Polyglot and Speech Corpus Tools: a system for representing, integrating, and querying speech corpora

Michael McAuliffe[1], *Elias Stengel-Eskin*[1], *Michaela Socolof*[2], *Morgan Sonderegger*[1]

[1]Department of Linguistics, McGill University, Canada
[2]Department of Linguistics, University of Maryland, USA

michael.mcauliffe@mail.mcgill.ca, elias.stengel-eskin@mail.mcgill.ca, msocolof@umd.edu, morgan.sonderegger@mcgill.ca

Abstract

Speech datasets from many languages, styles, and sources exist in the world, representing significant potential for scientific studies of speech—particularly given structural similarities among all speech datasets. However, studies using multiple speech corpora remain difficult in practice, due to corpus size, complexity, and differing formats. We introduce open-source software for *unified corpus analysis*: integrating speech corpora and querying across them. Corpora are stored in a custom 'polyglot persistence' scheme that combines three sub-databases mirroring different data types: a Neo4j graph database to represent temporal annotation graph structure, and SQL and InfluxDB databases to represent meta- and acoustic data. This scheme abstracts away from the idiosyncratic formats of different speech corpora, while mirroring the structure of different data types improves speed and scalability. A Python API and a GUI both allow for: enriching the database with positional, hierarchical, temporal, and signal measures (e.g. utterance boundaries, f0) that are useful for linguistic analysis; querying the database using a simple query language; and exporting query results to standard formats for further analysis. We describe the software, summarize two case studies using it to examine effects on pitch and duration across languages, and outline planned future development.

Index Terms: speech database management, speech analysis corpus phonetics, laboratory phonology

1. Introduction

A huge and ever-increasing pool of speech data annotated at least with an orthographic transcription exists in the world, from public speech datasets, academic laboratories, online media archives, and other sources. This pool spans languages, speech styles, and historical time. At the same time, increasingly accurate automatic tools exist to align these speech corpora [1, 2, 3, 4] and measure the standard variables used in language research (e.g. vowel formants, f0, VOT: [1, 5, 6]). This confluence of 'big data' and speech processing tools has significant scientific potential, by enabling linguists and speech scientists to study spoken language at a much larger scale than previously possible [7, 8]. While such 'large scale' studies have begun to be conducted (e.g. [9, 10]), they remain limited to technically-skilled researchers and corpora of similar formats, due to practical barriers described below. Realizing this potential requires software for *unified corpus analysis*: integrating speech corpora, enriching them with measures of interest, and querying across them.

This paper introduces Polyglot and SCT, open-source software tools for unified corpus analysis. Polyglot-SCT consists of a Python API (Polyglot) and a GUI (Speech Corpus Tools: SCT).[1] These tools are motivated by two aspects of speech corpora that act as barriers to large-scale studies.

First, speech corpora are large. While any individual corpus can be processed on a modern laptop, storage and processing time become issues when working with many corpora at once. Thus, key goals of Polyglot-SCT are *scalability* and *speed*: performance in reasonable time as the amount of data grows.

Second, speech corpora are complex and heterogeneous. Directory structure, metadata, and annotation files can all be highly structured. Dozens of formats have been used to store speech corpora over the past 25+ years. These factors make studies using data from many corpora practically difficult, with researchers writing extensive scripts to perform similar operations on different corpora, despite substantial structural similarities across all speech corpora. The scripting involved in corpus work is time-consuming for technical users, and a deal-breaker for non-technical users. Thus, two goals of Polyglot-SCT are *minimizing scripting* and *abstraction* away from corpus format. Users should need minimal technical skill, and should be able to interact with corpora without understanding particularities of their formats (as for textual corpora in NLTK [11]). Technically-skilled users should be able to avoid rewriting scripts with similar functionality.

Each of these goals informs the current implementation and planned future development of Polyglot-SCT, which we describe below. We then summarize two case studies demonstrating the software's feasibility for large-scale studies and unified corpus analysis.

2. Background

Annotation graphs are a formal model for linguistic transcription using directed acyclic graphs. Annotation graphs draw on the logical structure underlying all annotated linear signals, such as transcribed speech. Nodes are points in time, and annotations are edges between those nodes, representing intervals over which an annotation occurs [12]. Annotation graphs are used as the data model in several speech corpus management systems, described below. Annotation graphs are implemented in our system using a graph database [13], motivated by a key design principle of Polyglot-SCT.

Representation and storage of data in Polyglot-SCT relies on the principle of *polyglot persistence* [14]: different databases are used for different data types, with each database closely matching the format of its data. Polyglot persistence improves the speed and scalability of both development and use of the

[1] https://github.com/MontrealCorpusTools/PolyglotDB and http://montrealcorpustools.github.io/speechcorpustools/

system, because each database is already set up in a way that is optimized for the structure of each data type. We use three databases corresponding to different types of data for speech corpora, described in Section 3.

2.1. Other systems

Several other systems for management and analysis of speech corpora exist (e.g. [15, 16, 17, 18, 19]), including three systems which are most similar to Polyglot-SCT.

Phon [15] is a system for creating and querying corpora. Phon uses a relational database to store data, but does not adopt the annotation graph formalism. Phon is integrated with Praat [20], and allows for a range of acoustic analyses and linguistic analyses (e.g. syllabification) across many languages. LaBB-CAT [16] stores recordings and associated transcriptions as annotation graphs in a relational database. In addition to import, export, and querying, LaBB-CAT can enrich a corpus in various ways (e.g. forced alignment, syllabification), and offers integration with Praat and lexical databases. EMU-SDMS [17] is a system consisting of an R [21] package, to simplify the full pipeline of corpus research to a single environment for data preparation and analysis, and a web application for annotation and file inspection. EMU also uses annotation graphs, which are stored in JSON files, as are subsequent measurements (f0, etc.) made using a signal processing library. Querying is done through a custom query language.

Polyglot-SCT differs from other systems in its goals: it is optimized for large-scale studies across many corpora, maximizing scalability, speed, and ease of use. Polyglot-SCT is not integrated with annotation or statistical data analysis, assumes that human annotation is complete, and carries out only data processing that can be done automatically. This design anticipates planned future development, to allow working with corpora without access to the raw data.

3. Implementation and features

Fig. 1 (left) shows the architecture of Polyglot-SCT: (1) Data from speech corpora is stored in three database types, described below (*import*). (2) Positional, hierarchical, temporal, and signal measures are added to the databases using speech processing tools, internal algorithms, and external resources (e.g. lexicons) (*enrichment*). (3) The user executes a *query* over the databases. (4) The returned information is saved to a data file (*export*). Fig. 1 (right) shows an example of the system's use.

Steps (1)–(4) are described further in 3.1–3.3 below. (1)–(4) can be carried out using either Polyglot, a Python API, or Speech Corpus Tools (SCT), a graphical user interface that can be used by non-technical users without writing Python scripts. SCT contains various interfaces and dialogues to facilitate use, and is extensively documented, including a tutorial.

The system uses three kinds of database, for the three principle data types associated with speech corpora. First, the linear linguistic annotation is modeled as annotation graphs in a NoSQL *graph database*, implemented in Neo4j [13]. This database mirrors annotation graphs' formal model of directed acyclic graphs. To work better with Neo4j, the annotation graph formalism has been modified. Annotations are nodes (rather than edges) with precedence edges between them. Other edge types are used to other relationships, including hierarchical relationships between different levels (i.e. words and phones, speakers and files), and type-token relationships (i.e. lexical items and their productions by speakers).

Second, acoustic measurements which have values at fixed intervals over time (e.g. f0, intensity, formants) are stored in InfluxDB [22], a NoSQL *time-series database*. This kind of database is optimized for operations over such time-series data.

Third, tabular data—such as corpus metadata and properties of words and segments (e.g. word frequency or phonological features)—are stored in SQLite [23], a traditional *relational database*, which is best suited for storing this kind of data.

3.1. Import

The import step implements Polyglot-SCT's goal of abstraction away from corpus format: speech corpora in different formats (as in Fig 1b) are imported into a standardized database format, using pre-written *importers*. Currently the default importer loads Praat [20] TextGrids, and allows for structuring 'tiers' in the TextGrid into the more meaningful hierarchy defined in the database format (e.g. each phone token belongs to a corresponding word). TextGrid-based formats which are output from various programs are also supported: the MFA [3], Prosodylab-Aligner [4], and FAVE [1] forced aligners, as well as LaBB-CAT. Importers also exist for corpora in BAS Partitur format [24], as well as for the TIMIT and Buckeye corpora [25, 26].

However, there are many currently unsupported formats, including standardized (akin to Partitur) and idiosyncratic formats (akin to TIMIT). While we plan to expand the set of supported formats, the import pipeline has also been designed to make writing new importers maximally easy. A new importer involves only writing Python to parse the corpus into an intermediate Python object—no knowledge of the database systems (Neo4j, InfluxDb, SQLite) is required.

3.2. Enrichment

A freshly-imported corpus will result in a database containing at a minimum the word and phone levels (see Figure 1 right) and any other information from the corpus' annotation files. Any other information is added in the enrichment phase.

Polyglot-SCT databases can be enriched in many ways, by adding structure and measures that are often used in linguistic studies. First, new annotations can be created. Larger connected speech chunks, termed *utterances*, can be created as parents of word annotations (as in Fig. 1 right). Utterances are created by encoding speech versus non-speech elements in each file, then specifying the minimum duration of non-speech elements corresponding to an utterance boundary. Syllable annotations, which are parents of phones, can also be created using the maximum onset algorithm. We plan to add other algorithms for marking boundaries and for syllabification in the future.

Second, measures based on hierarchical relations can be calculated and stored. For instance, once utterances and syllables have been created, speech rate can be calculated as syllables per second in the utterance, and stored as a property of the utterance. Count and position of lower elements within higher elements can be encoded, such as syllable position within a word or number of syllables in a word (properties of the syllable and word, respectively).

Third, properties of lexical items, segments, and metadata about speakers or sound files can be added—such as from lexicons (e.g. frequency, part of speech) or files listing properties of phones (e.g. phonological features).

Fourth, acoustic measurements from the sound files can be calculated and stored. Currently, f0 (using Praat or Reaper [5]), intensity, and formants (using Praat) are supported. Other acoustic measurements will be added in future work, by incor-

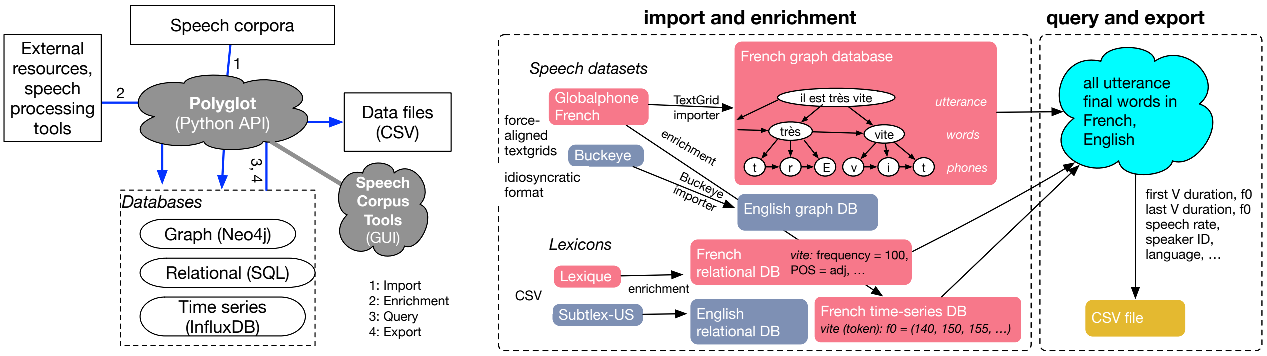

Figure 1: *Left: System architecture. Grey shapes = software, white shapes = elements used or created by the software, blue lines = steps in using the software (Sec. 3.1–3.3). Right: Schematic example of system use to carry out a study examining vowel duration in utterance-final words in French and English corpora of differing formats.*

porating external tools (e.g. for VOT, pitch accent detection: [6, 27]) and further integration with Praat.

Finally, 'relativized' versions of measures calculated in enrichment can be calculated. For example, it is often of interest in phonetic studies to know how long a phone (token) is relative to its mean duration in the corpus, or an utterance's speech rate relative to the speaker's mean rate.

Anything encoded as part of enrichment is saved and can be queried in the future. The intended use case for Polyglot-SCT is for import and enrichment to be done *once* per corpus. These steps can be slow (see Table 1), but require minimal input from the user. By contrast, querying and exporting are fast, and can be done many times, in different studies with different goals. This design allows users to not repeat work like recalculating measures (e.g. pitch tracks) for each new study, which is important for scalability to large-scale studies.

3.3. Query and Export

Queries consist of two parts: how to narrow results to a set of linguistic objects (*filters*), and what information to return about them (*columns*). Filters can target aspects of an annotation, such as 'phone label = b'. Aspects of nodes associated with the annotation can also be filtered on, such as the speaker or the following phone. Hierarchical aspects can also be filtered on, such as where an annotation occurs in the parent annotation (e.g. phone position in word), as well as on properties of higher annotations (e.g. orthography of the parent word). User-defined subsets, such as 'syllabics', can be defined and used in subsequent queries. Similarly to filters, columns can be properties of an annotation or linked nodes. Acoustic columns can also be specified, such as mean f0 or the full track of f0 measurements over an annotation. The returned results can be manipulated in Python or exported to an external file for further analysis. Currently, only export to CSV files is supported, but support for formats such as JSON and Feather is planned.

In Polyglot, queries are specified in a custom query language in Python style, so that users only need to know Python—and not the query languages of Neo4j, InfluxDB, and SQLite. In the SCT GUI, queries are built up using drop-down menus.

4. Case studies

We summarize two case studies, showing two types of common corpus studies Polyglot-SCT can be used to carry out at large scale: factors affecting duration of linguistic units, and examinations of acoustic measures such as f0. The studies together exemplify unified corpus analysis by using 15 corpora in two different formats. Each study demonstrates the workflow of using Polyglot-SCT: import, enrichment, query, and export.

Table 1 summarizes analysis time (on a desktop using 12 3.4-Ghz processors, 32 GB memory) and size of the input (speech corpora) and output (data files) for each study. Most running time (91–95%) is spent on import and enrichment, as opposed to query and export, in keeping with the intended use case where database setup is performed only once (Sec. 3.2).

Table 1: *Analysis time and size of input/output data for case studies. h=hours, m=minutes.*

Case	Size	Analysis time		Export
study	of	Import +	Query +	row
	corpora	Enrichment	Export	count
Vow. duration	40h	30m	1.4m	4,703
Intrinsic F0	275h	20h	2h	94,890

4.1. Obstruent voicing effects on vowel duration in English

This study examines the effect of following obstruent voicing on vowel duration in the Buckeye corpus of spontaneous speech [26], controlling for other factors. A more complex version of this case is described in detail in the SCT tutorial, using the LibriSpeech corpus of read speech [28] instead of the Buckeye corpus.[2] The effect of following obstruent voicing on vowel duration is thought to have the same direction across languages (voiced > voiceless) [29], and dialects of English [30]. However, obstruent voicing is one of many factors affecting vowel duration, and the robustness of the effect in spontaneous speech is unclear. We examine how large and reliable the effect of obstruent voicing is relative to other factors (speech rate, word frequency) in English spontaneous speech.

Import, Enrichment: The Buckeye corpus is imported into a database using the `Buckeye importer`. In enrichment we must add speech rate, consonant manner and voicing, syllable structure, and word frequency, none of which are included in Buckeye. Speech rate and syllables are calculated using the pipeline described in Sec. 3.2, with 'utterances' defined as speech segments bounded by non-speech intervals of >150 msec. Information about following consonant manner/voicing

[2] The analysis steps are identical for Buckeye and LibriSpeech due to the system's abstraction from corpus format.

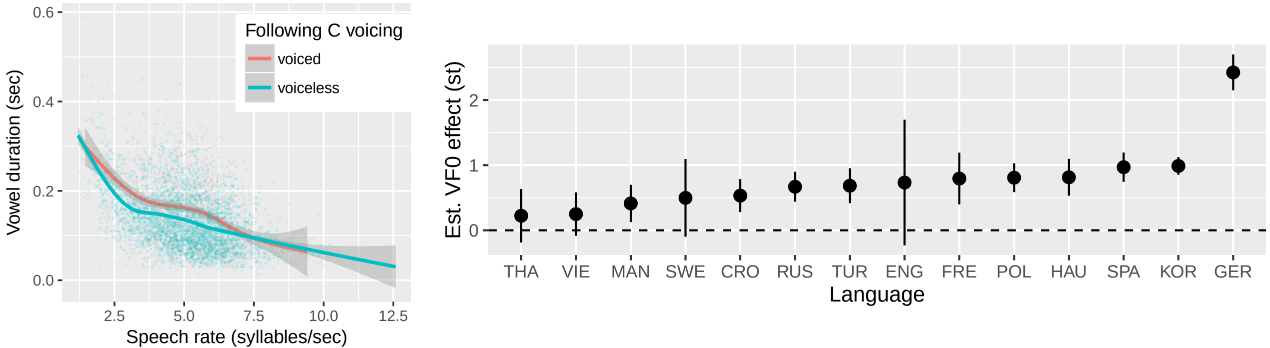

Figure 2: *Case study results. Left: Vowel duration in utterance-final CVC words in the Buckeye corpus, as a function of following consonant voicing and speech rate. Right: Difference in f0 (semitones) between high and low vowels for each of 14 languages, predicted by a statistical model controlling for other factors.*

and word frequency are added via CSV files that encode phonological featural information and lexical statistics [31].

Query, Export : To examine the effects of following C voicing on V duration, we find: all vowels in CVC words (fixed syllable structure), where the second C is a stop (to examine voicing effects independent of C manner), at the end of utterances (fixed prosodic position). The tutorial shows how filters are applied to narrow down to this set of vowel tokens, and how the following columns are selected to export to a CSV file: voicing of the following consonant, vowel identity, word orthography and frequency, speaker ID, speech rate, and identity and voicing of the following consonant.

Results: Consonant voicing affects vowel duration in the expected direction (voiced>voiceless), shown in Fig. 2 (left), but it is small. The effect of consonant voicing is also small compared to the effect of word frequency. These findings were confirmed in a linear mixed-effects regression, suggesting that following stop voicing may have a small effect on vowel duration in spontaneous English.

4.2. Intrinsic F0 across languages

'Intrinsic F0' (IF0) refers to effects of vowel height and following consonant voicing on f0. While IF0 effects have been documented for many languages and proposed to be universal [32], their robustness across languages and across speakers using comparable data and methods is unclear. This study examines cross-linguistic and interspeaker variability in IF0 effects across 14 languages. We use read sentences, from corpora of 13 languages from GlobalPhone [33]—Croatian, French, German, Hausa, Korean, Mandarin, Polish, Russian, Spanish, Swedish, Thai, Turkish Vietnamese (∼20 hrs/language)—and English (using a ∼2 hr subset of LibriSpeech), all force-aligned using the Montreal Forced Aligner [3]. The datasets contain speech from 67–113 speakers per language. Details of the study can be found on the last author's website; here we focus on its implementation using Polyglot-SCT.

Import, Enrichment: The corpus for each language was imported into a database using the `MFA importer`. A number of properties needed for the case study are added in enrichment. Information about vowel height, consonant voicing, and speaker gender was loaded from CSV feature files containing this information (available on the Polyglot-SCT site). Speech rate was calculated via the same pipeline as above. Pitch tracks were added to each database via integration with Praat (see Sec. 3.2), using gender-adjusted pitch ranges.

Query, Export: To focus on the effects of vowel height and preceding consonant voicing on f0—while controlling for effects of intonational context, consonant manner, syllable structure, and the languages' different vowel inventories—our query finds all {a,i,u} vowel tokens (present in each language) in utterance-initial obstruent-vowel syllables. We export to one CSV per language all information for each vowel token: the pitch track, preceding consonant voicing and identity, vowel height and identity, language, speaker ID, speech rate, vowel height, and word identity.

Results: The f0 data was further processed in R, including transforming to semitones and data cleaning to exclude unreliable f0 measures, then used to examine consonant voicing and vowel height effects on f0, as well as individual differences. We focus only on the effect of vowel height on f0 here. Fig. 2 (right) shows the estimated effect for each language, from a linear mixed-effects model controlling for other relevant factors (including following consonant voicing). The effect is positive in most languages (f0 higher in high than in low vowels), confirming the near-universality of vowel height effects on f0 [32], but its magnitude differs greatly across languages. The smallest effects are for languages that use f0 contrastively (tonal or pitch accent: Thai, Vie, Man, Swe, Cro) in line with work suggesting that IF0 is actively attenuated in some tone languages [34].

5. Conclusion

We have described Polyglot and SCT, new open-source tools for unified corpus analysis, and demonstrated the kind of large-scale studies of speech they can be used to perform. Future development will further optimize and enhance functionality of each step of a Polyglot-SCT analysis: import, enrichment, query, and export. For example, we plan to add support for point annotations (e.g. ToBI), voice onset time (VOT), user-customization of enrichment by allowing for arbitrary Praat scripts to encode acoustic measure, and to expose more parameters of existing enrichment to the user.

6. Acknowledgements

We acknowledge funding from SSHRC #430-2014-00018, FRQSC #183356 and CFI #32451 to MS, as well as McGill ARIA and McGill-Glasgow Mobility grants to ESE and MM. We thank Jane Stuart-Smith, Michael Wagner, Yvan Rose, Rafael Winkelmann, and participants in the Easier Speech Corpus Analysis workshop at U. Glasgow for useful feedback.

7. References

[1] I. Rosenfelder, J. Fruehwald, K. Evanini, and J. Yuan, "FAVE (forced alignment and vowel extraction) program suite," http://fave.ling.upenn.edu, 2011.

[2] T. Kisler, F. Schiel, and H. Sloetjes, "Signal processing via web services: the use case WebMAUS," in *Digital Humanities Conference 2012*, 2012.

[3] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal Forced Aligner [computer program]," https://montrealcorpustools.github.io/Montreal-Forced-Aligner/, 2017.

[4] K. Gorman, J. Howell, and M. Wagner, "Prosodylab-aligner: A tool for forced alignment of laboratory speech," *Canadian Acoustics*, vol. 39, no. 3, pp. 192–193, 2011.

[5] D. Talkin, "REAPER: Robust Epoch And Pitch EstimatoR [computer program]," https://github.com/google/REAPER, 2015.

[6] J. Keshet, M. Sonderegger, and T. Knowles, "AutoVOT: A tool for automatic measurement of voice onset time using discriminative structured prediction [Computer program]," Version 0.91. Available at https://github.com/mlml/autovot/, 2014.

[7] M. Liberman, "A new golden age of phonetics," Presentation at Center for Speech and Language Processing, Johns Hopkins University. Available at: https://vimeo.com/32571929, 2009.

[8] J. Coleman, M. Liberman, G. Kochanski, L. Burnard, and J. Yuan, "Mining a year of speech," in *Proceedings of VLSP 2011: New Tools and Methods for Very-Large-Scale Phonetics Research*, 2011, pp. 16–19.

[9] J. Yuan, M. Liberman, and C. Cieri, "Towards an integrated understanding of speaking rate in conversation," in *Proceedings of Interspeech*, 2006, pp. 541–544.

[10] J. Yuan and M. Liberman, "Automatic measurement and comparison of vowel nasalization across languages," in *Proceedings of ICPhS*, vol. 17, 2011, pp. 2244–2247.

[11] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*, 2006, pp. 69–72.

[12] S. Bird and M. Liberman, "A formal framework for linguistic annotation," *Speech communication*, vol. 33, no. 1, pp. 23–60, 2001.

[13] Neo4j Developers, "Neo4j: Graph NoSQL database [computer program]," https://neo4j.com/, 2017.

[14] P. J. Sadalage and M. Fowler, *NoSQL distilled: a brief guide to the emerging world of polyglot persistence*. Pearson Education, 2012.

[15] Y. Rose, B. MacWhinney, R. Byrne, G. Hedlund, K. Maddocks, P. O'Brien, and T. Wareham, "Introducing Phon: A software solution for the study of phonological acquisition," in *Proceedings of the 30th Annual Boston University Conference on Language Development*, vol. 2006, 2006, pp. 489–500.

[16] R. Fromont and J. Hay, "LaBB-CAT: An annotation store," in *Australasian Language Technology Association Workshop 2012*, vol. 113, 2012, pp. 113–117.

[17] R. Winkelmann, J. Harrington, and K. Jänsch, "EMU-SDMS: Advanced speech database management and analysis in R," *Computer Speech & Language*, 2017.

[18] T. Schmidt and K. Wörner, "EXMARaLDA–creating, analyzing and sharing spoken language corpora for pragmatics research," *Pragmatics*, vol. 19, no. 4, pp. 565–582, 2009.

[19] B. Bigi, "SPPAS - multi-lingual approaches to the automatic annotation of speech," *The Phonetician*, vol. 111–112, pp. 54–69, 2015.

[20] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [computer program]," 2017.

[21] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016. [Online]. Available: https://www.R-project.org/

[22] InfluxData, "InfluxDB: An open-source time series database [computer program]," https://www.influxdata.com/open-source/\#influxdb, 2017.

[23] M. Owens and G. Allen, *SQLite*. Springer, 2010.

[24] F. Schiel, S. Burger, A. Geumann, and K. Weilhammer, "The Partitur format at BAS," in *Proceedings of the First International Conference on Language Resources and Evaluation*, 1998, pp. 1295–1301.

[25] J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, and N. Dahlgren, *TIMIT Acoustic-Phonetic Continuous Speech Corpus*. Philadelphia: Linguistic Data Consortium, 1993.

[26] M. Pitt, L. Dilley, K. Johnson, S. Kiesling, W. Raymond, E. Hume, and E. Fosler-Lussier, *Buckeye Corpus of Conversational Speech (2nd release)*. Columbus: Department of Psychology, Ohio State University, 2007.

[27] A. Rosenberg, "AuToBI-a tool for automatic ToBI annotation," in *Proceedings of Interspeech*, 2010, pp. 146–149.

[28] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proceedings of ICASSP 2015*, 2015, pp. 5206–5210.

[29] M. Chen, "Vowel length variation as a function of the voicing of the consonant environment," *Phonetica*, vol. 22, no. 3, pp. 129–159, 1970.

[30] J. Tauberer and K. Evanini, "Intrinsic vowel duration and the postvocalic voicing effect: some evidence from dialects of north american English," in *Proceedings of Interspeech*, 2009, pp. 2211–2214.

[31] K. Vaden, H. Halpin, and G. Hickok, "Irvine Phonotactic Online Dictionary, Version 2.0. [Data file]," http://www.iphod.com, 2009.

[32] D. H. Whalen and A. G. Levitt, "The universality of intrinsic F0 of vowels," *Journal of Phonetics*, vol. 23, no. 3, pp. 349–366, 1995.

[33] T. Schultz, N. T. Vu, and T. Schlippe, "Globalphone: A multilingual text & speech database in 20 languages," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8126–8130.

[34] B. Connell, "Tone languages and the universality of intrinsic F0: evidence from Africa," *Journal of Phonetics*, vol. 30, no. 1, pp. 101–129, 2002.