Multilingual phonological corpus analysis

The tools behind the PhonBank Project

Yvan Rose

In this chapter, I describe the PhonBank database initiative within the larger CHILDES project. After a brief introduction to these inter-related database systems, I move the focus on the types of corpus annotations and analyses which we support within PhonBank, through the Phon software program. Phon greatly facilitates a number of tasks required for the analysis of phonological development. It supports multimedia data linkage, unit segmentation, multiple-blind transcription, automatic labelling of data, and systematic comparisons between target (model) and actual (produced) phonological forms. Building on this description, I then provide a practical illustration in the context of a multilingual study of phonology, taking as an example the analysis of data on the phonological adaptation of linguistic borrowings (loanwords).

1. Introduction

Current questions about language, its use, acquisition, and related disorders can nowadays be tested on empirical grounds that are firmer than ever. Among other developments, the democratization of computer-assisted approaches to linguistic analysis and the fostering of more forward-thinking attitudes toward data sharing have led new and exciting research opportunities. These advances, and the outcomes they yield, can in turn be used to feed more traditional but still fundamental debates in theoretical linguistics (e.g. contributions to Ewen et al. 2011 in the area of phonology). From a more practical standpoint, modern approaches to corpus-based linguistic studies call for robust systems for databasing and analyzing broad corpora of linguistic data. The CHILDES project (Child Language Data Exchange System; <http://childes.psy.cmu.edu/>) which has, over the last 25 years, provided the grounds for thousands of publications in areas ranging from language acquisition to computational linguistics, provides a model example of this.

In this chapter, I describe a recent initiative within CHILDES, the PhonBank project, and discuss how the tools developed to support this project can be used for multilingual, corpus-based research in phonology. I begin in Section 2 with a brief overview of the PhonBank project. In Section 3, I describe the main functions supported by Phon, the software program designed for the transcription and analysis of PhonBank corpora. In Section 4, I provide a practical illustration of Phon's data compilation functions, based on a corpus of French loanwords adapted in Kinyarwanda. I conclude in Section 5 with a brief section on future developments and related implications for research.

2. PhonBank

The PhonBank project broadens the scope of the CHILDES system to include the analysis of phonological development in first and subsequent languages for learners with and without language disorders.[1]

Since its inception in 2006, PhonBank has contributed computational and databasing facilities for corpus-based studies of phonological development, many of which can also be used in other areas of corpus phonology (e.g. Durand et al. to appear), including fieldwork and multilingual studies. Using these tools, researchers can perform systematic investigations based on large, fully transcribed corpora of speech data, a number of which also incorporate audio or video recordings. In the next section, I focus on the most central functions supported by Phon for corpus-based studies in phonology.

3. Phon

Phon is a free and open-source program for phonological corpus building and analysis. Developed in Java and packaged to run on Mac OS X, Windows and Unix/Linux platforms, Phon is fully Unicode-compliant, a feature required for the sharing of data transcribed with phonetic or other specialized symbols across these platforms. The main interface for data transcription and annotation in Phon is illustrated in Figure 1. It consists of a series of view panels, each of which

1. Current development of Phon and PhonBank is supported by the National Institute of Health. Earlier development of Phon was funded by grants from the National Science Foundation, the Canada Fund for Innovation, the Social Sciences and Humanities Research Council of Canada, the Petro-Canada Fund for Young Innovators as well as the Office of the Vice-President (Research) and the Faculty of Arts at Memorial University of Newfoundland.

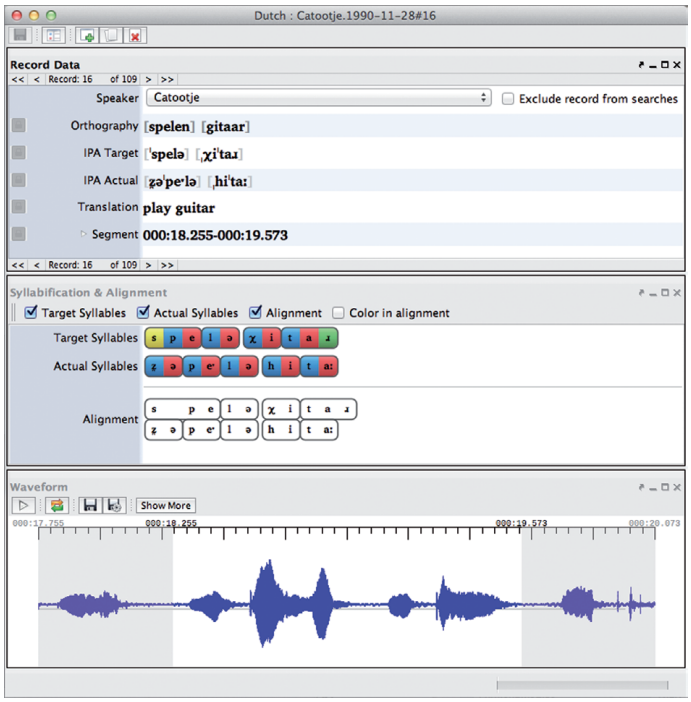


Figure 1. Phon's Session Editor: general interface

supports particular functions (e.g. orthographic and phonetic transcription; waveform visualization).

This general interface is fully configurable. View panels can be added or removed from the interface; view panels can also be docked at the bottom of the application's window or superposed within tabbed interfaces. The user can thus personalize the interface to streamline Phon's functions, many of which are described in the next subsection.

3.1 Main functions

In this section, I describe a number of functions provided by Phon for phonological corpus building and analysis. In the interest of clarity, I introduce the functions in an order that generally corresponds to the building of a corpus from the ground up. Note however that Phon does not impose any particular workflow. Also, while more detailed descriptions of Phon are available elsewhere (e.g. Rose 2010; Rose & MacWhinney to appear), the content below offers an up-to-date look at Phon and serves as a basis for the ensuing section, which highlights improvements brought in version 1.5 of the application.

3.1.1 *Project management*

Corpora documenting phonology or phonological development typically consist of the combination of a number of data transcripts, for example from a single learner over a given period of time (longitudinal) or from multiple groups of learners (cross-sectional). Independent of any particular research protocol, Phon offers an interface to create and manage sets of data transcripts. The Project Manager is illustrated in Figure 2.

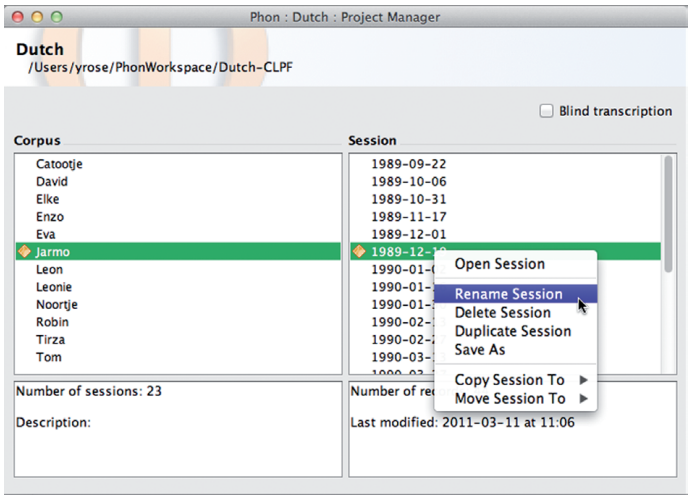


Figure 2. Project Manager

As illustrated, a Phon project contains one or many corpora (listed in the left column), each of which consists of one or many data transcripts (listed in the right column). A transcript typically corresponds to a data recording (or gathering) session or part thereof.[2] Corpus transcripts can be managed from within the Project Manager, through options available from a useful contextual menu.

3.1.2 *Segmentation (time alignment)*

One of the first tasks in the building of a transcribed corpus typically consists of the identification of the portions of the recording which must be transcribed for later analysis. Phon offers functions to associate transcribed data with the time intervals they correspond to within a recorded (audio or video) media file. This file

2. Note that Phon does not require media files. Diary or fieldwork notes can be typed or imported into Phon without any need for associated media recordings. Also, as it is the case of many functions within Phon, one can imagine other uses, for example the transcription of elicited lists of pronounced forms whereby each corpus could correspond to a given speaker or dialect. Similarly, sessions within a corpus could be used to separate a given set of forms into individual groupings, depending on specific research needs.

is never edited by Phon; only time intervals are recorded by the application, each of which corresponds to a speech segment assigned to an individual data record. After they are identified, media segments can be played back directly from both the Record Data and the Waveform view panels within the Session Editor for all tasks related to data transcription, annotation and related verifications. Whenever needed, the user can fine-tune the start and/or end time values of media segments using functions available in each of these panels. Phon also supports the exporting of individual media segments from the original media file into various audio or video formats. Again here, the original file is not edited; the audio/video data contained within the exported segment is extracted from it into a separate file. The exported clips can then be used in third-party software for further analysis or presentation purposes.

3.1.3 *Data transcription*

As was illustrated in Figure 1, the Session Editor incorporates support for orthographic and phonetic data transcription, other textual annotations as well as phone syllabification and alignment (the latter two are discussed in separate subsections below). Phon also supports user-defined data tiers. Built-in and user-defined tiers can be ordered to accommodate specific data visualization needs. Support for tier-specific fonts and typefaces is also provided, a feature particularly useful for work on languages written with non-Roman orthographies. Phonetic transcriptions follow the symbols and conventions of the International Phonetic Association (IPA). An IPA character chart built into the application facilitates access to IPA symbols which cannot be typed through regular keyboard input.

IPA transcriptions are stored as strings of phonetic symbols and diacritics, each of which is automatically associated with a set of descriptive features (e.g. 'Voiced'; 'Velar'; 'Stop'). These features are central for data identification based on natural classes of phones. For example, the consonants [p, t, k] can be simultaneously identified with the features 'Voiceless' and 'Stop' (see Section 4.3 for a concrete example).

Phon is also equipped with functionality to automatically insert IPA citation forms, based on electronic dictionaries of pronounced forms. While the IPA forms provided by these dictionaries typically do not provide fine-grained characterizations of particular languages or dialects, they do offer useful general baselines. When used in appropriate ways, these dictionaries can speed up transcription-related tasks as well as preliminary corpus investigations.

3.1.4 *Multiple-blind transcription and transcript validation*

Phon also offers a fully-integrated system to support multiple-blind, consensus-based approaches to IPA transcription. Each IPA transcriber involved in a

multiple-blind transcription project must log into the Session Editor using a specific username. Upon login, a transcriber can visualize all of the regular corpus data tiers. The crucial exception to this is that the IPA Target and IPA Actual transcription tiers are unique to each transcriber: a given transcriber can see his/her own IPA transcriptions, and thus remains 'blind' to those of other transcribers.

Blind IPA transcriptions performed in this setting are stored within the corpus transcript files but cannot be accessed by Phon's other functions at this stage. The next step in this workflow consists of consensus-based transcript validation. Using the Transcript Validation view panel, a team of two (or more) transcript validators can listen to speech segments and simultaneously visualize all corresponding transcriptions produced by the blind transcribers. The validation team then selects the transcription deemed the most accurate. Whenever necessary, the selected transcription can be modified to address details noticed by the transcript validators.

Multiple-blind transcription (and its associated validation process) is optional. If the user decides not to perform blind transcriptions, then IPA transcriptions are entered directly into the transcript and, thus, do not require validation. In case the researcher wants to quickly transition out of the blind transcription workflow, the 'Auto-validate' function can be used to batch-convert a blind transcript into a validated one. Among other benefits, this function is useful for quick assessments of blindly-performed transcription data.

3.1.5 *Word grouping within the utterance*

In Phon, a data record typically documents a spoken utterance or speech turn. In the context of multiple-word utterances, Phon provides basic functionality to divide the transcribed string into smaller subsets (or 'word groups'), which can correspond to specific domains such as phrases or individual words, depending on the researcher's needs. Word grouping enforces a logical organization between the Orthography, IPA Target and IPA Actual tiers, the latter two are daughter tiers under their parent Orthography tier.[3] Word groups, generated from the Orthography tier, are fully aligned across all daughter tiers. This system of tier dependency enables the identification of patterns specific to certain grammatical categories or positions within the utterance. (Grammatical category labels can be assigned to particular word groups through a group-aligned tier.) For example, back to the screenshot in Figure 1, each of the two words of the utterance is contained within its own group, an organization which restricts the focus of certain functions to

3. User-defined tiers can also be set as additional daughters.

individual words, for example syllabification and phone alignment, described in the next two subsections.

3.1.6 *Syllabification*

Upon entry of transcriptions in the IPA Target or IPA Actual tiers, transcribed phones are assigned descriptive syllable labels (visually represented with colours) such as 'Onset', 'Nucleus' or 'Coda' (e.g. Selkirk 1982). Different syllabification algorithms are provided to accommodate different languages. These algorithms can also be parameterized to suit various theoretical models. For example, the label 'OEHS' (for 'Onset of empty-headed syllable') can optionally be used by researchers working within approaches to syllabification which formally recognize this constituent (e.g. Kaye, Lowenstamm & Vergnaud 1990). Additional algorithms (for other languages or based on different assumptions about syllabification) can easily be added to the program, upon request.

The availability of different syllabification options offers the researcher means to test various hypotheses for any given dataset. Just like features are used to describe articulatory properties of phones, syllabification labels are also merely descriptive. No analysis is imposed by the program; data interpretation remains the sole responsibility of the researcher. Finally, as automatic labelling may at times produce spurious results (for example in the case of phone sequences which significantly depart from the types of strings expected in a given language), the researcher can quickly modify the outcomes of automatic syllabification through the Syllabification interface, whenever needed.

Syllabification labels are most useful in data queries, for example to return phones located in specific positions within the syllable. This level of functionality, combined with annotations pertaining to word stress, is central to the study of positional effects in phonology (e.g. Inkelas & Rose 2008 and references therein).

3.1.7 *Phone alignment*

After syllabification, a second algorithm performs automatic, phone-by-phone alignment of IPA Target with IPA Actual forms. For example, back to the screenshot in Figure 1, we can see that both vowel epenthesis (*spelen* [ˈspelə] produced as [z̪əˈpeːlə]) and consonant deletion (*gitaar* [ˌχiˈtaɹ] as [ˌhiˈtaː]) are represented as alignment of the relevant phones with an empty space in the corresponding form. Similarly, using this alignment, the user can detect the voicing discrepancy between aligned target [s] and actual [z] as well as the production of target [χ] as [h].

After transcription and related annotations are complete and verified, the user is in a position to perform database queries, as described in the next section.

3.1.8 *Database query*

The query system of Phon can be loosely described as a plug-in architecture. Built-in query scripts are provided for general purposes, as listed in (1).

(1) Query types built into Phon
   a. Aligned Groups (group-aligned forms)
   b. Aligned Phones (aligned phones and their features)
   c. CV Sequence (sequences of consonants, vowels and glides)
   d. Data Tiers (textual or IPA forms within specific tiers)
   e. Harmony (consonant or vowel harmony)
   f. Metathesis (consonant metathesis in aligned IPA forms)
   g. Word Shapes (stress patterns in words)

Each of these query scripts incorporates user-fillable forms to enter specific search criteria as well as options for selecting word groups, word or syllable positions, stress as well as participant name and age, where applicable. This system supports searches based on text strings (e.g. orthographic or IPA annotations), regular expressions and phonological expressions. The latter are formulated using Phonex, a special language which allows the searching of phones or metaphones (e.g. stress markers; word and syllable boundary markers) at all levels of data annotation described above. A practical example of the use of Phonex is provided in Section 4.3. Whether the user needs to query text strings, regular expressions or phonological expressions, queries can be executed on one or more sessions from the Project Manager, or on the currently-opened session transcript if the search is initiated from within the Session Editor.

Beyond these built-in scripts, the user can create custom scripts with no need to reprogram the application. Phon provides a Script Editor for creating custom queries. All queries are written in JavaScript. Users familiar with JavaScript can take advantage of having a full programming language for creating customized queries.[4] Also available from the Project Manager is the Query History, which provides access to all of the queries performed within a given project. Query results can be 'starred' for quick reference and re-opened at later dates.

3.1.9 *Report generation*

Query reports are exported as flat, comma-separated value (CSV) text files, which can then be opened in (UTF-8 compliant) third-party applications. Reports can be generated after each query, or through the Query History.

4. Additional scripts developed to support specific research needs can also be downloaded from a public script library <http://www.phon.ling.mun.ca/phontrac/wiki/search/scriptlibrary> and opened through the Script Editor.

3.2 New in Phon 1.5

Phon 1.5, released in the spring of 2011, offers significant improvements over previous versions. While the general look and feel of Phon 1.5 has its roots in version 1.4 of the application (e.g. user-configurable interface; waveform visualization; script-based data query and reporting functions), many details have been streamlined, both under the hood and within the graphical user interface.[5]

3.2.1 *Technical improvements*

Perhaps the most important change within the program's design is the replacement of its former Media Player internal engine (QuickTime for Java, a deprecated technology no longer supported by Apple) with VLC, an open-source media playback system built by a vibrant community of developers worldwide <http://www.videolan.org/vlc/>. In addition to providing a more stable application, the move to VLC also made possible the release of Phon for Unix/Linux platforms, a significant addition, especially given the open-source advocacy behind our project.

Phon 1.5 also comes with a significant change in our approach to file storage. While previous versions of Phon were combining project and transcript files into compressed archives identified with the '.phon' extension (effectively hiding these files from the user), Phon 1.5 keeps all corpus files explicitly listed and available through the user's computer file browsing system. In order to facilitate data backup and exchanges among researchers, we also added functions to archive (compress) and extract (expand) corpus files. I describe some of these functions as well as other interface improvements in the next subsection.

3.2.2 *Interface improvements*

Starting with interface adjustments, our newly introduced VLC-based Media Player comes with an interface which now integrates the functions needed for time segmentation (as described in Section 3.1.2). The Session Editor also offers a series of improvements, from colour marking of CHAT codes in the Orthography tier to better and more flexible visuals for the display of phone syllabification and alignment.

We also added new components to support new functions. Chief among these is the Workspace window, illustrated in Figure 3. As we can see from this screenshot, the Workspace window, which opens each time the application is launched, provides access to the list of corpora contained in the user's workspace folder as well as to a number of settings and actions. The user can set locations for Phon

5. A full list of the technical improvements brought to Phon 1.5 can be found at: <http://www.phon.ling.mun.ca/phontrac/wiki/phon1_5/ReleaseNotes>

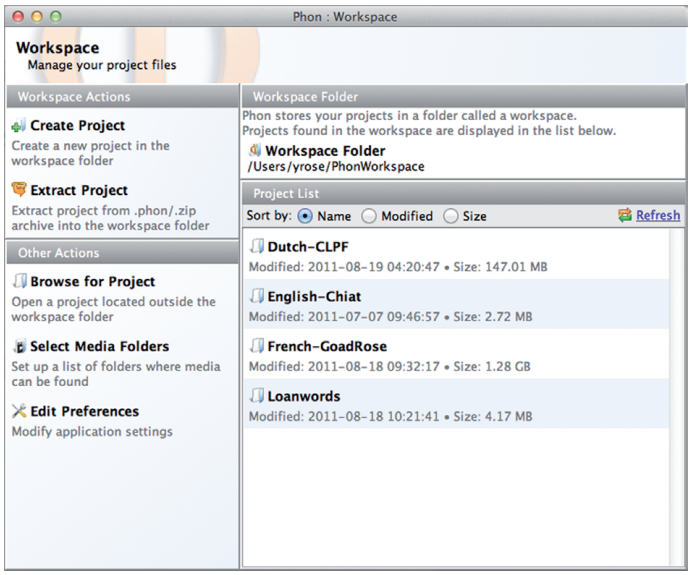


Figure 3. Workspace window

corpora and their respective media files (different default locations can be set for each), and perform general operations such as archiving corpora or extracting them from compressed archives. These functions facilitate corpus file management and backup as well as related tasks such as file sharing among members of a research team or with the research community.

3.3 Interaction between Phon and CLAN

Going back to the larger context of PhonBank, Phon also offers a high degree of connectivity with CLAN, its cousin application within CHILDES. Just as PhonBank supplements the CHILDES database, Phon supplements CLAN's general functionality, which has traditionally focused on non-phonological domains of analysis (e.g. morpho-syntax; discourse).

CLAN and Phon are now better integrated than ever. Corpora created with CLAN can be imported into Phon and, conversely, Phon corpora can be converted for use into CLAN. Conversions between the CLAN and Phon formats are performed via the more general TalkBank XML format. For example, a corpus created within CLAN must first be converted into TalkBank XML and from there into the Phon format. Following the same logic, a Phon corpus will first be converted into TalkBank XML before it is converted into CLAN. Conversions between CLAN and TalkBank XML are performed with the Chatter utility, while conversions between Phon and TalkBank XML are performed with the PhonTalk

utility. More information about these conversions and related utilities can be accessed at <http://www.talkbank.org/software/>.

Through these conversion methods, researchers can take advantage of specialized functions of each application without having to transcribe their data again. These conversions also allow the fast incorporation of already-existing CHILDES corpora into PhonBank: old data can now be used in new ways, often with only minimal coding efforts needed. These conversions also avoid unnecessary duplication of functions within CHILDES/PhonBank: the overlap in functionality between CLAN and Phon is indeed minimal and mostly pertains to corpus building (e.g. time segmentation and data transcription), while most analytical functions are unique to each program.

3.4 Other goodies

The core functions supported by Phon are supplemented by a number of other facilities. For example, Phon supports the merging of data multiple transcript sessions into single transcripts, which is useful for incremental corpus building (e.g. Voormann & Gut 2008) or for transcription activities involving multiple individuals. Phon also supports the importation of transcribed corpora (orthographic and/or phonetic) using a general CSV format, as well as the exportation of transcript data into this format. Data can thus easily enter or exit the realm of Phon. Finally, all functions are described in the user manual included with the application, now in PDF format and available directly from Phon's Help menu.

In the next section, I demonstrate some of the analytical flexibility that Phon offers, taking as an example corpus-based research in loanword phonology.

4. A practical illustration

As mentioned in the introduction, even though Phon was built to facilitate research in phonological development, it can easily be used in a number of other contexts, for example in fieldwork studies (e.g. Buchan 2011). Phon also has demonstrated potential in computational linguistics (Rose et al., to appear), as well as in loanword phonology, to which we now turn.

4.1 Loanword phonology

Loanwords, or linguistic borrowings, can be loosely defined as words that are introduced from a foreign (source) language into a borrowing language, typically

with a number of phonological adaptations required to make these words compatible with the phonotactics (possible sounds and sound combinations) of the borrowing language. Once integrated into a borrowing language, a loanword may be used by monolingual speakers of this language, independent of any knowledge of the foreign language the loanword originates from. However, the actual process of loanword adaptation is arguably performed by bilingual speakers of both the source and the borrowing languages (e.g. Paradis & LaCharité 1997, 2008). Bilinguals can also play a role in social conventions regulating the ways in which foreign sounds are adapted into the borrowing language (e.g. Grosjean 1982; Grosjean & Soares 1986). For example, English [θ, ð] are typically adapted as [t, d] in Québec French while they are adapted as [s, z] in European French, a distinction that transcends phonological considerations proper, as the alternatives found in each of the dialects could in principle occur in the other dialect (see Paradis & LaCharité 1997: 421–423 for further discussion).

The study of loanword phonology thus provides insights into the linguistic systems of bilingual speakers as well as into more general issues that may arise in multilingual (and multicultural) societal contexts. Without going into further considerations (e.g. contributions to Kenstowicz & Uffmann 2006 for recent discussions), I use loanword data in the next section to illustrate how loanwords and the phonological adaptations they involve can be studied within Phon.

4.2 The corpus

For the needs of this illustration, I converted an already-existing corpus of French loanwords in Kinyarwanda (Rose 1995) from its original format (Microsoft Word) into Phon. In order to do so, I first converted the original file into the CSV format (using OpenOffice Calc), and then imported the CSV file into a Phon session using the CSV import facility built into Phon. In the process, I created a number of user-defined tiers and converted the original IPA transcriptions into the more modern Unicode font format used by Phon. Upon data import, the IPA transcriptions documenting the source (French) and adapted forms (into Kinyarwanda) were assigned to the IPA Target and IPA Actual tiers, respectively. Other data were assigned to a combination of default and user-defined tiers. The following screenshot (Figure 4) illustrates a few of these tiers, alongside the syllabification and phone alignment data, for one data record.

We can also observe in this example segmental discrepancies between aligned phone pairs (e.g. [p] ↔ [m͡h]), as well as epenthesis of a vowel at the end of the adapted form. Each of these observations naturally lends itself to one or more

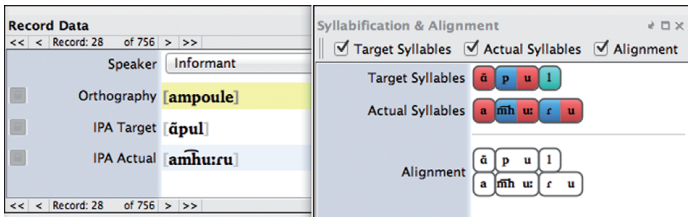


Figure 4. Loanword data: sample record

research questions, for example concerning the stability of an observed phenomenon or its implications for theoretical phonology.

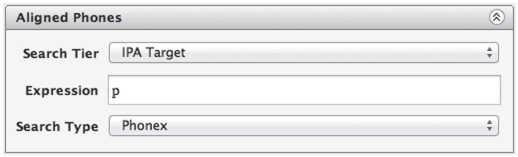
In the next subsection, I use this corpus to demonstrate some of the functions described in Section 3. While none of the compilations presented below should be considered as definitive (this is merely an illustrative sketch), the reader interested in taking a systematic look at these data is welcome to download the full corpus from the BilingBank database <http://www.talkbank.org/data/BilingBank/>, another offspring of the original CHILDES project. Additional compilations and related discussions can also be found in Rose (1995).

4.3 Data query and reporting: A brief demonstration

As observed above, the behaviour of [p] might raise certain questions, as this consonant is part of the Kinyarwanda consonantal inventory (Jouannet 1983). Through a simple query using the Aligned Phones search script, illustrated in (2a), we can observe that [p] actually yields a number of different adaptations, as per the summary breakdown in (2b).

(2) Adaptations of French [p] (n = 135)

a. Aligned Phone query



b. Results summary

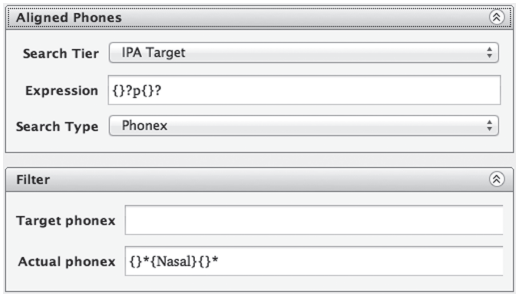
p ↔ p	114	p ↔ m͡h	8
p ↔ β	8	p ↔ m͡b	1
p ↔ k^w	1	p ↔ m͡p^h	1
p ↔ Ø	1	p ↔ m	1

While the majority of occurrences of target [p] are integrated in Kinyarwanda as expected, i.e. without any modification (n = 114), a few examples remain

unaccounted for at this stage. Focusing in particular on the data in the second column in (2b), we observed occurrences (n = 11) which retain the labiality of the original [p] in some form but also acquire some nasal characteristic. First off, the laryngealization pattern observed in most of these examples (e.g. [p] ↔ [m͡h]) relates to the native phonology of Kinyarwanda, which prevents the realization of pre-nasalized voiceless stops. As reported by Kimenyi (1979) and Jouannet (1983), these consonants undergo 'debuccalization' (loss of supralaryngeal articulator), with their place of articulation expressed on the pre-nasal portion of the phone. While this phenomenon relates to the native phonology of Kinyarwanda, one question remains in the context of the loanwords, about what strings of phones in the original French words might trigger the observed nasalization. In order to investigate this issue further, one possibility is to explore what segments appear in the phonological environment of [p] in the source and corresponding adapted forms. A query similar to that in (2a) can be performed to address this question, however this time with a filter specified to restrict the output to nasal outcomes and their environments, as per the screenshot in (3a).[6] The results of this query reveal that all of the 11 nasalized outcomes in (2b) originate from the presence of nasal vowels immediately preceding [p] in the French forms, as we can see with the exhaustive listing in (3b).

(3) French [p] leading to nasalized outcomes (n = 11)

a. Aligned Phone query and result filter



b. Results (exhaustive listing)

ɑ̃pa ↔ aːm͡ha:	ɔ̃pl ↔ oːm͡hurˌ	ɔ̃pɛ ↔ uːm͡beː
ɑ̃pj ↔ aːmp͡ʰij	ɔ̃ps ↔ omus	ɔ̃p ↔ oːm͡h
ɑ̃pu ↔ am͡huː	ɔ̃pu ↔ oːm͡huː	ɛ̃pʀ ↔ im͡hirˌ
ɔ̃pa ↔ oːm͡ha	ɔ̃pɔ ↔ iːm͡ho	

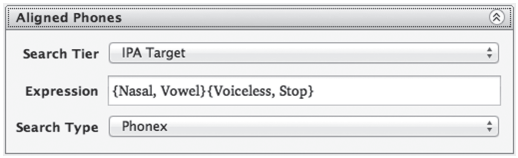
Finally, given that, according to Kimenyi (1979) and Jouannet (1983), debuccalization should affect all members of the voiceless stop family, one more query can be performed, still using Phonex, to verify the behaviour of this natural class

6. See the User Manual for further information about filtering query results.

of phones. This time, since we have already confirmed that the preceding nasal vowel is what causes pre-nasalization, a simple Phonex query (without filters) in (4a), which specifies a sequence of any nasal vowel followed by any voiceless stop, returns the expected matches. Representative results are listed in (4b).

(4) French voiceless stops leading to nasalized outcomes (n = 44)

a. Aligned Phone query and result filter



b. Results (representative, non-exhaustive listing)

i. ɑ̃p ↔ am͡h
ɑ̃t ↔ an͡h
ɑ̃k ↔ aŋ͡h

ii. ɑ̃p ↔ a:m͡h
ɑ̃t ↔ a:n͡h
ɑ̃k ↔ a:ŋ͡h

Other questions still remain at this stage, for example about the continuant outcomes of [p] in (2b) or the fact that certain vowels before the pre-nasalized laryngeals can be either long or short, in (4b), a difference which can be explained by the position of the vowel within the word, as word-initial vowels cannot be long in Kinyarwanda (e.g. Kimenyi 1979). Each of these questions lends itself to additional queries within the database. For example, in the case of the length distinction observed in (4b), which makes reference to initial word boundaries, one can supplement the expression in (4a) with an initial word boundary marker '#' (`#{Nasal, Vowel}{Voiceless, Stop}`).

As we can see from this brief demonstration, Phon's data annotation and query functions offer easy but powerful ways to perform systematic investigations of phonetically-transcribed datasets. Only a small subset of query types was covered within this section, however. Many more options are available and, as mentioned above, other query scripts can be programmed to accommodate additional needs.

5. Outlook

The software program and other facilities described in this chapter are the result of approximately seven years of development efforts. Looking ahead, we are now planning a series of extensions to these tools, the most important of these aims at providing support for acoustic measurement data. In order to achieve this goal, we will create an interface between Phon and Praat <http://www.fon.hum.uva.nl/praat/>. Using this new interface, researchers already working with Praat will

be able to take advantage of some of Phon's unique functions, especially for phonetic transcription and phonological data query, and researchers using Phon will be able to integrate acoustic measurements for both corpus preparation and data analysis. This new system will offer significant new support to researchers, who will be in a position to systematically verify the consistency of phonetic transcriptions (through acoustic data measurement) or use phonetic transcriptions as labels to organize and query their acoustic data.

Alongside this system, we will also expand Phon's data annotation and compilation functions in order to fully integrate acoustic data into Phonbank. Similar to all of the facilities described above, these expected outcomes should offer new and compelling solutions for corpus-based research in several areas of phonetic and phonological investigation.

References

Buchan, H. 2011. Phon: Free software for phonological transcription and analysis. *Language Documentation and Conservation* 5: 81–87.

Durand, J., Gut, U. & Kristoffersen, G. (eds). To appear. *The handbook of corpus phonology*. Oxford: OUP.

Ewen, C. J., Hume, E., van Oostendorp, M. & Rice, K. (eds). 2011. *The Blackwell companion to phonology*. Malden MA: Wiley-Blackwell.

Grosjean, F. 1982. *Life with two languages: An introduction to bilingualism*. Cambridge MA: Harvard University Press.

Grosjean, F. & Soares, C. 1986. Processing mixed languages: Some preliminary findings. In *Language processing in bilinguals: Psycholinguistic and neuropsychological perspectives*, J. Vaid (ed.), 145–179. Hillsdale NJ: Lawrence Erlbaum Associates.

Inkelas, S. & Rose, Y. 2008. Positional neutralization: A case study from child language. *Language* 83: 707–736.

Jouannet, F. 1983. Phonétique et phonologie: Le système consonantique du kinyarwanda. In *Le kinyarwanda: langue bantu du Rwanda*, F. Jouannet (ed.), 55–73. Paris: SÉLAF.

Kaye, J., Lowenstamm, J. & Vergnaud, J.-R. 1990. Constituent structure and government phonology. *Phonology* 7: 193–231.

Kenstowicz, M. & Uffmann, C. (eds). 2006. *Loanword phonology: current issues.* Special Issue of *Lingua* 116: 921–1194.

Kimenyi, A. 1979. *Studies in Kinyarwanda and Bantu Phonology* [Current Inquiry into Language and Linguistics]. Edmonton: Linguistic Research.

Paradis, C. & LaCharité, D. 1997. Preservation and minimality in loanword adaptation. *Journal of Linguistics* 33: 379–430.

Paradis, C. & LaCharité, D. 2008. Apparent phonetic approximation: English loanwords in Old Quebec French. *Journal of Linguistics* 44: 87–128.

Rose, Y. 1995. Minimalité, préservation et tolérance dans les emprunts français en kinyarwanda. MA thesis, Université Laval.

Rose, Y. 2010. The PhonBank initiative and second language phonological development: Innovative tools for research and data sharing. In *English pronunciation: Issues and practices (EPIP) – Proceedings of the first international conference*, A. Henderson (ed.), 223–241. Chambéry: Université de Savoie.

Rose, Y., Hedlund, G. J., Byrne, R., Wareham, T. & MacWhinney, B. To appear. Phon 1.4: A computational basis for phonological database building and model testing. In *Cognitive aspects of computational language acquisition*, A. Alishahi, A. Villavicencio, T. Poibeau & A. Korhonen (eds). Berlin: Springer.

Rose, Y. & MacWhinney, B. To appear. The PhonBank initiative. In *Handbook of corpus phonology*, J. Durand, U. Gut & G. Kristoffersen (eds). Oxford: OUP.

Selkirk, E. O. 1982. The syllable. In *The structure of phonological representation*, H. van der Hulst & N. Smith (eds), 337–385. Dordrecht: Foris

Voormann, H. & Gut, U. 2008. Agile corpus creation. *Corpus Linguistics and Linguistic Theory* 4: 235–251.