Open Speech Analytic Technologies Pilot Evaluation

OpenSAT Pilot

Evaluation Plan: version 1.0

1 Introduction

OpenSAT is a new speech analytic technology evaluation series organized by NIST that will begin with a pilot evaluation in the Spring of 2017. The pilot will involve three tasks: *Speech Activity Detection (SAD)*, *Key Word Search* (KWS), and *Automatic Speech Recognition (ASR)*, and will evaluate these speech analytics technologies across three data domains: i) low-resourced languages, ii) speech from video, and iii) public safety communications. The data selected for the OpenSAT evaluation series will target domains expected to be challenging for the current state of technology to process with high accuracy.

Interested researchers may choose to participate in any or all three of the tasks and for any or all three of the data domains[1]. For maximum flexibility the pilot evaluation period will be 1-month and the evaluation framework will permit researchers to upload multiple submissions throughout the month for scoring.

The pilot evaluation will have many of the same characteristics as other NIST Human Language Technology evaluations (registration, data agreements, system descriptions, reporting of results) and will be designed to have a low barrier to entry, to encourage researchers across diverse communities to participate. There will not be a dedicated workshop until after the first formal evaluation that follows the pilot evaluation. The pilot is planned for April 2017 and the first formal evaluation is tentatively planned for February 2018.

Please contact opensat_poc@nist.gov:

- For any OpenSAT relevant information not covered in this document
- To join our general purpose OpenSAT mailing list where future evaluation announcements will be made

Site registration will be required in order to participate. NIST will send an announcement to the OpenSAT_list@nist.gov mailing list with registration instructions once registration is open.

2 Objective

The objective of the pilot evaluation is to establish a baseline of performance levels for existing speech analytics when exposed to a variety of challenging data domains. The reference data for each domain and task will be released for continued R&D after the pilot evaluation period.

The goals of the pilot evaluation include: (1) to develop and exercise an online evaluation framework; (2) to provide a forum for the community to further test and develop speech analytic technologies; and (3) to bring together developers of different speech analytics that may have worked independently from each other; and to promote opportunities for sharing, leveraging or collaboration in system development.

[1] Researchers are strongly encouraged to submit results across all three domains in an effort to identify current challenges to system performance.

3 Schedule

Key Pilot Evaluation Milestones	Date
Registration period	(tbd[2]) through March 31, 2017
Evaluation data available	April 3, 2017
Scoring server active for OpenSAT Pilot evaluation	April, 2017
NIST releases reference data	May 19, 2017
Scoring Server active for continued development	June 1, 2017

4 Data

4.1 Training Data

OpenSAT participants may use any data that is (theoretically) publically available in order to develop and train their system. This includes data that may require a small fee or membership dues in order to access the data. Training data should be thoroughly described in the system description document to assist in the sharing and transferring of research knowledge.

4.2 Development Data

A small amount of development data from each (evaluation) domain will be distributed to registered participants. This dataset is referred to as NISTSAT_DEV01 and can be used for system development or training. The purpose of this dataset is to provide participants with examples of domain data, not necessarily large enough to serve as a true development-test dataset.

4.3 Evaluation Data

The three domains of data are described here. For the pilot evaluation, participants are strongly encouraged to process and upload system outputs for data from all three data domains, but they may choose to process only a specific domain.

Low Resourced Languages (LRL)

NIST will use ***exposed*** data drawn from the IARPA Babel collection, that is a collection of speech recordings from low resource languages. For the 2017 pilot, data will be drawn from the Pashto (Language ID-Name: 104 Pashto) language set. The data consists of conversational telephone speech (CTS). This data permits NIST to leverage previous efforts developed for the IARPA Babel program.

The LRL Babel data presents the following challenges:

- Foreign language (one language per evaluation event)
- Conversational telephone speech (CTS)
- Multiple microphones
- Natural environments

[2] The registration period will open immediately after all data use agreements are finalized.

Speech from Video (SV)

NIST will use ***unexposed*** data drawn from the Video Annotation for Speech Technologies (VAST) database. The VAST data is audio extracted from internet video recordings and presents the following challenges:

- Audio compression
- Diverse topics
- Diverse recording equipment
- Diverse background and environment scenarios

Public Safety Communications (PSC) (Sofa Super Store Fire dispatcher logs)

NIST will use ***exposed*** data of dispatcher logs from the Sofa Super Store Fire that occurred June 18, 2007 in Charleston, South Carolina. This data represents real fire-response operational data that cannot be duplicated through controlled scientific collection. Note, that this data does contain sensitive and disturbing content (e.g., pleas from trapped fire fighters) and sensitivity should be observed when using this data. The data presents multiple challenges:

- Land Mobile Radio (LMR) transmission effects
- Speech under cognitive and physical stress
- Varying background noise types
  Varying background noise levels

The content and makeup of the **evaluation data** is described in Table 1.

Table 1: Data for use in OpenSAT Pilot Evaluation

Domain	Tasks	Language
LRL (Babel)	SAD	Pashto
	ASR	Pashto
	KWS	Pashto
SV (VAST)	SAD	Arabic Mandarin English
PSC	SAD	English
	ASR	English
	KWS	English

5 Tasks - Overview and Performance Metrics

5.1 Speech Activity Detection (SAD)

The goal in the SAD task is to automatically detect the presence of speech segments in an audio recording. Audio recordings will be of variable duration. System output will be scored based on comparing system-identified speech segments (start and end times) to a human reference (annotation of the audio recordings). Correct, incorrect, and partially correct segments will determine error probabilities for the system.

SAD performance will be measured by the Detection Cost Function (DCF) value that is a function of false acceptance (false alarms) and false rejection (missed detections) rates of speech against the reference. System developers will determine and select their system setting (i.e., the detection threshold θ) with the goal of minimizing the DCF value.

5.2 Key Word Search (KWS)

The goal of the KWS task is to automatically detect all occurrences of a "keyword" (pre-defined single word or phrase) in an audio recording, transcribed in a language's original orthography (i.e., spelling convention), with beginning and end time-stamps for each detected keyword.

KWS performance will be measured by the Term-Weighted Value (TWV) that is a function of false acceptance (false alarms) and false rejection (missed detections) rates of a keyword relative to the reference. Actual Term-Weighted Value (ATWV) will be a measure of the calculated hypothetical optimal system setting.

5.3 Automated Speech Recognition (ASR)

The goal of the ASR task is to automatically produce a verbatim, case insensitive transcript of all words spoken in an audio recording.

ASR systems will output a stream of Conversation Time Marked (CTM) words (lexical tokens) reporting the token's begin and end time within the recording, a confidence score value in the range [0:1] indicating the system's confidence that the token is correct, and lexical subtype information.

ASR performance will be measured using word error rate (WER), calculated as the sum of errors (deletions, insertions and substitutions) divided by the total number of words from the reference.

6 Evaluation Rules

There is no cost to participate in the OpenSAT evaluation series. Participation in the pilot evaluation is open to all who are able to comply with the evaluation rules set forth in this plan.

The 2017 OpenSAT pilot evaluation is an open evaluation where the test data is sent to the participants who will process the data locally and submit their system outputs to NIST for scoring. As such, the participants have agreed to process the data in accordance with the following rules:

- Investigation of the evaluation data prior to submission of all systems outputs is not allowed. Human probing is prohibited.

- 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. e.g., the search results for each keyword are to be output prior to performing detection on the next keyword.
- *Language Specific Peculiarities (LSP) Resources*, the LSP documentation contains a full inventory of phones for the language. Evaluation participants are allowed to leverage that information. The LSP may include links to resources that can be utilized without using the Other LR designation. (There is no guarantee that phonemes for all borrowings are covered in the LSP.)

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

  - Participants can publish results for their own system but will not publicly compare their results with other participants (ranking, score differences, etc.) without explicit written consent from the other participants.
  - 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.
  - At the conclusion of the evaluation, NIST will generate a report summarizing systems results for conditions of interest. These results/charts will not contain participant names of the systems involved. Participants may publish or otherwise disseminate these charts, unaltered and with appropriate reference to their system.
  - The report that NIST creates cannot be construed or represented as endorsements for any participant's system or commercial product, or as official findings on the part of NIST or the U.S. Government.

7 Evaluation Protocol

To facilitate information exchange between the participants and NIST, all evaluation activities will be conducted over a web-interface.

7.1 Evaluation Account

Participants must sign up to establish an evaluation account to perform required activities:

- Register for the evaluation
- Signing the data license agreement
- Data access
- Upload the submission and system description.

To sign up for an evaluation account, go to https://sat.nist.gov[3]. The password must be at least 12 characters long and must contain a mix of upper and lowercase letters, numbers, and symbols.

After the evaluation account is confirmed, the participant is asked to join a site or create one if it does not exist. The participant is also asked to associate his site to a team or create one if it does not exist. This allows multiple members with their individual accounts to perform activities on behalf of their site and/or team (e.g., make a submission) in addition to performing their own activities (e.g., requesting workshop invitation letter).

- A site is defined as a single organization (e.g., NIST)
- A team is defined as a group of organizations collaborating on a task (e.g., Team1 consisting of NIST and LDC)
- A participant is defined as a member or representative of a site who takes part in the evaluation (e.g., John Doe)

7.2 Evaluation Registration

One participant from a site must formally register his site to participate in the evaluation by agreeing to the terms of participation, described in Section 6 Evaluation Rules above.

7.3 Data License Agreement

One participant from each site must sign the data license agreement to obtain the data.

7.4 Submission Requirements

Each team is required to submit a system description. The evaluation results are given only after the system description is received and verified to conform to the system description guidelines.

[3] This website will be activated as soon as this project clears NIST Human Subjects Protection Office.

Appendix I – System Input

SAD system inputs include:

- Audio file
- Test Definition File

KWS system inputs include:

- Audio file
- Experiment Control File (ECF)
- KW List file

ASR system inputs include:

- Audio file
- Experiment Control File (ECF)

Audio files

Currently, audio files will be in SPHERE format.

Test Definition Files (SAD)

Test definition files are XML formatted files that define the test to be performed on the audio files.

In the File element:

- the `id` attribute's value ties the Test Definition to the system output
- the `file` attribute is a filename in a directory, usually with a directory path (relative to the current directory).

Test Definition File example:

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

Experimental Control Files (ECF) - KWS and ASR

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 to evaluate the system on. 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 XML scheme for an ECF file can be found in the F4DE software package. 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.

ECF example:

```
<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>
```

KWList Files

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

```
/KWlist
    /LRL/[keyword1 in xml format].xml
    /SV/[keyword2 in xml format].xml
    /PSC/[keyword3 in xml format].xml
```

- **KWList File Format Description**

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 scheme for a KWList file can be found in the F4DE software package (make this a footnote). 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.

The following is an example 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>
```

`

Appendix II – System Output and Submission

SAD System output submission:

SAD system output will be formatted as a tab-separated ASCII text file with nine columns described in Table 2.

Table 2: SAD system output

Column	Output	Description
1	Test	Test Definition File name (name of an XML file containing the test definition content)
2	TestSet ID	contents of the `id` attribute of the `TestSet` tag
3	Test ID	contents of the `id` attribute of the `TEST` tag
4	Task	`SAD` <== a literal text string, without quotation marks
5	File ID	contents of the `id` attribute of the `File` tag
6	Interval start	an offset, in seconds, from the start of the audio file for the **start** of a speech/non-speech interval
7	Interval end	an offset, in seconds, from the start of the audio file for the **end** of a speech/non-speech interval
8	Type	In system output: "speech" or "non-speech" (with no quotation marks). In the reference: S, NS, or NT (for Speech, Non-Speech, and No Transmission).
9	Confidence (*optional*)	A value in the range 0.0 through 1.0, with higher values indicating greater confidence about the presence/absence of speech

Table 3: Four lines shown as an example for a SAD system output file:

1	2	3	4	5	6	7	8	9
Test	TestSetID	TestID	Task	SampleID	Start Time	Stop Time	Type	Confidence
Example output file contains:								
Cluster_01	OpenSAT17_01	Babel	SAD	20703_2017	0.0	4.61	non-speech	1
Cluster_01	OpenSAT17_01	Babel	SAD	20703_2017	4.61	7.08	speech	1
Cluster_01	OpenSAT17_01	Babel	SAD	20703_2017	7.08	7.49	non-speech	1
Cluster_01	OpenSAT17_01	Babel	SAD	20703_2017	7.49	9.34	speech	1

KWS System output submission:

KWS system output will be formatted as three hierarchically organized xml nodes in a 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 (e.g., KWSEval-kwslist.xsd) can be found in the F4DE software package available at https://www.nist.gov/itl/iad/mig/tools. The schema is the authoritative source located under /F4DE-3.3.0/KWSEval/data/.

Note: For participants who prefer to work in csv (tab-delimited ASCII) format, see Appendix VII.

The three nodes for a KWSList file are:

1. kwslist – the system inputs and parameters used to generate the results.
2. detected_kwlist – a collection of "kw" nodes which are the putative detected keywords.
3. kw – six attribute fields for the location and detection score for each detected keyword.

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.

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:

- kwid: The keyword id from the KWList file.
- search_time: (optional for backward compatibility) A floating point number indicating the number of CPU seconds spent searching the corpus for this particular keyword.
- 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".

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="YES"/>
    <kw file="NIST_20020214-1148_d05_NONE" channel="1" tbeg="45.5"  dur="0.3"  score="4.65"  decision="NO">
    </kw>
</detected_kwlist>
</kwslist>
```

ASR System output submission:

ASR system output will be formatted as tab-separated six column ASCII text Conversation Time Marked (CTM) files and use the .ctm extension with six columns as shown below. Each line represents a single token emitted by the system.

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.

Table 5: Four lines as an example for an ASR system output file:

1	2	3	4	5	6
file	chnl	tbeg	tdur	ortho	conf
Example output file contains:					
7654	A	11.34	0.2	YES	-6.763
7654	A	12.00	0.34	YOU	-12.384530
7654	A	13.30	0.5	CAN	2.806214
7654	A	17.50	0.2	AS	0.537922

Appendix III - System Descriptions and Auxiliary Condition Reporting

Documenting each system is vital to interpreting evaluation results and disseminating systems 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. As such, each submitted system, (determined by unique experiment identifiers), must be accompanied by a system description. An acceptable system description should include the following information:

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

In order to make system description preparation as simple as possible, developers are encouraged to write a single detailed description (see below) using the IEEE ICASSP template.

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
A few paragraphs on the major differences between this system and a "conventional" system. Questions often answered are: How is this system different from a system published in a conference proceedings a few years ago? How is it different from all the other teams' submissions?

Section 3: Data Resource
This section describes the data resources used by the system and for which major components the resources were used.

Section 4: Algorithmic Description
Sufficient detail should be provided for each component of the system such that a practitioner in the field can understand how each phase 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, there should be subsections for each major phase. They may be excluded if not relevant or if only standard methods are used (e.g. no need to describe how MFCCs are computed or tell us 25ms window and 10ms step). They may also refer to other subsystems or referent 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

The performance of the submission systems on the "dev" set should be reported, using the scoring software provided by NIST (to enable across system comparisons). Teams are encouraged to quantify the contribution of their major system components that they believe resulted in significant performance gains.

Section 6: Hardware description

Requirements on the description of architecture will be here. Reporting of the following environment elements relate directly to the reporting of time and memory requirements.

- 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)

System execution times to process a single recording must be reported for the various system components as well.

Appendix IV – 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), (False Alarm) - system incorrectly identifies speech in a segment where the reference identifies the segment as non-speech, and
4. False Negative (FN), (False Reject) - 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 in a system's segmentation of the test audio files. Missing (failing to detect) actual speech is considered a more serious error than identifying speech a little before it actually begins or a little passed after it actually ends. Accordingly, a half-second (0.5 second) collar at the beginning and end of each speech region will be not scored. If a segment of non-speech between collars does not last at least a tenth of a second (0.1 sec) then the collars involved are expanded so that they will merge (for example, no resulting non-speech segment with a duration of just 0.099 seconds). 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 a tenth of a second or else the collar will expand. In all other circumstances the collars will be exactly the nominal length.

Figure (1) illustrates the relationship between human annotation, scoring regions resulting from application of the collars, a possible system output, and the resulting time intervals from the four system output possibilities shown above.

The scoring collars also compensate for ambiguities in noisy channel annotation. Non-speech collars of two seconds in length, shown above the annotation, define regions that will not be scored. As can be seen, collars are applied to the annotations to determine the parts of the speech and non-speech that are scored.

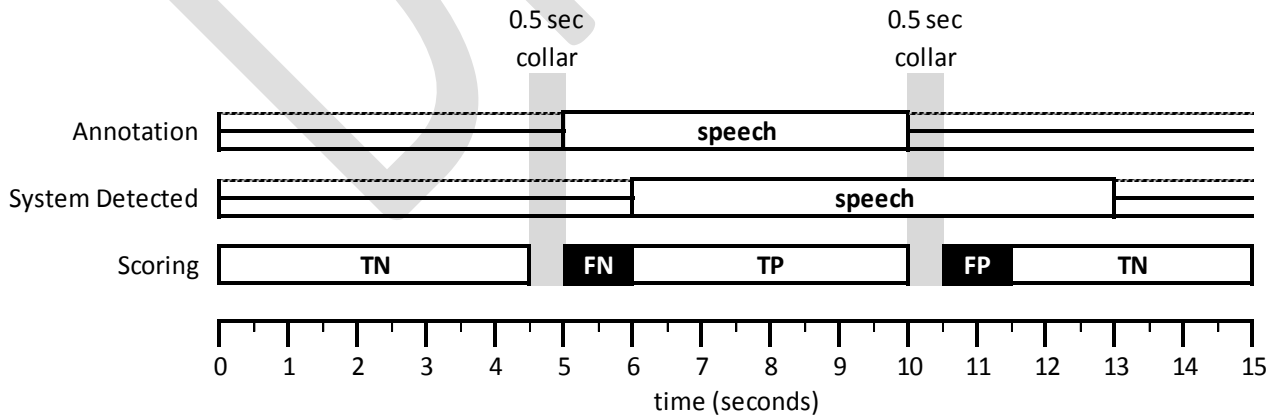

Figure 1: After collar application, systems are not scored on false alarms within the 0.5 second collar regions from speech boundaries.

The four system output possibilities determine probability of a false alarm (P_{FA}) and probability of missed speech (P_{Miss})

P_{FA} = (false positive) – detecting speech where there is no speech

P_{Miss} = (false negative) - not detecting speech where there is speech

$$P_{FA} = \frac{total\ FA\ (false\ positive)\ time}{annotated\ total\ nonspeech\ time}$$

$$P_{Miss} = \frac{total\ Miss\ (false\ negative)\ time}{annotated\ total\ speech\ time}$$

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

$$DCF\ (\theta) = 0.75 \times P_{\text{Miss}}\ (\theta) + 0.25 \times P_{\text{FA}}\ (\theta)$$

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

Developers are responsible for determining a hypothetical optimum setting (θ) for their system that minimizes the DCF value, e.g., analysis of a Detection Error Tradeoff (DET) curve, Receiver Operating Characteristic (ROC) curve using the system's P_{Miss} and P_{FA} rates, or by any other alternative self-preferred method.

Appendix V – KWS System Output Evaluation

Keyword detection performance will be measured as a function of Missed Detection and False Alarm 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) or (Miss) - system misses detection or location of a keyword, or miss-spells a keyword
4. (FP) or (FA) - system detects a keyword that is not in the reference or not in the correct location

Scoring protocol will be the "Keyword Occurrence Scoring" protocol that evaluates system accuracy based on the three steps below.

1. Reference-to-system keyword alignment
   - The KWS evaluation uses the Hungarian Solution to the Bipartite Graph matching problem[9] 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 (or FA), and Miss (or 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)

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

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 searches for the system's optimum decision score threshold
  - Called the "**Maximum Term Weighted Value**" (MTWV)

Appendix VI – 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 - system output misses the detection of a reference lexical token,
3. Insertion - system outputs a lexical token where it does not exist (no mapping) in the reference,
4. Substitution - system output correctly locates but miss-spells a lexical token compared to the mapped reference token.

Scoring Procedures

NIST will use the NIST SCTK toolkit scoring software to calculate WER. The SCTK scoring software, available at https://www.nist.gov/itl/iad/mig/tools, 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:

There are three types of tokens that are considered in scoring:

1. Token normalization, filtering for:
   - Scorable tokens (i.e., reference tokens that are expected to be recognized by the system),
     - All words transcribed as specified by the Babel Data Specification Document.
   - 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>).
   - non-scored tokens (i.e., reference tokens 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).

- Non-scored Speech Segments

  Segments containing the <overlap>, unintelligible [(()) tags], and <prompt> tags will not be scored.
  In addition, segments containing transcript tokens that were not able to be force aligned in the reference will not be scored.

2. Reference-to-System token 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
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):

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

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 VII – Converting KWS System Output csv files to xml

KWS system output formatted as a tab-separated csv text file with 12 columns is described in Table 4. A csv to xml conversion tool will be available at https://sat.nist.gov.

Column	Output	Description
1	KWList file name	The name of the KWList file used to generate this system output
2	Language	Language of the source material
3	System ID	A text field supplied by the participant to describe the system
4	KW ID	The keyword id from the KWList file
5	Search time	(optional for backward compatibility) A floating point number indicating the number of CPU seconds spent searching the corpus for this particular keyword.
6	OOV count	(optional) 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"
7	file	The basename of the audio file as specified in the ECF file
8	channel	The channel of the audio file where the keyword was found
9	tbeg	Offset time from the start (0.0 secs) of the audio file where the keyword starts
10	dur	The duration of the keyword in seconds
11	score	The detection score indicating the likelihood of the detected keyword
12	decision	[YES | NO] The binary decision of whether or not the keyword should have been detected to make the optimal score

KWS system output: Below is an example system output in csv format for keyword ID "dev06-0001":

1	2	3	4	5	6
KWList file name	Language	System ID	KW ID	Search Time	OOV count
Example output file contains:					
dev06.tlist.xml	English	System1	dev06-0001	24.3	0

7	8	9	10	11	12
File	Channel	tbeg	dur	score	Decision
Example output file contains:					
NIST_20020214_d05	1	6.956	0.53	4.115	YES

Tab-separated:
dev06.tlist.xml<TAB>English<TAB>System1<TAB>dev06-0001<TAB>24.3<TAB>0<TAB>NIST_20020214_d05<TAB>1<TAB>6.956<TAB>0.53<TAB>4.115<TAB>YES