Phon 1.2: A Computational Basis for Phonological Database Elaboration and Model Testing

Yvan Rose[1], Gregory J. Hedlund[1], Rod Byrne[2], Todd Wareham[2], Brian MacWhinney[3]

[1]Department of Linguistics
Memorial University of Newfoundland

[2]Department of Computer Science
Memorial University of Newfoundland

[3]Department of Psychology
Carnegie Mellon University

yrose@mun.ca, ghedlund@cs.mun.ca, rod@cs.mun.ca, harold@cs.mun.ca, macw@cmu.edu

Abstract

This paper discusses a new, open-source software program, called Phon, that is designed for the transcription, coding, and analysis of phonological corpora. Phon provides support for multimedia data linkage, segmentation, multiple-blind transcription, transcription validation, syllabification, alignment of target and actual forms, and data analysis. All of these functions are available through a user-friendly graphical interface. Phon, available on most computer platforms, supports data exchange among researchers with the TalkBank XML document format and the Unicode character set.. This program provides the basis for the elaboration of PhonBank, a database project that seeks to broaden the scope of CHILDES into phonological development and disorders.

1 Introduction

Empirical studies of natural language and language acquisition will always be required in most types of linguistic research. These studies provide the basis for describing languages and linguistic patterns. In addition to providing us with baseline data, empirical data allow us to test theoretical, neurological, psychological and computational models. However, the construction 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/), researchers in areas such as morphology and syntax have enjoyed a convenient and powerful method to analyze the morphosyntactic properties of adult 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 conduct phonological modeling, computational simulations based on a variety of theoretical approaches, and articulatory synthesis.

In this rapidly-expanding software universe, phonologists interested in the organization of sound systems (e.g. phones, syllables, stress and intonational patterns) and their acquisition have not yet enjoyed the same level of computational support. There is no developed platform for phonological analysis and no system for data-sharing parallel to that found in CHILDES. Unfortunately, this situation negatively affects the study of natural language phonology and phonological development. It also undermines potential studies pertaining to interfaces between various components of the grammar or the elaboration of computational models of language or language development.

It is largely accepted that the grammar is hierarchically organized such that larger domains (e.g. a sentence or a phrase) provide the conditioning environments for patterns occurring in the domains

Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition, pages 17–24,
Prague, Czech Republic, June 2007 ©2007 Association for Computational Linguistics

located lower in the hierarchy (e.g. the word or the syllable), as indicated in Figure 1.

Figure 1: General grammatical hierarchy

This hierarchical view of grammatical organization allows us to make reference to factors that link phonology to syntax. For example, in English, the phonological phrase, a domain that constrains phonological phenomena such as intonation, is best described using syntactic criteria (e.g. Selkirk 1986). Data on the acquisition of these grammatical structures and their phonological consequences can help us understand how they are learned and assimilated by the learner.

In this paper we discuss Phon 1.2, the current version of an 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 elaboration and analysis. On the other hand, its ability to integrate with other open-source software will facilitate the construction of complete analyses across all levels of grammatical organization represented in Figure 1.

The paper is organized as follows. In section 2, we discuss the general motivation behind the Phon project. In section 3, we discuss the current functionality supported in Phon 1.2. In section 4, we offer a glance at future plans for this project. Section 5 provides a final summary.

2 The PhonBank Project

PhonBank, the latest initiative within the CHILDES project, focuses on the construction of corpora suitable for phonological and phonetic analysis. In this section we first describe the goals and orientations of PhonBank. We then describe Phon, the software project 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 second languages for language learners with and without language disorders. To achieve this goal, we will create a new phonological database called PhonBank and a program called Phon to facilitate analysis of PhonBank data. Using these tools, researchers will be in 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 Macintosh (Mac OS X 10.4+) and Windows (Vista not tested yet) platforms.[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 is available as free download directly from CHILDES (http://childes.psy.cmu.edu/phon/).

At the time of writing these lines, Phon is available in its version 1.1, an iteration of the program that offered a proof of concept for the application envisioned (see Rose et al., 2006). Over the past year, however, 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 workflow. Despite what the minor version increment (1.1 to 1.2) may imply, the new version, which is currently being tested internally and due for public release in June 2007, offers significant improvements as well as novel and innovative functionality.

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

3 Phon 1.2

As illustrated in Figure 2, the general interface of Phon 1.2 consists of a media centre (top left of the interface), a section for metadata (e.g. recorded participants and their linguistic profiles; bottom left) and a Transcript Editor, the interface that provides access to most of the functionality (right).

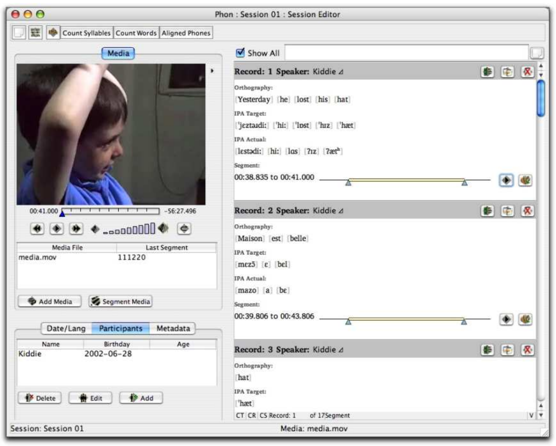

Figure 2: Phon 1.2 General Interface

One of the most significant improvements brought to version 1.2 comes from the integration of common tasks within the same user interface. In the previous version, completely separate interfaces had to be accessed to achieve the following tasks, all of which are required in the elaboration of any corpus:

- 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.

As a result the user often had to navigate between various modules in order to accomplish relatively simple operations. For example, a simple modification to a transcription required, in addition to the modification itself, revalidation of the data, and then a verification of the syllabification and alignment data generated from this revised transcription, each of these steps requiring access to and subsequent exit from a separate module.

In Phon 1.2, most of this hurdle has been alleviated through an integration of most of the functions into the Transcript Editor, while the others (e.g. media linkage and segmentation; transcript validation) are accessed directly from the general interface, without a need to exit the Transcript Editor. In the next subsections, we describe the main functions supported by the application.[2]

3.1 Media linkage and segmentation

As mentioned above, 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/). In addition to its integrated interface, Phon 1.2 offers support for linking different portions to a single transcript to different media files.

3.2 Data transcription

The Transcript Editor now incorporates in a single interface access to data transcription and annotation, transcription segmentation, syllabification and alignment. This module is illustrated in more detail with the screen shot of a data record (corresponding to an utterance) in Figure 3.

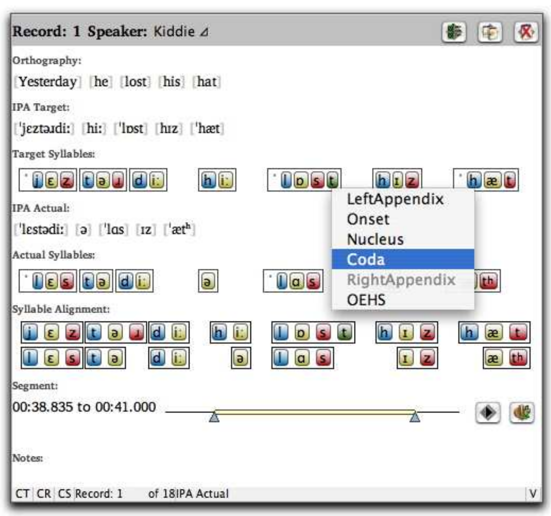

Figure 3: Data record in Transcript Editor

[2] Additional functions, such as user management, are also supported by Phon; we will however restrict ourselves to the most central functions of the program.

As can be seen, the interface incorporates tiers for orthographic and phonetic transcriptions as well as other textual annotations. Phon also provides support for an unlimited number of user-defined fields that can be used for all kinds of textual annotations that may be relevant to the coding of a particular dataset. All fields can be ordered to accommodate specific data visualization needs. Phonetic transcriptions are based on the phonetic symbols and conventions of the International Phonetic Association (IPA). A useful IPA character map is easily accessible from within the application, in the shape of a floating window within which IPA symbols and diacritics are organized into intuitive categories. This map 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 as needed to fit special research needs.

Phon 1.2 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 English and French. The English forms were obtained from the CMU Pronouncing Dictionary (www.speech.cs.cmu.edu/cgi-bin/cmudict); the French forms were obtained from the Lexique Project database (www.lexique.org).

In cases when more than one pronunciation 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 wanted form.

Of course, idealized citation forms do not provide accurate fine-grained characterizations 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.

1.1 Media playback and exporting

Actual forms (e.g. the forms produced by a language learner) must be transcribed manually. Transcript validation, the task described in the next section, also requires access to the recorded data. To facilitate these tasks, Phon provides direct access to the segmented portions of the media for playback in each record (see the 'Segment' tier in Figure 3). The beginning and end times of these segments can be edited directly from the record, which facilitates an accurate circumscription of the relevant portions of the recorded media. Finally, Phon can export the segmented portions of the media into a sound file, which enables quick acoustic verifications using sound visualizing software such as Praat (http://www.fon.hum.uva.nl/praat/), SFS (http://www.phon.ucl.ac.uk/resource/sfs/), Signalyze (http://www.signalyze.com/) or CSL (http://www.kayelemetrics.com/).

1.2 Transcript validation

In projects where only a single transcription of the recorded data is utilized, this transcription can be entered directly in the Transcript Editor. In projects that rely on a multiple-blind transcription method, each transcription for a given form is stored separately. To appear in the Transcript Editor, a blind transcription must be selected through the Transcript Validation mode. This interface allows the transcription supervisor (or, in a better setting, a team of supervisors working together) to compare competing transcriptions and resolve divergences. Alternative, non-validated transcriptions are preserved for data recoverability and verification purposes. They are however unavailable for further processing, coding or analysis.

1.3 Transcription segmentation

Researchers often wish to divide transcribed utterances into specific domains such as the phrase or the word. Phon fulfills this need by incorporating a text segmentation module that enables the identification of strings of symbols corresponding to such morphosyntactic and phonological domains. For example, using the syllabification module described immediately below, the researcher can test hypotheses about what domains are relevant for resyllabification processes across words. Word-level segmentation is exemplified in Figure 3, as can be seen from the gray bracketing circumscribing each word. Not readily visible from this interface however is the important fact that the bracketing enforces a logical organization between Orthographic, IPA Target and IPA Actual forms, the latter two being treated as daughter nodes directly related to their corresponding parent bracketed form in the Orthography tier. 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.

In addition to the textual entry fields just described, the Transcript Editor contains color-coded graphical representations of syllabification information for both IPA Target and IPA Actual forms as well as for the segmental and syllabic alignment of these forms.

1.4 Syllabification algorithm

Once the researcher has identified the domains that are relevant for analysis, segmentation at the level of the syllable is performed automatically: segments are assigned descriptive syllable labels (visually represented with colors) such as 'onset' or 'coda' for consonants and 'nucleus' for vowels. The program also identifies segmental sequences within syllable constituents (e.g. complex onsets or nuclei). Since controversy exists in both phonetic and phonological theory regarding guidelines for syllabification, the algorithm is parameterized to allow for analytical flexibility. The availability of different parameter settings also enables the researcher to test hypotheses on which analysis makes the best prediction for a given dataset. Phon 1.2 contains built-in syllabification algorithms for both English and French. The algorithm for English incorporates fine distinctions such as those proposed by Davis and Hammond (1995) for the syllabification of on-glides. Both algorithms are based on earlier work by, e.g. Selkirk (1982) and Kaye and Lowenstamm (1984), the latter also documenting the most central properties of French syllabification. While these algorithms use specific syllable positions such as the left appendix (utilized to identify strident fricatives at the left-edge of triconsonantal onset clusters; e.g. 'strap'), a simple syllabification algorithm is also supplied, which restricts syllable position to onset, nucleus and coda only. Additional algorithms (for other languages or assuming different syllable constructs) can easily be added to the program.

Our currently-implemented syllabification algorithms use a scheme based on a composition-cascade of seven deterministic FSTs (Finite State Tools). 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 metrical structure. The initial FST in the cascade places syllable nuclei and the subsequent FSTs establish and adjust the boundaries of associated onset- and coda-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 accuracies of almost 100%.

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. Syllabification is generated on the fly upon transcription of IPA forms; the researcher can thus quickly verify all results and modify them through a contextual menu (represented in Figure 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 (see also Figure 4 for a closer look at the display 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 realized 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 2003).

1.5 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' > 'a_cot', the algorithm aligns the first and final syllables of each form, and identifies the middle syllable ('pri') as truncated. This is illustrated in Figure 4. Similarly, in cases of renditions such as 'blow' > 'bolow' the alignment algorithm relates both syllables of the actual form to the only syllable of the target form and diagnoses a case of vowel epenthesis.

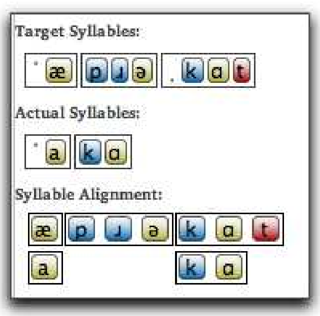

Figure 4: Syllabification and Alignment

In this alignment algorithm, forms are viewed as sequences of phones and syllable-boundary markers and the alignment is done on the phones in a way that preserves syllable integrity. 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 enhanced 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 Maddocks (2005) and Hedlund et al. (2005). Preliminary tests on attested data from the published literature on Dutch- and English-learning children (Fikkert, 1994; Pater, 1997) indicate an accuracy rate above 95% (96% for a Dutch corpus and 98% for an English corpus). As it is the case with the other algorithms 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 pair-wise 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 the algorithms implemented at this stage are used to assist data compilation; because every result generated by the algorithms can be modified by the user, no data analysis directly depends on 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 determined how they can be used in truly automated analyses.

1.6 Database query

Phon sports a simple search function built directly in the main interface (see Figure 2 above). More complex queries are now supported through a series of built-in analysis and reporting functions.

Using these functions, the research can identify records that contain:

- Phones and phone sequences (defined with IPA symbols or descriptive feature sets).
- Syllable types (e.g. CV, CVC, CGV, …).[3]
- Word types (e.g. number of syllables and the stress patterns that they compose).
- Segmental processes (obtained through featural comparisons between Target-Actual aligned phones; e.g. devoicing, gliding).
- Syllabic processes (obtained through comparisons between target-actual aligned syllables e.g. complex onset reduction).

Using these functions, the researcher can quickly identify the records that match the search criteria within the transcript. The reported data are visualized in tables which can be saved as comma-separated value text files (.csv) that can subsequently be open in statistical or spreadsheet applications. Using an expression builder, i.e. a system to combine simple searches using functions such as intersection and union, the researcher can also take advantage of more elaborate search criteria. The expression builder thus enables the study of interaction between factors such as feature combinations, stress, position within the syllable, word or any other larger domain circumscribed through the utterance segmentation function described above.

2 Future projects

Phon 1.2 now provides all the functionality required for corpus elaboration, as well as a versatile system for data extraction. In future versions, we will incorporate an interface for the management of acoustic data and fuller support for data querying and searching. At a later stage, we will construct a system for model testing. We discuss these plans briefly in the next subsections.

2.1 Interface for acoustic data

In order to facilitate research that requires acoustic measurements, Phon will also incorporate full interfacing with Praat and Speech Filing System, two software programs designed for acoustic analysis of speech sounds. As a result, researchers that utilize these programs 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 these two applications.

2.2 Extension of database query functionality

The search and report functions described in section 3.8 provide simple and flexible tools to generate general assessments of the corpus or detect and extract particular phonological patterns. However, to take full advantage of all of the research potential that Phon offers, a more powerful query system will be designed. This system will take the form of a query language supplemented with statistical functions.

Such a system will enable precise assessments of developmental data within and across corpora of language learners or learning situations. The query language will also offer the relevant functionality to take full advantage of the module for management of acoustic data described in the preceding subsection.

2.3 Platform for model testing

As presently implemented, Phon will allow us to continue with the construction of PhonBank and will provide tools for analyzing the new database. Once this system is in place, we will begin to develop additional tools for model testing. These new systems will formalize learning algorithms in ways that will allow users to run these algorithms on stored data, much as in the "Learn" feature in Praat. This new model-testing application will include 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 learn-

[3] C=consonant; V=vowel; G=glide.

ing algorithms could be facilitated using a Java API/interface-class combination specifying subroutines provided by Phon. The outputs of a given computational model could be compared against adult productions stored in Phon using the alignment algorithm described in Section 3.7 (which internally produces but does not output a score giving the similarity of the two forms being aligned). Finally, the outputs of a sequence of algorithm-produced grammars relative to a given target word could be compared against the sequence of productions of that word made over the course of acquisition by a particular learner by aligning these production sequences. Such an alignment could be done using the alignment algorithm described in Section 3.7 as a *sim()* function for matching up production-pairs in these sequences. In this case, more exotic forms of alignment such as local alignment or time-warping may be more appropriate than the global alignment used in Section 3.7. For a full description of such alignment options, see Gusfield (1997) and Sankoff and Kruskal (1983).

3 Discussion

In its current form, Phon 1.2 provides a powerful system for corpus transcription, coding and analysis. It also offers a sound computational foundation for the elaboration of the PhonBank database and its incorporation to the CHILDES system. Finally, it sets the basis for further improvements of its functionality, some of which was discussed briefly in the preceding section.

The model-testing tool design sketched above is ambitious and perhaps premature in some aspects —for example, should we expect the current (or even next) generation of language learning algorithms to mimic the longitudinal behavior of actual language learners? This question is 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.

References

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

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

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

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

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

Inkelas, S. and Y. Rose (2003). Velar Fronting Revisited. *Proceedings of the 27th Boston University Conference on Language Development*. Somerville, MA: Cascadilla Press. 334-345.

Kaye, J. and J. Lowenstamm (1984). De la syllabicité. *Forme sonore du langage*. Paris: Hermann, 123-161.

Kondrak, G. (2003) Phonetic alignment and similarity. *Computers and the Humanities* 37: 273-291.

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

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

Pater, J. (1997). Minimal Violation and Phonological Development. *Language Acquisition* 6, 201-253.

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

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

Selkirk, E. (1982) The Syllable. *The Structure of Phonological Representation*. Dordrecht: Foris, 337-385.

___ (1986) On Derived domains in Sentence Phonology. *Phonology* 3: 371-405.