IWCLUL 2021

The Seventh International Workshop on Computational Linguistics of Uralic Languages

Proceedings of the Workshop

September 23–24, 2021

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-954085-82-4

Preface

The seventh edition of ACL SIGUR's meeting IWCLUL is organised in conjunction with Electronic writing of the peoples of the Russian federation (EWPRF 2021) in Syktyvkar, Russia, the actual event was organised Online only due to the situation in 2021. This is the seventh event in the series and second hosted in Russia.

For the current proceedings of The Seventh International Workshop on Computational Linguistics for Uralic Languages, we accepted 8 high-quality submissions about topics ranging from overviews and insights into traditional language technology and resources all the way to modern neural network approaches in Uralic context and speech technology. The papers cover a wide range of Uralic languages from Finnish and North Sámi to Udmurt and Komi-Zyrian with several papers giving insight on whole range of Uralic langauges. Whereas some papers describe language-specific research, others compare different languages or work on small Uralic languages in general. These contributions are all very important for the preservation and development of Uralic languages as well as for future linguistic investigations on them.

As the conference is organised in collaboration with EWPRF, we have two full days of presentations as well as a round table, a regular business meeting of ACL SIGUR and time for discussions. The current proceedings include written papers of all of the IWCLUL oral presentations.

— The board of ACL SIGUR, October 13, 2021, Online / Syktyvkar

Organizing Committee

Association of Computational Linguistics' Special Interest Group for Uralic Languages (ACL SIGUR: `https://acl-sigur.github.io`)

Local organisers at Syktyvkar: `http://conference.krags.ru`

Programme Committee

- Flammie A Pirinen, UiT norgga árktalaš universitehta
- Timofey Arkhangelskiy Universität Hamburg
- Trond Trosterud, UiT
- Thierry Poibeau, LaTTiCe-CNRS
- Andrew Krizhanovsky, Institute of Applied Mathematical Research of the Karelian Research Centre of the Russian Academy of Sciences ordinary
- Svetlana Toldova, Higher School of Economics
- Mika Hämäläinen, University of Helsinki
- Francis M. Tyers, Indiana University Bloomington
- Csilla Horváth, Research Institute for Linguistics, Hungarian Academy of Sciences
- Jeremy Bradley, Ludwig Maximilian University of Munich
- Veronika Vincze, Hungarian Academy of Sciences, Research Group on Articial Intelligence
- Michael Rießler, Albert-Ludwigs-Universität Freiburg
- Andrey Kutuzov, University of Oslo
- Jack Rueter, University of Helsinki
- Joshua Wilbur, Univerity of Tartu
- Tommi Jauhiainen, University of Helsinki

Table of Contents

Conference Program

Thursday, September 23, 2021

13:45–14:00 *A never-published atlas of Udmurt dialects*
László Fejes

14:00–14:15 *Digitizing print dictionaries using TEI: The Abaev Dictionary Project*
Oleg Belyaev, Irina Khomchenkova, Julia Sinitsyna and Vadim Dyachkov

Friday, September 24, 2021

11:30–11:45 *Keyword spotting for audiovisual archival search in Uralic languages*
Nils Hjortnaes, Niko Partanen and Francis M. Tyers

11:45–12:00 *Evaluating Transferability of BERT Models on Uralic Languages*
Judit Ács, Dániel Lévai and Andras Kornai

12:00–12:15 *No more fumbling in the dark - Quality assurance of high-level NLP tools in a multi-lingual infrastructure*
Linda Wiechetek, Flammie A Pirinen, Børre Gaup and Thomas Omma

12:15–12:30 *Low-Resource ASR with an Augmented Language Model*
Timofey Arkhangelskiy

12:30–12:45 ***Discussion***

12:45–13:00 *The Current State of Finnish NLP*
Mika Hämäläinen and Khalid Alnajjar

13:00–13:15 *Overview of Open-Source Morphology Development for the Komi-Zyrian Language: Past and future*
Jack Rueter, Niko Partanen, Mika Hämäläinen and Trond Trosterud

14:30-15:30 ***ACL SIGUR meeting***

A never-published atlas of Udmurt dialects

Fejes, László
Hungarian Language Center for Linguistics
fejes.laszlo@gmail.com

Abstract

In the first decade of the 21th century, an atlas of Udmurt dialects was prepared for publication. Although hundreds of maps and legends were completed, due to no hope for publication, the project was never finished. The paper describes the material the atlas was based on, how the collection of exercise books was digitized and prepared for the purpose of a dialectal atlas, and how the atlas was generated from the data. The paper also presents some decisions that had to be made during the preparation of the atlas. Finally, the never-published atlas is compared to the published atlas of Udmurt dialects. Despite that the history of the atlas is far from a success story, it shows that, if data are available, a linguistic atlas can be produced even using low-budget tools, in a do-it-yourself way.

Пуштросэз

Кызь одӥгетӥ даурлэн нырысетӥ аръёсаз удмурт диалектъёсъя атлас поттыны дасямын вылэм. Кӧня ке сю карта но солы символъёсын валэктонъёс лэсьтэмын вылэм но, сое поттыны осконлык ышем бере, ужез пумозяз вуттӥллямтэ. Та статья маде, кыӵе материал-тодэтъёс вылэ пыкъяськыса атлас лэсьтэмын вылэм, кызьы но кыӵе тетрадьёс та атласлы шуыса дасямын но дигитализировать каремын вал. Статьялэн пумаз поттымтэ атлас мукетыныз, удмурт диалектъёсын поттэм атласэн ӵошатэмын. Атласлэн историез азинэс ӧй вал ке но, со возьматыны быгатэ: тодэтъёс вань дыръя кылъя атласэз, дунтэм тӥрлыкъёсты уже кутыса но, “киуж амалэн” дасяны луэ.

1 Introduction

Usually, IWCLUL papers present current achievements in computational approaches to Uralic languages. This paper is exceptional in the sense that it presents a more-than-a-decade-old project, which got stuck in its final phase, although it could have produced an (almost) unprecedented result: an atlas of the Udmurt dialects (based on its working title, *Удмурт вераськетъёсъя атлас*, henceforward УВА). The word ”almost” indicates that the first volume of another atlas of Udmurt dialects (Насибуллин et al. (2009), henceforward ДАУЯ) was published approximately at the same time when the discussed atlas should/could have been published. Interestingly, the two atlases are so different in their aims and methods, that they cannot even be considered competitors. As ДАУЯ was the first atlas of the Uralic languages of the Russian

Seventh international workshop on computational linguistics of Uralic languages, pages 1–11
September 23–24, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-1-954085-82-4_001

Federation, it could have been an interesting situation that Udmurt, the only Uralic language with a dialect atlas, could have immediately two of them.

The reasons for the project got stuck are complex. First of all, there was no hope to get financial support for publication. Online publication in PDF format was out of question for several reasons. The main reason is that if the atlas is available online, it is even more hopeless to get financial support to publish it in print. The prestige of an online publication is much lower even today than the prestige of a publication in print, and the difference was even more considerable more than a decade ago. It seemed reasonable to wait for better circumstances. Moreover, the author had permission from the Department of General and Finno-Ugric Linguistics of the Udmurt State University to use the data collected by them for the purpose of publishing a printed atlas. For the same reason, the publication of the bare database was also out of the question. In addition, the author had to leave academia in 2010 and worked outside academia for a living, without time and force to work on the atlas, including search for financial support for publication. When the author could return to research in 2016, he had very different tasks and could find time at least to document the former project only recently.

Section 2 presents where the idea for УВА came from. In Section 3, it is described how a digitally processable data set was produced from the available material. Section 4 discusses the way of generating an atlas from these data. Section 5 outlines the differences between УВА and ДАЯЛ. Section 6 contains some thoughts on the possible future of the УВА project.

2 Background

Ever since the middle of the 1980s, the students of Udmurt philology at the Udmurt State University have had to collect dialect materials from their home village in the second year of their study, and almost every year, they have gone to an expedition together at the end of the year to collect similar materials. The collected material, hand-written into exercise books, consists of two types: texts and the answers to a questionnaire, which will be shortly presented in 2.1. In 2004, the author learnt that a large number of questionnaires were stocked in the rooms of the Department of General and Finno-Ugric Linguistics of the Udmurt State University, not used for any linguistic purposes. Despite that the reliability of the material can be questioned (see 2.2), the author thought this collection was too valuable to be left untouched. The most straightforward idea was to make a dialect atlas based on the material. To make an atlas, survey sites have to be chosen — in 2.3, the applied method will be presented.

2.1 The 400 word program

The questionnaire mentioned above was put together by Valentin Kelmakovich Kelmakov, a (if not the) leading specialist of Udmurt dialectology. It is difficult to determine when the survey was assembled or first published, but the earliest exercise book with the questionnaire is dated to 1983. For the atlas, Кельмаков (2002) was used as a reference.

The questionnaire consists of 400 questions (the name *the 400 word program* — Udmurt 400 кылъем программа — comes from this). In most of the cases, the fieldworker says a Russian word or phrase, which the consultant has to translate into Udmurt. The fieldworker has to try to find a form relevant to the phenomenon the question serves to observe. E.g. question 221 should observe the use of affricates in the given dialect, and asks for the word ‘good’ (‘добрый, хороший’): /d͡ʑet͡ɕ/ ∼ /d͡ʑeɕ/. However, in some dialects, this word is absent or used in a very restricted way, in some greeting forms. If the consultant answers with another word meaning ‘good’, the fieldworker has to record the given form but also to try to ask for synonyms, or ask for the greeting forms containing the searched word. Of course, these rules are applicable for the questions on phonological and morphological phenomena, but not for lexical questions. In addition, there are also semantic questions, when the consultants are given an Udmurt word and they have to translate it into Russian.

The questionnaire consist of 309 questions on phonology,[1] 69 questions on morphology,[2] 18 questions on the lexicon (vocabulary) and 4 questions on semantics. In addition to the 400 questions, the fieldworker has to record 12 paradigm forms (present tense, positive and negative 1SG …3PL forms) of two verbs (*тодыны* ‘to know’, *кутскыны* ‘to begin’), i.e. there are 24 additional questions in the questionnaire.

2.2 The material collected

Between 1983 and 2004, more than 3000 exercise books were filled with answers on the questionnaire. The material is geographically unbalanced: since most of the students come from Southern Udmurtia, especially from the environs of Izhevsk, it is not rare that there are more than five, sometimes more than a dozen surveys from the same settlement. Northern Udmurtia is much less documented, while data from the dialects outside Udmurtia are rather sporadic.

In addition, the quality of the data is sometimes questionable. Data were collected by students, not professional fieldworkers. Theoretically, they are checked by the teachers, but, on the one hand, some exercise books seem to be unchecked (or checked but not corrected); on the other hand, for lack of sound recordings, the teachers cannot check whether the written data correspond to the answers given by the consultants. In some cases, it is clear that the student did not understand the task (the recorded answers are irrelevant to the studied phenomenon), or could not consistently record the data. A typical case is when in the answers to the first questions, which aim to reveal whether the dialect has *ы* /ɨ/, *ы̆* /ĭ/ or *ъ* /ə̆/, *ъ* or *ы̆* is recorded in all the cases, but later, in answers to other questions, only *ы* occurs. In addition, some exercise books are clearly copied from others (self-evidently, these were not used for the project), and it is possible that in some other cases, copying is not so conspicuous. Nonetheless, basically the material seems to be reliable. Data from the same settlements usually show more differences than one would expect if students simply copied the exercise books from each other; however, they are quite consistent to be done at random. Since the students usually document their own dialect, in a certain sense, they are more competent fieldworkers than well-trained but outsider linguists.

[1] In fact, in the Udmurt and Russian texts, they are called *phonetic* issues, but it seems that Udmurt (Russian?) linguistic tradition does not always make such a strict distinction between phonetics and phonology as the western one. In any case, most of the problems observed by these questions should be classified as phonological in the western tradition.

[2] In many cases, these are rather (morpho)phonological questions related to certain suffix morphemes.

2.3 Preparing for an atlas: the choice of survey sites

Theoretically, all of the documented settlements could have been survey sites of the atlas. This choice could have had two disadvantages. First of all, all the available data should have been digitized, although many of these are redundant, because they come from neighbouring settlements without considerable linguistic differences. In addition, too dense survey sites make the map less readable. Moreover, since different areas are documented at a different level, in some areas survey sites could have been dense, while in other areas sparse. Even worse, the density differences would have reflected the number of the students from the area, not the number of the Udmurt settlements (or speakers).

Therefore, a rectangular grid was formed on the map with squares about 15×15 kms. Each square got a two-character code: the first character (`a–v`) showed its latitude (`a` is the southernmost, the latitude of Naberezhnye Chelny, while `v` the northernmost border of Udmurtia), the second one (`A–N`) showed its longitude (`A` is the westernmost, `N` is the easternmost border of Udmurtia). For each square, one representative settlement was chosen, usually the one which was documented by the most surveys. Minimally two filled questionnaires were needed to appoint a survey site for the atlas. Unfortunately, in some cases, the chosen villages, although belonging to different squares of the grid, are quite close to each other, while some territories seem to be uncovered. Finally, 81 survey sites where chosen in the territory of Udmurtia. Later, two survey points were added from the Kirov Oblast (since here there were no settlements documented by two questionnaires, the data from two different but close villages were contracted in both cases) and two from Tatarstan, represented on the same map. In addition, nine survey points were added from Tatarstan, five from Bashkortostan and one-one from the Mari El and the Perm Oblast (nowdays Krai), respectively, which were represented outside (under) the map .

Every survey site had a four-character location code consisting of two letters and two digits. The first two characters showed which grid square it belonged to. In the case of the sites represented outside the map, their first character was `x`, the second corresponded to their relative position as they are represented under the map, which more-or-less reflected their relative longitudinal position, but ignored the actual distances. The last two characters were digits, and they reflected the relative position of the site in the grid square. Every square was divided into nine equal numbered (5×5 km) squares: 5 was the central square, 1 is the northwestern and 9 is the southeastern corner. The third character reflected in which ninth the site lies in. In a similar way, every 5×5 km square was divided into nine squares, and the position of the site was specified further by the fourth character. This way, every site could be located with 1–2 km accuracy (see Table 1).

3 From exercise books to data

After the survey sites had been chosen, the material of the exercise books had to be digitized. Each exercise book was represented by one text file, containing exclusively ASCII characters. The data (and the metadata) were simply typed in by the author of the current article. The data were usually written in a well-readable hand, in addition, as the possible answers to the questions formed an almost closed set, it was usually relatively easy to find out what had been intended by the fieldworker. On the contrary, metadata were sometimes written in a hardly readable cursive, and it was difficult to

11	12	13	21	22	23	31	32	33
14	15	16	24	25	26	34	35	36
17	18	19	27	28	29	37	38	39
41	42	43	51	52	53	61	62	63
44	45	46	54	55	56	64	65	66
47	48	49	57	58	59	67	68	69
71	72	73	81	82	83	91	92	93
74	75	76	84	85	86	94	95	96
77	78	79	87	88	89	97	98	99

Table 1: The place of the survey site further specified by two numbers inside the territory specified by two letters

а	a	*е*	e	*и*	i	*о*	o	*у*	u
я	ja	*э*	\e	*ы*	y	*ё*	jo	*ю*	ju
б	b	*в*	v	*г*	g	*д*	d	*ж*	zh
з	z	*й*	j	*к*	k	*м*	m	*н*	n
п	p	*р*	r	*с*	s	*т*	t	*ф*	f
х	kh	*ц*	c	*ч*	ch	*ш*	sh	*щ*	sch
ь	'	*ъ*	"						

Table 2: The transcription applied in the text files for metadata and meaning given in Russian

find out what is intended to be written (especially with personal names).

The files began with the metadata: every line contained one piece of data, beginning with the data identifier (field name), followed by a colon and the data. The identifiers were abbreviations based on Udmurt phrases, e.g. `gunim`: the name of the village in Udmurt (***г**уртлэн* ***у**дмурт* ***ним**ыз*), `infvar`: the year of birth of the consultant (***и**информантлэн* ***вор**дскем* ***ар**ез*), `ljuk`: the collector (fieldworker) (***люк**ась*) etc. For the transcription for the metadata, see Table 2.

The linguistic data followed the metadata. Every line contained a three digit code of the question and the answer, separated by a space. The paradigm forms for *тодыны* 'to know' and *кутскыны* 'to begin' were numbered 400–424.

The Cyrillic-based transcription used in the exercise books was transliterated to a specific code inspired by the Prószéky code. The Prószéky (named after its inventor, Gábor Prószéky)[3] is an ASCII-based code developed originally for Old Hungarian texts. The basic idea is that every letter missing from the English alphabet is encoded with a combination of a letter and one or two digits, e.g. *á*: `a1`, *ö*: `o2`, *ő*: `o3`, *č*: `c12`, *æ*: `a36`, *ʃ*: `s43`, *ð*: `d50`, etc. In the transcription applied (see Table 3), a Roman letter or a Roman letter and a digit corresponds to the original Cyrillic letter. However, there are also some exceptions, e.g. some digits (`8`, `9`) correspond to letters themselves, some punctuation marks (`"`, `.`) are also applied (since the data are words or, rarely, phrases, these are not needed otherwise), and some other marks are also used (`%`, `'`).

If the lack of a form was indicated in the exercise book in any way, a mark hyphen (`-`) was typed into the place of the data. If the form occurred just in a given phrase

[3] The first description of the transcription can be found in the unpublished manuscript Prószéky (1985). The earliest use of the term *Prószéky code (Prószéky-kód)* is attested in Kornai (1985).

а	a	а°	a0	ä	a2	и	i	$\overset{ə}{и}$	i6
^{и}	i3	о	o	$\underset{\wedge}{о}$	o1	ȯ	o6	ӧ^{ъ}	o3
ӧ	o2	ӧ°	o4	ӧ̥	o5	у	u	ӱ	u2
^{у}	u3	у̇	u5	ы	y	ы°	y0	ӹ	y2
^{ы}	y3	ы̆	y7	ъ	9	^{ъ}	93	ь	8
э	e	$\underset{\wedge}{э}$	e1	э̬	e3	^{э}	e3	·	%
б	b	в	v	β	W	w	w	ў	u7
г	g	д	d	Д	D	д′	d1	д′^{ӟ}	d5
ж	zh	ӝ	xh	ӝ′	x4	з	z	з′	z1
з″	z"	ӟ	x	ӟ″	x"	й	j	^{й}	j7
к	k	$\overset{,}{к}$	k3	л	l	л′	l1	l	lh
м	m	^{м}	m7	н	n	н′	n1	η	q
н̣	n.	п	p	р	r	с	s	с′	s1
с″	s"	$^{с′}$	s6	т	t	т′	t1	т′^{ч}	t5
^{т}	t7	ф	f	х	X	ц	C	ц′	C1
ч	c	ч′	c1	^{ч}	c7	ӵ	ch	ӵ′	c4
^{ӵ}	c6	ш	sh	щ	s1s1	’	'		

Table 3: The transcription applied in the text files for Udmurt dialect data

(as /d͡ʑet͡ɕ/ ‘good’ in /d͡ʑet͡ɕ lu/ ‘good bye’), the phrase was presented after a backslash (\).

If a synonym was given instead of the expected form, the hyphen was followed by an equals sign (=) and then came the synonym. If the answer was missing (but the lack of the asked item was not indicated), a question mark (?) was written. Any evidently wrong data were written following a question mark as well.

If there were more variants given to the question, they were separated by a comma (,). If the meaning of the word was given in Russian in the exercise book, it was encoded following a hashmark (#) in the transcription similar to the one used for metadata. If the verbal paradigm forms contained a personal pronoun as well, they were written after the verb form separated by an at sign (@).

4 From data to atlas

The idea was to generate an atlas from the text files as automatically as possible. It is important, because this way a new version of the atlas can be done any time (after correcting mistakes, adding new files or even survey sites, changing the way of data representation, the structure of the atlas, etc.). Therefore, a modular process was designed, in which a Unix shell script managed the whole process (all the work was done in Linux), calling Perl scripts and using simple shell commands (such as `uniq` and `sort`).

In principle, the basic task was rearrangement. While the source text files contained the answers given at one survey site at one occasion, in the atlas, answers given to the different questions had to be represented on a different map each; on each map, data for each survey had to be presented, grouped due to the survey sites, even similar data for the same survey site must be grouped together (symbolized by the same sign). In addition, for every map, each type of data must be associated with a map sign (manually, at least for the first time), and for each map, a legend must be generated, which must enlist all the used signs and all the data they are associated with.

The result was a LaTeX source file, which had to be compiled by LaTeX, and the DVI file could be converted to PostScript or PDF, which was ready for printing. An example of an atlas map is presented in Figure 1.

It must be stressed that flexibility is an essential property of the whole approach to the atlas. This means that most of the things done in a particular way could have been done in a different way. However, the description of the decisions made can also demonstrate the possibilities.

The *processed material* was restricted to the first 396 questions of the questionnaire. These ask for an Udmurt equivalent of a Russian word or phrase, i.e. the answer is an Udmurt word (or phrase). Questions 397–400 ask for the meaning of a given Udmurt word, that is, the answer is a Russian word (or phrase). Therefore, a different code is needed to process these answers, the coding of which was delayed, and later, seeing no hope for publication, the needed script was never written. The maps for the paradigm forms of two verbs (*тодыны* ‘to know’, *кутскыны* ‘to begin’) were omitted for a different reason. While all other questions are targeted to explore a given dialectal phenomenon, in these cases, there is no explicit problem the data should answer to. The maps could have been done from several standpoints, but asking novel research questions was out of the scope of the project; therefore, these maps were not prepared. (Representing all variants on the map had no sense, see below.)

The atlas basically consisted of the maps and the legends belonging to them, there were no accompanying comments. Despite that, the atlas had a title and contained some texts; therefore, the *language* of it had to be chosen. It was decided that the atlas will be bilingual: Udmurt for the sake of the language community and English for the international public.

Since there were 396 maps derived and all of them had to have a separate legend (although theoretically the legend could have been placed on the map, for the sake of readability and for aesthetic reasons, this solution was rejected), the length of the whole atlas was about 800 pages. Moreover, the legend sometimes was much shorter than a page, sometimes it exceeded a page length. In addition, it had to be prepared for the addition of explanations to each map. Since every map begins a new page, and each map should be presented on the same side, it could take up very much place. Therefore, it was decided that the map will be presented in two volumes: the first contains the maps, the second one contains the legends (and, desirably, the explanations in the future).

As it was mentioned above, the *maps showed the territory* of Udmurtia, and the Periferic Southern Dialects (PSDs), spoken farther from Udmurtia, were represented under the maps. This solution was chosen because if the PSDs had been represented on the map, the territory of Udmurtia would have been overly compressed. Moreover, PSDs are relatively dispersed, and their representation on their exact place does not add much to our understanding of the dialectal distribution of the given phenomena. In addition, PSDs are poorly represented in our material. Nonetheless, it would have been possible to represent every survey site on their exact place. Similary, it would have been possible to ”magnify” any territory on the map and examine the isoglosses more closely (especially where more survey sites could be added).

The survey sites are not represented on an exact geographical map as a background. As orientation points, six significant settlements of Udmurtia (Izhevsk, Glazov, Votkinsk, Sarapul, Mozhga, Igra) and Agryz (which belongs to Tatarstan, but whose area protrudes into the territory of Udmurtia) are indicated. In addition, the north-

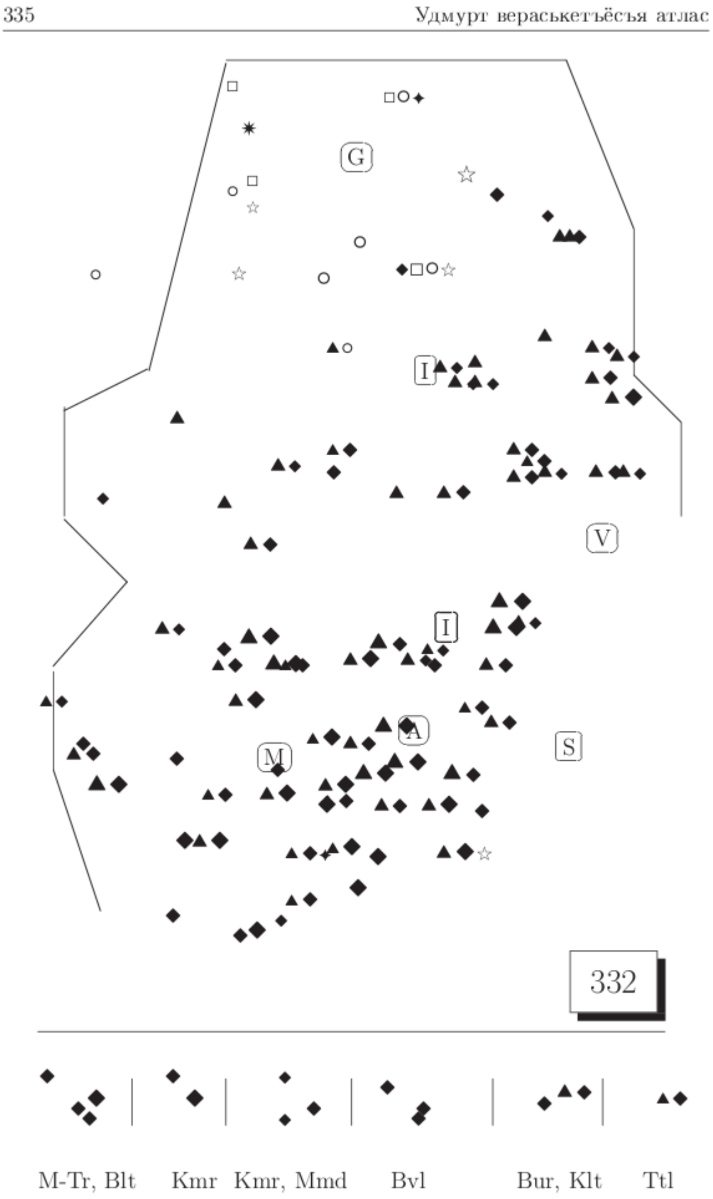


Figure 1: A map from the atlas (332. ‘по оврrагу’ – ‘along/through the ravine’, for the legend, see Figure 2)

eastern, northern and western borders of the Udmurt Republic is also schematically represented.

Many dialect atlases tend to show just one form (meaning, etc.) for one survey site. However, our everyday experience shows that dialects and even individual speakers exhibit variability. Variability can be attested in the material of the atlas as well. In many cases, data collected by different fieldworkers and from different consultants differ; the exercise books sometimes contain more possible answers for the question. It was decided that the atlas should reflect the *local diversity* of dialects; therefore, all data must be represented. However, since the dialect of some sites are documented in more than a dozen exercise books, it makes no sense to put all data on the map. On the other hand, if a form is documented seven times at a site, and another only once, it would be misleading to represent them in the same way. Therefore, if a piece of data on a site occurred just once, it was smaller, if more than four times, bigger than the sign for two to four pieces of data.

However, only *relevant diversity* was reflected on each map. That is, if the question of the questionnaire asked for the quality of a consonant, the differences of vowels were not reflected by the signs. On the one hand, this is practical for the sake of readability; on the other hand, it helps to eliminate the errors of the fieldworkers similar to the one mentioned in Section 2.2.

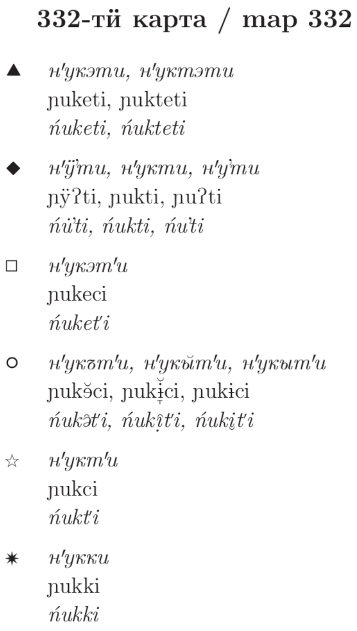
For the sake of keeping the printing cost low, *no colours* were used in the atlas. The signs chosen to represent the data where taken from the `MnSymbol` package of LaTeX: triangles (turned into different directions, filled and unfilled), squares, diamonds, circles (all filled and unfilled, containing different patterns), stars (asterisks, different forms and number of points). The signs were chosen in a way that their similarities could reflect the similarities of the linguistic data (e.g., data represented by filled triangles and filled circles resemble one another in a way; while data represented by filled triangles and unfilled triangles are similar in another way.).

The linguistic data are presented in three *transcriptions*: in Cyrillic based Udmurt dialect transcription – see Кельмаков (1998, 44–50) or Кельмаков (2002, 49–56) – for the language community, IPA for the international audience, and Finno-Ugric transcription for western traditional Finno-Ugrists. An example is represented on Figure 2.

5 The differences between the two dialect atlases of Udmurt

An important difference is that while ДАУЯ aims to present a full and balanced picture of the Udmurt dialects, the purpose of УВА is to make use of an incomplete and unbalanced, but already existing collection. Moreover, this collection is constantly growing, and hopefully will grow until Udmurt is spoken or Udmurt philology is taught at the Udmurt University.

By digitizing new exercise books, new survey sites can be added, and the maps can be completed with data for the territories undocumented up to this point relatively easily. On well-documented areas, the survey sites can be made more dense, and more detailed maps of these territories can be produced by relatively small modifications of the scripts. The data for УВА have been collected during a long period (at the time of the preparation, about twenty years, but since then more than thirty years), that is different survey sites can be represented by data from different times. However, on



332-тй карта / map 332

▲ *н′укэти, н′уктэти*
ɲuketi, ɲukteti
ńuketi, ńukteti

◆ *н′ӱ̆’ти, н′укти, н′у’ти*
ɲÿ̆ʔti, ɲukti, ɲuʔti
ńü’ti, ńukti, ńu’ti

□ *н′укэт′и*
ɲukeci
ńuket́i

○ *н′укъ̆т′и, н′укы̆т′и, н′укыт′и*
ɲukə̆ci, ɲukɨ̞̆ci, ɲukɨci
ńukət́i, ńukı̣̂t́i, ńukı̯t́i

☆ *н′укт′и*
ɲukci
ńukt́i

✷ *н′укки*
ɲukki
ńukki

Figure 2: An example of the legend (332. ‘по оврrагу’ – ‘along/through the ravine’, for the map, see Figure 1

well-documented territories, it might be possible to do longitudinal analysis and to reveal linguistic change.

Unfortunately, some territories, especially outside Udmurtia and Northern Udmurtia, are poorly represented by УВА. Although the number of the blank spots can be decreased, especially by organizing ”expeditions” to these territories, it is a cost-sensitive and time-consuming issue. Moreover, even if special attention is paid to the less-documented areas, the documentation level of different territories will never be balanced. On the contrary, there are no similar problems with ДАУЯ.

From an aesthetic point of view, УВА falls short of ДАУЯ, and even the exact identification of the survey sites on the map is challenging. However, the main purpose is not documenting the survey sites, but to give a general impression on the distribution of certain forms.

The most important difference is that while УВА presents the distribution of phonological and morphological features, and only minimally considers lexical differences, ДАУЯ deals exclusively with lexical issues. As a consequence, the two atlases complement each other, and together they can provide a more complete picture of the Udmurt dialects.

Finally, УВА was prepared based on an existing material and using low-budget tools in a do-it-yourself way. Evidently, the circumstances have changed since the atlas was made, and many things should be done in a very different way today (and even could have been done better at that time). But as this case study shows, making an atlas is not an unachievable purpose even for individual researchers if the linguistic material is available.

6 The future

The simplest way to finish the project would be to find financial support for a printout version, generate the missing maps, possibly improve the appearance, and publish the atlas. However, knowing the circumstances, this scenario seems to be unrealistic.

If we think about online publication, publication in a PDF format is not expedient. It would be much more reasonable to take advantage of the opportunities offered by technology, and to publish maps in an interactive format (e.g. based on OpenStreetMaps), when the user can zoom in and out depending whether they are interested in a specific territory or the general view. However, this would need a completely new way of generating maps from the data, although based on the same principles.

Nonetheless, such a decision is quite risky because of the fast change in technology. For example, in 2006 it could seem a good idea to publish an atlas on CD ROM, which, depending on the technological details, could be completely unusable today. While the preservation of printout books has established standards, web sites easily perish and vanish from the Internet, especially when nobody is involved in them in the hosting institute, if not as an author, at least as a user. As a consequence, online-only publication is not always a completely responsible decision even today.

Acknowledgements

The development of the atlas was financed by the project *NKFI K48309 Permic linguistic databases* led by Sándor Csúcs. I thank the collegues at the Department of General and Finno-Ugric Linguistics of the Udmurt State University, especially Valentin Kelmakovich Kelmakov, for their support. I am grateful to my Izhevsk family of that time for the ideal circumstances I could work in. I am also thankful to Nóra Wenszky for her assistance in preparing this paper and her useful advice.

References

András Kornai. 1985. Szótári adatbázis az akadémiai nagyszámítógépen. In *Tanulmányok a nyelvtudomány és társtudományai köréből*, MTA Nyelvtudományi Intézet, Budapest, pages 65–79.

Gábor Prószéky. 1985. Automatizált morfológiai elemzés a nagyszótári munkálatokban. Kézirat, MTA Nyelvtudományi Intézet.

В. К. Кельмаков. 1998. *Краткий курс удмуртской диалектологии.* Издательство Удмнуртского Университета, Ижевск.

В. К. Кельмаков. 2002. *Удмурт дуиалектология. Студентъёслы но аспирантъёслы юрттос.* 5-тйез, тупатъяса, выльдыса но ватсаса поттэмез. «Удмурт университет» книга поттон корка, Ижкар.

Р. Ф. Насибуллин, С. А. Максимов, В. Г. Семёнов, and Г. В. Отстановна. 2009. *Диалектологический атлас удмуртского языка. Вып. I.* Научно-издательский центр «Регулярная и хаотическая динамика», Ижевск.

Digitizing print dictionaries using TEI: The Abaev Dictionary Project

Oleg Belyaev
Lomonosov Moscow State University
Institute of Linguistics RAS
belyaev@ossetic-studies.org

Irina Khomchenkova
Vinogradov Russian Language Institute RAS
Lomonosov Moscow State University
irina.khomchenkova@yandex.ru

Julia Sinitsyna
Lomonosov Moscow State University
jv.sinitsyna@yandex.ru

Vadim Dyachkov
Institute of Linguistics RAS
hyppocentaurus@mail.ru

Abstract

We present the results of a year-long effort to create an electronic version of V. I. Abaev's Historical-etymological dictionary of Ossetic. The aim of the project is two-fold: first, to create an English translation of the dictionary; second, to provide it (in both its Russian and English version) with a semantic markup that would make it searchable across multiple types of data and accessible for machine-based processing. Volume 1, whose prelimiary version was completed in 2020, used the TshwaneLex (TLex) platform, which is perfectly adequate for dictionaries with a low to medium level of complexity, and which allows for almost WYSIWYG formatting and simple export into a publishable format. However, due to a number of limitations of TLex, it was necessary to transition to a more flexible and more powerful format. We settled on the Text Encoding Initiative — an XML-based format for the computational representation of published texts, used in a number of digital humanities projects. Using TEI also allowed the project to transition from the proprietary, closed system of TLex to the full range of tools available for XML and related technologies. We discuss the challenges that are faced by such large-scale dictionary projects, and the practices that we have adopted in order to avoid common pitfalls.

1 Introduction

Digital lexicography is currently experiencing rapid development. With the transition to computerized publishing, most dictionaries are from the start conceived of as structured databases, with the print version being only one medium of many — and not a primary one at that. This, in most cases, presupposes a structure of lexical entries that is considerably different from that of earlier print dictionaries, where automatic processing was not an issue and the data were structured so as to be accessible in printed form. Major continuing publications (such as, for example, the Oxford English Dictionary (OED, 2021)) have already made the transition to digital formats. However, this is mainly true for large languages, where dictionaries are regularly published by stable teams having reliable financial support from state research institutions or private companies. For smaller languages, especially for minority languages, many dictionaries still only remain available in print (at best, scanned) form, with no possibility of automatic digitization due to the complexity of their structure and the inherent irregularity of their practical decisions (entry structure, choice of typefaces, etc.). Even when new dictionaries are published by local research teams, they are often prepared for typesetting as monolithic word-processor documents, making them largely equivalent to traditional print dictionaries prepared from card-catalogues — searchable by text, but without any semantic markup or more complex query mechanisms. This situation severely biases the range of lexicographical data available to researchers working on individual languages and in lexical typology — even when the dictionaries exist and are of a considerably high quality, they are virtually unavailable for automatic query and analysis.

This paper describes an attempt to fill this gap for Ossetic — an Iranian language spoken in the Caucasus by approximately 500 000 people. Ossetic is relatively well-documented lexicographically: bilingual (Abaev, 1970; Kasaev and Guriev, 1993; Takazov, 2003) and monolingual (Gæbæraty et al., 1999)

Seventh international workshop on computational linguistics of Uralic languages, pages 12–19
September 23–24, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-1-954085-82-4_002

dictionaries exist for both major dialects (Iron and Digor), and due to the effort of Ossetic language enthusiasts these have been converted into the ABBYY Lingvo format and an online searchable database (Iriston.com, 2004), which, while not ideal for research purposes and having some limitations, may at least be queried by headword.

However, the main lexicographic resource for Ossetic is still Vasily Ivanovič Abaev's fundamental, four-volume *Historical-etymological dictionary of Ossetic* (Abaev, 1958–1989) (henceforth AbD). This dictionary is not only one of the best etymological dictionaries available for any Iranian language (Zgusta, 1991), but also a very detailed descriptive, bilingual (Ossetic-Russian) dictionary — with the quality of definitions and the number of illustrative examples far surpassing that of all other Ossetic dictionaries. This dictionary still lacks a digital version, for obvious reasons: the structure of entries is complex and not trivial to capture in a standard dictionary format; the etymologies include examples from many different languages with diverse scripts that cannot be reliably OCR'd; manual verification should be undertaken. A further problem is that AbD is not available in English, making it unaccessible to scholars who do not fluently read Russian. By both digitizing and translating AbD, one would automatically provide a solid basis to further digital lexicographic work on Ossetic while also providing scholars with an English-based lexicographic resource for this language.

Therefore, with the encouragement of the Moscow Ossetian Fraternity (whose help and support we gratefully acknowledge), in the end of 2020 we began preliminary work on the project of both translating and digitizing AbD (including the Russian version, which should in any case be available as a benchmark against which uncertain parts of the translation can be verified). By the end of 2020, a first draft of the translation and database was prepared, published in a small number of copies in book form (Èto Kavkaz, 2020). This paper documents our experience with this project while highlighting the advantages and drawbacks of different approaches, and attempting to establish a best practice that could hopefully be used This was preceded by a preliminary analysis of the structure of Abaev's lexical entries, described in section 2. Section 3 describes the choice of TshwaneLex (TLex) (Joffe et al., 2021) as the software platform and the general structure as it was implemented by the end of 2020, and the disadvantages of TLex for a dictionary with a structure like AbD's. Finally, in section 4 we describe the transition to the Text Encoding Initiative (TEI) (The TEI Consortium, 2021) framework and the corresponding workflow, which solves most of the problems that we had with TLex and can serve as a useful foundation for further work on similar lexicographic projects — in particular for Uralic languages, given that a large number of similar legacy dictionaries are available for many of these languages, and Ossetic itself (unlike most other Iranian languages) is typologically similar to Uralic.

2 The structure of a dictionary entry in AbD

The overall structure of a mid-sized AbD lexical entry (that includes all the core elements but lacks additional complexities) can be illustrated by the lexeme *ad* 'taste', shown in Figure 1.

The entry can be subdivided into several clearly distinguished elements:

1. The **headword**, with a possible dialectal Digor form (separated from the main word by a vertical line).

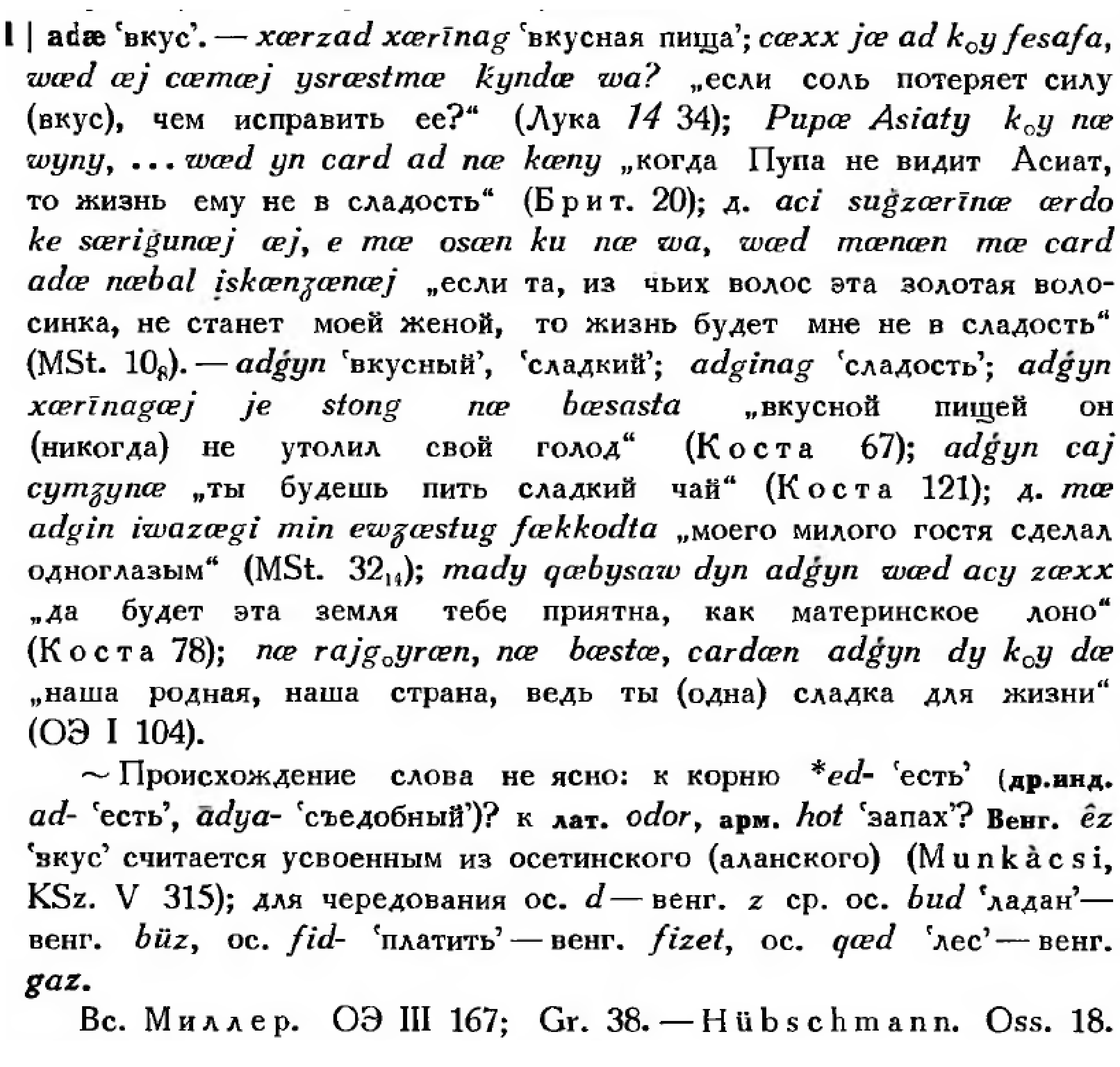
2. One or more **senses**, which consist of, most frequently, of short glosses in quotation marks, with possible additional comments.

3. An optional set of one or more **subentries** (idiomatic expressions or derivates from the headword), separated from each other by commas or semicolons; each subentry is a "mini-entry" in its own right, which may include several senses and its own examples.

4. One or more **groups of examples**; the group itself is separated by the surrounding content (including other example groups) by a dash, and examples are separated from each other by semicolons. The logic that stands behind using several groups of examples, rather than putting all exampels in one group, is in the general case not discernable. Sometimes both senses and example groups are numbered, in which case the group correspond to senses with the same numerical index.

5. An optional additional set of **subentries.**

6. A possible additional **example group** following the second set of subentries; only occurs



I | **adæ** ‘вкус’. — *xærzad xærīnag* ‘вкусная пища’; *cæxx jæ ad k$_o$y fesafa, wæd æj cæmæj ysræstmæ kyndæ wa?* „если соль потеряет силу (вкус), чем исправить ее?“ (Лука *14* 34); *Pupæ Asiaty k$_o$y næ wyny, . . . wæd yn card ad næ kæny* „когда Пупа не видит Асиат, то жизнь ему не в сладость“ (Брит. 20); д. *aci suǵzærīnæ ærdo ke særiǵunæj æj, e mæ osæn ku næ wa, wæd mænæn mæ card adæ næbal ịskænʒænæj* „если та, из чьих волос эта золотая волосинка, не станет моей женой, то жизнь будет мне не в сладость“ (MSt. 10_8). — *adǵyn* ‘вкусный’, ‘сладкий’; *adginag* ‘сладость’; *adǵyn xærīnagæj je stong næ bæsasta* „вкусной пищей он (никогда) не утолил свой голод“ (Коста 67); *adǵyn caj cymʒynæ* „ты будешь пить сладкий чай“ (Коста 121); д. *mæ adgin iwazægi min ewʒæstug fækkodta* „моего милого гостя сделал одноглазым“ (MSt. 32_{14}); *mady qæbysaw dyn adǵyn wæd acy zæxx* „да будет эта земля тебе приятна, как материнское лоно“ (Коста 78); *næ rajg$_o$yræn, næ bæstæ, cardæn adǵyn dy k$_o$y dæ* „наша родная, наша страна, ведь ты (одна) сладка для жизни“ (ОЭ I 104).

~ Происхождение слова не ясно: к корню **ed*- ‘есть’ (**др.инд.** *ad*- ‘есть’, *ādya*- ‘съедобный’)? к **лат.** *odor*, **арм.** *hot* ‘запах’? **Венг.** *êz* ‘вкус’ считается усвоенным из осетинского (аланского) (Munkàcsi, KSz. V 315); для чередования ос. *d* — венг. *z* ср. ос. *bud* ‘ладан’— венг. *büz*, ос. *fid*- ‘платить’ — венг. *fizet*, ос. *qæd* ‘лес’ — венг. ***gaz***.

Вс. Миллер. ОЭ III 167; Gr. 38. — Hübschmann. Oss. 18.

Figure 1: AbD entry for *ad* ‘taste’.

when the first block of example group(s) is also present.

7. The etymology, preceded by the tilde sign, which is essentially rich text which includes citations of forms from Ossetic and other languages (with the abbreviated language name typeset in bold) and bibliographic references.

This overall structure is of course a simplification: deviations from it are found in the dictionary, which is rather natural considering that the lexical entries were compiled by hand. However, in general, apart from the etymology, it is clear that the structure is relatively rigid so that it can be captured by a dictionary platform that allows custom data structures.

3 The TLex implementation

There are many lexicographic tools for linguists available today; the most popular ones are SIL Toolbox (SIL International, 2010) and Lexique Pro (SIL International, 2009), based on the Standard Format (SFM); and a more complex system implemented in SIL FieldWorks Language Explorer, or FLEx (SIL International, 2021). All these tools, while powerful and user-friendly, are aimed at field linguists documenting previously undescribed languages, and are ill-suited for a dictionary with such a non-standard structure as AbD. The standard tool for etymological dictionaries, StarLing (Starostin and Starostin, 2003), while powerful, is not suited for our purposes: it is rather deterministic, with the main aim being to capture exact etymological relationships, while AbD is in many cases ambiguous as to the exact etymology. Making an exact choice for an etymon is already an analytic decision that is beyond the scope of a digitization / translation project. Furthermore, StarLing provides little in terms of semantic markup. Therefore, we decided to choose another tool, also popular among digital lexicographers: TshwaneLex, or TLex (Joffe et al., 2021). This platform has been successfully used for numerous dictionary projects, notably the Beserman Udmurt dictionary, which has a complex structure comparable to that of AbD (Serdobolskaya et al., 2021). It is essentially a frontend to a highly customizable XML data structure. In particular, it is possible to define not only additional fields (as in FLEx), but a system of nested elements; importantly, the elements may contain mixed content with tags and PCDATA — this is essential for markup in the etymology to work correctly, as, of course, no rigid structure can account for the free-form text in Abaev's etymological descriptions. Accordingly, TLEx was used to implement a general dictionary structure that mimics the structure of Abaev's entries:

LemmaSign as an attribute (according to TLex usage), with optional LemmaVariant (for comma-separated orthographic / phonetic variants), Participle (verbs are quoted with participle forms, which are generally irregular) and DigorForm (for Digor dialectal forms, if they differ from Iron).

PreSubentryGroup (0+) a group of one or more subentries that precedes the first group of examples.

ExampleGroup (0+) the first set of example blocks;

PostSubentryGroup (0+) the second block of subentries;

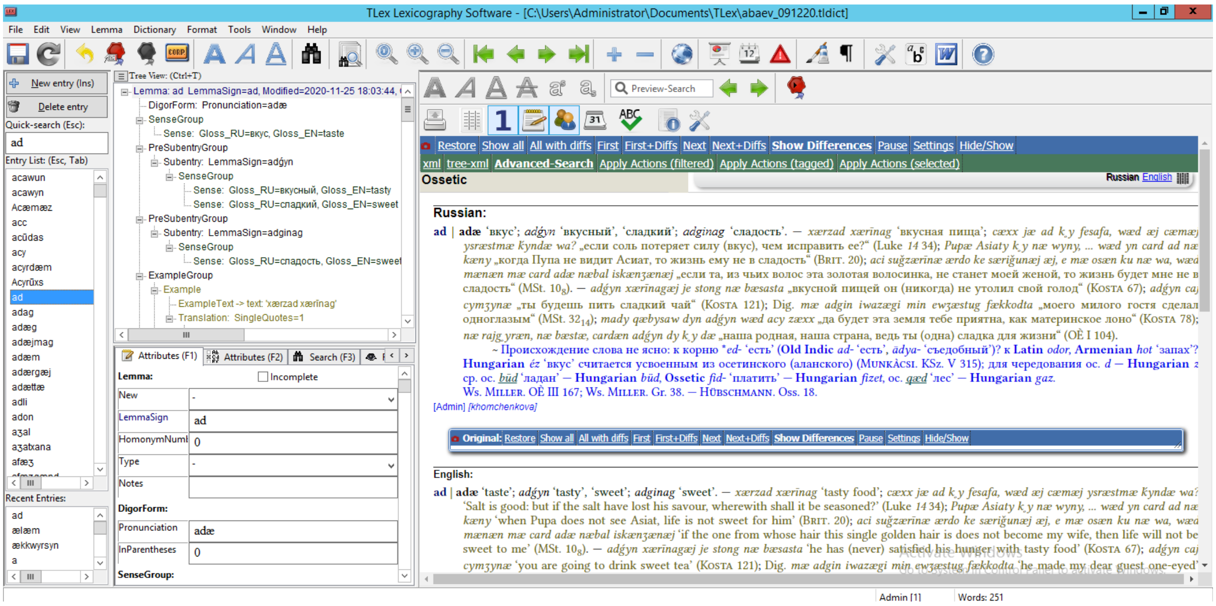
ExampleGroup (0+) the second set of example blocks;

Etymology with mixed content.

The structure of the same lexical entry *ad* 'taste' in TLex is shown in Figure 2.

The structure approximates AbD's structure relatively well and was used to successfully finish the translation of Vol. 1 of the dictionary. However, even from this general description of the structure, some problems are immediately apparent. For example, the difference between PreSubentryGroup and PostSubentryGroup seems to be completely artificial: these elements have exactly the same internal structure and display style. Following the logic of XML markup, they should definitely be assigned to the same element type.

The reason for this representation is the way TLex handles the order of elements: Unlike plain XML, where document order is always relevant, TLex ignores the order of elements in the XML source, only the structure is taken into account. This provides the advantage of being able to freely reorder elements using the built-in styling system. But the disadvantage is that it is practically impossible to differentiate between two or more elements that stand in different positions.


TLex Lexicography Software - [C:\Users\Administrator\Documents\TLex\abaev_091220.tldict]
File Edit View Lemma Dictionary Format Tools Window Help
New entry (Ins)
Delete entry
Quick-search (Esc):
Entry List: (Esc, Tab)
Recent Entries:
Tree View: (Ctrl+T)
Attributes (F1)
Attributes (F2)
Search (F3)
Preview-Search
Restore | Show all | All with diffs | First | First+Diffs | Next | Next+Diffs | Show Differences | Pause | Settings | Hide/Show
xml | tree-xml | Advanced-Search | Apply Actions (filtered) | Apply Actions (tagged) | Apply Actions (selected)
Original: Restore | Show all | All with diffs | First | First+Diffs | Next | Next+Diffs | Show Differences | Pause | Settings | Hide/Show
Russian English
Admin [1]
Words: 251

Quick-search (Esc): ad

acawun
acawyn
Acæmæz
acc
acūdas
acy
acyrdæm
Acyrūxs
ad
adag
adæg
adæjmag
adæm
adærgæj
adættæ
adli
adon
aʒal
aʒatxana
afæʒ

Recent Entries:
ad
ælæm
ækkwyrsyn
a

```
Lemma: ad  LemmaSign=ad, Modified=2020-11-25 18:03:44,
  DigorForm:  Pronunciation=adæ
  SenseGroup
    Sense:  Gloss_RU=вкус, Gloss_EN=taste
  PreSubentryGroup
    Subentry:  LemmaSign=adǵyn
      SenseGroup
        Sense:  Gloss_RU=вкусный, Gloss_EN=tasty
        Sense:  Gloss_RU=сладкий, Gloss_EN=sweet
  PreSubentryGroup
    Subentry:  LemmaSign=adginag
      SenseGroup
        Sense:  Gloss_RU=сладость, Gloss_EN=sweet
  ExampleGroup
    Example
      ExampleText -> text: 'xærzad xærīnag'
      Translation:  SingleQuotes=1
```

Lemma: ☐ Incomplete

New	-
LemmaSign	ad
HomonymNuml	0
Type	-
Notes	

DigorForm:

Pronunciation	adæ
InParentheses	0

SenseGroup:

Ossetic

Russian:

ad | **adæ** ‘вкус’; *adǵyn* ‘вкусный’, ‘сладкий’; *adginag* ‘сладость’. — *xærzad xærīnag* ‘вкусная пища’; *cæxx jæ ad k̦y fesafa, wæd æj cæmæj*
ysræstmæ ḱyndæ wa? „если соль потеряет силу (вкус), чем исправить ее?“ (Luke *14* 34); *Pupæ Asiaty k̦y næ wyny, ... wæd yn card ad næ*
kæny „когда Пупа не видит Асиат, то жизнь ему не в сладость“ (Brit. 20); *aci suğzærīnæ ærdo ke særiğunæj æj, e mæ osæn ku næ wa, wæd*
mænæn mæ card adæ næbal iskænʒænæj „если та, из чьих волос эта золотая волосинка, не станет моей женой, то жизнь будет мне не в
сладость“ (MSt. 10_8). — *adǵyn xærīnagæj je stong næ bæsasta* „вкусной пищей он (никогда) не утолил свой голод“ (Kosta 67); *adǵyn caj*
cymʒynæ „ты будешь пить сладкий чай“ (Kosta 121); Dig. *mæ adgin iwazægi min ewʒæstug fækkodta* „моего милого гостя сделал
одноглазым“ (MSt. 32_{14}); *mady qæbysaw dyn adǵyn wæd acy zæxx* „да будет эта земля тебе приятна, как материнское лоно“ (Kosta 78);
næ rajg̦yræn, næ bæstæ, cardæn adǵyn dy k̦y dæ „наша родная, наша страна, ведь ты (одна) сладка для жизни“ (OÈ I 104).
~ Происхождение слова не ясно: к корню **ed-* ‘есть’ (**Old Indic** *ad-* ‘есть’, *ādya-* ‘съедобный’)? к **Latin** *odor*, **Armenian** *hot* ‘запах’?
Hungarian *éz* ‘вкус’ считается усвоенным из осетинского (аланского) (Munkàcsi. KSz. V 315); для чередования ос. *d* — **Hungarian** *z*
ср. ос. *būd* ‘ладан’ — **Hungarian** *büd*, **Ossetic** *fid-* ‘платить’ — **Hungarian** *fizet*, ос. *qæd* ‘лес’ — **Hungarian** *gaz*.
Ws. Miller. OÈ III 167; Ws. Miller. Gr. 38. — Hübschmann. Oss. 18.
[Admin] *[khomchenkova]*

English:

ad | **adæ** ‘taste’; *adǵyn* ‘tasty’, ‘sweet’; *adginag* ‘sweet’. — *xærzad xærīnag* ‘tasty food’; *cæxx jæ ad k̦y fesafa, wæd æj cæmæj ysræstmæ ḱyndæ wa?*
‘Salt is good: but if the salt have lost his savour, wherewith shall it be seasoned?’ (Luke *14* 34); *Pupæ Asiaty k̦y næ wyny, ... wæd yn card ad næ*
kæny ‘when Pupa does not see Asiat, life is not sweet for him’ (Brit. 20); *aci suğzærīnæ ærdo ke særiğunæj æj, e mæ osæn ku næ wa, wæd*
mænæn mæ card adæ næbal iskænʒænæj ‘if the one from whose hair this single golden hair is does not become my wife, then life will not be
sweet to me’ (MSt. 10_8). — *adǵyn xærīnagæj je stong næ bæsasta* ‘he has (never) satisfied his hunger with tasty food’ (Kosta 67); *adǵyn caj*
cymʒynæ ‘you are going to drink sweet tea’ (Kosta 121); Dig. *mæ adgin iwazægi min ewʒæstug fækkodta* ‘he made my dear guest one-eyed’

Figure 2: The TLex representation of *ad* ‘taste’

This leads to another artificial solution: the splitting of <Comment> elements into <PreComment> (before parent element) and <PostComment> (after parent element). But even this sometimes leads to absurd situation. For example, in our model, example texts and translations were originally represented by the attributes @text and @tr(_ru,en). A PreComment would then precede the example and a PostComment would follow the translation. Some examples, however, have a comment that stands between the example and the translation:

```
<PreComment> @text <MidComment>??
@tr <PostComment>
```

Clearly, a proliferation of <MidComment>-like elements is undesirable, because the range of possible comment positions can never be fully accounted for. The eventual solution was to repesent the example text and translation by elements (<ExampleText>, <Translation>), not attributes — which is, in fact, the natural way for XML, but not for TLex, which heavily favours attribute values and where adding new nested elements is a cumbersome process that is prone to error.

Another problem is the handling of styles. Surprisingly for an XML-based system (where CSS is normally available), TLex has a rather simplistic style system that cannot account for the element or attribute's context in any way. As seen in the above example, AbD uses punctuation patterns that are by themselves rather regular, but difficult for human annotators to consistently handle without error. It is therefore desireable to insert such regular punctuation automatically. In TLex, this can be done only by scripts written in an internal lua-based language. For example, the following code inserts a space before a <Source> (reference to an example source) that follows a <PostComment> element.

```
local prev = gCurrentNode:GetPrevious();
if prev ~= nil then
    if prev:GetElementTypeID() == 10079
    then
        gCurrentStyle:SetBeforeG(" ");
    end
end
```

The same functionality is easily captured in CSS by a single line:

```
  PostComment + Source::before {
content: ''  ''}
```

The scripts are unnecessarily complex, written in a poorly documented language, and difficult to maintain; they may be adequate for comparatively minor dynamic styling, but as the project proceeded, it became clear that a large number of them is required. This made the dynamic punctuation practically unmanageable and difficult to debug.

A definite advantage of the TLex approach is support for controlled vocabularies, which here are called attribute lists (i.e. lists of possible values for certain attributes). However, this is not without a caveat: when server-based collaborative editing is used, any change to these attribute lists requires locking the whole database while making sure that all users have saved their data and logged out. This means that such trivial changes require an incomparable amount of effort, which complicates and slows down work on the dictionary.

To be sure the chief problem with using TLex for the AbD project is not that this is a bad piece of software — in fact, it is one of the best, if not the best, "off-the-shelf" dictionary creation tools currently available on the market. However, TLex's use of XML is more suitable for relatively flat database structures where most of the information is stored in attributes. The use of mixed data and nested tags is complex and is not something TLex has been designed for. It is an adequate tool for new dictionary projects that follow a more modern, sense-based structure, or for digitization projects that also overhaul the structure of the original. When the aim is to represent the original as faithfully as possible, TLex is not the right tool for the job.

4 The TEI approach

When Volume 1 was finished, work began on converting the dictionary format to Text Encoding Initiative (TEI) Guidelines [REF], which define a set of tags and constraints for representation of texts in digital form. An immediate advantage of TEI compared to TLex is that, unless new tags are defined (which is seldom needed, because TEI is a very detailed standard) or existing tags abused, each element has a well-described semantics that is immediately accessible to any external observer, due to the structure being associated with the TEI namespace. TEI also represents displayed content primarily in elements rather than attributes (consistent with XML practice, which is, after all, a *markup* language) and is fully compatible with mixed-content elements. Thus, the example above is represented in TEI as follows:

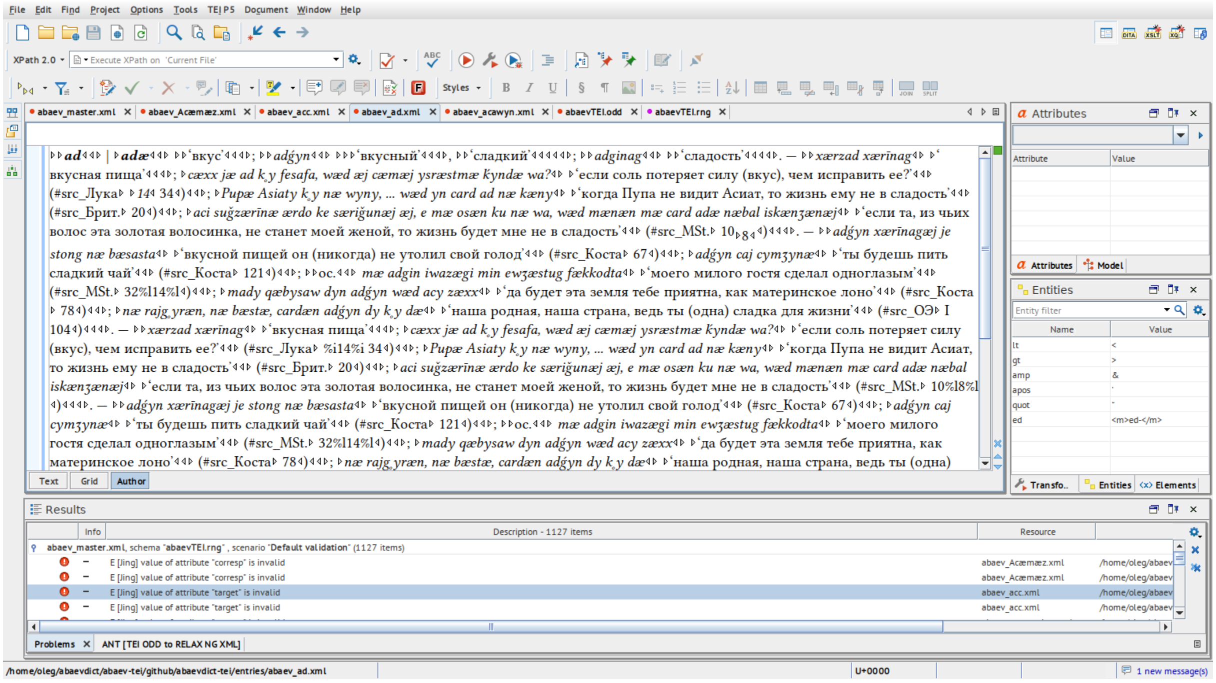
```
<cit type="example">
    <note type="comment">
        …(precomment)…
    </note>
    <quote>
        …(ex. text)…
    </quote>
    <note type="comment">
        …("midcomment")…
    </note>
    <cit type="translation"
        xml:lang=''ru''>
        …(translation)…
    </cit>
    <note type="comment">
        …("postcomment")…
    </note>
</cit>
```

Note the use of standard IETF BCP 47 (Network Working Group, 2009) language tags — this also allows interoperability and is implemented not only for English and Russian, but also for the Ossetic dialects and all languages cited in etymologies. The specific language strings can then be generate "on-the-fly" when the dictionary is converted (via XSLT or a similar transformation) into a publishable document.

An important feature of TEI is that it can be customize so that only the subset of all tags and attributes is selected that is actually required for a given project. This is done via files of a format called ODD (One Document Does (All)); our TEI customization is freely available in a GitHub repository: https://github.com/abaevdict/tei-abaev. This customization, of course, still remains rather redundant, allowing more than actually occurs; it could be constrained to resemble something like the rigid TLex schema above, but this is not required and in fact harmful, because further entries may include additional elements that have not been envisaged from prior experience (this being, after all, a legacy print dictionary).

The complex nested structures illustrated above can be edited in a user-friendly manner in modern XML editors such as Oxygen [REF], which we chose for this project. The editor natively supports TEI and allows "Author Mode" editing, which, styled with appropriate CSS, becomes almost a WYSIWYG model (see Figure 3). This significantly simplified work for the annotators, compared to TLex, where results are displayed in real-time, but the attributes and text values themselves have to be edited in a separate part of the screen.

The Oxygen customization, especially its CSS styles, are available on GitHub: https://github.com/abaevdict/abaev-tei-oxygen. The dictionary itself is split into multiple files, one file for each entry (generated from TLex using an XSLT transformation); the files are included in a single master file via XInclude. All dictionary data is also in a GitHub repo: https://github.com/abaevdict/abaevdict-tei. Collaborative editing can be done via standard



ad | **adæ** ‘вкус’; *adǵyn* ‘вкусный’, ‘сладкий’; *adginag* ‘сладость’. — *xærzad xærīnag* ‘вкусная пища’; *cæxx jæ ad k̥y fesafa, wæd æj cæmæj ysræstmæ k̓yndæ wa?* ‘если соль потеряет силу (вкус), чем исправить ее?’ (#src_Лука *14* 34); *Pupæ Asiaty k̥y næ wyny, ... wæd yn card ad næ kæny* ‘когда Пупа не видит Асиат, то жизнь ему не в сладость’ (#src_Брит. 20); *aci suǧzærīnæ ærdo ke særiǧunæj æj, e mæ osæn ku næ wa, wæd mænæn mæ card adæ næbal iskænʒænæj* ‘если та, из чьих волос эта золотая волосинка, не станет моей женой, то жизнь будет мне не в сладость’ (#src_MSt. 10_{8}). — *adǵyn xærīnagæj je stong næ bæsasta* ‘вкусной пищей он (никогда) не утолил свой голод’ (#src_Коста 67); *adǵyn caj cymʒynæ* ‘ты будешь пить сладкий чай’ (#src_Коста 121); ос. *mæ adgin iwazægi min ewʒæstug fækkodta* ‘моего милого гостя сделал одноглазым’ (#src_MSt. 32%l14%l); *mady qæbysaw dyn adǵyn wæd acy zæxx* ‘да будет эта земля тебе приятна, как материнское лоно’ (#src_Коста 78); *næ rajg̥yræn, næ bæstæ, cardæn adǵyn dy k̥y dæ* ‘наша родная, наша страна, ведь ты (одна) сладка для жизни’ (#src_ОЭ I 104). — *xærzad xærīnag* ‘вкусная пища’; *cæxx jæ ad k̥y fesafa, wæd æj cæmæj ysræstmæ k̓yndæ wa?* ‘если соль потеряет силу (вкус), чем исправить ее?’ (#src_Лука %i14%i 34); *Pupæ Asiaty k̥y næ wyny, ... wæd yn card ad næ kæny* ‘когда Пупа не видит Асиат, то жизнь ему не в сладость’ (#src_Брит. 20); *aci suǧzærīnæ ærdo ke særiǧunæj æj, e mæ osæn ku næ wa, wæd mænæn mæ card adæ næbal iskænʒænæj* ‘если та, из чьих волос эта золотая волосинка, не станет моей женой, то жизнь будет мне не в сладость’ (#src_MSt. 10%l8%l). — *adǵyn xærīnagæj je stong næ bæsasta* ‘вкусной пищей он (никогда) не утолил свой голод’ (#src_Коста 67); *adǵyn caj cymʒynæ* ‘ты будешь пить сладкий чай’ (#src_Коста 121); ос. *mæ adgin iwazægi min ewʒæstug fækkodta* ‘моего милого гостя сделал одноглазым’ (#src_MSt. 32%l14%l); *mady qæbysaw dyn adǵyn wæd acy zæxx* ‘да будет эта земля тебе приятна, как материнское лоно’ (#src_Коста 78); *næ rajg̥yræn, næ bæstæ, cardæn adǵyn dy k̥y dæ* ‘наша родная, наша страна, ведь ты (одна)

Name	Value
lt	<
gt	>
amp	&
apos	'
quot	"
ed	<m>ed-</m>

Info	Description - 1127 items	Resource	
	abaev_master.xml, schema "abaevTEI.rng" , scenario "Default validation" (1127 items)		
–	E [Jing] value of attribute "corresp" is invalid	abaev_Acæmæz.xml	/home/oleg/abaev
–	E [Jing] value of attribute "corresp" is invalid	abaev_Acæmæz.xml	/home/oleg/abaev
–	E [Jing] value of attribute "target" is invalid	abaev_acc.xml	/home/oleg/abaev
–	E [Jing] value of attribute "target" is invalid	abaev_acc.xml	/home/oleg/abaev

/home/oleg/abaevdict/abaev-tei/github/abaevdict-tei/entries/abaev_ad.xml

Figure 3: The representation of *ad* ‘taste’ in Oxygen’s Author mode

Git mechanisms, which is essentially error-proof, because each annotator keep a full local copy of the database on their machine. The use of GitHub also allows for undisruptively making modifications to the schema files: the annotators need only pull the relevant repositories, without the need to “lock” the database.

The elements corresponding to the TLex structure illustrated above are as follows:

form the head word (with the `@type = 'lemma'` property) or various variants (with `@type = 'dialectal'` or `@type = 'inflected'`, and various subtypes of inflected forms);

sense the sense information block that contains definitions or translations in Russian or English;

re corresponds to (Pre/Post)SubentryGroup; the xGroups are not actually needed because TEI XML is position-aware.

cit with `@type = 'exampleGroup'` is admittedly a slight deviation from TEI semantics, given that an example group is not an example itself. However, it is fairly close, because it may only containt `cit` elements which are examples.

etym the etymology block, with mixed content.

Thus, using TEI, the dictionary ends up with a structure that is more complex in some sense, but at the same time less rigid and having less limitations than the TLex model.

5 Conclusion

This paper describes the experience of our research group in an attempt to achieve a double aim: provide a translation of AbD and also digitize it, supplying it with semantic markup. Of course, this is not the first legacy dictionary project that utilizes TEI (Du Fresne Du Cange et al., 1883–1887; Littré, 1863–1873),[1] but the specific challenge is unique due to both its double aim and the complexity (and partial ambiguity) of AbD’s structure. In the talk, we will discuss the dictionary structure, its implementation in TLex and TEI, and the corresponding problems in more detail, attempting to provide a set of best practices for digitizing traditional etymological dictionaries.

References

Vasilij I. Abaev. 1958–1989. *Istoriko-ètimologičeskij slo-*

[1] There is even a modification of TEI for lexicography, called TEI-Lex-0 (DARIAH Working Group “Lexical Resources”, 2020), but it is too rigid for the purposes of faithfully representing an older print dictionary. In particular, mixed content is discouraged, which, in the case of Abaev’s etymologies, is impossible to follow.

var' osetinskogo jazyka. Nauka, Moscow, Leningrad. [Historical-etymological dictionary of Ossetic]. Vols. 1–4. In Russian.

Vasily I. Abaev. 1970. *Russko-osetinskij slovar'*. Nauka, Moscow. [Russian-Ossetic dictionary]. In Russian.

Charles Du Fresne Du Cange et al. 1883–1887. *Glossarium mediæ et infimæ latinitatis*. Niort. Online edition. accessed 07.03.2021.

DARIAH Working Group "Lexical Resources". 2020. *TEI Lex-0: A baseline encoding for lexicographic data*. Accessed 07.03.2021.

Èto Kavkaz. 2020. Istoriko-ètimologičeskij slovar' osetinskogo jazyka pereveli na anglijskij. [The Historical-Etymological dictionary of Ossetic has been translated into English] News item. Accessed 07.03.2021.

Nik'ala Gæbæraty, Tamerlan G̥yriaty, Nafi Ǯusojty, Šamil Ǯykkajty, and Xarum Taqazty. 1999. *Iron ævzaǯy æmbyryngænæn ʒyrdwat*. Tskhinval, Vladikavkaz, Vladikavkaz. [Explanatory dictionary of Ossetic]. In Ossetic.

Iriston.com. 2004. Slovari na iriston.com. [Dictionaries on IRISTON.COM] In Russian. Accessed 07.03.2021.

David Joffe et al. 2021. *TLex Lexicography*. Accessed 07.03.2021.

Alexander M. Kasaev and Tamerlan Aleksandrovič Guriev, editors. 1993. *Osetinsko-russkij slovar'*, 4th edition edition. Izdatel'stvo Severo-Osetinskogo instituta gumanitarnyx i social'nyx issledovanij, Vladikavkaz. [Ossetic-Russian dictionary.] About 28000 words. In Russian.

Émile Littré. 1863–1873. *Dictionnaire de la langue française*. Paris. Accessed 07.03.2021.

Network Working Group. 2009. *Tags for Identifying Languages (BCP 47)*. Accessed 07.03.2021.

OED. 2021. *The Oxford English Dictionary*. OUP, Oxford. Accessed 07.03.2021.

Natalia Serdobolskaya et al. 2021. Beserman-Russian dictionary. Accessed 07.03.2021.

SIL International. 2009. Lexique Pro. Accessed 07.03.2021.

SIL International. 2010. *Field Linguist's Toolbox*. Accessed 07.03.2021.

SIL International. 2021. *FieldWorks Language Explorer*. Accessed 07.03.2021.

Sergei A. Starostin and George S. Starostin. 2003. *The Tower of Babel: An international etymological database project*. Accessed 07.03.2021.

Fedar M. Takazov, editor. 2003. *Digorsko-russkij slovar'*. Vladikavkaz. [Digor-Russian dictionary]. In Russian.

The TEI Consortium. 2021. *P5: Guidelines for Electronic Text Encoding and Interchange*. Accessed 07.03.2021.

Ladislaw Zgusta. 1991. Typology of etymological dictionaries and V. I. Abaev's Ossetic Dictionary. pages 38–49.

Keyword spotting for audiovisual archival search in Uralic languages

Nils Hjortnæs
Indiana University
nhjortn@iu.edu

Niko Partanen
University of Helsinki
Helsinki, Finland
niko.partanen@helsinki.fi

Francis M. Tyers
Department of Linguistics
Indiana University
Bloomington, IN
ftyers@iu.edu

Abstract

In this study we investigate the potential of using Automatic Speech Recognition (ASR) for keyword spotting for four Uralic languages: Finnish, Hungarian, Estonian and Komi. These languages also represent different levels on the high and low resource continuum. Although the accuracy of the ASR systems show there is a long way to go, we show that they still have potential to be useful for downstream tasks such as keyword spotting. By using a simple text search after running ASR, we are already able to achieve an F_1 score of between 0.15 and 0.33, a precision of nearly 0.90 for Estonian and Hungarian, and a precision of 0.76 for Komi.

Tiivistelmä

Tutkimus käsittelee puheentunnistuksen käyttöä avainsanojen tunnistamisessa neljällä uralilaisella kielellä, joita ovat suomi, unkari, viro ja komisyrjääni. Nämä kielet ovat myös eri tasoilla saatavilla olevien resurssien määrän suhteen. Vaikka varsinaiset puheentunnistusjärjestelmät eivät välttämättä vielä toimi toivotulla tavalla, osoitamme, että näitä teknologioita voi jo hyödyntää eri tehtävissä, joista yksi on avainsanojen tunnistus. Kokeissamme avainsanat tunnistetaan suoraan puheentunnistuksen tuottamasta tekstistä. Näin saavutettu tarkkuus on verrattain korkea, mutta herkkyys yhä melko matala.

1 Introduction

Very large quantities of audio recordings exist for Uralic languages, as there is a long history of primary data collection. It is another question how large a portion of these materials are adequately archived, and if they are, whether they are findable and accessible. The situation is continuously improving, and as different archives digitize their collections, the material that can be used relatively easily will keep increasing in size. At the same time materials that are not transcribed, translated or annotated can be very challenging to work with. This problem is not unique to the Uralic language materials, nor linguistic materials in general, but touches archived data very widely.

Computational methods have been recognized as one approach to this issue, and several of the related technologies already give very good results (Blokland et al., 2019). When it comes to speech data, it still remains a challenge to develop high performance speech recognition for endangered or low-resource languages (Xu et al., 2020; Stoian et al., 2020). There has, however, been continuous progress in this field to build tools and methods that would allow integration of speech recognition technology into language documentation workflows (see i.e. Adams et al., 2020).

In this study we investigate the usability of using Automatic Speech Recognition (ASR) for keyword spotting for four Uralic languages: Finnish, Hungarian, Estonian and Komi. This way even ASR models that currently have a lower accuracy could be used effectively in some downstream tasks, of which keyword spotting is an important one. For example, there are often recordings that have accompanying notes or metadata, from which potential keywords can be extracted. In long recordings, locating these sections is, however, very tedious and slow to conduct manually. Keyword spotting would allow easier navigation and verification work with unannotated recordings.

Seventh international workshop on computational linguistics of Uralic languages, pages 20–26
September 23–24, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-1-954085-82-4_003

2 Wider context of archived multimedia

To contextualize even partly how large the scale of unannotated but existing multimedia is, we use Komi as the example in this section. Focus to Komi in this section is also motivated by the fact that the venue where our study is published is at Syktyvkar, Komi Republic, and Komi is the only endangered language which we address, and thereby the need to accurately locate Komi materials also more urgent. We are most familiar with the European archives, and focus to those, although most substantial Komi collections certainly are stored in Syktyvkar. The first audio recordings of Permian Komi and Udmurt were most likely done in 1911 (Денисов, 2014, 34), which is now 110 years ago. This tells that the materials have accumulated for a long period already.

The Archive of Estonian Dialects and Finno-Ugric Languages at the Institute of the Estonian Language (Ermus et al., 2019) contains a large number of recordings in various Uralic languages, and their online catalogue lists 212 Komi recordings that total in 19 hours. Most of their Komi materials have been collected by Anu-Reet Hausenberg and Adolf Turkin.

Similarly, the Institute for the Languages of Finland contains large Komi collections. These start with the work of Erkki Itkonen, who did a fieldwork trip to Syktyvkar in 1958 an (Itkonen, 1958, 70). Very soon after this Günter Johannes Stipa conducted similar trip (Stipa, 1962, 65–66). We also have to highlight the collections Muusa Vahros-Pertamo did in 1962 both with Zyrian and Permian Komi dialects (Vahros-Pertamo, 1963). These materials have not been published. In 1950s and 1960s Erik Vászolyi conducted similar work, and his recordings were later published (Vászolyi-Vasse, 1999), but also copied by Pertti Virtaranta to Helsinki. Also the recordings of Vászolyi do contain several hours of unpublished materials, primarily conversations. The case of Vászolyi is particularly interesting, as the same recordings must be currently copied in several locations: Helsinki, Syktyvkar, Budapest and Perth, Australia, where he was last located before his death. These recordings are approximately 20 hours.

3 Related work

Speech recognition has been previously studied on all of these languages, and some earlier work on keyword spotting also exists. For Finnish and Estonian ASR technologies have already been developed for a long period of time. Among the most recent studies in Finnish ASR is Jain et al. (2020), and for Estonian Alumäe et al. (2019). Enarvi et al. (2017) addressed both of these languages at the same time. A common point of research has been the need to address sub-word segmentation in various ways, as the agglutinative structure of these languages makes the number of unseen word forms potentially very high. At the same time, when the models have been trained with data from media broadcasts and parliamentary proceedings, the recognition of various conversational genres remains a challenge. Work on keyword spotting, or document retrieval in general, has been more scarce, but (Turunen and Kurimo, 2008) have studied the detection of morphemes from unsegmented Finnish audio recordings.

Several experiments for Komi ASR have been conducted, but the quality has not yet reached levels where the models are particularly useful. The steady progress the work has yielded, however, warrants optimism. In the first reported experiment the results were extremely bad, but demonstrated that in principle these systems can be trained with the currently available data, and some insight was shown to the roles the language models and transfer learning may have in the training process (Hjortnaes et al., 2020). A later study refined the language model with online materials, which improved the result considerably (Hjortnaes et al., 2020). All these models used English as the source language in transfer learning. Most recently an investigation was done about the possible use of other languages, and the transfer learning with Russian Common Voice data was tested (Hjortnæs et al., 2021). The results improved due to changes in the DeepSpeech architecture between different versions, but the English transfer learning still gave better results due to the quantity of data available. Further testing of these models by the authors has shown that producing an accurate transcript from a very clearly pronounced Komi speech can work relatively well. In real spontaneous speech the results are extremely sporadic. However, since there is also a clear ratio of correctly recognized words, or their parts, we believe testing the model in real world scenarios for other down stream tasks such as keyword spotting could be very beneficial. When we search for words we expect to occur in the text, we ignore the impact of entirely incorrectly recognized words, and by boosting the individual keywords we improve the possibility of recognizing the words we want to find even further. Unfortunately this scenario is not entirely realistic, as in many instances we cannot know what themes and words are present. However, there are also many instances where metadata containing keyword and topic information exists, and the researchers who have done the recordings often have acute information about the topics covered, which they may want to locate in the recordings more automatically.

Within the research of ASR at Uralic languages we can also mention the study on Samoyedic languages by Partanen et al. (2020), where relatively good accuracies were reported for single speaker scenarios. In the context of minority languages spoken in Russia, Wisniewski et al. (2020) also reported recently on their experiment with Bashkir. There have also been approaches to create keyword spotting without an ASR system at the background (van der Westhuizen et al., 2021).

4 Test data

In the test data we look at two compendia. The first is the Common Voice (Ardila et al., 2020) collection of the data for Hungarian and Estonian, and the second is the collection of available data for Finnish and Komi. The datasets are described below, with the first selection representing more artificial read literary language sentences, and the second containing spontaneous spoken language.

4.1 Common Voice

Common Voice (Ardila et al., 2020) is a project aimed at collecting speech data for all of the world's languages. One of the advantages of Common Voice is that, for the languages supported, it provides a very convenient way to contribute and distribute voice recordings. The data consists of short sentences, typically no longer than 10–15 tokens which are read by a range of different speakers. Readings longer than 10 seconds are discarded.

We followed the training process in Tyers and Meyer (2021) to train speech recognition models for Hungarian and Estonian using the Common Voice data. After training the models we extracted a number of keywords for the two languages from their test sets. We selected all tokens that appeared more than 5 times and that were 5 characters or longer. This second constraint was to try and avoid closed categories that would be unlikely to be used as keywords (e.g. Hungarian *és* 'and' or Estonian *on* 'is').

4.2 Real-word data

As the experiments with Common Voice demonstrate what can be done with read speech, we wanted to see how well the models would work with spontaneous speech of the type more typically found in language archives.

4.2.1 Finnish

The Finnish test data is taken from a CC-BY licensed Samples of Spoken Finnish corpus (Institute for the Languages of Finland, 2014), which contains 100 recordings of 50 Finnish dialects recorded primarily in the 1960s and 1970s. What makes this material particularly relevant is that the recordings originated in the Finnish dialect documentation program, which aimed to record 30 hours of dialect materials from each Finnish municipality. By the end of the 1970s the collections already contained 15,000 hours, and the currently available Finnish dialect materials, in the Institute for the Languages of Finland alone, number 24,000 hours[1]. The materials from which our sample is taken represents a tiny fragment of the recordings that have ever been published in any format.

We have selected five recordings from different dialect regions, and tagged the transcriptions for 100 keywords. The recordings chosen from the corpus were SKN03b_Palkane, SKN10b_Mikkeli, SKN12a_Salla, SKN13b_Pihtipudas and SKN18b_Rautalampi. The keyword tagging is applied on this dataset, and the accuracy is measured. We believe the Finnish results will be generalizable to the wider context of archived Finnish multimedia, at least what it comes to this portion of the dialect recordings. We used the normalized versions of the transcriptions, as those are available in the corpus we used. Those deviate in various ways from the original dialectal representation, but the high variation between word forms in different dialects would had made the comparison of keywords challenging. In the further work, the dialectal variants of the wordforms could be mapped together to allow more dialect-aware keyword search. At the same time, to our knowledge, no ASR system has yet been trained that would even start to address the phenomena met in the dialectal Finnish, and the target of these systems is usually modern literary Finnish. Also the current training data for our Finnish ASR model

[1] https://www.kotus.fi/aineistot/puhutun_kielen_aineistot

Source	Language	Autonym	Locale	Training	# Clips	# Speakers	$\|V\|$
Ardila et al. (2020)	Finnish	Suomi	`fi`	0:32:29	456	1	28
Ardila et al. (2020)	Hungarian	Magyar nyelv	`hu`	4:17:04	3339	2	36
Ardila et al. (2020)	Estonian	Eesti keel	`et`	5:00:16	2760	73	34
Hjortnaes et al. (2020)	Komi	Коми кыв	`kpv`	38:56:02	53711	232	60

Table 1: **Languages and data**. The datasets used in training the speech recognition models that were used in these experiments.

is basically in modern literary Finnish, as it was trained using the read sentences from Common Voice, making it poorly suited for dialectal data.

4.2.2 Komi

For Komi we used a story recorded by Erik Vàszolyi (for various versions of 'Ballad of the soft-haired sister' see Vászolyi-Vasse, 2001; Vászolyi-Vasse and Lázár, 2010), described in a recent study by (Blokland et al., 2021). This is a text that exists in two variants, as it has been recorded both as a sung and narrated version. The narrative version used in this experiment is 17 minutes long. This text is particularly relevant for testing keyword recognition, as it has culturally very relevant content to detect. However, the sang version of the text was already included in the training material of the model, invalidating any results obtained from testing on that data, and thereby excluded from comparison. Especially with the archival data, the same individual is often recorded numerous times, so a situation where some of their recordings are already included into the model is not entirely unrealistic. As always, further testing is obviously required with more speakers and text types. Also for Komi we manually selected 100 keywords that are represented in the text.

As this Komi text was recorded with a tape recorded in 1966, it is very representative of archived Komi materials that do exist in large quantities in different archives. We described the wider context of the archival recordings most familiar to us in Section 2. This illustrates how one central goal in work described here is to be able to better navigate and access untranscribed archival recordings. We describe the related methodology next.

5 Methodology

Keyword spotting is the task of finding specific words in a given audio stream, often containing continuous speech. This has a wide variety of uses, most notably *keyword search* and *wake-word detection*. Keyword searching is when you have a large collection of audio saved on disk, and you want to identify all the instances of certain word. This is especially useful for information retrieval scenarios, and is easily generalizable to the situations where we know something about the recordings, but not exactly where which topic is discussed.

The task discussed in this study, keyword spotting, is just one part of a larger pipeline that related technologies create. This involves text recognition of already written transcriptions, and forced alignment of the text with audio. Keyword spotting usually predates a well functioning ASR, as it can be, arguably, implemented before speech recognition is yet fully established. In the longer perspective keyword tagging is also related to subject indexing, where the topics and keywords are extracted from the document text. Such systems are already successfully in use with larger Uralic languages, such as Finnish (Suominen, 2019). Indeed, keyword spotting would regularly be conducted in a context where we have reasons to assume specific term of interest is used somewhere in the document, be that a text or recording.

While there are specific algorithms for keyword spotting, cf. Mazumder et al. (2021), we use a very simple approach. We decode the audio as if we are performing a normal Speech-to-Text transcription task, and then we do a simple text search over the transcript. In this study we did not use specific keyword boosting techniques, which would be an additional approach to improve the findability of a specific string. Such use cases also distinguish keyword spotting more clearly from speech recognition, as our current methodology essentially uses generated transcription as a starting point.

For the experiments, we took the test set for each language, and selected 10 words at random from a set of those words longer than four characters to favour content words over function words. The results are presented in Table 2.

Language	# Keywords	F_1	Prec	Rec
fi	100	0.15	0.41	0.09
hu	192	0.28	0.89	0.16
et	546	0.33	0.88	0.21
kpv	100	0.20	0.76	0.12

Table 2: **Keyword spotting**. We show the dataset size, precision, recall and F_1 score. In general the precision is high and recall is moderate to low.

6 Results

We will first explain the concepts we have used to measure the model's performance. Precision (Prec) is how often the model is correct when it identifies a keyword. Recall (Rec) is how many of the keywords in the test data the search is able to find. F_1 is a weighted average of precision and recall which tends towards whichever value is lower, meaning the best score is achieved by balancing precision and recall. This gives intuitively interpretable and comparative information about the experiments.

Our results were the best for Estonian and Hungarian. We believe this is largely connected to the narrow domain which was present in the Common Voice recordings, namely that the clips are read. The low accuracy of Finnish is probably related to the small amount of training data. Without an accurate model, the keywords may not be correctly transcribed and will not show up in the text search. In the case of Komi we reach a relatively high precision, on par with Hungarian and Estonian where the domain was narrow, and here the large amount of training data must have some role. However, the clips are from natural speech instead of read, which explains the lower accuracy when compared to Hungarian and Estonian despite the large quantity of training data. This is not an excellent result, but already a step toward a clearly functional system. As the recall is very low, it must stated that the system is not very successful in finding the keywords, but when it suggests them, those are often correct.

We expected Estonian and Hungarian to work relatively well, since the test data was not very realistic. However, the result with Komi comes relatively close to what we see with the test languages. Especially with Finnish experiments with more training data, possibly varying the training data size gradually, could help to understand how the ratio of the training data impacts to the model's performance. Similar experiment was previously conducted successfully for Kamas to evaluate changes in the accuracy (Partanen et al., 2020). We also have to emphasise that the Finnish data was much more strongly dialectal than what would be customarily encountered in the recordings today, and what is present in the Common Voice dataset. Even though such older dialect recordings exist in large quantities in Finnish archives, they must still be considered a special case within Finnish speech technologies in general.

Another challenge, and factor that makes our results less reliable, is that we selected the keywords from the corpora themselves. This was the only available approach, as we wanted to measure the accuracy, but it also targeted our experiment toward the existing inflected forms that do exist in the test data. With agglutinative Uralic languages, however, the most useful test scenario would be one where the desired keywords are listed by their lemmas, but may occur in a different shape in the real usage, and the keyword spotting would ideally still work.

7 Concluding remarks

Our research shows that keyword detection systems are in principle applicable for low resource settings, and even with a very small amount of training data the precision can be relatively high. It certainly is not possible to retrieve all keywords reliably under the current conditions, but even the accuracy we are now reaching could still be useful. Naturally, lots of work still remains to be done within this topic.

One of the most important further tasks would be to extend the experiment into entirely realistic conditions. We could, for example, use archived recordings and their keyword lists and summaries to create the keyword queries, and compare the result against manually verified data. This way we could move toward concrete evaluation of how well and realistically the system performs with various archived datasets. Also different fieldwork collections in Uralic languages could be very well suited for this task. Even though exact keyword and topic listings may not be very common in current metadata models, there is still a long tradition of compiling such topic indexes, and this is inarguably a very useful strategy to classify non-transcribed recordings. Combined to keyword spotting such index can be used to navigate the recordings as well. Our current study is a first step to that direction in a wider context of Uralic languages, and with the goal of trying to test the keyword detection in languages representing different branches of this language family.

Acknowledgments

Niko Partanen has produced this work within the project *Language Documentation meets Language Technology: The Next Step in the Description of Komi*, funded by Kone Foundation, Finland. This research was supported in part by Lilly Endowment, Inc., through its support for the Indiana University Pervasive Technology Institute.

References

Oliver Adams, Benjamin Galliot, Guillaume Wisniewski, Nicholas Lambourne, Ben Foley, Rahasya Sanders-Dwyer, Janet Wiles, Alexis Michaud, Séverine Guillaume, Laurent Besacier, et al. 2020. User-friendly automatic transcription of low-resource languages: Plugging ESPnet into Elpis. *arXiv preprint arXiv:2101.03027*.

Tanel Alumäe, Ottokar Tilk, et al. 2019. Advanced rich transcription system for Estonian speech. *arXiv preprint arXiv:1901.03601*.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common Voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215.

Rogier Blokland, Niko Partanen, and Michael Rießler. 2021. *This is thy brother's voice*. In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual facilitation*. University of Helsinki.

Rogier Blokland, Niko Partanen, Michael Rießler, and Joshua Wilbur. 2019. Using computational approaches to integrate endangered language legacy data into documentation corpora: Past experiences and challenges ahead. In *Workshop on Computational Methods for Endangered Languages, Honolulu, Hawai'i, USA*, volume 2, pages 24–30.

Seppo Enarvi, Peter Smit, Sami Virpioja, and Mikko Kurimo. 2017. Automatic speech recognition with very large conversational Finnish and Estonian vocabularies. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(11):2085–2097.

Liis Ermus, Mari-Liis Kalvik, and Tiina Laansalu. 2019. The Archive of Estonian dialects and Finno-Ugric languages at the Institute of the Estonian language. *Uralica Helsingiensia*, (14):351–366.

Nils Hjortnaes, Niko Partanen, Michael Rießler, and Francis M. Tyers. 2020. Towards a speech recognizer for Komi, an endangered and low-resource Uralic language. In Tommi A. Pirinen, Francis M. Tyers, and Michael Rießler, editors, *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 31–37. Association for Computational Linguistics.

Nils Hjortnæs, Niko Partanen, Michael Rießler, and Francis M. Tyers. 2021. The relevance of the source language in transfer learning for ASR. In Miikka Silfverberg, editor, *Proceedings of the 4th Workshop on Computational Methods for Endangered Languages*, volume 1, pages 63–69. University of Colorado Boulder.

Institute for the Languages of Finland. 2014. Suomen kielen näytteitä - Samples of Spoken Finnish [online-corpus], version 1.0. http://urn.fi/urn:nbn:fi:lb-201407141.

Erkki Itkonen. 1958. Komin tasavallan kielitieteeseen tutustumassa. *Virittäjä*, 62(1):66–66.

Abhilash Jain, Aku Rouhe, Stig-Arne Grönroos, and Mikko Kurimo. 2020. Finnish asr with deep transformer models. In *Conference of the International Speech Communication Association (INTERSPEECH)*, volume 21.

Mark Mazumder, Colby Banbury, Josh Meyer, Pete Warden, and Vijay Janapa Reddi. 2021. Few-shot keyword spotting in any language. *arXiv preprint arXiv:2104.01454*.

Niko Partanen, Mika Hämäläinen, and Tiina Klooster. 2020. Speech recognition for endangered and extinct samoyedic languages. In *Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation*.

Günter Johannes Stipa. 1962. Käynti syrjäänien tieteen tyyssijassa. *Virittäjä*, 66(1):61–68.

Mihaela C Stoian, Sameer Bansal, and Sharon Goldwater. 2020. Analyzing ASR pretraining for low-resource speech-to-text translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7909–7913. IEEE.

Osma Suominen. 2019. Annif: DIY automated subject indexing using multiple algorithms. *LIBER Quarterly*, 1(29):1–25.

Ville T Turunen and Mikko Kurimo. 2008. Speech retrieval from unsegmented Finnish audio using statistical morpheme-like units for segmentation, recognition, and retrieval. *ACM Transactions on Speech and Language Processing (TSLP)*, 8(1):1–25.

Francis M. Tyers and Josh Meyer. 2021. What shall we do with an hour of data? speech recognition for the un- and under-served languages of common voice.

Muusa Vahros-Pertamo. 1963. Syrjäänien asuinseuduilla. *Virittäjä*, 67(1):77–85.

E. Vászolyi-Vasse. 1999. *Syrjaenica*, volume One of *Specimina Sibirica*. Seminar für Uralische Philologie der Berzsenyi Hochschule.

Erik Vászolyi-Vasse. 2001. *Syrjaenica*, volume 2. Seminar für Uralische Philologie der Berzsenyi Hochschule.

Erik Vászolyi-Vasse and Katalin Lázár. 2010. *Songs from Komiland*. Reguly Társaság.

Ewald van der Westhuizen, Herman Kamper, Raghav Menon, John Quinn, and Thomas Niesler. 2021. Feature learning for efficient ASR-free keyword spotting in low-resource languages. *Computer Speech & Language*, page 101275.

Guillaume Wisniewski, Alexis Michaud, Benjamin Galliot, Laurent Besacier, Séverine Guillaume, Katya Aplonova, and Guillaume Jacques. 2020. Ouvrir aux linguistes «de terrain» un accès à la transcription automatique. *Groupement de Recherche Linguistique Informatique Formelle et de Terrain (LIFT)*, page 82.

Jin Xu, Xu Tan, Yi Ren, Tao Qin, Jian Li, Sheng Zhao, and Tie-Yan Liu. 2020. Lrspeech: Extremely low-resource speech synthesis and recognition. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2802–2812.

Виктор Николаевич Денисов. 2014. Из истории первых фонографических записей удмуртов и коми-пермяков в 1911-1912 гг. На территории верхнего Прикамья. *Ежегодник финно-угорских исследований*, (4).

Evaluating Transferability of BERT Models on Uralic Languages

Judit Ács
SZTAKI
Institute for Computer Science and Control
judit@sch.bme.hu

Dániel Lévai
Department of Digital Humanities
Eötvös Loránd University
levai753@gmail.com

András Kornai
SZTAKI
Institute for Computer Science and Control
kornai@sztaki.hu

Abstract

Transformer-based language models such as BERT have outperformed previous models on a large number of English benchmarks, but their evaluation is often limited to English or a small number of well-resourced languages. In this work, we evaluate monolingual, multilingual, and randomly initialized language models from the BERT family on a variety of Uralic languages including Estonian, Finnish, Hungarian, Erzya, Moksha, Karelian, Livvi, Komi Permyak, Komi Zyrian, Northern Sámi, and Skolt Sámi. When monolingual models are available (currently only et, fi, hu), these perform better on their native language, but in general they transfer worse than multilingual models or models of genetically unrelated languages that share the same character set. Remarkably, straightforward transfer of high-resource models, even without special efforts toward hyperparameter optimization, yields what appear to be state of the art POS and NER tools for the minority Uralic languages where there is sufficient data for finetuning.

A BERT- és más Transformer-alapú nyelvmodellek számos angol tesztadaton jobban teljesítenek, mint a korábbi modellek, azonban ezek a tesztadatok az angolra és néhány hasonlóan sok erőforrással rendelkező nyelvre korlátozódnak. Ebben a cikkben egynyelvű, soknyelvű és random súlyokkal inicializált BERT modelleket értékelünk ki a következő uráli nyelvekre: észt, finn, magyar, erza, moksa, karjalai, livvi-karjalai, komi-permják, komi-zürjén, északi számi és kolta számi. Az egynyelvű modellek – jelenleg csak észt, finn és magyar érhető el – ugyan jobban teljesítenek az adott nyelvre, általában rosszabbul transzferálhatóak, mint a soknyelvű modellek vagy a nem rokon, de azonos írást használó egynyelvű modellek. Érdekes módon a sok erőforráson tanult modellek még hiperparaméter optimalizálás nélkül is könnyen transzferálhatók és finomhangolásra alkalmas tanítóadattal csúcsminőségű POS és NER taggerek hozhatóak létre a kisebbségi uráli nyelvekre.

1 Introduction

Contextualized language models such as BERT (Devlin et al., 2019) drastically improved the state of the art for a multitude of natural language processing applications. Devlin et al. (2019) originally released 4 English and 2 multilingual pretrained versions of BERT (mBERT for short) that support over 100 languages including three Uralic languages: Estonian [et], Finnish [fi], and Hungarian [hu]. BERT was quickly followed by other large pretrained Transformer (Vaswani et al., 2017) based models such as RoBERTa (Liu et al., 2019) and multilingual models such as XLM-RoBERTa (Conneau et al., 2019). Huggingface released the Transformers library (Wolf et al., 2020), a PyTorch implementation of Transformer-based language models along with a repository for pretrained models from community contribution[1]. This list now contains over 1000 entries, many of which are domain-specific or monolingual models.

Despite the wealth of multilingual and monolingual models, most evaluation methods are limited to English, especially for the early models. Devlin et al. (2019) showed that the original mBERT outperformed existing models on the XNLI dataset (Conneau et al., 2018), a translation

[1] https://huggingface.co/models

Seventh international workshop on computational linguistics of Uralic languages, pages 27–36
September 23–24, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-1-954085-82-4_004

of the MultiNLI (Williams et al., 2018) to 15 languages. mBERT was further evaluated by Wu and Dredze (2019) for 5 tasks in 39 languages, which they later expanded to over 50 languages for part-of-speech (POS) tagging, named entity recognition (NER) and dependency parsing (Wu and Dredze, 2020). mBERT has been applied to a variety of multilingual tasks such as dependency (Kondratyuk and Straka, 2019) and constituency parsing (Kitaev et al., 2019). The surprisingly effective multilinguality of mBERT was further explored by Dufter and Schütze (2020).

Uralic languages have received relatively moderate interest from the language modeling community. Aside from the three national languages, no other Uralic language is supported by any of the multilingual models, nor does any have a monolingual model. There are no Uralic languages among the 15 languages of XNLI. Wu and Dredze (2020) do explore all 100 languages that mBERT supports but do not go into monolingual details. Alnajjar (2021) transfer existing BERT models to minority Uralic languages, the only work that focuses solely on Uralic languages.

In this paper we evaluate multilingual and monolingual models on Uralic languages. We consider three evaluation tasks: morphological probing, POS tagging and NER. We also use the models in a crosslingual setting, in other words, we test how monolingual models perform on related languages. We show that

- these language models are very good at all three tasks when finetuned on a small amount of task specific data,
- for morphological tasks, when native BERT models are available (et, fi, hu), these outperform the others on their native language, though the advantage over XLM-RoBERTa is not statistically significant,
- for POS and NER, the use of native models from related, even closely related languages, rarely brings improvement over the multilingual models or even English models,
- as long as the alphabet that the language uses is covered in the vocabulary of the model, we can transfer mBERT (or RuBERT) to the NER and POS tasks with surprisingly little finetuning data.

2 Approach

We evaluate the models through three tasks: morphological probing, POS tagging and NER. Uralic languages have rich inflectional morphology and largely free word order. Morphology plays a key role in parsing sentences. Morphological probing tries to recover morphological tags from the sentence representation from these models.

For assessing the sentence level behavior of the models we chose two token-level sentence tagging tasks, POS and NER. Part of speech tagging is a common subtask of downstream NLP applications such as dependency parsing. Named entity recognition is indispensable for various high level semantic applications such as building knowledge graphs. Our model architecture is identical for POS and NER.

2.1 Morphological probing

Probing is a popular evaluation method for black box models. Our approach is illustrated in Figure 1. The input of a probing classifier is a sentence and a target position (a token in the sentence). We feed the sentence to the contextualized model and extract the representation corresponding to the target token. Early experiments showed that lower layers retain more morphological information than higher layers so instead of using the top layer, we take the weighted average of all Transformer layers and the embedding layer. The layer weights are learned along with the other parameters of the neural network. We train a small classifier on top of this representation that predicts a morphological tag. We expose the classifier to a limited amount of training data (2000 training and 200 validation instances). If the classifier performs well on unseen data, we conclude that the representation includes the relevant morphological information.

We generate the probing data for Estonian and Finnish from the Universal Dependencies (UD) Treebanks (Nivre et al., 2020; Haverinen et al., 2014; Pyysalo et al., 2015; Vincze et al., 2010) and from the automatically tagged Webcorpus 2.0 for Hungarian since the Hungarian UD is very small. Unfortunately we could not extend the list of languages to other Uralic languages because their treebanks are too small to sample enough data.

The sampling method is constrained so that the target words have no overlap between train, validation and test, and we limit class imbalance to 3-to-1 which resulted in filtering some rare values. We

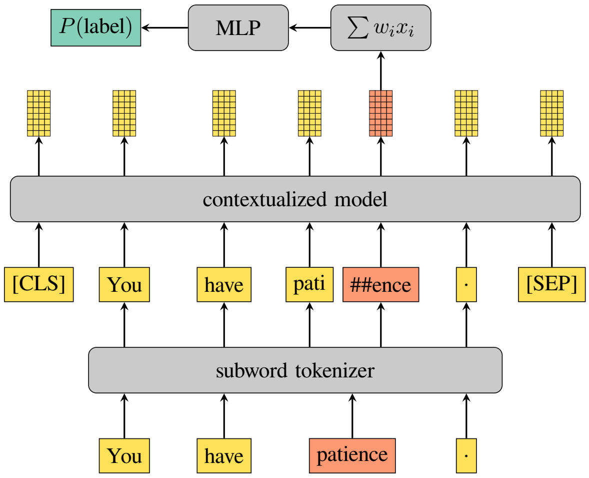


Figure 1: Probing architecture. Input is tokenized into subwords and a weighted average of the mBERT layers taken on the last subword of the target word is used for classification by an MLP. Only the MLP parameters and the layer weights w_i are trained.

were able to generate enough probing data for 11 Estonian, 16 Finnish and 11 Hungarian tasks, see Table 4 for the full list of these.

2.2 Sequence tagging tasks

Our setup for the two sequence tagging tasks is similar to that of the morphological probes except we train a shared classifier on top of all token representations. We use the vector corresponding to the first subword in both tasks. Although this may be suboptimal in morphology, Ács et al. (2021) showed that the difference is smaller for POS and NER. We also finetune the models which seems to close the gap between first and last subword pooling for morphology, see 4.1. For sequence tagging tasks, unlike for morphology, we found that the weighted average of all layers is suboptimal compared to simply using the top layer, so the experiments presented here all use the top layer.

We sample 2000 train, 200 validation and 200 test sentences as POS training data from the largest UD treebank in Estonian and Finnish, and from Webcorpus 2.0 for Hungarian. Aside from these three, Erzya [myv]; Moksha [mdf]; Karelian [krl]; Livvi [olo]; Komi Permyak [koi]; Komi Zyrian [kpv]; Northern Sámi [sme]; and Skolt Sámi [sms] have UD treebanks (Rueter and Tyers, 2018; Rueter, 2018; Pirinen, 2019; Rueter, 2014; Rueter et al., 2020; Partanen et al., 2018; Sheyanova and Tyers, 2017), but these are considerably smaller in size.

Language	Code	Morph	POS	NER
Hungarian	[hu]	26k	2000	2000
Finnish	[fi]	38k	2000	2000
Estonian	[et]	26k	2000	2000
Erzya	[myv]	0	1680	1800
Moksha	[mdf]	0	164	400
Karelian	[krl]	0	224	0
Livvi	[olo]	0	122	0
Komi Permyak	[koi]	0	78	2000
Komi Zyrian	[kpv]	0	562	1700
Northern Sámi	[sme]	0	2000	1200
Skolt Sámi	[sms]	0	101	0

Table 1: Size of training data for each language.

Although none of these languages are officially supported by any of the language models we evaluate, we train crosslingual models and find that the models have remarkable crosslingual capabilities.

Our NER data is sampled from WikiAnn (Pan et al., 2017). WikiAnn has data in Erzya, Estonian, Finnish, Hungarian, Komi Permyak, Komi Zyrian, Moksha, and Northern Sámi.[2] Similarly to the POS training data, we sample 2000 training, 200 validation and 200 test sentences when available, see Table 1 for actual training set sizes.

2.3 Training details

We train all classifiers with identical hyperparameters. The classifiers have one hidden layer with 50 neurons and ReLU activation. The input and the output dimensions are determined by the choice of language model and the number of target labels. The classifiers have 40 to 60k trainable parameters which are randomly initialized and updated using the backpropagation algorithm. We run experiments both with and without finetuning the language models. Finetuning involves updating both the language model (all 110M parameters) and the classification layer (end-to-end training).

All models are trained using the AdamW optimizer (Loshchilov and Hutter, 2019) with $lr = 0.0001, \beta_1 = 0.9, \beta_2 = 0.999$. We use 0.2 dropout for regularization and early stopping based on the development set. We set the batch size to 128 when not finetuning the models, and we use batch size 8, 12 or 20 when we finetune them.

The evaluated models, all from the

[2] WikiAnn also has Udmurt data, but the transcription is problematic: Latin and Cyrillic are used inconsistently, Wikipedia Markup is parsed incorrectly etc.

BERT/RoBERTa family, differ only in the choice of training data and the training objective. They all have 12 Transformer layers, with 12 heads, and 768 hidden dimensions, for a total of 110M parameters.

3 The models evaluated

Our goal is twofold: we want to assess monolingual models against multilingual models, and we want to evaluate the models on 'unsupported' languages, both typologically related and unrelated.

We pick two multilingual models, mBERT and XLM-RoBERTa. Our choices for monolingual models are EstBERT for Estonian, FinBERT for Finnish and HuBERT for Hungarian (See Table 2). As a control, we also test the English BERT as a general test for cross-language transfer. Since many Uralic speaking communities are in Russia and the languages are heavily influenced by Russian, we test RuBERT on these languages. Finally, we also test a randomly initialized mBERT. We do this because the capacity of the BERT-base models is so large that they may memorize the probing data alone. Many models have cased and uncased version, the latter often removing diacritics along with lowercasing. Since diacritics play an important role in many Uralic languages, we only use the cased models. We return to this issue in 3.1.

The models along with their string identifier are summarized in Table 2.

3.1 Subword tokenization

Subword tokenization is a key component in achieving good performance on morphologically rich languages. There are two different tokenization methods used in the models we compare: XLM-RoBERTa uses the SentencePiece algorithm (Kudo and Richardson, 2018), the other models use the WordPiece algorithm (Schuster and Nakajima, 2012). The two types of tokenizers are algorithmically very similar, the differences between them are mainly dependent on the vocabulary size per language. The multilingual models consist of about 100 languages, and the vocabularies per language apper sublinearly proportional to the amount of training data available per language: in case of mBERT, 77% of the word pieces are pure ascii (Ács, 2019).

The native models, trained on monolingual data, have longer and more meaningful subwords (see the bolded entries in Table 3). This greatly facilitates the sharing of train data, a matter of great importance for Uralic languages where there is little text available to begin with.

Both BERT- and RoBERTa-based models first tokenize along whitespaces, but the handling of missing characters differs significantly. In BERT-based models, if there is a character missing from the tokenizer's vocabulary, the model discards the whole segment between whitespaces, labeling it [UNK]. In cross-lingual cases many words are lost since monolingual models tend to lack the extra characters of a different language. In contrast, XLM-RoBERTa deletes the unknown characters, but the string that remains between whitespaces is segmented, so the loss of information is not as severe.

Table 3 summarizes different measures in language-model pairs. As a general observation, Latin script models (FinBERT, HuBERT, EstBERT) are unusable on Cyrillic text, as seen e.g. on Erzya, where Latin script models produce [UNK] token for 97.5% of the word types. This is also seen for Northern Sámi and Hungarian, which have many non-ascii characters (á, é, í, ó, ö, ő, ú, ü, ű for Hungarian, č, đ, ŋ, š, ŧ, ž for Northern Sámi) see the Hungarian-EstBert/FinBERT pairs and the Northern Sámi-FinBERT/HuBERT pairs.

The mean subword length generally lies between 3.0 and 3.5 for most pairs - naturally, the corresponding language-model pairs have much higher mean subword length, 5.0 to even 5.9. This range is true not only for Latin script languages, but for Cyrillic script languages as well, as indicated by Erzya, which has a mean subword length of 3.1 to 3.4 on the multilingual models and on RuBERT.

Fertility (Ács, 2019) is defined as the average number of BERT word pieces found in a single real word type. EstBERT on Estonian and FinBERT on Finnish have very similar fertility values (2.1 and 1.9), but HuBERT on Hungarian has much higher fertility. This is mainly caused by the different vocabulary sizes - the Finnic models have 50000 subwords in their vocabulary, HuBERT only contains 32000 subwords. The rest of the fertility values are mostly over 3. In extreme cases, a word is segmented into letters, which is the case for EngBERT on Erzya, but the non-Hungarian models on Hungarian also produce very high fertility values.

Model	Identifier	Language(s)	Reference
mBERT	bert-base-multilingual-cased	100+ inc. et, fi, hu	Devlin et al. (2019)
XLM-RoBERTa	xlm-roberta-base	100 inc. et, fi, hu	Liu et al. (2019)
EstBERT	tartuNLP/EstBERT	Estonian	Tanvir et al. (2021)
FinBERT	TurkuNLP/bert-base-finnish-cased-v1	Finnish	Virtanen et al. (2019)
HuBERT	SZTAKI-HLT/hubert-base-cc	Hungarian	Nemeskey (2020)
EngBERT	bert-base-cased	English	Devlin et al. (2019)
RuBERT	DeepPavlov/rubert-base-cased	Russian	Kuratov and Arkhipov (2019)
rand-mBERT	mBERT with random weights	any	described in Section 3

Table 2: List of models we evaluate.

	mBERT	RoBERTa	EstBERT	FinBERT	HuBERT	RuBERT	EngBERT
Vocab. size	120k	250k	50k	50k	32k	120k	29k
Missing [et] (%)	.0	.0	**.2**	.0	.5	.1	.2
Missing [fi] (%)	.0	.0	.0	**.0**	.4	.0	.0
Missing [hu] (%)	.1	.0	21.5	48.3	**.1**	2.7	.2
Missing [sme] (%)	.2	.0	15.0	47.4	5.1	4.8	.2
Missing [myv] (%)	.0	.0	97.5	97.5	97.5	.0	.0
Subword length [et]	3.7±1.4	4.2±1.7	**5.8**±2.6	3.7±1.4	3.1±1.2	3.1±1.2	3.5±1.4
Subword length [fi]	3.8±1.4	4.5±1.9	3.8±1.4	**5.9**±2.5	3.1±1.1	3.1±1.1	3.4±1.4
Subword length [hu]	3.5±1.5	4.2±2.0	3.3±1.2	3.1±1.1	**5.0**±2.4	3.0±1.1	3.3±1.4
Subword length [sme]	3.2±1.0	3.4±1.1	3.2±1.1	3.2±1.1	3.1±1.2	2.9±1.0	3.0±1.0
Subword length [myv]	3.1±1.2	3.2±1.0	1.0±0.0	1.0±0.0	1.0±0.0	3.4±1.2	1.1±0.4
Character length [et]	9.2	9.2	**9.2**	9.2	9.2	9.2	9.2
Character length [fi]	9.3	9.3	9.3	**9.3**	9.3	9.3	9.3
Character length [hu]	9.8	9.8	9.6	8.8	**9.8**	9.8	9.9
Character length [sme]	8.5	8.5	8.3	7.6	8.5	8.4	8.5
Character length [myv]	7.3	7.3	1.8	1.8	1.7	7.3	7.3
Fertility [et]	3.4	2.8	**2.1**	3.6	4.4	4.3	4.3
Fertility [fi]	3.3	2.7	3.5	**1.9**	4.6	4.4	4.5
Fertility [hu]	4.0	3.2	5.2	4.5	**2.8**	5.4	5.6
Fertility [sme]	3.7	3.6	4.1	3.3	4.5	4.6	4.7
Fertility [myv]	3.6	3.3	1.1	1.1	1.1	3.0	7.2

Table 3: Major characteristics of cross-language tokenization. Boldface font marks the corresponding language-model pairs.

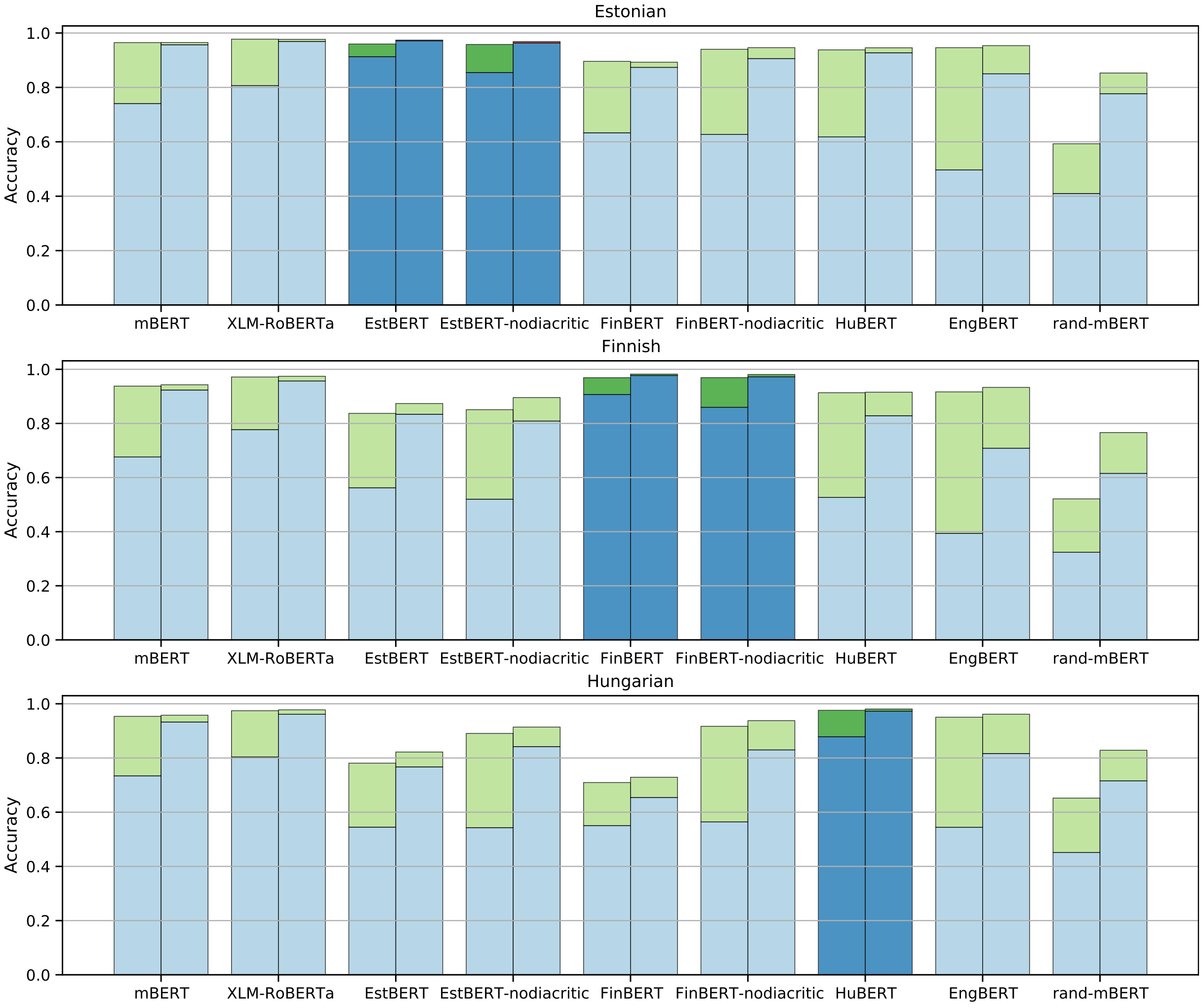


Figure 2: Mean accuracy of morphological tasks by language. The bars are grouped in two, the left one is the result of probing the first subword, the right one is the results of probing the last subword. Blue bars are without finetuning, green bars are with finetuning. Monolingual models are highlighted.

4 Results

4.1 Morphology

Morphological tasks are generally easy for most models and we see reasonable accuracy from crosslingual models as illustrated by Figure 2. Mean accuracies, especially after finetuning, are generally above 90%, except, unsurprisingly, for the randomly initialized models.

Subword choice We first start by examining the choice of subword on morphological tasks. We try probing the first and the last subword and we find that there is a substantial gap in favor of the last subword. This is unsurprising considering that Uralic languages are mainly suffixing. This gap on average shrinks from 0.21 to 0.032 when we finetune the models on the probing data (Figure 2 shows this gap in green). Without finetuning there is only one task, ⟨Hungarian, Degree, ADJ⟩, where probing the first subword is better than probing the last one for some models. This is explained by the fact that the superlative in Hungarian is formed from the comparative by a prefix.

Monolingual models are only slightly better than the two multilingual models, XLM-RoBERTa in particular. We run paired t-tests on the accuracy of each model pair over the 11 (et, hu) or 16 (fi) morphological tasks in a particular language and find that the difference between the monolingual model and XLM-RoBERTa is never significant, and for Estonian, neither is the difference between EstBERT and mBERT.

Cross-lingual transfer works only if we finetune the models. Interestingly, language relatedness does not seem to play a role here. FinBERT transfers worse to Estonian than HuBERT, and EstBERT transfers worse to Finnish than HuBERT. Interestingly, EngBERT transfers better to all three models than the other native BERTs, and for Finnish and Hungarian it is actually on par with mBERT.

Diacritics As seen from the first panel of Table 3, EstBERT and FinBERT replace words with unknown characters with [UNK] to such an extent that a large proportion of types end up being filtered. We try to mitigate this issue by preemptively removing all diacritics from the input. It appears that this has little effect on the original language, but cross-lingual transfer is improved for Finnish. In the sequence tagging tasks that we now turn to, we remove the diacritics when we evaluate EstBERT or FinBERT in a cross-lingual setting.

4.2 POS and NER

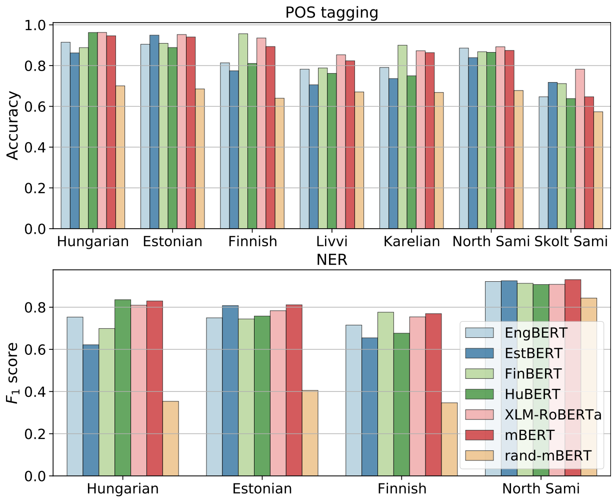


Figure 3: POS and NER results on languages that use the Latin alphabet.

We extend our studies to all Uralic languages with any training data (see Table 1) and we limit the discussion to finetuned models since cross-lingual transfer does not work without finetuning. We split the languages into two groups, Latin and Cyrillic, and we only test models with explicit support for the script that the language uses. Multilingual models support both scripts. Figures 3 and 4 show the results by language.

National languages We generally find the best performance in the three languages with native support: Estonian, Finnish and Hungarian. Monolingual models perform the best in their respective language but the two multilingual models are also very capable.

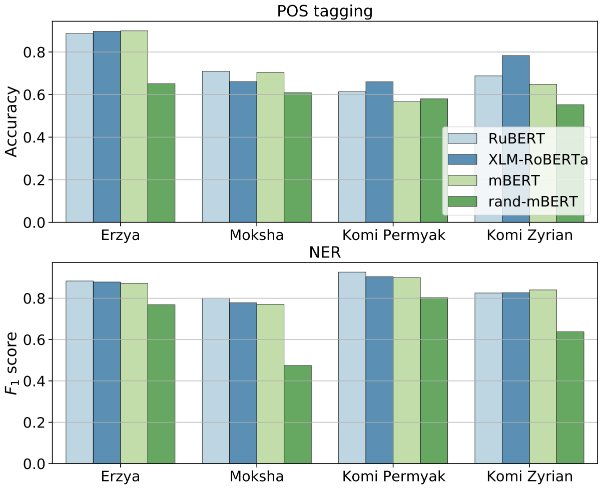


Figure 4: POS and NER results on languages that use the Cyrillic alphabet.

Cross-lingual transfer does not seem to benefit from language relatedness, EngBERT transfers just as well as other monolingual models. Even extremely close relatives such as Livvi and Finnish do not transfer better than XLM-RoBERTa to Livvi. On the other hand, FinBERT is the best for Karelian POS, another close relative of Finnish. The writing system and shared vocabulary also seem to play an important role, as seen from RuBERT's usefulness on unrelated but Cyrillic-using Uralic languages, see Figure 4.

XLM-RoBERTa is generally a strong model for cross-lingual transfer for all Uralic languages. We suspect that this is due to its large subword vocabulary, which may provide a better generalization basis for capturing the orthographic cues that are often highly indicative in agglutinative languages.

North Sámi Both POS and NER in North Sámi are relatively easy as long as the orthographic cues can be captured (i.e. the Latin script is supported). rand-mBERT is suprisingly successful at NER in North Sámi, suggesting that orthograpic cues (rand-mBERT uses mBERT's tokenizer) are highly predictive of named entities in North Sámi.

5 Conclusion

Altogether we find that it is possible, and relatively easy, to transfer models to new languages with finetuning on very limited training data, though extremely limited data still hinders progress: compare Erzya (1680 train sentences) to Moksha (164 train sentences) on Fig. 4.

Morph tag	POS	Estonian	Finnish	Hungarian
Case	adj	8 classes	11 classes	
Case	noun	15 classes	12 classes	18 classes
Case	propn		8 classes	
Case	verb		12 classes	
Degree	adj		Cmp, Pos, Sup	Cmp, Pos, Sup
Derivation	adj		Inen, Lainen, Llinen, Ton	
Derivation	noun		Ja, Lainen, Minen, U, Vs	
InfForm	verb		1, 2, 3	
Mood	verb		Cnd, Imp, Ind	Cnd, Imp, Ind, Pot
Number psor	noun			Sing, Plur
Number	a/n/v	Sing, Plur	Sing, Plur	Sing, Plur
PartForm	verb		Pres, Past, Agt	
Person psor	noun			1, 2, 3
Person	verb	1, 2, 3		1, 2, 3
Tense	adj	Pres, Past		
Tense	verb	Pres, Past	Pres, Past	Pres, Past
VerbForm	verb	Conv, Fin, Inf, Part, Sup	Inf, Fin, Part	Inf, Fin
Voice	adj	Act, Pass		
Voice	verb	Act, Pass	Act, Pass	

Table 4: List of morphological probing tasks.

EngBERT and RuBERT, which we introduced as a control for language transfer among genetically unrelated languages, transfer quite well: in particular the Latin-script EngBERT transfers better to Hungarian than FinBERT or EstBERT.

We note that we did not perform monolingual hyperparameter search or any preprocessing, and there is probably room for improvement for each of these languages. The biggest immediate gains are expected from extending the UD and WikiAnn datasets, and from careful handling of low-level characterset and subword tokenization issues. There are many Uralic languages that still lack basic resources, in particular the entire Samoyedic branch, Mari, and Ob-Ugric languages, are currently out of scope. Another avenue of research could be to work towards a stronger mBERT interlingua, or perhaps one for each script family, as the charset issues are clearly relevant.

Our data, code and the full result tables will be available along with the final submission.

Acknowledgements

This work was partially supported by the Ministry of Innovation and the National Research, Development and Innovation Office within the framework of the Artificial Intelligence National Laboratory Programme.

References

Judit Ács, Ákos Kádár, and Andras Kornai. 2021. Subword pooling makes a difference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2284–2295, Online. Association for Computational Linguistics.

Khalid Alnajjar. 2021. When word embeddings become endangered. *Multilingual Facilitation*, page 275–288.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, 48:493–531. Open access.

Nikita Kitaev, Steven Cao, and Dan Klein. 2019. Multilingual constituency parsing with self-attention and pre-training. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3499–3505, Florence, Italy. Association for Computational Linguistics.

Dan Kondratyuk and Milan Straka. 2019. 75 languages, 1 model: Parsing universal dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for russian language.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Dávid Márk Nemeskey. 2020. *Natural Language Processing Methods for Language Modeling*. Ph.D. thesis, Eötvös Loránd University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Niko Partanen, Rogier Blokland, KyungTae Lim, Thierry Poibeau, and Michael Rießler. 2018. The first Komi-Zyrian Universal Dependencies treebanks. In *Proceedings of the Second Workshop on Universal Dependencies (UDW 2018)*, pages 126–132, Brussels, Belgium. Association for Computational Linguistics.

Tommi A Pirinen. 2019. Building minority dependency treebanks, dictionaries and computational grammars at the same time—an experiment in karelian treebanking. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 132–136.

Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. 2015. Universal Dependencies for Finnish. In *Proceedings of NoDaLiDa 2015*, pages 163–172. NEALT.

Jack Rueter. 2014. The Livonian-Estonian-Latvian Dictionary as a threshold to the era of language technological applications. *Journal of Estonian and Finno-Ugric Linguistics*, 5(1):251–259. ESUKA – JEFUL 2013, 5–1: 253–261 Volume: Proceeding volume:.

Jack Rueter. 2018. Erme ud moksha.

Jack Rueter, Niko Partanen, and Larisa Ponomareva. 2020. On the questions in developing computational infrastructure for Komi-permyak. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 15–25, Wien, Austria. Association for Computational Linguistics.

Jack Rueter and Francis Tyers. 2018. Towards an open-source universal-dependency treebank for erzya. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 106–118.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Mariya Sheyanova and Francis M. Tyers. 2017. Annotation schemes in north sámi dependency parsing. In *Proceedings of the 3rd International Workshop for Computational Linguistics of Uralic Languages*, pages 66–75.

Hasan Tanvir, Claudia Kittask, Sandra Eiche, and Kairit Sirts. 2021. Estbert: A pretrained language-specific bert for estonian.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Veronika Vincze, Dóra Szauter, Attila Almási, György Móra, Zoltán Alexin, and János Csirik. 2010. Hungarian dependency treebank. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo. 2019. Multilingual is not enough: BERT for Finnish.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual BERT? In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 120–130, Online. Association for Computational Linguistics.

Judit Ács. 2019. Exploring bert's vocabulary. `http://juditacs.github.io/2019/02/19/bert-tokenization-stats.html`. Accessed: 2021-05-14.

No more fumbling in the dark – Quality assurance of high-level NLP tools in a multi-lingual infrastructure

Linda Wiechetek linda.wiechetek@uit.no
Flammie A Pirinen tommi.pirinen@uit.no
Børre Gaup boerre.gaup@uit.no
Thomas Omma thomas.omma@uit.no

Divvun, UiT Norgga árktalaš universitehta

Abstract

We argue that regression testing is necessary to ensure reliability in the continuous development of NLP tools, especially higher level applications like grammar checkers. Our approach is rule-based, building on successful work for a number of low-resourced languages over the last 20 years. Instead of working with a black box, we choose a method that allows us to pinpoint the exact reasons for failures in the system. We present a tool for regression testing for *GramDivvun*, the rule-based open source North Sámi grammar checker. The regression tool is available for any of the 135 languages in the Giella-LT infrastructure and can be applied when respective tools are built. An evaluation of the system shows how the precision of the regression tests improves with almost 20% over a time span of 1.5 years. We also illustrate that the regression tool can detect undesired effects of rule changes that affect the performance of the grammar checker.

Abstrákta

Mii ákkastallat ahte regrešuvdnaiskosat leat dárbbašlaččat jus galgá sáhttit ráhkadit luohtehahtti NLP-reaidduid, erenoamážit reaidduid nugo grammatihkkadárkkisteddjiid, mat sorjástit máŋga eará prográmmaide. Min bargu lea njuolggadusvuođđuduvvon, huksejuvvon barggu ala mii lea dahkkon smávva-resursagielaiguin maŋemuš 20 jagi. Dan sajis go bargat ”čáhppes bovssain”, mii válljet vuogi man bokte mii dalán oaidnit gokko vuogádagas meattáhus čuožžila. Mii čájehit reaiddu mii iská leatgo *GramDivvumis* regrešuvnnat. *GramDivvun* lea njuolggadusvuođđuduvvon davvisámi rabas gáldokoda grammatihkkadárkkisteaddji. Regrešuvdnaiskanreaidu lea olámuttus visot 135 gillii mat leat GiellaLT-infrastruktuvrras ja dan sáhttá vuodjit go gullevaš reaiddut leat huksejuvvon. Vuogádatevalueren čájeha ahte regrešuvdnaiskosiid bohtosat leat buoránan measta 20 %:in beannot jagis. Mii maid čájehit ahte regrešuvdnaiskanreaidu gávdná meattáhusaid maŋŋá rievdadusaid mat váikkuhit grammatihkkadárkkisteaddji bohtosiidda.

Tiivistelmä

Tässä artikellissa esitämme että regressiotestaus on välttämätöntä kielitekonologiatyökalujen, eritoten korkeampitasoisten sovellusten kuten kieliopiontarkistinten, jatkuvassa kehityksessä. Meidän lähestymislähtökohtamme on sääntöpohjainen, ja rakentuu aiemmalle vähäresurssisten kielten työlle viimeisen 20 vuoden ajalta. Musta laatikko -lähestymistavan sijaan käytämme menetelmiä joiden avulla voimme suoraan paikantaa ongelmakohdat järjestelmässä. Esittelemme työkaluja joilla regressiotestataan *GramDivvunia*, sääntöpohjaista pohjoissaamen kieliopintarkistinta. Regressiotestaus on valmiina käytettävissä 135 kielelle, joita kehitetään GiellaLT-infrastruktuurissa ja sitä voi hyödyntää vastaavissa työkaluissa. Järjestelmää evaluoimalla huomaamme että tarkkuus kasvaa 20 % 1,5 vuoden seurantajakson aikana. Sen lisäksi tuomme esille kuinka regressiotesteillä voi havaita säännöstömuutosten vaikutuksia kieliopintarkistimen suorituskykyyn.

Seventh international workshop on computational linguistics of Uralic languages, pages 37–46
September 23–24, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-1-954085-82-4_005

1 Introduction

This paper illustrates an efficient way to quality check high level rule-based NLP applications for low resource languages with complex morphology like North Sámi. In particular, we develop a powerful regression testing tool for the rule-based open source North Sámi grammar checker *GramDivvun* (Wiechetek et al., 2019a) that provides statistics of precision and recall specific to each error type[1] and a detailed analysis of each sentence including one or more (nested) errors[2], together with an advanced system of error mark-up that allows us to properly identify each error type module that is successful enough to be included in the grammar checker released to the public.

GramDivvun has been released by Divvun as a free plugin for Microsoft Office and Google Docs[3]. A grammar checker, as opposed to a spellchecker, is a tool that verifies and corrects errors in writing that are not mere mistyped non-words, but real words where the error is dependent on the whole sentence-context and its grammatical features.

North Sámi is a minority language in a bilingual language community, which faces challenges as regards writing proficiency. In this context, a reliable grammar checker can therefore also serve as a tool to improve writing skills. However, it is a difficult task to make a precise tool that meets users needs. If it underlines too many or even any correct sentences, the user will easily be frustrated and switch off the grammar checking. Regression testing resolves this problem in a robust and uniform way and ensures high quality of the tools.

North Sámi is a Uralic language spoken in Norway, Sweden and Finland by approximately 25,700 speakers (Simons and Fennig, 2018). It is a synthetic language, where the open parts of speech (PoS) – e.g. nouns, adjectives – inflect for case, person, number and more. The grammatical categories are expressed by a combination of suffixes and stem-internal processes affecting root vowels and consonants alike, making it perhaps the most fusional of all Uralic languages. In addition to compounding, inflection and derivation are common morphological processes in North Sámi. Due to its morphological complexity and, in addition, a large amount of homonymous forms or similar forms that can be confused in writing, there are many different grammatical error types. Similarly to other low-resource languages, there is little to no error marked-up data available for it, and the available data is seldom quality checked with regard to spelling and grammar. This poses a challenge to automatic grammar checking and testing.

Regression testing within software programming practice is defined as testing that ensures that recent code changes do not have any negative effects on existing features.[4] While regression testing is not a new idea and has been applied for some decades, to our knowledge, there are no in-detail publications of the challenges and practical solutions for it in grammar checking. However, Butt and Holloway King (2003) describe different testing strategies and their necessity for syntactic parsing. Since 2003, complexity of Natural Language Processing (NLP) tools has increased, which also requires adapting appropriate testing routines.

The rule-based model enables us to be very precise in locating the shortcomings of our grammar checker, and the regression tests ensure that the grammar checker keeps improving as new rules and tests to check them are added. The novelty in our approach to building grammar checkers lies in the workflows of simultaneously building the grammar checker rules, the error corpus and the regression testing suite. This workflow is an efficient approach to both building regression data and constructing our tools. The features of our tool are powerful enough to handle these multi-modular applications as well as an advanced mark-up system for a real world corpus that includes some spelling, morphological, syntactic, punctuation, space, real-word errors as well as nested errors per sentence. Also, the regression tool provides a detailed error analysis and not just overall regression statistics. It outputs error-specific statistics, including error subtypes, and enables efficient debugging of the system. The regression tools come with a database of tests, including several thousand sentences marked-up manually per error type.

2 Background

2.1 Framework

We are using a NLP development infrastructure called *GiellaLT* (Moshagen et al., 2014), which is

[1] More information on the different error types covered in *GramDivvun* can be found in (Wiechetek, 2017) and (Wiechetek et al., 2019b)

[2] Nested errors are errors within errors (typically with different scopes), for example a typo within an agreement error.

[3] `https://divvun.no/korrektur/gramcheck.html`

[4] `https://www.guru99.com/regression-testing.html` (Accessed 2021-03-23)

at present used by 135 languages. It consists of systems capable of building, testing and deploying a large range of NLP applications – including spelling and grammar checkers among others – based on finite-state morphology (Beesley and Karttunen, 2003) and Constraint Grammar (Karlsson, 1990). We apply a rule-based approach, which has a long tradition for the previously mentioned 135 languages, but is not as wide-spread as neural network approaches these days. Neural networks have shown to provide good results for many higher level NLP applications. However, they are also known to require large amounts of high quality or marked-up data, which for North Sámi would mean a manual quality check or mark-up as this data is not available. Our current error marked-up corpus (for all error types including nested errors) contains 120,459 words—a typical amount for training a neural network is at least several millions, and for a morphologically complexer language possibly more. Considering the amount of different types of errors there are and that not all of the sentences contain an error at all, this is very little data to train any kind of model.

Our work strategy consists in minimizing the workload by a combination of developing rule-based tools that reliably annotate and quality check our data and searching for and annotating example sentences from the corpus that give us further insight in the grammatical issue we are dealing with.

There is current work on neural network error detection/correction for specific 'simpler' grammatical errors (i.e. compound errors) in North Sámi that do not involve changing morphological forms or restructuring of a whole sentence (Wiechetek et al., 2021). However, rule-based tools were used, both to prepare the data and to access PoS information. Furthermore, its insertion of non-sense words restricts its usability for a community of real users. A full-fledged neural network grammar checker - that is not based on the rule-based grammar checker - is not to be realized in the near future.

Rule-based methods have the advantage of formalizing concise rules about the grammatical structure of a language. This gives us detailed insights in the language - as opposed to the black box of a neural network. This knowledge is necessary for defining errors in the first place, especially in cases where normative descriptions do not exist. It is also a prerequisite for debugging errors in our system. As we are able to translate language insights into formal grammar rules, we can pinpoint the exact causes of errors in our system. In other words, we can write a grammar that is both machine-, and to some extent, human-readable, which means that our knowledge can be used in other contexts outside of the grammar checker.

In the context of grammar checking tasks, specifically for morphologically complex and/or low-resourced languages, we would like to discuss two relevant tasks for neural network approaches, i.e. the systems for Latvian (Deksne, 2019) and Russian (Rozovskaya and Roth, 2019). The evaluation of Latvian neural network grammar checker shows a good performance with precisions between 78% and 98.5% (evaluated on a corpus of 115,000 sentences) depending on the error type. However, judging from their regular expressions to insert artificial errors, most of their error types seem to be fairly local errors that can be resolved based on shorter n-grams. The Russian system, on the other hand, focuses on more advanced error types, including case and agreement. However, precision (evaluated on a 206,258 token learners' corpus) is significantly lower — between 22% and 56%, only gender agreement reaches 68%. The corpus is rather small with regard to the task of correcting a large variety of errors. None of these two approaches deal with the advanced syntactic constructions we resolve in our approach, requiring an analysis of the whole sentence, valencies, semantic cues, etc.

The testing approach described here, while used in conjunction with a rule-based system, is agnostic of underlying technology, and could well be applied in the context of a neural system as well, should there be one that allows for correcting the errors the system makes.

2.2 Continuous integration and deployment

In order to provide a consistent grammar checking experience but also automatic updates and improvement, we apply stringent testing and combine that with a *continuous integration / deployment* (CI/CD) environment. To our knowledge, there are no publications on how to apply CI / CD to NLP product pipelines such as grammar checking, so in this article we lay out some guidelines and good practices. However, in the text books for the development of NLP applications we find some recommendations on the use of regression tests to compare different versions of the same application. (Grove, 2009, p.222) There have also been some workshops on regression testing in NLP, e.g. (Farrow and Dzikovska, 2009), however, these ideas have not found popular use, yet. One of the scientific contributions of our work is not only that we can provide the end users with products that work as expected, but also we can maintain scientific integrity of the systems in terms of *reproducibility*. We can apply the CI methods to ensure that systems can reproduce comparable results at all times. This is especially attractive for our case, since we apply mainly rule-based methods for grammar checking and correction, the results should stay relatively stable for the same versions of the system. In the recent years, the reproducibility has been brought to focus of the NLP research, with famous works like Pedersen (2008).

Typically, continuous development of rule-based NLP applications involves unexpected breakage. With regression tests for each error type in the grammar checker, regressions are caught quickly. This means that refactoring or larger changes to the code can be done without decreasing the overall quality of the grammar checker.

The main motivation behind introducing regression testing came from the need of automatizing the grammar checker evaluation. Manual evaluation to calculate precision and recall got rather cumbersome. This led to the development of a more powerful tool for testing grammar checking automatically (Wiechetek et al., 2019b), and there was parallel work and methodological in-depth study on corpus mark-up. Based on this work, we did not have to make a big leap to get regression testing. We reused the evaluation tool and turned it into a proper tester, with detailed statistics of the performance of the tool and sentence-by-sentence analysis that provides a basis for debugging.

2.3 The North Sámi grammar checker

The grammar checker for North Sámi (*GramDivvun*) performs both spell- and grammar checking – i.e. requiring full sentence analysis to identify local and global syntactic errors – in addition to punctuation and format checking. It includes a version of the open-source spelling checker that has been freely distributed since 2007[5], cf. also Gaup et al. (2006). It uses the HFST-based spelling mechanism described in Pirinen and Lindén (2014) for a number of modules, and in addition includes six Constraint Grammar modules, cf. Figure 1. These are:

- Two valency grammars applied before and after spellchecking (*valency.cg3* and *valency-postspell.cg3*)
- A tokenizer (*mwe-dis.cg3*)
- Two morpho-syntactic disambiguators applied before and after spellchecking (*grc-disambiguator.cg3* and *after-speller-disambiguator.cg3*)
- A module for more advanced grammar checking (*grammarchecker-release.cg3*)

The current version of the grammar checker module in *GramDivvun*[6] includes 313 error detection rules, 4 purely morpho-syntactic rule types, 17 morpho-syntactic rule types that are caused by general real-word rule types, 17 idiosyncratic real word error rule types, 14 punctuation or space error rule types and one spelling error rule type. A real word error is typically a misspelling, but unlike regular typos it results in (similar) real word rather than a non-word. Therefore, an analysis of the sentence is necessary to identify the error. In English language, *dessert* can be a real word error of *desert* and vice versa.

As in English, there are numerous idiosyncratic real word error types in North Sámi, made by native speakers for various reasons (i.e. dialectal phonetic differences that do not coincide with the written norm, vowel and consonant errors based on confusion of different forms, etc.) But some of these errors are more systematic, such as the confusion of case-marked (locative case) vs. attributive adjective forms. This is the case in ex. (1)[7], where the locative form *álkis* should be an attributive one, i.e. *álkes*, and the only distinction between these forms is the vowel - *e* vs. *i*.

(1) Snoranuohtti lea gehppes ja **álkis**
Danish seine be.3SG light and simple.LOC
veahkkeneavvu.
tool
'Danish seine is a light and simple tool.'

Instead of resulting in a simple non-word, in North Sámi vowel confusion can have grammatical

[5] http://divvun.no/korrektur/korrektur.html

[6] https://github.com/giellalt/lang-sme/releases/tag/naacl-2021-4

[7] All examples are original examples or fragments from *SIKOR* and are most likely native speaker texts or translations.

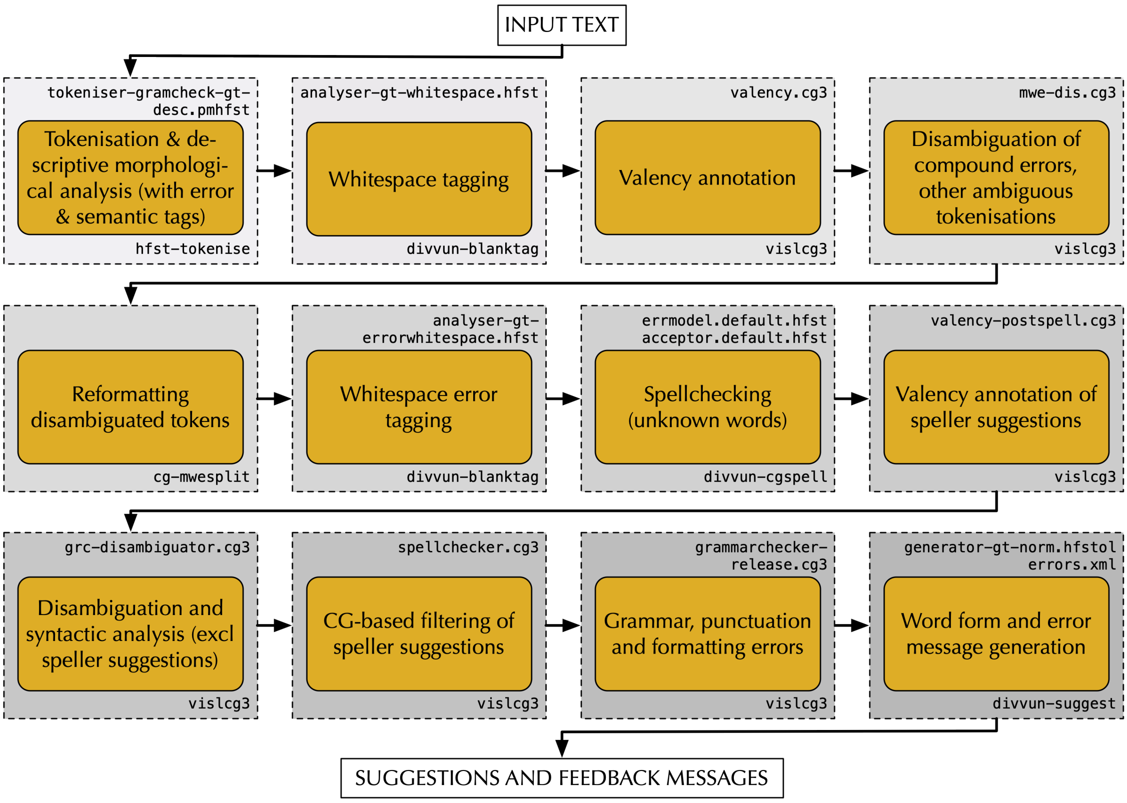

INPUT TEXT
tokeniser-gramcheck-gt-desc.pmhfst
Tokenisation & descriptive morphological analysis (with error & semantic tags)
hfst-tokenise
analyser-gt-whitespace.hfst
Whitespace tagging
divvun-blanktag
valency.cg3
Valency annotation
vislcg3
mwe-dis.cg3
Disambiguation of compound errors, other ambiguous tokenisations
vislcg3
Reformatting disambiguated tokens
cg-mwesplit
analyser-gt-errorwhitespace.hfst
Whitespace error tagging
divvun-blanktag
errmodel.default.hfst
acceptor.default.hfst
Spellchecking (unknown words)
divvun-cgspell
valency-postspell.cg3
Valency annotation of speller suggestions
vislcg3
grc-disambiguator.cg3
Disambiguation and syntactic analysis (excl speller suggestions)
vislcg3
spellchecker.cg3
CG-based filtering of speller suggestions
vislcg3
grammarchecker-release.cg3
Grammar, punctuation and formatting errors
vislcg3
generator-gt-norm.hfstol
errors.xml
Word form and error message generation
divvun-suggest
SUGGESTIONS AND FEEDBACK MESSAGES

Figure 1: System architecture of *GramDivvun*

consequences. That means that a certain grammatical form can be confused with another grammatical form of the same lemma. Since both forms regard the same lemma, these errors can be detected and corrected systematically. Apart from that, other (morpho-phonetic) criteria decide which forms are eligible for this error type. These are lemma endings (e.g. *-it*, *-at*, or *-ut*), number of syllables (even vs. uneven), and consonant gradation class membership.[8] Table 1 illustrates one of the consonant gradation classes with examples.

Nominal derivations of certain types of verbs (i.e. with a particular ending and a specific consonant gradation pattern In ex. (2), the vowel confusion (*u/o*) regards derived nouns (that should be past participle forms) from consonant gradation class 4D (cf. Table 1). Here, the (derived) noun *vákšun* '(the act of) observing' is confused with the past participle *vákšon* 'observed'.

(2)	Politiijat	leat	otne	**vákšun**	johtolaga	Guovdageainnus.
	police	be.3PL	today	observing.NOM	traffic	Guovdageaidnu.LOC

'The police has conducted a traffic control in Guovdageaidnu today.'

Consonant center		Example		Translation
kc	vcc	ci**kc**ut	ci**vcc**ui	'(to) pinch – s/he pinched'
kč	včč	go**kč**at	go**včč**at	'(to) cover – you cover'
ks	vss	oa**ks**i	oa**vss**it	'branch – branches'
kst	vstt	tea**kst**a	tea**vstt**at	'text – texts'
kš	všš	di**kš**ut	di**všš**un	'(to) take care – I take care'
kt	vtt	[…]		

Table 1: Consonant gradation group 4D according to Nickel (1994, p. 30)

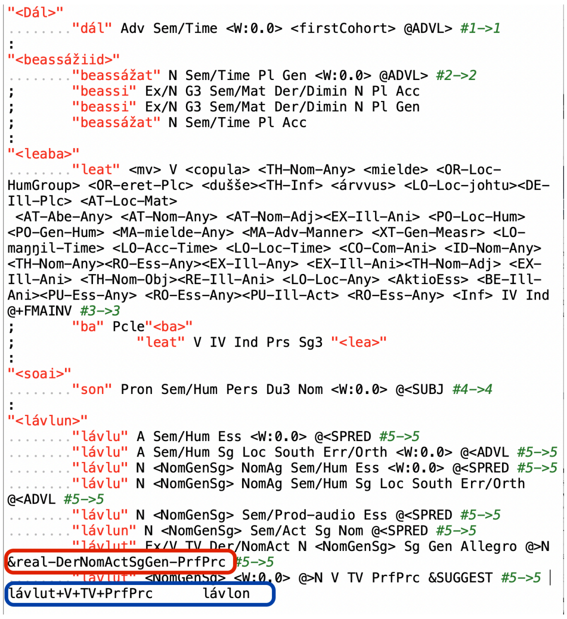
The complex structure of the grammar checker shows that there are modifications in many different modules that can be responsible for possible mishaps, since changes in one module can affect the input to subsequent modules.

The input for the grammar checker are unmarked sentences. The input for the regression tests are sentences with an error mark-up like in ex. (3).

[8] A number of Finno-Ugric languages use stem-internal morpho-phonological changes in addition to suffixes to mark case and other morphological processes. In North Sámi there are 123 consonant gradation patterns (Nickel, 1994, p.23-30)

(3) Dál beassážiid leaba soai
now Easter have they.DU
{lávlun}¢{lávlon} dáid sálmmaid
singer sing.PASTP these psalm.ACC.PL
girkuin Guovdageainnus,
church.LOC.PL Guovdageaidnu.LOC,
Kárášjogas ja Mázes.
Kárášjohka.LOC and Máze.LOC
'This Easter they have sung these psalms in the churches of Guovdageaidnu, Kárášjohka and Máze.'

Figure 2 shows the output for the grammar checker including error detection (red rectangle) and error correction (blue rectangle). The sentence is tokenized and reads from the top to the bottom. Word forms are in angle brackets, indented lines are homonymous analyses of each form, including lemmata, morphological, semantic and syntactic tags followed by numerical dependencies.



```
"<Dál>"
........"dál" Adv Sem/Time <W:0.0> <firstCohort> @ADVL> #1->1
:
"<beassážiid>"
........"beassážat" N Sem/Time Pl Gen <W:0.0> @ADVL> #2->2
;       "beassi" Ex/N G3 Sem/Mat Der/Dimin N Pl Acc
;       "beassi" Ex/N G3 Sem/Mat Der/Dimin N Pl Gen
;       "beassážat" N Sem/Time Pl Acc
:
"<leaba>"
........"leat" <mv> V <copula> <TH-Nom-Any> <mielde> <OR-Loc-
HumGroup> <OR-eret-Plc> <dušše><TH-Inf> <árvvus> <LO-Loc-johtu><DE-
Ill-Plc> <AT-Loc-Mat>
 <AT-Abe-Any> <AT-Nom-Any> <AT-Nom-Adj><EX-Ill-Ani> <PO-Loc-Hum>
<PO-Gen-Hum> <MA-mielde-Any> <MA-Adv-Manner> <XT-Gen-Measr> <LO-
maŋŋil-Time> <LO-Acc-Time> <LO-Loc-Time> <CO-Com-Ani> <ID-Nom-Any>
<TH-Nom-Any><RO-Ess-Any><EX-Ill-Any> <EX-Ill-Ani><TH-Nom-Adj> <EX-
Ill-Ani> <TH-Nom-Obj><RE-Ill-Ani> <LO-Loc-Any> <AktioEss> <BE-Ill-
Ani><PU-Ess-Any> <RO-Ess-Any><PU-Ill-Act> <RO-Ess-Any> <Inf> IV Ind
@+FMAINV #3->3
;       "ba" Pcle"<ba>"
;               "leat" V IV Ind Prs Sg3 "<lea>"
:
"<soai>"
........"son" Pron Sem/Hum Pers Du3 Nom <W:0.0> @<SUBJ #4->4
:
"<lávlun>"
........"lávlu" A Sem/Hum Ess <W:0.0> @<SPRED #5->5
........"lávlu" A Sem/Hum Sg Loc South Err/Orth <W:0.0> @<ADVL #5->5
........"lávlu" N <NomGenSg> NomAg Sem/Hum Ess <W:0.0> @<SPRED #5->5
........"lávlu" N <NomGenSg> NomAg Sem/Hum Sg Loc South Err/Orth
@<ADVL #5->5
........"lávlu" N <NomGenSg> Sem/Prod-audio Ess @<SPRED #5->5
........"lávlun" N <NomGenSg> Sem/Act Sg Nom @<SPRED #5->5
........"lávlut" Ex/V TV Der/NomAct N <NomGenSg> Sg Gen Allegro @>N
&real-DerNomActSgGen-PrfPrc #5->5
........"lávlut" <NomGenSg> <W:0.0> @>N V TV PrfPrc &SUGGEST #5->5 |
lávlut+V+TV+PrfPrc      lávlon
```

Figure 2: Output of *GramDivvun* in the command line

3 Regression testing for grammar checking

Regression testing for grammar checking is based on an error marked-up corpus. We have collected an error corpus of representative errors in *Yaml*-formatted[9] files specific to each error type. At the current date in august 2021, these include 17,800 sentences. Typically, each regression file contains several hundred sentences, some up to 4,300 sentences. There should be a balance of correct and erroneous sentences covering the same phenomena so that one can test for false positives and false negatives. Test sentences should cover a variety of syntactic contexts and pay attention to long-distance relationships between syntactic functions. They should include coordination, (inserted) subclauses, complex noun phrases, multiple adverbials, idiomatic constructions, multiple errors, punctuation, and other phenomena that can alter the status of the error/correct form. The collected errors are designed to cover a maximally large amount of real-world errors that people make when writing texts, in order to keep the grammar checker usable for people. The file naming is now error-specific,[10] but as they come from an authentic corpus, they can contain multiple errors per sentence including other types of errors and nested errors.

Yaml is a mark-up language with a simple syntax that makes writings of the tests convenient and co-operation with programmers and linguists easier. We chose to use the Yaml format for grammar testing because of positive experiences with the use of the same format for spell checker testing.[11] The original test framework for morphology testing initiated by Brendan Molloy can be found on GitHub.[12]

The regression test script measures both error detection and error correction and whether they match the manual error mark-up. False negatives of the type fn_1 are correctly detected errors that do not receive any corrections by the grammar checker. False negatives of the type fn_2 are undetected errors. The same goes for false positives, where: fp_1 are correctly detected errors with a wrong correction, and fp_2 are error detections that are not manually marked up. True positives (tp), on the other hand, are detected and corrected errors that match with the manual mark-up. In our final evaluation, we will not distinguish between these and only take into account successful vs. unsuccessful error correction in terms of false negatives and true/false positives. The tester script is implemented in Python and can be downloaded from GitHub[13].

[9] https://yaml.org/spec/1.2/spec.html

[10] current examples: https://github.com/giellalt/lang-sme/tree/main/tools/grammarcheckers/tests

[11] https://giellalt.uit.no/infra/infraremake/AddingMorphologicalTestData.html#Yaml+tests

[12] https://github.com/apertium/apertium-tgl-ceb/blob/master/dev/verbs/HfstTester.py

[13] https://github.com/giellalt/giella-core/blob/master/scripts/gramcheck-test.py

The grammar checker makes a list of each error that consists of the erroneous word, the position of the error (start and end), a list of suggestions and error type. The error mark-up is then converted to the same structure so that manual and grammar checker mark-up can be compared. For each of these test sentences, three things are collected: the erroneous version of the error marked-up sentence, the error marked-up version of the errors in the sentence and the errors detected by the sending the erroneous sentence through the grammar checker. The tester prints the outcome of each of the tests in a detailed manner, sentence by sentence and with references to the particular error types involved. The final report contains the number of total passes, fails, true and false positives/negatives, precision, recall and F_1-score. On exit, the script returns 0 or 1, 0 meaning all tests succeeded, 1 otherwise.

The test script is fast and light-weight enough to be part of a CI/CD system, even with processor time and RAM limitation, e.g. testing 300 sentences on the developers' machines takes about 30 seconds.

The error mark-up formalism has earlier been used to automatize spellchecking for Greenlandic, Icelandic, North, Lule and South Sami.

The error mark-up follows a number of guidelines[14] based on earlier corpus mark-up (Moshagen, 2014) and applies eight different general error types, each of them marked by a different sign: orthographic, real word, morpho-syntactic, syntactic, lexical, formatting, foreign language, and unclassified errors. The error is enclosed in curly brackets, followed by its correction in another set of curly brackets. The second curly bracket may or may not include a part of speech, morpho-syntactic criteria and a subclassification of the error type.

Orthographic errors (marked by $) include non-words only. They are traditional misspellings confined to single (error) strings, and the traditional speller should detect them. *Real word errors* (marked by ¢) are misspellings that cannot be detected by a traditional speller, they are an analysis of the surrounding words. *Morpho-syntactic errors* (marked by £) are case, agreement, tense, mode errors. They require an analysis of (parts of) the sentence or surrounding words to be detected. *Syntactic errors* (marked by ¥) require a partial or full analysis of (parts of) the sentence or surrounding words. They include word order errors, compound errors, missing words, and redundant words. *Lexical errors* (marked by €) include wrong derivations. *Foreign language* (marked by ∞) includes words in other languages that do not require a correction. *Formatting errors* (marked by ‰) include spacing errors in combination with punctuation. Unclassified errors are marked with §.

In ex. (4), the tokens involved in the error are nouns, the syntactic error is a missing word and the correction is adding the subjunction *ahte* 'that'.

(4) Illá {jáhkken}¥{missing|jáhkken ahte}
hardly think.PAST.1SG
lei duohta.
be.PAST.3Sg true
'I hardly thought that it was true.'

Regarding the span of an error, we typically mark as little as possible, even if larger parts of the sentence are responsible for the identification of the error. This is done to facilitate matching error mark-up with grammar checker marking of the error, and it has direct effect on automatic evaluation. Most of the frameworks we use to process language material in context, e.g. Constraint Grammar takes a token-based approach to language processing, and therefore marking several words can get cumbersome and should be avoided if possible.

Ex. (5) shows the mark-up of nested errors. There is both a morpho-syntactic error, the case of *linjá* 'line' should be accusative instead of nominative, and a compound error, *njuolggo* and *linjjá* should be written as one word.

(5) Sárggo {**njuolggo**
draw.IMPRT.2SG straight
{linjá}£{noun,obj,accsg,nomsg,case|linjjá}}
line
¥{noun,cmp|njuolggolinjjá} dán guovtti
(straightline) these two
čuoggá gaskka.
points between.
'Draw a straight line between these two points.'

4 Evaluation

We performed two measurements of the system quality: firstly we have the well-curated and targeted regression test suite that is summarized in Table 2. Secondly, we measure an overview of how the system fares for texts in the whole corpora in the wild in Table 3. The first test suite verifies our system's quality in the regression test sense, and the second test ensures that the system works for open text case.

[14] https://giellalt.uit.no/proof/spelling/testdoc/error-markup.html

	naacl-1	**naacl-2** **baseline**	**naacl-4**
Precision	70.9%	68.9%	88.8%
Recall	66.9%	84.0%	91.0%
F_1**-score**	68.8	75.7	89.9

Table 2: Evaluation results from the regression tests.

4.1 Quantitative evaluation

In Table 2 we show the results of the regression tests at the same three stages of the development. We measure the success percentage in terms of the number of the tests passed from the overall tests. The regression test corpus we use is a set of tests selected to have a representative coverage of the various error types and contexts. With the carefully selected grammar tests we can control the quality of the overall system, the overall aim for these grammar tests is to keep the correctness at 100 %. The correctness measure C here is $C = \frac{\text{tp}}{\text{CS}}$ where CS is the corpus size.

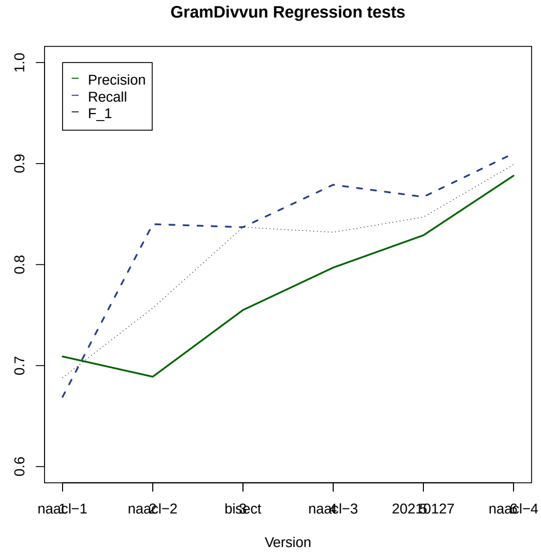


Figure 3: Development of *GramDivvun* precision and recall in the regression tests

In Table 3, we show the overall performance of *GramDivvun* at three stages over the course of approximately one and a half years of continuous development. This means that all grammatical errors are included, also the ones that the grammar checker does not have any module for yet. The tests are done on an error marked-up evaluation-corpus of approx. 26,000 words. The first test is made with the North Sámi grammar checker from 2019-11-21[15] before the introduction of the Yaml-tests (naacl-1). The second test uses the version from 2020-11-20[16] (naacl-2 - Yaml baseline) from when we had first introduced the regression tests. The third test uses the North Sámi grammar checker from 2021-03-20[17] (naacl-4) where we have taken into account results from the regression tests in the form of general rule changes.

The results show that the overall performance of the grammar checker on a small error marked-up corpus improves only slightly. This is due to the frequency of the errors we worked on. The corpus to test these error types in particular needs to be substantially bigger to show a change in performance. However, especially recall has improved by 6% showing an increased coverage of the error types covered in the grammar checker.

Figure 3 shows a number of stages of the performance of the grammar checker after developing regression tests. There was a significant drop in precision (naacl-2) and a number of drops in recall (bisect).[18] These coincided with the addition of test sentences (the regression tests grew from a couple of sentences to larger corpora of several thousand sentences), introducing new contexts that required stricter rules. Stricter rules typically lower recall to ensure stable precision. New, more specific rules need to be introduced to get recall up again. This explains the ups and downs in the graph. After the introduction of Yaml tests, however, we can see that precision has steadily been going up, and by that proves the main objective of regression tests right.

4.2 Qualitative evaluation

One can generally see, that rule types that have been prioritized in the grammar checker improved after involving regression testing.

Precision got better in ex. (6), where the nominalization *dovdan* 'feeling' is confused with the first-person singular form *dovddan* 'I know', forms that are distinguished by a change in the consonant centre only.

(6)	Buohkat,	geaid	**dovdan**,	oaivvildit
	All,	who.ACC.PL	feeling,	think.3PL

[15] https://github.com/giellalt/lang-sme/releases/tag/naacl-2021-1

[16] https://github.com/giellalt/lang-sme/releases/tag/naacl-2021-2

[17] https://github.com/giellalt/lang-sme/releases/tag/naacl-2021-4

[18] https://github.com/giellalt/lang-sme/commit/216d00d37bff6ebbd34a1529eb822b61b50a3f08

	naacl-1	naacl-2 baseline	naacl-4
Precision	80.0%	75.3%	82.3%
Recall	59.8%	65.9%	65.1%
F_1**-score**	68.4	70.2	72.7
TP	391	439	430
FP	98	144	92
FN	263	227	231

Table 3: Performance of *GramDivvun* over the span of a year, before and after introducing regression tests

seamma:
same
'Everybody I know thinks the same'

In ex. (7), *GramDivvun* finds the locative adjective form *oktageardánis*, which by analogy is confused with the nominative form *oktageardán*.

(7) Skuvllas berreše maiddái leat
school.LOC should.COND.3PL also be
oktageardánis
simple
'The school should also have simple'

A number of error type rules are causing false positives in certain contexts such as ex. (8), where the infinitive *oastit* 'buy' is a correct form. However, it is homonymous with a second-person plural imperative reading of the same verb, and is falsely corrected to the third-person plural reading *ostet*.

(8) Golut **oastit** dárbbašlaš girjjiid
expenditure.PL buy necessary book.ACC.PL
čađahit prošeavtta.
carry.through project.ACC
'Expenditure to buy necessary books to carry through the project.'

Some errors that are dealt with in the grammar checker are not recognized in certain syntactic contexts, such as the compound error *guovddáš doaimmat* that should be written as one word in ex. (9).

(9) Movttiidahttin ja bagadeapmi leat
motivation and instruction be.3PL
SOR:a prošeakta-jođiheaddji deatalaš ja
SOR.GEN project-leader important and
guovddáš doaimmat.
central task.PL
'Motivation and instruction are important and central tasks for SOR's project leader'

In addition, there are error types that the grammar checker does not deal with at all, which is why they are not recognized, and the result are false negatives. This is the case of the syntactic error ex. (10), where the subjunction *vai* 'so that' before the finite verb *beassaba* 'get to' is missing.

(10) Máŋgii vahkkui viežžá son vierrobeatnaga
often week.ILL fetch.3SG s/he foreign.dog
beassaba vázzit.
get.to3DU walk
'Many times a week she fetches the foreign dog so that they get to walk.'

5 Discussion and future outlook

In this paper we have shown that regression testing is necessary to provide reliable results (i.e. in particular a stable precision) for the users of higher level NLP applications like grammar checkers. A rule-based approach is successful for applications like grammar checking which require a high level of systematicity and reliable results. For low-resourced languages, where availability of resources such as expert-curated error-correction corpora are scarce, the development of rule-based tools is the most efficient approach. We showed that by using comprehensive regression testings we can keep developing the grammar checking and correction on a day-to-day basis and provide the end users with the newest updates without worrying about their quality. In the future we would like to see if it is possible to gather enough resources for a neural network based grammar checking and correction. Regression testing of the kind we described is applicable for neural network approaches as well. However, neural network systems do not allow for specific adjustments within the error types, which is rather a weakness of the system itself. It is therefore natural to apply these regression tests for neural network models as well, and we expect that the system will work in conjunction to neural network without any major changes.

We have started with neural network-approaches (forthcoming) for the correction of certain error types from our rule-based grammar checker. These require a preparation of the data by means of our existing rule-based tools, both for part-of-speech tagging and marking up error data.

One of the interesting features of a rule-based system, that has been brought to focus on the NLP community recently, is the energy-footprint of the used models. In case of our models, the rules can be compiled into finite-state automata on an average consumer desktop within minutes, and the actual models can be run on low-end mobile devices, so the energy footprint is trivially multiple orders of magnitude lower than that of any neural language models.

Acknowledgements

We thank Ritva Nystad for marking up part of the evaluation corpus, testing *GramDivvun* and contributing to critical discussions about grammatical errors. We also thank Sjur Nørstebø Moshagen and Brendan Molloy for the initial test setup.

References

Kenneth R Beesley and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI publications.

Miriam Butt and Tracy Holloway King. 2003. *Grammar Writing, Testing, and Evaluation*, pages 129–179. CSLI Publications, Stanford.

Daiga Deksne. 2019. Bidirectional lstm tagger for latvian grammatical error detection. In *Ekštein K. (eds) Text, Speech, and Dialogue. TSD 2019. Lecture Notes in Computer Science, vol 11697. Springer.*

Elaine Farrow and Myroslava O. Dzikovska. 2009. Context-dependent regression testing for natural language processing. In *Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009)*, pages 5–13, Boulder, Colorado. Association for Computational Linguistics.

Børre Gaup, Sjur Moshagen, Thomas Omma, Maaren Palismaa, Tomi Pieski, and Trond Trosterud. 2006. From Xerox to Aspell: A first prototype of a north sámi speller based on twol technology. In *Finite-State Methods and Natural Language Processing*, pages 306–307, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ralph F Grove. 2009. *Web Based Application Development*. Jones & Bartlett Publishers.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing unrestricted text. In *Proceedings of the 13th International Conference of Computational Linguistics*, volume 3, pages 168–173, Helsinki.

Sjur Moshagen. 2014. Test data and testing of spelling checkers. Presentation at the NorWEST2014 workshop.

Sjur Moshagen, Jack Rueter, Tommi Pirinen, Trond Trosterud, and Francis M Tyers. 2014. Open-source infrastructures for collaborative work on under-resourced languages. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC*, pages 71–77.

Klaus Peter Nickel. 1994. *Samisk grammatikk*, second edition. Davvi Girji, Kárášjohka.

Ted Pedersen. 2008. Empiricism is not a matter of faith. *Computational Linguistics*, 34:465–470.

Tommi A. Pirinen and Krister Lindén. 2014. State-of-the-art in weighted finite-state spell-checking. In *Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing - Volume 8404*, CICLing 2014, pages 519–532, Berlin, Heidelberg. Springer-Verlag.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of russian. In *Transactions of the Association for Computational Linguistics, vol. 7, pp. 1–17, 2019.*

Gary F. Simons and Charles D. Fennig, editors. 2018. *Ethnologue: Languages of the World*, twenty-first edition. SIL International, Dallas, Texas.

Linda Wiechetek. 2017. *When grammar can't be trusted – Valency and semantic categories in North Sámi syntactic analysis and error detection*. PhD thesis, UiT The Arctic University of Norway.

Linda Wiechetek, Sjur Nørstebø Moshagen, Børre Gaup, and Thomas Omma. 2019a. Many shades of grammar checking – launching a constraint grammar tool for north sámi. In *Proceedings of the NoDaLiDa 2019 Workshop on Constraint Grammar - Methods, Tools and Applications*, NEALT Proceedings Series 33:8, pages 35–44.

Linda Wiechetek, Sjur Nørstebø Moshagen, Børre Gaup, and Thomas Omma. 2019b. Many shades of grammar checking - Launching a Constraint Grammar tool for North Sámi. In *Proceedings of the NoDaLiDa 2019 Workshop on Constraint Grammar - Methods, Tools and Applications (NoDaLiDa 2019)*, pages 35–44.

Linda Wiechetek, Tommi A Pirinen, Mika Hämäläinen, and Chiara Argese. 2021. Rules ruling neural networks - how can rule-based and neural models benefit from each other when building a grammar checker? In *forthcoming*.

Low-Resource ASR with an Augmented Language Model

Timofey Arkhangelskiy
Universität Hamburg
timarkh@gmail.com

Abstract

It is widely known that a good language model (LM) can dramatically improve the quality of automatic speech recognition (ASR). However, when dealing with a low-resource language, it is often the case that not only aligned audio data is scarce, but there are also not enough texts to train a good LM. This is the case of Beserman, an unwritten dialect of Udmurt (Uralic > Permic). With about 10 hours of aligned audio and about 164K words of texts available for training, the word error rate of a Deepspeech model with the best set of parameters equals 56.4%. However, there are other linguistic resources available for Beserman, namely a bilingual Beserman-Russian dictionary and a rule-based morphological analyzer. The goal of this paper is to explore whether and how these additional resources can be exploited to improve the ASR quality. Specifically, I attempt to use them in order to expand the existing LM by generating a large number of fake sentences that in some way look like genuine Beserman text. It turns out that a sophisticated enough augmented LM generator can indeed improve the ASR quality. Nevertheless, the improvement is far from dramatic, with about 5% decrease in word error rate (WER) and 2% decrease in character error rate (CER).

Abstract

Ваньзылы тодмо, умой лэсьтэм кыл модель вераськемез асэрказ тодманлэсь ӟечлыксэ трослы будэтыны быгатэ шуыса. Озьы ке но, куке вераськон мынэ пичи кылъёс сярысь, кызьы ке распознавателез дышетон понна волятэм куара, озьы ик умой кыл моделез дышетон понна текстъёс ӵемысь туж ӧжыт луо. Ӵапак таӵе югдур удмурт кыллэн гожъяськеттэм бесерман вераськетэныз кылдэмын. Ки уламы вань 10 час пала волятэм но расшифровать карем куара но 164 сюрс пала уже кутэм кылъёсын текстъёс. Та тодэтъёсын дышетскыса, Deepspeech система возьматэ 56,4% WER (мыдлань распознать карем кылъёслэн процентсы). Озьы ке но бесерман вераськетъя вань на мукет кылтодон ванёсъёс: бесерман-ӟуч кыллюкам но шонеррадъян морфологи анализатор. Та ужлэн целез — валаны, луэ-а вераськемез асэрказ тодманлэсь ӟечлыксэ будэтон понна та ватсам ресурсъёсты уже кутыны. Кылсярысь, соос вылэ пыкъяськыса, турттэмын кыл моделез паськытатыны, со понна кылдытэмын вал трос зэмос луисьтэм шуосъёс, кудъёсыз куд-ог ласянь тупало зэмос бесерман текстлы. Шуосъёсты кылдытӥсь генератор тырмыт «визьмо» ке, распознаванилэн ӟечлыкез зэмзэ но будэ вылэм. Озьы ке но та умоян шӧдскымон луэ шуыса, вераны уг луы: WER возьматон усе 5%-лы пала, нош CER (мыдлань распознать карем букваослэн процентсы) — 2%-лы.

Abstract

Известно, что хорошая языковая модель может существенно повысить качество автоматического распознавания речи. Однако если речь идёт о некрупном языке, зачастую имеется не только слишком мало выровненного звука для

Seventh international workshop on computational linguistics of Uralic languages, pages 47–53
September 23–24, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-1-954085-82-4_006

обучения распознавателя, но и слишком мало текстов для обучения хорошей языковой модели. Именно таков случай бесермянского – бесписьменного диалекта удмуртского языка. В нашем распоряжении имеются около 10 часов звука, выровненного с расшифровками, и тексты объёмом около 164 тыс. словоупотреблений. Обучившись на этих данных, система Deepspeech демонстрирует WER (процент неправильно распознанных слов), равный 56,4%. Однако для бесермянского существуют другие лингвистические ресурсы, а именно бесермянско-русский словарь и правиловый морфологический анализатор. Цель этой работы – выяснить, можно ли использовать эти дополнительные ресурсы для улучшения распознавания речи. В частности, предпринимается попытка расширить с их помощью языковую модель путём порождения большого количества ненастоящих предложений, которые в некоторых отношениях похожи на настоящий бесермянский текст. Оказывается, что если генератор предложений достаточно “умён”, качество распознавания после этого действительно возрастает. Однако это улучшение вряд ли можно назвать существенным: показатель WER падает примерно на 5%, а CER (процент неправильно распознанных букв) – на 2%.

1 Introduction

The key to reaching good ASR quality is having lots of data, i.e. thousands or at least hundreds of hours of text-aligned sound recordings. For most languages in the world, however, resources of that size are unavailable. With only a dozen hours of sound at hand, it is currently impossible to reach a WER low enough for the system to be usable in real-world applications. Nevertheless, a system with a WER, which is high, but lower than a certain threshold (e.g. 50%), could still be used in practice. Specifically, the primary motivation behind this research was the need to transcribe large amounts of spoken Beserman for subsequent linguistic research. If an ASR system, despite its high WER, could facilitate and accelerate manual transcription, that would be a useful practical application, even if limited in scope. Other possible applications of such undertrained noisy ASR systems have been proposed by Tyers and Meyer (2021). This is why it makes sense to experiment with datasets that small.

A number of techniques have been used to achieve better results in low-resource ASR systems. This includes pre-training the model on the data from another (possibly related or phonologically similar) language (Stoian et al., 2020), augmenting the sound data with label-preserving transformations (Tüske et al., 2014; Park et al., 2019), and training the LM on a larger set of texts taken e.g. from a written corpus (Leinonen et al., 2018). That a good language model can play an important role can be seen e.g. from the experiments on ASR for varieties of Komi, a language closely related to Udmurt, as described by (Hjortnaes et al., 2020b) and (Hjortnaes et al., 2020a). Replacing a LM with a larger and more suitable one (in terms of domain) can decrease WER significantly.

Beserman is traditionally classified as a dialect of Udmurt (Uralic > Permic) and is spoken by around 2200 people in NW Udmurtia, Russia. Unlike standard Udmurt, it lacks a codified orthography and is not used in the written form outside of scientific publications. This paper describes experiments with training Deepspeech (Hannun et al., 2014) on transcribed and elicited Beserman data. I am particularly interested in augmenting the LM with the help of linguistic resources that exist for Beserman: a Beserman-Russian dictionary and a morphological analyzer. The former is used, among other things, to transfer information from a model trained on Russian data. Same kinds of data augmentation could be relevant for many other under-resourced languages and dialects, since bilingual dictionaries and rule-based tools often exist for varieties, which are poor in raw data.

The paper is organized as follows. In Section 2, I describe the dataset and lay out the reasons why improving the LM could be challenging. In Section 3, the training setup is outlined. In Section 4, I describe how the artificially augmented LM was generated. In 5, the original results are compared to that of the augmented LM. This is followed by a conclusion.

2 The data

The Beserman dataset I have at hand consists of about 15,000 transcribed sound files with recordings from 10 speakers, both male and female, totaling about 10 hours (with almost no trailing silence). Most of them come from a sound-aligned Beserman corpus, whose recordings were made in 2012–2019 and have varying quality. Another 2,700 files, totaling 2.5 hours, come from a sound dictionary and contain three pronunciations of a headword each. The duration of most files lies between 1 and 5 seconds. In addition to the texts of the sound-aligned corpus, there are transcriptions of older recordings, which are not sound-aligned as of now, and a corpus of usage examples based on the Beserman-Russian dictionary[1] (Arkhangelskiy, 2019). All these sources combined contain about 27,400 written Beserman sentences (some very short, some occurring more than once), with a total of 164K words.

Such amount of textual data is insufficient for producing a well performing LM. Since Beserman is a morphologically rich language, most forms of most lexemes are absent from the sample and thus cannot be recognized, being out-of-vocabulary words. Unlike in some other studies mentioned above, it is hardly possible to find Beserman texts elsewhere. One way of doing that would be to use texts in literary Udmurt, which are available in larger quantities (tens of millions of words). Although I have not explored that option yet[2], I doubt it could have the desired effect because the available Udmurt texts belong to a completely different domain. While most Beserman texts are narratives about the past or the life in the village, or everyday dialogues, most Udmurt texts available in digital form come from mass media. There is a pronounced difference between the vocabularies and grammatical constructions used in these two domains.

Instead, I attempt to utilize linguistic resources available for Beserman: a Beserman-Russian dictionary comprising about 6,000 entries and a morphological analyzer. The latter is rule-based and is based on the dictionary itself. Apart from the information necessary for morphological analysis, it contains some grammatical tags, such as animacy for nouns and transitivity for verbs. The analyzer recognizes about 97% of words in the textual part of the Beserman dataset. A small set of Constraint Grammar rules (Karlsson, 1990; Bick and Didriksen, 2015) is applied after the analysis, which reduces the average ambiguity to 1.25 analyses per analyzed word.

The idea is to inflate the text corpus used to produce the LM by generating a large number of fake sentences, using real corpus sentences as the starting point and the source of lemma frequencies, and incorporating data from the linguistic resources in the process.

3 Deepspeech training

All Beserman texts were encoded in a version of the Uralic Phonetic Alphabet so that each Unicode character represents one phoneme. Although there are a couple of regular phonetic processes not reflected in the transcription, such as optional final devoicing or regressive voicing of certain consonants, the characters almost always correspond to actual sounds. Therefore, CER values reported below must closely resemble PER (phone error rates)[3]. All sound files were transformed into 16 KHz, single-channel format.

Deepspeech architecture (Hannun et al., 2014) (Mozilla implementation[4]) was used for training. This involves training a 5-layer neural network with one unidirectional LSTM layer. After each epoch, the quality is checked against a development dataset not used in training. After the training is complete, the evaluation is performed on the test dataset. The train/development/test split was randomly created once and did not change during the experiments. The development dataset contains 1737 sentences; the test dataset, 267 sentences. No sound dictionary examples were included in either development or test datasets, otherwise their unnaturally high quality would lead to overly optimistic WER and CER values. It has to be pointed out though that the training dataset contains data from all speakers of the test dataset. This is in line with the primary usage scenario I had in mind, i.e. pretranscription of field data, because most untranscribed recordings in my collection are generated by the same speakers. However, for a real-world scenario where the set of potential speakers is unlimited, this setting would

[1] Available for search at http://beserman.ru; a large part of it has been published as Usacheva et al. (2017).

[2] There are certain phonological, morphological and lexical differences between the standard language and the Beserman dialect. Before an Udmurt model can be used in Beserman ASR, the texts should be "translated" into Beserman. Although such attempts have been made (Miller, 2017), making the translations look Beserman enough would require quite a lot of effort.

[3] This property is the reason why UPA rather than Udmurt Cyrillic script was used for encoding. Otherwise, the choice of encoding is hardly important because UPA can be converted to Cyrillics and vice versa.

[4] https://github.com/mozilla/DeepSpeech

produce an overly optimistic estimate. No transfer learning was applied.

A number of hyperparameter values were tested: learning rate between 0.00005 and 0.001, dropout rate 0.3 or 0.4, training batch size between 16 and 36. These value ranges have been demonstrated to yield optimal results on a dataset of similar size by (Agarwal and Zesch, 2019). The results did not depend in any significant way on these values, except for almost immediate overfitting when the learning rate was close to 0.001. Under all these settings, the training ran for 8 or 9 epochs, after which the loss on the development dataset started rising due to overfitting. The model used for the evaluation was trained with the following parameters: learning rate 0.0002, droupout rate 0.4, training batch size 24.

The output of the trained Deepspeech model is filtered using kenlm, an n-gram language model (Heafield, 2011). 3-gram and 4-gram models were tried, with no substantial difference; the figures below refer to the 4-gram models. When using the model with Deepspeech, there are two adjustable parameters, α and β. α (between 0 and 1) is the LM weight: higher values make the filter pay more attention to the n-gram probabilities provided by the model. β defines the penalty for having too many words: higher values increase the average number of words in transcribed sentences and decrease the average length of a word. A number of α and β combinations were tested (see below).

4 Augmented language model

As could be immediately seen from the test results, at least one of the reasons why the automatic transcription was wrong in many cases is that the corpus used to train the LM simply lacked the forms. Since Beserman is morphologically rich, a corpus of 164K words will inevitably lack most forms of most lexemes. Thankfully, this gap can be filled relatively easily, since Beserman morphological analyzer and dictionary can be turned into a morphological generator. (Another option, not explored here, would be to use subwords instead of words (Leinonen et al., 2018; Egorova and Burget, 2018).) However, if one just generated all possible word forms and added them to the corpus packed into random sentences, that would completely skew the occurrence and co-occurrence probabilities of forms, which would lead to even worse performance. The real trick would be to add the lacking forms without losing too much information from the original model, i.e. without significantly distorting the probabilities. Specifically, one would need to make the following values as close to the original ones as possible:

- relative frequencies of lemmata;
- relative frequencies of affix combinations, such as “genitive plural”;
- constraints on co-occurrence of certain grammatical forms (e.g. “verb in the first person is not expected after a second-person pronoun”);
- lexical constraints on contexts (e.g. “mother eats apples“ should be fine, while “apple eat mother“ should not).

Of course, traditional word-based text generation models strive to achieve exactly that. However, they could hardly be applied here because the objective of correctly generating a lot of previously unseen forms would be missed. Instead, I developed a sentence generator that utilizes not only the texts, but also the linguistic resources available for Beserman.

After a series of sequential improvements, the resulting sentence generator works as follows.

First, the sentences from the Beserman corpora are morphologically analyzed and turned into sentence templates. In a template, content words (nouns, verbs, adjectives, adverbs and numerals) are replaced with “slots”, while the rest (pronouns, postpositions etc.) are left untouched. The idea is that the lemma in a slot can be replaced by another lemma with similar characteristics, while the remaining words should not be replaced with anything else. Certain high-frequency or irregular verbs or adverbs are also not turned into slots, e.g. negative verbs or discourse clitics. Templates where less than one-third of the elements were turned into slots, or that contain fewer than three words, are discarded.

A slot contains the inflectional affixes the word used to have, its tags (e.g. “N,anim” for “animate noun”), as well as the original lemma.

Second, the data from the grammatical dictionary of the analyzer is processed. For each item, its lemma, stem(s) and tags (part of speech among them) is loaded. A global frequency dictionary is created. If a lemma is present in the corpora, its total number of occurrences is stored as its frequency; for the remainder of the lemmata, the frequency is set to 1.

Third, semantic similarity matrices are created for nouns, verbs and adjectives separately. The semantic similarities are induced from the Russian translations the lemmata have in the Beserman-Russian dictionary. Each translation is stripped of usage notes in brackets and parentheses and of one-letter words. After that, the first word of the remaining string is taken as the Russian equivalent of the Beserman word. The similarities between Russian translations are then calculated with an embedding model `ruwikiruscorpora_upos_skipgram_300_2_2019`[5] trained on the data of Russian National Corpus and Russian Wikipedia (Kutuzov and Kuzmenko, 2017). The resulting pairwise similarities are then condensed into a JSON file where each Beserman lemma contains its closest semantic neighbors together with the corresponding similarity value. The similarity threshold of 0.39 was set to only keep lemmata which are sufficiently similar to the lemma in question in terms of their distribution. After that, an average lemma contains about 66 semantic neighbors.

After these preparatory steps, the sentence generation starts. A template is chosen at random, after which each slot is filled with a word. If a slot contains multiple ambiguous analyses, one of them is chosen at random with equal probability, apart from several manually defined cases where one of the analyses is much more probable than the others. The original lemma of the slot is looked up in the list of semantic neighbors. If found, its semantic neighbors are used as its possible substitutes. Neighbors whose tags differ from the slot tags (e.g. inanimate nouns instead of animate) are filtered out. A random similarity threshold is chosen, which can further narrow down the list of substitutes. This way, more similar lemmata have a higher chance of ending up on the list of potential substitutes. When the list is ready, a lemma is chosen at random, with probability of each lemma proportional to its frequency in the global frequency list. Its stem is combined with the inflectional affixes in the slot, taking certain morphophonological alternations into account. The resulting word is added to the sentence. Template elements that are not slots are generally used as is, but words from a certain manually defined list can be omitted with a probability of 0.2 (this mostly includes discourse particles).

The sentences generated this way do not always make sense, but many of them at least are not completely ungrammatical, and some actually sound quite acceptable.

5 Results and comparison

I did not check how the size of the training dataset affects the quality of the model. However, it is interesting to note that the addition of 2.5 hours of triple headword pronunciations from the sound dictionary apparently did not add to the quality. The results were almost the same when they were omitted from the training set.

As already mentioned in Section 3, the output of a trained Deepspeech model is filtered with an n-gram model trained on a text corpus, with parameters α and β. I evaluated the model on the test dataset with three kenlm models: based only on the real Beserman sentences (`base`), and two augmented models trained on real and generated sentences (`gen`). The first augmented model was trained on 2M additional sentences (about 170K word types), the second, on 10M additional sentences (about 300K word types). The difference between the two augmented models was almost nonexistent. The larger model performed slightly better than the smaller for most parameter values, except in the case of $\alpha = 1.0$; the difference in WER in most cases did not exceed 0.5%. The figures below are given for the larger model.

The following α values were tested: 1.0, 0.9, 0.75, 0.6, 0.4. The values of β between 1.0 and 7.0 with the step of 1 were tested. The WER values for $\beta \geq 5.0$ were always worse than with lower β values and are not represented below.

One can see that the values obtained with the augmented model are better than the baseline across the board, so the sentence generation has had a positive effect on ASR quality. Also, the augmented model tolerates larger β values, whereas the baseline model starts producing too much short words in place of longer words absent from its vocabulary in that case. Nevertheless, the difference is not that large: the best `gen` value, 51.4, is lower than the best `base` value, 56.4, only by 5%. The difference in CER is even less pronounced:

A more in-depth analysis of the data reveals that the effect of LM augmentation is most visible on longer sound files. If only tested on sentences whose ground-truth transcription contained at least 6 words, the best WER value for `gen` equals 52.1, as

[5] https://rusvectores.org/en/models/

	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$
$\alpha = 1.0$	57.3 / 55.6	56.5 / 54.1	57.7 / 52.7	58.8 / 52.4
$\alpha = 0.9$	57.0 / 53.3	56.8 / 52.7	58.4 / 51.6	59.6 / 53.5
$\alpha = 0.75$	56.4 / 52.9	57.2 / 51.9	58.7 / **51.4**	60.9 / 53.4
$\alpha = 0.6$	57.1 / 52.9	58.9 / 51.9	60.4 / 53.8	64.9 / 56.3
$\alpha = 0.4$	59.6 / 55.5	62.3 / 56.4	67.0 / 59.8	75.4 / 66.1

Table 1: WER for `base` (before slash) and `gen` (after slash) models with different α and β values.

	$\beta = 1$	$\beta = 2$	$\beta = 3$	$\beta = 4$
$\alpha = 1.0$	35.0 / 34.4	33.9 / 33.3	33.3 / 32.0	32.8 / 30.9
$\alpha = 0.9$	34.0 / 32.5	33.4 / 32.1	33.0 / 30.7	32.7 / 30.2
$\alpha = 0.75$	32.9 / 31.5	32.4 / 30.4	32.0 / 29.7	31.9 / **29.2**
$\alpha = 0.6$	32.2 / 30.1	31.5 / 29.9	31.5 / 29.6	31.7 / 29.3
$\alpha = 0.4$	31.6 / 30.1	31.3 / 29.6	31.3 / 30.3	31.8 / 30.8

Table 2: CER for `base` (before slash) and `gen` (after slash) models with different α and β values.

opposed to only 59.5 for `base`. On short files, however, the added benefit of having plausibly looking n-grams in the corpus stops playing any role. For sentences (or, rather, sentence fragments) that contained at most 3 words, the best WER value for `gen` equals 60.5, compared to 61.5 for `base`.

As we can see, the LM augmentation did improve the ASR quality, even if marginally. The most important takeaway from this experiment, however, was that using a bilingual dictionary and a Russian model for approximating semantic similarity was a crucial part of the LM augmentation. Without that step, the generated LM did not visibly differ from `base`, even when lemma frequencies and tags were taken into account.

Since, to the best of my knowledge, no Deepspeech (or any other) ASR models existed for standard Udmurt when the experiments were conducted, it was impossible to compare ASR quality for Beserman and standard Udmurt.

6 Conclusion

There is a famous statement by Frederick Jelinek, made exactly in the context of ASR development, “Whenever I fire a linguist our system performance improves”. Indeed, contemporary ASR is largely an engineering enterprise and relies on algorithms and large amounts of data rather than on any linguistic insights. Still, if there is not enough data, can linguistic resources – resources created by linguists and for linguists – be of any help at all? The results of the experiments with the Beserman data are not conclusive. On the one hand, linguistic intervention did improve the ASR results, lowering WER by 5% and even more so in the case of longer sentences. Linguistic resources, such as the rule-based analyzer turned into a generator, and the Beserman-Russian dictionary, as well as the corpus of usage examples, seemed indispensable in the process. On the other hand, the result is yet another experimental model for a low-resource language with suboptimal performance, which might be not good enough even for auxiliary uses. In order to make it usable, one would still have to either add more data or change the algorithm (e.g. (Baevski et al., 2021) report results for comparable amounts of Tatar and Kyrgyz data that almost look like magic). It would be interesting to see if the “linguistic” LM augmentation adds anything in that case.

Acknowledgments

This research was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project ID 428175960.

References

Aashish Agarwal and Torsten Zesch. 2019. German end-to-end speech recognition based on DeepSpeech. In *KONVENS*.

Timofey Arkhangelskiy. 2019. Corpus of usage examples: What is it good for? In *Proceedings of the 3rd Workshop on the Use of Computational Methods in the Study of Endangered Languages Volume 1 (Papers)*, pages 56–63, Honolulu. Association for Computational Linguistics.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition.

Eckhard Bick and Tino Didriksen. 2015. Cg-3 – beyond classical constraint grammar. In *Proceedings of the 20th Nordic Conference of Computational Linguistics, NODALIDA*, pages 31–39. Linköping University Electronic Press, Linköpings universitet.

Ekaterina Egorova and Lukáš Burget. 2018. Out-of-vocabulary word recovery using fst-based subword unit clustering in a hybrid asr system. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5919–5923.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep Speech: Scaling up end-to-end speech recognition. *arXiv e-prints*, page arXiv:1412.5567.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Nils Hjortnaes, Timofey Arkhangelskiy, Niko Partanen, Michael Rießler, and Francis Tyers. 2020a. Improving the language model for low-resource ASR with online text corpora. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 336–341, Marseille, France. European Language Resources association.

Nils Hjortnaes, Niko Partanen, Michael Rießler, and Francis M. Tyers. 2020b. Towards a speech recognizer for Komi, an endangered and low-resource uralic language. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 31–37, Wien, Austria. Association for Computational Linguistics.

Fred Karlsson. 1990. Constraint grammar as a framework for parsing unrestricted text. In *Proceedings of the 13th International Conference of Computational Linguistics*, volume 3, pages 168–173, Helsinki.

Andrey Kutuzov and Elizaveta Kuzmenko. 2017. *WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models*. Springer International Publishing, Cham.

Juho Leinonen, Peter Smit, Sami Virpioja, and Mikko Kurimo. 2018. New baseline in automatic speech recognition for Northern Sámi. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 87–97, Helsinki, Finland. Association for Computational Linguistics.

Eugenia Miller. 2017. Avtomaticheskoe vyravnivanie slovarej literaturnogo udmurtskogo i jazyka i besermjanskogo dialekta [Automatic alignment of literary Udmurt and Beserman dictionaries]. In *Elektronnaja pismennost narodov Rossijskoj Federacii: opyt, problemy i perspektivy [Electronic writing of the peoples of the Russian Federation: Experience, challenges and perspectives]*, Syktyvkar, Russia. KRAGSiU.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech 2019*, pages 2613–2617.

Mihaela C. Stoian, Sameer Bansal, and Sharon Goldwater. 2020. Analyzing ASR pretraining for low-resource speech-to-text translation. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7909–7913.

Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney. 2014. Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages. In *INTERSPEECH-2014*, pages 1420–1424.

Francis M. Tyers and Josh Meyer. 2021. What shall we do with an hour of data? Speech recognition for the un- and under-served languages of Common Voice.

Maria Usacheva, Timofey Arkhangelskiy, Olga Biryuk, Vladimir Ivanov, and Ruslan Idrisov, editors. 2017. *Тезаурус бесермянского наречия: Имена и служебные части речи (говор деревни Шамардан) [Thesaurus of the Beserman dialect: Nouns and auxiliary parts of speech (Shamardan village variety)]*. Izdatelskie resheniya, Moscow.

The Current State of Finnish NLP

Mika Hämäläinen
Faculty of Arts
University of Helsinki
and Rootroo Ltd
mika.hamalainen@helsinki.fi

Khalid Alnajjar
Faculty of Arts
University of Helsinki
and Rootroo Ltd
khalid.alnajjar@helsinki.fi

Abstract

There are a lot of tools and resources available for processing Finnish. In this paper, we survey recent papers focusing on Finnish NLP related to many different subcategories of NLP such as parsing, generation, semantics and speech. NLP research is conducted in many different research groups in Finland, and it is frequently the case that NLP tools and models resulting from academic research are made available for others to use on platforms such as Github.

Tiivistelmä

Suomen kielen koneelliseen käsittelyyn on tarjolla paljon valmiita työkaluja ja resursseja. Tässä artikkelissa tarkastelemme viimeaikoina julkaistuja tieteellisiä artikkeleita, joissa keskitytään suomen kielen kieliteknologiaan. Tarkastelemme kieliteknologian eri alaluokkia, kuten jäsentämistä, tuottamista, semantiikkaa ja puheetta. kieliteknologista tutkimusta tehdään Suomessa monissa eri tutkimusryhmissä, ja usein akateemisen tutkimuksen tuloksena tuotetut kieliteknologian työkalut ja mallit julkaistaan muiden käytettäväksi esimerkiksi Githubissa.

1 Introduction

There is no doubt that, within the Uralic language family, Finnish is one of the most well-resourced languages in terms of natural language processing (NLP). This has, however, not always been the case. Currently, NLP research conducted for Finnish has started to fragment into research outputs of several different research groups, and there is no survey paper out there that would describe the current state of Finnish NLP.

We hope that this survey paper clarifies the current situation and makes it clearer for people working in the academia outside of Finnish universities or in the industry and also for students. As it has been discussed before (Hämäläinen, 2021), Finnish is certainly not a low-resourced language, and our current survey further proves this point.

It is also important for researchers working on other smaller Uralic languages to see what has been done for Finnish in terms of NLP to see what the possible and meaningful directions are for further developing the resources needed. Especially since Uralic language share the same feature of rich morphology, which is something that commonly causes problems for computers.

2 Finnish NLP

In this section, we present a survey on the current state of Finnish NLP. We have tried to gather most of the current research on the topic, but we are certain that there are some research out there we have not been able to find. We have categorized the surveyed research outputs into parsing, generation, semantics and speech.

2.1 Parsing

Starting from morphology, stemming and spell checking Finnish is well supported in multiple commercial applications such as Microsoft and Google products. In the open-source world, low-level tasks such as stemming and spell checking can be conducted with Voikko[1].

Omorfi (Pirinen, 2015)[2] is currently the most well supported tool for morphological analysis (in-

[1] https://voikko.puimula.org/

[2] https://github.com/flammie/omorfi

Seventh international workshop on computational linguistics of Uralic languages, pages 54–61
September 23–24, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-1-954085-82-4_007

cluding lemmatization) and generation. It is an FST (finite-state transducer) based tool developed on HFST (Helsinki finite-state technology) (Lindén et al., 2013) and it works together with constraint grammar (CG) based disambiguators and syntactic parsers available in the Giellatekno (Moshagen et al., 2014) repositories[3].

FinnPos[4] (Silfverberg et al., 2016) is another morphological tagger and lemmatizer tool based on CRF (conditional random field). There have been recently more data driven approaches focusing on Finnish (Silfverberg and Hulden, 2018).

While rule-based tradition has been strong in the past[5], there are several machine learning driven dependency parsers for Finnish, such as the statistical one[6] (Haverinen et al., 2014) and neural one[7] (Kanerva et al., 2018) by TurkuNLP.

Out of the aforementioned tools Omorfi (and the CG disambigator) and the machine learning based parsers are available to use through a Python package named UralicNLP[8] [9] (Hämäläinen, 2019).

As Finnish data is available in several multilingual datasets, there are many multilingual approaches for parsing (Qi et al., 2020)[10] (Honnibal et al., 2020)[11] and morphology (Aharoni and Goldberg, 2017; Nicolai and Yarowsky, 2019; Silfverberg and Tyers, 2019; Grönroos et al., 2020).

The fact that spoken Finnish is very different to standard Finnish has drawn some attention in the past (Jauhiainen, 2001) and recently (Partanen et al., 2019). The latter leading to a Python library called Murre[12] for automatic normalization of dialectal Finnish.

Non-standard data has been an issue in digital humanities (DH) projects (Mäkelä et al., 2020), and lately there have been efforts in automatically correcting OCR errors in existing historical datasets (Kettunen, 2015; Drobac and Lindén, 2020; Drobac, 2020; Duong et al., 2020).

Named entity recognition has also been under study with FiNER[13] and its recently released data (Ruokolainen et al., 2019). There is also another recent BERT (Devlin et al., 2019) based approach[14] to the topic (Luoma et al., 2020).

There have been several approaches to language detection including detection of Finnish from web corpora (see Jauhiainen et al., 2021). Similarly, native Finnish has been automatically identified from learner's Finnish (Malmasi and Dras, 2014).

In summary, parsing has been researched on different levels of language such as syntax, morphology, POS and NER tagging, and lemmatization. It has been mainly focusing on standard well-formed Finnish, although there are methods for coping with dialectal Finnish and OCR errors as well.

2.2 Generation

The lowest level of natural language generation is surface realization (see Reiter, 1994), and for that there are tools such as Omorfi and Syntax Maker[15] (Hämäläinen and Rueter, 2018). The latter uses Omorfi for morphological inflection while it takes care of higher level morphosyntax such as case government and agreement.

There is a strong computational creativity focus in Helsinki and it also shows in Finnish NLG, as there are several poem generators such as Keinoleino[16] (Hämäläinen, 2018b), Poeticus (Toivanen et al., 2012) and others (Hämäläinen and Alnajjar, 2019a,b). There is also an interactive poem generator tool called *Runokone* (Poem Machine)[17] (Hämäläinen, 2018c).

Recently there have been several approaches to enhancing existing news headlines (Alnajjar et al., 2019; Rämö and Leppänen, 2021). And some approaches to generating entire news articles automatically (Kanerva et al., 2019; Haapanen and Leppänen, 2020).

Paraphrase generation (Sjöblom et al., 2020) has also become a researched topic with the availability of monolingually aligned parallel corpora (Creutz, 2018). There is also an approach to converting standard Finnish text into different dialects (Hämäläinen et al., 2020).

Finnish is a typical language for machine translation tasks and it is not uncommon to see it featured in several papers that deal with multiple languages. However, there are several papers that fo-

[3] https://github.com/giellalt/lang-fin/tree/main/src/cg3

[4] https://github.com/mpsilfve/FinnPos

[5] See Pirinen, 2019b for some comparison between rules and neural networks

[6] https://turkunlp.org/Finnish-dep-parser/

[7] http://turkunlp.org/Turku-neural-parser-pipeline/

[8] https://github.com/mikahama/uralicNLP

[9] https://github.com/mikahama/uralicNLP/wiki/Dependency-parsing

[10] https://stanfordnlp.github.io/stanza/

[11] https://spacy.io/

[12] https://github.com/mikahama/murre

[13] https://github.com/Traubert/FiNer-rules/blob/master/finer-readme.md

[14] https://turkunlp.org/fin-ner.html

[15] https://github.com/mikahama/syntaxmaker

[16] https://github.com/mikahama/keinoleino

[17] http://runokone.cs.helsinki.fi/

cus on Finnish in particular (Hurskainen and Tiedemann, 2017; Hämäläinen and Alnajjar, 2019c; Pirinen, 2019a; Tiedemann et al., 2020).

There is also a recent approach to dialog generation in Finnish (Leino et al., 2020). Also non-native language learner's errors have been corrected successfully automatically (Creutz and Sjöblom, 2019).

To summarize the approaches, there are several generators for poetry and news that benefit from the available surface realizers. Paraphrasing, dialect adaptation, dialog generation and learners' error correction are domains with some research with potential for new discoveries in the future. Machine translation gets frequently attention from different researchers. There are several more NLG tasks (see Gatt and Krahmer 2018) that have not been researched at all in Finnish, which means that there is a lot of room for more research on this topic.

2.3 Semantics

Vector representations of meaning have become common place in NLP and Finnish is no exception with the availability of pretrained word2vec[18] [19] (Laippala and Ginter, 2014; Kutuzov et al., 2017) and fastText[20] (Bojanowski et al., 2017) models.

BERT models have also become available as part of the multilingual BERT model[21] (Devlin et al., 2019) or trained separately for Finnish[22] [23] (Kutuzov et al., 2017; Virtanen et al., 2019). Even Elmo models have been made available for Finnish[24] (Ulčar and Robnik-Šikonja, 2020).

In addition to the standard vector-based representations of meaning, there is another statistical model called SemFi[25] (Hämäläinen, 2018a). The model is a relational database that captures semantic relations of words based on their syntactic cooccurencies.

Before the era of machine learning, there were two prominent projects for modeling meaning computationally which have been translated into Finnish WordNet (Lindén and Carlson, 2010) and FrameNet (Lindén et al., 2019).

With the similar ideology to the hand crafted resources, there have been several different linked data projects in Finland representing semantics in structured ontologies (Hyvönen et al., 2006; Nyrkkö, 2018; Thomas et al., 2018; Koho et al., 2019). Many of the linked data projects are available on the Linked Data Finland website[26].

There is a Python library called FinMeter[27] (Hämäläinen and Alnajjar, 2019b) that has some higher level semantic tools for Finnish such as metaphor interpretation, word concreteness analysis and sentiment analysis. Sentiment analysis for Finnish has also been studied later on[28] (Öhman et al., 2020; Vankka et al., 2019; Lindén et al., 2020). There is also research on topic modeling methods (Ginter et al., 2009; Hengchen et al., 2018; Loukasmäki and Makkonen, 2019).

Finnish is well supported by traditional representations of semantics and latest vector based models. There is a vast amount of linked data resources in a variety of domains. Higher-level semantics such as metaphor interpretation and sentiment analysis also have received their share of research interest, although there are many more questions related to pragmatics and figurative language that have not been researched, such as sarcasm detection, multihop reasoning and fake news detection to name a few.

2.4 Speech

Apart from Finnish speech being supported by companies, there are some open-source tools that can synthesize Finnish. Festival[29] has a Finnish voice named Suopuhe[30], and eSpeak-ng[31] can even generate IPA characters for Finnish.

There are several more modern approaches to speech recognition (Enarvi et al., 2017; Varjokallio et al., 2021) and speech synthesis (Raitio et al., 2008, 2014). Although, speech synthesis has not gained much interest in the recent years.

There are several approaches to analyzing speech prosody (Virkkunen et al., 2018; Šimko et al., 2020). There is also some work on detecting different accents in spoken Finnish (Behravan et al., 2013, 2015) and named entity recognition (Porjazovski et al., 2020).

In summary, several approaches exist for speech processing in Finnish relating to recognition, ac-

[18] http://vectors.nlpl.eu/repository/

[19] https://bionlp.utu.fi/finnish-internet-parsebank.html

[20] https://fasttext.cc/docs/en/pretrained-vectors.html

[21] https://github.com/google-research/bert

[22] http://vectors.nlpl.eu/repository/

[23] https://github.com/TurkuNLP/FinBERT

[24] https://www.clarin.si/repository/xmlui/handle/11356/1277

[25] https://github.com/mikahama/uralicNLP/wiki/Semantics-(SemFi,-SemUr)

[26] https://www.ldf.fi/

[27] https://github.com/mikahama/finmeter

[28] a dataset https://github.com/Helsinki-NLP/XED

[29] https://www.cstr.ed.ac.uk/projects/festival/

[30] http://urn.fi/urn:nbn:fi:lb-20140730144

[31] https://github.com/espeak-ng/espeak-ng

cents and prosody. However, speech synthesis has received a surprisingly small amount of attention in the recent past. With the emergence of neural models, new research on synthesis could reach to potentially interesting new contributions.

3 Discussion and Conclusions

In this survey, we have gathered research conducted on different aspects of NLP. We have included links to models and code implementations for most of the research papers. It has been a pleasant thing to notice that not only Finnish NLP research exists but also it is often not conducted in a closed fashion, but the actual research outputs have been made openly available for a wider community of people even outside of academia. This is crucial for any language that is relatively small, like Finnish. If Finnish academics did not release their research, there would not be many other people in the world that would produce high-quality tools for Finnish.

Digital extinction is something that many endangered languages are facing right now (see Kornai 2013). Therefore, it is important to ensure that NLP resources become openly available for endangered Uralic languages as well. Availability itself is not enough, however, as the resources need to be easy to find and use. Despite the fact that we have open NLP tools for Finnish, we are still far a way from a world where machines use our language fluently. Finnair's in-flight entertainment system still announces happliy: **saavumme kohteeseen Helsinki* (*we arrive in destination Helsinki) instead of expressing it correctly, *saavumme Helsinkiin* (we arrive in Helsinki), Google Doc's spell checker does not recognize mostly any inflectional form with a possessive suffix and predictive text in mobile keyboards suggest overly formal normative Finnish only.

While Finnish NLP has come far in terms of academic research and tools built as a result, we as a nation are still far away from having Finnish language technology fully integrated into the systems we use every day. Many of the problems have been solved already, it is just the matter of the industry finding out about the NLP tools that are out there.

We have limited our survey to NLP tools and methods only. We know that there are a plethora of language resources available for Finnish as well. Based on our experiences, many corpora are well hidden and digging them up is a time consuming effort worthy of a separate survey paper. Unfortunately the Finnish practice of describing data on Metashare[32] is very unhelpful in this respect because the metadata descriptions in the service hardly ever contain information about where to access the data, how to cite it and who the real authors are.

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2004–2015, Vancouver, Canada. Association for Computational Linguistics.

Khalid Alnajjar, Leo Leppänen, Hannu Toivonen, et al. 2019. No time like the present: methods for generating colourful and factual multilingual news headlines. In *Proceedings of the 10th International Conference on Computational Creativity*. Association for Computational Creativity.

Hamid Behravan, Ville Hautamäki, and Tomi Kinnunen. 2013. Foreign accent detection from spoken finnish using i-vectors. In *INTERSPEECH*, volume 2013, page 14th.

Hamid Behravan, Ville Hautamäki, and Tomi Kinnunen. 2015. Factors affecting i-vector based foreign accent recognition: A case study in spoken finnish. *Speech Communication*, 66:118–129.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Mathias Creutz. 2018. Open subtitles paraphrase corpus for six languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018).*

Mathias Creutz and Eetu Eetu Sjöblom. 2019. Toward automatic improvement of language produced by non-native language learners. In *Proceedings of the 8th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2019), September 30, Turku Finland*, 164, pages 20–30. Linköping University Electronic Press.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[32] https://metashare.csc.fi/

Senka Drobac. 2020. *OCR and post-correction of historical newspapers and journals*. Ph.D. thesis, University of Helsinki, Finland.

Senka Drobac and Krister Lindén. 2020. Optical character recognition with neural networks and post-correction with finite state methods. *International Journal on Document Analysis and Recognition (IJDAR)*, 23(4):279–295.

Quan Duong, Mika Hämäläinen, and Simon Hengchen. 2020. An unsupervised method for ocr post-correction and spelling normalisation for finnish. *arXiv preprint arXiv:2011.03502*.

Seppo Enarvi, Peter Smit, Sami Virpioja, and Mikko Kurimo. 2017. Automatic speech recognition with very large conversational finnish and estonian vocabularies. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(11):2085–2097.

Albert Gatt and Emiel Krahmer. 2018. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Filip Ginter, Hanna Suominen, Sampo Pyysalo, and Tapio Salakoski. 2009. Combining hidden markov models and latent semantic analysis for topic segmentation and labeling: Method and clinical application. *International journal of medical informatics*, 78(12):e1–e6.

Stig-Arne Grönroos, Sami Virpioja, and Mikko Kurimo. 2020. Morfessor em+ prune: Improved subword segmentation with expectation maximization and pruning. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3944–3953.

Lauri Haapanen and Leo Leppänen. 2020. Recycling a genre for news automation: The production of valtteri the election bot. *AILA Review*, 33(1):67–85.

Mika Hämäläinen. 2018a. Extracting a semantic database with syntactic relations for finnish to boost resources for endangered uralic languages. *The Proceedings of Logic and Engineering of Natural Language Semantics 15 (LENLS15)*.

Mika Hämäläinen. 2018b. Harnessing nlg to create finnish poetry automatically. In *Proceedings of the ninth international conference on computational creativity*. Association for Computational Creativity (ACC).

Mika Hämäläinen. 2018c. Poem machine-a co-creative nlg web application for poem writing. In *The 11th International Conference on Natural Language Generation Proceedings of the Conference*. The Association for Computational Linguistics.

Mika Hämäläinen and Khalid Alnajjar. 2019a. Generating modern poetry automatically in finnish. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing Proceedings of the Conference*. The Association for Computational Linguistics.

Mika Hämäläinen and Khalid Alnajjar. 2019b. Let's face it. finnish poetry generation with aesthetics and framing. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 290–300.

Mika Hämäläinen and Khalid Alnajjar. 2019c. A template based approach for training nmt for low-resource uralic languages-a pilot with finnish. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, pages 520–525.

Mika Hämäläinen, Niko Partanen, Khalid Alnajjar, Jack Rueter, and Thierry Poibeau. 2020. Automatic dialect adaptation in finnish and its effect on perceived creativity. In *11th International Conference on Computational Creativity (ICCC'20)*. Association for Computational Creativity.

Mika Hämäläinen and Jack Rueter. 2018. Development of an open source natural language generation tool for Finnish. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 51–58, Helsinki, Finland. Association for Computational Linguistics.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation*, 48:493–531. Open access.

Simon Hengchen, Antti Olavi Kanner, Jani Pekka Marjanen, and Eetu Mäkelä. 2018. Comparing topic model stability between finnish, swedish, english and french. In *Digital Humanities in the Nordic Countries*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Arvi Hurskainen and Jörg Tiedemann. 2017. Rule-based machine translation from english to finnish. In *Proceedings of the Second Conference on Machine Translation*, pages 323–329.

Eero Hyvönen, Tuukka Ruotsalo, Thomas Häggström, Mirva Salminen, Miikka Junnila, Mikko Virkkilä, Mikko Haaramo, Eetu Mäkelä, Tomi Kauppinen, and Kim Viljanen. 2006. Culturesampo–finnish culture on the semantic web: The vision and first results. In *Developments in Artificial Intelligence and the Semantic Web-Proceedings of the 12th Finnish AI Conference STeP*, pages 26–27.

Mika Hämäläinen. 2019. UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software*, 4(37):1345.

Mika Hämäläinen. 2021. Endangered languages are not low-resourced! In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*. Rootroo Ltd.

Tommi Jauhiainen. 2001. Using existing written language analyzers in understanding natural spoken Finnish. In *Proceedings of the 13th Nordic Conference of Computational Linguistics (NODALIDA 2001)*, Uppsala, Sweden. Department of Linguistics, Uppsala University, Sweden.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2021. Suomalais-ugrilaiset kielet ja internet -projekti 2013-2019. In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*. Rootroo Ltd.

Jenna Kanerva, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. Turku neural parser pipeline: An end-to-end system for the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

Jenna Kanerva, Samuel Rönnqvist, Riina Kekki, Tapio Salakoski, and Filip Ginter. 2019. Template-free data-to-text generation of finnish sports news. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 242–252.

Kimmo Kettunen. 2015. Keep, change or delete? setting up a low resource ocr post-correction framework for a digitized old finnish newspaper collection. In *Italian Research Conference on Digital Libraries*, pages 95–103. Springer.

Mikko Koho, Esko Ikkala, Petri Leskinen, Minna Tamper, Jouni Tuominen, and Eero Hyvönen. 2019. Warsampo knowledge graph: Finland in the second world war as linked open data. *Semantic Web*, (Preprint):1–14.

András Kornai. 2013. Digital language death. *PloS one*, 8(10):e77056.

Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press.

Veronika Laippala and Filip Ginter. 2014. Syntactic n-gram collection from a large-scale corpus of internet finnish. In *Human Language Technologies-The Baltic Perspective: Proceedings of the Sixth International Conference Baltic HLT*, volume 268, page 184.

Katri Leino, Juho Leinonen, Mittul Singh, Sami Virpioja, and Mikko Kurimo. 2020. Finchat: Corpus and evaluation setup for finnish chat conversations on everyday topics. *Proc. Interspeech 2020*, pages 429–433.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A Pirinen, and Miikka Silfverberg. 2013. HFST a system for creating NLP tools. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 53–71. Springer.

Krister Lindén and Lauri Carlson. 2010. Finnwordnet–finnish wordnet by translation. *LexicoNordica–Nordic Journal of Lexicography*, 17:119–140.

Krister Lindén, Heidi Haltia, Antti Laine, Juha Luukkonen, Jussi Piitulainen, and Niina Väisänen. 2019. Finntransframe: translating frames in the finnframenet project. *Language Resources and Evaluation*, 53(1):141–171.

Krister Lindén, Tommi Jauhiainen, and Sam Hardwick. 2020. Finnsentiment–a finnish social media corpus for sentiment polarity annotation. *arXiv preprint arXiv:2012.02613*.

Petri Loukasmäki and Kimmo Makkonen. 2019. Eduskunnan täysistunnon puheenaiheet 1999-2014: miten käsitellä lda-aihemalleja? *Politiikka*, 61(2).

Jouni Luoma, Miika Oinonen, Maria Pyykönen, Veronika Laippala, and Sampo Pyysalo. 2020. A broad-coverage corpus for finnish named entity recognition. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4615–4624.

Eetu Mäkelä, Krista Lagus, Leo Lahti, Tanja Säily, Mikko Tolonen, Mika Hämäläinen, Samuli Kaislaniemi, Terttu Nevalainen, et al. 2020. Wrangling with non-standard data. In *Proceedings of the Digital Humanities in the Nordic Countries 5th Conference Riga, Latvia, October 21-23, 2020*. CEUR-WS. org.

Shervin Malmasi and Mark Dras. 2014. Finnish native language identification. In *Proceedings of the Australasian Language Technology Association Workshop 2014*, pages 139–144.

Sjur Moshagen, Jack Rueter, Tommi Pirinen, Trond Trosterud, and Francis M. Tyers. 2014. Open-source infrastructures for collaborative work on under-resourced languages. The LREC 2014 Workshop “CCURL 2014 - Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era”.

Garrett Nicolai and David Yarowsky. 2019. Learning morphosyntactic analyzers from the Bible via iterative annotation projection across 26 languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1765–1774, Florence, Italy. Association for Computational Linguistics.

Seppo Nyrkkö. 2018. Building a finnish som-based ontology concept tagger and harvester. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 18–25.

Emily Öhman, Marc Pàmies, Kaisla Kajava, and Jörg Tiedemann. 2020. Xed: A multilingual dataset for sentiment analysis and emotion detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6542–6552.

Niko Partanen, Mika Hämäläinen, and Khalid Alnajjar. 2019. Dialect text normalization to normative standard Finnish. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 141–146, Hong Kong, China. Association for Computational Linguistics.

Tommi Pirinen. 2019a. Apertium-fin-eng–rule-based shallow machine translation for WMT 2019 shared task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 335–341, Florence, Italy. Association for Computational Linguistics.

Tommi A Pirinen. 2015. Development and use of computational morphology of finnish in the open source and open science era: Notes on experiences with omorfi development. *SKY Journal of Linguistics*, 28:381–393.

Tommi A Pirinen. 2019b. Neural and rule-based finnish nlp models—expectations, experiments and experiences. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 104–114.

Dejan Porjazovski, Juho Leinonen, and Mikko Kurimo. 2020. Named entity recognition for spoken finnish. In *Proceedings of the 2nd International Workshop on AI for Smart TV Content Production, Access and Delivery*, pages 25–29.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 101–108, Online. Association for Computational Linguistics.

Tuomo Raitio, Heng Lu, John Kane, Antti Suni, Martti Vainio, Simon King, and Paavo Alku. 2014. Voice source modelling using deep neural networks for statistical parametric speech synthesis. In *2014 22nd European Signal Processing Conference (EUSIPCO)*, pages 2290–2294. IEEE.

Tuomo Raitio, Antti Suni, Martti Vainio, and Paavo Alku. 2008. Hmm-based finnish text-to-speech system utilizing glottal inverse filtering. In *Interspeech 2008, Brisbane, Australia, September 22-26, 2008*.

Miia Rämö and Leo Leppänen. 2021. Using contextual and cross-lingual word embeddings to improve variety in template-based nlg for automated journalism. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 62–70.

Ehud Reiter. 1994. Has a consensus nl generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation*.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2019. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, pages 1–26.

Miikka Silfverberg and Mans Hulden. 2018. Initial experiments in data-driven morphological analysis for Finnish. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages*, pages 98–105, Helsinki, Finland. Association for Computational Linguistics.

Miikka Silfverberg, Teemu Ruokolainen, Krister Lindén, and Mikko Kurimo. 2016. Finnpos: an open-source morphological tagging and lemmatization toolkit for finnish. *Language Resources and Evaluation*, 50(4):863–878.

Miikka Silfverberg and Francis Tyers. 2019. Data-driven morphological analysis for uralic languages. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 1–14, Tartu, Estonia. Association for Computational Linguistics.

Eetu Sjöblom, Mathias Creutz, and Yves Scherrer. 2020. Paraphrase generation and evaluation on colloquial-style sentences. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1814–1822.

Suzanne Elizabeth Thomas, Anna Pia Frederike Wessman, Esko Ikkala, Jouni Antero Tuominen, Mikko Koho, Eero Antero Hyvönen, and Ville Rohiola. 2018. (co-) creating a sustainable platform for finland's archaeological chance finds: The story of sualt. In *Digital Heritage and Archaeology in Practice*. University press of Florida.

Jörg Tiedemann, Tommi Nieminen, Mikko Aulamo, Jenna Kanerva, Akseli Leino, Filip Ginter, and Niko Papula. 2020. The FISKMÖ project: Resources and tools for Finnish-Swedish machine translation and cross-linguistic research. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3808–3815, Marseille, France. European Language Resources Association.

Jukka Toivanen, Hannu Toivonen, Alessandro Valitutti, Oskar Gross, et al. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the third international conference on computational creativity*. University College Dublin.

Matej Ulčar and Marko Robnik-Šikonja. 2020. High quality elmo embeddings for seven less-resourced languages[33] In *Proceedings of The 12th Language*

[33] We don't agree with the title of the paper declaring Finnish as a "less-resourced" language. As we have seen in this paper Finnish does have a bunch of resources!

Resources and Evaluation Conference, pages 4731–4738.

Jouko Vankka, Heikki Myllykoski, Tuomas Peltonen, and Ken Riippa. 2019. Sentiment analysis of finnish customer reviews. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 344–350.

Matti Varjokallio, Sami Virpioja, and Mikko Kurimo. 2021. Morphologically motivated word classes for very large vocabulary speech recognition of finnish and estonian. *Computer Speech & Language*, 66:101141.

Päivi Johanna Virkkunen, Juraj Simko, Heini Henriikka Kallio, Martti Tapani Vainio, et al. 2018. Prosodic features of finnish compound words. In *Proceedings of the 9th International Conference on Speech Prosody 2018*. International Speech Communications Association.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. Multilingual is not enough: Bert for finnish. *arXiv preprint arXiv:1912.07076*.

Juraj Šimko, Martti Vainio, and Antti Suni. 2020. Analysis of speech prosody using WaveNet embeddings: The Lombard effect. In *Proc. 10th International Conference on Speech Prosody 2020*, pages 910–914.

Overview of Open-Source Morphology Development for the Komi-Zyrian Language: Past and Future

Jack Rueter
University of Helsinki
jack.rueter@helsinki.fi

Niko Partanen
University of Helsinki
niko.partanen@helsinki.fi

Mika Hämäläinen
University of Helsinki & Rootroo Ltd
mika.hamalainen@helsinki.fi

Trond Trosterud
Norwegian Arctic University
trond.trosterud@uit.no

Abstract

This study describes the on-going development of the finite-state description for an endangered minority language, Komi-Zyrian. This work is located in the context where large written and spoken language corpora are available, which creates a set of unique challenges that have to be, and can be, addressed. We describe how we have designed the transducer so that it can benefit from existing open-source infrastructures and therefore be as reusable as possible.

Дзеньыдӧн

Тайӧ гижӧдын сёрни мунӧ канму коми кыв технология йылысь, кӧнi сетӧмаӧсь коми морфологиялы помысь-помӧдз автомат. Уджыс сэтшӧм контекстын, кӧнi ыджыд гижан да сёрнисикас корпусъяс босьтанног. Та вӧсна чужӧны торйӧн юалӧмъяс, кодлы выль воча кывъяс коланаӧсь. Петкӧдлам, мый эм кыдзи аддзыны колана воча кывъяс. Серпасалам анализатор-автоматлысь сӧвмӧдӧм процесс да вӧзйӧмным ӧтлаӧдны анализаторсӧ паськыдджык восся кодъяса ӧтувтечасӧ-инфраструктураӧ, медым уджыс уналаздоръясын вӧдитчыны.

1 Introduction

This study discusses open-source morphology development, which has greatly benefited from open-source projects most notably achievements attributed to the GiellaLT infrastructure (Moshagen et al., 2014), i.e. Giellatekno & Divvun at the Norwegian Arctic University in Tromsø, Norway. Specifically we discuss the infrastructure for the Komi-Zyrian language. We describe the work done up until now, and delineate some of the tasks we deem necessary in the future. There are features of Komi morphosyntax that need special attention, in regard to both of their linguistic and computational descriptions. This contribution aims to bring that discussion forward, and delineate the current status of the work.

Rueter (2000) describes the initial creation of the transducer, and the work discussed here continues that same undertaking, essentially providing an update of the changes done in the last decade, and a plan for the future. The transducer is available on GitHub for Komi-Zyrian.[1] The nightly builds are available through a Python library called UralicNLP[2] (Hämäläinen, 2019). Easy and efficient access to the traducers and their lexical materials has been the main designing principle, and we consider current approach very successful.

Komi-Zyrian has a growing representation in online corpora. There is a large written corpus that is accessible online[3]; it has been created by FU-Lab in Syktyvkar. The Giellatekno infrastructure provides a Korp implementation (Ahlberg et al., 2013) hosting numerous Uralic Wikipedia corpora, among which Komi can also be found[4]. At the Language Bank of Finland, parallel Bible corpora are available with possibilities for comparing different translations (Rueter and Axelson, 2020). While literary language often reflects astute professional language users, social media provides written language that may be more closely related to the vernacular, this type of Komi is found with minority languages of the adjacent Volga-Kama region[5] and as described in Arkhangelskiy (2019). In a simi-

[1] https://github.com/giellalt/lang-kpv

[2] https://github.com/mikahama/uralicNLP

[3] http://komicorpora.ru

[4] http://gtweb.uit.no/u_korp/#?lang=en

[5] http://komi-zyrian.web-corpora.net/index_en.html

Seventh international workshop on computational linguistics of Uralic languages, pages 62–72
September 23–24, 2021. ©2021 Association for Computational Linguistics
https://doi.org/10.26615/978-1-954085-82-4_008

lar vein, a Spoken Komi corpus containing mainly Izhma dialect has been created in a Kone Foundation funded research project (Blokland et al., 2014–2016), and it is also available online for community and research access.[6] Written and spoken language corpora are different in many ways, but together they form a large and representative description of the Komi language. Thereby they both need to be accounted for when the transducer is further developed. Electronic corpora have an important role in the research of Komi in general, and their significance most certainly will only grow when access and practices improve (for discussion about the use of electronic corpora, see Федина, 2019; Чупров, 2018; Блокланд et al., 2014).

There are also numerous dialect materials in Komi, and their progressing digitization gives us access to an increasing number of materials hitherto unavailable in digital format. When this process advances and we inevitably encounter more dialectal texts, we must also consider wider dialectal features of Komi when we develop the transducer.

Additionally, as there are two main Komi varieties with written standards and their dialects, Zyrian and Permyak, we must acknowledge that infrastructures for these languages cannot be developed in isolation, but rather that both language variants must be taken into consideration in different ways (Rueter et al., 2020c). At the same time, the respective written standards have needs for their own tools and resources that are still independent, so the whole question of how to best handle pluricentric language varieties such as Komi still needs additional planning.

The study is structured so that we first describe the work that has been done for modeling the morphosyntax of the Komi-Zyrian language. Then we discuss individual features and their role in the description, and aim to illustrate the types of challenges they present. As we believe that computational modeling of the language is directly connected to the linguistic description itself, we also discuss different phenomena and the ways our description is directly connected to the grammar.

2 Development history of the Komi-Zyrian FST

The FST described here is primarily built by Jack Rueter, beginning with work in the 1990s. The Komi-Zyrian finite-state description began with a trilingual glossary *Ӧшкамӧшка ичӧт кыввор, комиа-англискӧя-финскӧя* (Rueter, 1995), designed for use by Finnish and English speaking students of Komi, without previous knowledge of Russian, to accompany the **коми кыв** '*Komi language*' reader (Цыпанов, 1992), used for instruction in the Universities of Helsinki and Turku. Later, with a scholarship from the Kordelin Foundation, this vocabulary was augmented. First, the extension was intended to complement a second Komi reader by Манова (1994), and then to outline the Komi stem vocabulary of the Komi-Russian dictionary by Лыткин and Тимушев (1961). A large portion of the work done with this dictionary was only possible with the painstaking hours spent by Vera Chernykh. Thus, the approximately 3000-word glossary providing the lexical base for a finite-state description of Komi-Zyrian, presented at **Permistika 6** at the Udmurt State University in Izhevsk, 1996 (published in Rueter, 2000), was extended to over 6000 lexical entries.

In 2004 Trond Trosterud invited Rueter to Tromsø to learn more about the Xerox Finite-state technology (XFST) being implemented at Giellatekno as described in (Trosterud, 2004) and for Komi in (Trosterud, 2004b). Here the Komi transducer and lexicon were to be developed further than before, and to be connected to an infrastructure that was compatible with a larger array of languages.

To summarise some of the new improvements, there were no longer problems with Cyrillic letters requiring representation as conversions from Latin letters. It was now possible to write rules directly addressing elements of the Komi orthography. This direct use of the vernacular in the code may have, in fact, contributed to the belief of the developer that only the normative language needed description. (It was not until many years later that work with other under-resourced languages, such as Mansi (2015–present), Olonets-Karelian (2013–present), Skolt Saami (2015–present) and Võro (2014–present), made it obvious that non-standard words also require description.)

One of the most important items at this point was that the lexicon and morphology were open-source. This meant, in turn, that Komi could be worked on by others and tested in projects. Here, Komi was ideal. The morphology is very concatenative, and the orthography contains only two more letters than the Russian, i.e. problems with some rarer Cyrillic letters could be evaluated and solved.

[6] `http://videocorpora.ru`

In 2012–2016 Paula Kokkonen worked in conjunction with one of Rueter's projects, where she improved the Finnish translations and inspected the English translations for Komi lexemes. This work significantly increased the coverage of Finnish translations in the multilingual dictionary that was created in this point.

During the period 2012–2021, FU-Lab and Giellatekno collaboration has featured active FST development, including multiple use, and especially improvement in the disambiguation rules and lexical coverage. Morphological analysis is a central component in a modern corpus, and issues such as ambiguity are also always present when FST is used in this context (Öньӧ Лав, 2015, 140). Collaboration may also lead to unforeseeable development. When two infrastructures are aligned, there are often competing priorities. This has also been the case here, i.e. whereas FU-Lab has demonstrated immediate interest in the facilitation of writing, spell checking, dictionaries and corpora for the language community, Giellatekno has pushed for research-related morphological description, analysis and lexica for the research community, but which can, in fact, later be applied to the production of spell checking and other derivative tools. This divergence in priority lead to some duplicate work in morphology.

Helsinki Finite-State Technology (HFST) (Lindén et al., 2013) at Giellatekno with multi-use priorities was pitted against the quick but single-use Hunspell strategies practiced at FU-Lab. Thus, some of the technical complexities on the Giellatekno side had to be simplified so that one set of lexica might be shared. Giellatekno had plenty to gain from the lexical work done at FU-Lab, on the one hand, but it was not able to capitalize on its own sophisticated two-level description as a result of it, on the other. As regards morphophonological descriptions, stem-final variation had to be moved one step away from the initial LEMMA + COLON + STEM + CONTINUATIONLEXICON declaration in the code.

While Jack Rueter has often quickly followed the suggestion of XML maintenance of lexical materials, it has turned out that collaboration pulls away from this write-only-once policy. The more people there are working with one data set, the more documentation required for maintaining mutual working principles. Simple and complex XML systems alike require a working front-end, otherwise, as has been the case here, the workers opt out of the XML database and end up working more on materials that cannot be readily integrated back into the system.

At the moment the XML transformation is not being used in FST development. Instead, other solutions for database implementation are being worked on, see Alnajjar et al. (2020a); Hämäläinen et al. (2021). Only time will reveal which directions of development have contributed the most to the infrastructure.

In 2018-2021, Niko Partanen has been improving the dialectal lexicon coverage of the transducer while conducting his doctoral studies in Komi dialectology. In connection to this work, in 2020-2021, Jack Rueter has improved the coverage of dialectal morphology, specifically taking into account the phenomena found in the Izhma dialect. This work by both of them was done within a Kone Foundation funded research project *Language Documentation Meets Language Technology: The Next Step in the Description of Komi*. The work shows that it is a feasible strategy to improve the analyser so that the work aligns with specific goals and needs of an individual project or dataset. It does create an imbalance in to which degree different dialects are represented, but for a language as large as Komi doing everything at the same time is not possible either.

Mika Hämäläinen's role has been central in building more widely accessible computational infrastructure to access these transducers (Hämäläinen, 2019). In the recent work to create an online editing platform that would allow improved access to the lexical materials, Khalid Alnajjar has been in an irreplaceable position (Alnajjar et al., 2020a). This all shows that managing a transducer for a language like Komi is a multi-partnered operation that calls for wide collaboration between different groups and even infrastructures.

Since 2017, work has been conducted within the Universal Dependencies project to better cover Komi varieties, most recently (Zeman et al., 2021), see also Nivre et al. (2020). There are two Zyrian treebanks (Partanen et al., 2018), and work with Permyak progresses at many levels (Rueter et al., 2020c). Especially in the initial phase of the treebank, building the finite-state descriptions is in a pivotal role, and maintaining interoperability between the FST and treebank development allows very efficient use of both systems. A similar approach has also been systematically used for other languages, such as Karelian (Pirinen, 2019a) and both Mordvinic languages (Rueter et al., 2020a). Indeed, managing systematic and comparable use of tags and conventions across languages is one of the primary concerns in our work as well, and there have been specific surveys that try to track the progress of different Uralic treebanks (Rueter and Partanen, 2019). We can also mention that the practices described here have also be adopted for the development of Amazon minority language description for Apurinã in Helsinki-Belém (Rueter et al., 2021). In the approach discussed here, this harmonization starts at the transducer level and the documentation therein.

In the context of concrete applications of the Komi FST, we can highlight work by Gerstenberger et al. (2017), where the analyser was integrated into the popular multimedia annotation software ELAN. In addition, the most significant Komi online resource, the National Komi Corpus, contains annotations done with the transducer[7].

Next we describe some of the challenges and important phenomena that have been addressed in various ways when creating the Komi analyser.

3 On describing regular morphology

Komi regular morphology affects word forms in several parts of speech. In addition to verbal conjugation and nominal declension, there is an abundance of regular morpheme-sememe alignment in derivation. Whereas verbal conjugation is, indeed, limited to the indicative (in four synthetic tenses) and imperative moods, the complex noun-phrase head is associated with the categories of number (singular and plural), possessive marking for three persons and two numbers as well as nearly thirty syntactic entity markers or cases. Regular derivation can be observed in aspect, mediopassive and causative marking of verbs, as well as comparative and diminutive marking of nominals. There is a plethora of single-syllable nouns and derivational suffixes, and, at times, the boundary between compounding and derivation becomes obscure.

3.1 Stem variation

The Komi-Zyrian language is known to display a typologically common l-vocalization, which is a process where a lateral approximant is replaced by a labiodental fricative /v/ or labiodental approximant /ʋ/. In the Komi grammaticography this is known as l/v variation. Another comparable stem-alterating phenomena are the paragogic consonants in some word stems. These phenomena can be dealt with in much the same way, as they share a common trigger. Words with l/v or paragogic consonant variation in their stems can be identified on the basis of whether the stem is followed by an vowel-initial suffix, on the one hand, or a consonant-initial suffix (alternatively word boundary), on the other.

In the description of these words it has been suggested that erroneous forms be specifically identified. Special tags indicating the absence of paragogic consonants or substandard realization of the stem-final l/v have been implemented for Komi-Zyrian and reflect parallel tags previously implemented in the FST descriptions of other languages in the GiellaLT infrastructure, Northern and Skolt Saami, Erzya, Moksha, Võro to mention a few.

When we include more dialectal materials in the description, we also have to account for processes where l-vocalization triggers vowel lengthening. There are also secondary types of l-vocalization, influencing stems ending in the sequence /*-el*/, and triggering change /*-ej*/. Currently this is treated at the lemma level, so that the non-standard forms are connected to the standard lemmas, with an additional tag indicating dialectal form or error. Even the dialectal variants where neither types of the variation are met are exceptions from the point of view of the standard language. We have devised a tagging system for various subtypes, but the exact implementation is still being designed and planned further. We discuss in Section 4.2 related challenges in more detail.

3.2 Case

As mentioned above, there are nearly thirty syntactic entity markers or cases associated with complex noun phrases. The distinction drawn here of cases versus derivations lies in the complexity of the noun phrase, i.e. compatibility with the category of number or presence of modifiers has been underlined as a possible boundary (see Rueter, 2010, 74–75; cf. Ylikoski (2020)). If a denominal adverbial derivation does not take adjectival or determiner modifiers, there is no syntactic need to distinguish it from other opaque adverbials. On the contrary, it may be noted, syntactic elements that can take this kind of modifiers should be classified according to their syntactic merits. (The term CASE should not be reguarded as a title of estate but as a useful indication of syntactic class membership.)

[7] http://komicorpora.ru

Here, we will further note that according to the SIL Glossary of Linguistic Terms[8] case is defined as a grammatical category determined by the syntactic or semantic function of a noun or pronoun. If we apply this to a regular morphological description of the Komi languages, we may choose to distinguish between derivational endings applied to simple NP heads and inflectional endings applied to complex NP heads. By distinguishing these two varieties of inflection, we can arrive at a syntactic criterion for classifying different types of inflection, whereas the complex NP, which also takes marking for number, might be readily integrated into the enumeration of nominal modifiers, i.e. cases.

For nearly one and a half centuries, the 16 and 1 dependent cases as defined by Castrén (1844) have represented the canonical cases addressed in grammars of the Komi-Zyrian language. The seventeenth case, the comitative, is addressed as a postposition, but all examples of it show it as integrated morphology in the noun. Некрасова (2000) ('The Modern Komi Language', ÖKK), published in 2000 broke with this tradition by including a set of compounded cases (seven).

The 26 cases shown by the latest Komi grammar, may be further augmented to 29 by introducing the PROPRIETIVE, ABESSIVE and LOCATIVE cases, in *-а*, *-тӧм* and *-са*, respectively. The TEMPORAL in *-ся* might, as a function, be simply attributed to the already existing COMPARATIVE case. Similar questions of case definition have been treated by one of the authors, Rueter (2010), where he regards syntactic entity complexity as sufficient grounds for casehood, (see also Ylikoski, 2020).

Tauli (1956), it should be noted, provides numerous references to researchers dealing with affixes, inclusive derivation and case, there does not seem to be any standards for distinction between case and derivation. The Komi-Zyrian PROPRIETIVE refered to also as a *nomen possessoris* suffix *a*, which occurs as a "comitative" (Tauli, 1956), provides a challenge for the those wishing to distinguish Kom proprietive *-а*, comitative *-кӧд* and instrumental *-ӧн*.

Not unlike the PROPRIETIVE, the ABESSIVE, LOCATIVE and even the temporal function of the COMPARATIVE case are almost entirely limited in use to the adnominal range. The ABESSIVE has a predicative counterpart in the CARATIVE *-тӧг*, while the LOCATIVE has a predicative counterpart in the INESSIVE *-ын*. Perhaps this range distinction has also played a part so-called case classification. The adnominal TEMPORAL marker, however, seems to have no morphological counterpart for use in the predicative.

3.3 Accusative versus object marking

One of the dilemmas in Komi morphosyntax is where to introduce the object of a sentence. Actual non-ambiguous accusative forms are attested for pronouns and other NP heads, but the accusative is not the only case used for indicating the object, the ZERO marker strategy is also used for this purpose. Hence, one might readily speak of object marking with the nominative.

Canonic practice in the Komi grammaticography has been to include the nominative, ZERO form, as an additional accusative case form. If we introduce ZERO as an accusative case marker as well, we, essentially, be introducing ambiguity on the text on the analysis level.

Komi is known for its use of singular possessive suffixes in the accusative for marking different degrees of identifiability; zero, i.e. nominative marking, is also a possibility. When we also have the full syntactic dependency tree, the ambiguity between nominatives and unmarked accusative is resolved, as the object relation is unambiguously marked and connected to the root verb. The current solution in the morphological modeling has been to resolve all unmarked wordforms as nominatives, and to leave the nominative-accusative distinction into a later step of the analysis. None the less, we recognize this is only one of the various ways this can be analysed, and when the full analysis comes, we essentially have all the information to transform the material to match various existing traditions.

3.4 Nominal morpheme ordering

This section will investigate the ordering of morphological constituents typically associated with nominals and convey meaning associated with the categories of number, possession and case.

In initial collaboration with FU-Lab, a singular set of morpheme ordering was adopted for each individual combination of possessor & case marking. Hence, it was determined that the word form *батьӧйлӧн* « бать-ӧй-лӧн 'father.N-PxSg1-Gen' featuring the *ӧй* marking for the first person singular possessor could be distinguished from the possessive suffix *ым* in *гортӧдзым* « горт-ӧдз-ым 'home.N-Ter-PxSg1' on the basis of complementary distribution, i.e. there was no need to label the possessive suffixes as separate entities.

[8] https://glossary.sil.org/term/case

In later development, however, a different issue was observed in which case and possessive formatives might show varied ordering. Although this phenomenon is not as prevalent as in the Meadow and Eastern Mari language (cf. Luutonen (1997)), it did merit recognition and distinction for the facilitation of further resource.

The distinguishing tags strategy implemented for Meadow Mari and Hill Mari has been adapted for use with Komi-Zyrian with two tags. One tag indicates segment ordering where the possessive marker precedes the case marker (+So/PC), and the other indicates the case marker precedes the possessive marker (+So/CP), e.g. *кӧзяиныслань* ← кӧзяин-ыс-лань ’owner.N-PxSg3-Apr’ *кӧзяинланьыс* ← кӧзяин-лань-ыс ’owner.N-Apr-PxSg3’.

In addition to this relatively infrequent type of ordering variation of cases versus possessive suffixes, there also appears to be use of the accusative possessive suffix markers for second *-мӧ* and third *-сӧ* person on noun and adjective phrase heads, where the accusative case would not be syntactically compatible. In fact, these same endings are found in connection with other parts of speech as well. It has been maintained that these morphological constituents convey discourse meaning, but there is still much to investigate and establishing tagging practices for these features will contribute to better research materials in the future.

3.5 Numeral derivations

In Komi, numerals are regularly derived to form subgroups in cardinals ZERO, ordinals *-ӧд*, distributives *-ӧн*, iteratives *-ысь*, ordinal iteratives *-ӧдысь* and distributional iteratives *-ысьӧн* (Rueter et al., 2020c). As such, it is often novel or even confounding that we find the syntactic adverbial role found across languages is attributed to a regularly derived adverb *кыкысь* ’twice’ on the Komi side, on the one hand, and a noun phrase *fifty times* ’ветымынысь’, on the other.

Like other adnominal modifiers, it should be noted, numerals may also be promoted to NP head position in instances of contextually motivated ellipsis.

4 Development plan

We have recently moved into primarily data-driven development practice for Komi, where new lexicon and morphology is described primarily based on gaps we find through analysed language materials. At the same time we have developed further tests to check the validity of the output, and in the long term these approaches naturally will live on in parallel. Needless to say, using more natural texts has also forced us to take into account more spoken language and dialect phenomena, which moves the work into quite new directions, which we have already discussed partially above.

After reporting our experiments with the written corpus data, we discuss our plan to integrate the dialectal materials and tags better to the currently discussed Komi analyser.

4.1 Developing on the basis of unrecognized words and word forms

From a corpus of 1,415,210 unique word forms (2020-11-11) 520,180 were not recognized by the analyzer. Aside from the Russian words, apparently from quoted text, and words written entirely in upper case, the most frequent words not to be recognized by the FST seem to all involve hyphens. The use of hyphenation is best illiustrated by *Рытыв-Войвыв* the preposed modifier for direction ’north northwest’ (1377 times), a drawn out pronunciation *Но-о* ’Well-l’ (1177 times), and the orthographic practice of adding *-мӧд* ’another’ in *здук-мӧд* ’yet another moment’ (942 times).

Since over a third of the unique word forms had gone unrecognized, a strategy was developed for improving the model. This would be carried out for nominals initially and subsequently verbs. As described below, a very large portion of unrecognized forms involved various plurals. How they were dealt with is described below, as it illustrates well the challenges we have encountered and their possible solutions.

In the Komi-Zyrian morphology there are two separate plural markers associated with nominal declension. One is the NP plural marker *яс* and the other is the copula complement plural marker *ӧсь*. 20604 unrecognized word forms ended in *яс*, and in 11441 of these the plural marker was preceded by a Cyrillic hard sign ъ. This number was was further delimited by removing all instances of hyphenation and *v* followed by Cyrillic hard sign and word-final *яс*. Where the hyphen may have meant compound words for simple hyphenation in the text, the removal of *v* meant we could automatically avoid the problem of determining whether the word stem contained the notorious *l/v* variation or not. Our resulting figure was 8766.

After entering 15,101 new stems the number of unrecognized unique word forms dropped to 422,227, which was nearly a nineteen per cent improvement over the previous 520,180. In the future we plan to go further through the frequency list of unknown word forms and improve the analyzer so that individual yet frequent phenomena is adequately described and addressed.

4.2 Treatment of dialectal elements

Currently the FST is designed so that dialectal elements are recognized, but they come with an additional error or dialect tag which prevents them being suggested in tools such as spellcheckers. We have also experimented with approaches where Zyrian, Permyak and Russian analysers are run on top of one another, so that unknown forms may be captured by one of the systems with appropriate language tags returned. Since some Zyrian dialectal phenomena is also present in Permyak standard language, already this solution helps to improve the coverage.

Eventually, however, we consider it important that the analyser could capture nuances of individual dialects. In principle this could be accompanied with dialect specific tags, but this approach is also problematic. Many of the features are not strictly found in singular dialects, but cover larger regions. At the same time the speech of any individual is not necessarily limited to any specific variety. Moreover, we believe that further research in Komi dialect isoglosses may be necessary to exactly point for each feature where they definitely occur. Some rough areal boundaries, however, are well known and clear cut, which would make some areal tags potentially useful.

Features that currently are not included are especially those found from southern and eastern Zyrian dialects, mainly because nobody has attempted to use an FST with those varieties yet. We must also recognize that Permyak and Zyrian dialects overlap in their features in various ways, and especially the creation of infrastructure that handless all Komi varieties and both standards remains a challenge.

5 Future directions and Conclusions

In recent years many neural network based approaches have been becoming popular and also shown good results. In a recent study by Pirinen (2019b) the neural models were better than the traditional rule-based approaches for Finnish. Our team is always following new developments of the field, but we also believe that different approaches can be successfully combined.

We already see studies emerging where a neural network has been used to learn to generate predictions from an FST (Hämäläinen et al., 2021). Their research is also used the Komi-Zyrian FST presented in this paper. The results were promising and we are eager to see how this ideology of using neural networks and rule-based systems side by side rather than as competing systems plays out in the future. For the NLP pipeline of Komi the most important new developments will be connected to improvements in the dependency parsing side of the analysis, ideally in connection to automatic and rule-based methods of disambiguation. Komi Constraint Grammar has currently focused to disambiguation, and the tagging and parsing sections are largely missing. It remains to be seen what kind of an approach will be the most successful here. At the same time Komi Universal Dependencies treebanks have started to be large enough that their further modeling with deep learning starts to be an attractive and possibly fruitful task.

Komi texts are also present in many different orthographies, and taking all of them into account is a large and important task (Rueter and Ponomareva, 2019). Since the corpora of Latin Komi texts are also now available[9], the future for these lines of research is exciting and promising. This also connects to various transcription systems used in linguistic publications and text collections: these materials should be republished in the contemporary orthography in order to make them maximally useful for the language communities themselves.

Yet another future task is to provide access to the multilingual Komi lexicon the FST is based on in a form that is truly accessible and openly available. One solution could be to use online dictionary editing platforms, which are strongly linked to the FST development work, and thereby benefit it directly (Alnajjar et al., 2020b). These lexicons have already been published in Zenodo (Rueter et al., 2020b), and already their earliest version has been published in print (Rueter, 1995). Thereby the work described here in various ways continues an already 25 years old progress at morphological modeling of the Komi language, and explores new ways to connect various threads of existing work to one another, especially in ways that takes into account the tech-

[9] http://latina.komicorpora.ru/

nological and practical changes that these decades have shown. We believe this line of investigation of the Komi language will boldly continue the next 25 years, but also hope the reports of how the work progresses will become even more regularly.

We also foresee that further development of the Komi FST will bring new tools to benefit both the public and research communities. Such might be machine translation, on the one hand (Tiedemann, 2021), and translation studies, on the other (cf. Цыпанов, 2021). This, of course, does not close the circle, but merely the ever continuous spiral of development.

Acknowledgements

As described in the study above, this work on Komi has been funded by Alfred Kordelin Foundation and Kone Foundation.

References

Malin Ahlberg, Lars Borin, Markus Forsberg, Martin Hammarstedt, Leif-Jöran Olsson, Olof Olsson, Johan Roxendal, and Jonatan Uppström. 2013. Korp and Karp–a bestiary of language resources: the research infrastructure of Språkbanken. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 429–433.

Khalid Alnajjar, Mika Hämäläinen, and Jack Rueter. 2020a. On editing dictionaries for Uralic languages in an online environment. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 26–30.

Khalid Alnajjar, Mika Hämäläinen, Jack Rueter, and Niko Partanen. 2020b. Ve'rdd. narrowing the gap between paper dictionaries, low-resource nlp and community involvement. *arXiv preprint arXiv:2012.02578*.

Timofey Arkhangelskiy. 2019. Corpora of social media in minority Uralic languages. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 125–140.

Rogier Blokland, Vasily Chuprov, Maria Fedina, Marina Fedina, Dmitry Levchenko, Niko Partanen, and Michael Rießler. 2014–2016. Spoken Komi Corpus. The Language Bank of Finland version.

M.A. Castrén. 1844. *Elementa Grammatices Syrjaenae*. Ex officina typographica heredum Simelii, Helsingforsiae.

Ciprian Gerstenberger, Niko Tapio Partanen, Michael Rießler, and Joshua Wilbur. 2017. Instant annotations: Applying NLP methods to the annotation of spoken language documentation corpora. In *International Workshop for Computational Linguistics of Uralic Languages*, pages 25–36. The Association for Computational Linguistics.

Mika Hämäläinen, Khalid Alnajjar, Jack Rueter, Miika Lehtinen, and Niko Partanen. 2021. An online tool developed for post-editing the new Skolt Sami dictionary. In *Electronic lexicography in the 21st century (eLex 2021). Proceedings of the eLex 2021 conference*, pages 653–664, Czech Republic. Lexical Computing CZ s.r.o.

Mika Hämäläinen. 2019. UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software*, 4(37):1345.

Mika Hämäläinen, Niko Partanen, Jack Rueter, and Khalid Alnajjar. 2021. Neural Morphology Dataset and Models for Multiple Languages, from the Large to the Endangered. In *Proceedings of the the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa 2021)*.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A Pirinen, and Miikka Silfverberg. 2013. HFST a system for creating NLP tools. In *International Workshop on Systems and Frameworks for Computational Morphology*, pages 53–71. Springer.

Jorma Luutonen. 1997. *The Variation of Morpheme Order in Mari Declension*, volume 226 of *Suomalais-Ugrilaisen Seuran Toimituksia*. Suomalais-Ugrilainen Seura, Helsinki.

Sjur Moshagen, Jack Rueter, Tommi Pirinen, Trond Trosterud, and Francis M. Tyers. 2014. Open-source infrastructures for collaborative work on under-resourced languages. The LREC 2014 Workshop "CCURL 2014 - Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era".

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Niko Partanen, Rogier Blokland, KyungTae Lim, Thierry Poibeau, and Michael Rießler. 2018. The first Komi-Zyrian Universal Dependencies treebanks. In *Second Workshop on Universal Dependencies (UDW 2018), November 2018, Brussels, Belgium*, pages 126–132.

Tommi A Pirinen. 2019a. Building minority dependency treebanks, dictionaries and computational grammars at the same time—an experiment in Karelian treebanking. In *Proceedings of the Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019)*, pages 132–136.

Tommi A Pirinen. 2019b. Neural and rule-based finnish NLP models—expectations, experiments and experiences. In *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*, pages 104–114.

Jack Rueter. 2000. Hel'sinkisa universitetyn kyv tujalys' Ižkaryn perymsa simpozium vylyn lydd'ömtor. In *Permistika 6 (Proceedings of Permistika 6 conference)*, pages 154–158.

Jack Rueter. 2010. *Adnominal person in the morphological system of Erzya*. Number 261 in Suomalais-ugrilaisen seuran toimituksia. Suomalais-Ugrilainen Seura, Finland.

Jack Rueter and Erik Axelson. 2020. Raamatun jakeita uralilaisille kielille: rinnakkaiskorpus, sekoitettu, korp [tekstikorpus].

Jack Rueter, Marília Fernanda Pereira de Freitas, Sidney Da Silva Facundes, Mika Hämäläinen, and Niko Partanen. 2021. Apurinã Universal Dependencies treebank. In *Proceedings of the First Workshop on Natural Language Processing for Indigenous Languages of the Americas*, pages 28–33, Online. Association for Computational Linguistics.

Jack Rueter, Mika Hämäläinen, and Niko Partanen. 2020a. Open-source morphology for endangered Mordvinic languages. In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*. The Association for Computational Linguistics.

Jack Rueter, Paula Kokkonen, and Marina Fedina. 2020b. Komi-zyrian-to-x dictionary work. Zenodo data repository, version 0.5.1.

Jack Rueter and Niko Partanen. 2019. Survey of Uralic Universal Dependencies development. In *Third Workshop on Universal Dependencies (UDW, SyntaxFest 2019) Proceedings*. The Association for Computational Linguistics.

Jack Rueter, Niko Partanen, and Larisa Ponomareva. 2020c. On the questions in developing computational infrastructure for Komi-Permyak. In *Proceedings of the Sixth International Workshop on Computational Linguistics of Uralic Languages*, pages 15–25.

Jack Rueter and Larisa Ponomareva. 2019. Komi latin letters, degrees of UNICODE facilitation. *Proceedings of the Language Technologies for All (LT4All)*.

Jack Michael Rueter. 1995. *Komia-angliskӧj-finskoj = Komi-English-Finnish = Komilais-englantilais-suomalainen*. Self-published.

V. Tauli. 1956. The origin of affixes. *Finnisch-ugrische Forschungen*, XXXII(Heft 1–2):170–225.

Jörg Tiedemann. 2021. The development of a comprehensive data set for systematic studies of machine translation. In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*, pages 248–262. Rootroo Ltd., Helsinki. This book has been authored for Jack Rueter in honor of his 60th birthday.

Trond Trosterud. 2004. Porting morphological analysis and disambiguation to new languages. In *SALTMIL Workshop at LREC 2004: First Steps in Language Documentation for Minority Languages*, pages 90–92.

Trond Trosterud. 2004b. Porting morphological analysis and disambiguation to new languages. In *Poster presented at SALTMIL Workshop at LREC 2004: First Steps in Language Documentation for Minority Languages*.

Jussi Ylikoski. 2020. Kielemme kääpiösijoista: prolatiivi, temporaali ja distributiivi. *Virittäjä*, (4):529–554.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, H͡órunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon. Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phương Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, KyungTae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ̀ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Rachele Sprugnoli, Steinh͡ór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. Universal dependencies 2.8.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Р Блокланд, М Рисслер, Н Партанен, А Чемышев, and М Федина. 2014. Использование цифровых корпусов и компьютерных программ в диалектологических исследованиях: теория и практика. In *Актуальные проблемы диалектологии языков народов России: материалы XIV всеросс. науч. конф., посвященной*, pages 20–22.

В.И. Лыткин and Д.А. Тимушев. 1961. *Коми-*

русский словарь. Государственное издательство уностранных и национальных словарей, Москва.

Н.Д. Манова. 1994. *Учимся говорить по-коми. Самоучитель коми языка*. Коми книжное издательство, Сыктывкар.

Г. Некрасова. 2000. Эмакыв. In Г. В. Федюнёва, editor, *Öнія коми кыв, морфология*. Россияса наукаяс академия, Коми наука шöрин, Сыктывкар.

Марина Серафимовна Федина. 2019. Корпус коми языка как база для научных исследований. In *II Международная научная конференция «Электронная письменность народов Российской Федерации: опыт, проблемы и перспективы» проводится в рамках реализации Государственной программы «Сохранение и развитие государственных языков Республики Башкортостан и языков народов Республики Башкортостан» на 2019–2024 гг. Ответственный редактор: Ахмадеева АУ*, page 45.

Евгений Александрович Цыпанов. 1992. *Коми кыв: самоучитель коми языкаю*. Коми кн. изд-во, Сыктывкар.

Йöлгинь Цыпанов. 2021. Питирим Сорокинлысь «a long journey» небöг комиöдöмын шыбöльяс. In Mika Hämäläinen, Niko Partanen, and Khalid Alnajjar, editors, *Multilingual Facilitation*, pages 94–103. Rootroo Ltd., Helsinki. This book has been authored for Jack Rueter in honor of his 60th birthday.

Василий Пантелеймонович Чупров. 2018. Электронный корпус ижемского диалекта коми языка как ресурс для исследования речи ижемских коми. In *Говоры Республики Коми и сопредельных областей*, pages 158–170.

Öньö Лав. 2015. Видзам-сöвмöдам коми кыв! *Арт*, (3):135–144.

Author Index