Computer-Assisted Language Comparison in Practice

Tutorials on Computational Approaches to the History and Diversity of Languages

Typing Special Characters as a Key Skill for Linguists

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

Most linguists have to type special characters that are not available on an ordinary keyboard on a regular basis. Reflecting about the general problems involved in typing special characters, I review different solutions and argue that linguists should not only be able to type special characters on their computers, but that they should also have some basic knowledge about their technical aspects and know how to expand and customize them. In order to improve the training of young scholars, it is important to discuss special character typing more openly in linguistics, especially in the classroom and with doctoral students, sharing individual solutions openly.

1 Typing Special Characters

Most linguists encounter special characters already in their early graduate studies. These characters that are not available on the regular keyboard of a computer are typically used to transcribe particular sounds from the world's languages with specific symbols, usually following the system proposed by the International Phonetic Alphabet (IPA 1999). However, since linguists like variation, since not all linguists follow the tradition of the IPA, and since the IPA has been extended and modified various times, the number of characters used in the literature largely exceeds those that one can find in a typical IPA Chart. As a result, linguists who want to cite the work of their colleagues, as well as linguists who want to assemble data on individual languages will sooner or later have to type these special characters in their own work.

When I wrote my first term papers as a student of comparative linguistics at Freie University Berlin in 2003, I had a computer with a Windows operation system and was naturally writing with the help of the Word program. Special characters already occurred in my very first term paper, where I discussed the origin of the feminine gender in Indo-European (List 2003) and had to type quite a few superscript and subscript letters used to represent the Proto-Indo-European sound system, such as the famous three laryngeals [h_1, h_2, h_3] — sounds whose phonetic value is unclear (Meier-Bruegger 2002). While it was annoying to type them, it was not difficult to do so, given my knowledge about

Computer-Assisted Language Comparison in Practice
Volume 7, Number 2
URL: https://calc.hypotheses.org/7806 DOI: 10.15475/calcip.2024.2.5
Published under a Creative Commons Attributions 4.0 LICENSE
Published on 04/11/2024

writing digital texts at that time. I just switched to superscript or subscript mode, as offered by most text processing tools, typed the numbers, or the letter w and that was it.

Only later, when I began my PhD thesis and started to program, I made acquaintance with Unicode and the particular pitfalls of typing characters on the computer. Only then I became aware that there is a fundamental difference between typing a letter in superscript or subscript mode in word and typing a superscript or subscript letter as such in a text file, that is, a file that does not allow for various kinds of markup, such as bold or italic font, or underlined text. Specifically starting to write texts in LaTeX — XeTeX in particular, a LaTeX version that allows to type symbols directly in Unicode — changed my view on special characters fundamentally and taught me about the important to know and understand how to type special characters in different contexts.

2 X-SAMPA and Key-Maps

Since at least 2010, I have been using a key-map when writing texts with the help of the VIM text editor (https://github.com/bpj/bpj-vim-keymaps, Jonsson 2017). This key-map allows me to switch into a special mode when typing, in which combinations of characters that I type are directly converted to special symbols. The basic representation of my VIM key-map uses X-SAMPA (Gibbon et al. 1998), but I have extended the key-map myself over the years, adding and modifying key mappings. When I now write a text in text editor (as I do most of the time, since I do not often use Word or LibreOffice), I only have to press the F8 button and when I then type the sequence `"t_hOxt@R"`, it will immediately be rendered as `"tʰɔxtəʁ"`. Switching back to normal text mode only requires me to press the F10 button.

Although this little plugin has helped me a lot (I assume I would not have been able to write my dissertation in time without it), it comes with a great disadvantage, since I can only use it when writing texts with a text editor. When collaborating with colleagues, writing documents in Word or other systems for collaborative writing that require the use of a web browser, I cannot use my key-map system. When having to writer longer sequences or examples with special characters, I often end up opening a text file, typing the sequences there, and then copying the content into the document. Since this does not happen too often, I have never complained about it, neither have I actively tried to find a solution to avoid it.

I have noticed, however, that it definitely has a certain impact on my productivity, albeit a small one. Being able to type special character sequences considerably fast (probably much faster than many of my colleagues), it always makes me nervous when I end up typing special characters in one program and then copying it ot the other. For this reason, I also included key-maps as an option into the EDICTOR software tool (https://edictor.org, List 2017, List and van Dam 2024), although I always knew that most colleagues would not use it, since typing special characters has always been a private issue in linguistics, something nobody wants to talk about with their colleagues.

3 Compose Keys

Only recently, I found a solution that allows me to type special characters freely on all programs of my computer. I learned — by coincidence — that the Linux system of Compose keys can be freely extended, using any private key-map that one defines. The Compose key system on Linux operation systems is a very straightforward way to type special characters that I myself have been using for years now. The idea is that one key on the keyboard (in my case the CAPS LOCK key) is reserved as special Compose key. When typed, the compose sequence is initiated, allowing to type special character combinations afterwards that are then turned into a special character. I have been using the Compose key specifically to type German Umlaut when using an English keyboard. Thus, in order to type an a, I would type `'CAPS LOCK + " + a'` on my computers, in order to type the schwa sound, I would type `'CAPS LOCK + e + e'`, and so on.

While I always liked and enjoyed compose keys, it never occurred to me that I could actively modify their behavior on my computers. It turns out, however, that this is extremely easy (as I learned from a post on the internet of the Debian Wiki). All one hast to do is to create a file `.XCompose` in one's home directory and add Compose key combinations, following the following syntax, where `'CHAR1'` and `'CHAR2'` refer to any alphabetic characters, and `'RESULT'` to any special characters in Unicode.

```
<Multi_key> <CHAR1> <CHAR2> : "RESULT"
```

In order to guarantee that the original Compose key settings are not overwritten (which was particularly important for myself), the file should start with the line `'include "%L"'`. In my particular case, I first defined mappings for superscript characters that allow me to type superscript letters as a combination of the letter to be turned into its superscript counterpart, preceded by the letter `'x'` (indicating we are in X-SAMPA mode) and the circumflex character `'^'`.

```
include "%L"
<Multikey> <x> <asciicircum> <a> : "ᵃ"
<Multikey> <x> <asciicircum> <b> : "ᵇ"
<Multikey> <x> <asciicircum> <c> : "ᶜ"
<Multikey> <x> <asciicircum> <d> : "ᵈ"
<Multikey> <x> <asciicircum> <e> : "ᵉ"
<Multikey> <x> <asciicircum> <f> : "ᶠ"
<Multikey> <x> <asciicircum> <g> : "ᵍ"
<Multikey> <x> <asciicircum> <h> : "ʰ"
<Multikey> <x> <asciicircum> <i> : "ⁱ"
<Multikey> <x> <asciicircum> <j> : "ʲ"
```

With these settings, I can easily type sequences like `[tʰ]`, `[kʲ]`, and the like, and what is even more important, I can expand the settings according ot my future needs. It may sound silly to non-linguists, but the fact that I can finally get rid of the problem of typing phonetic transcriptions freely and quickly across all programs in my computer, has given me an unexpected spur of enthusiasm during the last days that I usually only feel when learning new techniques in programming or when finally understanding mathematical or algorithmic problems that I had been fighting with before.

4 Typing Skills for Linguists

The enthusiasm about the unexpected solution for my typing problems has made me think more closely about the problem of typing special characters for linguists. In linguistics, we have by now many courses at universities or summer schools that help us to develop basic skills needed for a career in the language sciences. Among these are courses and course books that introduce programming with R and Python, introductions to linguistic field work, or introductions to experimental studies in psycholinguistics. What is lacking — to the best of my knowledge — are detailed introductions into the typing of special characters. There is a very good introduction into Unicode by Moran and Cysouw (2018), there is X-SAMPA as a system for typing the International Phonetic Alphabet with ASCII letters, there are several keyboard layouts for Windows that allow to type IPA characters, but I have never seen general recommendations on typing IPA or other symbols in the context of linguistics.

It seems to me, that the problem of handling special characters is handled in a way similar to bibliographies. It is seen as something belonging to the private domain. Everybody should learn how to handle large numbers of references in their papers, everybody should be able to construct some kind of an internal knowledge base, but there is no attempt to make these skills part of the curriculum of students. I have been wondering a lot, why bibliography management is so rarely discussed among linguists. I suspect that it is due to a strange mixture of embarrassment and protectionism. On the one hand, scholars feel very personal about their bibliographies and internal knowledge bases. They might consider them as mediocre, not optimal, and therefore feel that they are not worth to be shared with others. On the other hand, scholars may see their personal knowledge base as something they have been curating a long time and that helps them to write their papers and prepare their classes. As a result, they are reluctant to share their resources and techniques with their colleagues, as they consider them as a certain advantage that they are not willing to share for free.

With typing, it might be similar. I barely hear anybody discussing openly, how they type all those strange characters that appear in their databases. At times, I feel that scholars are getting even obsessed with certain superscripts and tend to overuse them, just in order to show that they can type them nicely. As a result, we find scholars who type [t^s] instead of [ts] or [d^z] instead of [dz] when transcribing alveolar affricates, just because they develop some obsession with the superscript, which is completely unnecessary in this context and not warranted at all by the International Phonetic Alphabet. Whenever I do a closer analysis of the transcriptions that colleagues produce, I find typical errors in their data, resulting from an uncontrolled and unreflected typing of special characters, ignorant of Unicode.

In order to cope with the character typing problem, scholars invent their own combinations, resort to Greek or Russian keyboard layouts, or copy-paste symbols from where they can find them in the internet. The result is an extreme amount of variation, as we have documented in our Cross-Linguistic Transcription Systems project (https://clts.clld.org, Anderson et al. 2018) and as one can easily see when checking individual datasets of our Lexibank repository (https://lexibank.clld.org, List et al. 2022). Thus, a huge part of the variation in phonetic transcriptions and in the codings of those transcriptions in their digital representation can — I am quite sure about this — be attributed to the fact that scholars use ad-hoc solutions for the typing of special characters. If scholars decided to openly share their solutions — specifically with younger scholars — and to interact with their colleagues in order to improve their typing skills, we would see a lot less variation in all the datasets compiled by different scholars.

Similar to systems for the management of bibliographic entries, however, it can be see as equally impossible and also undesirable to impose one system upon the whole field. It should not make a difference if people store their bibliographies in BibTeX format and use tools like JabRef to maintain them, or if they use bibliography managers like Zotero. What is important, however, is that scholars are involved in bibliography management, that they share their solutions with younger people, in order to give them guidance, and that they exchange about the ways in which they organize their knowledge in order to improve not only the transparency of the output of scientific endeavor but also of the daily work that creates leads to this output.

In the same way, I would welcome an open discussion about special character typing in linguistics. I would like to know what tools colleagues use to type IPA characters, whether they consider it important to improve their typing speed, or whether they bother about digital details. It would also be very interesting for me to hear how senior colleagues deal with data typing in their teams and if they require their students to use certain solutions or whether they let them find solutions on their own. Last but not least, it would also be important to discuss about teaching how to type special characters in linguistics in the classroom.

For young scholars who are still in the beginning of their career, I recommend to take the typing of special characters seriously and to invest time into the search for good solutions and into the improvement of their typing skills. When searching for solutions, I recommend to look for tools that can be used across programs and that can be extended and customized easily. The Compose key system that I mentioned seems to be an ideal candidate for me. It seems that similar systems are available also for operation systems other than Linux, including Windows (https://wincompose.info, Hocevar 2024) and Mac OS (Goutte-Gattat 2023). Due to its widespread use across different fields, I furthermore recommend that all linguists should acquire basic knowledge of X-SAMPA. Basic knowledge of Unicode and how Unicode characters can be identified is als a “must know” for all those who are interested in linguistics.

5 Outlook

Although scientific practice has reached a stage now in which scholars who hide their code and their data are often no longer taken seriously. Scientific practice, however, still has many secret workflows and techniques which scholars are reluctant to share. In linguistics, one of these techniques seems to be the typing of special characters. While various solutions exist, special character typing is barely discussed in the classroom or among colleagues. No matter why scholars tend to be silent about special character typing, I think it is time to break the silence and to make sure that solutions and workflows are actively discussed and perceived as part of our scientific work.

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. https://doi.org/10.2478/yplm-2018-0002

Gibbon, Dafydd and Moore, Roger and Winski, Richard (1997): Spoken language reference materials. Berlin and Boston: De Gruyter Mouton.

Goutte-Gattat, Damien (2023): Emulating the Compose key on macOS. Incenp (Blog) 12.1. https://incenp.org/notes/2023/emulating-compose-key-on-macos.html

IPA (1999): Handbook of the International Phonetic Association. Cambridge: Cambridge University Press.

List, Johann-Mattis (2004): Inkongruenz als Indiz – G.G. CORBETTS Kongruenzhierarchie als Mittel der Rekonstruktion, dargestellt am Beispiel der Entstehung von Femininum und Plural im Indogermanischen. Term Paper. Freie Universitat Berlin. https://doi.org/10.17613/vvv6-c466

List, Johann-Mattis (2017): A web-based interactive tool for creating, inspecting, editing, and publishing etymological datasets. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. System Demonstrations. 9-12. https://aclanthology.org/E17-3003

List, Johann-Mattis and Forkel, Robert and Greenhill, Simon J. and Rzymski, Christoph and Englisch, Johannes and Gray, Russell D. (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 and van Dam, Kellen Parker (2024): Computer-Assisted Language Comparison with EDICTOR 3 [Invited Paper]. In: Proceedings of the 5th Workshop on Computational Approaches to Historical Language Change. 1-11. https://aclanthology.org/2024.lchange-1.1

Meier-Brugger, Michael (2002): Indogermanische Sprachwissenschaft. Berlin and New York: de Gruyter.

Moran, Steven and Cysouw, Michael (2018): The Unicode Cookbook for Linguists: Managing writing systems using orthography profiles. Berlin: Language Science Press. https://langsci-press.org/catalog/book/176

Hocevar, Sam (2024): WinCompose [Software, Version 0.9.10]. http://wincompose.info

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.