Usage of the Label Tool v 0.91

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, though.

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 8kHz 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.
Other samplerates are possible by changing the labelTool's parameter configuration in the "labelTool.html" 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 without arguments:

```
java -cp bin recserver.RecServer
RecServer version 1.8
    USAGE
    RecServer <textlist with audiofiles>.
        Format: filePath label_1 label_2 ... label_i
    [path to sympaConfig].
        If given, Sympalog libraries are loaded.
    OPTIONS:
    -pst . Print sympalog training to stdout.
        Format: filepath <A|N>.
        Unsure or garbage files are disregarded.
    -psr . Print sympalog result to stdout.
        Format:
         filepath <reference: A|N> <non-anger 0..1> <anger 0..1>.
    -pl . Prints labels to stdout.
        Format: filepath label_1 label_2...label_i
    -pt . Prints transcriptions to stdout.
        Format: filepath transcript
    -pnt . Prints files without transcriptions to stdout.
        Format: filepath
    -at . Adds transcriptions from textlist.
        Format: filepath transcript
    -ps . Prints string labels to stdout.
        Format: filepath filesize SL_all SL_l1 SL_l2...SL_li
        (SL=StringLabel,
        "A" is for anger,
        "N" for non-anger,
        "U" for unsure and
        "G" for garbage).
    -classify . Classifiy all files. [path to sympaConfig] must be given
    -addLabels . Adds given labels in textlist to all files
    -addSympaResults . Adds given sympalog results in textlist
        (format: see "-psr" option) to all files
    -removeAnnotationFiles . Removes all annotation files
        (containing transcirption and labels)
    -removeLabels . Removes all labels for files given in textlist
```

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.

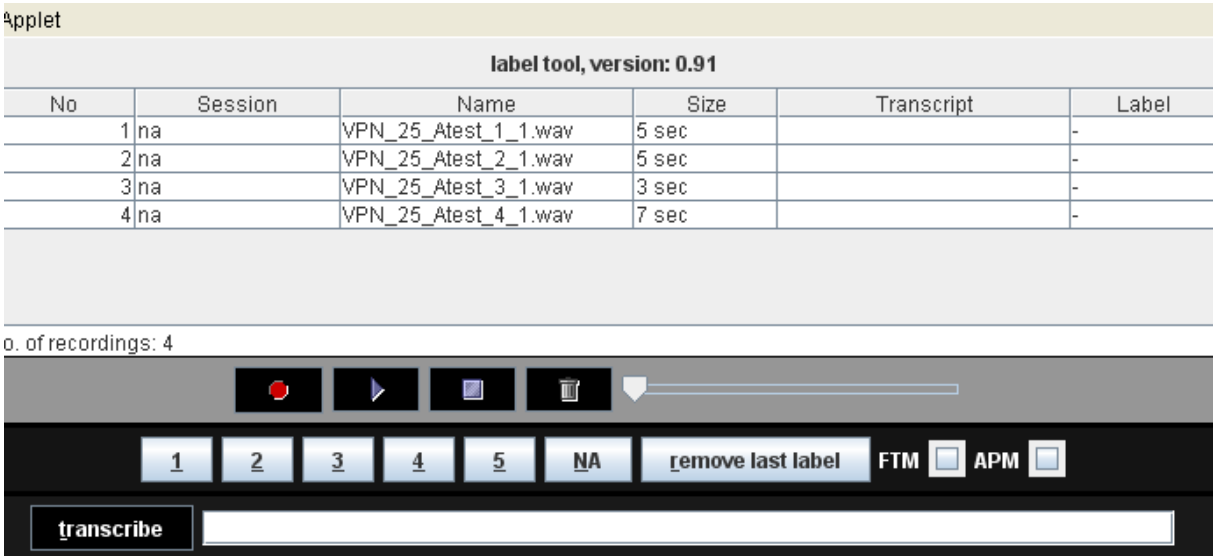
./data/SMX/2006/03March/01/15-40-27-SYS6TN3-39/utt00000001.wav

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 labelToolProperties.txt file:

sympalog|false -> if true, EXTERNAL sympalog emotion recognizer is initialized
port|6666 -> port for the server
labelIdentifier|LABELS: -> string to start the label line
transcriptIdentifier|TRANSCRIPTION: -> string to start transcription line
audioFormat|wav -> audio format and file extension
#audioFormat|pcm
#audioFormat|raw
#audioFormat|snd
recordingDir|recordings -> local parent directory for tool-made recordings

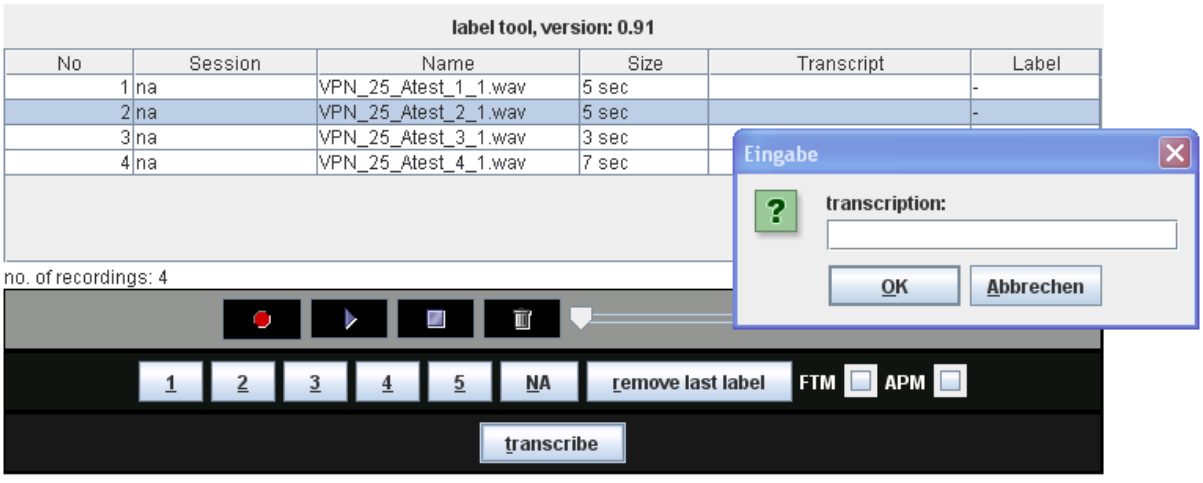
Client

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.

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.

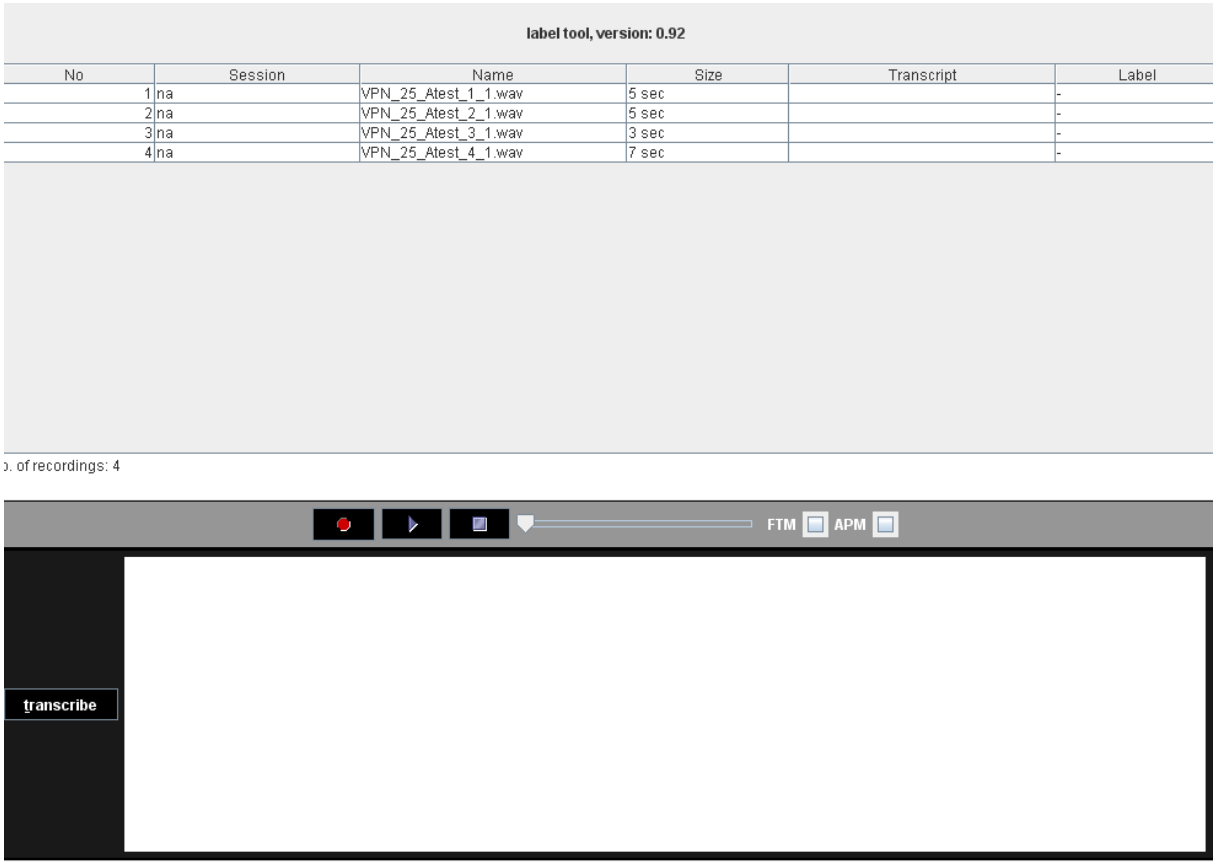
Fast mode

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:



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 textarea, followed by the <Control> key, the value is inserted, replacing the key..

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.