Working with CHAT transcripts in Python*

Jackson L. Lee, Ross Burkholder, Gallagher B. Flinn, and Emily R. Coppess
University of Chicago

January 28, 2016

Abstract

This report introduces the Python library PyLangAcq for working with CHAT transcription data in Python. The library interfaces with speech data transcribed in the CHAT format, which is adopted by the CHILDES database for child language development research. Built in a Python infrastructure, PyLangAcq has direct access to a multitude of computational and statistical tools for language acquisition research. As the CHAT format is also used for other speech transcription databases, PyLangAcq will be useful for researchers in other linguistically related fields such as conversational analysis, corpus linguistics, and clinical linguistics.

1 Introduction

Natural language data often come in the form of conversations transcribed in some specific format for the purposes of linguistic research and other domains that require a consistent representation of conversational data. A very commonly used format is the CHAT transcription format (MacWhinney 2000). CHAT (Codes for the Human Analysis of Transcripts) is the transcription format particularly developed for CHILDES (Child Language Data Exchange System) for language acquisition research. As CHAT is well documented and can have very rich annotations, it is also used more generally outside the field of language acquisition, for areas such as conversational analysis, corpus linguistics, and clinical linguistics.

Research using data in the CHAT format necessitates tools for extracting information and doing analysis in an efficient and automatic manner. This is particularly relevant for the computational modeling of language acquisition, a growing field of study across linguistics, psychology, and computer science (cf. Alishahi 2010; Villavicencio et al. 2013). The CHILDES project has the associated tool CLAN (Computerized Language Analysis), a widely used toolkit with a graphical user interface which facilitates both transcription and analysis of conversational linguistic data. As a standalone computer program, however, CLAN does not straightforwardly allow customized manipulation and analysis of CHAT transcripts that deviates from the functionalities directly provided by CLAN itself. To this end, a solution would be to come up with something that parses CHAT transcripts and allows researchers to devise any tools and programs for their purposes. For instance, it would be desirable to be able to parse CHAT data, perform computational and statistical analyses, as well as visualize data and results all in one single system.

*Technical Report, Department of Computer Science, University of Chicago
Corresponding author: Jackson L. Lee (jsllee@uchicago.edu)

Indeed, the Python-based NLTK (Natural Language Toolkit; Bird et al. 2009) has a CHILDES corpus reader (by Tomonori Nagano and Alexis Dimitriadis, presented as Nagano and Valian 2011) and, thanks to Python being a general-purpose programming language, this allows virtually anything to be done with the parsed data structure. There is, however, one crucial criterion if one would like to use NLTK to handle CHAT transcripts: NLTK currently requires the XML version (a mark-up schema devised by the CHILDES team) of the CHAT transcripts. Such a requirement adds an additional layer of work and effort, thereby increasing the chance of introducing errors in the workflow. Although CHILDES does provide tools that convert CHAT transcripts into their specified XML format, this requires that the CHAT format specifications and the associated tools (all updated from time to time) be mutually compatible, which could be overlooked in actual use and create confusion. Moreover, it is clear that human researchers work most comfortably and conveniently with CHAT transcripts directly, not with the derived XML version with rich mark-up language that is not intended to be handled by humans.

Given this background, there is a need for a general tool that parses CHAT transcripts and allows researchers to write their own scripts and programs to interact with the parsed data structure. In this report, we introduce the Python library PyLangAcq for exactly these purposes.[1] Our choice of programming language is due to the widespread use of Python in computational linguistics and natural language processing. PyLangAcq makes it possible that the great variety of machine learning as well as other computational and statistical tools available via Python can be used to model any phenomena of interest with respect to CHAT datasets. As the CHAT format is used for speech transcriptions more generally, PyLangAcq will be useful for researchers of many other linguistically related fields.

PyLangAcq is ever expanding and evolving, with its official detailed documentation hosted online and regularly updated (`http://pylangacq.org/`). At the time of writing, PyLangAcq is fully operational for parsing CHAT transcripts and extracting information of interest, including but not limited to the following:

- participants (e.g., `CHI` (target child), `MOT` (mother)) and their demographic information
- age (of the target child, most typically)
- transcriptions in various data structures
- word frequency information and ngrams
- word search and concordance
- dependency graphs (based on `%gra` tiers)
- standard language development measures such as type-token ratio (TTR), mean length of utterance (MLU), and index of productive syntax (IPSyn)

The reader is directed to the online documentation of the library for any of these items and more. They are the building blocks of advanced modules and functions currently being developed and added to the library.

In the rest of this report, we illustrate the use of PyLangAcq for measuring the mean length of utterance in morphemes (MLUm; section 2), studying bilingualism (section 3), and exploring phonological development (section 4).

[1] `http://pylangacq.org/`

2 The mean length of utterance in morphemes (MLUm)

We illustrate how the mean length of utterance in morphemes (MLUm) can be computed by PyLangAcq for some given CHILDES dataset in language acquisition research. Our example uses Eve's data from the Brown portion (Brown 1973) of CHILDES. As MLUm is often used as a measure of language development, we may ask if MLUm is correlated with age in Eve's data. Figure 1 shows the results:

Filename	Age (months)	MLUm
`eve01.cha`	18	2.267
`eve02.cha`	18	2.449
`eve03.cha`	19	2.763
`eve04.cha`	19	2.576
`eve05.cha`	20	2.859
`eve06.cha`	21	3.177
`eve07.cha`	21	3.123
`eve08.cha`	21	3.374
`eve09.cha`	22	3.818
`eve10.cha`	22	3.792
`eve11.cha`	23	3.866
`eve12.cha`	23	4.157
`eve13.cha`	24	4.239
`eve14.cha`	24	3.960
`eve15.cha`	25	4.450
`eve16.cha`	25	4.424
`eve17.cha`	26	4.466
`eve18.cha`	26	4.288
`eve19.cha`	27	4.348
`eve20.cha`	27	3.163

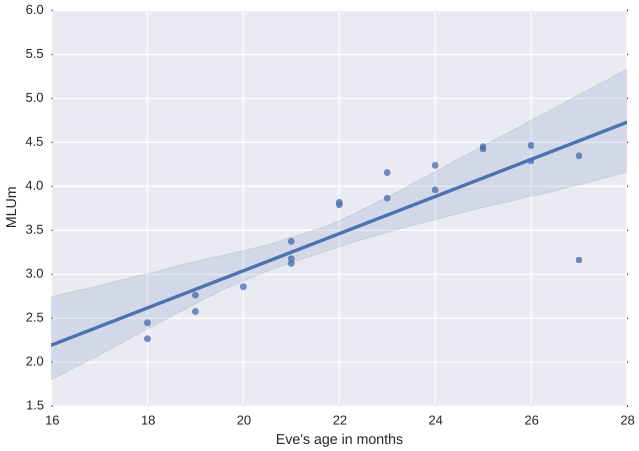

Figure 1: Eve's MLUm at different ages (Pearson's $r = 0.84, p < 0.001$)

The results show that Eve's age is significantly (and positively) correlated with MLUm (Pearson's $r = 0.84, p < 0.001$).

The code for MLUm computation illustrated above is available online. The computation of MLUm is performed entirely in Python from reading the dataset all the way to data analysis and visualization, combining PyLangAcq with other Python libraries and tools.[2]

3 Language dominance measured by MLUw

As the mean length of utterance is commonly used as a measure of language development, it is used in a wide variety of research topics in language acquisition. We illustrate how PyLangAcq can be used in research on bilingualism, specifically in the area of bilingual first language acquisition.

[2]The complete code is here: `http://pylangacq.org/papers/tech-report-2016.html`

Other libraies and tools of Python (van Rossum and Drake Jr 1995) we have used are IPython Notebook (Pérez and Granger 2007), SciPy (Jones et al. 2001–), pandas (McKinney 2010), and Seaborn (Waskom 2015) (built upon matplotlib (Hunter 2007)).

An important aspect of bilingualism concerns how various factors might contribute to the developmental trajectories of different languages spoken by a bilingual speaker. Essential in this research area is a reliable means of measuring language dominance, for whether (and by how much) a bilingual speaker is more competent in one language than in another. Here, we use PyLangAcq to replicate some of the results of—and possibly provide new insights for—Yip and Matthews (2007) for language dominance.

We focus on the datasets from the three siblings Timmy (eldest), Sophie, and Alicia (youngest) from the "YipMatthews" corpus in the "Biling" section of CHILDES. Born and raised in Hong Kong, they are Cantonese-English bilinguals whose mother is a native speaker of Cantonese and whose father is a native speaker of English. Following Yip and Matthews (2007: 73-81), we compare patterns of language dominance (measured by MLUw, the mean length of utterance in words) of these three children acquiring Cantonese and English simultaneously:[3]

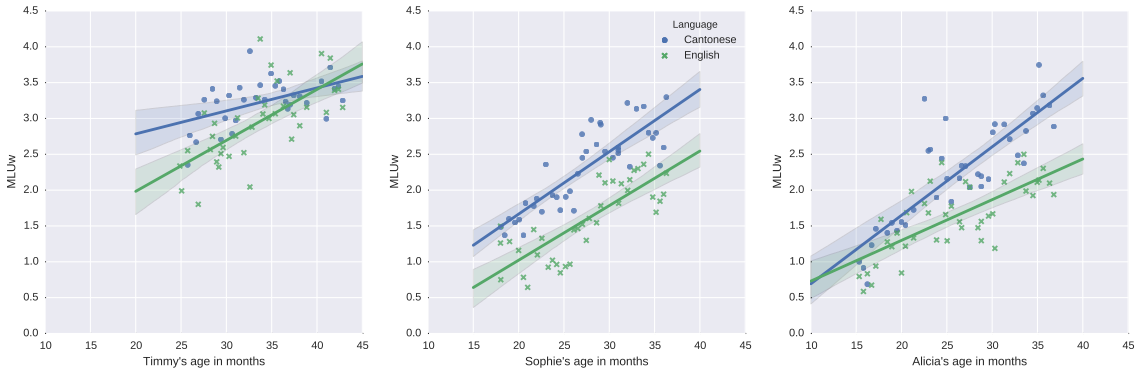

Figure 2: MLUw of Timmy, Sophie, and Alicia from CHILDES YipMatthews

For the purposes of comparison, the three plots in Figure 2 are produced with the same axes and ranges of values. For each language of each child, the best fit lines showing the overall trajectory (with shaded error regions at the 95% confidence interval) are also given.

Based on Figure 2, several observations are borne out, which are also discussed by Yip and Matthews. The three children appear to exhibit Cantonese dominance in general, but with interesting differences. Timmy, the eldest sibling, shows higher competence in Cantonese early on, but his English caught up quickly during the period of study. For Sophie and Alicia, Yip and Matthews (2007: 77) point out that they show a consistent pattern of Cantonese dominance. This seems to be the case as shown in Figure 2, although Alicia shows the pattern of relatively increasing preference of Cantonese, whereas Sophie's competence in Cantonese and English matures in a more or less comparable rate. And yet Sophie and Alicia's patterns contrast sharply with Timmy's, whose preferential growth of English competence during the period of study is unobserved in his sisters. It is interesting to see how divergent bilingual development can be – even within the same family. While these finer-grained observations are possibly tangential to the particular research that Yip and Matthews (2007) focus on, the fact that more detailed statistical analyses and data visualization are available in a purely Python environment incorporating PyLangAcq shows that PyLangAcq can facilitate language acquisition research for large datasets and more sophisticated computational and statistical analysis.

[3]The complete code is also available online. See footnote 2.

4 Phonological development

PyLangAcq also facilitates research by use in conjunction with other Python tools developed particularly for linguistics. Continuing with Cantonese, one of the languages exemplified above, we use PyCantonese (Lee 2015), a Python library for Cantonese linguistic research. In the following, we briefly explore phonological development – child tone production in particular; Cantonese is a tone language with six tones. In this example, PyLangAcq handles the CHILDES Cantonese monolingual child development data from Lee and Wong (1998), and PyCantonese parses Cantonese romanization for extracting tone information.

We use the data from the child MHZ. There are altogether 16 CHAT data files, with the age range of 24.5-32.2 months. As a first step for future work, we briefly explore the distribution of tones produced by MHZ. For each file, we count the number of times a particular tone is produced by MHZ. The results are presented in the following heatmap:[4]

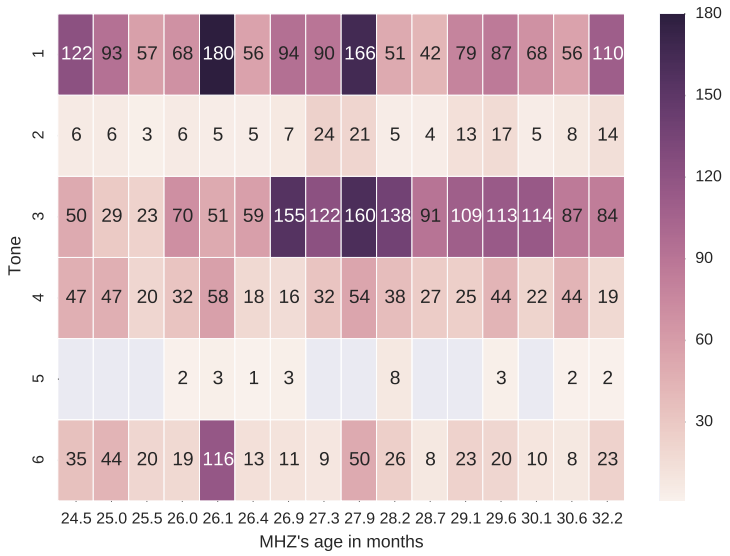

Figure 3: MHZ's tone production

With the aid of heatmap visualization in Figure 3, we can see that the level tones (tones 1, 3, 6) and the low-falling/level tone 4 appear to be most frequently produced by the child. The observation that the level tones are empirically more frequent than contour tones is similar to findings with regards to adult Cantonese corpus studies (Leung et al. 2004). Possible further research which could be performed using PyLangAcq includes comparing children's speech, as shown above, to child-directed speech, and modeling the development of tone production distribution over time.

5 Conclusion

PyLangAcq opens the door for reproducible, accessible, and extensible research in language acquisition and any work that involves CHAT transcripts. Written in Python, the lingua franca in

[4]The complete code for this part is also available online. See footnote 2.

computational linguistics and natural language processing, PyLangAcq facilitates computational and statistical modeling as it has direct access to many other Python tools for computationally demanding research. While PyLangAcq is still at its infancy, this report has provided some basic examples of its use. Many more functions and modules will be implemented as the library grows and evolves.

References

Alishahi, Afra. 2010. *Computational Model of Human Language Acquisition*. Morgan & Claypool Publishers.

Bird, Steven, Edward Loper and Ewan Klein. 2009. *Natural Language Processing with Python*. O'Reilly Media Inc.

Brown, Roger. 1973. *A first language: The early stages*. Harvard University Press.

Hunter, John D. 2007. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* 9(3). doi:10.1109/MCSE.2007.55.

Jones, Eric, Travis Oliphant, Pearu Peterson et al. 2001–. SciPy: Open source scientific tools for Python.

Lee, Hun-Tak Thomas and Colleen Wong. 1998. Cancorp: the Hong Kong Cantonese Child Language Corpus. *Cahiers de Linguistique Asie Orientale* 27: 211–228.

Lee, Jackson L. 2015. PyCantonese: Cantonese linguistic research in the age of big data. Talk at the Childhood Bilingualism Research Centre, the Chinese University of Hong Kong.

Leung, Man-Tak, Sam-Po Law and Suk-Yee Fung. 2004. Type and token frequencies of phonological units in Hong Kong Cantonese. *Behavior Research Methods, Instruments, and Computer* 36(3): 500–505.

MacWhinney, Brian. 2000. *The CHILDES project: Tools for analyzing talk*. Mahwah, NJ: Lawrence Erlbaum Associates.

McKinney, Wes. 2010. Data structures for statistical computing in Python. In Stéfan van der Walt and Jarrod Millman (eds.), *Proceedings of the 9th Python in Science Conference*, 51 – 56.

Nagano, Tomonori and Virginia Valian. 2011. Is fully-automated corpus-based language acquisition research feasible? Poster presentation at the Architectures and Mechanisms of Language Processing (AMLaP).

Pérez, Fernando and Brian E. Granger. 2007. IPython: A system for interactive scientific computing. *Computing in Science & Engineering* 9(3): 21–29. doi:10.1109/MCSE.2007.53.

van Rossum, Guido and Fred L Drake Jr. 1995. *Python reference manual*. Amsterdam: Centrum voor Wiskunde en Informatica.

Villavicencio, Aline, Thierry Poibeau, Anna Korhonen and Afra Alishahi (eds.). 2013. *Cognitive aspects of computational language acquisition*. Springer.

Waskom, Michael. 2015. Seaborn: v0.6.0. doi:10.5281/zenodo.19108.

Yip, Virginia and Stephen Matthews. 2007. *The Bilingual Child: Early Development and Language Contact*. Cambridge University Press.