Exploring Finno-Ugric linguistics through solving IT problems

Tobias Weber,*[*] Jeremy Bradley[‡]

[*]Ludwig Maximilian University of Munich
Institut für Finnougristik / Uralistik
Ludwigstraße 31/III
D-80539 München
weber.tobias@campus.lmu.de

[‡]University of Vienna
Institut EVSL, Abteilung Finno-Ugristik
Campus AAKH, Hof 7-2, Spitalgasse 2-4
A-1090 Wien
jeremy.moss.bradley@univie.ac.at

Abstract

This paper seeks to introduce our approach of integrating computational methods, digital resources, and computer literacy skills into the curriculum of Finno-Ugric (Uralic) linguistics. Our starting point is the class Digital Resources in Linguistics, which we taught at the Institute of Finno-Ugric/Uralic Studies at LMU Munich in 2017; our eventual aim is the compilation of teaching materials (a textbook with supplementary online materials) on this subject matter and their integration into Finno-Ugric curricula. While there are numerous high-quality textbooks on computational linguistics, our endeavour is more tied to the framework of Digital Humanities, stressing the background in humanities and social sciences rather than details of specific technologies, and attempting to be conscientious to the specific needs, interests, and skills of our students. This endeavour is happening within the context of the ongoing internationalization of our research discipline, exemplified by the Erasmus+-strategic partnership INFUSE (Integrating Finno-Ugric Studies in Europe, 2015–2018), which in its next iteration, COPIUS (Community of Practice in Uralic Studies, 2018–2021) will also focus on the development and pooling of teaching materials.

1. Introduction

In order to discuss the challenges of conveying computer literacy skills in classrooms of subjects traditionally associated to the humanities, we need to consider our targeted audience first and explore their access to computers and programming. An overview of existing literature will help shed some light onto issues our students may experience in using these materials. Based on these observations we will outline our approach by highlighting overlaps between the topics which are already covered in our courses and computational methods and tools that can be used in relation to them. Finally, we will present ways of integrating our concept into the curricula of the European institutes for Uralic studies and exemplify them using the curriculum at LMU Munich, where we have taught our pilot course.

1.1. Target audience

To understand our role in this endeavour, let us step back and discuss our professional relationship to our students and consider their needs and interests in acquiring computer literacy skills. As educators at university level in the digital era, we, in spite of teaching in a discipline traditionally seen as part of the humanities, instruct students who have been exposed to advanced technologies throughout their previous educational careers as well as in their social lives. This means that we are not building knowledge from scratch in absence of a preexisting foundation. However, our students are enrolled in linguistic, philological, or ethnographic courses, and are generally aiming to acquire an education within the domain of the humanities, rather than in more technically-oriented subject fields. They generally are users of applications which they find online or through recommendations by teachers or supervisors but have no expertise in developing applications of their own, or the intention of doing so.

From conversations with our students prior to our course, we had gathered that most regard computational methods as too abstract and not relevant enough for their own research, and that they are reluctant to use applications or technologies with which they do not feel confident in their research. At the same time, it was obvious to us that many tasks our students face on a regular basis could be streamlined if they could overcome these reservations. Our approach is informed by the discord between reservations our students - who do not consider themselves “tech people” - feel, and the profit they could garner from basic IT literacy in their work. Our course aims to create a basic understanding of how computers handle language data, and help students develop a critical view of the possibilities and limitations in electronic data processing. We are educating “cross-disciplinary thinkers” (Furman, 2015, p. 4). Whether students go on to take classes in computational linguistics of digital humanities or not, they will see computers in a different light. Even students who do not themselves start working with software tools we cover in our courses will profit from this better understanding, as it will enable better communication with programmers and coders in collaborative efforts.

1.2. Approaches in existing literature

Computer-Assisted Language Learning constitutes one important field of study within language pedagogy for linguists. While this field is only peripherally related to our endeavour, it comprises many insightful essays on the use of computers in classrooms of language and other "soft sciences". Among these papers, there are also critical voices calling for considerate use of computers and technology, a position which is derived from the incorporation of insights from media science and philosophy. Richard Kern (2014) uses the term *pharmakon* to describe the role of technology in a language classroom – it can be both the cure for an ailment and a poison. This point of view calls upon educators and students to deal with computers and technology critically, i.e. to acknowledge pitfalls as well as benefits.

The amount of literature on NLP and computer skills for linguists has increased immensely over the last decade. Some books use the "dummies" approach and assume as little prior knowledge as possible or try to convey practical skills in using computers and particular software. Among those, some authors claim to teach broad skills for the labour market or to "prepare you for success in a modern world full of computers" (Wempen, 2014, p. 1). While the role of computer literacy in private economy cannot be underestimated, it appears that a university course should primarily tackle specific issues within the discipline but also convey skills from which students will profit in their later lives regardless of the career paths they choose. Another subset of literature is directly aimed at students of computational linguistics, e.g. (Jurafsky and Martin, 2009; Carstensen et al., 2009), usually coming from a technology background and with the goal of teaching skills for software applications as well as programming and coding languages. These books are, without doubt, the benchmark for textbooks on computational linguistics and it should be the hope of every educator in digital humanities that their students can go on to read and understand this set of literature or acquire practical knowledge of a programming language, should it match their interests and needs. However, as outlined above, our target audience is not particularly interested in writing programmes and would feel easily intimidated when confronted with theoretical concepts from computer science or mathematics (e.g. discrete mathematics for the description of automata or formal languages). Hence, our objective is rather to foster a general understanding and awareness of how applications relevant to our subject field work.

It should furthermore be the goal of our course and teaching materials to explain these abstract concepts as practical knowledge and formalism as an excursus rather than primary content of the class. This becomes most relevant in assessment, where such knowledge should not be tested explicitly – students should be enabled to understand the concepts but not be tested on formalisms. A few textbooks take this approach, e.g. (Dickinson et al., 2012), i.e. that "[t]he goal of our courses is to show students the capabilities of [NLP] tools, and especially to encourage them to take a reflective and analytic approach to their use." (Dickinson et al., 2012, p. xiii). This textbook, like the teaching materials we aim to create, puts emphasis on the discussion of computational methods, on students' own work, as well as in the academic community – questioning the "engineering mentality" (Popoveniuc, 2010) and the sociocultural aspect of technology in scientific discourse, cf. (Schmidt, 2010).

2. Why does it matter?

The relevance of our project, in spite of the large existing body of textbooks on computational linguistics, stems from the discrepancy between the issues and approaches followed in mainstream textbooks and topics relevant to our target audience: first of all, literature covering NLP issues on Uralic languages is still scarce. This is not to say that there is no scientific output on NLP pertaining to Uralic languages, but that publications are either very specific in their target language (mostly on the three Uralic national languages of Europe, Hungarian, Finnish, and Estonian) or have a strong focus on technical issues (e.g. publications by Giellatekno, the Centre for Saami language technology; publications in the Northern European Journal of Language Technology). We consider our project to act as a bridge between "paper and pencil" linguistics and the research carried out by computational linguists by facilitating students' access to this field of study, or at least educating them about the range of possibilities offered to the study of Uralic languages by NLP.

Furthermore, Uralic linguistics has a long research tradition which has given rise to peculiarities in terminology or practices in handling language data, e.g. the Finno-Ugric Transcription FUT (Setälä, 1901), which predates IPA as a transcription standard. This means that working with Uralic language data requires the researcher to know about these conventions. While transliteration between transcription systems does not pose an obstacle to a computational linguist, cf. (Bradley, 2017), it can dishearten scholars outside the Uralic scientific community to work with our data, cf. (Widmer, 2004). These peculiarities are not only potentially alienating to scholars outside of our discipline, they also give rise to difficulties when using software applications not specifically designed for Uralic languages, for example transcription software or programs used for linguistic annotation. For example, as many values in FUT lack Unicode code points of their own, they can only be represented using combining characters. The appropriate usage of these often poses an insurmountable hurdle to students not trained in appropriately dealing with such issues.

A further argument in favour of our project pertains to the ethical duties of linguistics – as researchers on social subjects, we need to ensure a good reciprocal relationship between us and our informants, cf. (Moran, 2016), and need to prioritise the communities' needs and rights, cf. (Austin, 2010). This means that we, as instructors, need to ensure that all of our students are familiar with best practices in handling language data and using available electronic resources: Many Uralic languages are endangered, have little electronic resources, and thus require more documentary research. Should our students aspire to conduct fieldwork, it is imperative that they know about technological methods in archiving and transcribing, see (Austin, 2006; Gippert, 2006; Dry, 2008). They might also be asked to contribute to revitalisation efforts, which nowadays also include technological contributions such as learning or dictionary applications, cf. (Grenoble and Whaley., 2006; Grenoble et al., 2008; Dauenhauer and Dauenhauer, 1998; Hinton et al., 2018).

3. Contents of the syllabus

While creating the syllabus for our course, we have followed the usual sequence of topics introduced in courses in computational linguistics or basic computer science. This includes overviews of "how computers think", data types and structures, character encoding and digital language data representation, basic concepts of programming, use of corpora, regular expressions, and comments on current topics in computational linguistics like machine translation, machine learning, or OCR. This discussion of current issues is essential for giving our students an idea of the work carried out in neighbouring disciplines: should they require IT assistance with one of their projects, or if they are working in interdisciplinary teams, they will need a basic understanding of the capabilities and limitations of certain computational tools in order to communicate their needs and to give informed comments on project work. It should also help them in planning their work to consider whether computational methods could reduce their workload or make their tasks easier – and to give them an idea of what can be expected of linguistics software, which issues it can help to solve and where its limits lie. If someone asks them why they have or have not used a particular method or software tool, they should be able to give a confident, reasoned answer and not have to say that they did not consider a methodology because they were afraid to use a computer or did not understand the results an application delivered. For the topics covered in the class, we plan to add practical work, not in active programming, but in working with code, to our course and teaching materials: supplying basic code segments and explaining how they work, and how they can be tweaked to deliver the desired results.

An example would be the creation of a frequency list. We would supply a script in a more-or-less human-readable code with comments and highlighted code fragments which have to be replaced in order to, for example, get the output in a desired alphabetical order (NB! The alphabetical order differs between different Uralic alphabets), or to read data from a specific file. All of these examples will come with a sandbox (or equivalent applications which are available online) as well as test files so that students can practise with dummy files before working on actual data for their own research tasks.

4. Linking with the curriculum

During our pilot course, most students were enrolled in later years of undergraduate or graduate programmes. While it is desirable that all students learn about methods in digital humanities and acquire some skills during their studies, it is questionable whether this task should be left for advanced students or rather tied into first-year modules. An important factor in favour of the second option is that an understanding of basics like Unicode can make a student's work easier and prevents them from using obsolete custom solutions, e.g. fonts to encode special characters, copy-pasting often incorrect symbols (e.g. Cyrillic and Latin characters that are visually identical, but not so for language processing tools). It can furthermore open their minds to computational approaches in our discipline and might lead them to consider introductory classes in computational linguistics, digital humanities, or computer science as their electives. It also reduces the likelihood of last-minute efforts to learn a software or a method for a term paper or dissertation. Computers become increasingly involved in our everyday work, and knowledge about how to use them correctly becomes more and more a requirement than an optional skill.

Most curricula in linguistics and philology contain at least one class on research methodology in an early semester (first and/or second). Such courses cover basics of literature search, citing techniques, note taking, or presentational skills – solid skills of information literacy (cf. `informationskompetenz.de`). But – moving into an increasingly digital age where everyday tasks like citations are assisted by software – why should information literacy be taught detached from computer or digital literacy skills? This does not mean that we aim to replace courses in information literacy – on the contrary, students should be able to do all the tasks their computer will do for them, to avoid over-reliance on technology. Therefore, we propose a system of mutual learning, where "paper and pencil" methods are mirrored in computational tasks.

5. Bringing it all together

As outlined above, we are aiming to foster awareness of computational methods among undergraduate students, ideally in their first year. First-year undergraduate students at the Institute of Finno-Ugric/Uralic Studies at LMU Munich must attend: a two-semester introduction to Finno-Ugric studies with a focus on history of the field, development of the languages, and basic typology; modules on phonetics and phonology; an introduction to linguistic theory; practical courses on scientific writing and information literacy; a language class. The possibilities to tie in computational approaches are numerous:

5.1. Introduction to Finno-Ugric studies

Within the introduction to Finno-Ugric studies, students learn about concepts of stem alternations – present in Finnic, most Saamic languages, and Nganasan – and vowel harmony – present in some shape or form in more Uralic languages than not, for example in Hungarian: the dative suffix has two variants used depending on the quality of the vowels of the base words, *-nak* after back vowels (*barát* 'friend' → *barátnak* 'to the friend') and *-nek* after front vowels (*zseb* 'pocket' → *zsebnek* 'to the pocket'). The resulting allomorphy can be used to explain search functions using regular expressions (e.g. implementing a search function that will find both variants of the aforementioned dative suffix) and highlight how computer applications have to be constructed to allow for this allomorphy (e.g. in lemmatising/stemming, morphological analysis).

5.2. Phonetics and Phonology

An integral part of the phonetics and phonology modules is acquiring practical skills in transcribing spoken language into a conventional transcription system, either IPA, FUT, or a standard orthography of the language. While there are numerous possible excursuses from reading a spectrogram and working with applications like Praat to speech recognition and synthesis, these topics are mostly covered by entire modules at the Institute of Phonetics and Speech Processing where our students can take classes as electives. However, in the compulsory class at our institute, students practise writing and reading phonetic transcriptions, and become acquainted with the Cyrillic alphabet used by the majority of Uralic languages spoken in the Russian Federation (e.g. Mari, Udmurt). This enables us to address three issues: firstly, students will have to submit coursework throughout the semester, and handing it in electronically or in type-written form will require the use of special characters on a computer. They should know about Unicode and encodings, as well as tools for writing and transcribing with Unicode characters (e.g. `transcribe.mari-language.com`). Secondly, they will want to use the right characters in writing, so teaching them how to write Cyrillic characters on their keyboards becomes relevant (in addition to special characters found in Uralic languages utilizing the Latin alphabet, e.g. Hungarian <ű>, <ő>). Thirdly, we can present tools for transliteration between orthographies and explain how these simple search-and-replace processes are coded.

5.3. Information Literacy and Scientific Writing

The module on information literacy and scientific writing can be amended by a brief overview of linguistics software, or how to make best use of basic tools like using the regular expression function in Word. Furthermore, information literacy should also include a discussion about research ethics, stressing the importance of using best practice in scientific work. In fact, the information literacy course is most easily converted into an information and digital literacy course and should be seen as a platform for teaching the desired skills mentioned above, and also introducing students to more modern methods of finding and organizing references, e.g. Google Scholar and bibliographic software.

5.4. Language Classes

Lastly, students taking a language class will face the challenge of writing in their target language using all special characters of its orthography (see above). Moreover, they will want or need to use electronic resources like dictionaries, spell-checkers, morphological synthesis and analysis, or even corpora. We feel that we should do more than hand them a list of available resources and referring them to documentation on these but should rather give our students ideas of how to utilise the digital resources appropriately and how to get the desired results out of them. This will also help them with other course work in the following years.

5.5. Summary

As could be seen from the discussion above, the first-year modules in our undergraduate programme already give a basis for weaving in basic IT skills through highlighting the computational side of the (paper and pencil) processes or techniques presented. Within the course of the first year, this approach would cover Unicode, special characters and encodings, transcription and transliteration, regular expressions, morphological analysis and synthesis, electronic resources, and excursuses and discussions of issues in computational linguistics. This covers all the basics which would be covered in the initial chapters of any computational linguistics or computer science textbook. Students could then go on to continue along this pathway by taking classes of computational linguistics, or by using these methods for their own research. Either way, they will have acquired important knowledge about using computers efficiently and can use this knowledge throughout their studies.

6. Evaluation of the pilot

We taught our course "Digital Resources in Linguistics" as a pilot project in May, June, and July 2017 as a seminar, open to undergraduate and graduate students of Uralic studies and neighbouring disciplines as an elective course. Our course consisted of six four-hour sessions with a limited workload worth 3 ECTS points. Six students enrolled for our course. As is common practice in universities, our students were asked for their evaluation of the module at the end of the term. Given the experimental nature of our course, we requested some more detailed information on students' evaluation of their learning progress, and obtained permission to reproduce their answers in print. This feedback showed that the students felt that they had learned much ("I now feel more confident in handling the more technical side of my research") and found the workload appropriate. While such an evaluation cannot guarantee the success of an approach, it demonstrates that our idea gets positive feedback ("would greatly recommend") and that this course, which was held as an optional module, managed to provide interesting insights. The greatest point of criticism raised by students is that it was not offered at the optimal point in their studies: "The only regret I have about the course is that it came too late in my studies, namely in my last semester of my Master's. Would it have taken place in my second or third semester of my Bachelor's, I may have considered taking more courses in that direction." As there are no elective courses in the first two years of our Bachelor's programme, however, we currently have no possibility of offering this course at an earlier stage of studies. Changes to the curriculum would be necessary for this to happen in future.

Students' interest also showed in their independent projects which they had to conduct for receiving the credits on this module: brief case studies where computational methods are currently used or could be used in the future to solve problems in their field of interest including a discussion of potential benefits and pitfalls ("As a conversational analyst, I still very much rely on the 'pen and paper'-method of data analysis; however, I now have a better understanding of how to work with the raw data in the transcription phase").

Students enjoyed learning about computer basics, discussions of software use, and thematic excursuses. These topics inspired conversations with students and prompted individual questions (a student working part-time as a proofreader: "This gain in cross-disciplinary work also proved to be useful in working life"). Teaching principles of programming and working with scripts turned out to be more difficult for our students and might require reworking of our teaching materials. Furthermore, we see the necessity to address commonly used applications (e.g. ELAN, R, LaTeX) more directly, as it appeared that our students had already heard about such software before our class (in some cases, competencies with these software packages were expected from them) but without receiving instruction in their usage and range of capabilities. There were no issues with the thematic progression and the order of topics.

7. Outlook

We hope to teach this course again in future, as budget and student interest allows, both in Munich and at our partner institutions across Europe. For future iterations of this course we are creating a script and reading list to give the course a more solid outline. Eventually, we intend to create a textbook (to be published online in an openly accessible manner) and supplementary online materials on the basis of our course materials.

Our efforts so far have been happening within the framework of the Erasmus+-strategic partnership INFUSE (Integrating Finno-Ugric Studies in Europe, 2015–2018, cf. `www.infuse.finnougristik.uni-muenchen.de`), which is administered by the Institute of Finno-Ugric/Uralic Studies at LMU Munich, and consists of eight European Finno-Ugric departments (Hamburg, Helsinki, Munich, Szeged, Tartu, Turku, Uppsala, Vienna). Funding has been guaranteed for a continuation of this strategic partnership, COPIUS (Community of Practice in Uralic Studies, 2018–2021, cf. `www.finnougristik.uni-muenchen.de/aktuelles/nachrichten/copius`; Budapest has now joined our consortium). In COPIUS, our focus will lie on the development and pooling of teaching materials. We have committed ourselves to creating an openly accessible integrated online learning platform for our subject field, including a general introduction to Finno-Ugric studies, and a number of expansion modules (e.g. on fieldwork methods, etymology, individual Finno-Ugric languages). Our teaching materials can constitute an additional module in this learning platform.

We would like to reiterate that this does not mean that we are trying to replace traditional computer science or computational linguistics courses. We are rather aiming to close the gap between students with prior knowledge of computer science and students without exposure to principles of computing. This enables the latter to make best use of available tools and resources and gives all students guidelines for basic NLP tasks pertaining to Uralic languages, while simultaneously presenting an apparatus of computational methodologies.

We hope that our contribution will help students to see the "mysticism" of computers and computational methods in a different light, and thereby help to bridge the digital divide.

8. References

Peter K. Austin. 2006. Data and language documentation. In Jost Gippert, Nikolaus P. Himmelmann, and Ulrike Mosel, editors, *Essentials of Language Documentation*, pages 87–112. de Gruyter, Berlin.

Peter K. Austin. 2010. Communities, ethics and rights in language documentation. *Language Documentation and Description*, 7:34–54.

Jeremy Bradley. 2017. Transcribe.mari-language.com: Automatic transcriptions and transliterations for Mari, Tatar, Russian, and more. *Acta Linguistica Academica*, 64(3):369–382.

Kai-Uwe Carstensen, Christian Ebert, Cornelia Ebert, Susanne Jekat, Hagen Langer, and Ralf Klabunde, editors. 2009. *Computerlinguistik und Sprachtechnologie: Eine Einführung*. Spektrum, Heidelberg, third edition.

Nora Marks Dauenhauer and Richard Dauenhauer. 1998. Technical, emotional, and ideological issues in reversing language shift: examples from southeast alaska. In Lenore A. Grenoble and Lindsay J. Whaley, editors, *Endangered languages: Language loss and community response*, pages 57–98. Cambridge University Press, Cambridge.

Markus Dickinson, Chris Brew, and Detmar Meurers. 2012. *Language and Computers*. John Wiley & Sons, Chichester.

Helen Aristar Dry. 2008. Preserving digital language materials: Some considerations for community initiatives. In Wayne Harbert, Sally McConnell-Ginet, Amanda Miller, and John Whitman, editors, *Language and Poverty*, pages 202–222. Channel View Publications, Bristol.

Robert L. Furman. 2015. *Technology, Reading, and Digital Literacy. Strategies to Engage the Reluctant Reader*. International Society for Technology in Education, Eugene, OR.

Jost Gippert. 2006. Linguistic documentation and the encoding of textual materials. In Jost Gippert, Nikolaus P. Himmelmann, and Ulrike Mosel, editors, *Essentials of Language Documentation*, pages 337–361. de Gruyter, Berlin.

Lenore A. Grenoble and Lindsay J. Whaley. 2006. *Saving languages: an introduction to language revitalization*. Cambridge University Press, Cambridge.

Lenore A. Grenoble, Keren D. Rice, and Norvin Richards. 2008. The role of the linguist in language maintenance and revitalization: Documentation, training and materials development. In Wayne Harbert, Sally McConnell-Ginet, Amanda Miller, and John Whitman, editors, *Language and Poverty*, pages 183–201. Channel View Publications, Bristol.

Leanne Hinton, Leena Huss, and Gerald Roche, editors. 2018. *The Routledge Handbook of Language Revitalization*. Routledge, New York - London.

Dan Jurafsky and James H. Martin. 2009. *Speech and language processing: an introduction to natural language* *processing, computational linguistics, and speech recognition*. Pearson Education International, Prentice Hall, Upper Saddle River, NJ.

Richard Kern. 2014. Technology as Pharmakon: The Promise and Perils of the Internet for Foreign Language Education. *The Modern Language Journal*, 98(1):340–357.

Mary H. Moran. 2016. The digital divide revisited: Local and global manifestations. In Roger Sanjek and Susan W. Tratner, editors, *eFieldnotes: The Makings of Anthropology in the Digital World*, pages 65–77. University of Pennsylvania Press, Philadelphia.

Bogdan Popoveniuc. 2010. What is a technological mentality? In Viorel Guliciuc and Emilia Guliciuc, editors, *Philosophy of Engineering and Artifact in the Digital Age*, pages 125–135. Cambridge Scholars Publishing, Cambridge.

Colin T. A. Schmidt. 2010. Cognitive life re-engineered. In Viorel Guliciuc and Emilia Guliciuc, editors, *Philosophy of Engineering and Artifact in the Digital Age*, pages 67–80. Cambridge Scholars Publishing, Cambridge.

Eemil Nestor Setälä. 1901. Über die transskription der finnisch-ugrischen sprachen. *Finnisch-ugrische Forschungen*, 1:15–52.

Faithe Wempen. 2014. *Computing Fundamentals: Digital Literacy Edition*. John Wiley & Sons, Chichester.

Anna Widmer. 2004. Reconnecting and Reconsidering: Remarks on the Final Discussion of the International Linguistic Symposium “Reconnecting Finnic”. *Linguistica Uralica*, 2004(3):197–212.