Phon: A Computational Basis for Phonological Database Building and Model Testing

Yvan Rose, Gregory J. Hedlund, Rod Byrne, Todd Wareham, and Brian MacWhinney

Abstract This paper describes Phon, an open-source software program for the transcription, coding, and analysis of phonetically transcribed speech corpora. Phon provides support for multimedia data linkage, utterance segmentation, multiple-blind transcription, transcription validation, syllabification, and alignment of target and actual forms. All functions are available through a user-friendly graphical interface. This program provides the basis for the building of PhonBank, a database project that seeks to broaden the scope of CHILDES into phonological development and disorders.

Key words: Phonology, Database, Software, Transcription, Annotation, Language Acquisition

Yvan Rose
Department of Linguistics, Memorial University of Newfoundland
St. John’s, NL A1B 3X9 Canada
e-mail: yrose@mun.ca

Gregory J. Hedlund
Department of Linguistics, Memorial University of Newfoundland
St. John’s, NL A1B 3X9 Canada
e-mail: ghedlund@mun.ca

Rod Byrne
Department of Computer Science, Memorial University of Newfoundland
St. John’s, NL A1B 3X5 Canada
e-mail: rod@mun.ca

Todd Wareham
Department of Computer Science, Memorial University of Newfoundland
St. John’s, NL A1B 3X5 Canada
e-mail: harold@mun.ca

Brian MacWhinney
Department of Psychology, Carnegie Mellon University
Pittsburgh, PA 15213 USA
e-mail: macw@cmu.edu

1 Introduction

The topic of this chapter may appear as a slight oddity in the context of the current publication. While most of the contributions to this volume focus on computational methods applied to language learning problems, our paper centers on a recently-introduced tool for the building of phonetically-transcribed speech corpora. This is relevant in a number of respects. Empirical studies of natural language and language acquisition will always be required in most types of linguistic research, as these studies provide the basis for describing languages and linguistic patterns. In addition to providing us with baseline data, corpora also allow us to test models of various kinds, be they theoretical, neurological, psychological or computational. However, the building of natural language corpora is an extremely tedious and resource-consuming process, despite tremendous advances in data recording, storage, and coding methods in recent decades.

Thanks to corpora and tools such as those developed in the context of the CHILDES project (http://childes.psy.cmu.edu/), computational scientists interested in morphology and syntax have enjoyed convenient and powerful methods for analysing the morphosyntactic properties of natural languages and their acquisition by first and second language learners. In the area of phonetics, the Praat system (http://www.fon.hum.uva.nl/praat/) has expanded our abilities to test optimality-theoretic as well as neural network-based learning models, in addition to providing a breadth of support in the areas of speech analysis and synthesis.

In this rapidly-expanding software universe, phonologists interested in the organisation of sound systems (e.g. phones, syllables, stress and intonational patterns) and their acquisition had not enjoyed the same level of computational support prior to the inception of the PhonBank project within CHILDES. There was no developed platform for phonological analysis and no system for data sharing. This situation negatively affected the study of natural language phonology and phonological development. It also undermined potential studies pertaining to interfaces between various components of the grammar or the elaboration of computational models of language and language development.

It is widely accepted that a spoken utterance consists of more or less one sentence. Utterances can contain one or more phonological phrases, which can serve as reference domains for intonational purposes or relate to independent aspects of syntactic constituency. Phonological phrases are typically made of series of one or more prosodic words and associated morphemes, with each of these meaningful units consisting of syllables which can themselves be broken down into individual phones. This general grammatical organisation allows us to make reference to factors that link the smallest phonological units to morphological and/or syntactic levels of grammatical patterning. For example, in English, the phonological phrase, a domain that constrains phonological phenomena such as intonation, must typically be described using syntactic criteria; in a similar way, the analysis of stress patterns in this language requires references to large-domain morphological boundaries (e.g. [Selkirk, 1986]). Studying the acquisition of these grammatical structures, and of

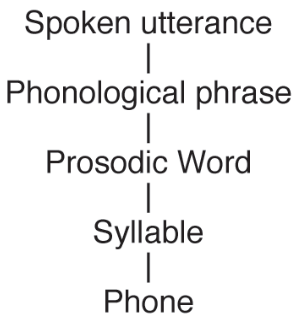


Fig. 1 General organisation of a spoken utterance

their phonological components, can help us understand how linguistic knowledge emerges in the grammar of a language learner.

In this paper we discuss Phon, an innovative open-source software program that offers significant methodological advances in research in phonology and phonological development. On the one hand, Phon provides a powerful and flexible solution for phonological corpus building and analysis. On the other hand, its ability to integrate with other open-source software facilitates the construction of complete analyses across all levels of grammatical organisation represented in Fig. 1. Although the primary target group for this tool was originally L1 researchers, the core functions of Phon are equally valuable to other speech researchers who are interested in analysing language variation of any type (e.g. cross-dialectal variation, L2 speech, speech pathologies, evolution of consonant inventories; on this last application, see Mukherjee et al. in this volume).

The paper is organised as follows. In Sect. 2, we discuss the general motivation behind the Phon project. In Sect. 3, we describe the functionality supported in Phon. In Sect. 4, we focus on the query and reporting systems that are built into the application. We then summarize in Sect. 5 currently planned extensions to Phon, including both the integration of acoustic data analyses and a greatly expanded database query functionality that will ultimately assist in both language acquisition model testing and derivation. Concluding remarks are offered in Sect. 6.

2 The PhonBank Project

PhonBank, one of the latest initiatives within the CHILDES project, focuses on the construction of corpora suitable for phonological and phonetic analysis. In this section we first describe the goals of PhonBank. We then describe Phon, the software program designed to facilitate this endeavor.

2.1 PhonBank

The PhonBank project seeks to broaden the scope of the current CHILDES system to include the analysis of phonological development in first and subsequent languages for learners with and without language disorders. To achieve this goal, we have created a new phonological database called PhonBank and a program called Phon to facilitate the analysis of PhonBank data. Using these tools, researchers are in a position to conduct a series of developmental, crosslinguistic, and methodological analyses based on large-scale corpora.

2.2 Phon

Phon consists of inter-connected modules that offer functionality to assist the researcher in important tasks related to corpus transcription, coding and analysis. (The main functions supported are discussed in the next section.)

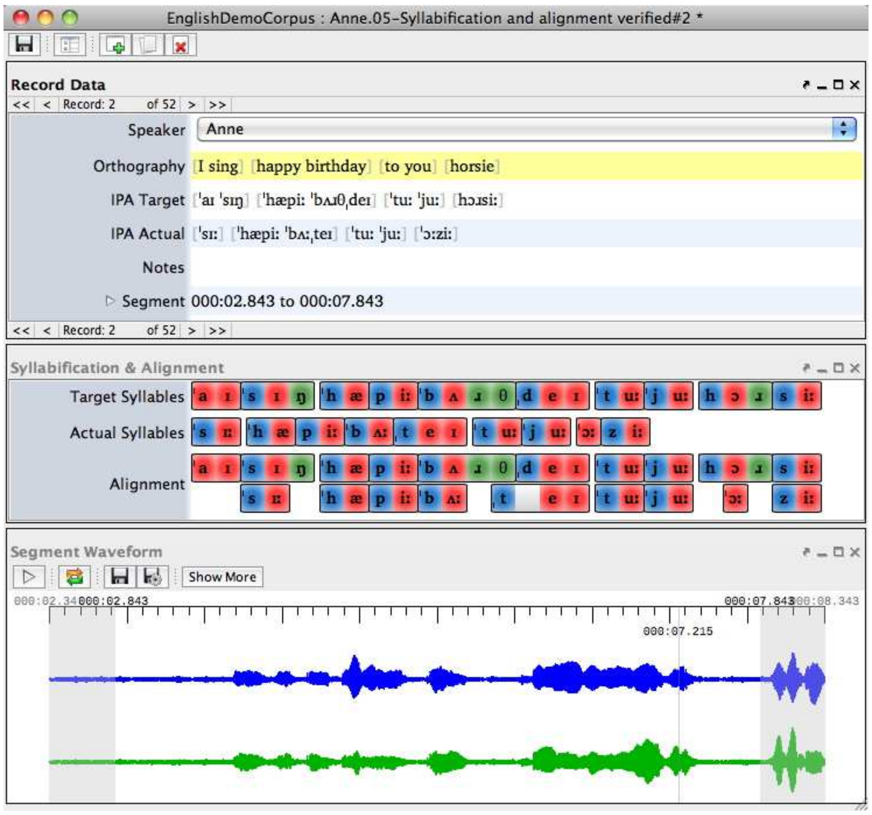
The application is developed in Java and is packaged to run on Mac OS X and Windows platforms that support Java 1.6.[1] Phon is Unicode-compliant, a required feature for the sharing of data transcribed with phonetic symbols across computer platforms. Phon can share data with programs which utilize the TalkBank XML schema for their documents such as those provided by the TalkBank and CHILDES projects.

Phon was introduced approximately 5 years ago (see [Rose et al., 2006]). Since then, we have thoroughly revised significant portions of the code to refine the functionality, ensure further compatibility with other TalkBank-compliant applications, and streamline the interface for better user experience and improved support for the general workflow involved in phonological corpus building. We also added novel and innovative functionality for corpus query and reporting. An advanced beta version of this application is publicly available online as a free download directly from the CHILDES website (http://childes.psy.cmu.edu/).

3 Phon

The general interface of Phon is exemplified in Fig. 2. It consists of a series of view panels, each of which supports particular aspects of corpus manipulation (e.g. session-level information, orthographically or phonetically transcribed data, other data annotations). In Fig. 2, three view panels are displayed (Record Data, Syllabification and Alignment, and the Waveform of the speech segment transcribed in this record). Additional view panels can be docked horizontally or vertically, or super-

[1] Support for the Unix/Linux platform is currently compromised, primarily because of licensing issues related to the multimedia functions of the application.


EnglishDemoCorpus : Anne.05-Syllabification and alignment verified#2 *

Record Data

Record: 2 of 52

Speaker: Anne

Orthography: [I sing] [happy birthday] [to you] [horsie]

IPA Target: [ˈaɪ ˈsɪŋ] [ˈhæpiː ˈbʌɹθˌdeɪ] [ˈtuː ˈjuː] [hɔɹsiː]

IPA Actual: [ˈsɪː] [ˈhæpiː ˈbʌːˌteɪ] [ˈtuː ˈjuː] [ˈɔːziː]

Notes:

Segment: 000:02.843 to 000:07.843

Fig. 2 Phon General Interface

posed as tabbed interfaces within single docks. This user-configurable interface is one of the key improvements brought to the current version. Using this interface, the user can perform a series of tasks related to the building of phonological corpora:

- Media linkage and segmentation.
- Data transcription and validation (including support for multiple-blind transcriptions).
- Segmentation of transcribed utterances (into e.g. phrases, words).
- Labeling of transcribed forms for syllabification.
- Phone and syllable alignment between target (expected) and actual (produced) forms.

In the next subsections, we describe the main functions supported by the application.

3.1 Project Management

The building of a corpus of transcribed phonological data often requires the combination of a number of data transcripts, be they from a single learner over a period of time or from multiple learners. Phon offers functions to create and manage sets of data transcripts, following a general corpus structure whereby a project contains one or many corpora, each of which contains a set of data transcripts. Session transcripts can be copied or moved across corpora or project files through the Project Manager interface.

3.2 Media Linkage and Segmentation

Linkage of multimedia data and subsequent identification of the portions of the recorded media that are relevant for analysis are now available directly from the application's main interface. These tasks follow the same logic as similar systems in programs like CLAN (http://childes.psy.cmu.edu/clan/). A transcript in Phon generally corresponds to a data recording session. The created media segments can be played back directly from the graphical user interface (GUI). Whenever needed, the user can also fine-tune the segments start and/or end time values, a task made much easier with the incorporation of waveform visualisation.

3.3 Data Transcription

As we saw in Fig. 2, the Session Editor incorporates in a single interface access to data transcription and annotation, transcription segmentation, syllabification and alignment. Phon also provides support for an unlimited number of user-defined fields that can be used for various kinds of textual annotations that may be relevant to the coding of a particular dataset. All data tiers can be ordered to accommodate specific data visualisation needs. Support for tier-specific fonts is also provided, a feature particularly useful for work based on non-Roman data transcripts. Phonetic transcriptions are based on the phonetic symbols and conventions of the International Phonetic Association (IPA). A useful IPA character chart is easily accessible from within the application, in the shape of a floating window within which IPA symbols and diacritics are organised into intuitive categories. This chart facilitates access to the IPA symbols for which there is no keyboard equivalent.

Target and actual IPA transcriptions are stored internally as strings of phonetic symbols. Each symbol is automatically associated with a set of descriptive features generally accepted in the fields of phonetics and phonology (e.g. bilabial, alveolar, voiced, voiceless, aspirated) [Ladefoged and Maddieson, 1996]. These features are extremely useful in the sense that they provide series of descriptive labels to each transcribed symbol. The availability of these labels is essential for research involving the grouping of various sounds into natural classes (e.g. voiced consonants; non-high front vowels). The built-in set of features can also be reconfigured to fit particular research needs through the query system (see Sect. 4.5 for further discussion).

Phon is also equipped with functionality to automatically insert IPA Target transcriptions based on the orthographic transcriptions. Citation form IPA transcriptions of these words are currently available for Catalan, German, English, French, Icelandic, Italian, Dutch and Spanish. IPA dictionary files from these various languages were generously provided by independent open-source projects and subsequently formatted for use into Phon.

In cases when more two or more pronunciations are available from the built-in dictionaries for a given written form (e.g. the present and past tense versions of the English word ‘read’), the application provides a quick way to select the required form. Of course, idealized citation forms do not provide accurate fine-grained characterisations of variations in the target language (e.g. dialect-specific pronunciation variants; phonetic details such as degree of aspiration in obstruent stops). They, however, typically provide a useful general baseline against which patterns can be identified.

3.4 Multiple-blind Transcription and Transcript Validation

Phon offers a fully-integrated system for multiple-blind, consensus-based IPA transcriptions. Multiple-blind transcription is in essence identical to the double-blind protocol: it consists of the IPA transcription of recorded utterances by two or more (hence, multiple) transcribers. Within Phon, the IPA transcribers are effectively ‘blinded’ from each other’s transcriptions in that they must perform their IPA transcriptions without being able to visualise the transcriptions of other transcribers. Each IPA transcriber logs into the blind transcription interface using a specific username. Upon login, a transcriber can visualise the regular corpus data records, including orthographic transcriptions and other annotations, with the crucial exception that the visible IPA transcription tiers are unique to each transcriber.

After the blind transcriptions are performed, they are then ready for the next step in the workflow, which consists of consensus-based transcript validation. This step is necessary as, under the blind transcription protocol, none of the user-specific transcriptions can immediately be considered valid for research. We developed an interface within Phon which facilitates record-by-record comparisons of the blind transcriptions. Using this interface, a team of two (or more) transcript validators can listen to the record’s speech segment, and then visualise in parallel all corresponding transcriptions produced by the blind transcribers. The transcription deemed the most accurate by consensus between the transcript validators is then selected with a simple mouse click. Whenever necessary, the selected transcription can be further adjusted according to the details noticed by the transcript validators during the comparison process. While this method is relatively onerous both in time and human resources, its combined steps (blind transcription followed by consensus validation) help to maximise transcription reliability for research purposes.

Employing blind transcription and its associated validation systems is optional. If the user decides not to perform blind transcriptions, the phonetic transcriptions are entered directly into the transcript and, as such, do not require subsequent validation. Similarly, the decision to protect each set of blind transcriptions with a password, which may be overkill in many situations, is left to the user. Note as well that only data which have been validated or directly entered into the transcript can be further annotated or compiled. Non-validated blind transcriptions are saved as part of the project file but cannot be used for research. Whichever the mode of entry into the session editor (multi-blind or direct), the interface for data entry remains identical.

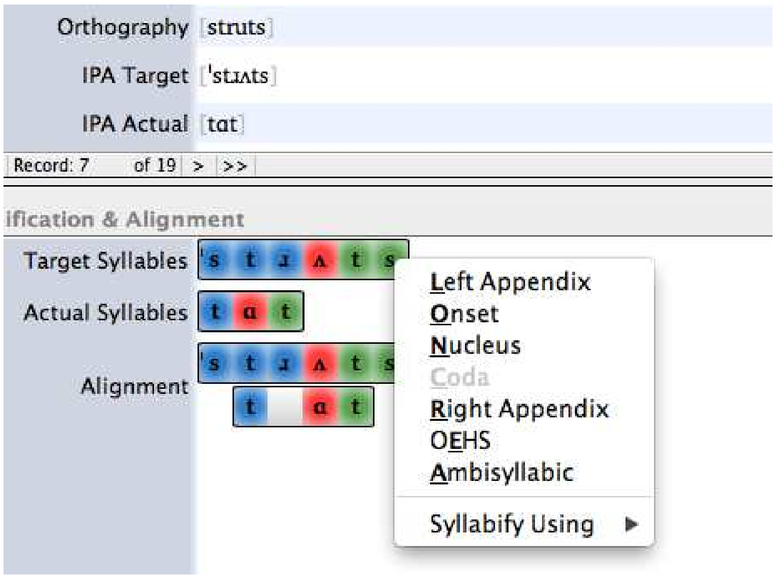
Of course, as with anything related to phonetic transcription, whichever method selected by the user is no panacea. Regardless of the amount of care put into it, and in spite of its crucial role in creating readable transcripts of spoken forms, the symbolic representation of speech sounds remains a methodological compromise. Nonetheless, at all steps involving the transcription of spoken utterances into IPA notation and/or the validation of phonetic transcripts, the user (transcriber or validator) can always export the relevant speech samples as individual sound clips for visualisation in speech analysis software programs for further assessment of the properties of the speech signal. This functionality further contributes to the attainment of the most representative data transcripts possible.

3.5 Transcribed Utterance Segmentation

Researchers often wish to divide transcribed utterances into specific domains such as the phrase or the word. Phon provides basic functionality to address this need by incorporating a text segmentation module that enables the identification of strings of symbols corresponding to such morphosyntactic and phonological domains, which we loosely call ‘word groups’. An important feature of word grouping is that, if used, it strictly enforces a logical organisation between Orthography, IPA Target and IPA Actual tiers, the latter two being treated as daughter nodes directly related to their corresponding parent bracketed form in the Orthography tier. Word groups are also supported in user-defined tiers. This system of tier dependency offers several analytical advantages, for example for the identification of patterns that can relate to a particular grammatical category or position within the utterance.

3.6 Syllabification Algorithm

Once IPA transcriptions are entered into the transcript, Phon performs syllable-level annotation automatically: segments are assigned descriptive syllable labels (visually



Orthography [struts]

IPA Target [ˈstɹʌts]

IPA Actual [tɑt]

Record: 7 of 19

Fig. 3 Syllable-level annotations

represented with colors) such as ‘onset’ or ‘coda’ for consonants and ‘nucleus’ for vowels, as can be seen in Fig. 3.

Our general approach to syllable-level annotations is based on models of syllable representation developed within generative phonology, which provide a particularly useful framework for its focus on structural description (e.g. [Selkirk, 1982, Kaye and Lowenstamm, 1984]; see [Fikkert, 1994] and [Goad and Rose, 2004] for applications to child phonology). However, because some degree of controversy exists in both phonetic and phonological theory regarding the very notion of syllabification and the types of syllable constituents allowed across formal models, the algorithm can be easily parameterized to suit various models. As can be seen in Fig. 3, descriptive models supported by Phon can be highly articulated, with positions such as initial (left) and final (right) appendices, or less refined, for example with all pre-vocalic consonants as onsets and post-vocalic ones as codas. Note as well that the use of syllable-level annotation can be used in a number of ways. On the one hand, the availability of different parameter settings makes it possible to test various hypotheses for any given dataset, for example, about formal distinctions concerning the status of on-glides in English (e.g. [Davis and Hammond, 1995]). On the other hand, labels can be used in a strictly descriptive fashion, to ease tasks such as precise data compilation, but yet have no formal implications for the researcher’s theoretical approach (and related analysis).

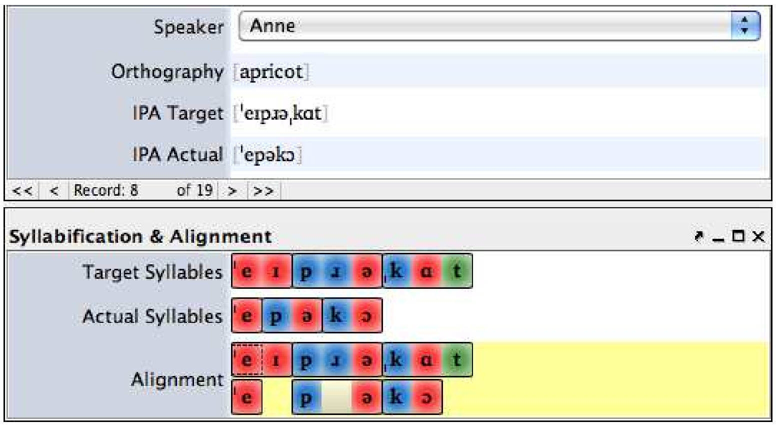
Syllabification algorithms are provided for several languages, with multiple algorithms readily available for English, French and Dutch. Additional algorithms (for other languages or based on different assumptions about syllabification) can easily be added to the program, upon request. These algorithms use a scheme based on a composition-cascade of seven deterministic FSTs (Finite State Transducers). This cascade takes as input a sequence of phones and produces a sequence of phones and associated syllable-constituent symbols, which is subsequently parsed to create the full multi-level prosodic annotation. The initial FST in the cascade places syllable nuclei. Subsequent FSTs establish and adjust the boundaries of associated syllable onset and offset domains. Changes in the definition of syllable nuclei in the initial FST and/or the ordering and makeup of the subsequent FSTs give language-specific syllabification algorithms. To ease the development of this cascade, initial FST prototypes were written and tested using the Xerox Finite-State Tool (xFST) [Beesley and Karttunen, 2003]. However, following the requirements of easy algorithm execution within and integration into Phon, these FSTs were subsequently coded in Java. To date, the implemented algorithm has been tested on corpora from English and French, and has obtained extremely high accuracy rates.

Occasionally, the algorithm may produce spurious results or flag symbols as unsyllabified. This is particularly true in the case of IPA Actual forms produced by young language learners, which sometimes contain strings of sounds that are not attested in natural languages. Since syllabification annotations are generated on the fly upon transcription entry within the IPA Target or IPA Actual tiers, the researcher can quickly verify all results and modify them through a contextual menu (represented in Fig. 3) whenever needed. Segments that are left unsyllabified are available for all queries on segmental features and strings of segments, but are not available for queries referring to aspects of syllabification.

The syllabification labels can then be used in database query (for example, to access specific information about syllable onsets or codas). In addition, because the algorithm is sensitive to main and secondary stress marks and domain edges (i.e. first and final syllables), each syllable identified is given a prosodic status and position index. Using the search functions, the researcher can thus use search criteria as precisely defined as, for example, complex onsets realised in word-medial, secondary-stressed syllables. This level of functionality is central to the study of several phenomena in phonological acquisition that are determined by the status of the syllable as stressed or unstressed, or by the position of the syllable within the word (e.g. [Inkelas and Rose, 2008]).

3.7 Alignment Algorithm

After syllabification, a second algorithm performs automatic, segment-by-segment and syllable-by-syllable alignment of IPA-transcribed target and actual forms. Building on featural similarities and differences between the segments in each syllable and on syllable properties such as stress, this algorithm automatically aligns corresponding segments and syllables in target and actual forms. It provides alignments for both corresponding sounds and syllables. For example, in the target-actual word pair ‘apricot’ $>$ ‘apico’, the algorithm aligns the phones contained in each corresponding syllable, as illustrated in Fig. 4.



Speaker: Anne
Orthography: [apricot]
IPA Target: [ˈeɪpɹəˌkɑt]
IPA Actual: [ˈepəkɔ]
<< < Record: 8 of 19 > >>

Syllabification & Alignment
Target Syllables: ˈe ɪ p ɹ ə ˌk ɑ t
Actual Syllables: ˈe p ə k ɔ
Alignment: ˈe ɪ p ɹ ə ˌk ɑ t
ˈe p ə k ɔ

Fig. 4 Phone alignment

In this alignment algorithm, forms are viewed as sequences of phones and syllable-boundary markers. Alignment is performed on the phones in a way that preserves the integrity of syllable-level annotations. This algorithm is a variant of the standard dynamic programming algorithm for pairwise global sequence alignment (see [Sankoff and Kruskal, 1983] and references therein); as such, it is similar to but extends the phone-alignment algorithm described in [Kondrak, 2003].

At the core of the Phon alignment algorithm is a function $sim(x,y)$ that assesses the degree of similarity of a symbol x from the first given sequence and a symbol y from the second given sequence. In our $sim()$ function, the similarity value of phones x and y is a function of a basic score (which is the number of phonetic features shared by x and y) and the associated values of various applicable reward and penalty conditions, each of which encodes a linguistically-motivated constraint on the form of the alignment. There are nine such reward and penalty conditions, and the interaction of these rewards and penalties on phone matchings effectively simulates syllable integrity and matching constraints. Subsequent to this phone alignment, a series of rules is invoked to reintroduce the actual and target form syllable boundaries.

A full description of the alignment algorithm is given in [Hedlund et al., 2005, Maddocks, 2005]. As it is the case with all of the other algorithms for automatic annotation included in the program, the user is able to perform manual adjustments of the computer-generated syllable alignments whenever necessary. This process was made as easy as possible: it consists of clicking on the segment that needs to be realigned and moving it leftward or rightward using keyboard arrows.

The alignment algorithm, as well as the data processing steps that precede it (especially, syllabification), are essential to any acquisition study that requires pairwise comparisons between target and actual forms, from both segmental and syllabic perspectives. Implicit to the description of the implementation of the syllabification and alignment functions is a careful approach whereby specialized algorithms are implemented in ways that facilitate data annotation; because every result generated by the algorithms can be modified by the user, no ensuing compilation of these data directly depends upon them. The user thus has complete control on the processing of the data being readied for analysis. After extensive testing on additional types of data sets, we will be able to optimize their degree of reliability and then determine how they can be used in truly automated analyses.

Once the data are processed through the modules described in the preceding subsections, they are ready to be used for research. We describe in the next section the search and reporting functions supported by Phon.

4 Database Query

The newly-introduced query system can be loosely described as a plug-in system based on JavaScript. Built-in query scripts are provided for general data query purposes. Using these scripts, the researcher can identify records that contain:

- Phones and phone sequences (defined with IPA symbols or descriptive feature sets).
- Syllable types (e.g. CV, CVC, CGV, etc. where C = consonant, V = vowel, and G = glide).
- Word types (e.g. number of syllables; stress patterns).
- IPA Target-Actual phone and syllable-level comparisons, obtained through phone and syllable alignment (e.g. phone substitutions; complex onsets reduction; syllable epenthesis).

Beyond these built-in search functions, the user can create search scripts without any need to reprogram the application. In this section, we discuss the processes of executing a query using this system, reviewing the results, and creating a report of (or an export based on) those results. This discussion also briefly covers how queries are specified using JavaScript and how to find more information about this system.

4.1 Terminology

Several terms used in the following sub-sections must first be defined

- **Query**: The set of criteria (or pattern) used to match results. Each query executed in Phon is given a unique ID.
- **Search**: The execution of a query on a particular session. Each search is also given a unique ID.
- **Search Metadata**: A set of key/value pairs which contains data particular to the search being performed.
- **Result**: A result is a single instance of the given patten found in a session.

- **Result Metadata**: A set of key/value pairs which contains data related to a result.
- **Query History**: A history of all queries performed for the current project. Queries can be 'starred' for later reference and be re-opened at later dates. The Query History window can be accessed via the Project >> Query History menu option in the Project Manager window.
- **Script Editor**: Phon provides a basic script editor for creating custom queries. The Script Editor can be accessed via the Project >> Script Editor menu option in the Project Manager window.

4.2 Executing a Query

Phon's search and report functions are separated into 3 phases:

1. Specifying Query: Query parameters are entered and the search is executed on one or more selected sessions.
2. Review Intermediate Results: All results are stored in a relational database. Result sets can be opened in Phon and modified using the Session Editor.
3. Generate Report: Results can be exported as reports into a variety of formats for use in other applications and further analysis.

Phase 1: Specifying Query Phon provides several stock searches with the installer. Each of these searches has a form which can be invoked by using the Project >> Search menu (for project-level searching) and View >> Search menu (within the Session Editor). The stock searches included with Phon are:

- **Aligned Groups**: A search for tiers which are organised into phonetic groups. This is useful for performing searches where special coding is used inside user-defined tiers which is associated with data in the Orthography or IPA tiers.
- **Aligned Phones**: This search is provided for searching patterns in the alignment data of records. Patterns are specified using Phon's phonex language for matching phone sequences.
- **CV Sequence**: Used for searching CV(G) patterns in IPA data.
- **Data Tiers**: This search is provides for generic searching of any tier.
- **Harmony**: A special function search for locating instances of harmony (consonant and/or vowel) in aligned IPA forms.
- **Metathesis**: A special function search for locating instance of metathesis in aligned IPA forms.
- **Word Shapes**: Used for searching stress patterns in words.

Each search form includes options for selecting group, word, and syllable position; syllable stress; and participant name and age, where applicable. Once the options in the form have been specified, the query can be executed on one or more sessions in the currently open project (or on the current session if initiated from the Session Editor.)

Phase 2: Review Intermediate Results Once all searches have been completed result sets can be reviewed using the Session Editor. Results are displayed in a table with corresponding metadata. Results can be removed from the result set; however such removals cannot be undone. This process is especially useful for queries which can potentially return false-positives, such as the Harmony and Metathesis queries.

Phase 3: Generate Report Query reports can come in two formats: a flat-export Comma-Separated Values (CSV) file; and a printable format which can be exported into a variety of formats. The user can choose to create a query report once project-level searches have been completed. Access to the reporting function is also available in the search list of the Query History window.

CSV reports can organise all results in either a single file or a set of files (each of which corresponds to an individual session). The user also has the option of selecting the columns they want included in the report. Columns for session data, speaker information, tier data and result data are available. This report type is most useful for inputting data into other applications (such as SPSS) for analysis.

For more detailed reporting Phon provides a configurable report template which can generate printable reports in PDF, Microsoft Word/Excel, OpenOffice, and formatted CSV. When creating these reports a default option is provided for the user. This default option can be changed by adding/removing report sections in the provided interface. Currently there are sections for printing parameters used for a query, search summary, comments, inventories, and result listings. Each section has configuration options allowing further customization of the report.

4.3 Creating a Query

Phon queries are written in a language called JavaScript. Advanced users can take advantage of having a full programming language for creating customized queries. Essentially any query can be defined using this system but there are restrictions as to what can constitute a result. The application programming interface (API) for queries can be found by using the help button in the Script Editor.

A minimal script for a Phon query implements the function `query_record(record)`. This method is executed once for each record in a session. The provided variable ‘record’ is a reference to the current record. A global variable ‘results’ is also provided for adding results to the current search’s result set. Optionally, the user can implement the functions `begin_search(session)` and `end_search(session)` which are executed at the beginning and end of a search, respectively. These methods are useful for initializing and reporting any custom global variables.

4.4 An Illustrative Example

In this section we present a concrete illustration of data representation and querying within Phon. This illustration draws on the Goad-Rose corpus of Quebec French development available through PhonBank, aspects of which are analysed in Rose (2000). First, we illustrate in Fig. 5 an early production of the name "Gaspard".

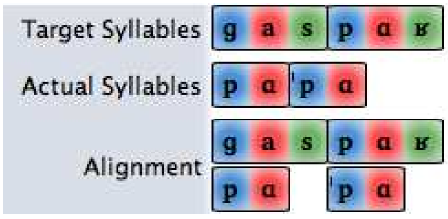
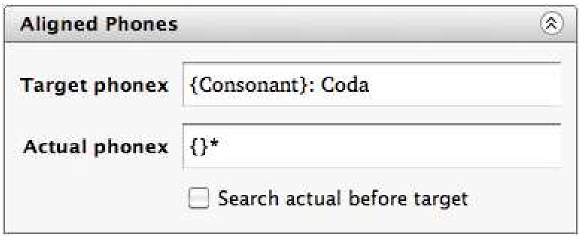


Fig. 5 Pronunciation of French name "Gaspard".

As we can see in this illustration, Phon enables the identification of segmental discrepancies between the target (model) pronunciation and its actual production by the French learner through an alignment of all IPA Target phones with their IPA Actual counterparts, thereby setting the stage for investigations of the segmental and syllabic properties of the child production (e.g. target [g] produced as [p]; deletion of both syllable-final consonants). For example, focusing on patterns of coda (syllable-final) consonant deletion, one can use the Aligned Phones search system to identify all of the relevant cases in the database, as shown in Fig 6.


Aligned Phones

Target phonex {Consonant}: Coda

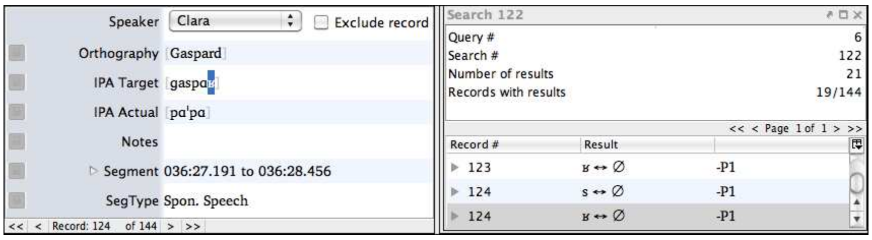
Actual phonex {}*

☐ Search actual before target

Fig. 6 Aligned Phones query: Realisation of target codas.

In this example, the user invokes the phonex language to search for target consonants in syllable codas and associated productions (or deletions) in the child's forms. As we can see in Fig. 7, the application identifies matching patterns, each of which can be visualised from the application's GUI.

As mentioned in Phase 3 of Section 4.2 above, such results can be compiled for entire recording sessions or corpora (collections of recording sessions) in the form of reports generated in various formats. Portions of Phon-generated reports are illustrated in the next two figures from a report generated in the Excel format.



Speaker	Clara
	Exclude record
Orthography	Gaspard
IPA Target	gaspaʁ
IPA Actual	pɑˈpɑ
Notes	
Segment	036:27.191 to 036:28.456
SegType	Spon. Speech

<< < Record: 124 of 144 > >>

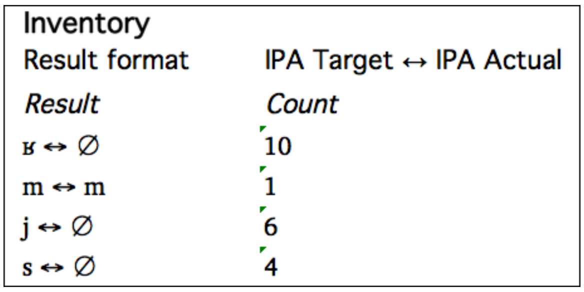
Search 122

Query #	6
Search #	122
Number of results	21
Records with results	19/144

<< < Page 1 of 1 > >>

Record #	Result	
123	ʁ ↔ ∅	-P1
124	s ↔ ∅	-P1
124	ʁ ↔ ∅	-P1

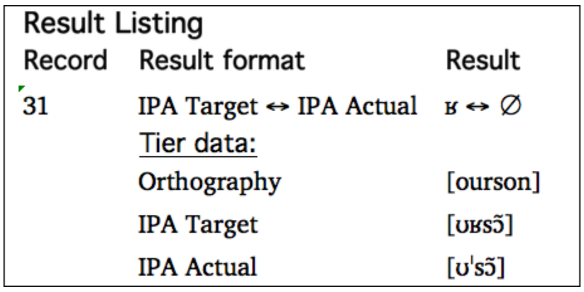
Fig. 7 Search results visualised in the Transcript Editor.



Inventory

Result format	IPA Target ↔ IPA Actual
Result	*Count*
ʁ ↔ ∅	10
m ↔ m	1
j ↔ ∅	6
s ↔ ∅	4

Fig. 8 Phon-generated data report: Inventory of results.



Result Listing

Record	Result format	Result
31	IPA Target ↔ IPA Actual	ʁ ↔ ∅
	Tier data:	
	Orthography	[ourson]
	IPA Target	[ʊʁsɔ̃]
	IPA Actual	[ʊˈsɔ̃]

Fig. 9 Phon-generated data report: Record-by-record result listing .

All such reports are also divided into specific sections. For example, as illustrated in Fig. 8, sections reporting general inventories of results are useful to observe general trends in the data. In this particular case, we can see that all continuant consonants in coda undergo deletion (e.g. all 10 instances of target rhotics in coda), while coda [m], a stop consonant, is realised in a target-like fashion.

In order to fully appreciate the extent of such patterns, the user can also study each instance of a given pattern through individual result listings such as the one illustrated in Fig. 9. While this example only contains data from three tiers (Orthography, IPA Target and IPA Actual), all tiers contained in data records can be listed in the reports.

While the above examples provide a simple illustration of the query and reporting system implemented in Phon, many more types of query and data reports can be specified by the user. Through combinations of quantitative information (as in Fig. 8) and associated qualitative characterizations (as in Fig. 9), the user can achieve the desired level of observational detail to address various types of research hypotheses.

4.5 Additional Information

More information on searching in Phon can be found in the application manual available via the Help menu. A support forum is also available, which can be accessed at http://phon.ling.mun.ca/phontrac/discussion/3. For additional query scripts visit http://phon.ling.mun.ca/phontrac/wiki/search/scriptlibrary. Finally, as mentioned in Sect. 3, support is also provided for data compilations based on particular feature sets for transcribed phones. Information on this topic can be found at http://phon.ling.mun.ca/phontrac/wiki/search/customfeatures.

5 Future Projects

As described above, Phon provides all the functionality required for corpus building as well as a versatile system for data extraction. In future versions, we plan to incorporate an interface for the management of acoustic data and fuller support for data querying and searching; the latter can, among other things, be used to create systems for testing and deriving language acquisition models. We will discuss all of these plans in the following subsections.

5.1 Interface for Acoustic Data

In order to facilitate research that requires acoustic measurements, Phon will interface fully with Praat [Boersma and Weenink, 2011], a software program designed for acoustic analysis of speech sounds. Using conduits between Praat and Phon, researchers that use these programs will be able to take advantage of some of Phon's unique functions and, similarly, researchers using Phon will be able to integrate acoustic measurements for both corpus preparation and data analysis.

We will first develop an interface within Phon for the alignment of phonetic transcriptions with their corresponding waveforms and spectrograms. This process will be semi-automated using the CSLU Toolkit (http://cslu.cse.ogi.edu/toolkit), which provides a method for aligning phonetic transcriptions with their corresponding spectrographic representations. Researchers will also be able to use similar Praat-compatible plug-ins such as EasyAlign, which can be accessed through http://latlcui.unige.ch/phonetique/. The transcription-spectrogram alignment will provide the start and end points of data measurements for each sound or sound sequence targeted by the researcher. The researcher can then activate a command to send the relevant portion of the recorded media for analysis in Praat, which can compute a wide variety of acoustic analyses, such as F0 and formant tracking and spectral analysis through FFT or LPC. After acoustic analysis, Phon will import the results into an interface that will accommodate acoustic measurement data.

The system described above will offer unprecedented support for investigations requiring the combination of refined phonological classifications and detailed acoustic characterizations of developmental data. Among other advantages, this system will offer a means to systematically verify phonetic transcriptions of recorded speech, through mapping impressionistic transcriptions with their acoustic correlates. It will also enable systematic extractions and compilations of acoustic measurements of speech sounds relative to their positions within the spoken utterance. For example, the researcher will be able to study the development of vocalic systems by simultaneously compiling longitudinal acoustic data relative to prosodic positions such as stressed versus unstressed syllables. As a result, researchers that utilize Praat will be able to take advantage of some of Phon's unique functions and, similarly, researchers using Phon will be able to take advantage of the functionality of both Praat and Phon.

5.2 Extensions of Database Query Functionality

The search and report functions described in Sect. 4.2 provide simple and flexible tools to generate general assessments of the corpus or detect and extract particular phonological patterns occurring in specified data tiers in individual sessions. However, to take full advantage of all of the research potential that Phon offers, a more powerful query system is required. The first steps towards such a system will involve modifying the existing query language to include (1) standard statistical functions and (2) methods for specifying queries that incorporate the acoustic data described in the previous subsection. This will enable precise initial assessments of developmental data within and across corpora of language learners or learning situations.

More comprehensive assessments are possible if the query mechanism is further modified to allow the specification and matching of richer types of patterns. One such type of particular interest is a pattern that describes a (possibly summarized) portion of the time-series of sessions comprising the longitudinal data for a particular language learner. Each such pattern is effectively a hypothesis about the nature and course of language acquisition. The hypotheses associated with these patterns treat linguistic phenomena of variable extent, from local features of language acquisition that occur in a particular time-interval, e.g., the so-called ‘vocabulary spurt’, to global characterizations of the whole acquisition process, e.g., the acquisition order of the target phone inventory.

Given such a pattern and a learner time-series, the degree to which the pattern matches the time-series corresponds to the degree with which the hypothesis encoded by that pattern is consistent with and hence supported by that time-series.

Alternatively, two learner time-series could be matched against each other to assess their similarity and hence the degree to which those learners are following the same developmental path. Such matches can be done with variants of the target-actual form alignment function described in Sect. 3.7. Many types of time-series pattern matching have been defined and implemented within computational molecular biology (see [Sankoff and Kruskal, 1983, Gusfield, 1997] and references) and temporal data mining (see [Roddick and Spiliopoulou, 2002, Keogh, 2008, Mitsa, 2010] and references); it seems likely that some of these can be either used directly or modified for the purposes of language acquisition research.[2]

While useful in itself, such a time-series pattern matching capability is the building block of even more exotic analyses, e.g.,

- **Language Acquisition Model Testing**: Given appropriate formalizations of language acquisition models as algorithms that produce ‘actual’ output analogous to that produced by learners, such models can be automatically evaluated against learner time-series stored in Phon (in a manner analogous to the ’Learn’ function in Praat) using functions such as:
  - Run an arbitrary language learning algorithm.
  - Compare the results of the grammar produced by such a language learning algorithm against actual language data.
  - In the event that the learning algorithm provides a sequence of grammars corresponding to the stages of human language learning, compare the results of this sequence of grammars against actual longitudinal language data.

  By virtue of its software architecture, form-comparison routines, and stored data, Phon provides an excellent platform for implementing such an application. Running arbitrary language learning algorithms can be facilitated using a Java API/interface-class combination specifying subroutines provided by Phon, and the outputs of a given model could be compared against target productions stored in Phon using either the alignment algorithm described in Sect. 3.7 or the more general time-series pattern matching algorithms described above.
- **Language Acquisition Model Derivation**: Consider applying time-series pattern matching as described above in reverse – namely, given a set of two or more learner time-series, find those patterns that are best supported by and hence characteristic of the those time-series. Such patterns may be used as developmental benchmarks for deriving language acquisition models or (in the case of very rich types of time-series patterns) function as models themselves.

[2] Previous experience in computational molecular biology and data mining suggests that, given the large amounts of data involved, various specialized algorithmic techniques will probably have to be invoked to allow time-series pattern matching to run in practical amounts of time and computer memory. The typical approach described in [Keogh, 2008] is to simplify the given data, derive approximate analysis-results relative to this simplified data, and (hopefully with minimal effort) reconstruct exact analysis-results relative to the original data. However, there may be other options, such as using so-called fixed-parameter tractable algorithms [Downey and Fellows, 1999, Niedermeier, 2006] whose running times are impractical in general but efficient under the restrictions present in learner time-series datasets.

Analyses such as those sketched above would be much more comprehensive than what has been the norm thus far in the field, especially given past problems encountered in verifying let alone deriving trustworthy models of language acquisition relative to small (and possibly wildly unrepresentative) sets of learners. However, given the diversity of analysis techniques available within computational molecular biology and data mining in general, providing a platform like Phon for implementing such analyses may have the (perhaps ultimately more important) long-term effect of introducing previously unimagined analytical possibilities and related research opportunities.

6 Discussion

Phon offers a sound computational foundation for the management of corpus-based research on phonology and phonological development, media linkage and segmentation into transcript-annotated time intervals, multiple-blind IPA transcription, IPA transcription validation, target (adult) IPA form insertion, automatic phone alignment between target (model) and actual (produced) forms, automatic syllabification, utterance segmentation into smaller units, database query, data import, and data export. Finally, it provides a strong computational foundation for the implementation of additional functions.

Beyond acoustic data analysis capabilities, the order in which new functionalities will be implemented in future versions of Phon is still unclear. For example, the model-testing tool sketched in Sect. 5.2 is ambitious and perhaps premature in some respects, e.g., should we expect the current (or even next) generation of language learning algorithms to mimic the longitudinal behavior of actual language learners? Such issues are especially relevant, given that some language behaviors observed in learners can be driven by articulatory or perceptual factors, the consideration of which implies relatively more complex models. That being said, the above suggests how Phon, by virtue of its longitudinal data, output-form comparison routines, and software architecture, may provide an excellent platform for implementing the next generation of computational language analysis tools.

Acknowledgements We would like thank the co-organisers of the original ACL workshop (namely, Afra Alishahi, Thierry Poibeau, Anna Korhonen and Aline Villavicencio) for their help and support through all the steps that brought us to this publication and Carla Peddle for assistance in preparing the final version presented here. We are also grateful to two anonymous reviewers for their useful feedback. Current development of Phon and PhonBank is supported by the National Institute of Health. Earlier development of Phon was funded by grants from National Science Foundation, Canada Fund for Innovation, Social Sciences and Humanities Research Council of Canada, Petro-Canada Fund for Young Innovators, and the Office of the Vice-President (Research) and the Faculty of Arts at Memorial University of Newfoundland. TW would also like to acknowledge support provided through NSERC Discovery Grant 228104.

References

[Beesley and Karttunen, 2003] Beesley, K.R. and Karttunen, L. (2003). *Finite-State Morphology*. CSLI Publications, Stanford, CA.

[Boersma and Weenink, 2011] Boersma, P. and Weenink, D. (2011). Praat: doing phonetics by computer [Computer program]. Version 5.2.18, retrieved 10 March 2011 from http://www.praat.org/

[Davis and Hammond, 1995] Davis, S. and Hammond, M. (1995). On the Status of Onglides in American English. *Phonology*, 12: 159–182.

[Downey and Fellows, 1999] Downey, R.G. and Fellows, M.R. (1999). *Parameterized Complexity*. Springer, New York.

[Fikkert, 1994] Fikkert, P. (1994). *On the Acquisition of Prosodic Structure*. ICG Printing, Dordrecht.

[Goad and Rose, 2004] Goad, H. and Rose, Y. (2004). Input Elaboration, Head Faithfulness and Evidence for Representation in the Acquisition of Left-edge Clusters in West Germanic. In: R. Kager, J. Pater, and W. Zonneveld (eds.). *Constraints on Phonological Acquisition*, pages 109–157. Cambridge University Press.

[Gusfield, 1997] Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

[Hedlund et al., 2005] Hedlund, G.J., Maddocks, K., Rose, Y., and Wareham, T. (2005). Natural Language Syllable Alignment: From Conception to Implementation. In: *Proceedings of the Fifteenth Annual Newfoundland Electrical and Computer Engineering Conference (NECEC 2005)*.

[Inkelas and Rose, 2008] Inkelas, S. and Rose, Y. (2008). Positional Neutralization: A Case Study from Child Language. *Language*, 83: 707–736.

[Kaye and Lowenstamm, 1984] Kaye, J. and Lowenstamm, J. (1984). De la syllabicité. In: *Forme sonore du langage*, pages 123–161. Hermann, Paris.

[Keogh, 2008] Keogh, E. (2008). Indexing and Mining Time Series Data. In: S. Shekhar and H. Xiong (eds.) *Encyclopedia of GIS*, pages 493–497. Springer, New York.

[Kondrak, 2003] Kondrak, G. (2003). Phonetic alignment and similarity. *Computers in the Humanities*, 37: 273–291.

[Ladefoged and Maddieson, 1996] Ladefoged, P. and Maddieson, I. (1996). *The Sounds of the World's Languages*. Blackwell, Cambridge, MA.

[Maddocks, 2005] Maddocks, K. (2005). *An Effective Algorithm for the Alignment of Target and Actual Syllables for the Study of Language Acquisition*. B.Sc.h. Thesis. Memorial University of Newfoundland.

[Mitsa, 2010] Mitsa, T. (2010). *Temporal Data Mining*. Chapman and Hall/CRC, Boca Raton, FL.

[Niedermeier, 2006] Niedermeier, R. (2006). *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.

[Roddick and Spiliopoulou, 2002] Roddick, J.F. and Spiliopoulou, M. (2002). A Survey of Temporal Knowledge Discovery Paradigms and Methods. *IEEE Transactions on Knowledge and Data Engineering*, 14: 750–767.

[Rose, 2000] Rose, Y. (2000). *Headedness and Prosodic Licensing in the L1 Acquisition of Phonology*. Ph.D. Dissertation. McGill University.

[Rose et al., 2006] Rose, Y., MacWhinney, B., Byrne, R., Hedlund, G.J., Maddocks, K., O'Brien, P., and Wareham, T. (2006). Introducing Phon: A Software Solution for the Study of Phonological Acquisition. In: *Proceedings of the 30th Boston University Conference on Language Development*, pages 489–500. Cascadilla Press, Somerville, MA.

[Sankoff and Kruskal, 1983] Sankoff, D. and Kruskal, J.B. (eds.) (1983). *Time Warps, String Edits, and Macromolecules: The Theory and Practice of String Comparison*. Addison-Wesley, Reading, MA.

[Selkirk, 1982] Selkirk, E. (1982). The Syllable. In: *The Structure of Phonological Representation*, pages 337–385. Foris, Dordrecht.

[Selkirk, 1986] Selkirk, E. (1986). On Derived Domains in Sentence Phonology. *Phonology*, 3: 371–405.