The *trs2txt* script: A conversion tool to produce time-aligned Toolbox files

Hiram Ring
hiram1 AT e D0T ntu D*T edu D@T sg

April 12, 2017

Abstract

This data paper gives some initial background and describes the function and use of a program bundled as a Python script and Windows executable file and intended to be used by Field Linguists. The program has been developed as a standalone solution inspired by Andrew Margetts' online converter. The purpose is to take a Transcriber file annotated with timecodes and convert it to a Toolbox document in order to preserve the timecodes, allowing playback of a linked sound file in a Toolbox project. *[UPDATES (12Apr2017): The configuration file is now deprecated in favor of asking for user input to create a config file. The repository link has also been updated and various bugs have been fixed to improve functionality on Windows.]*

1 Introduction

Field linguists and language documenters use a variety of software and database solutions for their research on under-described (minority) languages. Common programs include Praat,[1] ELAN,[2] Toolbox,[3] and FLEx,[4] among others. While these programs are supported, other programs such as Transcriber[5] are not. These tools are created for slightly different specialized uses, and as such have slightly different user groups. This means that a linguist will often use each program for slightly different purposes, as some do things that the others do not.

[1] http://www.fon.hum.uva.nl/praat/ (Boersma, 2001)

[2] http://tla.mpi.nl/tools/tla-tools/elan/ (Max Planck Institute for Psycholinguistics, The Language Archive, Nijmegen, The Netherlands; Wittenburg et al., 2006)

[3] http://www-01.sil.org/computing/toolbox/

[4] http://fieldworks.sil.org/flex/

[5] http://sourceforge.net/projects/trans/files/transcriber/1.5.1/

Praat, for example, is particularly useful for phoneticians and phonologists who are looking at speech sounds. The program makes sound visualization and measurement relatively simple and allows users to annotate speech sounds with TextGrids in order to run automated scripts. ELAN allows users to annotate audio or video with different tiers so that timecode information is preserved and notes can be made on conversational turns. Toolbox allows the linguist to analyze sentences and words morphologically in order to understand grammatical structure, and this is also the purview of FLEx, which has a more complex database system. Transcriber, on the other hand, is a simpler segmentation tool that aligns text to specific time-alignments on an audio file.

These tools also can have different outputs, though the majority support text files. Praat outputs tables of data or images, ELAN outputs a variety of file types including HTML, Toolbox outputs text in various formats, and FLEx outputs XML as well as a few other file types. Transcriber outputs a text-readable *.trs* file.

One of the desires of linguists doing language description is being able to time-align audio or video to text, analyze the grammatical structure of the text, and produce a document in which analyzed text occurs along with the linked media. ELAN is particularly good at producing time-aligned tiers and producing a web-ready HTML document with the linked media. However, the time-alignment is time-consuming, it can be difficult to set up tiers, and ELAN does not enable text analysis (that is, there are no linked database files, so the user must develop their own system of analysis within the tiers that ELAN allows).

Toolbox, on the other hand, is an excellent database system for text analysis but the user cannot time-align a media file from within the program, though Toolbox does support timecodes and playback of audio files. For analysis in Toolbox, having a linked audio file for a transcription is a useful tool. This means that the linguist does not need to leave their analysis in order to search for that particular sentence in their recordings. This ensures that the analysis, linked to an existing file, can be double-checked immediately and corrected, such as in places where the transcription was incorrect. FLEx is similarly an excellent database system for text analysis, but unfortunately does not easily support timecodes and playback of files.[6]

At the same time, ELAN does support importing from Toolbox text files, and can also export text files that Toolbox can read. This has allowed for the development of a workflow that uses both programs (along with Transcriber) to produce time-aligned analyzed texts that can be exported from ELAN as a web-ready HTML page.

[6]As a point of clarification, FLEx does seem to support playback of audio files but not timecoded segments of audio files. That is, individual segments corresponding to sentence entries must be cut from a larger sound file and linked to the sentence entries.

1.1 Description of workflow

This workflow seems to have originally been developed by Andrew Margetts as part of Shoebox[7] documentation[8] around 2005. The workflow that I describe here is somewhat modified, as it moves directly from Transcriber to Toolbox and then to ELAN as a final step. This is also not a tutorial or how-to, but simply a general description of the process as an introduction to the converter being described further below.

The basic steps of the workflow require an audio file that either needs to be transcribed or already has been transcribed. Ideally this will correspond to a video file of the exact same length. The audio file is opened in Transcriber[9] and the user transcribes text on the lines (or copy-pastes existing text), which correspond to timecodes in the audio file. Once the audio has been fully transcribed and timecoded the user saves the file. This file is then converted to a format that Toolbox can read, and is imported into a Toolbox project. Within Toolbox, the transcribed lines can be interlinearized and analyzed, and then the file is exported. The file is then imported into ELAN and the video file (if it exists) is re-linked to the analyzed document. The complete ELAN document is then exported as an HTML file which allows for playback of the linked video file along with corresponding text analysis.

1.2 Need for conversion

A crucial step for this workflow is the conversion from Transcriber format to Toolbox. For many linguists, Andrew Margetts' converter[10] will suffice, particularly as it allows for multiple configurations depending on the linguist's own database conventions. However, if the site goes down or a linguist is in the field with a bad or nonexistent internet connection, this becomes a problem. One solution is an offline converter which will perform much the same as the existing online converter. This is the motivation behind the *trs2txt* program which will be described in the remainder of this document.

2 The program

The program is currently hosted on GitHub,[11] a storage and collaboration website for coders, and will be for the foreseeable future. Interested coders are invited to collaborate

[7] Shoebox is an earlier version of what is now called Toolbox.

[8] See: http://languages-linguistics.unimelb.edu.au/thieberger/RNLD/IntroductionShoebox.pdf

[9] Transcriber version 1.5.2 was newly repackaged for Mac, and works well. A stable version for Windows is Transcriber 1.5.1, which can also be run in a Windows XP virtual box (see http://www.virtualbox.org/).

[10] Located at: http://linguisticsoftwareconverters.zong.mine.nu/

[11] At http://github.com/lingdoc/trs2txt/

on this project and make suggestions or edits to the code. The project site is built around the *trs2txt.py* script which is written in Python and serves as the primary code which runs in Python and does the conversion. The release version[12] includes a Windows binary (executable file) which is built with a basic Python library, so the user does not need to have Python installed. This release is distributable under a Creative Commons Attribution 4.0 International license[13] and can be freely copied, adapted, and shared. The following sections describe the contents of the zipped folder which can be downloaded at the release page, as well as explaining what each file is.

2.1 Contents of the .zip file

ELAN.typ (a database file that Toolbox requires to understand the markers)

README.txt (a readme file with brief explanations)

trs2txt.exe (the executable script file)

trs2txt.py (the Python script from which the executable is built)

2.2 Explanations of each file

ELAN.typ: This file is a database file that Toolbox uses in order to interpret the markers in the text file which the script generates. Each linguist may have a different setup and thus a slightly different **.typ* file that identifies how their Toolbox project interprets interlinear text files. This file is included in the zipped file as an example of a **.typ* file that is configured to allow markers which refer to a linked audio file and timecodes. This particular **.typ* file is also set up with markers that allow easy import into ELAN. This ELAN.typ file (or a corresponding version) should be placed in your Toolbox project settings folder.

README.txt: This file is a brief explanation of the contents of the folder and also contains simple usage instructions. It should be copied with any distribution and edited accordingly.

***trs2txt.cfg:** This file used to be included with the distribution, but is now deprecated, since it was difficult to format for cross-platform use. It is now replaced with a user prompt that requests configuration information from the user, and then the script generates a configuration file in the folder where the program is run if there is no existing configuration file.

[12] At http://github.com/lingdoc/trs2txt/releases

[13] http://creativecommons.org/licenses/by/4.0/

trs2txt.exe: This file is an executable program for Windows which was built directly from the Python script. It was compiled using PyInstaller[14] on a Windows XP virtual box and should run on any Windows platform.

trs2txt.py: This file is the Python script from which the executable is built, and can be run from the Python command line. It can also be easily edited, though this is not recommended for the average user, as editing may break the code. Since it is Python code, this file is cross-platform, working on any operating system with a Python build.

3 Steps to running the script

The script can be easily run by using either the Python script or the Windows executable. In order to run, the script/executable should be placed in a folder with all the **.trs* (Transcriber) files that the user wants to convert. The program will prompt the user for basic Toolbox settings/markers and create a configuration file in the same folder. The script then creates a corresponding **.txt* file for each **.trs* file in the folder.

3.1 The configuration file

The configuration file is created when the program is first run, or if there is no existing configuration file in the folder. The file containes definitions for the following markers (this configuration file is text-editable), and the markers should correspond to those that refer to the same fields in the user's own Toolbox texts:

\ref - This marker generally refers to a sentence, and often consists of a line name and number.

\ELANBegin - This marker identifies the timecode beginning, and is named to ease importing into ELAN.

\ELANEnd - This marker identifies the timecode ending, and is named to ease importing into ELAN.

\t - This marker identifies the vernacular text line. The Toolbox default is **\tx** but other linguists may use a different marker.

\f - This marker identifies the free translation line. The Toolbox default is **\ft** but other linguists may use a different marker.

[14] http://www.pyinstaller.org/

These markers should also correspond to the descriptions in the ELAN.typ file or the corresponding database file that Toolbox uses to interpret your database files.

3.2 Notes on running the script

As noted above, to use the script it should be run in a directory that contains Transcriber **.trs* files. The script/executable will create a corresponding **.txt* file for each **.trs* file in the directory, using the newly created (or existing) configuration file as a guide. Below are several things to note:

1. Existing **.txt* files with the same name will be overwritten, so it is recommended that the **.trs* files be placed in a separate folder than existing **.txt* files, to ensure nothing gets overwritten accidentally.

2. If markers are written in the configuration file which don't correspond to those in the **.typ* file in the Toolbox project settings folder, the resulting **.txt* files will not be viewable in Toolbox. This can be fixed by running a simple find/replace in a text editor for each marker.

3. Other markers and information that will be added by the script automatically are the linked sound file and participant turn information. Participant turn information will only be populated if such information is available in your Transcriber file. Otherwise this entry will be left blank - you can add this information manually in Toolbox. This information is used by ELAN to import Toolbox interlinear texts.

4. The script populates the **\id**, **\ref**, **\ELANParticipant** and **\sound** fields (as well as the timecode information) directly from the linked file in your Transcriber session. It is a good idea to save the Transcriber timecoded file in the same directory as your sound file in order to avoid having to find/replace more information in the resulting text file.

5. I do not know whether Toolbox supports sound formats other than *.WAV* - since it is generally bad practice in linguistics to record in formats other than *.wav*, I consider this a non-issue. In the case of non-wav formats I recommend converting the file to *.wav* before time-aligning in Transcriber.

6. This script treats all time-aligned text as belonging to the **\tx** (\t) line. However, it adds an **\ft** (\f) line to the resulting Toolbox *.txt* file to facilitate adding of this information. If you have a file with both vernacular text and free translation, you can leave it all on the same line in your Transcriber time-alignment, and simply copy-paste from the **\tx** line to the **\ft** once it is in Toolbox.

4 Viewing the output

Each **.txt* file can be opened directly in a text editor. The newly generated file will include blocks of text with Toolbox markers which identify timecode information, linked audio files annotated by the timecodes, and transcribed sentences corresponding to those timecodes. There is one additional step needed, however, before Toolbox can read the information.

4.1 Enabling Toolbox to read your information

In order for Toolbox to read the information, the header information must be correct. This is usually the first two or three lines of a Toolbox text file. The header information will look different depending on the linguist's particular setup. The current script/executable writes a basic possible header for each file. In order to verify the header information, the linguist should open an existing text in an existing Toolbox project. If the project is a new project, open the *Texts.txt* file within that project. Compare this existing text with the text just produced by the converter, and note the differences in the first two or three lines.

To import the texts into Toolbox, the texts need to be modified to look the same. This can be done by copying the header information from an existing project text into the converted text file. It can also be done by copying the converted content (from the header **\id** to the end) and pasting it into an existing Toolbox *Texts.txt* file. Then the text should be able to be opened from within a Toolbox project.

To play the sound file from your Toolbox project, the sound file should be in the same directory as the text(s), which should be located in the project directory. It may be possible to put the file in a separate folder within your projects directory, and change all the paths in the generated Toolbox file, though I have not tried this. The option to play a linked file or file segment can be found under the 'Tools' menu in an open Toolbox project.

4.2 Importing to ELAN

The **.txt* file resulting from the conversion process seems also to be in a format that can be imported to ELAN. This particular part of the workflow will not be dealt with here. A tutorial for this process should be available at the Margetts document identified above in a footnote. I hope to expand this aspect of the document at some future date.

5 Conclusion

This program supports an existing workflow by speeding up the conversion of files from Transcriber to a format that works with Toolbox. One of the benefits of linking files to an analysis is the ease with which it enables checking of an existing analysis. It is hoped that this script and executable file will be of some use to Field Linguists and others who desire time-aligned transcriptions of audio files that can be played directly from a Toolbox project.

References

Boersma, Paul. 2001. Praat, a system for doing phonetics by computer. *Glot International*, 5(9/10):341–345.

Wittenburg, P., H. Brugman, A. Russel, A. Klassmann, & H. Sloetjes. 2006. Elan: a professional framework for multimodality research. In *Proceedings of LREC 2006*. Fifth International Conference on Language Resources and Evaluation.