Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Towards a Unified Conversion Table for Semitic Transcriptions and Transliterations

Carlo Meloni / Johann-Mattis List
Chair for Multilingual Computational Linguistics
University of Passau

In this study we present a preliminary conversion table that can be used for transcriptions and transliterations across different Semitic languages. We introduce the basic idea behind the table, show how it can be used, and explain how we hope to expand it in the future.

1 Introduction

Transcription and transliteration practices vary drastically across languages, language groups, and language families (Anderson et al. 2018). As a result, it is often difficult for scholars who are new to a given linguistic area or subgroup to understand how symbols are used to represent sounds. In the following, we will give a very short overview on the transcription practices that emerged for the Semitic language family and propose a conversion table that can be used along with a dedicated software package to convert from individual transcription systems employed to document most Semitic languages to phonetic transcriptions in the International Phonetic Alphabet.

2 Background

Semitic transcription and transliteration practices developed in nineteenth-century European scholarship, particularly within German-speaking Semitic philology. What is known as the Semitological transcription system did not emerge as a single fixed standard, but as a shared framework for representing Semitic languages such as Hebrew, Aramaic, Arabic, Akkadian, and Ethiopic in the Latin alphabet. Its primary purpose was to support comparative and historical analysis rather than to provide fine-grained phonetic detail (Weninger et al. 2011).

Computer-Assisted Language Comparison in Practice
Volume 8, Number 2
URL: https://calc.hypotheses.org/9109 DOI: 10.15475/calcip.2025.2.6

Published on 17/12/2025

Early Semiticists such as Wilhelm Gesenius, Heinrich Ewald, and later Theodor Nöldeke developed a set of conventions that adapted the Latin alphabet through the systematic use of diacritics and modified letters. Dots below, macrons, carons, and other special characters made it possible to represent consonantal distinctions lacking direct equivalents in European languages, most notably emphatic consonants (pharyngealized or ejective) and pharyngeals / laryngeals segments. From the outset, these conventions were intended to encode phonological categories and historical identities rather than narrow phonetic detail. A symbol such as <ṭ>, for example, was meant to designate an emphatic /t/ as a structural category, without committing the scholar to a specific articulatory analysis such as pharyngealization or ejection (/tˁ/, /tʼ/). By the late nineteenth and early twentieth centuries, the system had become widely established in grammars, dictionaries, and epigraphic editions, with a largely shared core despite minor national variations.

Within this general Semitological framework, language-specific transliteration systems were developed for individual languages and subgroups (cf., for example, Brockelmann and Ronkel 1935). Arabic studies produced influential conventions such as the DMG system and DIN 31635, while Hebrew, Ethiopic, Ethio-Semitic, and Modern South Arabian traditions adapted the same principles to their own linguistic features (vowel length and quality contrasts for Hebrew, labialized and palatalized consonants for Ethio-Semitic). In all cases, the emphasis remained on structural and historical comparability rather than phonetic precision.

The strength of the Semitological system lies in this abstraction, which has ensured its long-term stability and cross-linguistic applicability and has allowed it to adapt smoothly to digital typography and Unicode. At the same time, the system has clear limitations. Its deliberate avoidance of phonetic specificity can obscure real differences in pronunciation between languages and dialects, and its categories reflect the priorities and assumptions of nineteenth-century European scholarship, which are not always aligned with modern descriptive or community-based approaches to language documentation. For phonetic analysis and fieldwork, IPA-based transcription is therefore indispensable, and the two systems are best seen as complementary rather than competing (Huehnergard and Pat-El 2019).

3 Materials and Methods

The general idea that we have in mind is to come up with an initial orthography profile that could serve as a general basis to turn transcriptions and transliterations used in particular contexts to transcribe lexical data in Semitic languages into standardized transcriptions of the International Phonetic Alphabet, or – more specifically – the particular version of the IPA underlying the Cross-Linguistic Transcription Systems initiative Anderson et al. (2018).

To achieve this conversion and to run the tests, we make use of the possibility to convert original strings written in individual transcription traditions typical for the handling of individual Semitic languages, with the help of conversion tables as introduced in the LinSe software package (Forkel and List 2024, https://pypi.org/project/linse). Conversion tables in *LinSe* build on the idea of Orthography Profiles presented originally by Moran and Cysouw (2018). The differences between orthography profiles are conversion tables are mostly conceptually. Although orthography profiles predate conversion tables, conversion tables can be thought of as the more abstract concept, in so far as they serve for the conversion of strings drawn from one alphabet into strings represented by different alphabets with the help of rudimentary replacement rules that are applied in a greedy fashion.

Conversion tables in *LinSe* have a very flexible format. All that one needs to create a conversion table are data in tabular form represented in CSV format. In the conversion table, one column reflects the alphabet in which the original strings are represented, and additional columns can be used to provide replacement values. When employing a conversion table, one can either simply parse the original string or the original set of strings into chunks defined as graphemes in the column representing the original alphabet, or one can convert the values directly to the desired replacement values.

Our initial conversion table for Semitic transliteration and transcription consists of roughly 150 graphemes, i.e., strings consisting of one or more characters, along with their most general counterpart in the B(road)IPA system of the CLTS standard for phonetic transcription based on the International Phonetic Alphabet (IPA 1999) along with the name of the respective sound in the CLTS system (see https://clts.clld.org for details). In three cases, one sequence corresponds to two sounds. In these cases, the corresponding IPA sounds are separated by a white space (following the basic conventions used in software tools, such as LingPy, see List et al. 2018) and the names of the sounds are separated by a + symbol.

The conversion table itself can be found on Codeberg (https://codeberg.org/digling/semitic-transliterations), from where it can be freely downloaded and used along with the LinSe package or with other software solutions.

4 Examples

4.1 Introducing the SegmentGrouper Object in LinSe

A conversion table in LinSe can be initiated in two major ways. One can load it from file, or one can pass it as a two-dimensional list. The following conversion table identifies `a`, `b`, `ab`, and `abab` as valid segments of a sequence and will group them together in a greedy fashion, if it identifies them in a string. By calling the function with a string as an argument, the instantiated SegmentGrouper will split the input string into chunks recognized from the alphabet.

```
>>> from linse.convert import SegmentGrouper
>>> sg = SegmentGrouper.from_table([["Sequence"], ["a"], ["b"],
["ab"], ["abab"]])
>>> sg("aba")
["ab", "a"]
```

When instantiating the SegmentGrouper with an additional column, this column can serve as the replacement table. To convert a sequence into another sequence, one must pass the name of the column as argument when calling the function.

```
>>> sg = SegmentGrouper.from_table([["Sequence", "Out"], ["a",
"A"], ["b", "B"], ["ab", "C"], ["abab", "D"]])
>>> sg("aab", column="Out")
["A", "C"]
```

4.2 Employing the SegmentGrouper on Semitic Data

The initial conversion table is supplemented with this study in the form of a CSV file that can be downloaded from Codeberg (https://codeberg.org/digling/semitic-transliterations). From there, one can either directly download the file semct.csv or clone the repository. In the following, we show how it can be used to retrieve IPA transcriptions from the transliteration of Arabic numerals from one to five. We assume that the terminal is opened in the same folder in which the file resides.

```
from linse.convert import SegmentGrouper
from tabulate import tabulate

sg = SegmentGrouper.from_file('semct.csv', delimiter=",")

# words are taken from
# https://en.wiktionary.org/wiki/Appendix:Arabic_Swadesh_list
words = ["wāḥid", "ʾiṯnān", "ṯalāṯa", "ʾarbaʿa", "ḵamsa"]

for word in words:
    table += [[" ".join(sg(w)), " ".join(sg(w, column="IPA"))]]

print(tabulate(table, tablefmt="pipe", headers=["Original",
"IPA"]))
```

This code produces the results shown in Table 1. While it is clear that there are quite a few different ways how these could have been achieved, we think that conversion tables offer a particular simple way to get started with sequence manipulation, especially also because they can be easily tested and expanded.

Original	IPA
w ā ḥ i d	w aː ħ i d
ʾ i ṯ n ā n	ʔ i θ n aː n
ṯ a l ā ṯ a	θ a l aː θ a
ʾ a r b a ʿ a	ʔ a r b a ʕ a
ḵ a m s a	x a m s a

Table 1: Result of the sequence conversion routine of Arabic numerals from one to five.

5 Conclusion

Due to the tabular format of conversion tables, individual replacement columns for individual languages can be easily added and defined, helping us to account for potential ambiguities resulting from the ambiguous coding of elements. In the future, we hope to expand the current system by providing more language-specific conversion information and expanding the initial collection of graphemes. We do not think that this initial conversion table is correct in all cases, nor do expect it to serve as a competitor for targeted conversion tools for individual languages, such as PanPhon (Mortensen et al. 2016). However, we take the table as a hopefully useful starting point from which we intent to see if we can start to populate a larger collection of etymologies in Semitic languages that we want to investigate in more detail along with their phonetic representations.

References

Anderson, Cormac, Tiago Tresoldi, Thiago Costa Chacon, Anne-Maria Fehn, Mary Walworth, Robert Forkel, and Johann-Mattis List. 2018. "A Cross-Linguistic Database of Phonetic Transcription Systems." Yearbook of the Poznań Linguistic Meeting 4 (1): 21–53. https://doi.org/10.2478/yplm-2018-0002.

Brockelmann, Carl, and Philippus Samuel van Ronkel. 1935. Die Transliteration Der Arabischen Schrift in Ihrer Anwendung Auf Die Hauptliteratursprachen Der Islamischen Welt: Denkschrift Dem 19. Internationalen Orientalistenkongreß in Rom. Leipzig: Deutsche Morgenländische Gesellschaft.

Forkel, Robert, and Johann-Mattis List. 2024. "A New Python Library for the Manipulation and Annotation of Linguistic Sequences." Computer-Assisted Language Comparison in Practice 7 (1): 17–23. https://doi.org/10.15475/calcip.2024.1.3.

Huehnergard, John, and Naʻama Pat-El, eds. 2019. The Semitic Languages. 2nd ed. Abingdon and New York: Routledge.

IPA, ed. 1999. Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet. Cambridge: Cambridge University Press.

List, Johann-Mattis, Cormac Anderson, Tiago Tresoldi, and Robert Forkel. 2021. Cross-Linguistic Transcription Systems. Version 2.1.0. Jena: Max Planck Institute for the Science of Human History. https://doi.org/10.5281/zenodo.3515744.

List, Johann-Mattis, Mary Walworth, Simon J. Greenhill, Tiago Tresoldi, and Robert Forkel. 2018. “Sequence Comparison in Computational Historical Linguistics.” Journal of Language Evolution 3 (2): 130–44. https://doi.org/10.1093/jole/lzy006.

Moran, Steven, and Michael Cysouw. 2018. The Unicode Cookbook for Linguists: Managing Writing Systems Using Orthography Profiles. Berlin: Language Science Press. https://langsci-press.org/catalog/book/176.

Mortensen, David R., Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori S. Levin. 2016. “PanPhon: A Resource for Mapping IPA Segments to Articulatory Feature Vectors.” In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 3475–84. ACL.

Weninger, Stefan, ed. 2011. The Semitic Languages: An International Handbook. With Geoffrey Khan, Michael P. Streck, and Janet C. E. Watson. Berlin/Boston: Walter de Gruyter.

Supplementary Material

Data and code can be found at https://codeberg.org/digling/semitic-transliterations.

Funding Information

This project has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (Grant agreement No. 101044282). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.