PyToBI: a Toolkit for ToBI Labeling under Python

Mónica Domínguez[1], *Patrick Louis Rohrer*[1,2], *Juan Soler-Company*[1]

[1]University Pompeu Fabra, Barcelona, Spain
[2]Université de Nantes, CNRS, Laboratoire de Linguistique de Nantes, UMR-6310, France
monica.dominguez@upf.edu, patrick.rohrer@upf.edu, juan.soler@upf.edu

Abstract

PyToBI is introduced as a user-friendly toolkit for the automatic annotation of intonation contours using the Tones and Breaks Indexes convention, known as ToBI.

Index Terms: speech prosody, ToBI, automatic annotation, speech corpora, acoustic parameters, Praat, open-source software, Python

1. Introduction

Natural Language Processing (NLP) has got a wide variety of libraries implemented under Python, such as NLTK or Spacy. These libraries offer many functionalities that are relevant for speech research. However, applications dealing with the automatic annotation of speech prosody are often not prepared to be used within NLP pipelines under Python. In this scenario, PyToBI emerges as an accessible and ready-to-use toolkit to foster cross-disciplinary and multilingual research in the field of speech prosody for both developers and linguists.

PyToBI has thus been developed with two main objectives: (i) to automatically annotate prosodic contours using the ToBI convention [1], and (ii) to provide a data structure under Python, which will foster the usability of prosodic information in both speech and NLP pipelines and processes. Our toolkit uses Praat for the computation of acoustic parameters, and the algorithm for ToBI annotation is implemented in Python.

PyToBI[1] is an open-source application distributed under a GNU License v.3.0. The main contribution of PyToBI is the automatic annotation based on acoustic parameters, which is relevant for speeding up the compilation of prosody annotated corpora (often regarded as a time-consuming task and as well as a perception dependent process). The resulting output is presented as a TextGrid, and Python data structures allow exporting relevant prosodic information to any other format. All in all, this toolkit is meant to be instrumental in both linguistic and computational research contexts.

2. Methodology

The following subsections explain how PyToBI has been devised. Special attention is drawn to the labeling heuristics of tones and break indexes.

2.1. Processing Pipeline

The proposed processing pipeline is sketched in Figure 1. PyToBI requires as input a sound file in wav format and a TextGrid with the aligned words and phones. Several tools can be used for this alignment, such as EasyAlign [2], or Montreal Forced Aligner [3], as long as they output a TextGrid with two aligned tiers for words and phones, respectively. Montreal Forced Aligner has been used to align input files in our experiments.

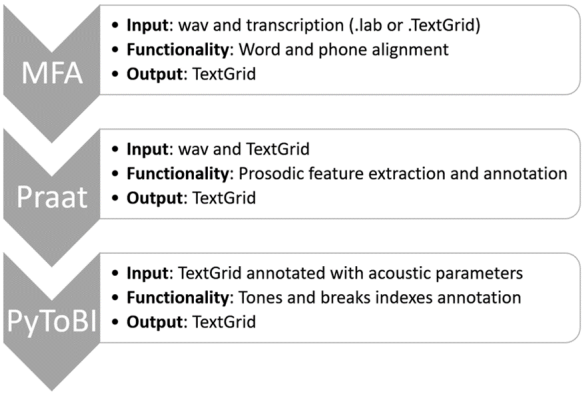

Figure 1: *Processing pipeline*

Praat needs to be installed locally in the PyToBI folder to perform the second step of the processing pipeline. The latest version of Praat can be downloaded from the original website[2]. PyToBI includes a bash file to call Praat and four Praat scripts included in the toolkit. These scripts perform different processes under Praat, namely:

1. Annotation of silences and intensity peaks.
2. Annotation of intensity valleys.
3. Word and phone export and annotation of prosodic parameters as features.

The next step of the pipeline is fully run in Python. The following functionalities are provided:

1. Python data structures created from TextGrid input.
2. Annotation of tones.
3. Annotation of breaks.
4. Conversion of the output structure to TextGrid format.

The following sections explain in more detail the Python functionalities and advantages of this methodology.

2.2. Python implementation

Once relevant acoustic parameters have been computed under Praat, the input TextGrid is transformed into Python data structures. Several classes have been defined for processing data from TextGrid, tiers, annotations and features. The *annotation* class represents both interval and point annotations and contains the start and end time, the head and a set of features. This functionality of offering parsable text labels for feature annotation

[1]https://github.com/monikaUPF/PyToBI

[2]http://www.fon.hum.uva.nl/praat/

is inspired in previous work on automatic prosody annotation by Domínguez et al. [4] and an online implementation of Praat called "Praat on the Web" [5].

Once the TextGrid class is instantiated with the path to a TextGrid file as parameter, all data structures are filled and made available for further processes.

These classes contain methods to access and create new tiers, add and get annotations from specific tiers and add and get features from an annotation. These functionalities can be used by the developer to add information to the data structures, and to export everything as a new updated TextGrid file or any other format.

2.3. Annotation of tones

We have envisaged the tones tier as a scaffolding of prosodic contours to be used off-the-shelf as an overall automatic annotation based on prosodic features. However, this automatic approach might also be used as a preliminary process to speed up manual annotation of dedicated ToBI catalogs for each language.

The extensive catalog of possible ToBI labels (different in each language) has been reduced to what we call a unified catalog that serves to characterize intonation in terms of falling-rising-flat contrasts within an intonational phrase (IP). This unified catalog of ToBI labels for pitch accents and boundary tones is summarized in Table 1. Further labels could be implemented given that a parametric description is provided.

Table 1: *Unified catalog of ToBI labels implemented in PyToBI.*

Contour	Pitch Accents	Boundary Tones
Rising	L*+H L+H*	LH- L-H%
Falling	H*+L	HL- H-L%
Flat	L* H* !H*	L- LL- L-L%

Z-scores of acoustic values across intonational phrases as well as intra-word acoustic parameters (like the tendency of F0 slope, intensity range or number of phones) are used for the prediction of ToBI labels in the tone tier. The resulting prominent point is annotated with a ToBI label and a *prominence score* is added as a feature to the corresponding word interval. Such score is the mean value of normalized F0, intensity and duration elements computed in each word.

2.4. Annotation of breaks

PyToBI detects breaks[3] from 1 to 4 and, in case a break type 3 or 4 is detected, the corresponding boundary tone is labeled in the tone tier. Phonemic information is used in this module to detect whether the last phoneme belongs to the subgroup of either fricatives or unvoiced consonants. Such information is relevant for the detection of breaks of type 2.

3. Evaluation

A preliminary evaluation has been run to assess the system performance. An excerpt from a TED talk has been annotated manually by two expert linguists. A total of 830 words were annotated with tones and break indexes (approximately, 5 minutes of speech by one male speaker of American English). The tone tier of each annotator included a total of 459 and 473 labels respectively.

Two metrics are measured to assess the location precision and label matching. The location score counts as a match every time a pitch accent or a boundary tone is annotated in the same word. The label score measures whether the label assigned is exactly the same in each location. In the case of system performance, location and label scores (from 0 to 1) are computed when any annotator shows coincidence with the automatic annotation. Table 2 summarizes these two scores for both inter-annotator agreement and system performance .

Table 2: *Inter-annotator agreement and system performance scores.*

	Score	Tones	Breaks
Inter-annotator Agreement	**Location**	0.91	1.00
	Label	0.78	0.85
System Performance	**Location**	0.77	0.97
	Label	0.47	0.90

4. Conclusions and Future work

PyToBI showcases the capabilities of automatically annotating prosody contours with the ToBI convention based on an algorithmic approach. The authors are currently working on a thorough evaluation of the system across several languages and a comparison of this tool with a baseline using existing prosody annotation tools, such as AuToBI (for English) [6] and ANALOR (for French) [7].

5. Acknowledgements

This project has received partial funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 786731.

6. References

[1] K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg, "ToBI: A standard for labeling English prosody," in *Proceedings of Interspeech*, Makuhari, Japan, 2010, pp. 146–149.

[2] J. Goldman, "Easyalign: An automatic phonetic alignment tool under praat," in *Proceedings of the Interspeech*, 2011, pp. 3233–3236.

[3] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: trainable text-speech alignment using kaldi," 2017.

[4] M. Domínguez, M. Farrús, and L. Wanner, "An automatic prosody tagger for spontaneous speech," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, Osaka, Japan, 2016, pp. 377–387.

[5] M. Domínguez, I. Latorre, M. Farrús, J. Codina, and L. Wanner, "Praat on the web: An upgrade of praat for semi-automatic speech annotation," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, Osaka, Japan, 2016, pp. 218–222.

[6] A. Rosenberg, "AutoBI - A tool for automatic ToBI annotation," in *Proceedings of Interspeech*, Makuhari, Japan, 2010, pp. 146–149.

[7] M. Avanzi, A. Lacheret-Dujour, and B. Victorri, "ANALOR. a tool for semi-automatic annotation of french prosodic structure," in *Proceedings of the International Conference on Speech Prosody*, April 2008, pp. 119–122.

[3] Breaks of type 0 are not being annotated due to the inherent difficulty in automatically detecting phonetic events related to deletion and assimilation processes in continuous speech.