

EXTRA: User's Guide, Version 0.2

Martin Schickbichler

Creation date: September 19, 2010

1 ELAN-EXTRA: The ELAN extension

This section briefly explains the installation and usage of ELAN-EXTRA, an extension to the ELAN linguistic annotation software [WBR⁺06] for producing a draft phonetic transcription of single unknown audio samples.

1.1 Requirements

- The ELAN software version $\geq 3.9.0$ must be installed.
- Sufficient amount of free main memory (300MB should be enough for a larger example database).
- Sufficient amount of free hard disk space (35MB for the delivered example database).

1.2 Installation

1. Extract the downloaded `.zip` archive to the folder `extensions` in the ELAN directory. The directory structure should now look as follows:

```
ELAN_DIRECTORY
|-extensions
|  |-elan-extra
|    |-recognizer.cmdi
|    |-elan-extra.jar
|    ...
|-resources
|  |-sqlitedb.db
|  ...
```

2. Create a folder `resources` in the ELAN directory and copy the downloaded database file into it.
3. Increase the Java heap size when starting ELAN. This can be done as follows, depending on if you have installed the binary or the source version of ELAN.

Binary version: Open the `.lax` file in the ELAN directory (e.g. `ELAN.lax`). Search for the property `lax.nl.java.option.java.heap.size.max` and increase its value (e.g. to `320000000`). Save the file.

Source version: Open the ant build file `build.xml` in the ELAN directory. Search for the target `start` (the line starting with `<target name="start"...`). Add the line `<jvmarg value="-Xmx320M"/>` to the other `jvmarg` lines. Save the file.

4. You can now start ELAN.

1.3 Usage

To use the transcriber extension, open a new `wav` file in ELAN (via `File - New...`). Follow the instructions illustrated in figure 1 and commented in the following

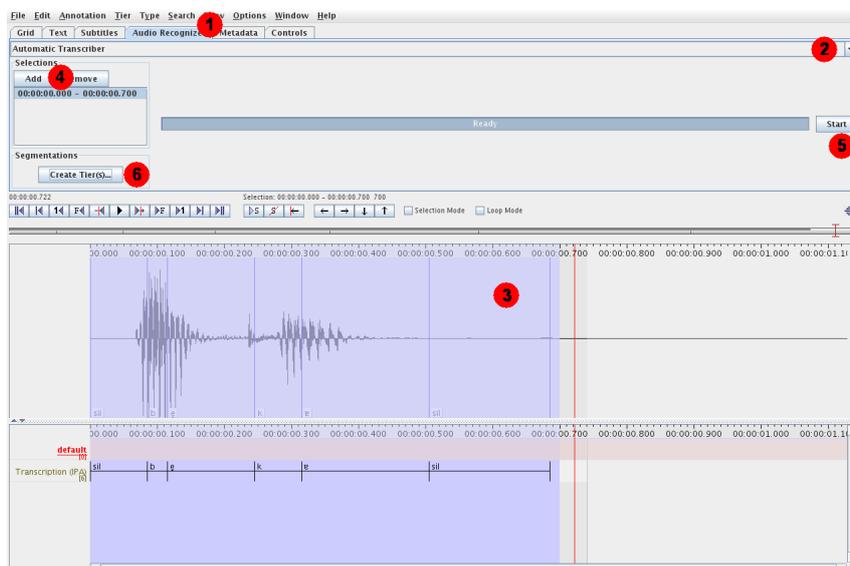


Figure 1: The ELAN Extension.

list:

- Go to the tab *Audio Recognizer* (Point 1).
- Select *ELAN-EXTRA* from the Drop-Down menu on the top (Point 2).
- Select the segment in the audio file to transcribe (Point 3).
- Click the *Add* button and choose your selection (Point 4).
- Click the *Start* button and wait until the transcription is done (Point 5).
- You can see the annotations in the waveform.
- Click *Create Tier(s)...* to create a tier with the new transcription (Point 6).

2 EXTRA: The standalone demonstrator for Example-based Automatic Phonetic Transcription

This section explains the usage of EXTRA for analysing the results of the Example-based Automatic Phonetic Transcription algorithm. EXTRA currently serves as a demonstrator and is not intended for professional use.

2.1 Requirements

- Sufficient amount of free main memory (300MB should be enough for a larger example database).
- Sufficient amount of free hard disk space (35MB for the delivered example database).

2.2 Installation

- Download the database file and examples and extract the zip file(s).
- Check if the database file is named `sqlitedb.db` and is placed in the EXTRA folder either directly or in a subdirectory named `resources`.
- You can now start EXTRA.

2.3 Usage

- Click *File – Open...* and select any number of the downloaded `.wav` files.
- Select the files to transcribe and press the *Transcribe* button.
- Wait until the transcription is done and select a result for further analysis from the results list (*Transcribed files* list box).
- To get the best matching examples for an individual frame, select this frame in the *Result frames* list. You can now see the n-best list in the right list box.
- To show the DTW warping path for a given example, select it in the right list box.

3 Creation of a segment database

The software currently requires the segment database to be in the sqlite format. Create an sqlite database file `sqlitedb.db` with the following database schema:

```
CREATE TABLE triphonedb (nr INTEGER PRIMARY KEY, id VARCHAR(255),
    int_transcription VARCHAR(255), transcription VARCHAR(255),
    features TEXT, phoneme_limits VARCHAR(32);
```

The individual fields have the following meanings:

nr: The primary key.

id: An identifier, in this case the filename of the audio.

int_transcription: An intermediate transcription that is reserved for future use and not necessary for this software.

transcription: The transcription.

features: The MFCCs as 2D-double array, converted to the java ObjectStream binary format. The first dimension represents the coefficients's number, the second dimension is the feature vector number.

phoneme_limits: The phoneme boundaries in multiples of 100ns units. The value represents the offset from the beginning of the current triphone segment.

1. Extract the MFCCs (HTK [YEG⁺06] style) from the audio files.
2. Now segment the audio data into triphones and label them (e. g. using the HTK).
3. Bring the labeled segments into a format that fits the database schema mentioned above, e.g.:

nr	id	int_.	transcr.	features	phoneme_limits
1	MADRID__01-03.mfc	-	sil m a	[MFCCs]	600000 1500000
2	MADRID__02-04.mfc	-	m a d	[MFCCs]	900000 2100000
3	MADRID__03-05.mfc	-	a d r	[MFCCs]	1200000 1700000
4	MADRID__04-06.mfc	-	d R i	[MFCCs]	500000 1200000
5	MADRID__05-07.mfc	-	R i t_h	[MFCCs]	700000 2600000
6	MADRID__06-08.mfc	-	i d sil	[MFCCs]	1900000 5700000

for the word *Madrid*, where [MFCCs] are the features as 2D-double array converted to the Java ObjectStream format.

4. Fill the database with entries from all your segments.
5. Copy the database to the `resources` folder in the ELAN directory.
6. The database is ready for use with the ELAN extension.

References

- [WBR⁺06] Peter Wittenburg, Hennie Brugman, Albert Russel, Alex Klassmann, and Han Sloetjes. ELAN: a professional framework for multimodality research. In *In Proceedings of Language Resources and Evaluation Conference (LREC)*, 2006.
- [YEG⁺06] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying (Andrew) Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. Cambridge University Engineering Department, 2006.