LANGUAGE DOCUMENTATION & CONSERVATION

Vol. 3, No. 1 (June 2009), pp. 51-86
http://nflrc.hawaii.edu/ldc/
http://hdl.handle.net/10125/4426

Using Toolbox with Media Files

Andrew Margetts

Monash University

This article focuses on our documentation project's use of Toolbox with media files, i.e., the source audio/video material that our transcripts are based on: why we set things up the way we do, and how. The process begins with an appropriate media file. This is marked up in Transcriber to produce a series of time-aligned annotations containing transcripts and speaker names, which correspond to intonation units in the recording. The resulting file is converted to a text format that can be used natively in Toolbox and easily imported into ELAN. The article also covers techniques for managing and querying the resulting data, both within Toolbox and with spreadsheets and relational databases. Further, it discusses some other language-oriented programs (especially Transcriber and ELAN) insofar as they affect our use of Toolbox. When Toolbox is used in close conjunction with source media files, it becomes particularly powerful. Some common tasks become easier, and new types of enquiry are possible. This is largely the result of Toolbox's ability to play discrete segments from a sound file. There is no single established methodology for creating such a conjunction, and there are a multitude of possibilities for using the results. This paper offers one account.

1. INTRODUCTION. This paper details some technical aspects of a documentation project of Saliba-Logea, an Oceanic language of Papua New Guinea. The project is part of the Documentation of Endangered Languages (DoBeS) Program funded by the Volkswagen Foundation.[1]

For several years, some users of Toolbox[2] have been recording time information alongside participant and transcription data in their text files, thus establishing a link between the transcription and the source audio via this timecode.

Such "text/media linking" has been achieved by using Toolbox in conjunction with other software tools that can "mark up" a media file, thus associating text with timecode "chunks" (each chunk defined by a start and an end time). The information is passed to Toolbox and incorporated in the standard Toolbox file format. Other information, e.g., free translations and interlinear glosses, is then added as usual.

Toolbox preserves the relationships among these different kinds of information, but until quite recently it hasn't provided the most obviously useful integration, i.e., the abil-

[1] For extensive and very helpful comments on, and contributions to, earlier drafts of this paper I would like to thank Nick Thieberger, Birgit Hellwig, Anna Margetts, Sebastian Drude, and Albert Bickford.

[2] http://www.sil.org/computing/toolbox

E-ISSN 1934-5275

ity to actually play the "soundbites" that the timecode represents. For that, one has had to use other programs—perhaps the same ones used to mark up the timecode (Transcriber[3] or ELAN[4]), or perhaps a dedicated tool like Audiamus.[5]

This situation changed with version 1.5 of Toolbox, which can now play these segments, provided that the timecode and audio file information is recorded in a particular way. For such a useful feature, this addition to Toolbox's armory received very little fanfare. Perhaps this was because the user-base that has timecode data and associated media files is still relatively small.

There is no single established methodology for creating the necessary connections between various software components. Section 2 of this article gives a sketch of one solution, one set of processes based on our own workflow, and outlines a few variants that might be more applicable to some users. Section 3 discusses some of the connections one can make between texts and lexica, with special reference to what can be done with the added audio information both in Toolbox and other frameworks.

Section 4 deals with two mechanisms within Toolbox for examining patterns in the data: concordances and filters, while section 5 outlines what can be achieved by extending the scope of the data beyond Toolbox using external mechanisms, namely spreadsheets and relational databases. Although such techniques are applicable to all kinds of data, the focus will continue to be on text/media linkage topics. Finally, section 6 offers a few suggestions for new features that would enhance media-related capabilities of Toolbox.

There are a number of appendices covering peripheral topics, or adding more detail to the main discussion. This is very far from an exhaustive treatment of the subject; not only is it incomplete but by the very nature of software it is likely to be out of date. I refer the reader to the final appendix for a brief overview of other useful online resources not introduced elsewhere in the text.

2. TYPICAL WORKFLOW. The prerequisite for our process is a suitable digital audio file, preferably in uncompressed pulse-code modulation (PCM) format, e.g., the most common type of "Wave" (.wav) file.

This is not to imply that video files are not appropriate; indeed in our own project nearly all recordings are digital, or digitized, video from which we extract an audio-only version (i.e., a ".wav" file). The point is that two of the programs we employ, Transcriber and Toolbox, can, at this stage, only handle audio files.[6] (Our video files are still useful, as we can later link them to the transcripts in ELAN.)

---

[3] http://trans.sourceforge.net/en/presentation.php

[4] ELAN (EUDICO Linguistic Annotator). http://www.lat-mpi.eu/tools/elan

[5] Audiamus. http://www.linguistics.unimelb.edu.au/thieberger/audiamusdemo.htm

[6] At least in the manner we need. Toolbox can play entire video files via an external player, but not discrete chunks. Video support in Transcriber has been announced in the "roadmap" but not yet implemented.

The workflow we then follow boils down to this sequence: transcribe in Transcriber; convert Transcriber file to Toolbox file with a dedicated online conversion tool; open and edit in Toolbox; use Toolbox files in ELAN (import file into ELAN, review and/or edit in ELAN, export again to Toolbox if necessary).

2.1 TRANSCRIBING IN TRANSCRIBER. The first thing to note is that there are (at least) two alternatives to using Transcriber: ELAN and CLAN.[7] Both can be used to achieve the same task and are arguably better in some cases: ELAN is the obvious choice for people who are subsequently going to work mainly in ELAN; likewise for CLAN. These two programs are different from Transcriber in that they are actually useful for certain kinds of linguistic analysis, whereas Transcriber is really just a vehicle for marking up the texts with timecode (it has other uses too, but these do not concern us here).

The second thing to note is that it has been announced that Transcriber is being completely redeveloped. According to the website the "new version will be based on Annotation Graph, which will be the default format of the new annotation files" and "should be released by Q2 09." At the time of writing it is fair to surmise that this change will temporarily break all existing converters, since the format is going to change. However it is likely that older versions of Transcriber will continue to be available for download and will continue to suffice for our needs. For the record, our online conversion tool[8] presented in the next section currently works with Transcriber up to version 1.5.1 (i.e., the most recent version).

So, why do we use Transcriber? Partly for historical reasons: when we started on this task, Transcriber was more stable than ELAN, and we had no experience with CLAN. Partly for convenience: Transcriber is much easier to learn, and hence to teach to transcription assistants. The stability issue no longer applies, but the usability one does, at least for us. We employ native-speaker transcribers who typically have no computer experience when they start; we have found they can learn Transcriber very quickly because it is so straightforward. The same does not apply to ELAN, which is so much more complex. Although we have not used CLAN, from looking at the files and talking to users I get the impression that it would not be quite as user-friendly as Transcriber. I hope that one day the ELAN team might develop a simplified version of ELAN, or rather a "skin" on top of ELAN to make it as simple to use for basic transcription as Transcriber, while still creating real ELAN files that could be worked on in ELAN or exported to Toolbox. That would remove Transcriber's advantage and the need to maintain it in one's system. Until that time we recommend Transcriber.

Notwithstanding our choice, you can instead use ELAN or CLAN. The caveat is that the next step, which is to convert Transcriber files directly to Toolbox files and for which we present a working solution, will not apply and you will need to find another method for converting your files to Toolbox format. I cover this topic briefly in Appendix A.

---

[7] CLAN (Computerized Language Analysis). http://childes.psy.cmu.edu/clan

[8] Linguistic Software Converters. http://linguisticsoftwareconverters.zong.mine.nu

There are a couple of things to be aware of in using Transcriber. First, events, sections, and backgrounds should not be used if you plan to use the online converter. In my view, they are not necessary for our purposes, which, at this stage, are purely to associate speech events with timecode. If you do use them, the converter we use will just ignore them: i.e., they will not be included in the converted file.[9] If you feel you do need some or all of these features, then you could consider using a different converter. See Appendix A.

Second, Transcriber is limited in how it can represent overlapping speech events in that it provides only one tier for assigning participants to speech units. That means that units with more than one speaker can only be rendered in the form "Speaker1 + Speaker2" rather than having different units (but with the same timecode) for "Speaker1" and "Speaker2." As a way to represent such occurrences in a database this is far from ideal, since if you do a search on, say,"Speaker1," it will not find the unit/s marked "Speaker1 + Speaker2" even though they are relevant. Worse, Transcriber's overlapping speech function limits you to two speakers at a time, presumably because it would become impossibly clumsy to allow more. In our experience, this is certainly not enough.

The good news is that both Toolbox and ELAN can, in their different ways, represent any number of participants for a single unit in a manageable, and searchable manner; one could say that they both have a multi-tier model. Some of the conversion tools available can make sense of Transcriber overlapping speech data and reformat it in the multi-tier manner, but they cannot help you overcome Transcriber's shortcomings regarding the number of participants. So if you need to mark more than two speakers for a unit you will have to resort to some kind of convention to create separate units for each speaker. For example, in Transcriber you might enter "{spkr1} Blah {spkr2} Blah blah {spkr3} More blah," and then after conversion you would copy, paste, and edit the corresponding record to create the three separate records with specific transcription data for each speaker (i.e., in Toolbox and/or a text editor). This process is much simpler than it sounds, but if the nature of your work means that you very often need to resort to this tactic, then you might be better off using ELAN as your "transcriber," since it can handle multi-tier speech events natively.

Generally we use Transcriber to mark up new audio files, i.e., ones that have not been worked on before, but another option is to import old transcripts and match them with their source audio files. A good way to do this is to start with a pre-existing Toolbox (or Shoebox) text, because then you already have a fair indication of where the "chunk" boundaries are. The tool for this is Econv;[10] see the next section.

As a convention, I suggest that your participant names in Transcriber do not contain spaces, e.g., that you use the form "Firstname_Lastname" rather than "Firstname Lastname." With a bit of effort you can configure Toolbox concordances to return the participant name as well as the reference number for each record. A limitation of this technique is that in some versions and configurations Toolbox stops returning data in such a field when it encounters a space (because it considers that it has reached the end of the data). In such

[9] More information on why they are not included can be found on the online converter website.

[10] The Language Archiving Technology Portal. http://www.lat-mpi.eu/tools/unsupported

cases only "Firstname" would be returned, limiting the completeness of the data; using a regular character, e.g., an underscore, instead of a space, circumvents this problem.[11]

Finally, it should be noted that Transcriber is not foolproof; errors do occasionally occur that cause the program to freeze or, more problematically, to produce a corrupted file. An error that I have come across is one where the timecode suddenly becomes non-sequential. This normally seems to occur only over a few units. It's not always easy to spot this in the Transcriber environment, but it shows up well enough when you try to convert the file. The solution is to look carefully at the file (if you cannot spot the errors in Transcriber, then open it in a text editor and search on some text near the affected area), then delete and redefine the offending chunks; you should not have to redo much, as the problem is localized and does not flow on to following chunks.

Although you may never need to go back to your Transcriber files once they have been converted, I suggest that you keep them safe somewhere (they are tiny). Apart from the desirability of having every kind of backup of source material, it is possible that you might use Transcriber again to help you to extract parts of audio files, e.g., a single chunk that you wish to use as an example, at a much later stage; this will be discussed in section 3.2.

2.2 CONVERTING TRANSCRIBER FILES TO TOOLBOX FILES. In previous times, converting Transcriber files to Toolbox format was quite a complicated affair. The first tool available was Econv, developed by the Technical Group[12] at the Max Planck Institute for Psycholinguistics in Nijmegen, Netherlands. This dedicated conversion program has been largely superseded by features in ELAN itself (developed by the same team) and by other converters, but, as mentioned, it is still useful for converting old Shoebox/Toolbox texts into draft Transcriber files that can be time-aligned to a sound file preparatory to converting them back to Toolbox format. A description of this process can be found in section 17 of Margetts (2005).[13]

ELAN is under continual development and its conversion features (there are many, but we are concerned here only with import from Transcriber and export to Toolbox) are much improved. However I have found that some post-processing of the converted files

---

[11] Toolbox can be configured to overcome this limitation by attending to the data property for the marker in question: right click on the marker to open the Marker Properties dialogue box, then Data Properties > Field data consists of > a single item (possibly consisting of multiple words). In many instances, Toolbox then treats the content as if the spaces were not significant. Moreover, since version 1.5.4 of Toolbox, the concordance can handle spaces in the first two reference layers. However there are still problems (e.g.,if there are spaces in the second layer, then sometimes the third layer may not be returned and instead the field content is replaced by "0"). Therefore I still recommend that spaces be avoided.

[12] Technical Group, Max Planck Institute for Psycholinguistics. http://www.mpi.nl/resources/tools

[13]In short, the Econv tool can prepare a Shoebox/Toolbox text for use in Transcriber in such a way that each record is assigned to a chunk. Each such chunk is an arbitrary one-second long. It is then just a matter of adjusting chunk boundaries to suit the audio; this is certainly quicker than starting from scratch. There are some tricks to getting it to work that are discussed in Margetts 2005.

involving search-and-replace operations is always necessary before we can use the results in Toolbox. See Margetts 2005 for an account of the problems and earlier work-around solutions to them.

Also, in ELAN the import/export utilities involve multiple dialog boxes, which makes conversion a slow (and potentially error-prone) affair. Furthermore its export process makes no provision for adding the marker that will allow Toolbox to play the sound chunks; it only writes the timecode information in the manner that will be used by ELAN when re-importing the Toolbox file.

In an attempt to simplify matters I developed a script for making conversions directly from Transcriber to Toolbox without the need for ELAN or any other processing step. I then adapted this script to run as a more generalized application so that others could use it. The result is our online converter. This is the tool I recommend for the job. There is extensive information available on the site. Two advantages of this tool are that you are not locked into one conversion scheme (i.e., you can choose your own output marker names), and that you can save your own settings so that it is simple and quick to use once you have set them up.[14]

Being modelled in part on the ELAN export process, the converter sets up markers for timecode and participant of the type that ELAN expects; this means that the resulting files can be imported into, and played in, ELAN. But it also adds a marker dedicated to playing the sound chunks in Toolbox—the "sound" marker. The format of this assumes that your sound file will be in the same folder as the related text file, and that it has the same name but with a ".wav" extension.

One other thing to note about the converter is the nature of the values provided for the reference marker; these take the form "Filename_Number" (e.g., Interview-A_0001, where the source file is Interview-A.trs and the converted file is Interview-A.txt). This convention of concatenating a reference to the text (the file name) with a record number ensures that each chunk is uniquely identified and that it can be easily associated—both by human and machine—with its source material.[15]

The converter site can also provide you with matching Toolbox "type" (.typ)[16] and ELAN "marker" (.mkr) files to assist you in opening the converted files in Toolbox and

[14] A criticism of the converter is that it is available only as an online service, which limits its usefulness in remote field sites. The decision to design it as an online application was driven by the desire to make it cross-platform and by the author's limited programming ability. A stand-alone version would certainly be preferable, and should this be developed, all registered users of the site will be notified by email.

[15] Note that it uses underscore characters rather than spaces between the file name and the number. This is for the same reasons that I advise against spaces in participant names: it helps avoid problems when making Toolbox concordances and also when linking the results with texts. To benefit from this, your file names should also not feature spaces.

[16] It does not however provide a Toolbox "language encoding" (.lng) file for your texts (i.e., for the transcription marker; the other markers can usually remain simply "Default"). These are best created in Toolbox if needed. However there is one situation where this lack might cause confusion

importing/exporting them to/from ELAN, respectively. I believe that if you are using Transcriber there is good reason to use this tool.

However there are at least two alternatives to this route: using ELAN's import/export utilities and using Toolbox's process for importing Transcriber files. We do not use these in our workflow, but I have provided an overview in Appendix A.

2.3 OPENING AND EDITING TOOLBOX FILES IN TOOLBOX. Assuming that you have followed the above procedures, you should have so-called Standard Format (SF) files that will open easily in Toolbox. Note, however, that any database file in Toolbox (e.g., a text or lexicon) belongs to a "type" that defines its properties. A database type is defined in a separate ".typ" file, usually with the same name as the type itself, which applies for all databases of the same kind. This file needs to be located in the same settings folder as your Toolbox project. Typically, this is in a subfolder, called Settings, of the folder where your database files are stored (i.e., when using the Start New Project command available with recent versions of Toolbox). However some users may have it elsewhere on their system in a folder called Shoebox Settings, My Shoebox Settings, My Toolbox Settings, or something similar (hereafter I simply refer to it as the "Settings" folder). It is possible to develop this type file in Toolbox itself, but instead I recommend downloading one from the converter site. As mentioned above, this type file will be appropriate for your Toolbox text files because it uses the same user settings to create its marker definitions, so it is a convenient way to get started. However, if you already have a working type file and have adapted your user settings to match, then you will not need this. Note that it is important that every text file, whether generated by some conversion mechanism or created with Toolbox itself, should be of the same database type as every other text (similarly for lexicons, concordances, or any other class of database).

The site-generated type files are set up to use the Reference marker (the default is "ref") as the "record marker." This is one of the options in Toolbox, the other being to use a marker called "id" as the record marker. The difference is that in our arrangement (using "ref") each unit is a primary record, while in the other (using "id"), the whole text is the record (or more accurately, each portion of it defined by an "id" marker), and the units are sub-records. We find the "ref" solution simpler; it fits in better with ELAN and has advantages for general Toolbox use. (The browse view can be confusing and the presentation of filtered results is often unhelpful when using "id"). If you decide to use "id" you will have to provide your own type file, and you should edit each Toolbox file that the converter generates to include the "id" marker/value pairs you need (typically adding one such "id" marker before the first "ref" marker).[17]

and frustration when using the converter. If a transcription made with Transcriber contains UTF-8 encoded characters beyond ASCII these may not display correctly after opening the converted file in Toolbox, depending upon the configuration of the relevant language encoding (probably Default unless specifically changed). To rectify this, adjust the encoding thus: Project > Language Encodings… > (select language type) > Modify > Options > Advanced > Unicode (UTF-8).

[17] It is possible, and potentially useful, to associate a text with both "id" and "ref" style type files (but

Whichever way you acquire your type file, you should consider calling it "ELANExport," i.e., the default for the converter. This is the name on the first line of the type file itself; the converter would name such a file "ELANExportText.typ," and these would be the two names you would see in Toolbox under Project > Database Types...; "ELANExport" will also appear in the first line of each text, and in this way texts are associated with the type file. The reason is that this is the name that ELAN inserts by default as part of the first line in the Toolbox file it generates when one uses the "Export as Shoebox File" command (not though for the newer "Export as Toolbox File," which for some reason requires explicit input).[18] Therefore using this name can save you a little work and trouble when you make future exports from ELAN (which you may want to do, e.g., after editing timecode information in ELAN).

If you are using a type file provided by the converter, then you already have the makings of a typical interlinear setup for your texts. That is to say, it contains the three standard processes under the interlinear tab (select a text file > Database > Properties... > Interlinear). These are: one parse process from the text to the analysis in morphemes; and two lookup processes from morpheme to gloss and part-of-speech. I say more on this in section 3.1.

Once you have a suitable type file in your Settings folder, you can open the converted texts in any Toolbox project associated with that folder (i.e., any project whose ".prj" file is in that folder).[19] You can search and manipulate them with filters and custom sort orders as usual, and you can play the related sound chunks within Toolbox with Tools > Play Sound.[20] If you need more information on the basics of Toolbox operation, please refer to Appendix B.

Below is an example record from a Toolbox text file produced by the online converter followed by an explanation of the contents.

---

not simultaneously). For example we find that certain filters only work with the "ref" setup but that the "id" system gives a more useful overview of a merged text comprising many component texts. Our technique is to maintain both kinds of ".typ" file in the Settings folder and simply edit the file names and the first line to suit which one we wish the texts to associate with. Make sure that Toolbox is not open during such a changeover.

[18] In ELAN the difference between "Export as Shoebox File" and "Export as Toolbox File" has more to do with whether you require UTF-8 Unicode character encoding support in Toolbox than whether you use Toolbox as opposed to Shoebox; we still use "Export as Shoebox File" even though we use Toolbox.

[19] It is possible to have many different Settings folders to suit different purposes, e.g., different languages.

[20] By contrast, Tools > Play File will invoke an external media player (and can therefore be used to examine video as well as audio files), but this will not play just the targeted chunk but rather the whole file. One of the main reasons to subsequently use ELAN is that it can play such discrete portions of a video file.

(1)

\ref	AboutDialects_01DP_0002
\ELANBegin	2.724
\ELANEnd	5.629
\ELANParticipant	John_M
\sound	AboutDialects_01DP.wav 2.724 5.629
\t	kowa ku hedehedede na kabo yau ya bu
\f	

The "ref" or reference marker (discussed above) is used as the unique label for this record. The following three markers are, as the prefix implies, used by ELAN when one imports the Toolbox file. The "Begin" and "End" timecodes are not useful in Toolbox, but the "Participant" is; it reflects the choices made in Transcriber for each so-called "turn" (a block of consecutive units attributed to one speaker). The "sound" marker is simply a rewriting of the timecode, along with the name of the relevant sound file in a manner that enables Toolbox to play the chunk. I give more detail on this in Appendix B in case the form is not clear. The "t" marker is the transcript itself, while the "f" marker is provided for adding a free translation.

Not shown, but significant, are the "m," "g," and "p" markers for morphemic analysis, gloss, and part of speech, respectively, which are the usual interlinear markers. These are defined in the Toolbox "type" (.typ) and ELAN "marker" (.mkr) files that the online converter can also provide. In the former they provide the skeleton of the interlinear setup mentioned above, while in the latter they allow any interlinearized text to be correctly imported into ELAN. (This marker file is designed to handle the hierarchical nature of the relationships between these markers and the "t" marker.)

These are the default markers produced by the converter site. All of them can be renamed to suit individual preference. But it is important to note that the "ELAN…" markers and perhaps the "ref" marker probably should not be changed without good reason. There is a further discussion of this matter on the site itself.

Interesting possibilities arise from the combination of this new audio function with more standard Toolbox commands. You can sort and filter the units in your text files (and hence the corresponding audio chunks) and so listen to, for example, just the utterances by a particular speaker, or all the occurrences of a particular construction. There is no real equivalent for this in other programs that I know of; ELAN can achieve something similar, but in my view it is not quite as elegant. This functionality can be extended when you run concordances or merge text files. In these cases your queries can include multiple audio files representing your entire text collection. (See section 4 for further discussion.)

2.4 USING TOOLBOX FILES IN ELAN. There are many reasons for wanting to import your Toolbox data into ELAN. For instance, ELAN can be used as a "graphical timecode editor" for making adjustments to the chunks. ELAN also offers video playback. Further, ELAN's visual representation of the sound-wave (similar to Transcriber's but more informative) allows one, for instance, to measure pause lengths on screen. But much more sophisticated

audio analysis is also possible using Praat[21] (a tool for "doing phonetics by computer"), which can be used with ELAN in a very convenient manner. Again, ELAN has different search capabilities from Toolbox. This is just a small sample of ELAN's capabilities.

A recent development (2009), which should be of significant interest to all researchers who wish to distribute their texts as multimedia documents to a general audience, is CuPED.[22] This is an independent initiative, which, as the full name "Customizable Presentation of ELAN Documents" implies, provides a way to transform ELAN files (and hence any Toolbox files that can be imported into ELAN) into "formats more readily accessible to a general audience." One strategy is to create web page versions that are enriched with Flash and JavaScript technology to produce a clean, intuitive presentation of the data, combining text annotations with streaming video/audio content that plays in a regular browser. Although this technique could be used to present a complex document, such as a fully interlinearized text, perhaps its main value lies at the other end of the spectrum: it can produce uncluttered, but annotated, renderings of individual recording sessions for the language community.

There are various methods for importing Toolbox files into ELAN. The one I have found most reliable requires a so-called "marker" file. This is the other type of settings-file that the converter site provides in addition to the converted texts. This file does not need to be placed in any particular location, but for consistency's sake I again place it in the Settings folder. So, to import Toolbox files into ELAN using this method, follow this procedure in ELAN: File > Import > Shoebox File... > (browse to choose your Toolbox file) > Set field markers... > Load Markers... > (browse to choose your Marker file) > Select > Close > OK.

If you are using ELAN only for inspection or presentation of data, you do not have to worry about exporting to Toolbox again; your Toolbox file is still your primary database. If, however, you edit the ELAN file and want to incorporate this information into your Toolbox file, then you must export it again. ELAN now offers two "Export As" processes for this: "Shoebox File" and "Toolbox File (UTF-8)," which are subtly different. As mentioned earlier in a footnote, the distinction has not to do with whether you use Toolbox or Shoebox; rather, it concerns whether you use Unicode in Toolbox. I still use the Shoebox File process, i.e. File > Export As > Shoebox File... > (select and order tiers, choose whether to wrap blocks, etc.) > Define field markers... > Load Markers... > (browse to choose your Marker file) > Select > Close > OK > (choose file name and location) > Save.

A note of caution: The conversions that ELAN makes when importing and exporting Toolbox texts are rather complicated, particularly when one is dealing with interlinearized versions. Although these days ELAN seems to do a faultless job, there have been a few cases in the past of data getting lost in transit. The problem is that if such errors occur, they are likely to be silent—i.e., no notification is given. Therefore I recommend that if you do export from ELAN back to Toolbox, you keep a backup of your original Toolbox file,

---

[21] http://www.praat.org

[22] CuPED (Customizable Presentation of ELAN Documents) http://sweet.artsrn.ualberta.ca/cdcox/cuped

and that you then compare the new and old files. You can do this within Toolbox: Tools > Compare Files. Alternatively you can use a dedicated file-comparison utility such as that within the program Total Commander,[23] or WinMerge[24] (which is open source and free) on the Windows platform or Text Wrangler[25] on the Mac.

3. LINKING TEXT AND LEXICON FILES IN TOOLBOX. The purpose of linking text and lexicon files (which are considered different database types in Toolbox) is generally to facilitate interlinear glossing of the texts. An additional benefit is that you can use your texts to inform and even populate a lexicon; Toolbox can automatically insert each new morpheme into a lexicon as it is encountered during the interlinear process. Adding audio information to the texts, the lexica, or both enhances the relationship between them.

A Toolbox lexicon database is an obvious source for a dictionary. Toolbox provides a standardized setup for lexicon databases called the "Multi-Dictionary Formatter (MDF) marker set." With this set you can produce conventional printed dictionaries and finder-lists from your lexicon; you can also use Lexique Pro,[26] another SIL product, to generate computer-based, and even online versions of these documents. Dictionary making is not the topic of this article, but there are many resources to help explain the MDF framework, starting with the Toolbox Help file. The point here is that we use this set, and so this discussion on linking will make reference to standard MDF markers.

This section continues with some comments about setting up the links between lexica and texts, and then goes on to cover incorporating image and sound files in a lexicon and related dictionaries.

3.1 SETTING UP THE LINKS. As I mentioned, the online converter tool can provide a type file with the skeleton of an interlinear setup in place. To complete this setup you must map the appropriate markers in the text files to their correspondents in your lexicon file(s). (The reason why this is not already done is that the converter deals only with your text files; it has no way of knowing how your lexica are defined). If you are using your own type file and do not already have an interlinear setup established, note that our framework is essentially the same as that provided by Toolbox's quick set up method: i.e., in Toolbox, select any text file > Database > Properties... > Interlinear > Quick Setup. Likewise the default interlinear setup for texts created by the Start New Project command is almost identical: open the default project (probably "Toolbox Project.prj" in the Settings folder), select the default text, then Database > Properties... > Interlinear to examine the individual settings for each process.

---

23 http://www.ghisler.com

24 http://winmerge.org

25 http://www.barebones.com/products/textwrangler

[26] http://www.lexiquepro.com

In our own practice we use the MDF marker set. Therefore we use "lx" for the lexeme, "se" for a subentry, "ge" for the English gloss, and "ps" for the part-of-speech, plus "u" for underlying and "a" for alternate forms. Our interlinear setup is then as follows:

(2) t > m Parse process: Markers to Find = lx, a, se; Marker to Output = u.
m > g Lookup process: Markers to Find = lx, se; Marker to Output = ge.
m > p Lookup process: Markers to Find = lx, se; Marker to Output = ps.

This setup is very like that used in the sample files provided in Margetts 2005 and that used in Toolbox's wizards. The one real difference is the addition of "se" to our current practice, which we find useful for allowing parses from sub-entries.

Note that you do not have to use the MDF set. If you prefer a different arrangement then just substitute your equivalents for "lx," "se," "ge," and "ps" in the setup. Also, "u" and "a" are not really part of the MDF set (and so they do not appear in a generated dictionary); rather, they are related to the parse processing.

You are not limited to having only one lexicon as a source for your interlinearizing, and it is often convenient to create more than one lexicon. For example, we have one each for two dialects of the language plus another called "glossing," in which we place items that should not be in the printed dictionary but which are necessary for glossing. All three of these lexicon files are linked to the interlinear process. For this to work, all lexicon files have to be of the same database type, defined by one single ".typ" file.

Almost as useful as the interlinear setup is the "Jump Path" setup. This feature allows you to define a connection between markers in the text and markers in the lexicon such that a right-click on part of an entry in the text (e.g., on a particular morpheme) will cause the cursor to jump to the relevant entry in your lexicon. The sample texts in the Toolbox wizard project include a simple jump path from text morphemes ("mb") to lexicon lexemes ("lx"). The sample files in Margetts 2005 also include a jump path. The type files produced by the online converter do not include a jump path for the same reason that they cannot complete the interlinear setup: the converter does not know to what you want to jump. You can edit or add new jump paths quite easily in Toolbox.

3.2 INCORPORATING IMAGE AND SOUND FILES IN A LEXICON. The ability to add image files to a lexicon has long been a feature of Toolbox. Although this might be useful for any Toolbox lexicon, the real benefit comes when using the MDF set, since it allows pictures to be added to the Dictionary when it is printed or turned into a Lexique Pro file. The marker to use then is "pc." All image files are incorporated using a simple "(path)filename.extension" format similar to the sound markers we have already considered: e.g. d:\images\picture_1.jpg (the path is not needed if the file is in the same folder). In Toolbox it is played with the F4 command, which invokes an external image viewer.

Sounds can also be added to a lexicon, just as for texts. Not surprisingly, given its origins as a technique to prepare printed material, the MDF set does not define any particular media markers. However the Lexique Pro team has added a couple of markers to allow sounds to be played in their dictionaries (though not as yet in those published online, i.e., as web-pages, as far as I understand). These are "sf" for "pronunciation of lexical entry"

and "sfx" for "corresponding example, paradigm, or lexical function." (Lexique Pro also provides several markers for including video files.) Good sources for "sfx" markers are the already existing "sound" markers in your texts. Creating "sf" markers is a different matter. If you feel you still need a dedicated sound file for the lexical item itself, then it is perhaps better to elicit such audio files directly. To illustrate the idea, say you have the following item in your lexicon:

(3)
```
\lx      buluka
\ps      n
\ge      pig
...
```

And the following record in one of your texts:

(4)
```
\ref                Buluhagalagala_01DU_0008
...
\ELANParticipant    speaker#1
\sound              Buluhagalagala_01DU.wav 12.066 14.835
\t                  buluka lakilakina ye miyamiya
\f                  There once was a big pig
...
```

Using that record as an example for the lexical item, you might get:

(5)
```
\lx      buluka
\ps      n
\ge      pig
...
\rf      Buluhagalagala_01DU_0008
\xv      buluka lakilakina ye miyamiya
\xe      There once was a big pig
...
```

To add the audio information from your text, copy and paste the "sound" marker entry, editing the path appropriately (assuming your lexicon is not in the same folder as your text and its supporting audio files). So, say you kept all your texts in a folder at d:\toolbox texts:

(6)
```
\lx         buluka
\ps         n
\ge      pig
...
\rf      Buluhagalagala_01DU_0008
\xv      buluka lakilakina ye miyamiya
\xe      There once was a big pig
\sound   d:\toolbox_texts\Buluhagalagala_01DU.wav 12.066 14.835
...
```

This will enable the sound to be played in the usual way in Toolbox, but from the lexicon entry rather than from a text record. The example above uses an absolute path to the file, but note that relative paths are also permissible.

As it stands, this is of no use to you in Lexique Pro, because that program cannot currently play just a segment of a file defined by start and end times as Toolbox can. However it is a good starting point, which defines exactly the information you wish to convey and from which you can take several paths (of increasing complexity): (1) "publish" your lexicon within Toolbox, (2) derive an equivalent stand-alone sound file that will play in Lexique Pro, (3) derive an equivalent "timecode aware" hyperlink that will play the sounds in a web version of your dictionary. I expand on each option in Appendix C.

4. EXAMINING PATTERNS IN THE DATA IN TOOLBOX. Searching for patterns in texts, particularly interlinearized ones, is one of the chief reasons for using a program like Toolbox or ELAN. Toolbox offers an array of search tools, but I shall concentrate on those designed for returning batches of records that match specific criteria, i.e., Concordances and Filters. These provide useful insights when it comes to searching for patterns or testing hypotheses. I am not aiming to give a complete guide to these commands, but just to demonstrate how the addition of markers for audio and participants to your texts can enhance such pattern investigations. ELAN takes a rather different approach to searching, which is discussed briefly in Appendix D.

4.1 CONCORDANCES. The secret to running a Toolbox concordance is to realize that one must first add a Text Corpus to search and that the meat of the task is in creating or editing the set-up of this corpus and adding files to it. There is a particular command for this: Project > Text Corpora… takes you to the Text Corpora dialogue box, then click Add. (Alternatively the same dialogue box can be reached via the concordance process itself: Tools > Concordance… > Edit > etc.)

As part of the process of adding (or modifying) a text corpus you must define the marker or markers that the concordance will examine and process. You must also define those that are to be used as references (up to three levels or "layers," as Toolbox calls them).

Using the same markers as elsewhere in this document I would place "\t" in the box for "Markers for words to Process." This means simply that the concordance will only look in the transcription field, which is useful for a corpus that contains texts that have not yet been interlinearized; for some kinds of concordances it will be more appropriate to search

your morpheme field (e.g., "\m") in which case you can place that in the box instead. There is nothing to stop you from placing more than one marker in the box, but generally that is unhelpful, since you get duplicated results. Again, following the marker set used before, insert "\ref" for "Primary (textual ref)"; leave the other two reference marker fields empty for now.[27] You must now add some files for the concordance to work on; click the Edit Files List... button for that.

Running a concordance is simply a matter of typing in your search word or phrase in the Search For box, reviewing a few other options,[28] and then clicking Lookup. For a basic concordance, the text(s) in the corpus do not even have to be open in Toolbox. However, for the next stage—gaining access to the audio chunks for the results—they do have to be open. We look at that next, as it subtly changes what you can do with a concordance.

Assuming that your concordance is set up as described, you should see a window called "lookup.db" with the results. The first column will be a representation of your texts' "ref" markers (called "concref" by default in the database type for lookup.db). Right-click on any item in that column. As long as the text file for that item is open in Toolbox, and as long as you have the reference in an appropriate form, preferably avoiding spaces,[29] then the cursor will jump straight to the correct record in the correct text file.

Generally there seems to be no need to explicitly set up a Jump Path between the concordance and the texts for this to work. If you find that it does not, you can always create a Jump Path to manage the connection; however you will then have to add each text in your corpus to the Databases in Path section.

If you have a sound marker in the record to which you have just jumped, then you can play the audio chunk as usual. This means that your concordances can concern themselves with more than just the written form; you can now use them to search for and review speech phenomena. A discussion of a search where this feature was vital is included in Anna Margetts (this issue).

If you are prepared to learn a little about regular expressions,[30] you can customize the concordance so that it includes participant names in the results, which eliminates the need to jump to the source records in the texts for this information. That means you can sort and filter concordances based on individual speakers' utterances. You can even go beyond

---

[27] Notice that in this dialogue box, i.e., Text Corpus Properties, you must add the backslash before a marker name in the various fields. This is not always the case in other dialogue boxes, e.g., Interlinear and Jump Path setup; Toolbox is a bit inconsistent in this regard.

[28] The choices for string matching are "Middle," "Begin," "End," or "Whole," and there are options regarding alignment, case matching, number of results, and name/location of output file.

[29] As discussed in section 2.2 the ref marker needs to be a unique label, such as one formed by the file name and a number so that Toolbox can make the jump without ambiguity, e.g., "Buluhagalagala_01DU_0008."

[30] Regular expressions are a way of abstracting aspects of textual information so that operations such as complex search and replace can be performed.

this to include other related data, such as a speaker's dialect, directly in the concordance. I outline the steps needed in Toolbox, which are very easy, and the kind of manipulation you will need to do externally, in Appendix E.

4.2 Filters. Concordances are very convenient, but not flexible enough when it comes to certain kinds of queries, e.g., those imposing conditions on more than one marker (e.g., where the morpheme is "ye" and the part-of-speech is conjunction), those using features such as logical operators (And, Or, Not, With), or built-in variables (Word boundary, Skip, Wildcard). For such cases a filter is the tool to use. Filters have a couple of drawbacks compared with concordances, however. For one thing they do not present the information in the same neat way with the search-term in the middle (they cannot, in fact, since some filters will be searching for several things simultaneously). This is not a serious problem: a well-configured Browse View will generally present the information in a useful manner. More seriously, a filter is run on only one file at a time. You can run the same filter on any file of the same database type simultaneously, but you must still inspect each file individually for the results.

The obvious, almost crude, way around this is to merge all your texts together and then filter that master file. This is feasible so long as you have a reference naming convention that produces unique labels for records. If all your records are merely numbered and rely on an "id" or even just the name of the file for complete identification, then this will not work since you will no longer have any idea to which text any particular record belongs. A way of creating a master file which requires no scripting and can be done in Toolbox itself is to use the "Merge Database..." command. Again, only databases of the same type can be merged.

The benefit of working with such a master file is that you then really have the best of both worlds. You can run a concordance on the master (the corpus being just this one file). And you can also run any kind of filter you like on this file. Whether running a concordance or a filter, you always have ready access to the audio data for a record, either by a jump or directly, and you always know that all records are included in the search. You must place your audio files in the same folder as the master file or else rewrite, using relative or absolute paths, all the sound marker entries in the source text files. This need only be a series of simple search-and-replace operations, one for each sound file.

Note that a concordance is just another type of database as far as Toolbox is concerned, so you can run a filter on a concordance. (An example is a filter on a particular speaker if you have created a concordance of the type developed in Appendix E.)

5. EXTENDING THE SCOPE OF THE DATA. All the tools of both Toolbox and ELAN, powerful though they are, eventually run into a limitation imposed by flat-file databases: the difficulty of extending the scope of an investigation beyond the data contained in the texts.

In theory one could encode any amount of information with each Toolbox (or ELAN) record, simply by adding more and more markers/tiers. In practice, this is not only unworkable (it simply takes too long to add and edit each record) but also undesirable; inevitably a lot of that information is actually implied by information already marked. Such information should not ideally be added to the record as well, since it introduces unlimited

opportunities for human error to compromise the integrity of the data. The classic case in point for our type of dataset is information implied by the participant names. This includes age, gender, dialect, domicile, education, and anything else you care to take note of about a person. Such information should be recorded just once rather than repeated in each unit or text.[31]

When any of these traits are significant for a linguistic investigation, then it becomes desirable to gain access to the information for them via the participant names. While the best way to do this is through a relational database framework, the easy way (at least to begin with) is with a spreadsheet. The next two sections discuss each approach. Appendix F uses an example from our project to illustrate both in more detail.

5.1 SPREADSHEET APPROACH. This approach amounts to exporting filtered information from Toolbox, reformatting and adding to it manually in a spreadsheet, and then sorting and summing using the inbuilt functions to generate answers. It is very flexible and delivered the results we needed with reasonable certainty.

However there are a few problems with it. First, it is time-consuming to run (though not to set up), particularly with large datasets. Second, it is not easily replicable, either by oneself (e.g., at a later stage, using more data), or by a different researcher. Third, it is suited only to answering the particular question it was designed for. The reason is the same in all cases; the extra information about age, dialect, etc., is not linked to the Toolbox information digitally and so cannot be systematically manipulated using an abstract mechanism like Structured Query Language (SQL). In short, there is too much manual labor involved.

5.2 RELATIONAL DATABASE APPROACH. A relational database comprising a table containing the Toolbox data along with tables of other, linked information solves these problems but brings a significant new one. It takes much longer to configure such a system initially than simply to cut and paste data into a spreadsheet. Ultimately the effort is worthwhile, even for a seemingly simple question, because not only can the query be run again at any time with any dataset by anyone, but also other, completely different queries can be run.

Creating such a database allows many new kinds of queries, not only because you have a richer data set (your text data and all related metadata is searchable together), but also because SQL itself is a very flexible and powerful tool. However, it is not quite a panacea. Some kinds of Toolbox filters are not easy to model in SQL. What if your question involves that kind of task as well as information about, say, speaker age or dialect? Rather than trying to replicate the Toolbox filter in your database, an easier option would be to run

---

[31] Despite this assertion, it is certainly possible and sometimes desirable to add such information directly to the Toolbox file (as hinted in section 4.1 regarding adding a speaker's dialect to a concordance). This has the major advantage of convenience, the user being already familiar with Toolbox functions. The way to manage this is with another script that looks up data in a separate file and then adds it automatically to the Toolbox file. This limits the possibilities for boredom and human error. See Appendix E.

the appropriate filter in Toolbox, export the filtered data, convert and import this to your database, maybe to a special table you reserve for such temporary tasks, then run an SQL query targeting the additional constraints. It sounds involved, but actually it is just a couple of logical steps; and there is no compromise as far as the results are concerned.

A potential drawback to importing your Toolbox data into another database is that you do not automatically have access to your sound chunks anymore. The way around that (which has other benefits as well) is to configure your database to be a "back-end" to a browsable interface. A common approach is to use MySQL[32] as the database server, Apache[33] as the web server, and PHP[34] as the "glue" language that manages the connections between them and between the clients (and also offers additional methods for querying your data that go beyond SQL). With these technologies one can configure a form-based website to query or write to a remote database. Therefore any computer with a browser and an internet connection can use the database. Apart from the common advantages of this approach (multiple simultaneous users, platform independence, and connectivity to other web sites), it also offers solutions to the problem of media playback and adds the potential to use video as well as audio.

There are a number of techniques that can be pressed into service here. Perhaps the simplest is to create Flash[35] audio/video clips for every possible chunk and call them with standard hyperlinks. These hyperlinks can be generated quite easily from the results of a query when it is rendered to a web page. A script or two and a command-line media encoder that can do batch processing can make the task of producing the clips relatively painless. Other possible solutions include using Synchronized Multimedia Integration Language (SMIL),[36] HTML+TIME[37] (Microsoft's proprietary version of SMIL, which works quite well in Internet Explorer), or the emerging Annodex[38] framework, all of which can, in principal, target portions of a media file based on timecode information (i.e., just as Toolbox and ELAN can) and play the result in a browser (in some cases via a plug-in).

6. SUGGESTIONS FOR NEW FEATURES IN TOOLBOX. The addition of timecode-related playback of audio has transformed Toolbox into an altogether more useful program. Nevertheless, there are always things that could be added or done differently to the benefit of

---

[32] http://www.mysql.com

[33] http://www.apache.org

[34] http://www.php.net

[35] http://www.adobe.com/products/flash

[36] http://www.w3.org/AudioVideo

[37] http://msdn.microsoft.com/en-us/library/ms533112(VS.85).aspx

[38] http://www.annodex.net

the user. Below are a few ideas for improvements which are related to the foregoing discussion. It is worth noting here that SIL,[39] the developers of Toolbox, are currently working on SIL FieldWorks one component of which, FieldWorks Language Explorer (FLEx),[40] is likely in time to supersede Toolbox (it can import Standard Format (SF) files, i.e., the format that Shoebox/Toolbox uses). Perhaps it is more likely then that if any of the following suggestions are implemented, they will form part of FLEx rather than Toolbox.

6.1 TRANSCRIBER FUNCTION. If the use of time-aligned texts becomes popular with Toolbox users, then it would be useful if Toolbox itself could provide the kind of functionality that Transcriber does. There would then be no more need for Transcriber, nor for converters and similar work-arounds.

6.2 SLOW PLAYBACK. The ability to slow the audio down when playing chunks could ease the task of fine-tuning the transcriptions, particularly for overlapping speech. If this were to be implemented, then Toolbox should ideally adopt a method that does not alter the pitch when changing the speed. (ELAN already offers variable speed playback, but at the cost of variable pitch, which limits its usefulness.) This function would be even more valuable if combined with the "Transcriber function" just described, as it could be of great help in the initial transcription phase.

6.3 VIDEO WINDOW. The option to display a small video window to show the images that go with the sound, where they exist, would add much to the user's comprehension of the data. If this were possible it would seem to make sense to support primarily the same codecs that ELAN does, at least MPEG-1.[41]

6.4 EXTENSIONS TO THE CONCORDANCE.The concordance tool would be even more useful if it allowed one to add other fields to the output in addition to the reference markers. For instance, if one could add the participant marker in its own right (rather than as a secondary reference marker), then one would not need an external script of the type mentioned in 4.1 and developed in Appendix E. Also, one could easily include the sound marker in the results, which would allow audio clips to be played directly from the concordance file.[42]

---

[39] http://www.sil.org

[40] http://www.sil.org/computing/fieldworks/flex

[41] Perhaps the developers could adapt the ELAN code (the sources being available for non-commercial use).

[42] It is now possible (since Toolbox 1.5.4) to add the sound marker using the secondary reference but seemingly at the expense of adding anything else, e.g., the participant marker, in the tertiary reference.

6.5 MERGING OR APPENDING TEXTS. Toolbox does have a Merge Database... command that could be used to create a master text of the type described earlier, but a nice refinement might be to allow the selection of a batch of texts to be merged. (If one follows the recommended "ref" marker convention of Filename_Number, there will never be any completely duplicated records and so "merging" is in effect "appending," which is what we want.) A useful location for this command would be in the dialogue box for adding and removing files for a concordance or word list text corpus (Tools > Concordance > Text Corpus/Edit... > Modify > Edit Files List...). This would be a logical source for the list of files to merge. This would be especially useful if you intend to establish more than one corpus.

Appendix A – Alternatives to the Online Converter

This appendix is supplementary to sections 2.1 (Transcribing in Transcriber) and 2.2 (Converting Transcriber Files to Toolbox Files). It deals with some situations where the online converter is not applicable, either because software other than Transcriber is being used for transcribing or simply because you prefer to work with conversion utilities native to ELAN and Toolbox, perhaps because you wish to preserve features in Transcriber such as events, sections, and backgrounds.

If you are using ELAN for marking up your audio files rather than Transcriber, then the online converter is not relevant and you should use ELAN's own export functions instead. Despite improvements made to ELAN in this respect, you will probably find that whichever way you use the dialogue boxes, the Toolbox file created by this process is still not quite suitable.

For instance, instead of having one marker for the actual transcription (which we call "t") the ELAN converter will generate a unique marker for each participant, despite the fact that this information is already marked in the dedicated ELANParticipant field. Changing all these markers to "t" (or whatever your preferred transcription marker is), which you must do for your interlinear setup to work normally, is just a matter of running search-and-replace operations. But it is tedious work, since you cannot easily predict what you are changing them from; it varies according to the speaker. And it is hard to write a useful macro to do this work automatically without resorting to regular expressions. If you are completely baffled about what kind of transformations to make, refer to Appendix Two of Margetts 2005. The exact method described there is no longer completely relevant due to changes to the ELAN export options, but the general principles are still valid, so it is a reasonable place to start.

Note that the ELAN export process does not give you the option to add a Toolbox-style sound marker. That means you cannot play the audio chunks in Toolbox. Perhaps this option will be added in due course, but in the meantime you will probably have to do without, unless you can write a suitable script to add it.

If you use CLAN rather than Transcriber it is worth knowing that ELAN can also import CHAT files (a file format used by CLAN). In theory then it ought to be possible to use ELAN as a converter from CLAN to Toolbox. In practice, as with other import/export options this seems to be not so straightforward. However, since all the file formats involved are plain text, the same kind of search-and-replace techniques should suffice.

These days ELAN can render events, sections, and backgrounds when importing Transcriber files, and that might seem like a reason to use ELAN as your converter rather than the online converter. However, these features generally do not export again to Toolbox in a useful manner, and relatively complex scripting would be needed to rectify this. If you really do want to include such additional metadata then you might want to use ELAN rather than Toolbox; in that case you could perhaps use ELAN instead of Transcriber in the first

place, simply creating equivalent tiers to sections and backgrounds (and anything else of significance to you) within ELAN.

A fairly recent (2008) addition to conversion options is a process within Toolbox itself for importing Transcriber files. This is achieved with a special "consistent changes" file[43] that is applied when opening the Transcriber file in Toolbox.

The process is easy to apply and works well. In theory this should be the ideal solution, avoiding all third-party applications. In practice there are several problems. For instance, concerning participant marking only the records where a turn occurs are marked with a participant. This more or less follows Transcriber's system but it has unfortunate implications for the Toolbox data; e.g., one cannot filter on one particular speaker effectively because for this to work, participants must be explicitly marked for all records. Also it seems to have the consequence of depriving files with only one speaker of any participant data at all (presumably because there are no new "turns" in the Transcriber file). Further, the converter does not deal well with overlapping speech but preserves the non-optimal Transcriber system. Finally the names themselves are not applied to the output file but rather their equivalent Transcriber codes (e.g., spk1, spk2 etc).

There are other problems, but the participant handling is the most serious, as some of the above conventions would not be easy to fix without using another script. Like the ELAN converter, the Toolbox process can import Transcriber sections; however as with participants they are marked only where the section changes, so this is of limited usefulness. As one would expect, this Toolbox process does add a sound marker (called "wav" by default; the marker name is not important, only the form of the data), but it does not add the ELAN equivalents so you cannot import the file into ELAN without further tweaking.

In conclusion, I suggest that you try the online converter first, since it is designed to produce data that are compatible with both Toolbox and ELAN, and has several other features, e.g., file names included in the record markers that could enhance the usefulness of the data. If for some reason this does not work for you, then try either the ELAN or Toolbox internal solutions.

[43] http://www.sil.org/computing/toolbox/extras.htm

Appendix B – Supplementary Information on Using Toolbox

This appendix provides a background to section 2.3 (Opening and Editing Toolbox Files in Toolbox) for those readers who are not so familiar with Toolbox.

There are many step-by-step resources available. Sections 1–16 of Margetts 2005 provide some information, and the accompanying sample files could be used as a template to get you going: add some lexical data of your own, delete the example entries, import one or two of your own texts (replace the ELANExportText.typ file in the Some Shoebox Settings folder in the sample set with a new one from the converter site), and play around.

If you are new to the program, another possibility is to use Toolbox's Start New Project wizard. This will establish a new project containing linked lexicon and text databases, complete with an interlinear setup. The text database provided by the wizard, Texts.txt (or rather, its "typ" file) is different from the ones the converter site provides in that it uses "id" as the record marker. Also the labels for the typical markers are slightly different (e.g., "tx" instead of "t"). However, it is a good example of a working setup and you can certainly use it as a starting point for working with your converted texts; just make sure to place a copy of the relevant type file (i.e., the one generated by the converter) in the same folder as the project (if you followed the defaults in the wizard this is likely to be C:\Toolbox\Settings).

One thing that is not covered in either Margetts 2005 or in the Toolbox wizard project, is what form a Toolbox "sound" marker should take nor how to play the sound. Even the current Toolbox Help files are a little vague about the format, stating merely that "if there are numbers after the file name, these are taken to be the beginning and end of the segment to play. The numbers are interpreted as seconds, with up to three digits after the decimal point." Here is an example of such a marker, as produced by the online converter, to help visualize this statement. Notice that the three components (file name, "begin" time, and "end" time) are separated simply by spaces.

(7)
```
\ref        AboutDialects_01DP_0002
…
\sound      AboutDialects_01DP.wav 2.724 5.629
...
```

There are three prerequisites for using this function as it is shown above: (1) the audio exists as a wave file and this file has a ".wav" extension, (2) it has the same name as the text file (which, as you can see, is also the prefix of the "ref" marker data), (3) it resides in the same folder as the text file. The first point follows current information in the Toolbox Help files (maybe other file types/ extensions also work, but I haven't experimented). The remaining two conditions are conventions that the online converter follows, so that it can ignore variability in file names and paths.[44] If you want to use different names or locations you can, but you will have to edit the marker entries yourself.

[44] Otherwise these would have to be explicitly declared in the online converter form, which seems unnecessarily complicated to me.

Once these three conditions have been met, playing the sound for a record is a simple matter of pressing Shift + F4. This works both in normal and browse views, which means you can easily skip around a file and play the parts which are of interest. Just be careful not to press Ctrl + F4 or Alt + F4 instead, which will promptly close the window or exit Toolbox respectively. (I wish Toolbox would come up with a different keyboard shortcut.)

Appendix C – More on Incorporating Sound Files in a Lexicon

This appendix expands on section 3.2 (Incorporating Image and Sound Files in a Lexicon). At the end of that section we had incorporated a "sound" marker into the lexicon, but couldn't do much with it outside of Toolbox. Assuming that the goal was to distribute the lexicon with playable audio information to a general audience, I suggested that there were three different options (of increasing complexity for the creator but simplicity for the user).

1) "Publish" your lexicon within Toolbox.

2) Derive an equivalent stand-alone sound file that will play in Lexique Pro.

3) Derive an equivalent "timecode aware" hyperlink that will play the sounds in a web version of your dictionary in a standard browser (possibly with an appropriate plug-in).

"Publishing" your Lexicon within Toolbox: Toolbox offers the option to lock your projects to different degrees. The most stringent level effectively turns Toolbox into a file viewer. See the Advanced Consultant Features in Toolbox Help. Note however that this locked project is not password protected but rather hidden from naive users only by being buried deep in the documentation. It is therefore not suitable for a truly read-only distribution. Also, you should not rely on filters in Toolbox to hide sensitive information from users; although this is effective in Toolbox itself, anyone can open the source files in a text editor and see the unexpurgated versions. In other words, you must actually remove anything you consider not suitable for public consumption. (By contrast, in Lexique Pro you can encrypt your files, which among other things, removes access to your "private fields.") However, locking a Toolbox project remains a straightforward way to share your work with colleagues, and if you include your sound files with your project, then you open up the opportunity for others to make comments based on the sources rather than just your interpretations. In this case you can leave all "sound" markers in their original form. This method does require your audience to install Toolbox and, at least to a limited degree, learn some of its commands.

The second method is to use stand-alone sound files in Lexique Pro. It is not hard to extract and save a portion of a sound file that is the equivalent of the part that would be targeted by your "sound" marker. The simplest way to do this, to my knowledge, is to use Transcriber. For this you need your original Transcriber file for the recording in question. Since Transcriber has no concept of record numbers, the easiest way to find the appropriate chunk is simply to search for the text string it contains. In the example developed in 3.2, this would be "buluka lakilakina ye miyamiya." Transcriber has a function for such procedures: Edit > Find/Replace. Once you have located the correct string select the corresponding audio chunk in the row at the bottom (i.e., as if selecting to play that chunk). Then: File > Save audio selection as.... This opens a Save audio segment dialogue box and supplies a useful default name based on the name of the audio file and the timecodes, though only

to two decimal places. I suggest you save this file to the same folder as the lexicon (so that you do not need a path in the marker format). Then add a "sfx" marker to the lexicon with this filename. To complete the example from 3.2:

(8)
```
\lx       buluka
\ps       n
\ge       pig
...
\rf       Buluhagalagala_01DU_0008
\xv       buluka lakilakina ye miyamiya
\xe       There once was a big pig
\sound    d:\toolbox_texts\Buluhagalagala_01DU.wav 12.066 14.835
\sfx      Buluhagalagala_01DU_12.07-14.84.wav
...
```

This should now work with Lexique Pro. You can leave in or delete the old "sound" marker as you see fit; Lexique Pro simply will not use it. One refinement would be to convert your ".wav" file to ".mp3" format to save space. These files are small anyway, so this may not be particularly important. But if we do this then our new marker will be:

(9)
```
\sfx      Buluhagalagala_01DU_12.07-14.84.mp3
```

Incidentally, ELAN can make similar extractions, but geared towards video files. It uses a third-party command-line tool called M2-edit CL which is a bit more involved than simply using Transcriber. This method again requires the user to install a new program, but Lexique Pro has been designed as a viewer so it makes fewer demands than Toolbox.

The third option is to use something like the emerging Annodex framework to playtimecode targeted clips from an audio or video file over the internet via a streaming server; in other words, a web equivalent for what Transcriber, Toolbox and ELAN can do. At the moment it is not really ready for mass production or consumption; it requires additional modules at the server (where your files would be stored), a plug-in at the client (i.e., the browser), and some specialized code in between. However the technology is proven, and in principle, your "sound" markers could be converted to the appropriate format to take advantage of it (they contain enough information already). A working example, using Lexique Pro to provide the online dictionary and Annodex to enable the associated media playback via hyperlinks, has been created by Thieberger (2007). Somewhat similar solutions could be developed with Flash or SMIL/HTML+TIME technologies.

Of these three options, it is the easiest for the user, since it requires only a web-browser. It is also currently the only one to be truly platform independent (because it is web based); both Toolbox and Lexique Pro are Windows applications (although there are ways to get around this with emulators), while Lexique Pro's web-export feature does not at present support the inclusion of sound files.

Appendix D – Searches in ELAN

In section 4 (Examining Patterns in the Data in Toolbox) I briefly noted that ELAN offers a different approach to searching for data. This appendix provides a little more detail.

ELAN has a number of different query tools collected under the menu-item Search. In some ways they combine some of the advantages of Toolbox concordances and filters: the Search Multiple tools can look inside closed documents, like a concordance, and also presents its findings in a concordance like manner, but with rather more information. The Structured Search Multiple eaf… is very powerful but I find it less intuitive than the Toolbox filter tools. Where ELAN does score is in any kind of query that looks at time constraints. If your research deals with such things, ELAN searches should be useful to you.

Searches in interlinearized text in ELAN can be problematic. The reason lies in the way ELAN represents the interlinear relationships. In Toolbox the relationships between interlinear components are maintained, both visually and actually, by the spacing system, which keeps related objects (such as morphemes and their gloss and part of speech) vertically aligned at their starting position.

The important point is that each transcription chunk remains as one unbroken string, one record. By contrast, ELAN maintains the relationships between tiers by placing each unit in its own sub-record, in a hierarchical system. Searches cannot generally search across two records, and this constitutes the problem. Say you have the following Toolbox record and want to search for instances of *lau magai*:

(10) \t se lau magai se keno

Toolbox searches would have no problem finding the target. If you import this record to ELAN as it stands, an ELAN search would also find the target, because the whole string (*se lau magai se keno*) would still be contained in one record. Once a text has been interlinearized in Toolbox, however, the record system after importing to ELAN becomes more particular. Now each word is in a sub-record of the whole unit (and each morpheme is in a sub-record of a word unit). So *lau* and *magai* are isolated from each other by a record boundary and the search will return nothing. Only search patterns consisting of just one word will succeed.

It is now possible to overcome this to some extent using ELAN's Multiple Layer Search tool, but it is not easy to do this for a specific string of more than three words or morphemes (i.e., representing three consecutive units). One way around this is to duplicate the transcription marker in Toolbox before interlinearizing. This duplicate marker would not be subject to the interlinearizing process in Toolbox and so would not be broken down in ELAN as described above, but will instead appear in ELAN as an unbroken string, similar to, say, the free-translation tier. ELAN can then search on this duplicate tier with success. The other way around this is to do such searches in Toolbox instead.

Appendix E – Adding Participant Names (and More) to Concordance Results

Section 4.1 introduced the idea that it is possible to wring more value out of a Toolbox concordance than might at first seem possible. This appendix attempts to explain the nature of this process. It starts with a fairly detailed explanation of how to add the participant names to the concordance results, and goes on to discuss more generally how the same scripting techniques could be adapted to add further data from external sources.

The task begins in Toolbox. The first step is to check the participant names format. In section 2.1 I advised that these names, like references, should not contain spaces but instead have the form "Firstname_Lastname." This is not strictly necessary, as discussed earlier, but I feel it is still good practice: Since in certain situations Toolbox stops reading data at the space, if you had spaces you would not retrieve the full name but just the first part.

Assuming that you have taken care of that, the next step is to go back to the definition of your concordance text corpus and add the participant marker (for us it is "\ELANParticipant") to the "Secondary (numeric ref)" reference marker box. Now run a concordance again. Your "concref" items will look something like this:

(11) \concref Buluhagalagala_01DU_0008.Marjorie_Little

In the concref item the reference and the participant are now concatenated with a period.

At this point you should save the concordance to a file so that you can run a script on it. (Use the Save As… command and choose a name other than the default "lookup.db"; the actual name is unimportant though the extension should ideally be the same as for other concordances, i.e., ".db"). Close this file in Toolbox.

The next step, breaking the reference and participant into two separate markers, takes place outside of Toolbox. This is actually very easy to do with a basic regular expression script. There are many tools for all platforms that can handle regular expressions. We use a text editor called NoteTab,[45] of which there is a perfectly capable free version (though unfortunately, it runs only on Windows). NoteTab has a built-in scripting language that is ideal for text manipulations, particularly as it supports Perl Compatible Regular Expressions (PCRE)—which are just about the industry standard. This is the script that we use to convert a text with the above format to something more useful:

(12)
```
1  ^!Jump 1
2  :Loop
3  ^!Find "\concref" S
4  ^!IfError Finish ELSE Next
5  ^!Replace "(\\concref\s.+)\.(.+)$" >> "$1\r\n\\concspkr $2" R
```

[45] http://www.notetab.com

```
6       ^!Goto Loop
7       :Finish
8       ^!Jump 1
```

Here is a brief explanation of each step, to give you an idea of what it does so you can adapt it if necessary:

1. moves the cursor to the start of the first line of the text

2. is a label called "Loop"

3. finds the next line in the text that contains "\concref"

4. says that if the code in the line above fails then the script should jump to the label called "Finish" (in other words, it exits the loop when it is at the end of the file)

5. does the work of reformatting the information. There are too many concepts here to go into detail, but basically it isolates the part up to the period, (\\concref\s.+) and the part after it up to the end of the line, (.+), "remembers" those parts (the parentheses are important), and then writes them out again on two lines, with a new marker called "concspkr" for the participant

6. causes the code to run again from "2," i.e., causes it to loop (until "4" causes it to exit)

7. is a label called "Finish"

8. moves the cursor to the start of the first line again (the script is at an end)

This script results in the sample line being re-rendered as:

(13)
```
\concref    Buluhagalagala_01DU_0008
\concspkr   Marjorie_Little
```

The file is then saved and re-opened in Toolbox. A new marker called "concspkr" will have been automatically added to the database type file for concordances (one of the convenient features of Toolbox). You now have the information on the participants in the concordance and can include this marker when filtering and sorting.[46]

[46] As mentioned earlier, unfortunately it is not possible to adapt this technique to also include the sound marker information in the concordance (which would mean that jumping to the texts to hear the audio would not be necessary). At present one can capture either the participant or the sound marker but not both.

Once you have had a little experience in writing simple scripts like the one above it is not that hard to create more ambitious and really useful routines. An example, which we have used to good effect, is to look up information pertaining to each speaker (e.g. age, dialect and gender) in an external file and add it to a concordance or a text file record by record.

The process in outline is as follows:

1. find first speaker in Toolbox text or concordance (i.e., find the data following the first "ELANParticipant" or "concspkr" marker)

2. store value as a variable "X"

3. open external file with speaker data (in our case a text export from a database; can be a simple text document however, provided it is regularly arranged)

4. find speaker name corresponding to "X" in this file.

5. find and store data related to this speaker in an array "Y" (an array is a group of variables)

6. return to Toolbox file. Find "X" again

7. write out values from "Y" in useful manner (i.e. with corresponding marker names)

8. find next speaker … loop through steps 2 through 8 until end of file

9. exit script

Naturally there is some fiddling involved, but it is not at all a complex task. With such a script you can almost instantaneously populate a Toolbox file with all the extra data that otherwise requires a spreadsheet or a separate relational database (see section 5 and Appendix F). By adding an initial step whereby all such existing data is removed, it is very easy to update this file at any time.

It would seem in some ways a superior approach to those techniques involving external spreadsheets and databases because everything can be done within Toolbox. However there are drawbacks as well. First, such a file must be deliberately updated to ensure accuracy. Granted this is not an arduous task (just run the script again) but it introduces an opportunity for error that a relational database does not have. Second, and more serious, it suffers from the same limitations as all Toolbox processes, e.g., it is neither suited to running numerical queries on, say the speaker ages, nor to calculate totals and percentages and the like. Still, this technique will carry one a long way without the need for anything more than a simple text document of supporting information (this document could very readily be another Toolbox file of a different database type; i.e., a Standard Format file would do the job well).

Appendix F – Example of Using Spreadsheets and Relational Databases

Section 5 (Extending the Scope of the Data) discussed in general terms the benefits and drawbacks of exporting Toolbox data to other program types, namely spreadsheets and relational databases. This appendix is an account of our practical experience in these endeavors. It is based on our attempts to answer a particular linguistic question.

The idea was to see whether a certain grammatical feature in the language was a trait of a particular age group, say of speakers under forty, whether it was found in both dialects of the language, and with what frequency (i.e., with respect to the entire set) it occurred.

Investigating these questions with a spreadsheet required us first to run a Toolbox concordance on all available texts looking for the feature. We sorted and filtered the results by the participant names and thereby derived the total number of occurrences of the feature for each participant. We then set up a spreadsheet containing just this information, i.e., a list of participants with occurrences. This resulted, schematically, in something like the following:

(14)

Participant	Occurrences
SpeakerA	88
SpeakerB	25
SpeakerC	121
SpeakerD	13
...	...
Total	X

The next step was to make judgments for each participant about age and dialect and add these to the spreadsheet, resulting in:

(15)

Participant	Occurrences	Over/Under 40	Dialect
SpeakerA	88	U	L
SpeakerB	25	O	L
SpeakerC	121	U	S
SpeakerD	13	U	L
...	...	...	...
Total	X		

We made several copies of the sheet so we could rearrange the data in various ways, sorting and totalling by age, by dialect and by both age and dialect. Here's what the first one (i.e., by age) would look like:

(16)

Participant	Occurrences	Under 40
SpeakerA	88	U
SpeakerC	121	U
SpeakerD	13	U

...	...	...
SubTotal: Under 40	Y	
Participant	Occurrences	Over 40
SpeakerB	25	O
...	...	...
SubTotal: Over 40	Z	
Total	X	

The interesting data were the subtotals, or rather, their values as percentages of the grand total of occurrences (i.e., their relative spread according to our criteria), and also as percentages of the grand total of available intonation units (i.e., the frequency of occurrences in the texts). We also had to check the distribution of all units (i.e., unfiltered) with respect to our criteria (i.e., age and dialect distribution) so that we could assess the significance of the results.

This question could arguably be better tested using a relational database. I do not attempt to give a worked example of setting up an appropriate system; that would belong in a different article. Rather, I give a sketch of how we have gone about the task, with some particular attention given to the problems of getting our information out of Toolbox and into another database. This will give you an idea whether this is something you want to try yourself.

The first thing to point out is that we are not actually linking a Toolbox text or set of texts to another database: the organization of data in a Toolbox text file is not suited to that. Instead we convert the files to a different format, and then import these conversions into another database that already contains the extra information. The links between types of information among the different tables of the database are made internally in the relational database. This is perhaps not ideal, as in effect we are making a copy of our Toolbox data rather than linking directly to it. This means we have to be mindful of the need to update that copy—i.e., to import afresh from Toolbox from time to time, just as we have to recheck our interlinearizing periodically. In fact it becomes part of the same workflow: check texts against lexicon, export to database.

In our case we started with a Microsoft Office Access[47] database, which was set up initially just to track the progress of our own workflow: how many tapes had been processed, what sessions belonged to each tape, who and what are the participants, locations, topics in each session, and so on. One common factor between this database and our Toolbox texts was the list of participants (another was the list of session names).

If I were starting this task over, I am not sure that I would still choose Access, because of all the things that Access is not: not multi-user (except in a limited sense), not open-source, not free. However it must be said that it is fairly powerful and quite easy to learn to use and, most important, it is capable of importing information in XML format. Initially,

[47] http://office.microsoft.com/en-us/access/FX100487571033.aspx

this provided the missing link, because Toolbox can export to XML; XML became the corridor between the two programs.

It is work to maintain this kind of system, if only because of all the fiddling around with multiple export and import dialogue boxes in the two programs. One can minimize the task by adopting the earlier suggestion of a master text in Toolbox. This gives you just one file to export, one to import, and one set of data to replace (one complete table) in your new database.

The default XML export routine in Toolbox was a bit too enthusiastic for our purposes, in that it represents, in a hierarchy of tags, all the hierarchy of your interlinear text. This is not what we wanted, because then one needs to import to many different tables in Access, not just one, and then somehow establish links between them. I solved this problem by deleting the tags that create the hierarchy. Importing this into a single table in Access was quite straightforward, and I was able to link the text records to the participants' details via their name. As a proof of concept it succeeded.

How does this linking work in practice? In this case, two tables (one called "texts," the other "participants") include a field called "name" containing the participants' names. Access has a graphical way of modelling queries, but underlying it is a variant of SQL, the language that underpins most relational databases. An SQL query can ask a question that hinges on correspondences among tables by writing a sort of equation. For example, a query asking for all fields, for all records, from both tables, linked by the name and sorted (i.e., ordered) by the names in ascending order might look like this:

(17) SELECT * FROM texts, participants WHERE texts.name = participants.name ORDER BY participants.name ASC;

Restricting this set to, e.g., records where the morpheme *liga* is spoken by participants speaking the Logea dialect, the query might look like this:

(18) SELECT * FROM texts, participants WHERE texts.name = participants.name AND transcription LIKE '%liga%' AND dialect = 'Logea' ORDER BY participants.name ASC;

As you can see SQL is a relatively natural language to read and write, though as with any code, syntax is inflexible. There are differences between SQL "dialects" according to the program, but they are minor.

As I became more aware of shortcomings in Access, particularly with regard to multi-user and web access, I considered shifting to another database system. I settled on MySQL, for which there is abundant support; it is one of the most popular open-source (and free) relational databases around. This brought many benefits, but it raised a new challenge, because there is no obvious way to import XML documents into MySQL.

Although there are various solutions to this problem, I decided that the best approach would be to convert the texts into a format that is supported and even expected by MySQL. This format is the very common "Comma Separated Values (CSV)" format. This is an extremely simple system where each line contains one record and each field is separated from the next by a comma, which acts as a delimiter (another label for such a file is Comma

Delimited). You may have encountered it in spreadsheet programs, as it is one of the ways to move data (but not formulas) between different programs. It might have already occurred to you that problems would arise if the data themselves contain commas, triggering false field boundaries. There are a couple of ways around this problem. One is to use other symbols, typically inverted commas to surround text strings so that isolated commas used as genuine punctuation are not confused with the delimiters. The other is to use a different delimiter; a common alternative is the tab character.

Whatever exact configuration you settle on, the important thing to realize regarding the CSV format is that the obverse of its simplicity is a very strict demand for order and consistency. The format depends entirely on equivalent items being in exactly the same position, as dictated by the sequence of delimiters, on their respective lines. For example, a participant name might always occur after, say, the fourth comma from the start of a line.

That creates a problem when trying to convert from Toolbox files to CSV, because Toolbox has quite a relaxed approach to data encoding within each record. This manifests itself in two ways. First, data can be in a relatively random order, e.g., a "free translation" marker might come before or after a "notes" marker. Second, a field or fields can be missing or duplicated, e.g., no "free translation" marker and two "notes" markers. Such variations seldom affect how Toolbox behaves (and there are no restrictions on how you add markers), so they tend to creep into the files; but they will wreak havoc on your new data set if they are allowed to stand uncorrected in your CSV files. You will end up with data in all the wrong fields in your texts table in the relational database.

So, any converter has to be able to check for inconsistencies of this kind and either deal with them automatically (e.g., concatenating duplicated markers or providing an empty but valid marker where one does not exist) or alert the user so that they can be manually corrected. It also has to perform a peculiar task related to Toolbox's "Reshape" function. If you haven't encountered this before, "Reshape" wraps whole chunks of interlinearized text so that it is more readable on screen or on the page, but so that the vertical alignment of morphemes, gloss, and part of speech is preserved. It is a clever and effective process. The problem is that it is not completely reversible within Toolbox, and to the converter it appears that you have duplicated fields. Concatenating such fields is not a problem, but recreating the original vertical alignment is more of a challenge. It is not impossible, but it is a bit tricky.

In case the reader is motivated to make a similar converter, I would say that the most useful approach is to proceed record by record, and within each record to assign each marker/value pair to a temporary variable. That makes it easier to re-order, concatenate, and omit parts of the data when you write it out to the finished file.

Appendix G – Other Useful Online Resources

This appendix is inevitably incomplete, but it may help to fill out some of the gaps in the narrative.

As mentioned in the introduction, our project is part of the Documentation of Endangered Languages (DoBeS) Program[48] funded by the Volkswagen Foundation.[49] As well as providing the funding, the electronic archive, the infrastructure, and an ethical, academic, and practical framework for our field work, this program has produced and continues to produce a wealth of software tools[50] and technical literature.[51]

The Linguistic Data Consortium is a resource for all "language-related education, research and technology development," and its Linguistic Annotation page[52] provides a useful overview and source of links to relevant institutions, programs, and standards.

The E-MELD (Electronic Metastructure for Endangered Languages Data)[53] site is concerned with the dual tasks of preservation of endangered languages and development of effective infrastructure for electronic archives of such languages. The following E-MELD pages are particularly recommended:

The School of Best Practices[54]

Events, which includes proceedings of the workshops[55]

"Digitizing and Annotating Texts and Field Recordings in the Awetí Project" by S. Drude.[56] This document describes step-by-step the workflow from recording to archiving of language data, including technical matters and discussions of using Shoebox, Transcriber, and ELAN together. As such, it is a useful complement to the topics in this article, and a source of information on much more besides, e.g., advanced glossing of interlinear texts.

---

[48] http://www.mpi.nl/DOBES

[49] http://www.volkswagenstiftung.de

[50] http://www.lat-mpi.eu

[51] http://www.mpi.nl/DOBES/documents

[52] http://www.ldc.upenn.edu/annotation

[53] http://www.emeld.org/index.cfm

[54] http://www.emeld.org/school

[55] http://www.emeld.org/events/index.cfm

[56] http://www.emeld.org/workshop/2003/paper-drude.html

REFERENCES

MARGETTS, ANDREW. 2005. Introduction to Shoebox and Toolbox with notes on Econv, Transcriber and Elan. http://www.linguistics.unimelb.edu.au/thieberger/RNLD/IntroductionShoebox.pdf (sample files: http://www.linguistics.unimelb.edu.au/RNLD/Shoebox_Session.zip).

MARGETTS, ANNA. 2009. Data processing and its impact on linguistic analysis. *Language Documentation and Conservation* 3(1): 87-99.

THIEBERGER, NICK. 2007. Getting media into online dictionaries, an example workflow. http://wiki.arts.unimelb.edu.au/ethnoer/Main_Page#Getting_media_into_online_dictionaries.2C_an_example_workflow

Andrew Margetts
andrew.margetts@arts.monash.edu.au