Extending ELAN into Variationist Sociolinguistics

(updated July 2017 from 2015 Linguistics Vanguard version)

Naomi Nagy
Naomi.Nagy@utoronto.ca

Miriam Meyerhoff
Miriam.Meyerhoff@vuw.ac.nz

Abstract: Prior to the implementation of ELAN (tla.mpi.nl/tools/tla-tools/elan, Wittenburg et al. 2006), it was common for sociolinguists to use multiple software applications, and consequently multiple formats, along the route from recording participants to conducting statistical analyses of the data. We present a method which allows for transcription, extracting, coding, preparation for statistical analysis, calculation of some basic frequency statistics, and creation of a concordance all within one program. ELAN is well-established as a valuable tool for language documentation. ELAN is frequently used for transcription and multi-tier mark-up illustrating levels of linguistic structure as well as translations and glosses. We hope that this cross-over introduction will encourage the efficiency of documentary linguistics among sociolinguists and increase the interest in documenting variation among documentarians.

After providing an overview of ELAN's utility, we focus on extracting (or marking) and coding tokens of linguistic variables for quantitative analysis in the variationist sociolinguistic framework. This seamless connection between recording, transcript and coding of dependent and independent variables improves consistency, efficiency, utility, reliability and the accountability of our coding to the original recording. We illustrate a range of benefits and include step-by-step instructions accompanied by downloadable sample files to illustrate each step of the process (http://~~projects.chass.utoronto.ca/ngn~~/zip/Celeste_for_ELAN.zip).

ngn.artsci.utoronto.ca/

(and the same on 6 other appearances of this website in the document)

Table of Contents

1. Introduction

This paper motivates using ELAN (http://tla.mpi.nl/tools/tla-tools/elan, Wittenburg et al. 2006) to assist in the analysis of sociolinguistic variables.[1] ELAN has established itself as a valuable tool for language documentation and is frequently used for transcription and multi-tier mark-up illustrating levels of linguistic structure as well as translations and glosses. In this paper, we illustrate an extension to its utility: extracting and coding tokens of linguistic variables for quantitative analysis in the variationist sociolinguistic framework. This approach improves consistency, reliability and validity of our coding through direct links to the original recording. We illustrate the following benefits:

- seamless connections between recording, transcript, and coding of the dependent variable (response) and independent variables (predictors). This facilitates (i) revision and intercoder reliability testing and (ii) the workflow across different stages, providing better tracking across a collaborative research team.
- reuse of contextual factor coding (e.g. style, topic, interlocutor) as well as some structural (morphological, syntactic) tags
- wide-ranging exportability (e.g. Excel, R, Rbrul, Goldvarb, Praat, SPSS)
- importability of transcripts from Word/text files and many other transcription formats (e.g. text, Transcriber, Shoebox/Toolbox, CHAT, Praat)
- complex searches to speed up token extraction
- archivability of all mark-up related to each data file in a consistent manner that uses a small file-size format

2. Why ELAN?

Prior to the implementation of ELAN, it was common for sociolinguists to use up to four different software applications along the route from recording participants to conducting statistical analyses of the data. For example, we might use a dedicated transcribing program such as Transcriber, a specialized application such as Shoebox or Chat for CHILDES (http://childes.psy.cmu.edu/, MacWhinney 2000), or create transcriptions with a text editor or word processor (e.g. Microsoft Word). Transcriptions might include manually entered

[1] We appreciate the questions, feedback and encouragement from participants at the 3rd International Conference on Language Documentation and Conservation, NWAV 42, NWAV 43 and assorted workshops that helped develop this paper. Zsuzsanna Fagyal, Miklós Kontra, and Richard Cameron have been particularly helpful on all three counts. Will Barras and Han Sloetjes have shared invaluable knowledge about ELAN. We take full responsibility for existing errors, but ask readers to let us know when they find them. As our use of ELAN is constantly evolving, updates to our approach may be found at: http://projects.chass.utoronto.ca/ngn/HLVC/2_2_linguists.php.

timestamps, marked while playing back the recording for transcription. Some people transcribed directly in Praat, which provides some of the functionalities of ELAN.

The transcriptions would then be mined for data and the data exported to a spreadsheet or database program, bringing along some identification information such as page/line numbers or timestamps. Much context would be lost at this stage – the researcher had to decide how much text to copy to provide context for each token. If additional factors became relevant later in the analysis, the context brought along with each token was often not sufficiently rich to address the new questions, and this necessitated laborious re-sourcing of the original token and context.

Each extracted token had to be marked up for the speaker who produced it – this information did not transfer automatically from the transcription file to the token coding file. A common practice was to use separate files for each speaker or create a separate column in which the source is identified. The former makes an unwieldy collection of files and the latter introduces opportunities for additional error.

Coding of the dependent variable and independent linguistic and stylistic variables would then be conducted in additional columns of the spreadsheet. Often additional searches of the transcription file would be necessary to find necessary information for coding decisions. It was not always easy to make sure the correct token had been re-located.

After this, the coded data was converted into a format suitable for use in a statistical analysis package (commonly Goldvarb (http://individual.utoronto.ca/tagliamonte/goldvarb.html), a concordancer, etc. One such method is illustrated at http://individual.utoronto.ca/ngn/LIN/courses/LIN351/LIN351_project_GEs.htm#part6. At this point, only tenuous connections to the original recording or transcript existed – if an error was suspected or more information about a token was required, it was difficult and time-consuming to locate the original site of a token in the recording or transcript.

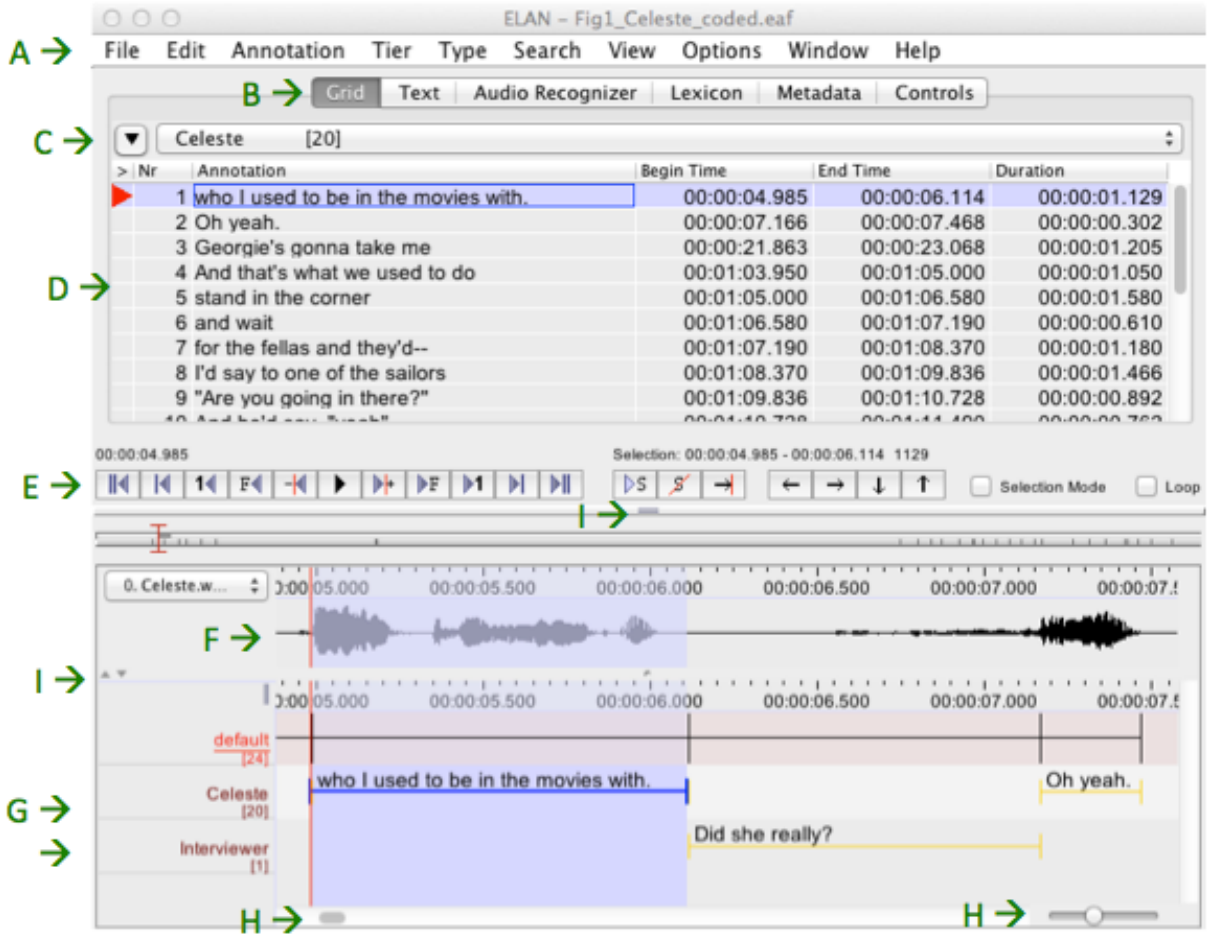
This article outlines an alternative method which allows for transcription, extracting, coding, and preparation of data for statistical analysis and calculation of some basic frequency statistics.

The article is structured in three main parts: an introduction to the basic structure of ELAN, a thumbnail overview of the use of ELAN for coding variation, a detailed step-by-step guide to the use of ELAN for coding variation, including links to prepared ELAN files that allow you to practice steps in the detailed guide. Appendices provide instructions for value-added features allowed by ELAN: setting importing legacy (.doc, .txt) files into ELAN (Appendix D); creating a concordance and lexical frequency information (Appendix E).

3. Basic structure and utility of ELAN

In this section, we introduce ELAN's basic structure and utility. This will allow us to demonstrate its usefulness for coding variation more clearly in the sections that follow.

Figure 1 is a screenshot showing one section of a time-aligned transcription. In this figure, the following sections are indicated, highlighting the way that the text transcription and the audio (or video) recordings are linked:



A: Menus at the top of the file.

B: Choice of displays for the top part of the window, with the "Grid" display selected.

C: Pull-down menu that allows you to select different tiers for display in the top part of the window.

D: A "grid" view of the transcript, in which the transcription for one segment is shown on each row, along with the timestamp. ELAN calls any text that is linked to a chunk of the media file an "annotation".
This is a clickable means of navigating through the file.

E: Playback and selection controls.

F: Soundwave with time-markers.

G: Two tiers providing transcription of two participants in a conversation (number of annotations shown under label).

H: Scrollbar to move through the recording and transcript and slider to change resolution.

I: Several up/down arrow pairs that allow for resizing the different parts of the display. Font size, color and style are also modifiable.

Figure 1: ELAN file with transcription. Letters indicate parts of the display referred to in text

There are a number of benefits to transcribing and coding data with ELAN, as opposed to the multi-media method described at the beginning of this section. These include, first and foremost, ELAN's **hierarchical structure.** This permits time-aligned linking of information, across many types of data. These include speaker transcription(s), translations, glosses, coding of dependent and independent variables, and style or topic coding that extends over longer portions of the transcript. This structure has the following benefits:

Context availability: You can see all the context you need, and hear it as you code each factor. This provides greater accuracy and cuts down on errors. When it is necessary to revise a coding plan and change the codes for some tokens, it is easy to find the token(s), edit, and quickly recreate any data analysis files.

Recyclability: The hierarchical structure of ELAN allows us to code multiple variables in a single file, even multiple dependent variables. This means that coding produced for the analysis of one variable is available for future work. If a transcript has been coded according to style or conversational topic, for example, those codes can be applied to the analysis of many variables – their timestamps can be used to match them with any coded tokens. Additionally, tier structure and format, controlled vocabulary, and codes can be transferred from one file to another, meaning that the reusability extends beyond a single file or project.

Searchability: ELAN's search function is very powerful because it uses regular expressions.[2] You can simultaneously search multiple files (e.g. your whole corpus), look for correspondences between items on different tiers and within different time periods relative to each other.

For example, you might want to find all tokens of an affricated /t/ that occur within five seconds of a non-standard for of negation, or you might want to only extract examples of null subjects occurring in narrative segments.

Everything linked to an annotation (for our purposes, all the coding associated with it that we will outline shortly) and its timestamp appear in the search results. The list of matches is live – you can click directly on the part of the transcript where each token is located.

Compatibility: In addition to being able to move seamlessly between ELAN and Praat, ELAN has many Import and Export functions allowing interoperability with other programs. Of key importance for variationist analysis, ELAN exports in tab-delimited and comma-delimited text file format. These are suitable for multivariate analysis using Rbrul or Goldvarb or virtually any statistical analysis package. Files can be exported in many formats (ELAN manual §4.4.1, accessed 12 September 2014).

It is best to refer to the ELAN manual as these options are updated often. The following formats are likely to be most useful for sociolinguists.

- Tab-delimited text file
- Interlinear text file

---

[2] If you are not familiar with the concept of searching with regular expressions, there are many online and book guides. Regular expressions are very powerful tools for finding patterns, as opposed to specific text strings, and take us well beyond the scope of this article.

- Traditional transcript file
- Praat TextGrid file

These are formats that can be used with forced alignment software such as FAVE (Rosenfeld et al. 2011), WebMAUS (Kisler et al. 2012), or ProsodyLab (Gorman et al. 2011) (WebMAUS is integrated into the latest versions of ELAN). The ELAN manual also lists the file formats that can be imported into ELAN. These include:

- CSV / Tab-delimited Text Files
- Praat TextGrid file

Over and above these attributes associated with its hierarchical structure, we find the following characteristics helpful:

Special fonts: ELAN allows for input and display of transcriptions in many orthographies (English, Arabic, Chinese, Georgian, Hebrew, Korean, Russian or Turkish). In the example in Figure 1, the IPA is used because Faetar is a language without a standard orthography. Several input methods are available: typing directly, optionally using a pop-up keyboard on the screen where you can see the special characters (IPA-SAMPA), and pop-up menus that provide choices of characters that physically resemble what you've typed (IPA-RTR).

Acoustic analysis with Praat: Praat (Boersma & Weenink 2014) and ELAN can interact with each other. As well as importing and exporting between these applications, it is possible to call up spectrograms, measure formants, etc., from within ELAN, benefitting from the power of Praat. If you want to examine parts of your file with Praat from inside ELAN, see Appendix E: Installing and using Sendpraat.

Automation of analysis: Time-aligned transcription and mark-up allow data to be subjected to automated forced alignment (at the segmental level) and thus to automated acoustic analysis (e.g. VOT measurements, formant extraction). We predict that using tools for detailed phonetic analysis such as FAVE (Rosenfelder et al. 2011) and WebMAUS (Kisler et al. 2012) should encourage advances similar in kind to those that resulted from the advent of automated tree-parsers for syntactic analysis.

Audio control: ELAN has a good built-in algorithm for changing the speed of audio playback without altering the pitch. This allows you to listen in faster than real time when you are segmenting very clear speech, if you are just searching for a topic, or trying to get a general sense of the content. Conversely, you can listen in slower than real time when you need to attend to detail. There is a "Loop mode," which will repeat the playback of a particular segment until you are ready to move to the next segment.

ELAN may also be used with **video recordings,** which has made it invaluable for analysis of variation in sign languages (British Sign Language Corpus Project 2012).

Creating audio (or video) clips with automated transcriptions for internet display: Chris Cox and Andrea Berez have created an application called CuPED (Customizable Presentation of ELAN Documents, Appendix A) that makes webpages that display voiceclips, along with a

(multi-tiered) transcription that scrolls in synchrony with the recording. The soundclips associated with the map at http://projects.chass.utoronto.ca/ngn/HLVC/4_1_map.php were easily created with this application, from files originally transcribed in ELAN.

4. An overview of the use of ELAN for coding variation

In this section, we give an overview of how ELAN works, focussing on elements essential to our practices as sociolinguists. In the next section, we provide step-by-step instructions to get started in ELAN and prepare data for variationist analysis.

ELAN is freeware that you can download from http://tla.mpi.nl/tools/tla-tools/elan/download/. It's available for a variety of platforms including Macintosh, Windows and Linux. It's easy to install and currently well-supported by various European institutes. There is a comprehensive manual for ELAN which you can search online and/or download. You'll find it at the same URL. There are many tutorials available online to explore additional uses of ELAN.

ELAN has three modes that are useful for the process of transcribing and coding sociolinguistic data. These are *segmentation, annotation* and *transcription*.

The first stage in creating a transcription is to **segment** the audio file. This consists of splitting up the .wav file into units. You can choose whatever units you prefer: sentences, clauses, breath groups, turns – they are all possible. In *Options: Segmentation Mode*, you listen to the soundwave (and/or watch the video), clicking the Return key as you hear each "boundary." This creates segments, and these will eventually contain the transcription associated with each unit.

To start transcribing you will need to go into *Annotation Mode* to create a dedicated space, or Tier, for each participant in the recording. Once your tiers are set up, you can transcribe in the convenient *Transcription Mode*.

Once you have transcribed (a portion of) your audio recording, you can begin isolating and **coding** your variable data. For this, you will create additional tiers, normally one for each dependent and independent variable to be considered. In *Annotation Mode* you can **prepare the types** of tiers, **the tiers** themselves, and (optionally) **controlled vocabularies.**

In the tier for each variable, you will create annotations that indicate the variant realizing each token. These are time-aligned to the corresponding portion of the transcription and recording.

When you have completed coding your variables, you will **export** the transcription annotations, timestamps and token codes to create a file suitable to use as input for a statistical analysis program such as Goldvarb, Rbrul or R.

Other things that are useful to learn to do (by exploring ELAN's manual) are listed in Appendix B. Technical details about the software are in Appendix C.

5. Step by step guide (workflow) to the use of ELAN for coding variation

This section provides detailed, step by step instructions to accomplish what has just been outlined. In addition to ELAN, you will need a spreadsheet editor such as Microsoft Excel and a statistical analysis program (e.g. Goldvarb, Rbrul, R). Headphones are also helpful.

Getting ELAN going on your computer

Starting ELAN

1. Download ELAN (http://tla.mpi.nl/tools/tla-tools/elan/download/) and save it in Applications or Programs.

2. Start ELAN.

3. Choose "New" from the "File" menu.

4. You will be asked to *Add [a] Media File*. In our example, choose "Celeste_Step4.wav".[3]

5. Immediately save your .eaf file (.eaf = ELAN annotation file). We created Celeste.eaf.

6. By default, ELAN opens in *Annotation Mode*. If you don't see a soundwave in the center of the window, Right/Command (⌘)-Click) where the soundwave should be (see Figure 1, it will appear as a horizontal line) and then choose a (large) number from the pop-up menu to Vertical Zoom in by. If it is still too quiet to hear, use the "Amplify" function in an audio editor like Audacity to edit the .wav. Then relink the modified .wav to your ELAN .eaf.

7. Once you can see the soundwave, test the volume. Check your computer's System Preferences or ◂ icon, as well as the Volume controls in ELAN. You find those by clicking Controls in the area marked "B" of the window displayed in Figure 1.

Segmentation

8. Select *Segmentation Mode* in the *Options* menu (*Options: Segmentation Mode*). In this mode, divide the recording into phrase/clause units. Press the Play button (in "E" of Figure 1). Press Return/Enter (the designated Segmentation key) as you hear the beginning of the first segment. Press Return again as you hear the end of each segment. The end of one annotation is set up as the default start of the following one.[4] If you make

---

[3] Files for the Celeste example are available at http://projects.chass.utoronto.ca/ngn/pdf/Celeste_LD&C.zip.

[4] There are several options for relating key presses to boundaries. You may wish to experiment with the other options: (1) press and release Return/Enter to begin an annotation, repeat to end the annotation or (2) press and hold Return/Enter for the duration of each annotation.

an error, pause, back up as necessary, and start again. New annotations overwrite existing ones. Boundaries can be adjusted, inserted and deleted later (in Annotation Mode). You decide how exact you need your alignment to be – you can zoom in on the soundwave to see additional detail (Slider button in "H" in Figure 1, bottom right). Once your file is segmented, you can start to think about transcribing. (See Celeste_segmented_Step8.eaf.)

Creating a hierarchical structure of tiers for transcription

9. Prior to creating the tiers where you will transcribe each speaker, you need to define the type of relationship the transcription tiers will have with the default tier (on which we segmented the file).[5] In the *Type* menu, choose *Add New Tier Type*. For ease of reference, we'll call our first new type "Transcription." As its Stereotype, select "Symbolic Association."

10. Create a tier in ELAN for each person to be transcribed in that file (one tier named "Celeste," in our example). To do this, go to *Tier: Add New Tier*. Specify the Default tier as its Parent Tier and select the Transcription type you created as its Tier Type. (You can add more tiers any time.)

Transcribing

11. To **transcribe** the content in each annotation field that you segmented, go to *Options: Transcription Mode*. In the dialogue box that appears when you select Transcription Mode, select "Transcription" as the type for each column until you have one column for each speaker in the file.[6] (You can return to this dialogue by pressing the "Configure…" button.)

12. In *Transcription Mode*, each column displays the annotation fields for one speaker. A soundwave corresponding to any selected field is displayed. Play a segment by clicking in its annotation field. Transcribe what you hear. When you finish, hit Return. This advances the recording to the next segment and moves the cursor so you are ready to type in the next annotation field. (See Celeste_transcribed_Step12.eaf.)

    You may choose the Loop mode if you want each segment to repeat automatically until you are done transcribing it.

---

[5] Documentary linguists tend not to transcribe in the default tier. If you automatically number each segment in the default tier (*Tier: Label and Number Annotations*), you have a concise and unique reference for any examples you might want to reproduce in a journal article or presentation later.

[6] If the necessary number of columns don't appear, select *Create Annotations on Dependent Tiers…*" from the tier menu. In the first pop-up menu, select "default" (the tier on which you segmented) as the parent. In the second pop-up menu, select each of the speakers you will transcribe ("Celeste" in our example). Click "Finish."

13. You can select how many columns to make visible (depending on whether you want to transcribe one speaker at a time or several) with the *Configure* button.

14. If you have an existing .doc or .txt transcription file, you can copy it in to this format, segment by segment. Appendix D provides an alternative method of converting a legacy transcript to ELAN format.

A few other useful things to know about ELAN annotations

- You can add an unlimited number of annotations and tiers to audio or video streams. You can hide them when you aren't actively working on this.
- An annotation can be a sentence, word or gloss, a comment, translation or a description of any feature observed in the media.
- An annotation can either be time-aligned to the media or it can refer to other existing annotations.
- ELAN provides several different views of the annotations (*Grid, Text, Subtitles*) in Annotation Mode. Each view is connected and synchronized to the audio/video media. Choose views from the part of the display indicated as "B" in Figure 1.

File preparation for variable rule analysis

Once a file is transcribed in ELAN, it is easy to "extract" and code tokens. Here, "extract" appears in double quotes because it's no longer necessary to extract or remove the tokens from their original context – the principal benefit of this approach. "Mark" might now be a better word.

At this point, we move from transcribing to the practices we have developed for preparing data for variationist analysis.

As indicated in Section 4, we code each variable on a separate tier, where the dependent variable is linked to the annotation segment that contains it. We then create a hierarchical structure that links codes for independent (predictor) variables to the code of the corresponding dependent (response) variable. You can see this architecture in the inset of Figure 2. As an example, we code the well-known sociolinguistic variable (ING), the alternation between velar and alveolar nasals in the English *–ing* suffix (cf. Hazen 2006). Figure 2 shows a screenshot of the sample file Celeste_coded_Step31.eaf, a transcript in which coding for (ING) is ongoing. Below the tier for the transcription of the speaker we have added four tiers:

- token: the tier showing the words containing tokens of the dependent variable;

- (ING): the tier on which the dependent variable is coded – abbreviations or codes indicate which variant (alveolar, velar, other) appears in each token;

- POS: this tier shows a code for the first independent variable considered, the part of speech of the word containing (ING);

- # syllables: this tier shows a code for the second independent variable, the number of syllables in the token word.

We will now walk you through the process of setting up this architecture for a dependent variable and its predictor variables.

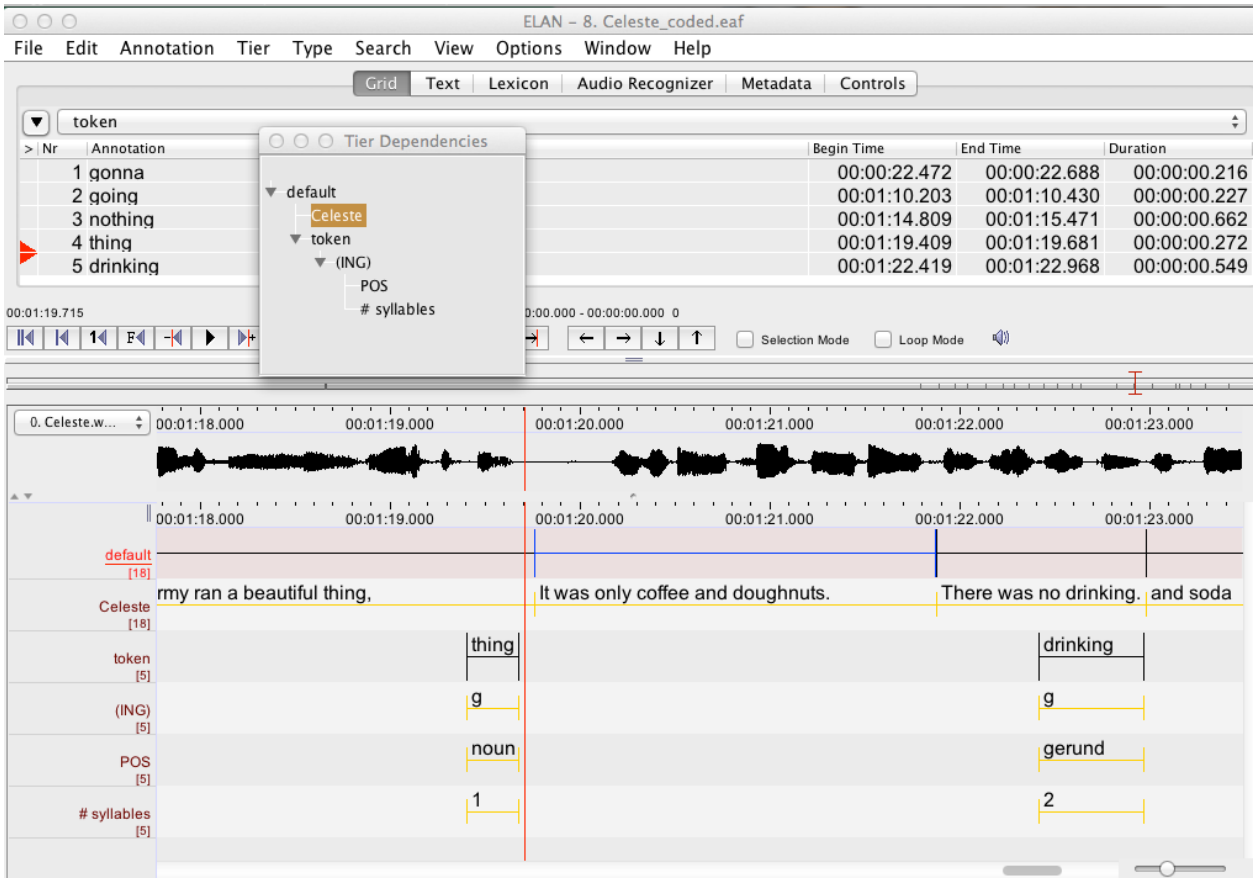


Figure 2: Coding & Extracting: One tier for each variable (at Step 31)

Creating a hierarchical structure of tiers for coding variables

The dendogram showing tier dependencies in Figure 2 is produced by selecting *View: Tier Dependencies*. It shows the hierarchical relationship among all tiers in the file. We first created a tier called "token," as child of "Celeste." We then created a child of the token tier for the dependent variable (ING), and one grandchild tier for each independent variable (POS, # syllables). Because these are linked to the original context in the transcript, if you double-click on the code of any variable you can immediately read and hear the surrounding context.

15. Prior to creating the tiers for coding each variable, we need to **define the types** of tiers – i.e. specify how they link to their parent tiers. This is fundamentally the same process we went through in Steps 9-10. Celeste_with_tiers_for_coding_Step25.eaf shows the complete set of tiers we are creating.

    The first tier we need is for the tokens we've identified. This will be a child of the transcription tier, the tier named "Celeste." We want to be able to have more than one token within a single transcription segment, e.g. if the segment were "she was working in the morning." In the *Type* menu, choose *Add New Tier Type*. We'll call this first new type

"token." For its *Stereotype*, select *Included In*, because this allows us to create multiple token annotations that are shorter than the parent annotation if we need to.[7]
Having a separate tier that annotates tokens can be useful if you want to investigate lexical effects.

16. Next we will want to create a tier for coding the dependent variable in each token. Your dependent variable probably has a finite set of forms (in our example, the alveolar nasal, the velar nasal and for some dialects of English an "Other" variant). We can save ourselves a lot of typing if we use *Controlled Vocabularies* for this. A Controlled Vocabulary (CV) is specified in the type for each tier.

    CVs are used to delimit the set of characters that can be typed into a particular tier, providing a pull-down menu of options, and disallowing other entries. This helps prevent typographical errors and increases efficiency in coding, especially across teams of collaborators. CVs are created from the Edit menu and may be shared among multiple tiers and files.

    In our example, we create Controlled Vocabularies for each variable. (If you don't use CVs, you can manually code in each annotation field.)

17. Set up the **Controlled Vocabulary (CV) for the dependent variable** by selecting *Edit: Edit Controlled Vocabularies.*

    We will name our first CV <(ING)> by typing this into the "CV Name" field and clicking the "Add" button in the top part of the window.

    In the "Entry value" field, type <n>, <g> and <o> for alveolar, velar and other variants, respectively, pressing return after each one. Include a useful "Entry description" to remind yourself of what the abbreviations mean.

18. Click the lower "Add" button when you have finished entering the variants.

19. Now, we create a new Tier Type for the dependent variable. We will call it "(ING)." Because each token will have one and only one value for the dependent variable, we need to define a different relationship between the parent (token) and child (ING) tier this time. Specify its *Stereotype* as *Symbolic Association*. This means that annotations created

---

[7] An alternative for some types of variables (especially when there may be more than one token per clause) is to "tokenize" (from the Tier menu) the transcription tier (here "Celeste"). This creates a new tier, a child of "Celeste," with each transcribed word in a separate annotation field. You can then code the dependent variable on a child tier of the tokenized tier without having to retype (or copy) the token word. Both the word and the parent segment can be exported. One issue that arises using the Tokenize command is that the resulting output is *not* exactly time-aligned to the audio. Rather, each transcribed segment is automatically divided into segments of equal duration, one per word. You can also experiment with the Recognizers included in ELAN, which will automatically segment your file into various types of units.

on this tier will have the same location and duration as the annotation on their parent tier ("token"). Specify the "(ING)" CV when you create this Tier Type.

20. Now go to the Tier menu. *Tier: Add New Tier*. This, too, can be named "(ING)." Specify it as a child of the "token" tier.

21. Next, we can **set up the tiers for the independent variables**. Again, we will first create CVs and Tier Types.

22. Create a CV called "POS," to code part of speech for each instance of (ING). Type in <noun>, <adjective>, <present participle>, <gerund> and <verb> as "Entry values." Again, give useful "Entry descriptions."

    Note that here we spelled out the variants. You will then be able to enter them as codes for each token by typing just the first letter (or selecting from the pull-down menu).

23. Create a CV for "# syllables" in the same way.

24. Now that we have created our CVs, we can **create a type for each independent variable tier** *Type: Add New Tier Type*.[8] All independent variable tiers will have the dependent variable tier (ING) as their parent tier. Associate the appropriate CV with each type as you create it.

25. Finally, **create a tier** (*Tier: Add New Tier*) for each independent variable to be coded. Specify the "(ING)" tier as the Parent Tier and select the relevant type for Linguistic Type. Name each tier for the variable to be coded on it (in our example, "POS" and "# syllables"). (See Celeste_with_tiers_for_coding_Step25.eaf.)

26. Additional tiers can be used for social or stylistic factors spanning longer portions of the transcript. They should be associated with segments on the default tier.

27. Additional sets of tiers can be created in the same file for additional dependent variables.

Tip: Tiers can be rearranged onscreen and/or temporarily hidden by right-clicking on the tier label so that only the tiers in current use are displayed.

Aside on types of tiers

Table 1 briefly describes the Tier Types in ELAN (from the ELAN manual), with an indication of how we use them.

---

[8] Having a different type for each independent variable allows us to take advantage of the different CVs. If you were coding without CVs, you could (in principle) define only one Type, e.g. "Independent variable.".

Table 1: Types of types in ELAN

Type of Type	Description and *Example of usage*
None	Annotation is linked directly to the time axis. *Ex: Default tier on which segmentation occurs*
Time Subdivision	Annotation on parent tier can be sub-divided into smaller units, which link to time intervals. No gaps allowed. *Ex: Sentence > words or clauses*
Symbolic Subdivision	Smaller units that cannot be linked to a time sub-interval. *Ex: Word > its semantic fields; also required for Tokenization*
Included In	All annotations fall within borders of parent tier. Gaps between the child annotations are allowed. for dependent variable *Ex: Sentence > Word tier for marking (or "extracting") dependent variable*
Symbolic Association	one-to-one correspondence between the parent annotation and its child annotation. *Exs: Sentence > translation; dependent variable > independent variable*

Isolating and coding tokens

28. Now you are ready to **code the data**. Coding in Annotation mode allows you to see the broadest context (zoom to any degree of detail), while coding in Transcription mode provides a format more like the traditional spreadsheet for coding. In that mode, you cannot see/hear the larger context. We will illustrate using Annotation mode, but for some variables that are "easy" to code (e.g., # of syllables in the token word), it may be more efficient to work in Transcription mode.

29. Move the cursor to the timepoint where you wish to start coding. Often sociolinguists don't use data from the first 15-20 minutes of the recording.

    You can Play one segment at a time (select a segment, then press blue triangle Play button, or Shift+spacebar) or Play from the cursor's current position (black triangle Play button, or hit the spacebar) until you come to the first word containing the first token of (ING). Stop the recording.

    Highlight that portion of the soundwave. Notice how it highlights in both the transcription and the token tier. Double-click on the highlighted portion in the "token" tier. This creates an annotation field associated to that portion of the recording. Type in the word.

30. Repeat until you are satisfied you have enough tokens of the dependent variable. The number of annotations created appears under the tier label at the left edge of the display window.

31. Now return to the first token you marked in the "token" tier. You can find it quite quickly now by using the Grid menu in the top half of the screen. Choose the token tier under Grid, and you can scroll back to the first one easily. (See Celeste_coded_Step31.eaf.)

    Select the first token in the "token" tier, and double-click in the dependent variable "(ING)" tier.

    Type the appropriate code (or select it from the pull-down menu if using Controlled Vocabularies). Remember that you can zoom in and out, listen to the associated annotation, and see extended context in the Grid or Text display in the top part of the window.[9]

32. The most efficient way to move among tokens is with the arrow keys on your keyboard. Type Option + up arrow and that will move you up a tier; Option + right arrow key moves to the next token.

    With the second token selected, double-click in the dependent variable tier to code it. Repeat across the whole tier, and for all subsequent tiers.

    In lieu of Option + arrow keys, you can mouse-click the arrow buttons to the right of the Play Selection button in the middle of the ELAN display or navigate using the Grid display mentioned in Step 31. You can, of course, also navigate entirely by mouse.

Tip: Some variables have very frequently occurring "default" values. It may be quicker to fill those in in Excel (or with "Find & Replace"), rather than typing/selecting their values in ELAN, e.g., if, as part of a cross-linguistic analysis, you are coding English pro-drop where 98% of the subjects are overt.

Export to statistical analysis program

When coding is complete, tiers can be exported to a text file and prepared for analysis. You may first wish to see a quick count of variants for each variable. Select *View: Annotation statistics* and then select the relevant tier, or explore the summary statistics available through the *Search* menu. Your codes, annotations, and timestamps can all be exported for quantitative analysis.

33. Go *File: Export as…: Tab delimited Text*. (If you are working with a lot of speakers, experiment with *Export Multiple Files*…)

34. Select all the tiers that have relevant codes or transcriptions in them (see Figure 3).

35. Select "Separate column for each tier" and "Repeat values…". This means you will get any cases where there are two or more tokens of your dependent variable within one segment.

---

[9] If you discover the need for additional variants as you code, you can add them to the Controlled Vocabulary at any time by selecting *Edit: Edit Controlled Vocabularies...*

36. Select at least the Begin Time in order to retain timestamps with each token and code. We find it useful to select both the hh.mm:ss.ms and the msec format. The first is easy to interpret and use to find tokens by timestamp. The latter is better for re-sorting tokens into chronological order in Excel (after sorting by other characteristics, see § *Export to statistical analysis program*, Steps 39-43).

37. Save as a .txt file. Make sure the filename ends in ".txt." It is useful to create a filename that identifies the speaker. (See Celeste coded exported_Step37.txt.)

38. Open the .txt file in a spreadsheet program like Excel (use *File: Import* and skip directly to "Finish").[10]

[10] We have not had good luck importing IPA symbols into Excel. Open Office does a better job. Selecting IPA-SAMPA as the Default Language for the tier alleviates this problem.

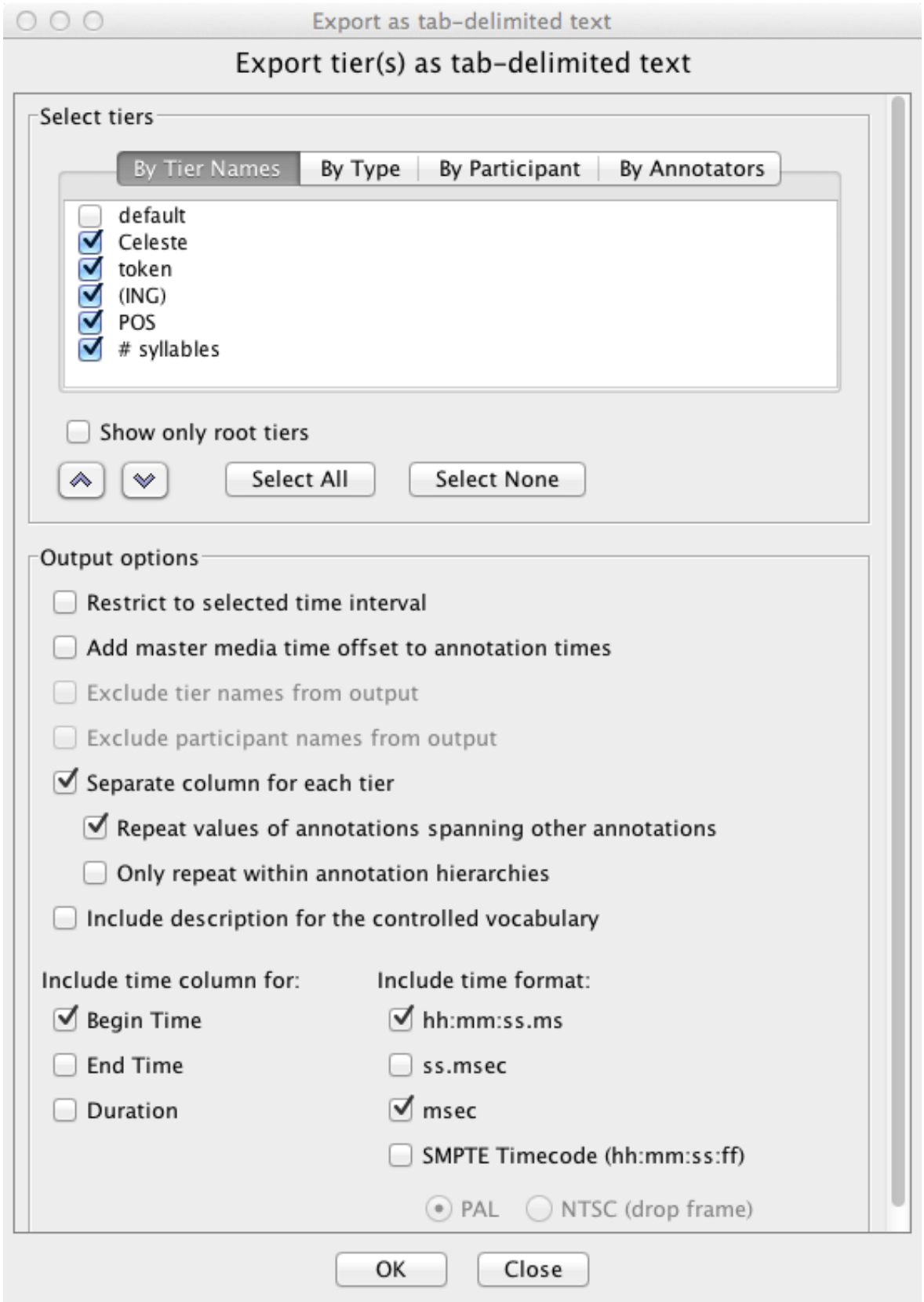
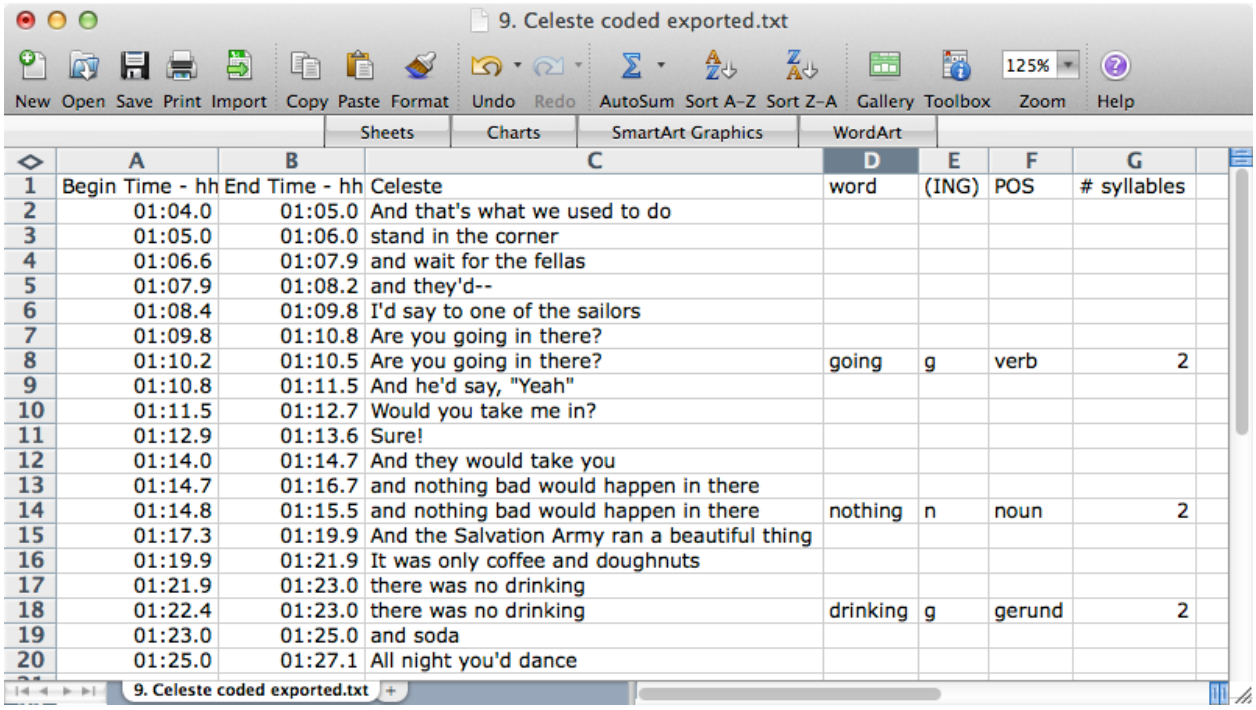


Figure 3: Screenshot showing export settings for preparing for statistical analysis (Step 33)

When you open the file in Excel, you will see a column labeled "Begin Time" with a timestamp (two timestamp columns, if you selected two time formats in the Export window), a column with the annotation (transcription) linked to each token, and columns for each coded variable. You will see many rows, only some of which contain a coded token. See Figure 4. The many extra rows (without token codes) are created by selecting to export the transcription tier – not every annotation in that tier has coded tokens associated to it. This is an easy problem to fix:



	A	B	C	D	E	F	G
1	Begin Time - hh	End Time - hh	Celeste	word	(ING)	POS	# syllables
2	01:04.0	01:05.0	And that's what we used to do				
3	01:05.0	01:06.0	stand in the corner				
4	01:06.6	01:07.9	and wait for the fellas				
5	01:07.9	01:08.2	and they'd--				
6	01:08.4	01:09.8	I'd say to one of the sailors				
7	01:09.8	01:10.8	Are you going in there?				
8	01:10.2	01:10.5	Are you going in there?	going	g	verb	2
9	01:10.8	01:11.5	And he'd say, "Yeah"				
10	01:11.5	01:12.7	Would you take me in?				
11	01:12.9	01:13.6	Sure!				
12	01:14.0	01:14.7	And they would take you				
13	01:14.7	01:16.7	and nothing bad would happen in there				
14	01:14.8	01:15.5	and nothing bad would happen in there	nothing	n	noun	2
15	01:17.3	01:19.9	And the Salvation Army ran a beautiful thing				
16	01:19.9	01:21.9	It was only coffee and doughnuts				
17	01:21.9	01:23.0	there was no drinking				
18	01:22.4	01:23.0	there was no drinking	drinking	g	gerund	2
19	01:23.0	01:25.0	and soda				
20	01:25.0	01:27.1	All night you'd dance				

Figure 4: Unsorted coding file exported from ELAN and opened in Excel (Step 38)

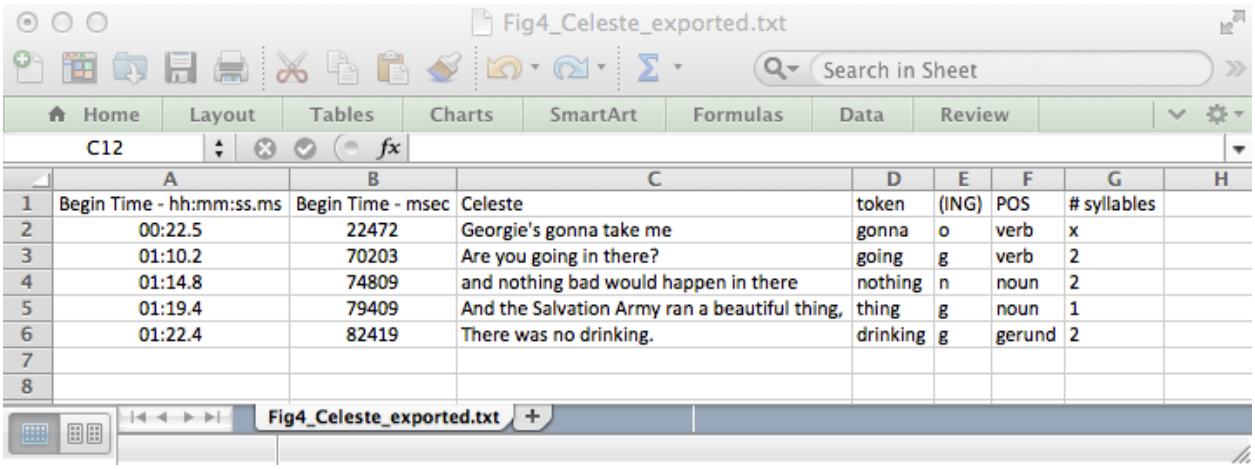
39. In the Excel file, Select All.

40. Sort by the Dependent Variable column (in the example in Figure 4, "(ING)" in Column E). This will place all rows with coded tokens together at the top of the table and all rows without coded tokens together below.

41. Select and delete all rows without coded tokens.

42. If you left any default values blank while coding in Excel, sort the rows by that column, and use the Fill Down function to fill in the default value for all blank cells in that column. Repeat for additional variables.

43. Select All and *Data: Sort*, this time by the "Begin Time" timestamp column. Use the column with msec formatting to avoid odd complications with timestamps greater than one hour. Your file should now appear as in Figure 5.



	A	B	C	D	E	F	G	H
1	Begin Time - hh:mm:ss.ms	Begin Time - msec	Celeste	token	(ING)	POS	# syllables	
2	00:22.5	22472	Georgie's gonna take me	gonna	o	verb	x	
3	01:10.2	70203	Are you going in there?	going	g	verb	2	
4	01:14.8	74809	and nothing bad would happen in there	nothing	n	noun	2	
5	01:19.4	79409	And the Salvation Army ran a beautiful thing,	thing	g	noun	1	
6	01:22.4	82419	There was no drinking.	drinking	g	gerund	2	
7								
8								

Figure 5: Sorted coding file (Step 43)

44. Your file is ready for analysis with a program like Rbrul that allows token codes in multiple-character string format. Save the file as .txt or .csv and then import it to your analysis program.

45. If you will be using Goldvarb, use Excel's Find and Replace function to create one-character codes for each variant of each variable. (You could also make these replacements in ELAN, before exporting.)

46. The final step if you will be using Goldvarb is to create a column that contains the full token string. This is done by typing a function that concatenates the contents of each column in a new cell in the first token's row, for example:

    =“(”&E2&F2&G2&“ ”&D2&“ ”&C2&“ ”&B2

    Goldvarb considers only the material between a left parenthesis and white space as a token. Any material to the right of the white space is ignored by the program but may be useful to the user.

47. Then use Fill Down (from the Edit menu) to copy that function to every row with a token in it. After checking that the function is referring to the correct rows and cells in your spreadsheet, copy *only* that column to a new .tkn file in Goldvarb. Figure 6 shows a concatenation function to create token strings.

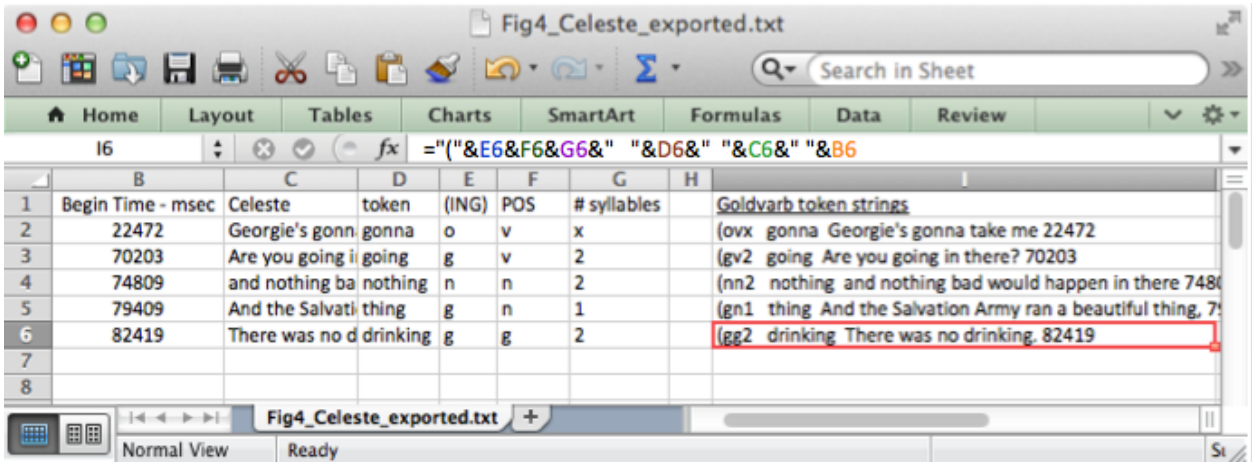


Figure 6: Preparing token strings for Goldvarb (Step 46)

6. Links to prepared files for practice

That's all there is to it. You can try these steps by using the sample files, available as a compressed package at http://projects.chass.utoronto.ca/ngn/zip/Celeste_for_ELAN.zip [3.4 MB]. Files have been created for each step of the process – much like the pre-baked pie that the TV chef can pull out of the oven just seconds after putting the raw pie in. They are:

Celeste_Step4.wav

Celeste_segmented_Step8.eaf

Celeste_transcribed_Step12.eaf

Celeste_with_tiers_for_coding_Step25.eaf

Celeste_coded_Step31.eaf

Celeste coded exported_Step37.txt

Celeste_transcript_ING_marked_AppD_Step1.doc

Celeste_Aligned_AppD_Step15.txt

Celeste_segmented_AppD_Step12.txt

Celeste_cleaned_AppD_Step11.txt

Many other features of ELAN can be learnt from the ELAN website and manual. In this paper, we have focused on ELAN's utility for preparing data for variationist analysis. Although this function was not envisaged by its creators, we have shown that it affords variationist sociolinguists a number of advantages over their previous practice: principally, retaining a close

and active link between coding and the original source recording, and secondly, allowing for export in a number of formats suited to subsequent analysis of variation.

By explaining in detail the steps that are involved in the process, we hope that we can inspire both variationists to expand their software repertoire to include ELAN and also inspire documentary linguists to add the systematic coding of variation to their work.

References

Boersma, Paul & David Weenink. 2014. Praat: doing phonetics by computer [Computer program]. Version 5.3.80. http://www.praat.org/ (08 July 2014.)

British Sign Language Corpus Project. 2012. http://www.bslcorpusproject.org/data/ (5 October 2014.)

Gorman, Kyle, Jonathan Howell & Michael Wagner. 2011. Prosodylab-Aligner: A tool for forced alignment of laboratory speech. *Proceedings of Acoustics Week in Canada*, Quebec City.

Hazen Kirk. 2006. IN/ING Variable. In Keith Brown (ed.) *Encyclopedia of Language & Linguistics, 2nd edn*, volume 5, 581-584. Oxford: Elsevier.

Kisler, Thomas, Florian Schiel & Han Sloetjes. 2012. Signal processing via web services: the use case WebMAUS, *Digital Humanities 2012, Hamburg, Germany*, pp. 30-34.

MacWhinney, Brian. 2000. *The CHILDES Project: Tools for analyzing talk. Third Edition.* Mahwah, NJ: Lawrence Erlbaum Associates.

Nagy, Naomi & Miriam Meyerhoff. 2013. Extending ELAN into Quantitative Sociolinguistics. 3rd International Conference on Language Documentation and Conservation, Manoa, Hawai'i.

Nagy, Naomi & Miriam Meyerhoff. 2013. Extending ELAN into Variationist Sociolinguistics. NWAV 42, Pittsburgh.

Nagy, Naomi & Miriam Meyerhoff. 2013. Extending ELAN into Variationist Sociolinguistics. NWAV 43, Chicago.

Rosenfelder, Ingrid, Joseph Fruehwald, Keelan Evanini, and Jiahong Yuan. 2011. FAVE (Forced Alignment and Vowel Extraction) Program Suite. http://fave.ling.upenn.edu (5 October 2014.)

Wittenburg, Peter, Hennie Brugman, Albert Russel, Alex Klassmann & Han Sloetjes. 2006. ELAN: a Professional Framework for Multimodality Research. In *Proceedings of LREC 2006, Fifth International Conference on Language Resources and Evaluation*.

Appendix A: Links to useful freeware mentioned in this paper

Audacity http://audacity.sourceforge.net/

CuPED http://sweet.artsrn.ualberta.ca/cdcox/cuped

ELAN http://tla.mpi.nl/tools/tla-tools/elan

FAVE http://fave.ling.upenn.edu/

Goldvarb http://individual.utoronto.ca/tagliamonte/goldvarb.htm

Praat http://www.fon.hum.uva.nl/praat/

R http://www.r-project.org

Rbrul http://www.danielezrajohnson.com/rbrul.html

tabber http://www.nowme.ca/lib/hlvc/tabber.html (Mac, Linux)

pctabber http://projects.chass.utoronto.ca/ngn/HLVC/tabber_for_legacy_files/pctabber.exe (Windows)

WebMAUS http://clarin.phonetik.uni-muenchen.de/BASWebServices/

Information and instructions for coding with ELAN:

http://projects.chass.utoronto.ca/ngn/HLVC > Resources > For Researchers

http://individual.utoronto.ca/ngn/pdf/ELAN_Handout_Barras_2013.pdf

Coding assignment with step-by-step instructions:

http://individual.utoronto.ca/ngn/LIN/courses/LIN351/LIN351_project.htm

To use for a class, request data files from the first author.

Appendix B: Other good things to learn to use in ELAN

- Vertical zoom & horizontal zoom in the .wav window (Control + click)
- Resizing display with slider at bottom right of display
- Navigate with "Grid" and "Text" (choose relevant tier from pull-down menu)
- Control speed and volume of playback in "Controls"
- List of "shortcuts" from the View menu (key combinations)
- Change order of tiers (by dragging tier names)
- Delete annotation (select it, Option+D)
- Change size of annotation (select it, then Option+Drag edge with mouse)
- Templates to set up tiers for many files

The ELAN manual gives clear instructions for all these.

Appendix C: ELAN's Technical details

- The textual content of annotations is in Unicode and the transcription is stored in an XML format.
- ELAN delegates media playback to an existing media framework, like Windows Media Player, QuickTime or JMF (Java Media Framework). As a result, a wide variety of audio and video formats is supported and high performance media playback can be achieved.
- ELAN is written in the Java programming language and the sources are available for non-commercial use. It runs on Windows, Mac OS X and Linux.

Appendix D: Creating ELAN files for legacy transcripts

This protocol was created for a particular corpus of Word .doc transcriptions.[11] You will need to adjust certain aspects, particularly in the first section, to the extent that the original transcribers of your corpus made different formatting decisions.

Clean up .doc transcriptions (in Word) - batch processing

1. In Word, open all the files that need to be "ELANized".
   Edit as necessary to clean up. You can open several/all of the files and then click select "All Open Documents" from the pulldown menu in the Search window in Word. Be careful!

   (The **red/bold** is exactly what you type in the "Find what" and "Replace with" boxes in Word, except the word "space" represents one blank space.)

   a. Remove all tabs: Replace **^t** with **SPACE**
   b. Remove double spaces: Replace **SPACE SPACE** with **SPACE** (Repeat as necessary)
   c. Make each sentence (or clause – your choice – these will be your annotation entries) start on a new line:
      Replace **[** with **^p[** (if "[ ]" mark speaker codes)
      Replace **.** with **. ^p**
      Replace **?** with **? ^p**
      Replace **!** with **! ^p**
      Replace **(…)** with **.^p**
      Replace **…** with **.^p**
      Replace **. ^p"** with **."^p** (for quoted speech)
      Replace **?^p"** with **?"^p**
      Replace **!^p"** with **!"^p**
      Replace **^p SPACE** with **^p**
      Replace **SPACE^p** with **^p**
      Replace **^p^p** with **^p** (repeat until Word finds 0)
   d. Optionally, delete: "(laughter)", "mhm" etc., ONLY if they are a full turn. Then**:**
      Replace **^p.^p** with **^p**
      Replace **- SPACE** with **SPACE**
      Replace **^p SPACE ^p** with **^p**
      Replace **[^#] SPACE ^p** with nothing
      Replace **[^#] SPACE .^p** with nothing
      Replace **[^#] SPACE [** with **[**
      Replace **[^#^#^#] SPACE ^p** with nothing
      Replace **[^#^#^#] SPACE .^p** with nothing

---

[11] The corpus is Hoffman and Walker's (2010) Contact in the City corpus of Toronto English. The authors thank Vina Law, Naomi Cui, Minyi Zhu and Vanessa Racine for their aid in developing this protocol.

Replace **[^#^#^#] SPACE [** with **[**

e. Make sure any other comments are enclosed in double parentheses, if you will be using FAVE to force align your files: **(( ))**

2. Save with one clause/phrase/intonation unit per line.
3. List the speakers on the first line of the file. Use this format and order :

   XXX *Main-participant-name*, 1 *Interviewer-name*, *2 Second-interviewer-name, 3 Any-other-participant-name, 4 etc.* *(Italicized information is optional.)*

   Note the lack of square brackets on this line and that speakercodes are separated by commas.
4. Save the .doc transcripts as SpeakerCode_cleaned.doc.

Segmenting audio in ELAN and editing individual .doc transcript files

5. In ELAN, create a new file and associate the appropriate audio or video file (e.g. .wav).
6. Switch to *Options*: *Segmentation Mode*. Segment recording on the default tier – one segment for each line (paragraph mark) in SpeakerCode_cleaned.doc. (You'll want that transcript .doc open next to ELAN on your screen.) Save the ELAN file as SpeakerCode_segmented.eaf.
7. In Word, divide up any super-long clauses on to separate lines (otherwise they will be hard to transcribe and analyze). After commas that indicate a division between clauses or breath groups (but not other commas), hit Return. Make sure this matches how you segment in ELAN.
8. Deal with overlapping speech. When you segment a line with overlapping speech, it will need three segments, all on one tier:
   a. The first segment covers the time that the first speaker talks before the interruption/overlap.
   b. The second segment covers the time when both are talking.
   c. The third is for the time after the interruption/overlap, when one person continues to talk.

		Segment 1	Segment 2	Segment 3
ELAN segmenting				
what you hear:	[007]	The man ate	an icecream cone with	sprinkles.
	[1]		He was so funny.	

what you read in the text file:	[007] The man ate [007] an icecream cone with [1] He was so funny. [007] sprinkles.

Figure 7: Aligning overlapping speech

Note: On the line in the text file with both speakers, the speakers must appear in the order listed at the top of the file.

9. If you make changes in Word, save the .doc transcripts as SpeakerCode_edited#.txt (Tab-delimited text file format; select Other encoding: Unicode UTF-8, NOT Insert line breaks). Use a new # for each version so that you can back-track if necessary.
10. Save files as SpeakerCode_cleaned2.doc.
11. Save the .doc transcripts as SpeakerCode_cleaned.txt (Tab-delimited text file format; When prompted, select Other encoding: Unicode UTF-8, NOT Insert line breaks).

Merge the timestamps from ELAN with the transcript text

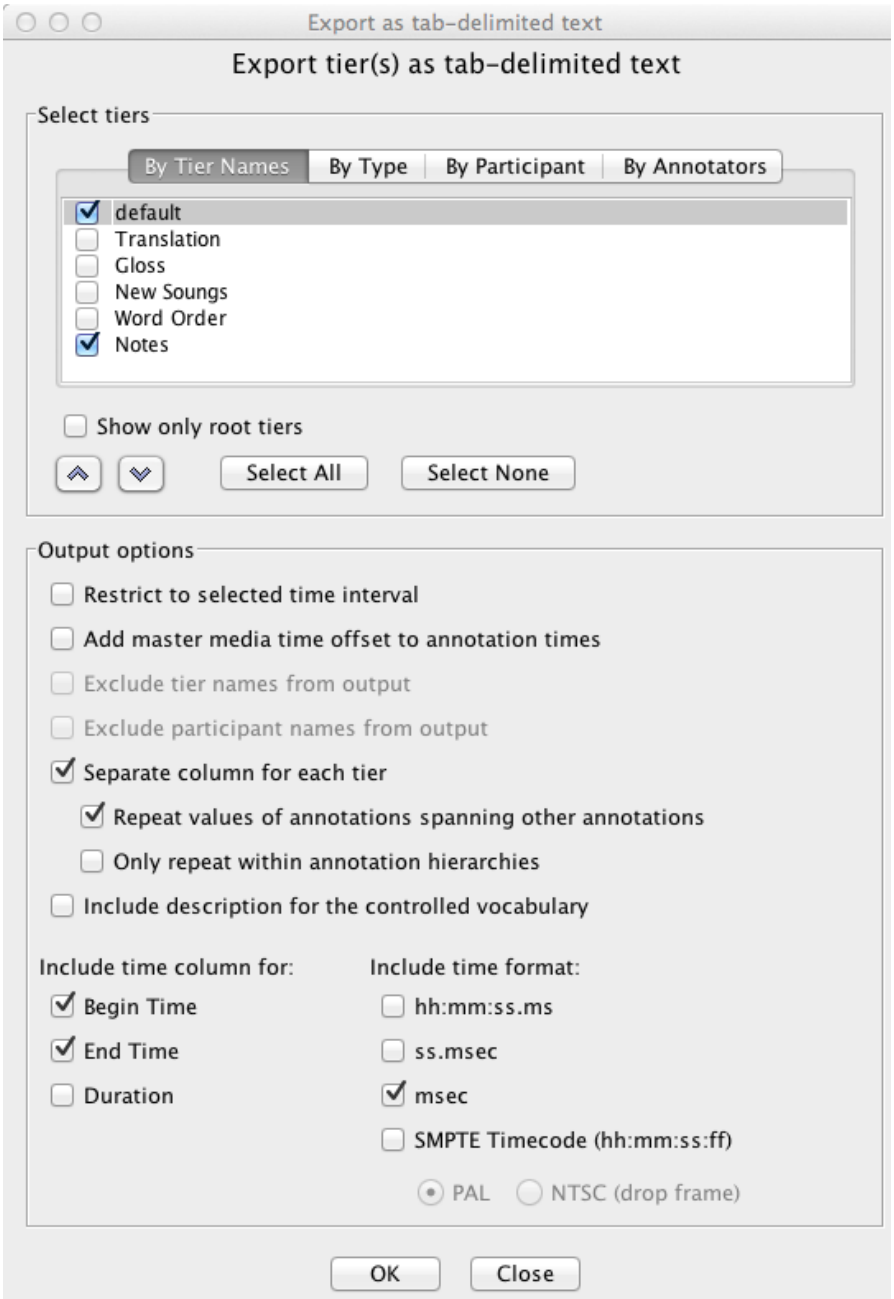


Figure 8: Settings for exporting tiers as tab-delimited text

12. Export the (timestamp) file from ELAN as a .txt file. Use these settings:
    a. Select the tier(s) that have segments marked on them in the top of the window and click OK.
    b. When prompted for text encoding, select "Unicode UTF-8" encoding.
    c. Save as SpeakerCode_segmented.txt. This provides only the timestamps (and any notes in the Notes tier).
    d. It may be helpful to sort the rows of SpeakerCode_segmented.txt in Excel, by duration, then delete the rows that are too short to possibly contain a word, e.g. < 150 msec. Then re-sort by Begin Time and combine with the text files. (This will get rid of some very short annotations created by very slight overlaps in annotations or accidental double-clicks in Segmenting mode.)
13. Download the script from http://www.nowme.ca/lib/hlvc/tabber.html (Mac, Unix) or http://projects.chass.utoronto.ca/ngn/HLVC/tabber_for_legacy_files/pctabber.exe (Windows). This insert tabs that align each speaker at a different indent level. Additional instructions are available at the same site for the tabber and at http://projects.chass.utoronto.ca/ngn/HLVC/tabber_for_legacy_files/pctabber_instructions.pdf for pctabber.

    On a Mac:
    a. Unzip and save tabber in folder with transcript files.
    b. Open Terminal.app (in Utilities)
    c. Go to directory where .txt files and script are (cd "Directory").
    d. Type ./tabber "FILENAME" or ./tabber "FOLDERNAME"

    In Windows:
    a. Put the pctabber.exe file in the same folder as the .txt transcripts to be tabbed.
    b. Double click the pctabber.exe file and it will open the command line.
    c. Enter the file names of the .txt files one by one, hitting "Enter" after each file name.

This will create a tabbed_SpeakerCode.txt for each SpeakerCode_cleaned.txt in the folder.

14. In Excel, combine the two .txt files for one speaker (the time stamps and the transcripts) so that the Begin and End time stamps (from SpeakerCode_segmented.txt) are on the same row as the corresponding annotations (from tabbed_SpeakerCode.txt).
15. Label columns with SpeakerCodes. Save as SpeakerCode_Aligned.txt.
16. Open ELAN. Import SpeakerCode_Aligned.txt and match up tiers and columns appropriately in the pop-up window. Include the msec Begin Time and msec End Time column. Include the Annotation column for each speaker. Unclick any other columns. Select the line on which the transcription begins in that window (Line 2, because the column labels are on Line 1). Save as SpeakerCode.eaf.
17. After importing, you will need to relink the SpeakerCode.wav file into your new SpeakerCode.eaf file. In ELAN, *Edit: Linked files*.

Quality control

18. Spot check throughout to make sure that the transcription matches the sound file. If it doesn't, go back and re-segment to over-ride in the problem area, or fix the row alignment in SpeakerCode_Aligned.txt.

Appdendix E: Concordance and frequency counts

1. Create Type: symbolic subdivision, call it "***tokenizing***" (File > Multiple File Processing > Edit Multiple Files)
2. Create Tier: call it "***tokenized***" as a child of the Speaker tier. Do for all speaker files. (File > Multiple File Processing > Edit Multiple Files, if your speaker tiers all share the name "Speaker")
3. Tokenize files (one-by-one?), using the "***tokenized***" tier as the landing tier for the process.
4. Change the case of all annotations on the "***tokenized***" tier (Tier > Change Case of Annotations) so that "cat" matches "Cat". (I think you have to do this in each file separately.)
5. Scrub white space (invisible spaces, tabs, etc.) from all files, but only from the ***tokenized*** tier. (File > Multiple File Processing > Scrub Transcriptions…)
6. Delete punctuation & numbers (e.g., Interview Question numbers) from ***tokenized*** tier of all files (Search > Find & Replace in Multiple files).

   To find all punctuation, use regular expression `\p{Punct}`. Replace with nothing.

   To find all numbers, use regular expression `\d` . Replace with nothing.

7. Save.
8. Files > Multiple File Processing > Statistics for Multiple Files (use only ***tokenized*** tiers)
9. Save Annotation Statistics as LANGUAGE_concordance.txt.
10. Open in Excel.
11. Alphabetize the file. (Data >Sort)
12. Save only the columns labeled "Tier", "Annotation" and "Occurrences".
13. Check that you have included only the ***tokenized*** tier.
14. Save.

Now you have a list of all the words in your corpus in one column and the frequency of their use in the next column.

To use these frequency data in an analysis:

15. Use the Excel function *vlookup* to create a lexical frequency factor column in your token file that references this new file. Create the formula in the first row of tokens.
16. Fill it down to all rows of tokens for that language.

    In my file, this *vlookup* function looks like this:

```
=VLOOKUP(Q2,'Naomi''s
iMac2008:Users:nagy:LIN456:[ukrainian concordance.xlsx]
freq'!$B$2:$C$14006,2,TRUE)
```

The blue argument indicates the cell of the token (word) in that row – the word for which we want the lexical frequency.

The green argument indicates the location of the two columns of information in the file exported from ELAN – the column that lists all the words and the column that lists the number of occurrences of each word. The orange parts need to be adjusted to check the frequency list for the appropriate language. Make sure all the relevant rows are selected. (OK to include blanks at the bottom.)

The red "2" indicates that we want to get the value from the second column of the green array.

The black "TRUE" means that we want an exact match. "FALSE" would match the closest word, even if not exactly the same.

You can either use lexical frequency as a continuous independent variable, or you can bin it into several categories (e.g., low, medium, high frequency) and have a discrete independent variable.