Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Parsing IPA Transcriptions with CLTS

Johann-Mattis List
Chair of Multilingual Computational Linguistics / Department of Linguistic and Cultural Evolution
University of Passau / Max Planck Institute for Evolutionary Anthropology

The Cross-Linguistic Transcription Systems (CLTS, https://clts.clld.org) project serves as a reference catalogue for speech sounds. At the core of the project is a generative method that parses existing IPA transcriptions (or transcriptions in other supported transcription systems) and checks if they conform to the principles and components laid out in the reference catalogue. As a result, Cross-Linguistic Transcription Systems is much more than a simple list of possible speech sounds transcribed in the International Phonetic Alphabet, but a system that allows to generate possible speech sounds and to check if sounds provided in various transcription systems contain problems. This study gives a short overview on the basic ideas that lead to the creation of the database and the parsing method and provides some examples showing how it can be employed in practice.

1 Introduction

With the publication of the Lexibank repository (List et al. 2022), a very large collection of wordlists from more than 2000 different language varieties was released that has recently also been published as a CLLD application (List et al. 2023, https://lexibank.clld.org). What makes this collection special is that for more than 2000 varieties, lexical entries are provided in a unified transcription system that is compatible with the International Phonetic Alphabet. This transcription system, that we call B(road)IPA was first presented in 2018 (Anderson et al. 2018) as part of the Cross-Linguistic Transcription Systems project (CLTS, https://clts.clld.org) and has since then been regularly updated and enhanced.

Along with the first publication of CLTS, there has been a lot of confusion among colleagues about the role that the project plays when it comes to phonetic transcriptions. Since On the one hand, it does not seem to be clear what we understand under a reference catalog (not to speak about a reference catalog for the transcription of speech sounds). On the other hand, the generative nature of the method by which CLTS parses phonetic

Computer-Assisted Language Comparison in Practice
Volume 6, Number 2, 87–92
URL: https://calc.hypotheses.org/6546/ DOI: 10.15475/calcip.2023.2.5
Published under a Creative Commons Attributions 4.0 LICENSE
Published on 20/11/2023

transcriptions in various transcription systems is often ignored. As a result, CLTS is considered as just another list of phonetic transcriptions with some metadata attached to it.

2 Reference Catalogs

The Lexibank repository provides a good starting point to illustrate the role that reference catalogs can play in comparative linguistics. Lexibank builds on three major catalogs, Glottolog (https://glottolog.org, Hammarstrom et al. 2023), Concepticon (https://concepticon.clld.org, List et al. 2023), and CLTS, in order to unify and standardize more than 100 different individual lexical datasets.

Thanks to the consequent linking of most of the Lexibank varieties to Glottolog, major types of metadata that Glottolog provides — such as geolocations, classifications, but also information on the documentation status of a language — can be integrated with the Lexibank wordlist collection. At the same time, Glottolog facilitates to search for duplicates or nearly identical varieties of the same language.

Thanks to the consequent mapping of concept elicitation glosses to Concepticon, the wordlists underlying individual datasets that made it into Lexibank can be easily compared and combined wordlists reflecting — for example — the majority of the concepts underlying the lists of Swadesh (1952 and 1955) can be easily extracted from the repository as a whole. In the past, many projects have tried to come up with their own "internal" version of the Concepticon, but it turned out that only the explicit investment into the creation of a full-fledged reference catalog that has been constantly maintained and updated from its first publication (List et al. 2016) until today made it possible to link datasets in this large scale.

CLTS helps us to unify speech sounds in Lexibank. The basic idea of CLTS was to provide unified transcriptions reflecting a strict subset of the IPA (called B(road)IPA, with "broad" referring to the fact that many nuances are covered). With this set of strict transcriptions tha twould not allow for variants as they are common in the practial application of the International Phonetic Alphabet (compare [ts] vs. [t͡s] vs. [t̪s̪]), we hoped to be able to handle all speech sounds that we would encounter in the individual datasets we tried to standardize for Lexibank. The more data we encountered, however, the clearer it became that the task of unifying the multiple ways in which IPA transcriptions are realized by scholars would not be feasible on the long run. While we could and can afford to maintain the Concepticon by expanding it whenever new concepts are encountered that are not yet handled by the resource, we realized that doing the same with phonetic transcriptions would be too tedious in practice. On the one hand, too many ad-hoc decisions would have to be carried out whenever adding individual sounds not yet reflected in the CLTS catalogue. On the other hand, it was clear that the

systematic structure of the International Phonetic Alphabet (as well as other transcription systems) should — in theory at least — make it possible to create a system that could generate sounds it encountered rather than eliciting them all explicitly before.

After several months of trying, it turned out that the creation of such an automatic system was indeed possible, and the method that we proposed first in 2018 (Anderson et al. 2018) has not been substantially modified since then. The CLTS algorithm for parsing speech sounds transcribed in IPA or other transcription systems proceeds in three basic steps of normalization, lookup, and generation.

3 How CLTS Represents Transcription Systems

The CLTS data for an individual transcription system consists of six files with tabular data. There are three tables which provide what we consider base sound symbols along with their feature descriptions (when occuring alone, without further modifiers), one for consonants (consonants.tsv), one for vowels (vowels.tsv), and one for tones. The base files provide a fixed set of features in column form (phonation, place, and manner for consonants and roundedness, height, and centrality for vowels) along with a column for additional feature values which are provided in the form of a comma-separated list of key-value pairs separated by a colon (column extra). For tones, we simply provide contour, start, middle, and end, reflecting the typical tone movements encountered mainly in South-East Asian languages (tones.tsv). An additional column allows to mark a certain sound as an alias (column alias). This means that the sound is frequently transcribed in the form given in the literature, but that it should rather be represented by another sound that we offer in the database and that resolves to the same features (consider, for example the voiced labiodental stop consonant [b̪] which is at times transcribed as [ɗb]). With these base sounds (which can well consist of more than one character and therefore also explicitly defined), we create a first basis of every transcription system from which all additional characters (also diphthongs and consonant clusters) can be derived.

A fourth file provides diacritics or combinations of diacritics that can precede or follow the base sounds (diacritics.tsv). Diacritics modify base sounds not only in the representation of the grapheme but also by modifying the internal feature representation of base sounds. A fifth file provides a list of markers, that is, symbols that do not refer to sounds, such as markers for morpheme boundaries (we use "+" for this purpose) or markers for word boundaries (CLTS accepts "_", but we discourage its use). The sixth file provides a list of normalization pairs (source, target, one character only as source) that are applied as the first step of the parsing algorithm (normalize.tsv). With these files in place (which can be provided for any alphabetical transcription system that follows

the basic logic behind the IPA with its combination of base characters and diacritic marks), various transcription systems (among them the B(road)IPA) can be processed.

When parsing a string representing one sound in a phonetic transcription, the parsing algorithm first normalizes the string, then tries to match it directly against a base vowel, consonant, or tone, and finally tries to generate the sound dynamically. In the first step, input data are normalized by applying Unicode's NFD normalization (Moran and Cysouw 2018: 16) and by applying the custom replacement table in which individual characters are looked up and replaced by their normalized counterparts (file normalize.tsv). This helps us to deal with frequently encountered Unicode lookalikes that scholars use in their work without paying much attention to them. Thus, many scholars use the colon ":" to mark vowel length, while the correct character is "ː".

Having checked in the second step, if a direct match with the base sounds for vowels, consonants, and tones can be made, the parsing algorithm first applies a regular expression to identify the base sound in the string. If this sound can be identified, the method then proceeds in steps to the left and the right of the base sounds and tries to match the diacritics provided in the table with diacritic symbols. If they can be identified, the features underlying the base sound are consecutively modified. If all symbols can be explained in this way, the orthographic representation of the sound is generated from the underlying collection of features (following strict rules for the order by which features are displayed on transcriptions) along with a human-readable name of the sound that follows the typical characterizations of sounds provided by the IPA.

As a result, each sound that is successfully parsed by CLTS has an underlying representation as a feature bundle, a unique name composed of these features in a fixed order, and a transcription composed of the individual features in a fixed order. Since CLTS not only parses a sound, but recreates the sound after parsing, the approach is very useful to check for common errors in phonetic transcriptions, due to the multiple tests that a transcription must pass before it is accepted. At the same time — since CLTS is able to generate transcriptions from feature bundles — CLTS can likewise create new sounds in their transcription when passing the human-readable name of the sound.

4 Example

In order to test this in concrete, you can just install the pyclts package (List et al. 2020) and download the most recent version of the CLTS data by cloning with GIT.

```
$ pip install pyclts
$ git clone https://github.com/cldf-clts/clts.git
```

Having done so, you can interactively use CLTS in a Python session by loading the CLTS class from the pyclts package and instantiating the class by passing the path to the CLTS dataset. Having done so, you can test the B(road)IPA in various ways by passing it names of sounds as well as transcriptions of sounds.

```
>>> from pyclts import CLTS
>>> clts = CLTS("clts")
>>> sound1 = bipa['pʲʰ']
>>> sound2 = bipa['pʰʲ']
>>> sound3 = bipa["voiceless palatalized aspirated bilabial stop consonant"]
>>> sound4 = bipa["palatalized aspirated voiceless bilabial stop consonant"]
>>> sound1 == sound2 == sound3 == sound4
True
```

You can see from this example, that CLTS handles both the inconsistent order of diacritics on sound1 and sound2 in our example, as well as the different order in the description of the name. Since transcriptions can be generated from names, CLTS can even translate sounds from one transcription system to another one, by passing the name obtained for one sound in one transcription to another transcription systems (this requires, of course, that the target system can express the relevant sounds).

```
>>> napa = clts.transcriptionsystem("napa")
>>> napa["voiceless palatalized aspirated bilabial stop consonant"]
'pʸʰ'
```

In its current form, CLTS is incredibly useful for the creation and maintanance and further expansion of our Lexibank wordlist collection. While before we were struggling hard to keep up with the multitude of traditions and attitudes towards phonetic transcriptions across various data sets, we now still have to standardize and normalize transcription systems through orthography profiles (Moran and Cysouw 2018), but the major work of deciding how to standardize individual sounds has significantly dropped from the moment on when we integrated CLTS into the Lexibank workflow for wordlist standardization.

5 Outlook

CLTS is still actively maintained and curated, but unlike with the Concepticon, where we add new concept lists on a regular basis, major modifications are only done when encountering new sounds that we could not handle so far. These modifications may also force us to rewrite parts of the code base, so we cannot do them on a regular basis. The

last major modification on CLTS was made in 2021, when we prepared for a new study in which we wanted to be able to represent existing collections of phoneme inventories (Anderson et al. forthcoming). However, even if I do not expect much modifications or enhancements with respect to the basic algorithm, I have the hope that we can soon add a method that would allow us to generate various kinds of numeric features from the feature system underlying CLTS. Having generated vector representations of sounds would offer various possibilities to work with the constantly growing data in Lexibank.

References

Anderson, Cormac and Tresoldi, Tiago and Chacon, Thiago Costa and Fehn, Anne-Maria and Walworth, Mary and Forkel, Robert and List, Johann-Mattis (2018): A Cross-Linguistic Database of Phonetic Transcription Systems. Yearbook of the Poznań Linguistic Meeting 4.1. 21-53.

Anderson, Cormac and Tresolid, Tiago and Greenhill, Simon J. and Forkel, Robert and Gray, Russell and List, Johann-Mattis (forthcoming): Measuring variation in phoneme inventories. Journal of Language Evolution 0.0. 1-16. DRAFT: https://doi.org/10.21203/rs.3.rs-891645/v1

List, Johann-Mattis and Tjuka, Annika and van Zantwijk, Mathilda and Blum, Frederic and Barrientos Ugarte, Carlos and Rzymski, Christoph and Greenhill, Simon J. and Robert Forkel (2023): CLLD Concepticon [Dataset, Version 3.1.0]. Leipzig: Max Planck Institute for Evolutionary Anthropology. https://concepticon.clld.org

Hammarstrom, Harald and Haspelmath, Martin and Forkel, Robert and Bank, Sebastian (2023): Glottolog. Version 4.8. Leipzig: Max Planck Institute for Evolutionary Anthropology. https://glottolog.org

List, Johann-Mattis and Cysouw, Michael and Forkel, Robert (2016): Concepticon. A resource for the linking of concept lists. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation. 2393-2400.

List, Johann-Mattis, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, Johannes Englisch, & Russell D. Gray (2022): Lexibank, A public repository of standardized wordlists with computed phonological and lexical features. Scientific Data 9.316. 1-31. https://doi.org/10.1038/s41597-022-01432-0

List, Johann-Mattis, Robert Forkel, Simon J. Greenhill, Christoph Rzymski, Johannes Englisch, & Russell D. Gray (2023): Lexibank Analysed [Data set, Version 1.0]. https://zenodo.org/records/7836668

Moran, Steven and Cysouw, Michael (2018): The Unicode Cookbook for Linguists: Managing writing systems using orthography profiles. Berlin: Language Science Press.

Johann-Mattis List and Cormac Anderson and Tiago Tresoldi and Robert Forkel (2020): PYCLTS. A Python library for the handling of phonetic transcription systems [Software Library, Version 3.0.0]. Geneva: Zenodo. https://pypi.org/project/pyclts

Swadesh, Morris (1952): Lexico-statistic dating of prehistoric ethnic contacts. With special reference to North American Indians and Eskimos. Proceedings of the American Philosophical Society 96.4. 452-463.

Swadesh, Morris (1955): Towards greater accuracy in lexicostatistic dating. International Journal of American Linguistics 21.2. 121-137.