Journal of Phonetics 71 (2018) 1–15

Contents lists available at ScienceDirect

Journal of Phonetics

journal homepage: www.elsevier.com/locate/Phonetics

Special Issue: Emerging Data Analysis in Phonetic Sciences, eds. Roettger, Winter & Baayen

Introducing Parselmouth: A Python interface to Praat

Yannick Jadoul [a,b,*], Bill Thompson [c,a], Bart de Boer [a,c]

[a] Artificial Intelligence Lab Brussels, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Elsene, Belgium
[b] ADReM Research Group, University of Antwerp, Middelheimlaan 1, 2020 Antwerpen, Belgium
[c] Language and Cognition Department, Max Planck Institute for Psycholinguistics, Wundtlaan 1, Nijmegen 6525 XD, The Netherlands

ARTICLE INFO

Article history:
Received 15 September 2017
Received in revised form 16 April 2018
Accepted 4 July 2018

Keywords:
Praat
Python
Data analysis
Acoustics
Phonetics
Software

ABSTRACT

This paper introduces Parselmouth, an open-source Python library that facilitates access to core functionality of Praat in Python, in an efficient and programmer-friendly way. We introduce and motivate the package, and present simple usage examples. Specifically, we focus on applications in data visualisation, file manipulation, audio manipulation, statistical analysis, and integration of Parselmouth into a Python-based experimental design for automated, in-the-loop manipulation of acoustic data. Parselmouth is available at https://github.com/YannickJadoul/Parselmouth.

1. Introduction

Data analysis in the phonetic sciences routinely relies upon the functionality of Praat (Boersma, 2001; Boersma & Weenink, 2018), an extensive software package which has subserved the day-to-day activities of phoneticians for more than two decades. This paper introduces Parselmouth, an open-source Python library that exposes major functionality of Praat into Python. Two principal advantages result from this integration: (1) users of Praat may now benefit from the expressive power of a large-scale language like Python, and its expansive ecosystem of scientific and computational libraries; and (2) users of Python may access the many tools and utilities for sophisticated acoustic analysis that Praat provides. Parselmouth is currently available as version 0.3.0, for use with Windows, macOS, and Linux-based operating systems, for Python versions 2 and 3. The package is under active development by the first author of this article, and can be downloaded from https://github.com/YannickJadoul/Parselmouth. Basic speech analysis methods from Praat are already available, while all other algorithmic functionality of Praat can be called indirectly. We are hopeful that others in the community of speech scientists and engineers will wish to contribute to the development.

The remainder of this paper is organised as follows: Sections 1.1, 1.2, and 1.3 respectively give detailed background and motivation, compare Parselmouth to other software packages, and provide technical information about Parselmouth. Sections 1.4, 1.5, and 1.6 then present more practical information on the functionality of Parselmouth, how to install the Python library, and where to find its online documentation and further resources. Section 2 presents five usage examples, focusing on what we imagine to be some of the most recurrent technical challenges speech scientists are likely to face: idiosyncratic visualisation of acoustic data (Section 2.1); reading, writing, and manipulating batches of acoustic files and data frames (Section 2.2); manipulation of audio files along complex acoustic dimensions (Section 2.3); statistical analysis of the output of acoustic analyses (Section 2.4); and integration of automated acoustic analysis into experimental design (Section 2.5). These examples are intended to be illustrative of the principles behind the package, rather than exhaustive demonstrations of Parselmouth's potential use cases, which we expect to grow indefinitely with the advance of Python and the scientific creativity it facilitates. We summarise the motivation and examples and present concluding remarks in

* Corresponding author at: Artificial Intelligence Lab Brussels, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Elsene, Belgium.
E-mail address: Yannick.Jadoul@ai.vub.ac.be (Y. Jadoul).

https://doi.org/10.1016/j.wocn.2018.07.001
0095-4470/

Section 3, after which Section 3.1 closes the paper with a brief discussion of the future of Parselmouth.

At this point we wish to stress that Parselmouth is built on the vast Praat collection of source code: as such, we encourage twin citation of both Praat and Parselmouth whenever Parselmouth is used for scientific research.

1.1. Motivation

The Python programming language is rapidly becoming the *lingua franca* of scientific computing. Python is used and supported by an enormous community of scientists, researchers, and engineers whose workflows are continuously improving thanks to integration of diverse computational utilities in a single programming language. For many, including us, Python is the go-to toolbox for data manipulation and analysis. However, for contemporary *speech* scientists, researchers, and engineers, major portions of our day-to-day activities – specifically, analysis of acoustic data using Praat functionality – remain difficult or time-consuming in Python; the necessary functionality is often unavailable or dispersed over multiple unrelated and sometimes incompatible libraries. We began developing Parselmouth as a solution to this problem. Parselmouth is not a replacement for Praat: it is an additional interface to Praat, making Praat's functionality available in Python. We have three principal goals in mind: to allow experienced users of Praat to more efficiently integrate acoustic analysis with scientific tools available in Python but not in Praat; to provide access to Praat's functionality for users who are comfortable with Python but unfamiliar with Praat; and to simplify or optimise the workflow of any users who would simply rather work in a single language.

Python is often used as *glue language* for scientific workflows, drawing together the "scientific stack" in a collection of widely used, robust scientific libraries (e.g., *NumPy*, *SciPy*, *pandas*, *scikit-learn*, *matplotlib*, etc.; see https://scipy.org/about.html). As Python is designed as an extensible programming language and framework, its use extends across many domains, and even across other programming languages. Scientists using Python have access to, for example: advanced statistical modeling libraries and probabilistic programing frameworks such as *Statsmodels*,[1] *PyMC3*,[2] *Pyro*,[3] and *Edward*;[4] deep learning libraries like *TensorFlow*[5] or *PyTorch*;[6] *Jupyter* notebooks[7] (formerly IPython); experimentation packages such as *PsychoPy*[8] or *Dallinger*;[9] the *rpy2*[10] module that provides easy access to R functionality; and the official *'MATLAB Engine API for Python'*,[11] which integrates MATLAB into Python programs. More generally, just like Praat, Python has functionality for writing universal data exchange formats – built-in, such as comma-separated values (csv) or JavaScript Object Notation (JSON), or through external libraries, such as HDF5[12] or SQL databases[13,14] – which makes it possible and convenient to use Parselmouth to combine the functionality of Praat and these Python libraries with almost any other computational framework.

While choosing any particular language is to some extent an arbitrary choice, Python is a popular and high-level, yet fully-fledged programming language. Python not only accommodates quick scripting but also provides support for more complex programming paradigms and performant implementations of algorithms. While the Praat scripting language is suitable for automating repeated workflows and calculations within the context of Praat, we believe the use of Python and Parselmouth can be advantageous in a broader range of applications. Python implements general programming principles, including a full and generic type system with built-in types (i.e., lists, tuples, sets, dictionaries, ...) and custom classes. As a result, Python is well-suited to be used in a more programming-intensive context. In these cases, *integrated development environments* with e.g. syntax highlighting, autocompletion functionality, and debugging tools, can assist in the development process.

Python is also an accessible language, useful for writing simple scripts. Python is often taught to students at their first encounter with programming, sometimes even before a specialisation in phonetics brings them into contact with Praat. We believe that Parselmouth can be attractive to this group of users that are already familiar with programming or Python, but not with the Praat scripting language. Python is supported by a large community of users who have written up many solutions to specific programming problems and frequent Python errors – see, for example, StackOverflow.[15] Fewer people have the necessary experience with Praat to answer questions and solve problems concerning Praat scripts, and fewer resources and tutorials exist to learn the Praat scripting language than to learn Python.

Finally, the Python project and the available libraries are *modular*. They are specialised in one area of functionality (i.e., being a programming language, plotting graphs, handling data tables, performing statistical analyses, etc.), yet are designed to be used and combined in larger and more complex projects. With Parselmouth, we aim to add the option of using the highly advanced, specialised functionality from Praat in combination with the already existing libraries in Python in this same manner.

1.2. Relation to previous software

Parselmouth is not the first attempt to port Praat functionality into Python. Other packages exist, together offering a range of Praat functionality. However, the previous projects we are aware of are generally restricted in important ways that Parselmouth is not, technically speaking. We see the diversity of preceding projects as testament to a clear but unfulfilled demand

[1] http://www.statsmodels.org/
[2] http://docs.pymc.io/
[3] http://pyro.ai/
[4] http://edwardlib.org/
[5] https://www.tensorflow.org/
[6] http://pytorch.org/
[7] http://jupyter.org/
[8] http://www.psychopy.org/
[9] http://docs.dallinger.io/en/v3.4.1/
[10] https://rpy2.readthedocs.io/
[11] https://mathworks.com/help/matlab/matlab-engine-for-python.html
[12] http://docs.h5py.org/
[13] http://www.sqlalchemy.org/
[14] https://pandas.pydata.org/pandas-docs/stable/io.html#io-sql
[15] https://stackoverflow.com/questions/tagged/python