ANALYSING SPEECH DATA WITH SATKIT

Pertti Palo[1*], Scott R. Moisik[2], and Matthew Faytak[3]

[1]Indiana University, Bloomington, [2]Nanyang Technological University, Singapore,

[3]University at Buffalo, Buffalo, NY

[*]pertti.palo@taurlin.org

ABSTRACT

A great variety of tools are available for analysing speech data, but few permit simultaneous annotation of audio, articulatory and other modalities. Being able to view several data modalities is key when analysing speech production data. The Speech Articulation ToolKIT (SATKIT) is a free open source tool for analysing and annotating speech data which aims to fill this gap. SATKIT has a command-line interface for batch processing data, and a Praat-like GUI for annotating articulatory and audio data. SATKIT has been written in Python, which it also uses as its scripting language. Modifying, extending, and connecting to other systems uses a well documented Python API. SATKIT already includes algorithms which are aimed at making analysing image based data – such as ultrasound – easier. Using SATKIT does not require any paid licenses, enhancing access to this type of annotation. This paper provides an in-depth introduction to this new tool.

Keywords: Articulatory and acoustic annotation, speech segmentation, free open source software

1. INTRODUCTION

Today, there is a great variety of tools available for analysing speech data. To mention a few, we have Praat [1] for annotating sound, and for example for ultrasound and tongue contours we have AAA [2] (which works with other modalities too), as well as GetContours [3], SLURP in 2D and 3D [4, 5], UltraTrace [6], WASL [7] among others. The list can be continued with various tools for analysing EMA data, and topped off with dedicated software from many equipment manufacturers to analyse data from their machines. Some of the availble tools and programs are open source, but none have the combination of being open source, free to use (i.e. no costly licenses required), with an easy-to-learn scripting language (such as Python), and dimensionality reduction and time domain annotation of the data.

It would seem that we should have a tool available for easily segmenting not just audio data, but articulatory modalities too. And that such a tool would be easily scriptable, mesh well with other analysis tools, and be extendable as well as modifiable. Most software is not as handy as Praat is with segmentation, but unfortunately Praat really only analyses audio data, not to mention that scripting Praat can be challenging, and integrating with other tools like Python or R needs a step of first saving the data into files such as TextGrids and then reading them with the system used to process them further. Good packages exist for using this route [8], but closer integration is still missing. And on the other hand, many of the tools available for processing articulatory data are either proprietary software (AAA) or rely on such - most prominently on MATLAB (GetContours, SLURP, WASL).

2. INTRODUCING SATKIT

The Speech Articulation Toolkit (SATKIT) is a free open source tool written in Python for analysing, visualising, and annotating speech data. Our aim is to provide a tool, which is intuitive to use via GUI, command line interface, and scripting interface alike.

SATKIT incorporates algorithms and metrics that are aimed at making analysing image based data – such as ultrasound – easier. It has a commandline interface for batch processing of data, and a Praat-like GUI for annotating articulatory and audio data.

SATKIT features a well documented Python API for extending and customising the toolkit. SATKIT has an object oriented structure which provides base classes for representing data, for writing new metrics to be applied to the data, and for representing annotations. The code is extensively commented and documented.

3. GUI

The SATKIT annotator GUI enables users to mark single point annotations on data, perform textgrid

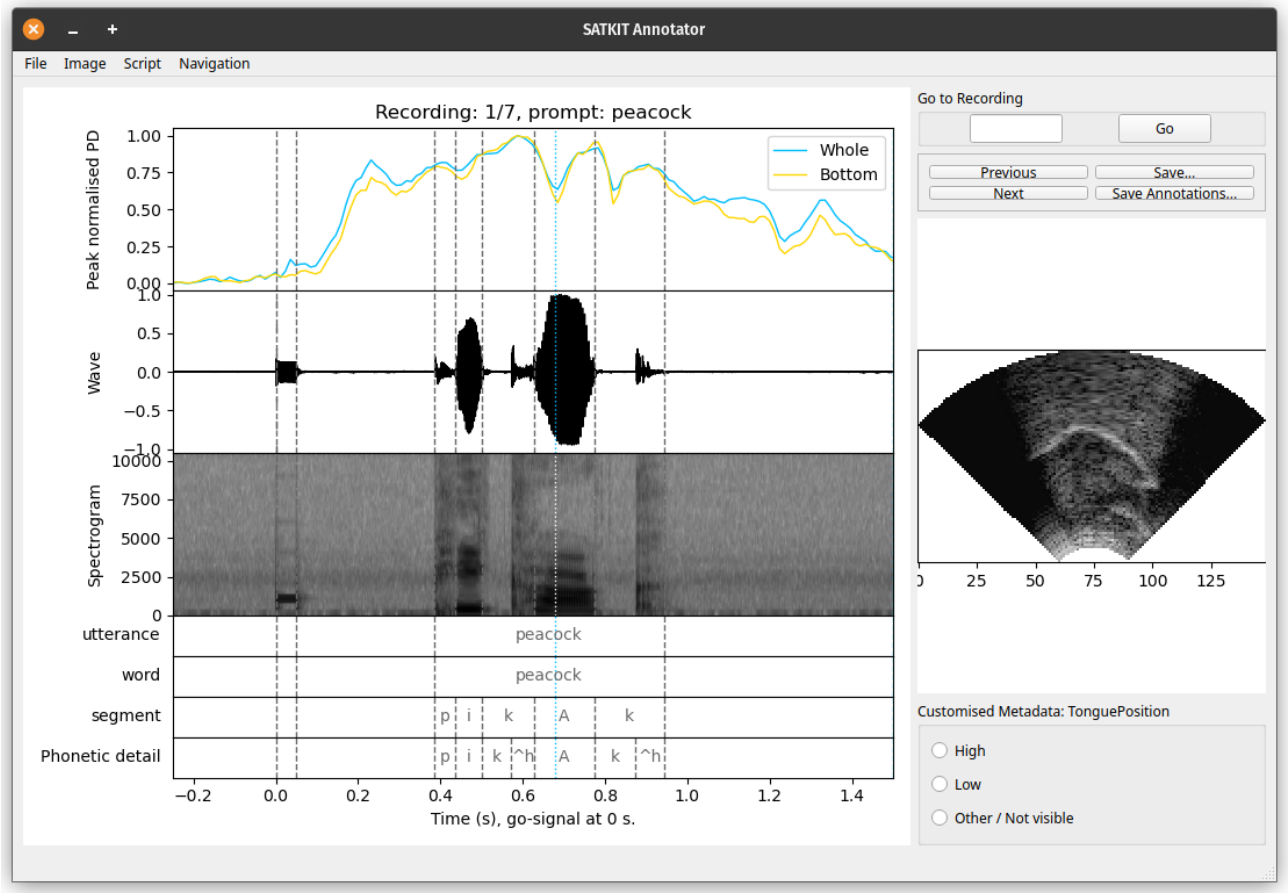

Figure 1: Annotating ultrasound data in SATKIT. Panels on the left from top to bottom are 1. peak normalised PD of both the whole raw ultrasound image and the bottom half, 2. waveform, 3. spectrogram and 4. acoustic boundaries read from a TextGrid. Right panels from top to bottom show widgets to switch between recordings and save data, the interpolated ultrasound frame corresponding to the cursor (dotted light blue vertical line) position within /a/, and a customisable metadata section here showing tongue position categories.

style annotations, and display of ultrasound frames at selected points. An example single point annotation with SATKIT is shown in Figure 1. The annotation GUI is designed for fast manual annotation of large numbers of recordings.

The initial TextGrid was populated with another python tool called Computer Assisted Segmentation Tool (CAST, available at https://github.com/giuthas/cast) [1].

4. DATA PROCESSING

4.1. Data formats

SATKIT can read sound from wav files, ultrasound tongue data exported from AAA (old and new metadata formats as well as splines), 3D ultrasound DICOM files, flow and sound data exported from EVA, avi videos, and Praat TextGrids. Internally data is represented with a layer of abstraction between import functions and the algorithms/metrics that are used to process and plot the data. This makes the list of supported formats readily expandable and the algorithm implementations data format agnostic.

4.2. Algorithms and metrics

SATKIT includes several ways to process data. Most of them provide ways of analysing complex data – such as ultrasound videos – in time without needing to examine the data frame-by-frame. SATKIT includes Pixel Difference (PD) for analysing over all change in an image sequence [9, 10], Optic Flow (OF) for analysing direction of local and/or global movement [11, 12, 13], and a combination of eigentongue analysis and discriminant analysis on ultrasound [14].

5. SATKIT CODEBASE

5.1. Design principles

SATKIT is built using object oriented programming. We aim for a class structure which is intuitive to use and easy to maintain: keep inheritance trees shallow, separate data storage and functionality when it makes sense. The core data classes represent organisational concepts which are likely to be mirrored by directories and files produced when recording data.

5.2. Core data structures

5.2.1. Recording

The Recording class (Fig. 2) represents a single trial from a single recording source. It has one or more Modalities such as MonoAudio and RawUltrasound. It also has metadata: participant, time of recording, etc. and a TextGrid [1, 15].

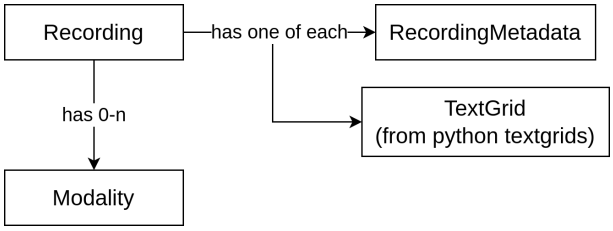

Figure 2: Recording class and its components.

5.2.2. Data Modalities

Different types of data are represented as Modalities derived from the base Modality class (Fig 3). The derived Modality classes provide a represantation of the specific qualities of a given data type.

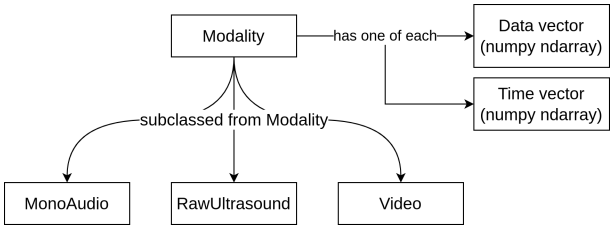

Figure 3: Components and example subclasses of Modality.

5.2.3. Derived Modalities

One of the main development goals SATKIT is to be able to easily run different algorithms on recorded data. The results of doing so are represented by derived Modalities (Fig 4). After creation, there is very little difference between the two Modality types, which means that algorithms and plotting will be as easy to run on derived data as on recorded data.

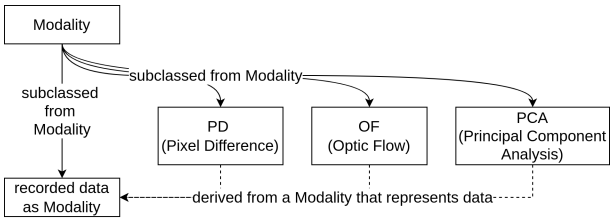

Figure 4: Derived data is represented by subclasses of Modality, which also contain the information on which other Modality class they were derived from.

6. INSTALLING

SATKIT is available on GitHub. The installation requires Python for running, and conda and mamba for managing the required packages. More detailed installation instructions are available in the repository on GitHub. After installing Python, the required python packages, and SATKIT either by forking/cloning from github, SATKIT can be started from the commandline. SATKIT runs on recent operating systems (Linux, MacOS, Windows).

7. ROADMAP BEYOND SATKIT 1.0

7.1. Expanding beyond a single session

Datasets are generally speaking more complex then just a single session and single speaker. To represent this, SATKIT is going to get classes for Datasets, Participants within a Dataset, and Sessions for Participants (Fig 5). The Trial class represents a multimodal recording that consists of one or more Recordings where each Recording represents data from a single data source such as AAA or an EMA machine.

7.2. Expanding to new Modalities

In general, one of our aims is to make extending SATKIT easy and to make it easy to use it with or as part of other tools. Adding new modalities – whether data modalities or new metrices represented by derived modalities – is going to be a routine thing with SATKIT. Writing a new data Modality for SATKIT is a matter of defining the Modality subclass and an importer to construct it. After doing

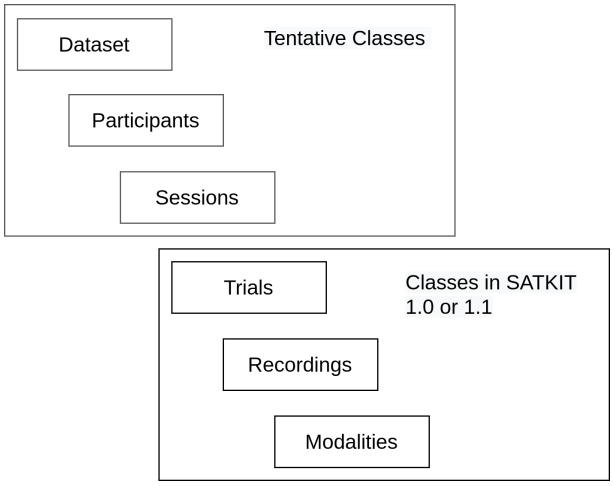

Figure 5: Hierarchy of database classes in SATKIT versions 1.0 and 1.1 and some tentative classes that potentially will become part of SATKIT in a later version.

so the algorithms and GUI will be automatically able to handle the new type of data.

While SATKIT does not generate tongue splines, it will able to process them by version 1.1. This will be done by implementing both a data Modality for splines and different derived Modalities for metrics from literature [16, 17, 18, 19]. Other data Modalities that could be implemented depending on user interest include MRI, EMA, and air flow data from EVA and other systems.

7.3. Other possible improvements

The GUI will definitely be streamlined over time as experience of using it points to good changes. Already at this stage it is clear that an undo function (with preferably an undo stack of more than one command) is a necessity. Adding Praat-style support of annotating data with IPA is easily supported by the underlying data classes but requires a bit more work on the GUI. Having an editable list of hotkeys allowing users to customise how they work with the GUI is another likely improvement in the not-too-distant future.

Currently SATKIT is installed as a Python package and a script. Having a stand-alone executable would be desirable, but selecting the best way to do this to provide multiplatform support requires careful consideration and testing.

8. CONCLUSION

SATKIT is a useful, easy-to-use, freely available tool. It is readily expandable to include new metrics on various data modalities adding to the existing ones for raw ultrasound and audio. We are interested in growing the user community and integrating with other tools and adding new functionality, while keeping the API robust.

9. REFERENCES

[1] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [computer program]," 2022, version 6.2.12, retrieved 14 April 2022 from http://www.praat.org/.

[2] *Articulate Assistant Advanced User Guide: Version 2.14*, Edinburgh, UK: Articulate Instruments Ltd, 2012.

[3] M. Tiede, "How to "getcontours" from ultrasound imaging," in *Proceedings of Ultrafest IX*, Bloomington, Indiana, 2020.

[4] S. Ghrenassia, L. Ménard, and C. Laporte, "Interactive segmentation of tongue contours in ultrasound video sequences using quality maps," in *SPIE Medical Imaging*, San Diego, United States, 2014.

[5] E. M. V. N. Karthik, E. Karimi, S. M. Lulich, and C. Laporte, "Automatic tongue surface extraction from three-dimensional ultrasound vocal tract images," *Journal of the Acoustical Society of America*, vol. 147, no. 3, pp. 1623 – 1633, 2020.

[6] K. Murphy, N. Z. Stern, D. Swanson, C. Ho, and J. Washington, "Ultratrace: A free/open-source cross-platform tool for manual annotation of ultrasound tongue imaging data," in *Proceedings of Ultrafest IX*, Bloomington, Indiana, 2020.

[7] S. M. Lulich, K. H. Berkson, and K. de Jong, "Acquiring and visualizing 3D/4D ultrasound recordings of tongue motion," *Journal of Phonetics*, vol. 71, pp. 410 – 424, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0095447017301481

[8] T. Bořil and R. Skarnitzl, "Tools rpraat and mpraat," in *Text, Speech, and Dialogue: 19th International Conference, TSD 2016, Brno, Czech Republic, September 12-16, 2016, Proceedings*, P. Sojka, A. Horák, I. Kopeček, and K. Pala, Eds. Cham: Springer International Publishing, 2016, pp. 367–374. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-45510-5_42

[9] P. Palo, "Measuring pre-speech articulation," Ph.D. dissertation, Queen Margaret University, Edinburgh, 2019.

[10] ——, "Computer assisted segmentation of tongue ultrasound and lip videos," *Journal of the Canadian Acoustical Association*, vol. 49, no. 3, pp. 44 – 45, 2021.

[11] J. H. Esling and S. R. Moisik, "Laryngeal aperture in relation to larynx height change: An analysis using simultaneous laryngoscopy and laryngeal ultrasound," in *Rhythm, melody and harmony in speech: Studies in honour of Wiktor Jassem*, D. Gibbon, D. Hirst, and N. Campbell, Eds. Polskie Towarzystwo Fonetykzne Poznan, 2012, vol. 14/15, pp. 117 – 127.

[12] S. R. Moisik, H. Lin, and J. H. Esling, "A study

of laryngeal gestures in mandarin citation tones using simultaneous laryngoscopy and laryngeal ultrasound (sllus)," *Journal of the International Phonetic Association*, vol. 44, no. 1, pp. 21 – 58, 2014.

[13] D. P. Z. Poh and S. R. Moisik, "An acoustic and articulatory investigation of citation tones in singaporean mandarin using laryngeal ultrasound," in *Proceedings of the 19th International Congress of Phonetic Sciences, Melbourne, Australia*, S. Calhoun, P. Escudero, M. Tabain, and P. Warren, Eds., 2019.

[14] M. Faytak, S. Liu, and M. Sundara, "Nasal coda neutralization in shanghai mandarin: Articulatory and perceptual evidence," *Laboratory Phonology*, vol. 11, no. 1, 2020.

[15] T. Nieminen, "praat-textgrids – praat textgrid manipulation in python [python package]," 2022, version 1.4.0, available on PyPi and Github.

[16] K. M. Dawson, M. K. Tiede, and D. H. Whalen, "Methods for quantifying tongue shape and complexity using ultrasound imaging," *Clinical Linguistics & Phonetics*, vol. 30, no. 3 – 5, pp. 328 – 344, 2016.

[17] P. Palo, "Can we detect initiation of tongue internal changes before overt movement onset in ultrasound?" in *Proceedings of the 12th International Seminar on Speech Production (ISSP 2020)*, Online / New Haven, CT, 2020, pp. 242 – 245.

[18] N. Zharkova and N. Hewlett, "Measuring lingual coarticulation from midsagittal tongue contours: description and example calculations using English /t/ and /ɑ/," *Journal of Phonetics*, vol. 37, pp. 248 – 256, 2009.

[19] N. Zharkova, F. E. Gibbon, and W. J. Hardcastle, "Quantifying lingual coarticulation using ultrasound imaging data collected with and without head stabilisation," *Clinical Linguistics & Phonetics*, vol. 29, no. 4, pp. 249 – 265, 2015.

[1] While the initial TextGrid from CAST looks like forced alignment, it is both dumber and more ergonomic for purposes of human correction. The idea is to provide – with a good deal of configuration options – an evenly spaced phonological segmentation that is at a reasonable position within the recording for a human annotator to move the boundaries to their correct locations.