A restricted freedom of choice: Linguistic diversity in the digital landscape

Trond Trosterud
University of Tromsø

Abstract:

The freedom of choosing what language to use in various contexts is restricted by a wide range of non-linguistic factors. One often-overlooked factor is the availability of a digital infrastructure for the languages in question. To put it bluntly: With no keyboard layout available there also will be no texts written. The article looks at different aspects related to minority languages and digital linguistic resources.

Background[1]

Seen from a purely computational point of view, all languages pose the same challenges for computational linguists, and there is no reason to treat languages differently according to the number or social status of their speakers. When it comes to doing language technology in practice, the situation is different. There is no economical demand to make language technology solutions for more than a handful of languages. For most languages the basic tools for making language technology applications are not readily available: there are no large amounts of texts available in electronic format, also reference grammars may be incomplete, if they even exist at all.

The freedom of choosing the language of your will is restricted in many ways. The topic of this article is the restrictions posed by computational resources: Without access to letter images, keyboard layouts, proofing or machine translation tools, information retrieval or synthetic speech software, we are to varying degrees prevented from using the language we want.

Written language

Read

To read you need the letters of the language. With the introduction of the encoding standard Unicode, this issue is in principle solved. Unicode may, with a slight exaggeration, be called the first milestone for printing after Gutenberg. It contains all writing symbols of all living and most dead languages, and auxiliary symbols for most linguistic and non-linguistic processing, transliteration alphabets, Braille, mathematical symbols, chess

[1] Thanks to Elisabeth Scheller for commenting upon an earlier draft of this article.

Nordlyd 39.2:89–104, CC BY-NC 2012 (http://creativecommons.org/licenses/by-nc/3.0/)
Special Issue on Language Contact, Edited by Hilde Sollid
CASTL, Tromsø. http://septentrio.uit.no/index.php/nordlyd/

symbols, etc. Unicode is what makes it possible to publish text in all languages.

Font providers endorse such a generous policy in order to include East Asian writing systems, such as Chinese. But also linguists devoted to making language representation possible were seminal in making Unicode into what it is today.

In order to be of any help, Unicode must still be taken into use. Failing to do so still results in obstacles to minority language usage. The Norwegian census registry (Folkeregisteret) is a case in point. Formally, it allows the Norwegian letters *a-z* and *æøå*, as well as the foreign letters *äéèôöü*. It does not, however, accept the Sámi letters *á* or *čđŋšŧž*. In practice, this means that Sámi parents cannot give their children names such as *Ánde, Behkká, Iŋgá* and *Máret*. Legally speaking, these common Sámi names are thus illegal. The most important letter here is *á*. 63 % of the Sámi first names in the Giellatekno lexicon[2] contain *á* (whereas 9.2% contain at least some other Sámi letters). The lesson learned for Sámi parents is of course that Sámi, literally speaking, has no legal status for the census authorities. This is even more serious, since the letter *á* stands out also in another respect: Contrary to the 6 other non-Norwegian Sámi letters, this letter actually is included in the repertoire of codepage ISO/IEC 8859-1, or Latin 1. In fact, the Norwegian census is based upon the same repertoire, and there are thus no rational reasons (except for the evident low status of Sámi) for blocking the use of á in the official census registry. To illustrate the priority of this issue, the census registry was originally given a transition period 1.1.2011 - 1.1.2020 to enlarge the character repertoire.

The Norwegian company registry (Brønnøysundregistra) is more tolerant than the Census registry, and allows the whole Latin 1 repertoire (thus also á), but it is still restricted to Latin 1, and does not accept the other Sámi letters. The newspaper *Ávvir* can thus be registered under its own name, whereas *Šillju Gatekjøkken Café Karasjok* cannot. Contrary to the ban on the letter á in the official census, the Latin 1 restriction in the company registry is a real problem, and the programming code of the registry will have to be changed in order to migrate to Unicode. The required change is still relatively trivial. The Norwegian company registry is planning to introduce Sámi letters in 2015[3].

---

[2] http://giellatekno.uit.no

[3] "Altinn og Brønnøysundregistrene er for eksempel godt i gang med å forberede overgangen til UTF-8, men vil allikevel ikke kunne tilby samiske tegn før i 2015." Cited from http://www.regjeringen.no/nb/dep/fad/tema/samepolitikk/samiske_sprak/samisk-sprak-og-it.html?id=86947

There are also languages with real difficulties, though, and one of them is Yoruba. The Niger-Congo language Yoruba has 20 million speakers, mainly in Nigeria, Benin and Togo, but still no official support on any operative system. There is work in progress on supporting Yoruba in Linux, though. Consider the non-ASCII Yoruba letters:

(1) á, à, é, è, ẹ, ẹ́, ẹ̀, í, ì, ó, ò, ọ, ọ́, ọ̀, ṣ, ú, ù

These letters have the accents placed right over the letters, and the dots placed right under them. Not all word processors and font providers are able to do this. For some, the accent is higher above the letter when there is a dot below, for others, the dot is not below, but slightly to the right of the letter. As long as the letter receives only one diacritical mark, the result is usually fine. But with two marks on the same letter, most word processors encounter difficulties.

Now, there is a discussion on skipping the diacritics. The motivation for this discussion (computer problems) is ill founded, though. Rather than settling for a bad orthography, the focus should be upon correcting the glyph rendering in the computer programs. Whether all tones always should be shown in all tone languages is another issue. The Norwegian orthography shows tone distinction for its large systematic homonymy pattern (infinitive / neuter), whereas several cases of lexical minimal pairs are not shown in writing (e.g. *gassen* "the gander / the gas"). To what extent suprasegmental opposition should be shown in the orthography is a complicated matter, and all too important to be governed by the rapidly changing condition of text processing software.

The opposite problem (where the correct letters are available, but the users tend to wrong symbols) is represented the case of the South-West African click letters. The Khoi-san languages boast a series of click sounds which are rendered with the following letters: *ʘ, ǀ, ǁ, ǂ, ǃ*. These letters look like symbols already found on a typewriter, but in Unicode they are represented by other characters, as shown in Table 1 below:

Unicode symbol	Hexadecimal value	Ad hoc symbol	Hexadecimal value	Bantu orthography	Click sound
ʘ	0298	Ø	00D8		bilabial
ǀ	01C0	|	007C	c	dental
ǁ	01C1	||	007C, 007C	x	lateral
ǂ	01C2	+	002B		alveolar
ǃ	01C3	!	0021	q	retroflex

Table 1: Click sounds in Unicode and ascii, and in Bantu orthography

Without a proper keyboard, users will write the alveolar and retroflex clicks not with the Unicode characters 01C2 and 01C3, but rather revert to

the ordinary plus sign and exclamation marks (Unicode 002B and 0021). For the alveolar click we can see the difference, but for the retroflex ones, the two glyphs look identical. But as characters they are distinct, and they behave differently in text processing. Consider the San sentence in (2), with (2)a. written with correct characters and (2)b. written with the ad hoc symbols (San is spoken in Namibia and Botswana).

(2) a. Tsií maátsekám ǁlóakas hòásàp ke ǂxam xam-à !árop !naa ǂ'oá tsií ǁ'iip tì laísìpà síí kèrè ǀnoóku náú ǀúrún ǀxáa.

b. Tsií maátsekám ǁlóakas hòásàp ke +xam xam-à !árop !naa +'oá tsií ǁ'iip tì laísìpà síí kèrè ǀnoóku náú ǀúrún ǀxáa.

These letters were designed by linguists equipped with typewriters. In the digital era, text shall not only be human-readable, but also processed by machines. When processing text, we are used to double-click on a word in order to mark, delete or move it. When pre-processing text, machines insert sentence boundaries after punctuation marks such as "!". For text set in narrow columns, words containing punctuation marks and not letters (like the invisible distinction between *!árop* and correct *!árop*, where the correct form contains 5 letters, and the erroneous one contains 4 letters and an exclamation mark) will be divided after the exclamation mark. In order to be able to process text in an efficient way, users should thus type letters rather than punctuation marks. Good text editing programs will then treat them accordingly. In order to do that, they will need keyboard drivers equipped with reference to the click letters.

Alternatively, one might do like the Bantu language Xhosa, and design an orthography without special letters (in this case, *qárop*). Writing *!árop* is in itself not more problematic than writing Scandinavian *æøåäö* or Sámi *áčđŋšŧž*. The principle should in any case be that computers should adjust to humans, and not vice versa. Minority languages also need stable norms. Introducing Bantu letters *q, c, x* etc. in order to resolve the conflict between the two ways of writing the click letters might as well result in a situation with three competing norms instead of the already existing two.

Write

In order to have text to read, someone must write it. This brings us to the topic of keyboards. For speakers of well-equipped languages this may seem as a trivial problem indeed, but this is not the case.

Nama speakers will need a keyboard to write ǂ, *!* instead of +, *!*. The best way of designing a Yoruba keyboard will probably be to have one non-spacing key ("Dead key") for each diacritical mark (acute, grave, dot below), probably also separate dead keys for the combinations acute and

dot below and grave and dot below. In this way, a letter with 2 diacritical signs will require 3 keystrokes (letter key + modifier key + dead key) instead of 5 (letter key + (modifier key + dead key) x 2).

For only a small fraction of the more than 3000 written languages of the world may the speakers actually turn on a computer, select the appropriate keyboard, and start writing. Table 2 shows the number of out-of-the-box keyboards (i.e. keyboards pre-installed in the operating system) on 3 different platforms (2004 (2011*)) some years ago.

Operative system	Keyboard layout	Graphical interface
Windows XP	51	33
Mac OS X	78*	-
Linux KDE	-	88

Table 2: Out-of-the-box localisation on major operative systems

Looking at the languages behind the numbers, we get the picture shown in Table 3. The largest languages without support are shown to the left, and the smallest languages with such support are shown to the right.

Largest lgs not support out-of-the-box				Smallest lgs with basic support or more			
Rank	Speakers	Name	Country	Rank	Speakers	Name	Country
26	41.0	Bhojpuri	India	2108	0.014	Inuktitut	Canada
33	30.0	Siraki	Pakistan	1971	0.017	North Sámi	Nordic
35	24.0	Maithili	India	1752	0.022	Cherokee	USA
37	23.0	Oriya	India	1344	0.047	Greenlandic	Greenland
39	22.0	Burmese	Myanmar	1343	0.047	Faroese	Denmark
40	22.0	Hausa	Nigeria	1304	0.050	Maori	NZ
44	20.3	Awadhi	India	991	0.940	Gaelic	Scotland
47	20.0	Yoruba	Nigeria	601	0.250	Icelandic	Iceland
51	17.0	Sindhi	Pakistan	517	0.330	Maltese	Malta
53	16.0	Nepali	Nepal	407	0.500	Breton	France
55	15.0	Amharic	Ethiopia	370	0.580	Welsh	UK
59	13.7	Assamese	India	292	0.910	Basque	Spain
60	13.0	Haryanvi	India	130	4.000	Georgian	Georgia

Table 3: The largest non-supported and the smallest supported languages

The haves and the have-nots of the linguistic scene are not arbitrarily distributed. The languages with basic IT supports consist of three different groups: languages with official status in an independent country and rich and monolingual speakers, (most) official state languages of India, and minority languages with a strong government backing them up (typically languages of Western Europe, Canada or New Zealand).

Languages with marginal or no IT support are then the remaining 6400 languages. As can be seen from Table 3, the largest languages without support are Indian languages other than the official state languages, and African languages. Further down this list come languages without official status in an independent country, especially in former British and French colonies.

The process of getting this support is not trivial. It may be illustrated with an example in which the present author participated, the one of North Sámi. North Sámi keyboard layout is now included, out of the box, no matter where you buy your computer, from Linux KDE 3.0, Mac OS 10.3, Win XP SP2 onwards. Behind this achievement lies a decade of hard work, involving experts and language users, consensus-seeking conferences among users, standardisation work, (ISO, CEN, national standards), pressure from state administrations upon the OS vendors, and support from the open source movement.[4] At the outset, there were many diverging keyboard layouts available. We compared them to each other, and let letters that had the same positions in all former keyboards keep their positions. Thus, on the left part of the keyboard there was no disagreement. For the right-hand side, all existing keyboards disagreed, and they sacrificed some, but not all, of the non-Sámi Nordic letters *æøåäö*. With the help of text frequency studies, we were able to show that in Sámi text, the most rarely used Sámi characters actually were less used than the letters *ø, å* (in Norway) and *ä* (in Finland), due to the high frequency of foreign names (*Synnøve, Nystø, Näkkäläjärvi*) in Sámi text. We also saw it as a desired option to be able to write the Nordic letters in the same way on the Sámi and the majority language keyboards, respectively. Where former keyboards had replaced some, but not all of the Nordic letters, we chose to keep them. The Sámi letters were then placed according to frequency, with the most common letters occupying the ergonomically best positions. Non-Nordic[5] *q, w, x* were replaced by Sámi letters, all existing layouts had made

[4] The standardisation work was carried out from around 1996 onwards, by Sámi dihtorlávdegoddi (The Committee for Sámi Computer Standardisation), consisting of Audun Lona (leader), Heikki Kangasniemi, Inger Marie Gaup Eira, Roy Amundsen and Trond Trosterud, as well as national standardisation organisations. Michael Everson participated in editing the non-Sámi parts of the keyboard layouts. The keyboard layouts arrived upon may be found at the site http://www.hum.uit.no/a/trond/smi-tastatur.html.

[5] The Swedish letter *x* represents an exception here. Contrary to Norwegian and Finnish, /ks/ is rendered with the letter *x* rather than with *ks* in Swedish. This makes it more cumbersome for Swedish Sámi speakers to use the Sámi keyboard for Swedish. With the sacrifice of the Y key this double use of the keyboard is in any case a lost cause.

this choice already. The replaced letters were kept on the same keys as they originally had, but accessed via modifier keys. So, when the key W gives *š*, modifier key + W gives *w*, etc. The present official keyboard deviates from the one we designed on one point, that of the placement of the letter *ŧ*. Today, it is found on the position of the non-Sámi letter *y*, whereas our solution was to render it as modifier key + T). A result of this change was that one can not type Finnish or Scandinavian text with the Sámi keyboard, this is even more regrettable as the letter *y* is 10 times more common than *ŧ* even in Sámi text, due to Nordic names (Yngve, Jyrki, ...), and the marginal status of the *ŧ* phoneme in Sámi. The unfortunate outcome of the process illustrates the importance of careful analysis of the language in question prior to keyboard design. Finally, care was taken to preserve non-letter symbols (@, §, ', etc.) on the same positions as for the respective national keyboards. Thus, distinct North Sámi keyboards were made for Norway, Sweden and Finland. Similar studies were conducted for the other Sámi languages as well.

The lesson learned from this and similar endeavours is that orthographies and keyboard layouts should be designed according to linguistic and ergonomic principles. We linguists invented these diacritic signs, now we should help the speakers out, and give them the possibility to write their own language. The issue is far from settled, as illustrated above. A case in point is Komi, whose orthography contains two non-Russian Cyrillic letters (*i* and *ӧ*). Of these letters, *ӧ* is the 3rd most common letter in running text, and even *i* is 100 times more common than the non-Komi Russian letter *ы* (and 4 times more common than *ӹ*). Today, these two common Komi letters are produced by shifting to the English keyboard, writing the corresponding Latin letters, and shifting back. Even disregarding the problems caused by mixing two alphabets, the writing process becomes cumbersome and far from optimal. The conclusion is again the same: Do not change the orthography, but change the computer, so that the possibility to choose to write one's mother tongue becomes a real alternative.

Language technology

After the basic prerequisites of reading and writing are in place, the language still has a long way to go. This will become more and more evident as more advanced options for the majority languages are taken into use also by ordinary users.

Basic grammatical analysis and generation

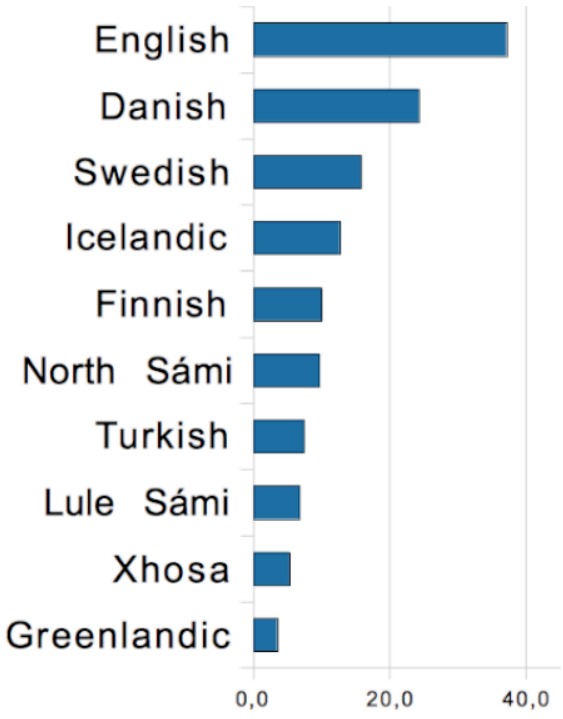
With basic grammatical analysers we shall refer to computational models of the grammar, models that are capable of giving any wordform a

grammatical analysis, and to generate any wordform from the base form (lexeme) and grammatical specification. Thus, such a model should be able to produce *went* from *go+V+Past*, and vice versa.

In many contexts, this in itself is not enough. As was the case for *go*, we may generate the past tense of *walk*, which is *walked*. But the analysis of the form *walked* is more complicated. In addition to being analysed as past tense, the form may also represent the participle, and occur in the same context as the verbform *gone*. The disambiguation of this type of grammatical homonymy is also a central part of the grammatical analysis. As human beings, we are able to distinguish between the two forms of *walked* by looking at the context within which it occurs. Thus, with a subject preceding it we interpret the form as past tense, but an intervening auxiliary verb leads us to believe it is a participle, and we get pairs like *She walked.* / *She has walked.*

Analysers/generators and disambiguators may be made in different ways. For languages with a not too extensive morphology, analysers and generators may be made in the form of lists of pairs of wordforms and grammatical words. Germanic irregular verb morphology may for example be listed. For languages with a richer morphology, this is not practical. The inflectional paradigm of a Finnish verb contains over thousand forms, since each of the participle forms is inflected for case and person-number. On top of this come the forms with clitic particles. Another challenge comes from languages with dynamic compounding, like the languages in Northern Europe. For them, a list-based approach is in practice impossible, since the size of the lexicon will be the product of itself. Languages with non-concatenative morphology will also need separate mechanisms to deal with that.

Disambiguation, or the choice between the two analyses of *walked*, may be done in several ways. A much used way is statistical: Based on a correct-tagged training corpus, the computer is able to find the most probable candidate in any context. This method works best for languages without a rich morphology. For languages with a rich morphology, the likelihood of seeing the same wordform again, and thereby of learning to choose it, is much lower. Cf. Figure 1, which shows the



token/type frequency for some languages.

Figure 1: Token/type frequency (Bible)

More advanced analysis gives rise to more possibilities. Finding the dependency relations within a given sentence makes it possible to abstract over superficial word order, and look for the grammatical relation between the words of the sentence. Enriching the vocabulary with semantic information makes it possible to group concepts together, and give them an adequate treatment. Marking synonyms and hyperonyms gives rise to a more robust information retrieval.

Having such resources becomes more and more important as the demand for language-awareness in language software grows. The outcomes of much of the research done within these areas today are confined to specialist areas and still not visible for the everyday user. Still, language technology applications based upon content analysis are becoming more common. In order to have access to language technology beyond tools for reading and writing, the basic resources must be in place. Shifting the focus from letters and keyboards to grammatical analysers we again find that only a small number of the world's languages have access to such tools. Cf. Table 4, which shows the situation for one of the central resource repositories, the one of the Association for Computational Linguistics[6].

[6] Table 4 is the result of comparing ACL's "List of resources by language" (http://aclweb.org/aclwiki/index.php?title=List_of_resources_by_language) with the Ethnologue's list of languages (http://ethnologue.com).

#	Spkrs	Name	#	Spkrs	Name	#	Spkrs	Name
6	168.0	Bengali	51	17.0	Sindhi	84	8.0	Bundeli
12	75.5	Javanese	53	16.0	Nepali	85	8.0	Ilocano
14	69.0	Telugu	54	15.0	Uzbek	86	8.0	Kazakh
16	61.0	Marathi	57	14.5	Hungarian	87	8.0	Rwanda
17	59.0	Tamil	60	13.8	Azeri (Iran)	88	7.5	Uyghur
18	59.0	Vietnamese	60	13.7	Assamese	90	7.1	Marwari
25	41.5	Gujarati	60	13.0	Haryanvi	91	7.1	Khmer
26	41.0	Bhojpuri	61	13.0	Sinhala	92	7.0	Neapolitan
33	30.0	Siraiki	62	12.2	Igbo	93	7.0	Akan
35	24.0	Maithili	63	12.0	Cebuano	95	7.0	Kurmanji
37	23.0	Oriya	70	10.7	Deccan	96	7.0	Shona
39	22.0	Burmese	70	10.5	Tagalog	97	7.0	Somali
40	22.0	Hausa	72	10.0	Magahi	98	7.0	Tatar
41	21.0	Thai	73	10.0	Zhuang	99	6.8	Azeri (Az.)
44	20.3	Awadhi	76	9.1	Lombard	100	6.5	Xhosa
47	20.0	Yoruba	80	8.2	Chattisgarhi	102	6.3	Luba-Kasai

Table 4: The 48 most commonly spoken languages not found on aclWiki

Machine translation and multilingualism

Machine translation (MT) may be characterised as the ultimate challenge within written language technology. For multilingual societies it also carries a key role: Multilingual text production will in an increasing degree rely upon machine translation, and languages without access to this technology run the risk of not being included in multilingual text production.

MT may be set to conduct two different tasks, commonly called assimilation and dissemination. Assimilation refers to the task of translating foreign text into a language the reader may understand, in order to understand the content, whereas dissemination refers to the task of producing your text in the target language. The tasks are different, and they put different demands upon the system.

At the outset, one would think that bilingual members of a minority language community had no need for an assimilation system. Contrary to the majority, they understand both languages, and will in most cases prefer to read the original majority language text, rather than taking the risk of running it through an imperfect machine translation system. But assimilation MT systems do play an important role for such communities, as follows: Without access to MT, the minority language speaker will have to publish his or her message both in the minority and in the majority language, in order to be read by the whole community. Faced with this

task, the pragmatic choice will often be to write and publish in the majority language only, in order to avoid the burden of producing the text twice. This will evidently have negative consequences upon minority language text production. With access to a working MT system translating to the majority language, the minority language speaker suddenly has gained the freedom of using his or her own language. To the extent that the majority language speakers are interested in the text, there is always the possibility of running it through the machine translation system, and the burden of bilingualism is thus shifted from the bilingual minority to the majority population. This may also give the minority language press a more central position, as it will become accessible to majority language readers («What do they write about me today?»).

Dissemination systems are harder to build. Whereas assimilation systems may tolerate also less well-formed output (as long as it is understandable and not misleading), dissemination systems must be quite good before a professional translator prefers correcting their output rather than translating manually from scratch. In practice, dissemination systems often coexist with other translation aid, such as translation memory systems and terminology resources, together forming computer-assisted translation (CAT) environments. Text translation is expensive and time-consuming, and the development of MT dissemination thus has the potential of receiving substantial funding.

In many parts of the world, minority languages are *Abstand* languages rather than *Ausbau* languages, in other words, the majority and minority language are often not related, and structurally very different. Circumpolar minority languages, and American and African languages are all morphologically complex. The dominating paradigm within machine translation is the statistical approach. Unfortunately, this approach is notoriously bad at translation into morphologically complex languages, as can be seen from Table 5, taken from Koehn 2005. The table shows the outcome of a fair competition: 11 languages (110 translation systems) were trained on the Europarl corpus, the translations of the meeting minutes of the European parliament. The details behind the evaluation method are not relevant in the present context, suffice is to say that for the commonly used BLEU score, which measures the distance between MT output and a (set of) reference translation(s), the result is better the higher the number is. Note also that BLEU scores always must be seen relative to each other in a given test setting, comparing them to test sets of other texts will be misleading.

Source Language	Target Language										
	da	de	el	en	es	fr	fi	it	nl	pt	sv
da	-	18.4	21.1	28.5	26.4	28.7	14.2	22.2	21.4	24.3	28.3
de	22.3	-	20.7	25.3	25.4	27.7	11.8	21.3	23.4	23.2	20.5
el	22.7	17.4	-	27.2	**31.2**	**32.1**	11.4	26.8	20.0	27.6	21.2
en	25.2	17.6	23.2	-	**30.1**	**31.1**	13.0	25.3	21.0	27.1	24.8
es	24.1	18.2	28.3	**30.5**	-	**40.2**	12.5	**32.3**	21.4	**35.9**	23.9
fr	23.7	18.5	26.1	**30.0**	**38.4**	-	12.6	**32.4**	21.1	**35.3**	22.6
fi	20.0	14.5	18.2	21.8	21.1	22.4	-	18.3	17.0	19.1	18.8
it	21.4	16.9	24.8	27.8	**34.0**	**36.0**	11.0	-	20.0	**31.2**	20.2
nl	20.5	18.3	17.4	23.0	22.9	24.6	10.3	20.0	-	20.7	19.0
pt	23.2	18.2	26.4	**30.1**	**37.9**	**39.0**	11.9	**32.0**	20.2	-	21.9
sv	**30.3**	18.9	22.8	**30.2**	28.6	29.7	15.3	23.9	21.9	25.9	-

Table 5: BLEU scores for the 110 translation systems trained on the Europarl corpus (Koehn 2005)[7]

The important point in this context is to look at the results for Finnish as a target language. Finnish is the only morphologically complex language in the set, and also the one that stands out with markedly worse results as target language, and a BLEU score only half as good as for the other languages. With one deviant exception, Finnish also shows the worst results as a source language, although here the differences are far smaller. The results are representative: Statistical translation into morphologically rich languages is hard.

Looking again at the task at hand, machine translation into minority languages in order to produce text, we may conclude that Google Translate and similar systems represent a poor starting point. A viable alternative is rule-based machine translation (RBMT), especially rule-based machine translation between closely related languages.

In a setting like the one in Norway or Russia, with several related minority languages, one might select one pivot language (North Sámi in Norway, and e.g. Meadow Mari and Tatar in Russia). Texts may then be translated from the majority language (Norwegian, Russian) into the pivot language, either manually or with the help of a combination of machine translation and translation memory. The resulting text will then be proofread and used as such. This text will then be used as a source text for machine translation from North Sámi to the other Sámi languages, from Meadow Mari to the other Finno-Ugric language, and from Tatar to the other Turkic ones. Similar results may be achieved within language families like e.g. Eskimo-Aleut and Bantu.

[7] The languages in Table 5 are: da = Danish, de = German, el = Greek, en = English, fr = French, fi = Finnish, it = Italian, nl = Dutch, pt = Portuguese, sv = Swedish.

An example showing the potential of such an approach is the Apertium machine translation platform (Armentano-Oller et al 2005, Forcada 2006). Apertium started out as an development of the InterNOSTRUM.com system (a Spanish-Catalan MT system), it was initially funded by the Spanish Ministries of Industry, Tourism and Commerce, of Education and Science, and of Science and Technology. From the outset, the philosophy was to keep all the parts of the system open source (both the translation engine and the linguistic resources), to invite anyone to participate, and to let eventual commercial interests make money on selling services to customers (like tuning the system to special needs) rather than on translation licenses. The resulting research milieu consists of several companies (Prompsit, Eleka, Imaxin|software), universities, researchers, and language activists.

The system is documented by its developers, on a volunteer basis, and the documentation is written as a wiki. The wiki presently has 69 registered authors[8], and 229 users have received right to commit changes to the source files (this right is, to quote one of the key persons, «basically given away as candy», the reason why it is controlled at all is to avoid unserious check-ins. The repository contains 41 stable language pairs, 26 language pairs on a beta stage, and approximately 160 language pairs in an initial stage. The source files are stored in a version control system, so that every change to the files is logged and may be reversed. During the last 4 years (2008-2012) the system has seen an average of 22 revisions a day.

Looking at usage, we see that the language pairs are unevenly used. Almost half of the translated texts are translated into or from Spanish, reflecting the Iberian focus. More surprisingly, the pair Nynorsk - Bokmål makes up for one third of the translations. This success is probably due to Norwegian schoolchildren being more aware of online resources than their teachers, but it still illustrates the potential in grammar-based machine translation of closely related languages for text production purposes.

[8] http://wiki.apertium.org/wiki/Category:Users

Language pair	Texts/week	Percent
Norwegian Bokmål – Norwegian Nynorsk	9623	34,82 %
Spanish – Catalan	4188	15,15 %
Portuguese – Spanish	3466	12,54 %
Spanish – Brazilian Portuguese	1966	7,11 %
Spanish – English	1549	5,60 %
Spanish – Portuguese	1054	3,81 %
English – Esperanto	824	2,98 %
Galician - Spanish	499	1,80 %
Esperanto – English	427	1,54 %
Other pairs	413	14,65 %

Table 6: Weekly traffic, Apertium translations (Forcada et al 2011)[9]

The core Apertium developers are computational linguists with a focus on RBMT. Most other contributors to Apertium are linguists engaged in work on a specific language or language family. For each language pair, work is then conducted as teamwork, with one or more core developers, and linguists doing the language-specific work.

As evident from the size of both the community and the volume of translations, this open network model works, also for a large and complex linguistic task such as machine translation. Since most of the world's languages neither have the text corpora nor the commercial potential needed for other approaches, this approach is a good candidate for linguistic software development.

Language technology as language documentation

Minority languages neither have the resources nor the commercial potential needed in order to achieve language technology tools on a par with English. What they do have is the potential for an alliance between different groups: Grammarians wanting to understand the grammar and lexicon of the languages in question, sociolinguists analysing the present status of the languages in society, and language activists wanting to support their use. For linguistics, languages with few speakers are as interesting as languages with many speakers. Even more so: Languages where you may be a pioneer may be more attractive than more well studied languages. The

[9] Table 6 shows the following languages: nn = Norwegian Nynorsk, nb = Norwegian Bokmål, es = Spanish, ca = Catalan, pt = Portuguese, ptBR = Brazilian Portuguese, en = English, eo = Esperanto, gl = Galician. Note that the table in the original publication quotes the translation direction from Nynorsk to Bokmål, upon personal communication with the authors I am informed that the correct direction shall be from Bokmål to Nynorsk, as shown here.

challenge for language technology is then to unify the two interests, and let the linguists write their linguistic generalisations in a form useful to language technology. It is not obvious to linguists that they should write their grammars in a machine-readable way, but if they do so it will have a double benefit: The machine may check the validity of their rules, and the resulting grammar/analyser will form the cornerstone of a wide array of end user programs, ranging from spell checkers via pedagogical programs to machine translation.

Lexicographic work should be conducted in a structured way, and if corpora are available, they could be annotated by a parser. Thus, the researcher and the language community have common interests. All this points to a new paradigm for linguistic work, with methods taken from open source program development, like the way Linux was developed. In addition to normal academic publishing, this paradigm is characterised by the following: Projects share resources openly: corpora, lexica, grammatical rules, and infrastructure. File sharing is done via version control systems. The systems are documented via open documentation pages, written collectively, like wikis.

Conclusion

There is now a will, and a way, to provide languages with necessary infrastructure. Better grammatical methods make our analysers robust, and interesting both for linguists and the language communities. Without these resources in place, the freedom of choosing the language of your desire remains an illusion.

The message relevant to sociolinguistics is thus to remember the material base for linguistic practice.

References

Armentano-Oller, Carme, Antonio M. Corbí-Bellot, Mikel L. Forcada, Mireia Ginestí-Rosell, Boyan Bonev, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Gema Ramírez-Sánchez, Felipe Sánchez-Martínez 2005. "An open-source shallow-transfer machine translation toolbox: consequences of its release and availability", in Proceedings of OSMaTran: Open-Source Machine Translation, A workshop at Machine Translation Summit X (Phuket, Thailand, September 12--16, 2005). http://www.dlsi.ua.es/~mlf/docum/armentano05p.pdf

Forcada, Mikel L. Open-source machine translation: an opportunity for minor languages. in Strategies for developing machine translation for minority languages (5th SALTMIL workshop on Minority Languages) (organised in conjunction with LREC 2006 (22-28.05.2006)) http://dlsi.ua.es/~mlf/docum/forcada06p2.pdf.

Forcada, Mikel L., Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez and Francis M. Tyers 2011: Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Koehn, Phillip. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. http://homepages.inf.ed.ac.uk/pkoehn/publications/europarl-mtsummit05.pdf.

Novák, Attila 2009: MorphoLogic's submission for the WMT 2009 Shared Task Proceedings of the Fourth Workshop on Statistical Machine Translation, pp. 155–159. http://www.mt-archive.info/WMT-2009-Novak.pdf

Tyers, F. M. and Wiechetek, L. and Trosterud, T. (2009) "Developing prototypes for machine translation between two Sámi languages". Proceedings of the 13th Annual Conference of the European Association of Machine Translation, EAMT09. http://xixona.dlsi.ua.es/~fran/publications/eamt2009a.pdf