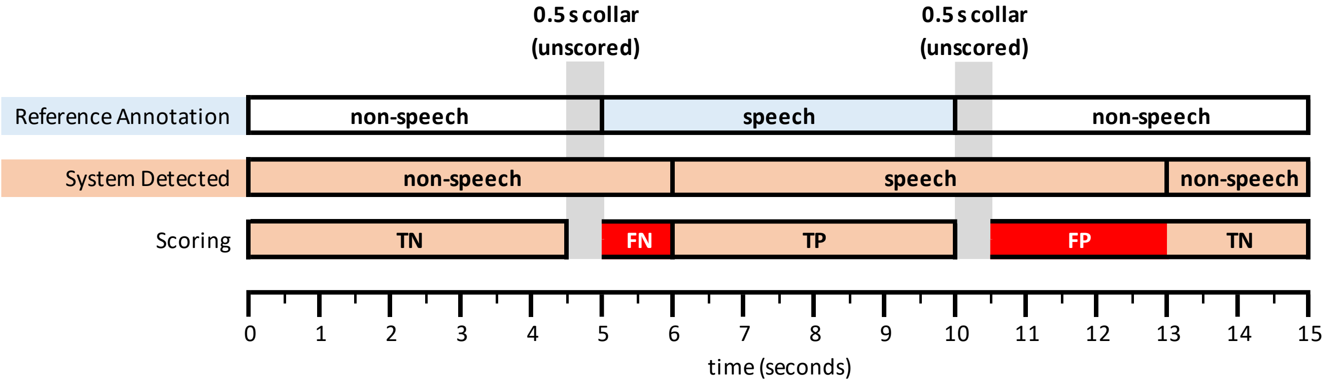


Figure 2 illustrates the relationship between human annotation, the scoring regions resulting from application of the collars, a hypothetical system detected output, and the resulting time intervals from the four system output possibilities. The scoring collars also help compensate for ambiguities in noisy channel annotation. Non-speech collars of half a second in length will define those regions that will not be scored. As can be seen, with collars applied to the annotation, parts of system-detected non-speech and potentially speech are not used in scoring.

Below illustrates an example of a system detected output and the resulting scoring zones relative to the annotation with 0.5 s collars applied. The figure shows the resulting four possibilities (TN, FN, TP, FP) considered in the scoring. The gray areas preceding and trailing the annotated speech are the 0.5 s collar regions.

Figure 2: Hypothetical system output compared to the annotation of the same file showing the resulting four possible outcomes that are used for scoring.



Scoring Procedure

Information for downloading the scoring software will be available at the OpenSAT website.
The four system output possibilities mentioned above determine the probability of a false positive (P_{FP}) and the probability of a false negative (P_{FN}). Developers are responsible for determining a hypothetical optimum setting (θ) for their system that minimizes the DCF value.
P_{FP} = detecting speech where there is no speech, also called a "false alarm"
P_{FN} = missed detection of speech, i.e., not detecting speech where there is speech, also called a "miss"

$$P_{FP} = \frac{total\ FP\ time}{annotated\ total\ nonspeech\ time}$$

$$P_{FN} = \frac{total\ FN\ time}{annotated\ total\ speech\ time}$$

DCF (θ) is the detection cost function value for a system at a given system decision-threshold setting.

$$DCF\ (\theta) = 0.75 \times P_{FN}\ (\theta) + 0.25 \times P_{FP}\ (\theta)$$

P_{FN} and P_{FP} are weighted 0.75 and 0.25, respectively,
θ - denotes a given system decision-threshold setting

Appendix II KWS

KWS System Input Format

1) **Audio file -** All audio files are currently expected to be in SPHERE format.
2) **KWS Experimental Control Files (ECF) -** ECF files are XML formatted files that define the excerpts within audio files to be used for a specific evaluation and the language/source type of each file.

   NIST-supplied ECFs are the mechanism the evaluation infrastructure uses to specify time regions within an audio recording, the language, and the source type specified for the experimental condition. A *system input ECF* file will be provided for KWS and ASR tasks to indicate what audio data is to be indexed and searched by the system. The evaluation code also uses an ECF file to determine the range of data needed to evaluate the system. In the event a problem is discovered with the data, a special *scoring ECF* file will be used to specify the time regions to be scored.

   ECF File Format Description - An ECF file consists of two, hierarchically organized, XML nodes: “ecf”, and “excerpt”. The following is a conceptual description of an ECF file.

   The “ecf” node contains a list of “excerpt” nodes. The “ecf” node has the following attributes:
   - source_signal_duration: a floating-point number indicating the total duration in seconds of recorded speech
   - version: A version identifier for the ECF file
   - language: language of the original source material. Each “excerpt” tag is a non-spanning node that specifies the excerpt from a recording that is part of the evaluation. The “excerpt” has the following attributes:
   - audio_filename: The attribute indicates the file id, excluding the path and extension of the waveform to be processed.
   - source_type: The source type of the recording either “bnews”, “cts”, “splitcts”, or “confmtg”.
   - channel: The channel in the waveform to be processed.
   - start: The beginning time of the segment to processes. The time is measured in seconds from the beginning of the recording which is time 0.0.
   - end: The ending time of the segment to processes. The time is measured in seconds from the beginning of the recording which is time 0.0.

   Example of an ecf file:

```
<ecf source_signal_duration=”340.00” version=”20060618_1400” language=”english” >
<excerpt audio_filename=”audio/dev04s/english/confmtg/NIST_20020214-1148” channel=”1” tbeg=“0.0”
            dur=”291.34” source_type=”confmtg”/>
<excerpt audio_filename=”audio/eval03/english/bnews/ABC_WNN_20020214_1148.sph” channel=”1”
            tbeg=”0.0” dur=”291.34” source_type=”bnews”/>
…
</ecf>
```

   XML schemas for KWS Input/output files - The XML schemas for the files below can be found at https://www.nist.gov/itl/iad/mig/tools, select F4DE-3.5.0, download the tar/zip file, then go to KWSEval/data/

   KWSEval-ecf.xsd for ecf file - input
   KWSEval-kwlist.xsd for kwlist file - input
   KWSEval-kwslist.xsd for kwslist file – ouput

3) KWS KWList Files

KWList files defines the keywords to search for in the indexed corpus, e.g.,

```
/KWlist
        /Babel/[keyword1 in xml format].xml
        /VAST/[keyword2 in xml format].xml
        /PSC/[keyword3 in xml format].xml
```

Keyword List files (KWList) are NIST-supplied, XML-formatted, text files that end with the ".kwlist.xml" extension. These files define the search keywords to be processed by a KWS system. Each keyword is identified by a keyword ID (kwid), which is used to track keywords through the evaluation process and specify keyword texts with a flexible set of attributes.

KWList files consist of three hierarchically organized XML nodes: "kwlist", "kw", and potentially several nodes under "kw". The XML schema for a KWList file can be found in the F4DE-3.5.0 software package at
F4DE-3.5.0/KWSEval/data/KWSEval-kwlist.xsd from https://www.nist.gov/itl/iad/mig/tools.

The following is a conceptual description of a KWList file. The "kwlist" node contains a list of "keyword" nodes and has the following attributes:

- ecf_filename: The basename of the ECF file associated with this Kwlist file. (Basename of a file excludes the directory names and extensions. For example, the basename of "the/directory/file.txt" is "file".)
- version: A version identifier for the file.
- language: Language of the original source material.
- encoding: The character encoding of the text data. Only "UTF-8" is currently accepted.
- compareNormalize: The function used to normalize the text before comparison. Current legal values are blank (which applies no normalization) and "lowercase".

Each "kw" node is a spanning XML tag that contains a set of additional XML nodes to specify the keyword. There is a single attribute 'kwid'.

- kwid: A string identifying the keyword.

The "kw" tag contains two sub-nodes "kwtext" (which is the keyword text) and the "kwinfo" tag (which contains a flexible attribute/value structure).

The "kwtext" tag is a spanning tag that contains the CDATA (character) string for the keyword. The leading and trailing white space of the keyword string is NOT considered part of the keyword while single internal white space(s) are.

The "kwinfo" tag is a spanning tag that contains one or more "attr" tags that specify an attribute name and value with a "name" and "value" tag respectively. Both contents of "name" and "value" tags are CDATA.

Example for a KWlist file:

```
<kwlist ecf_filename=”english_1” version =”20060511-0900” language=”english” encoding=”UTF-8”
compareNormalize=”lowercase”>
        <kw kwid=”dev06-0001”>
            <kwtext>find</kwtext>
                <kwinfo>
                    <attr>
                        <name>NGram Order</name>
                        <value>1-grams</value>
                    </attr>
                </kwinfo>
        </kw>
        <kw kwid=”dev06-0002”>
            <kwtext>many items</kwtext></kw>
               <kwinfo>
                   <attr>
                       <name>NGram Order</name>
                       <value>2-grams</value>
                   </attr>
               </kwinfo>
       </kw>
</kwlist>
```

KWS System Output Format

KWS system output is to be formatted as three, hierarchically organized, xml nodes in a single KWSList file as shown below and use the extension ‘kwslist.xml’. It contains all the runtime information as well as the search output generated by the system. Below is a content description of the XML nodes and attributes. The XML schema for a KWSList file can be found in the F4DE-3.5.0 software package at F4DE-3.5.0/KWSEval/data/KWSEval-kwslist.xsd from https://www.nist.gov/itl/iad/mig/tools.

The three nodes for a KWSList file are:

1. kwslist – the system inputs and parameters used to generate the results.

    The “kwslist” node contains a set of “detected_kwlist” nodes: one for each search keyword. The “kwlist” node contains three attributes:

    - kwlist_filename: The name of the KWList file used to generate this system output.
    - language: Language of the source material.
    - system_id: A text field supplied by the participant to describe the system.

2. detected_kwlist – a collection of “kw” nodes which are the putative detected keywords.

The “detected_kwlist” node has three attributes and contains the system output for a single keyword in “kw” nodes. The “detected_kwlist” node attributes are:

1. kwid: The keyword id from the KWList file.
2. search_time: (optional for backward compatibility) A floating point number indicating the number of CPU seconds spent searching the corpus for this particular keyword.
3. oov_count: An integer reporting the number of tokens in the keyword that are Out-Of-Vocabulary (OOV) for the system and/or the training and development language data. If the system does not use a word dictionary, the value should be “NA”.

3. kw – six attribute fields for the location and detection score for each detected keyword.

The “kw” node is a non-spanning XML node that contains the location and detection score for each detected keyword. The six “kw” node attributes are as follows:

- file: The basename of the audio file as specified in the ECF file.
- channel: the channel of the audio file where the keyword was found.
- tbeg: Offset time from the start (0.0 secs) of the audio file where the keyword starts
- dur: The duration of the keyword in seconds
- score: The detection score indicating the likelihood of the detected keyword.
- decision: [YES | NO] The binary decision of whether or not the keyword should have been detected to make the optimal score.

Below is an example of a KWS system output for keyword ID “dev06-0001”:

- file = NIST_20020214_d05
- channel = 1
- tbeg = 6.956
- dur = 0.53
- score = 4.115
- decision = YES

Below shows the above system output for keyword ID “dev06-0001” in KWSList xml file format for submission:

```
<kwslist
            kwlist_filename=”expt_06_std_eval06_mand_all_spch_expt_1_Dev06.tlist.xml”    language=”english”
            system_id=”Phonetic subword lattice search”>
 <detected_kwlist kwid=”dev06-0001”
                     search_time=”24.3” oov _count=”0”>
     <kw file=”NIST_20020214-1148_d05_NONE” channel=”1” tbeg=”6.956” dur=”0.53” score=”4.115”     decision=“NO“/>
     <kw file=”NIST_20020214-1148_d05_NONE” channel=”1” tbeg=”45.5”   dur=”0.3”   score=”4.65”   cision=“YES“/>
 </detected_kwlist>
</kwslist>
```

KWS System Output Evaluation

Keyword detection performance will be measured as a function of Missed Detection/False Negative (FN) and False Alarm/False Positive (FP) error types.

Four system output possibilities are considered for scoring regions:

1. TP – correct system detection of a keyword (matches the reference location and spelling)
2. TN - system does not detect a keyword occurrence where a keyword does not exist
3. FN - system misses detection or location of a keyword, or miss-spells a keyword
4. FP - system detects a keyword that is not in the reference or not in the correct location

Scoring Procedure

Scoring protocol will be the “Keyword Occurrence Scoring” protocol that evaluates system accuracy based on the three steps below. For more detailed information see the DRAFT KWS16 KEYWORD SEARCH EVALUATION PLAN, PLAN (KWS16-evalplan-V04), https://www.nist.gov/document-194, also available at https://www.nist.gov/itl/iad/mig/openkws16-evaluation.

Information for accessing and downloading the scoring software will be available at the OpenSAT website.

1. Reference-to-system keyword alignment

- The KWS evaluation uses the Hungarian Solution to the Bipartite Graph matching problem[1] to compute the minimal cost for 1:1 alignment (mapping) of reference keywords to system output keywords.

2. Performance metric computation (TWV, ATWV)
   - Uses probability values derived for FP, and FN.
   - System Actual TWV (ATWV): a measure of keyword detection performance at a given system's threshold setting (θ).
   - System Maximum TWV (MTWV): an oracle measure of keyword detection performance at the system's optimal θ setting. (The difference between ATWV and MTWV indicates the loss in performance due to a less-than-optimal system threshold (θ) setting for ATWV when determining the θ for ATWV.)
3. Detection Error Tradeoff (DET) Curves
   - Curve depicts the tradeoff between missed detections versus false alarms for a range of θ settings.

Term Weighted Value (TWV)

$$TWV(\theta) = 1 - \left[P_{FN}(\theta) + \beta \cdot P_{FA}(\theta)\right]$$

Choosing θ:

- Developers choose a decision threshold for their "Actual Decisions" to optimize their term-weighted value: All the "YES" system occurrences
  - Called the "**Actual Term Weighted Value**" (ATWV)
- The evaluation code also searches for the system's optimum decision score threshold
  - Called the "**Maximum Term Weighted Value**" (MTWV)

[1] Harold W. Kuhn, "The Hungarian Method for the assignment problem", *Naval Research Logistic Quarterly*, **2**:83-97, 1955.

Appendix III ASR

ASR System Input Format

1) **Audio Files -** Currently, all audio files are expected to be in SPHERE format.
2) **ASR Experimental Control Files (ECF)** (same as KWS) - ECF files are XML formatted files that define the excerpts within audio files to be used for a specific evaluation and the language/source type of each file.

   NIST-supplied ECFs are the mechanism the evaluation infrastructure uses to specify time regions within an audio recording, the language, and the source type specified for the experimental condition. A *system input ECF* file will be provided for KWS and ASR tasks to indicate what audio data is to be indexed and searched by the system. The evaluation code also uses an ECF file to determine the range of data needed to evaluate the system. In the event a problem is discovered with the data, a special *scoring ECF* file will be used to specify the time regions to be scored.

   ECF File Format Description - An ECF file consists of two, hierarchically organized, XML nodes: “ecf”, and “excerpt”. The following is a conceptual description of an ECF file.

   The “ecf” node contains a list of “excerpt” nodes. The “ecf” node has the following attributes:

   - source_signal_duration: a floating-point number indicating the total duration in seconds of recorded speech
   - version: A version identifier for the ECF file
   - language: language of the original source material. Each “excerpt” tag is a non-spanning node that specifies the excerpt from a recording that is part of the evaluation. The “excerpt” has the following attributes:
   - audio_filename: The attribute indicates the file id, excluding the path and extension of the waveform to be processed.
   - source_type: The source type of the recording either “bnews”, “cts”, “splitcts”, or “confmtg”.
   - channel: The channel in the waveform to be processed.
   - start: The beginning time of the segment to processes. The time is measured in seconds from the beginning of the recording which is time 0.0.
   - end: The ending time of the segment to processes. The time is measured in seconds from the beginning of the recording which is time 0.0.

   Example for an ecf file:

```
<ecf source_signal_duration=”340.00” version=”20060618_1400” language=”english” >
<excerpt audio_filename=”audio/dev04s/english/confmtg/NIST_20020214-1148” channel=”1” tbeg=“0.0”
            dur=”291.34” source_type=”confmtg”/>
<excerpt audio_filename=”audio/eval03/english/bnews/ABC_WNN_20020214_1148.sph” channel=”1”
            tbeg=”0.0” dur=”291.34” source_type=”bnews”/>
…
</ecf>
```

   XML schema for the ecf input file - The XML schema for the ecf file can be found at https://www.nist.gov/itl/iad/mig/tools, select F4DE-3.5.0, download the tar/zip file, then go to KWSEval/data/KWSEval-ecf.xsd.

ASR System Output Format

ASR system output will be a Conversation Time Marked (CTM) file using the .ctm extension and consisting of a stream of lexical tokens formatted in a tab-separated six column ASCII text and use the .ctm extension. Each line represents a single token emitted by the system. The six columns plus an example are shown below.

Table 4: ASR system output

Column	Output	Description
1	file	The waveform file base name (i.e., without path names or extensions).
2	chnl	Channel ID, The waveform channel (e.g., “1”).
3	tbeg	The beginning time of the token, in seconds, measured from the start time of the file.
4	tdur	The duration of the object, in seconds
5	ortho	The orthographic rendering (spelling) of the token.
6	conf	Confidence Score, the probability with a range [0:1] that the token is correct. If conf is not available, omit the column.

Example of four lines for an ASR system output file:

```
7654    A       11.34   0.2     YES     0.9
7654    A       12.00   0.34    YOU     0.8
7654    A       13.30   0.5     CAN     1
7654    A       17.50   0.2     ASK     0.75
```

ASR System Output Evaluation

Four system output possibilities are considered:

1. Correct - system correctly locates [system and reference map] and correctly spells a lexical token item (token) compared to the reference lexical token location and spelling,
2. Deletion (Del) - system output misses the detection of a reference lexical token,
3. Insertion (Ins) - system outputs a lexical token where it does not exist (no mapping) in the reference,
4. Substitution (Subst) - system output correctly locates but miss-spells a lexical token compared to the mapped reference token.

Scoring Procedure

NIST will use the NIST Scoring Toolkit (SCTK) scoring software to calculate WER. The SCTK scoring software is available at https://www.nist.gov/itl/iad/mig/tools. The SCTK software generates an optimum word-to-word mapping (lowest error) between the system output and the reference file.

Lexical Tokenization and Scoring

Lexical tokenization will use space as the delineator.
System scoring includes three steps:
Step 1 Token normalization, - filtering for three types of tokens:

1) Scorable tokens (i.e., reference tokens that are expected to be recognized by the system),

    - All words are transcribed as specified by domain-specific guideline.

2) Optionally deletable tokens (i.e., reference tokens that may be omitted by the system without penalty)

    - Fragments (marked with a -) in the reference transcript. System tokens with token-initial text matching the fragment’s text will be scored as correct (e.g. /theory/ would be correct for fragment/th-/). The same test is applied to the obverse, token-final fragments /-tter/ matching /latter/.
    - The hesitation tags (<hes>).

3) Non-scored tokens/speech segments (i.e., reference tokens/speech segments removed from both the reference and system transcripts prior to scoring)

- Codeswitch tags.
- Speaker change tags.
- Unintelligible speech tags.
- Non-lexical punctuation.
- Non-lexical, speaker-produced sounds (<lipsmack>, <cough>, <breath>, etc. as defined in the data specification document).
- Segments containing the <overlap>, unintelligible [(()) tags], and <prompt> tags.
- Segments containing transcript tokens that were unable to be force aligned in the reference.

Step 2 Reference-to-System alignment - Scorable reference tokens are aligned with system output tokens

- Alignment is performed using Levenshtein distances computed by Dynamic Programming Solution (DPS) to string the alignment
- System tokens are weighted per DPS priori transition costs for alignment computation
  - Substitution = 4, Insertions = 3, Deletions = 3, Correct = 0

Step 3 System performance metric computation

- An overall Word Error Rate (WER) will be computed as the fraction of token recognition errors per maximum number of reference tokens (scorable and optionally deletable tokens):

$$\text{WER} = \frac{(N_{Del} + N_{Ins} + N_{Subst})}{N_{Ref}}$$

where

N_{Del} = number of unmapped reference tokens (tokens missed, not detected, by the system)
N_{Ins} = number of unmapped system outputs tokens (tokens that are not in the reference)
N_{Subst} = number of system output tokens mapped to reference tokens but non-matching to the reference spelling
N_{Ref} = the maximum number of reference tokens (includes scorable and optionally deletable reference tokens)

Appendix IV SAD, KWS, and ASR - System Output Submission Packaging

Each submission shall be an archive file, for example, .tgz or .zip file, named as follows:

<SysLabel>.tgz
or
<SysLabel>.zip

Submit a separate .tgz or .zip file for each system output (e.g., a separate .tgz or .zip file for Primary, Contrastive1, and Contrastive2 systems).

<SysLabel> shall be an alphanumeric [a-zA-Z0-9] that is a performer-assigned identifier for their submission.

There should be no parent directory when the submission file is untarred. The tar command for .tgz or .zip files should be:

```
> tar MySystemSubmissionFile.tgz
```

or

```
> tar MySystemSubmissionFile.zip
```

respectively.

Prior to uploading the submission file to the NIST scoring server, performers will be asked for information about the submission. The scoring server will attach the following information to the submission filename to categorize and uniquely identify the submission:

Field	Information	Method
<TeamID>	[Team]	obtained from login information
<Task>	{SAD | ASR | KWS}	select from drop-down menu
<SubmissionType>	{primary | contrastive}	select from drop-down menu
<Training Condition>	{unconstrained}	default - hard-coded
<EvalPeriod>	{2020}	default - hard-coded
<DatasetName>	{PSC | VAST | Babel}	select from drop-down menu
<Date>	{YYYYMMDD}	obtained from NIST scoring server at submission date
<TimeStamp>	{HHMMSS}	obtained from NIST scoring server at submission time

Below is an example of a resulting filename:
NIST_ASR_primary_uncontrained_2020_PSC_20200415_163026_ MySystemSubmissionFile.tgz

The NIST scoring server will validate the submission file content to make sure the system output files conform to the format described in each task system output format section above.

Each team is required to submit a system description (See Appendix V) for a system's output submitted for evaluation in order to receive the system's score and ranking results. Evaluation results will be provided only after a system description is received and verified to conform to system description guidelines in Appendix V.

Appendix V SAD, KWS, and ASR - System Descriptions and Auxiliary Condition Reporting

Each submitted system must be accompanied by a system description. Documenting each system is vital to interpreting evaluation results and disseminating them to potential end users. System descriptions are expected to be of sufficient detail for a fellow researcher to both understand the approach and the data/computational resources used to train and run the system.

In order to make system description preparation and format as simple and consistent as possible, developers are encouraged to use one of the Microsoft Word IEEE Manuscript Templates for Conference Proceedings, https://www.ieee.org/conferences/publishing/templates.html.

For purposes of having comparable and informative system descriptions the following is recommended to be included.

Section 1: Abstract
Section 2: Notable highlights
Section 3: Data resources
Section 4: Algorithmic description
Section 5: Results on the DEV set
Section 6: Hardware description and timing report

Section 1: Abstract

A few sentences describing the system at the highest level. This should help orient the reader to the type of system being described and how the components fit together.

Section 2: Notable Highlights

For each task, a brief summary of what is different or any notable highlights. Examples of highlights could be differences among systems submitted; novel or unusual approaches, or approaches/features that led to a significant improvement in system performance.

Section 3: Data Resource

Data resources used by the system, including *Linguistic Data Consortium* (LDC) obtained, NIST-provided, or other publically available data.

Section 4: Algorithmic Description/Approach

Details on each component of the system and how each was implemented. You should be very brief or omit altogether components that are standard in the field. For system combinations, there should be a section for each subsystem.

For each subsystem, provide subsections for each major phase. Exclude this if not relevant or if only standard methods are used (e.g., no need to describe how Mel-frequency cepstral coefficients (MFCCs) are computed or 25 ms window and 10 ms step). They may also refer to other subsystems or reference system descriptions if they share components.

Suggested Subsections:

- Signal processing - e.g., enhancement, noise removal, crosstalk detection/removal.
- Low level features - e.g., PLP, Gabor filterbank.
- Speech/Nonspeech –
- Learned features – e.g., MLP tandem features, DNN bottleneck features, etc.
- Acoustic models – e.g., DNN, GMM/HMM, RNN, etc.
- Language Models – methods used
- Adaptation – e.g., speaker, channel, etc. Specify how much of the evaluation data was used as well as the computational costs (memory and time).
- Normalization - Normalizations not covered in other sections
- Lexicon – methods used to update
- Decoding – e.g., Single pass, multipass, contexts,etc.
- OOV handling – e.g., Grapheme, syllable, phoneme,etc.
- Keyword index generation –
- Keyword search –
- System combination methods – e.g., posting list, score, features, lattices.

Section 5: Results on the DEV Set

Report the performance of the submission systems on the "dev" set when using the scoring software provided by NIST (to enable across systems comparison). Teams are encouraged to quantify the contribution of their major system components that they believe resulted in significant performance gains, if any.

Section 6: Hardware Description

Description of the architecture elements that relate directly to system execution time and memory requirements. Report the system execution times to process a single recording for the various system components.

- OS (type, version, 32- vs 64-bit, etc.)
- Total number of used CPUs
- Descriptions of used CPUs (model, speed, number of cores)
- Total number of used GPUs
- Descriptions of used GPUs (model, number of cores, memory)
- Total available RAM
- RAM per CPU
- Used Disk Storage (Temporary & Output)