Usage of the Label Tool v 2.1

Contents

General

This is the description of a tool designed to label or transcribe telephone voice Human-Machine dialogs. There's special support as a testing tool for emotional speech classification. The tool comes as a client/server solution, i.e.. the audio samples get managed by a Java server application and the interface is provided by a Java applet. This has the advantage, that several annotators can work on the same audio database. A user management has not yet been realized.

Both, client and server, are controlled by configuration files.

Server: res/voiceClassServer.properties

Client: labeltool.html

Prerequisites

Software

Being in Java, the tool should run with all Java enabled platforms. It was tested with Windows XP and Linux Suse 10.4.

Audio format

The audio format currently supported is 16bit PCM linear Big Endian coded. A-law support is provided by the Java sound API but experimental and functionality like pausing or recording is not working then.

The sample rate can be set in the parameter configuration “labelTool.html” and res/voiceClassServer.properties file..

Start Up

Server

The server application gets started as a java command.
An example startup is given by the file “startServer.bat”.

Numerous options are available and get shown if invoked with argument “-h”:

```
$ java -jar VoiceClassifier.jar -h
VoiceClassVoiceClassServer version 1.7
Usage: program [-h ] [-cf <String>] [-rd <String>] [-fl <String>] [-aft
<String>] [-srt <String>] [-port <String>] [-pe ] [-pp ] [-pf ] [-pl ] [-pi
] [-pt ] [-pnt ] [-pc ] [-prtl ] [-al ] [-at ] [-ar ] [-gw ] [-rl ] [-wer ]
[-sclite ] [-mixAll ] [-removeAnnotationFiles ] [-classify ] [-train ] [-
noExtract ] [-eval ] [-removeLabels ] [-removePreds ]
```

```
      -h 'print usage'  Default: false
      -cf 'configuration file'  Default: res/voiceClassServer.properties
      -rd 'directory with recordings'  Default:
      -fl '<textlist with audiofiles>.
            Format: filePath label_1 label_2 ... label_i'  Default:
      -aft 'Set audio file type, e.g. wav or pcm'  Default:
      -srt 'Set audio sample rate, e.g. 8000 or 16000'  Default:
      -port 'Set port number, e.g. 6666'  Default:
      -pe 'Print evaluation format to stdout.
            Format: filepath <string label> <prediction category>.
'  Default: false
      -pp 'Print prediction to stdout.
            Format: filepath <prediction category>.
'  Default: false
      -pf 'Prints file info to stdout.
            Format: filepath size'  Default: false
      -pl 'Prints labels to stdout.
            Format: filepath label_1 label_2...label_i'  Default: false
      -pi 'Prints labels as integers to stdout.
            Format: filepath label_1 label_2...label_i'  Default: false
      -pt 'Prints transcriptions to stdout.
            Format: filepath _transcript'  Default: false
      -pnt 'Prints files without transcriptions to stdout.
            Format: filepath'  Default: false
      -pc 'Prints categories to stdout.
            Format: filepath filesize C_all C_l1 C_l2...C_li
            (C=category).'  Default: false
      -prtl 'Prints transcriptions and labels to stdout (if BOTH exist).
            Format: <filepath> <transcript> <label>
            example: recs/rec.wav "bla bla" -3'  Default: false
      -al 'Adds labels from textlist.
            Format: filepath label_1...label_n'  Default: false
      -at 'Adds transcriptions from textlist.
            Format: filepath transcript_1...transcript_n'  Default: false
      -ar 'Adds recognition results from textlist.
            Format: filepath recognized word_1...recognized word_n'
Default: false
      -gw 'Generate (syntheseize) wav-files in textlist according to
transcriptions'  Default: false
      -rl 'Replaces/adds given labels in textlist to all files'  Default:
false
      -wer 'compute word error rate for loaded audio files (must be
transcribed and recognized)'  Default: false
```

```
      -sclite 'sclite wer comuptation option'  Default: false
      -mixAll 'Mix sound to all files'  Default: false
      -removeAnnotationFiles 'Removes all annotation files
            (containing transcription and labels)'  Default: false
      -classify 'Classifiy all files.'  Default: false
      -train 'Train a model from all files.'  Default: false
      -noExtract 'If set, features will not be extracted before model
training.'  Default: false
      -eval 'Evaluate given list internally (samples MUST have associated
annotation files with labels and predictions)'  Default: false
      -removeLabels 'Removes all labels for files given in textlist'
Default: false
      -removePreds 'Removes all predictions for files given in textlist'
Default: false
```

The only mandatory argument is a file list containing paths to the audio files, given as in text file format, one line per entry, an example (testlist.txt) should be given with the distribution.

./data/SMX/2006/03March/01/15-40-27-SYS6TN3-39/utt00000001.wav

Note that the audio samples should NOT have to be under a common directory. The parent directory is used the dialog name, the whole parent path as the dialog identifier, thus enabling dialogs with the same name.
The results of the transcription/labeling will be written to a file with the same path and name as the audio file but extension ”txt”.
The server program is configurable via the res/voiceClassServer.properties file:

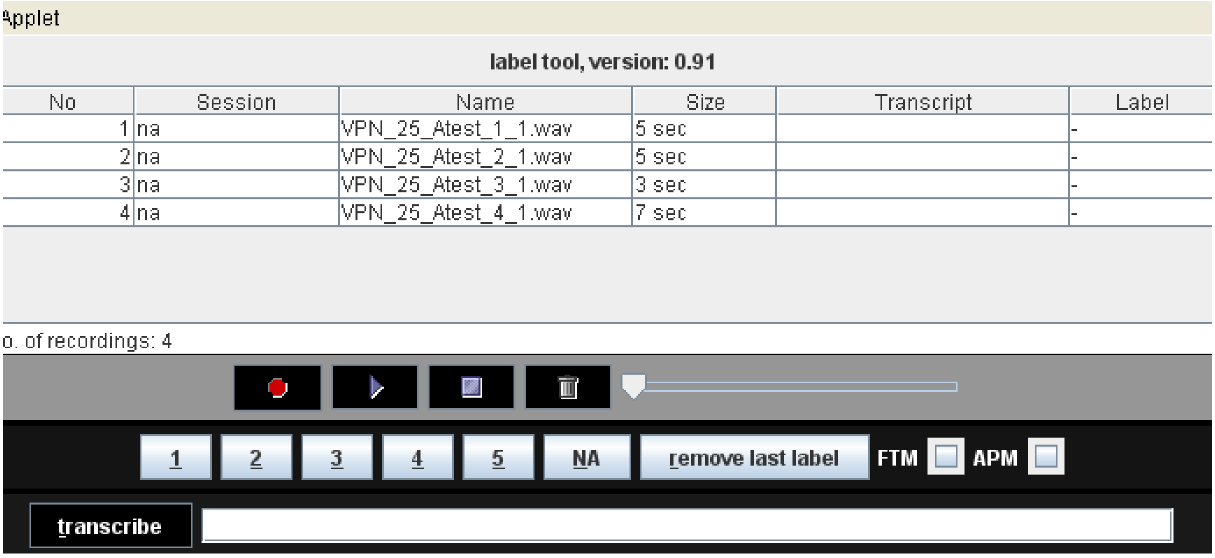
```
### MISC
   # port of recording server
port=6666
   # identifier for label-lines in annotation file
labelIdentifier=LABELS:
   # identifier for transcription-lines in annotation file
transcriptIdentifier=TRANSCRIPTION:
   # identifier for transcription-lines in annotation file
recognitionIdentifier=RECOGNIZED:
   # identifier for prediction-lines in annotation file
predictionIdentifier=PREDICTION:
   # extension for audio-accompanying annotation file
labelFileExtension=txt
   # extension of audio files (others will be disregarded)
audioFormat=wav
   # more debug output @see logConfig.xml
garrulous=false
   # sample rate of audio files
sampleRate=8000
   # feature extraction method
usePraat=false
useOpenEar=true
   # classification method
useWEKA=true
   # whether feature extraction before model building starts from scratch
   # or extracts only files that have no prediction yet.
additiveTraining=false
   # label to category mapping: pairs of category descriptors
   # assigned to minimum labels, e.g. 1,N;2,NA means category "N"
   # is assigned for values between 1>=x<2.
   # MUST be in ascending order!
categories=-1,NA;0,G;1,N;2,L;3,A
   # categories=-1,NA;1,N;3,A
#categories=-1,NA;1,x;2,f;3,m;4,f;5,m;6,f;7,m
# categories=-1,1;2,2;3,3;4,4;5,5;6,6;7,7
```

```
   # character encoding, e.g. ISO8859-1 or UTF-8
# charEnc=ISO8859-1
charEnc=UTF-8
   # classifier type, smo, j48 or naiveBayes
classifier=smo
   # command to execute on selected audiofiles
execCmd=mplayer
   # whether to use the spellchecker
withSpellChecker=false
#### PATHES
normalizeRules=res/normalizeRules.txt
normalizeVocab=res/normalizeVocab.txt
emlUrl=http://192.168.188.129:8080/EMLQueue/Transcribe
recognitionTmpDir=C:/temp/
recordingDir=recordings
hypothesisFile=hypothesis.txt
referenceFile=references.txt
resultFile=results.txt
scliteTool=tools/sclite.exe
tempFeatFile=tmp/tmpFeat.txt
praatCommand=./tools/praatcon.exe
praatScriptFile=res/extractFeatures.praat
trainFile=res/train.arff
testFile=res/test.arff
arffHeaderFile=res/ARFFHeader.txt
modelFile=res/classifier.model
#modelFile=res/agenderTrainIS10.model
logConfig=res/logConfig.xml
openEarCommand=./tools/SMILExtract.exe
openEarConfig=res/IS10.anger_functs.conf
openEarConfigAgenderIS10=res/IS10.agender_functs.conf
openEarConfigIS09=res/emo_IS09.conf
openEarConfigLarge=res/emobase.conf
useTTS=svox
#useTTS=emofilt
#useTTS=ivona
mbrola=tools/mbrola/mbrola
txt2pho=c:/Programme/Txt2pho/txt2pho.exe /G=
tmpTxtFile=tmp/tmp.txt
tmpWavFile=tmp/tmp.wav
tmpPhoFile=tmp/tmp.pho
tmpPhoEmotionFile=tmp/tmpE.pho
mbrolaDB=tools/Mbrola/voices/
mbrolaVoices=de6,male
# mbrolaVoices=de1,female;de2,male;de3,female;de4,male;de6,male;de7,female
emofilt=java -jar tools/emofilt/emofilt_fat.jar
emofiltDB=-cf tools/emofilt/emofiltConfig.ini
emotions=happiness
# emotions=boredom yawning despair happiness sadness joy fear hotAnger
neutral
wavGenOutPrefix=
formatOption=
svoxzh-CNfemale=svox-dz0co0zh-CN22.pil
ttsSex=male
ttsLang=de-DE
svoxde-DEfemale=svox-gl0co0de-DE22.pil
svoxde-DEmale=svox-ag5co0de-DE22.pil
svoxen-USfemale=svox-lh0co0en-US22.pil
svoxen-USmale=svox-mh5co0en-US22.pil
svoxCmd=wine /media/OS/tts/Svox/svox_431/svox.exe -d d:/tts/Svox/svox_431/
-f
mixFile=res/streetnoise16kHz.raw
mixFactor=0.5
mixExtension=_mix
ivonaCmd=ivonacl -f
```

`ivonaVoice=hans`

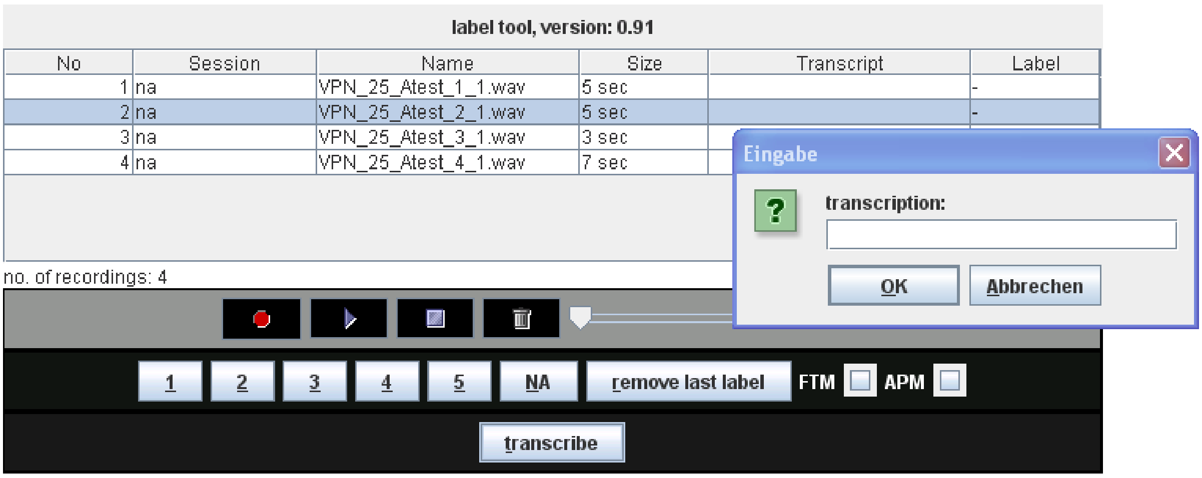
Client

The client application is started by either opening the “labelTool.html” file with the java appletviewer application or a Java-enabled Web-browser.
Here’s a screenshot



The audio files are displayed in a table, given name, dialog, size, transcription and possible labels. If the tool is used with an emotion prediction system (like Sympalog), this is shown also. The emotion recognition functionality can be disabled via the applet parameters.

The following screenshot shows the interface with external transcription window.



The client application can be configured by editing the “labelTool.html” file, all configurable parameters can be edited there. This would include

- the server’s address and port,
- the sample rate of the audio files
- the interface labels (and language)
- general usage features like e.g. turning emotion recognition support on/off, fast transcription mode on/off and the like..

Usage

Labeling

Once the files are loaded (depending on the number of files, this may take several seconds, the progress is shown on the output console), one can select an audio file and label it by clicking on the label buttons (1-5, NA) or, alternatively, pressing ALT-(1-5) or ALT-n.
The number and assignment of label buttons is specified in the parameter files.
In the labeltool.html file all buttons are specified like this

```
<PARAM Name="buttons" VALUE="s1 s2 s3 s4 s5 s6">
<PARAM Name="s1.label" VALUE="1">
<PARAM Name="s1.value" VALUE="1">
<PARAM Name="s1.tooltip" VALUE="press for zero anger">
<PARAM Name="s1.short" VALUE="1">
<PARAM Name="s2.label" VALUE="2">
<PARAM Name="s2.value" VALUE="2">
<PARAM Name="s2.tooltip" VALUE="press for not sure anger">
<PARAM Name="s2.short" VALUE="2">
….
```

i.e., an additional button must be added to the “buttons” parameter and then its values specified.
The assignment of numeric labels to category strings is done by a parameter called categories, which must be specified in the labeltool.html file as well as in the voiceClassServer.properties file.

```
   # label to category mapping: pairs of category descriptors
   # assigned to minimum labels, e.g. 1,N;2,NA means category "N"
   # is assigned for values between 1>=x<2.
   # MUST be in ascending order!
categories=-1,NA;1,x;2,f;3,m;4,f;5,m;6,f;7,m
```

and

```
<PARAM Name="categories" VALUE="-1,NA;0,G;1,N;2,U;3,A;4,A;5,A">
```

Transcribing

Depending on the parameter “transciptionInline” given in the “recorder.html” file, two modes are possible:

Inline

Type the transcription in the transcribe field and confirm with the enter key. The active audio file will be transcribed.

External

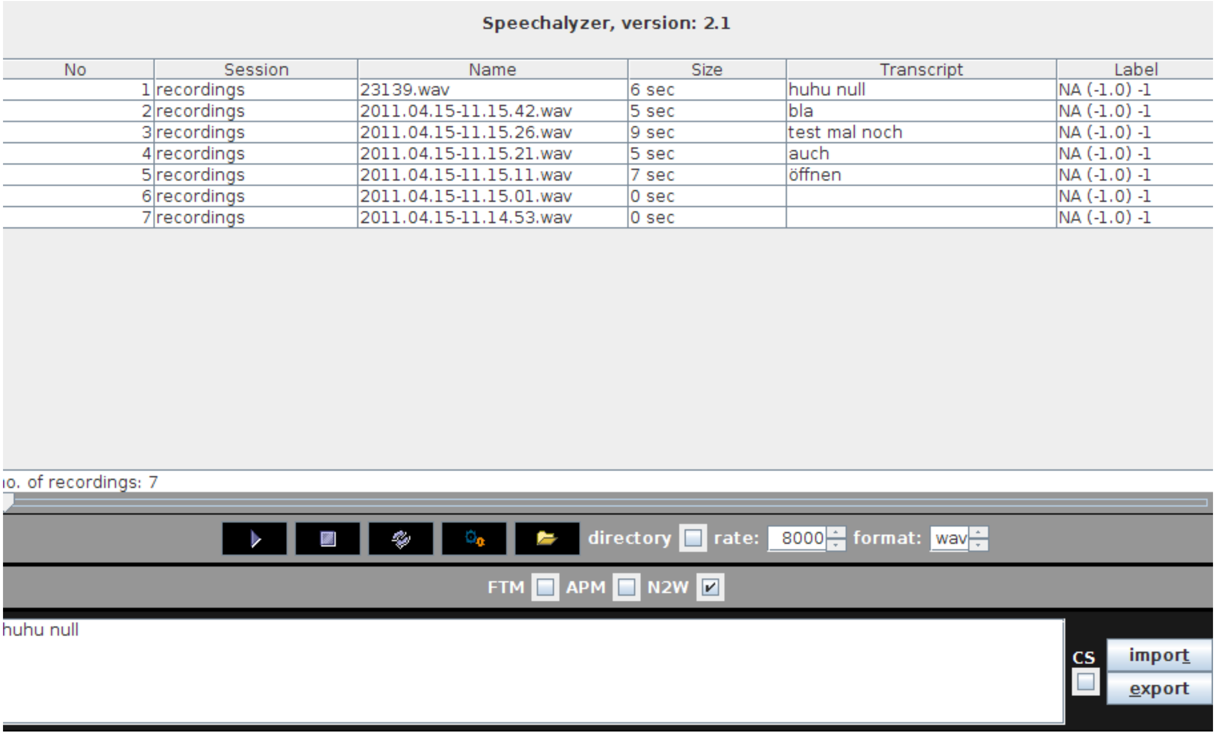
Click the transcribe button and an external window will appear.

Fast mode

If the “FTM” (short for “fast transcription mode”) checkbox is activated, the next audio file in the table will be selected and played automatically after transcription or labelling. Which of the actions triggers the fast mode, depends on the “fastModeSetsAnger” parameter given in the “recorder.html” file.

Transcription only

If the applet parameter “transcribeOnly” is set to “true”, all buttons that are not needed for transcription are invisible. The transcription area can be set to large values by the parameters “transcriptionField.width” and “transcriptionField..height”.
Here’s a screenshot with hidden number buttons and large transcription area:



Shortcuts

For the insertion of certain keywords, an applet parameter named “replacements” can be used. It consists of a list of colon-separated key-value pairs. If the key is typed in the text area, followed by the <ESC> key, the value is inserted, replacing the key.

Abbreviations

For the automatic replacement of strings, e.g. abbreviations, an applet parameter named “abbreviations” can be used. It consists of a list of colon-separated key-value pairs. If the key is typed in the text area, the value is inserted, replacing the key.

Tagging of words

For the automatic tagging of strings, i.e. surrounding them by xml-elements, an applet parameter named “classes” can be used. It consists of a list of colon-separated key-value pairs. If the key is typed in the text area followed by the CTRL key, the values is used as an XML element and surrounds the word that stands before the cursor position.

Prefixing words

To prefix a word with a special sign, e.g. changing the word “example” → “&example” , an applet parameter named “prefixes” can be used. It consists of a list of colon-separated key-value pairs. If the key is typed in the text area followed by the F1 key, the value is used as a prefix for the last word of the text.

Number conversion

If the “N2W” (number to word) checkbox in the GUI is activated, all digits get converted into words after a blank is typed.

Playback and Recording

Playback and recording can be controlled by the buttons in the middle of the interface, a pause function is provided by clicking the “stop” button during playback, it will then change to a

“resume-playback” button. This works only with raw PCM format, NOT for a-law coded audio. Recording also works only for PCM format.
The slider can be used to jump to specific audio positions.
If the “APM” (short for “audio playback mode”) checkbox is activated, the next file is selected automatically after playback.

Fast Playback mode

If the “APM” (short for “audio playback mode”) checkbox is activated, the next file is selected automatically after playback.