Computational support for early elicitation and classification of tone

Steven Bird, Haejoong Lee, Quynh-Chi Nguyen

University of Melbourne and University of Pennsylvania

Computational support for early elicitation and classification of tone

Abstract

Investigating a tone language involves careful transcription of tone on words and phrases. This is challenging when the phonological categories – the tones or melodies – have not been identified. Effects such as coarticulation, sandhi, and phrase-level prosody appear as obstacles to early elicitation and classification of tone. This article presents an approach to this problem along with open source software. Users listen to words and phrases of interest, before grouping them into clusters having the same tonal properties. The software continuously analyzes the clusters and suggests additions and corrections. In this manner, it is possible to quickly annotate words of interest in extended recordings, and compare items that may be widely separated in the source audio to obtain consistent labelling. Users have reported that it is possible to train one's ear to pick up on the linguistically salient distinctions. We illustrate the approach with examples from Eastern Chatino (Mexico) and Alekano (Papua New Guinea).

1. Introduction

During early elicitation, transcription practice evolves as we tune into the linguistically salient contrasts. For segmental distinctions, it is usually straightforward to begin with narrow phonetic transcriptions and gradually leave out details once they are found to be non-contrastive. For instance, after noting that voiceless obstruents are aspirated in syllable onset position, we may decide to stop marking aspiration. Over time, such conventions make it possible for transcription to proceed more quickly, and for the results to be more readable. Yet all the time, we try to remain open to detecting new contrasts (cf Hyman 2001).

The situation is often more acute for tone. To begin with, the IPA notation notation for tone is cumbersome, and it is also arbitrary with its five levels with the corresponding contours. In the experience of many, it is more effective to draw stylized contours, e.g [-_/]. The use of elicitation frames may effect the target word in unpredictable ways, and we have to sort out the various contributions of phrase-level prosody (e.g. phrase boundary tones), local phonological alternations, and phonetic interpretation (e.g. tonal coarticulation). Eyeballing F_0 traces sometimes helps, but these are often misleading.

In short, we are trying to uncover discrete underlying tonal categories like High and Low, without knowing much about the function which gave them their observable phonetic realization. When we detect a small pitch difference between consecutive syllables, we can't be sure whether this points to an underlying contrast, or whether it is just a case of tonal coarticulation. After a while, perhaps a week or a month, our language acquisition device has become engaged, and we begin to ``hear" the tone. Ideally, we would reach this stage more quickly and reliably, so that we can produce reliable transcriptions in a limited amount of time. Field trips often have a short duration, and so speeding up this ear-training process may have a significant impact on the quality and quantity of the transcriptions that can be made in the field, and this may in turn help identify gaps where further data collection would be useful. Our work is intended to occupy this niche of early elicitation.

This paper presents a free, open source software tool called Toney that is intended to support the early elicitation and classification of tone language data. Toney displays forms on a canvas, and the user can listen to the forms and group them into clusters. By reviewing the clustered items, it is easy to learn to hear the tonal categories and identify mis-classified items. The software analyzes the contents of each cluster and suggests how unclassified items should be grouped. By using this software, the user can quickly learn the linguistically salient tonal categories, and annotate extended audio recordings.

This paper is organised as follows. In section 2, we give an extended example of an early elicitation problem in Alekano. In section 3, we give a worked example of how the tool is used to classify words in isolation. In section 4, this is broadened to include sentence frames, multiple speakers, and further categorizations that may be useful. In section 5 we explain how the system "learns" the tonal categories and recommends additions and revisions to the categorizations made by the user. The paper closes with a discussion and conclusions.

2. Background: early elicitation of tone

In order to motivate our approach, we begin with an example of early elicitation in Alekano (ISO gah), a language spoken by about 20,000 people in the Eastern Highlands Province of Papua New Guinea. Consider the following sentence:

(1)   gènēzá àní'gùvè *I saw (a) tongue* [audio: geneza-F1]

The target word *gèn̄ēzá* (tongue) appears to have a rising sequence, which we have transcribed as low-mid-high. However, the position of the mid between L and H is suspicious: perhaps it is really a low tone that has been raised in the context of H. If so, we may may be able to drop the mid tone category, and write instead *genezá aní'guve* (leaving low tone unmarked), and posit a rule of phonetic interpretation in which a low tone is raised in the L_H environment.

After further elicitation we build up a picture of the inventory of tonal melodies on words with a fixed syllable shape, in this case CVCVCV words:

(2)
a. LLH geneză aní'guve *I saw tongue* [audio: geneza-F1]
b. LHH golání aní'guve *I saw blood* [audio: golani-F1]
c. LHL gosíha aní'guve *I saw snake* [audio: gosiha-F1]
d. HLH lágahá aní'guve *I saw fish* [audio: lagaha-F1]

Further possibilities for these words show up when we add a definiteness marker.

(3)
a. LLL geneza-má aní'guve *I saw the tongue* [audio: geneza-F2]
b. LHH golání-má aní'guve *I saw the blood* [audio: golani-F2]
c. LHL gosíha-má aní'guve *I saw the snake* [audio: gosiha-F2]
d. HLL lágaha-má aní'guve *I saw the fish* [audio: lagaha-F2]

Here, the final syllable of (3a) and (3d) becomes L, and we could posit a rule H->L/L_#H. The phonetic rule which raises this L seems to be variable, and we get both level and rising variants, e.g. [audio: geneza-F2, geneza-F2b].

There are problems with this approach. It establishes a sequence of hypotheses which purport to account for a selection of the data. However, we would like more than this. First, we want to be faithful to the data, confident that we are not deluding ourselves by transcribing the materials opportunistically, to supports our early hypotheses. Second, we want to be accountable, retaining the link between an individual transcribed form, the audio recording on which it is based, and the full set of forms that are transcribed the same way. Third, we would like to tune our ears to linguistically salient aspects of pitch (the usual acoustic correlate of tone) so that we can transcribe more quickly and reliably over time. These goals are challenging when we have not identified the tones, and don't know how to attribute putative contrasts to phonological categories or phonetic effects. In the following sections, we introduce our software tool and show how it addresses these problems.

3. Classifying words in isolation

We begin by illustrating the use of the software in the context of an early elicitation session. We assume that a set of words and glosses have been transcribed and reviewed, and then recorded at the end of the session. If multiple speakers are involved, the wordlist is recorded separately with each speaker, to minimize the risk that one will copy the intonation of the other. We label the words using acoustic analysis software (see Figure 1).

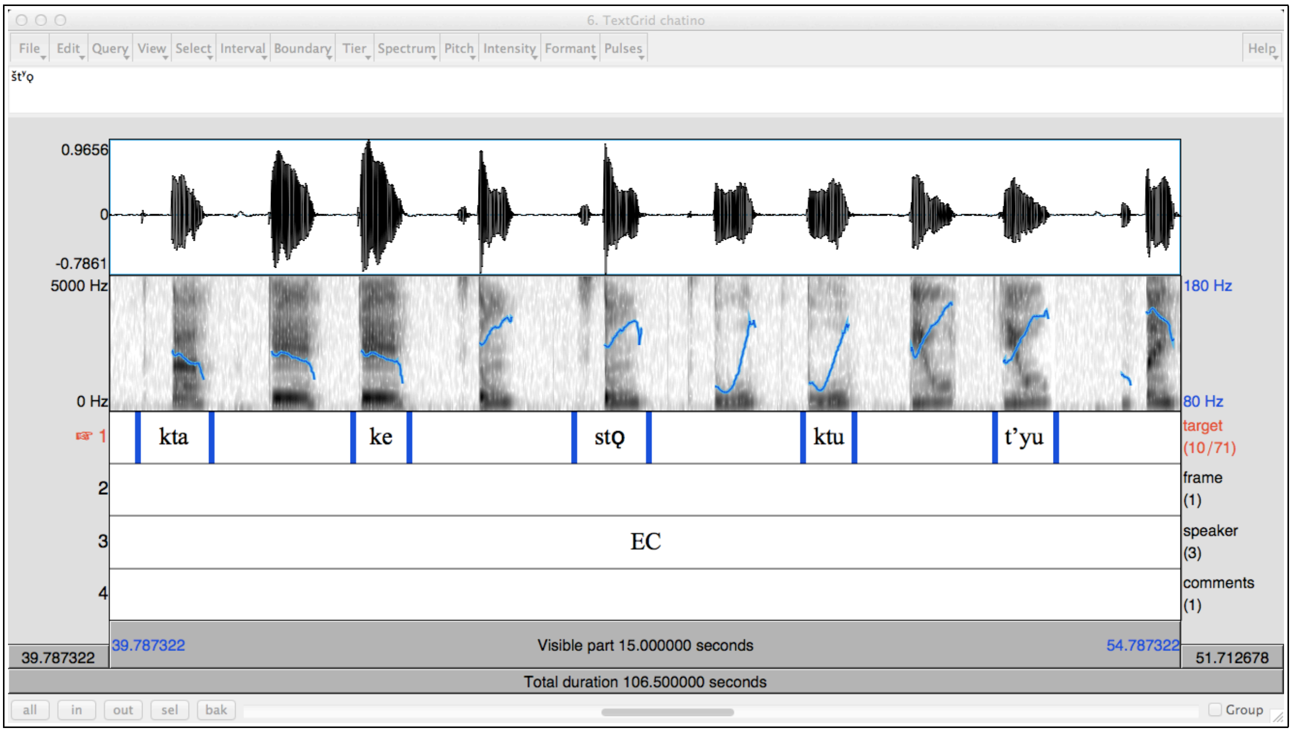


Figure 1: Praat annotation of selected lexemes, using the "target" tier. In this case, only one instance for each lexeme has been labeled.

The annotation is required to have at least the following three tiers: target, frame, and speaker. In this instance, the words were produced in isolation, and so no frame is specified. The speaker is the same for the entire file, so there is a single annotation with the speaker's initials (here, EC). Next, we open the file using Toney, and see the words scattered on a canvas, as shown in Figure 2.

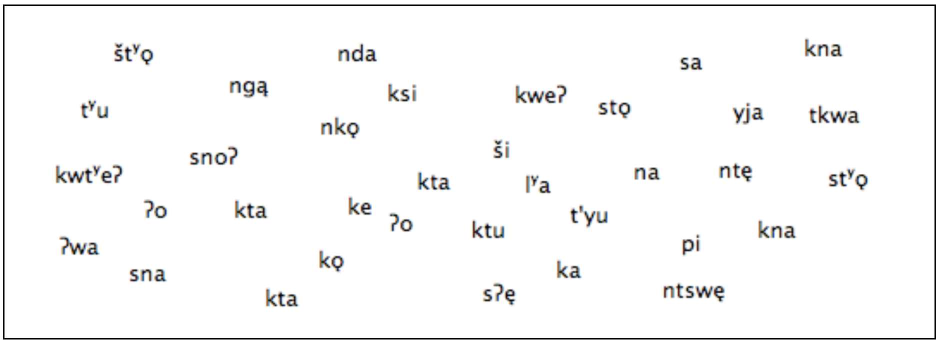


Figure 2: Toney's initial display of the Eastern Chatino nouns

The user can click on the words to hear them, and drag similar-sounding words into groups on the canvas. In Figure 3, the forms have been arranged into three groups, corresponding to rising, level, and falling melodies. (Note that the two instances of ʔo and the three instances of kta have different melodies.)

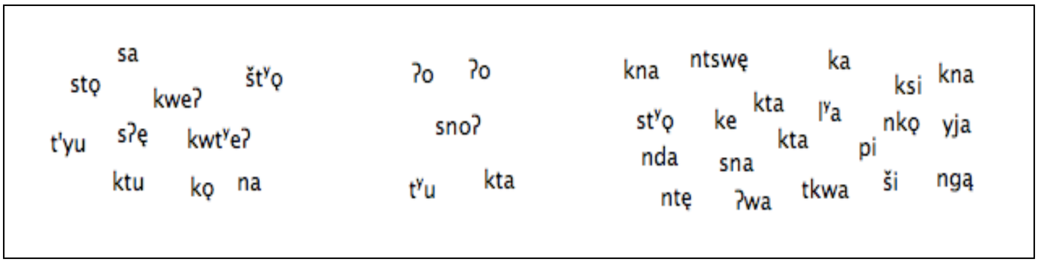
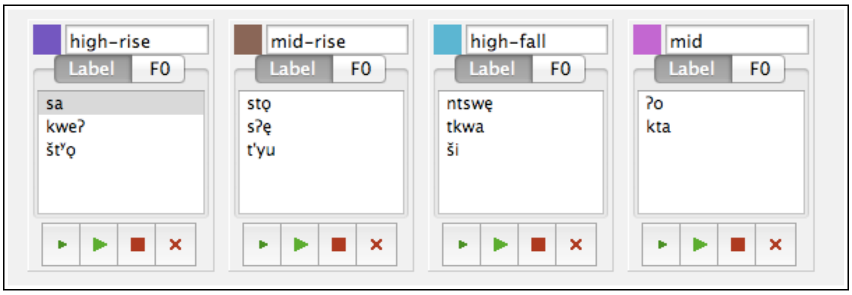


Figure 3: Manually arranging words into clusters on the canvas

Once a set of words with the same tone melody has been identified, it is moved into a cluster in the top half of the display, as shown in Figure 4. At the top of each cluster display is a unique color (to be explained later) and a user-assigned label. There is a row of four buttons at the bottom. The first button plays back all of the forms of the cluster in sequence. Usually, any outliers are immediately obvious, and they can be moved to another cluster or else back to the canvas. The second button plays all the forms with their elicitation frames, permitting them to be heard in context. The remaining buttons are for stopping playback and for deleting the cluster (respectively). Users can create an arbitrary number of clusters.



high-rise
sa
kweʔ
štʸǫ

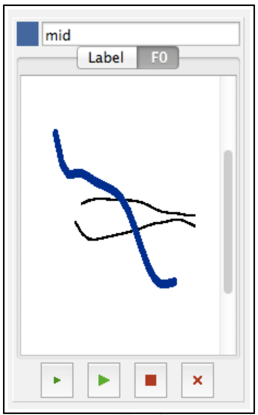
mid-rise
stǫ
sʔę
t'yu

high-fall
ntswę
tkwa
ši

mid
ʔo
kta

Figure 4: Establishing tone clusters

Each cluster also has two tabs, for the individual item labels (Figure 4) and for F_0 contours (Figure 5). The F_0 contours for all forms in the cluster are overlaid. Here we have added a falling-tone word to the mid cluster, and it stands out in the F_0 display for the mid tone. We can click on the contour to hear it, and switch back to the "Label" tab to see which word is highlighted and remove that word from the cluster.



mid

Figure 5: F_0 display

At the end of a session, the labels are saved back to the Praat file, and any forms that have been clustered will appear with a cluster label. For instance, *kweʔ*, from the first cluster will now appear in the Praat file as *kweʔ:high-rise*. (NB. it is useful to adopt compact labels for maximum readability in Praat, e.g. *HR* instead of *high-rise*).

4. Adding Information about Sentence Frames and Speakers

F_0 contours are scaled relative to a speaker's pitch range. The contour for a word is sensitive to its phrasal context. Thus, it is usual to elicit tone data by varying the target word within sentence frames, enabling us to identify the relative rather than absolute differences between tonal melodies. This also avoids the situation where, in order to be uttered in isolation, a word carries phrase- and utterance-level prosodic information, such as a final fall to the bottom of the speaker's pitch range. Since the tones of a sentence frame can interact with the tones of the target word, it is best to use a variety of frames, controlling for phonological and morphosyntactic context. An example of the frame annotation is shown in Figure 5.

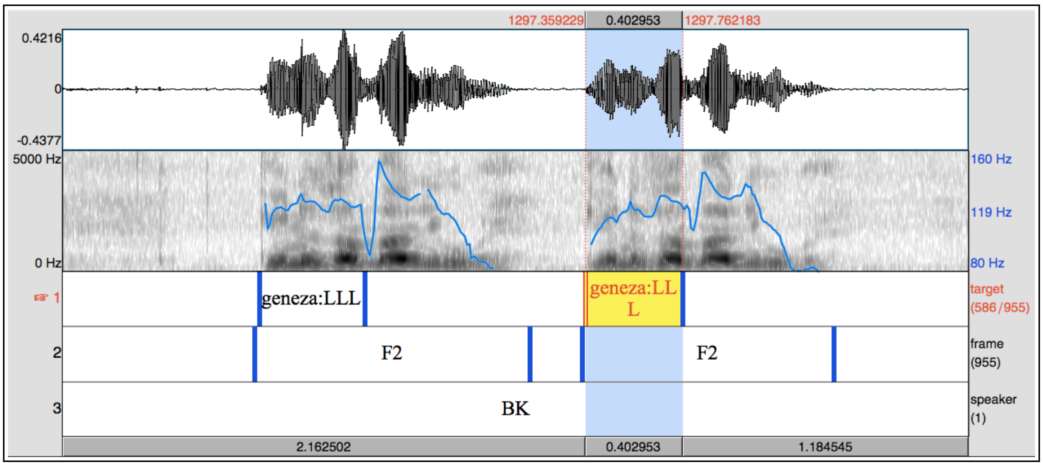
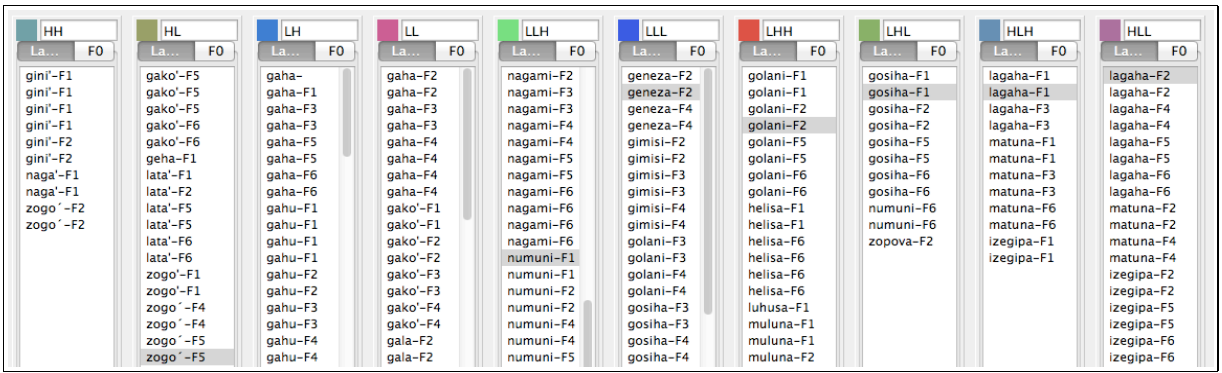


Figure 6: Alekano target words in a sentence frame

Toney supports playback of target words (by clicking) and whole frames (by right-clicking). Figure 6 shows a partial screenshot once all the items have been classified. We can listen to all forms in the LLH column to verify that the second L tone is raised, and confirm that these are distinct from the LHH column. We can observe that *luhusa-F1* is the only form to appear in both LLH and LHH columns, something which will need to be verified in the recordings of the other two speakers.



HH	HL	LH	LL	LLH	LLL	LHH	LHL	HLH	HLL
gini'-F1	gako'-F5	gaha-	gaha-F2	nagami-F2	geneza-F2	golani-F1	gosiha-F1	lagaha-F1	lagaha-F2
gini'-F1	gako'-F5	gaha-F1	gaha-F2	nagami-F3	geneza-F2	golani-F1	gosiha-F1	lagaha-F1	lagaha-F2
gini'-F1	gako'-F5	gaha-F3	gaha-F3	nagami-F3	geneza-F4	golani-F2	gosiha-F2	lagaha-F3	lagaha-F4
gini'-F1	gako'-F6	gaha-F3	gaha-F3	nagami-F4	geneza-F4	golani-F2	gosiha-F2	lagaha-F3	lagaha-F4
gini'-F2	gako'-F6	gaha-F5	gaha-F4	nagami-F4	gimisi-F2	golani-F5	gosiha-F5	matuna-F1	lagaha-F5
gini'-F2	geha-F1	gaha-F5	gaha-F4	nagami-F5	gimisi-F2	golani-F5	gosiha-F5	matuna-F1	lagaha-F5
naga'-F1	lata'-F1	gaha-F6	gaha-F4	nagami-F5	gimisi-F3	golani-F6	gosiha-F6	matuna-F3	lagaha-F6
naga'-F1	lata'-F2	gaha-F6	gaha-F4	nagami-F6	gimisi-F3	golani-F6	gosiha-F6	matuna-F3	lagaha-F6
zogo´-F2	lata'-F5	gahu-F1	gako'-F1	nagami-F6	gimisi-F4	helisa-F1	numuni-F6	matuna-F6	matuna-F2
zogo´-F2	lata'-F5	gahu-F1	gako'-F1	nagami-F6	gimisi-F4	helisa-F1	numuni-F6	matuna-F6	matuna-F2
	lata'-F6	gahu-F1	gako'-F2	nagami-F6	golani-F3	helisa-F6	zopova-F2	izegipa-F1	matuna-F4
	lata'-F6	gahu-F1	gako'-F2	numuni-F1	golani-F3	helisa-F6		izegipa-F1	matuna-F4
	zogo'-F1	gahu-F2	gako'-F3	numuni-F1	golani-F4	helisa-F6			izegipa-F2
	zogo'-F1	gahu-F2	gako'-F3	numuni-F2	golani-F4	helisa-F6			izegipa-F2
	zogo´-F4	gahu-F3	gako'-F4	numuni-F2	gosiha-F3	luhusa-F1			izegipa-F5
	zogo´-F4	gahu-F3	gako'-F4	numuni-F4	gosiha-F3	muluna-F1			izegipa-F5
	zogo´-F5	gahu-F4	gala-F2	numuni-F4	gosiha-F4	muluna-F1			izegipa-F6
	zogo´-F5	gahu-F4	gala-F2	numuni-F5	gosiha-F4	muluna-F2			izegipa-F6

Figure 6: Classification of Alekano words from six sentence frames; we have turned on the display of frame identifiers

Just as we classified the forms according to sentence frames, we can classify them according to speakers. As we have seen, the third Praat tier contains a speaker identifier. Such labels should span every frame that was produced by a given speaker. In our Alekano example, the speaker identifier spans the entire file, and we have three separate files, one per speaker. These can all be loaded at once, and the speaker id (e.g. *BK*) can be displayed alongside each form (e.g. As *gosiha-BK* or *gosiha-F4-BK*). Any systematic difference between speakers should manifest itself in the pattern of speaker identifiers and column labels.

To the system, the frame and speaker labels are just arbitrary categories that can be used for dividing up the data. We can add more dimensions to our labels. For example, we could cross-classify all forms for syllable weight, vowel height, and onset laryngealization. The Praat label would consist of colon-separated fields, e.g. *gahu:LH:light:low:no* (and so tone is in position 1, syllable weight is in position 2, and so forth). These extra fields can be created inside Toney by selecting a new "value position", and then populating clusters and labelling them.

The same method can be used to break a melody into its components. Thus, instead of classifying bisyllabic forms into one of *LL*, *LH*, *HL*, *HH*, we could establish two orthogonal categories, one for the first syllable and one for the second. Now, the user's classification task would consist of making two independent judgements, one per syllable.

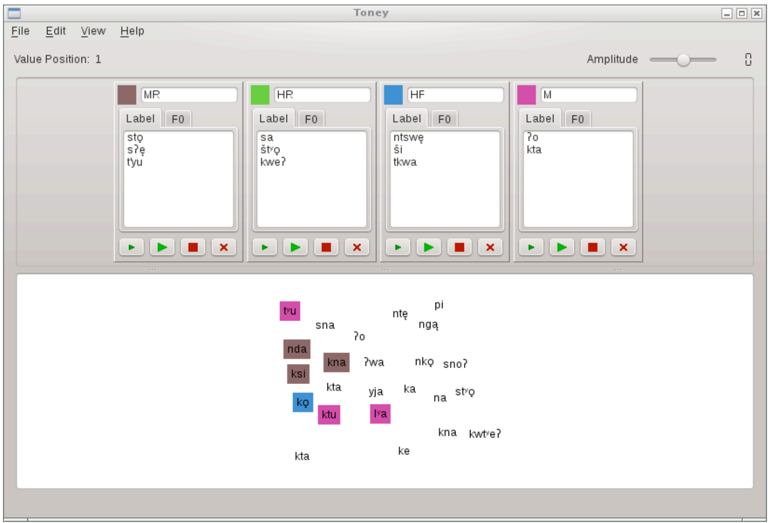
5. Automatic Classification

The automatic transcription of tone would seem to be an ideal way to avoid the difficulties of early elicitation, or the slow process of tone transcription. This prospect is as remote as ever, but there is some hope that we might be able to partially automate the process of classifying tones or tone melodies. The goal is to exploit the categories that have been established by the user and create a *model* for each tone melody, then detect uncategorized items that fit the model. For example, we could consider the F_0 contour for the mid tones shown in Figure 5, and then look for uncategorized items that are similar to these.

However, this approach is fraught with difficulties. For example, a male speaker will usually speak at a lower pitch (and thus with a lower F_0 contour) than a female, or the F0 contour of a syllable placed at a later position in a sentence may be affected by the tendency of the voice to fall in pitch through a sentence, or there may be local tonal coarticulation effects. Normalisation involves preprocessing of the data, reducing these effects, before studying the resultant contours in an attempt to cluster them into tones (Liberman 2010).

We employ partial least squares regression (PLS, Haenlein and Kaplan 2004) for automatically classifying tones. PLS transforms data by projecting it onto new axes (the so-called "latent variables"), with optional dimensionality reduction to reduce the chance of overfitting and to increase the speed of classification. PLS works by maximizing the covariance between the F_0 contour and the other factors (sentence position, speaker gender, tonal environment, etc) and the "response variables" (i.e. the tone labels). Dimensions of variability that are not relevant to tone classification are ignored, and so it should not be necessary to perform manual normalisation (Liberman 2010). We expect that the latent variables will correspond to the contribution of each factor, and give a better prediction for tone labels. The dimensions are not required to be orthogonal, and this may be useful when dealing with linearly dependent factors, such as when a lower pitch may be the result of a low tone, an environment containing low tones, depressor consonants, a male speaker, or a phrase-final position.

Toney uses PLS to generate tone classifications as shown in Figure 7. The classifications are presented as suggestions so that the user retains control of the process. Each time the user modifies a cluster by moving an item, the suggestions are updated.



Value Position: 1

MR: stǫ, sʔę, tʸu

HR: sa, štʸǫ, kweʔ

HF: ntswę, ši, tkwa

M: ʔo, kta

Figure 7: Automatically-generated suggestions for classification

6. Discussion and Conclusion

In their presentation at the Berkeley Tone Workshop in February 2011, Woodbury and Cruz demonstrated an approach to elicitation and transcription based on spreadsheets (Cruz and Woodbury 2011). Participants were given a hardcopy spreadsheet with one segmentally transcribed lexeme per row. Each row was numbered. Participants listened while the Chatino scholar (Cruz) produced each form and independently wrote down our tone transcriptions. After all forms were transcribed, we spent the bulk of the time asking Cruz to produce items in succession, by calling out pairs of row numbers, usually non-adjacent in the spreadsheet. After each such test, individuals would decide whether the pair should be grouped together, as having the "same" F_0 contour. The key insight to emerge from this activity was that early tone transcription is a classification task, an insight that underpins the work that has been reported here.

As we have observed, early elicitation of tone data is often difficult, thanks to a large number of influences on the F_0 contour that work to obscure the underlying tonal contrasts. We have developed software that is designed to fit into this niche of early elicitation, and help a linguist to identify the linguistically salient contrasts and annotate them consistently in extended recordings. Key features of this approach are as follows:

Collocation: Items that were widely separated in a field recording can be brought together, making it easy to check that they are transcribed appropriately. All items with the same transcription can be cross-checked, and any items that do not follow the pattern stand out and can be corrected right away.

Ear training: The software makes it easy for users to listen through lists of items having the same or similar tone melody. Similarly, a non-linguist native speaker can be alerted to the tonal contrasts of his/her language and will hopefully learn to alert the linguist to new contrasts, or to non-contrasts. Unlike a native speaker, the software does not tire of repeating the same set of forms over and over again.

Progressive elicitation: files from a series of elicitation sessions can be loaded at the same time, facilitating the growth of the collection over time without any need to re-record items. When an item is reclassified, its label is saved back to the file it came from. When a class label is modified, all items in that class are relabelled in all corresponding files.

Primary documentation: The source audio could contain primary documentation instead of controlled elicitation: the only requirement is for individual forms to be annotated with an optional context window.

Words and frames: We can listen to words with or without the surrounding sentence frame.

Audio annotation: The tone labels are stored as annotations of one or more original recordings, rather than a separate collection of audio clips that is disconnected from its source. If the segmental transcription of a word needs to be changed, this only needs to be done once, at the place where the word is located in the file. Similarly, if the tonal transcription of the word is changed in the Praat file, the word will be assigned to this new tonal category inside Toney. The Praat file can have any number of extra tiers, corresponding to other kinds of annotations required by the user, and these are left untouched when Toney saves tone labels back to the Praat file.

Automatic classification: Once some words have been categorized, the system models the tone contour of each category, with an emphasis on what distinguishes it from the other categories. These suggestions may speed up the classification work, and may suggest items that have been mis-classified.

Acknowledgements

We are grateful to Emiliana Cruz and Tony Woodbury for providing the Chatino data. The first prototype of this software was developed by X. The software and data samples are available for download from http://lp20.org/toney/. This research has been supported by grant XYZ.

References

Aston, John A.D., Chiou, Jeng-Min, and Evans, Jonathan P. (2010). Linguistic pitch analysis using functional principal component mixed effect models. *Journal of the Royal Statistical Society: C*, 59 (2): 297–317.

Cruz, Emiliana and Tony Woodbury. (2011). *Finding a way into a family of tone languages: The story of the Chatino Language Documentation Project (2003-)*. Presentation at the Berkeley Tone Workshop. http://www.prosodicsystems.org/workshop1/files/EasternChatinoToneExercise-Distribution.zip

Haenlein, Michael and Andreas M. Kaplan. (2004). A beginner's guide to partial least squares analysis, *Understanding Statistics* 3, 283–297.

Hyman, Larry M (2001). Fieldwork as a state of mind. In Paul Newman and Martha Ratliff (eds). *Linguistic Fieldwork*. Cambridge University Press.

Liberman, Mark (2010). COGS-501: FDA Homework #2
http://www.ling.upenn.edu/courses/Fall_2010/cogs501/FDA_HW2.html (retrieved 18 April 2013).