DAPPr: A (semi-)automated tool for pitch annotation

Emily Grabowski and Laura McPherson

UC Berkeley, Dartmouth College
emily_grabowski@berkeley.edu, Laura.E.McPherson@dartmouth.edu

ABSTRACT

This paper describes a novel computational toolkit for tonal analysis: DAPPr (Discrete Annotations for Pitch and Prosody). In particular, in this paper, we describe a first-pass method for automatically extracting target segments for analysis using a trained deep learning neural network. This automatic segment extractor is designed to work language-independently with minimal or zero training data. Together with the already-implemented DAPPr suite, these tools produce discrete pitch annotations that are consistent, objective, and compatible with other techniques and tools for analysis. While this tool was designed with tone languages in mind, the segment extraction and pitch analysis methods are relevant to the study of prosody more generally or even simply for automatic vowel detection. The aim of this analytical suite is to promote the inclusion of objective, replicable pitch data in documentary, descriptive, or theoretical materials.

Keywords: Tone, prosody, pitch, language documentation

1. INTRODUCTION

To date, it remains an open question how best to include pitch information in linguistic materials. For tone languages, the standard is to annotate data with more or less phonemic tone markings (diacritics, numerals, IPA tone levels, etc.). But, as recently highlighted by Remijsen [15], tone marking is not pre-theoretical; it is an analytical choice, which can obscure the surface realization of tones, or worse, may not be an accurate representation of the data. Similar points have been raised for intonation [9], where it is even less standard to include any sort of prosodic markings in most linguistic representations (unless focusing specifically on prosody). Worse still, many linguists are still uncomfortable with tone [13] and thus may exclude tone marking altogether, rendering the materials of little use to future researchers interested in the prosodic system. Past work on computational methods in linguistic description has also argued for the utility of including such a level of annotation in documentary materials, both to assist current analysis and to make the materials maximally useful for future research [7].

In light of this situation, we find a need for a standard method of including representations of phonetic pitch information in linguistic materials. We turn to discrete annotations as a way to capture broad phonetic information, much in the way of broad phonetic segmental annotations.

In making the analysis of pitch information more accessible, we find that it is essential to also make data processing itself accessible. To this end, we have begun development of a module to accompany the main DAPPr analysis software to assist in pre-processing the data. In this paper, we present the first iteration of the module, and find promising results.

To address these issues, we are developing a computational toolkit, DAPPr (Discrete Annotations of Pitch and Prosody). DAPPr takes as input a recording and outputs normalized annotations that can serve as an intermediate between raw phonetic information (i.e. f0 in Hz) and a phonemic analysis. The annotations created by DAPPr are designed to create an objective, replicable, and digital version of the system of dashes often found as a descriptive lingua franca for the surface realization of tone. It is designed to be used by anyone, regardless of technical training or familiarity with tone and prosody. In the process of producing discrete pitch levels, the toolkit also implements a range of related easy-to-use functions, including f0 extraction, correction, and normalization, duration measurements, and logging of an individual's pitch range across recordings, all functions that may be useful in a variety of research contexts. In this paper, we summarize the functionality of DAPPr for discrete pitch annotation, and will describe the development of automatic vowel extraction (AVE) to automatically pre-process data.

We first briefly address previous computational approaches to pitch annotation and vowel extraction in §2, then turn to a basic overview of the DAPPr workflow. §4 lays out the deep learning neural network for AVE, and in §5 we conclude.

2. PREVIOUS APPROACHES

Previous work in automatic tone annotation has primarily focused on identifying phonemic tone categories. Many methods have been focused on automating phonemic transcription for languages with known tone systems. These have included Hidden Markov Models [4][11][20][21], neural networks [1], and clustering [5].

Methods for analysis of unknown tone systems are less common. Language-independent clustering is implemented in the software Toney [3]. This tool aids the user in grouping perceptually similar tones together with the goal of faster identification of phonemic categories. However, it is still focused on speeding phonemic analysis, and less on preserving pitch information for future research. We aim to fill this gap with DAPPr.

Automatic identification of target sequences has also been addressed in previous work. Projects like ProsodyLab [6], Montreal Forced Aligner [12] and FAVE [16] have all made strides towards streamlining data processing. However, forced alignment typically requires significant training data for a specific language and often requires full, accurate transcriptions. Although the ideal case is to be able to exhaustively segment recordings, the amount of pre-processing necessary for these techniques can be prohibitive, especially for low-resource languages.

Previous language-independent implementations such as {Deep} Phonetics Tools [18], a deep neural network for vowel extraction, perform at a high level of accuracy. However, it also requires input segmented into CVC sequences, which does not cover the kind of data typically used in pitch extraction and analysis.

Thus, we identify a need for an automatic vowel extractor that is language-independent, easy to use, and can be paired with the pitch extraction algorithm to create completely automated pitch annotations.

3. PITCH EXTRACTION, NORMALIZATION, AND DISCRETIZATION

DAPPr is implemented in the open-source programming language Python, with additional support from Praat and Python packages scikit-learn [13] and Parselmouth [10]. The tool is equipped with a simple graphical user interface, making it accessible without the need to interact directly with code.

The pitch extraction tool takes as input an audio file (in .wav format) and accompanying TextGrid annotated to indicate target segments (generally vowels) for analysis. F0 measurements are extracted via a Praat script every 10ms throughout the segment.

A major benefit of DAPPr is that it will not only extract f0 but also apply algorithms to do some automatic cleaning for failure of the pitch extraction algorithm, with the aim of retaining as much data as possible for the normalization steps.

After the data are cleaned, the f0 values undergo normalization. We follow a widely practiced normalization procedure and normalize f0 to semitones [2][8][17]. We choose here to normalize to a speaker-specific f0, the mean of the speaker's range. This mean is tracked across multiple recordings to increase accuracy in normalization.

Finally, DAPPr discretizes the pitch data. This step considers all f0 information from all recordings for a given speaker and uses as a maximum and minimum value the 99th and 1st percentile of the speaker's range to reduce the effect of any remaining outliers. The speaker's range after normalization is divided into equal bins, or levels, the number of which can be set by the user as a parameter of the tool. The levels are labeled numerically such that 1 refers to the lowest level. Each token is sampled at 2-3 points, depending on the needs of the researcher, and each sample is then assigned to the appropriate level.

DAPPr outputs include tab-delimited text files containing all raw and normalized measurements, as well as TextGrids with a new tier of discrete level annotations, which can also be imported in ELAN and merged with other layers of annotation.

4. AUTOMATIC VOWEL EXTRACTION

4.1. Methodology

In this section, we present a tool for automatic vowel extraction in data processing. We approach the automatic vowel extraction tool with the following criteria in mind: it must be language-independent, require minimal, if any, training data, and be able to identify segments of interest for pitch analysis. Crucially, identification of vowel quality is not necessary for pitch-extraction, and since vowel quality annotations are to some degree language-dependent, we choose to simplify vowel extraction to a binary classification task: identification of whether the given acoustic material belongs to a vowel or not.

Our implementation takes as input an audio file with no annotations and produces a TextGrid with target segments marked as 'V'. The results of this can be inspected or fed into the main DAPPr analysis suite.

Training data totalling 130 minutes included eleven languages and 68 speakers. Several languages were drawn from the UCLA Voice Quality project [19], a dataset including wordlists recorded in a variety of languages by multiple speakers. The languages selected from this data set, Santiago Matatlan Zapotec, San Juan Guelavia Zapotec, Luchun, Yi, Bo, Black Miao, and Mazatec, were chosen because they were annotated to the level of detail required in the analysis of the tool. We supplement this data with field recordings in Seenu, Santo Domingo Albarradas Zapotec, Teochew, and Tommo So.

Thus the training dataset covers a large number of speakers from typologically different languages, and from different annotators. The purpose of these choices is to allow the model to generalize across different data to create a model that is not overly influenced by one subset of the data.

The training set is annotated such that vowels are marked as 1 and non-vowels as 0, including consonants, silence, and non-speech noise. The training data are featurized using the Praat algorithm to generate spectral coefficients and intensity measures via the Parselmouth interface. These data are then merged with human-made TextGrids delineating the boundaries of the target vowels to create a training set.

The model itself is a multilayer perceptron that is trained on the data described above. The model is then presented with data outside of the training set and predicts the locations of the vowels. The output of the model is then smoothed to reduce noise, and the results of the module are written to a TextGrid, where 'V' marks the locations identified as containing a vowel, and blank annotations in other intervals. This TextGrid can be directly input into the pitch analysis module of the DAPPr tool, resulting in fully automated pitch annotations that can be incorporated with other information.

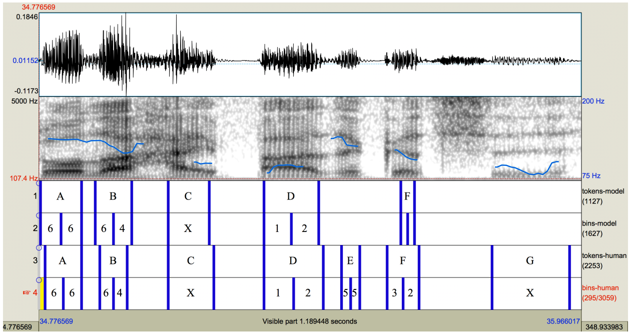
Because the training data does not require any phonological information such as the identity of the segments, training data can be made before phonological analysis is complete, allowing for use of DAPPr in early-stage linguistic description and documentation.

4.2. Model Results

In testing the model, we ran the tool on languages that were outside of the training dataset. For the purposes of qualitatively analysing the model's predictions, we take as a case study Mandarin, specifically a 350 sec recording of a wordlist elicitation. For Mandarin, we had human annotations for all of the vowels in the recording.

Figure 1 compares a representative DAPPr output for the model-identified vowels (top) and the human-identified vowels (bottom) for an utterance in Mandarin Chinese. This example shows the possible outcomes of using this tool in conjunction with DAPPr, and comparing the DAPPr annotation gives good insight into the quality of the vowel annotation for these purposes. From left to right, segments A, B, C and D are captured in both tools, although the start and end points for the model-predicted vowels are slightly different in some cases (around 10 ms). The DAPPr output for both are also the same. For the segment labelled C, DAPPr excluded the segment from pitch analysis (labelled as X) in both models because of poor pitch tracking.

Figure 1: DAPPr output for model-identified vs. human-identified vowels. Each vowel label is replaced with a capital letter to allow for easy comparison.



Segments E, F, and G demonstrate divergent behaviour between the model and human annotator. In segment E, the model failed to pick up on a segment that was annotated by a human annotator. In segment F, the model identified a segment with a start time that is 30 ms away from that of the human annotator. In addition, the resulting DAPPr annotations for the model output are (3,3) as opposed to (3,2) for the span annotated by a human. Finally, for G, the model failed to identify any vowel at all. However, the vowel identified in the human output is excluded in the DAPPr annotations, so there is no loss of information in this case.

Generally, in cases where the model fails to identify vowels, there appears to be some lack of robustness in the vowel, such as low amplitude or significant creak. Many of these factors also interfere with DAPPr predictions via poor pitch tracking. The correlation between poor vowel placement and poor pitch tracking means that despite some failures of the model in terms of absolute vowel identification, the output of the model is still largely usable for pitch analysis, since these spans would likely be excluded from DAPPr analysis anyway. Moreover, using automatic vowel extraction algorithms may also serve as a kind of filter that excludes acoustic information that would fail at a future analysis step. This approach would reduce reliance on later ad hoc data-cleaning algorithms in the data analysis pipeline.

All-in-all, there is relatively little information lost in using a generally trained vowel extraction model for data processing, and huge gains are made in the volume of data that can be processed.

The automatic vowel extraction module can be used in its general form for DAPPr-style tasks involving automatic data cleaning, pitch extraction, normalization and discretization. In addition, it is possible to tune the model to data from a specific language in order to create a within-language model.

To test the quality of the within-language model, we took a single recording (206 seconds) of a male speaker of Seenku reading target words in a frame sentence, and annotated it by hand. We then trained the model on different subsets of data (10%, 25%, 30%, 40%, and 80%) and tested it on three recordings: the same recording, a different recording done in the same style by the same speaker, and a similar recording by a different (female) speaker of Seenku. The results are given in Table 1.

In this case, we are interested in how closely a within-language model will match the human annotations. Thus, we measure performance using a metric called coverage, which is the percentage of segments annotated by the human researcher that have both a start and end timepoint within 20 ms of an interval in the model output.

Table 1: Degrees of coverage by amount of training data, within and across speakers.

% used	Same recording	Same speaker	Different speaker
10	0.84	0.64	0.59
25	0.9	0.76	0.57
30	0.91	0.80	0.57
40	0.91	0.83	0.57
80	0.91	0.83	0.57

We find that using as little as 10% of the training file (20 seconds of annotated data) can correctly identify around 85% of target vowels in the same recording, 65% in the same speaker, and 59% in a speaker of the language of the other gender. Increasing the proportion of training data improves the within-speaker results until around 40% (80 seconds of annotated speech) and gives a slight drop in performance for the other speaker. Thus, with this tool, it appears that even a small amount of training data can be used to bootstrap annotation of vowels within a project. While this requires some preparation of the data, it has the potential to reduce the amount of time necessary to create high-fidelity annotations within a set of similar recordings, a use that may prove helpful in a range of research contexts.

Both uses of the vowel extraction module are beneficial for certain tasks. The general model can be distributed pre-trained and represents ease of use, generalizability to a large number of languages, and data-cleaning properties. In addition, the overall quality of the general model is sufficient for, at the very least, broad phonetic analysis such as that in DAPPr. A within-language model requires some training data, which can be used to iteratively bootstrap a larger dataset for better models. However, a within-language model also provides a high level of control and the potential to fine-tune the model to specific desired properties of the language, with the trade-off of a higher cost to using and training the model.

5. CONCLUSION

In this paper, we have presented an automated workflow for discrete pitch level annotations that incorporates automatic vowel extraction. We find that even with a simple DNN, it is possible to vastly reduce the amount of human effort necessary to annotate recordings with pitch information. The preliminary results presented above demonstrate the potential of the tool in streamlining data processing for phonetic analysis of vowels, particularly pitch. Future directions for this project include: comparing machine learning algorithms, building a larger model, and using DAPPr in a wider range of languages and research questions.

Our ultimate goal in developing DAPPr is to remove both psychological and practical barriers for the incorporation of pitch information in linguistic materials. An automated system of discrete level annotations based directly on the acoustic signal promotes transparency and replicability of prosodic findings and helps ensure that important linguistic data is there for future generations.

7. REFERENCES

[1] Adams, O., Cohn, T., Neubig, G., Michaud, A. 2017. Phonemic transcription of low-resource tonal languages. *Proc. of Australian Language Technology Association Workshop,* 50–53..

[2] Baken, R. J. 1987. *Clinical measurement of speech and voice*. Boston: College Hill Press.

[3] Bird, S. 1994. Automated tone transcription, *Proc. of the First Meeting of the ACL Special*.

[4] Cooper-Leavitt, J. E. 2016. A computational classification of Thai lexical tones. *Journal of the Acoustical Society of Am*erica 139.

[5] Dockum, R. 2016. Tone analysis in Southeast Asia: computational modeling and traditional methods. *Talk presented at the 26th Annual Meeting of the Southeast Asian Linguistics Society* Manila.

[6] Gorman, K., Howell, J., Wagner, M. 2011. ProsodyLab-aligner: A tool for forced alignment of laboratory speech. *Journal of the Canadian Acoustic Association* 39, 192-193.

[7] Grabowski, E., McPherson, L. 2018. ATLAS (Automated Tone Level Annotation System): A

tonologist's and documentarian's toolkit . *Proc. 6th Int'l Symposium on Tonal Aspects of Languages* Berlin.

[8] Hart, J. T., Collier, R., Cohen, A. 1990. *A perceptual study of intonation: an experimental approach to speech melody*. Cambridge: Cambridge University Press.

[9] Hualde, J., Prieto, P. 2016. Towards an International Prosodic Alphabet IPrA," *Journal of Laboratory Phonology* 7.1.

[10] Jadoul, Y., Thompson, B., de Boer, B. 2018. Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics* 71, 1-15.

[11] Lee, T., Ching, P. C., Chan, L. W., Cheng, Y. H., Mak, B. 1995. Tone recognition of isolated Cantonese syllables. *IEEE Transactions on Speech and Audio Processing,* 3, 204–209.

[12] McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., Sonderegger, M. 2017. Montreal Forced Aligner: trainable text-speech alignment using Kaldi, *Proc. of the 18th Conference of the International Speech Communication Association*.

[13] McPherson, L. 2019. Tone: the present state and future potential. *Language: Commentaries* 95.1: e188-e192

[14] Pedgregosa, F., Varoquax, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Pasos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, 2825-2830.

[15] Remijsen, B. 2018. Investigating underdocumented tone systems. *Workshop given at the Annual Meeting of Phonology* UC San Diego.

[16] Rosenfelder, I., Fruehwald, J., Evanini, K., Seyfarth, S., Gorman, K., Prichard, H., Yuan, J. 2014. FAVE (Forced Alignment Vowel Extraction) Program Suite v1.2.2.

[17] Ross, E., Edmondson, J., Siebert, G. 1986. The effect of affect on various acoustic measures of prosody in tone and non-tone languages: a comparison based on computer analysis of voice. *Journal of Phonetics* 14, 283-302.

[18] Sonderegger, M., Keshet, J. 2012. Automatic measurement of voice onset time using discriminative structured predictions. *The Journal of the Acoustical Society of America* 132, 3965-3979.

[19] UCLA Voice Quality Project. http://www.phonetics.ucla.edu/voiceproject/voice.html.

[20] Xu, Y. 2004. Understanding tone from the perspective of production and perception. *Language and Linguistics* 5, 757-797.

[21] Yang, W. J., Lee, J. C., Chang, Y. C., Wang, H. C. 1988. Hidden Markov model for Mandarin lexical tone recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 36, 988–992.