Open-Source Infrastructures for Collaborative Work on Under-Resourced Languages

Sjur Moshagen⋆, Jack Rueter†, Tommi Pirinen‡, Trond Trosterud⋆, Francis M. Tyers⋆
⋆UiT Norgga árktalaš universitehta, †Helsinki university, ‡Dublin City University
⋆N-9037 Tromsø, †FIN-00014 Helsingin yliopisto, ‡IE-Dublin 9
sjur.n.moshagen@uit.no, jack.rueter@helsinki.fi,
tommi.pirinen@computing.dcu.ie, trond.trosterud@uit.no, francis.tyers@uit.no

Abstract

In order to support crowd sourcing for a language, certain social and technical prerequisites must be met. Both the size of the community and the level of technical support available are important factors. Many language communities are too small to be able to support a crowd-sourcing approach to building language-technology resources, while others have a large enough community but require a platform that relieves the need to develop all the technical and computational-linguistic know how needed to actually run a project successfully. This article covers the languages being worked on in the Giellatekno/Divvun and Apertium infrastructures. Giellatekno is a language-technology research group, Divvun is a product development group and both work primarily on the Sámi languages. Apertium is a free/open-source project primarily working on machine translation. We use Wikipedia as an indicator to divide the set of languages that we work on into two groups: those that can support traditional crowdsourcing, and those that do not. We find that the languages being worked on in the Giellatekno/Divvun infrastructure largely fall into the latter group, while the languages in the Apertium infrastructure fall mostly into the former group. Regardless of the ability of a language community to support traditional crowdsourcing, there is in all cases the necessity to provide a technical infrastructure to back up any linguistic work. We present two infrastructures, the Giellatekno/Divvun infrastructure and the Apertium infrastructure and show that while both groups of language communities would not be able to develop language technology on their own, using the infrastructures that we present they have been quite successful.

Keywords: crowdsourcing, infrastructure, minority languages

1. Introduction

Crowdsourcing (Howe, 2008; Surowiecki, 2005) is often thought of as being the leveraging of a group (or crowd) of non-experts to perform tasks previously only done by experts. This is exemplified by the Amazon *Mechanical Turk* platform.[1] Researchers assign tasks and pay small amounts for each task completed. When working with small language communities (often in the hundreds of people), there is not a sufficient mass of native speakers to be able to harness the power of the crowd in this way.

In this article we describe another approach to crowdsourcing. By our definition, a crowd is a group of people who are united by an interest in the development of language technology for a variety of ends.

This collaborative work is made possible by well defined and technically supported infrastructures. An infrastructure consists of the following components: a pre-established way of laying out linguistic data in files and directories, conventions for encoding the data, pre-defined tools for working with the data and building products, and documentation for working with the tools. It should also facilitate testing of both data and tools.

1.1. Language community size and morphological complexity

Language technology's equivalent of the elephant in the room is *the word*. Many language technology applications reduces this concept to a list, possibly a list of pairs (*walk, walk:walks, mouse, mouse:mice, ...*). For morphology-rich languages, like for example the circumpolar ones, this approach is a showstopper. In these languages, the word forms are, for practical and partly even theoretical purposes, not listable.

This is even more true considering the language community sizes of the languages described in the article. Whereas it is fully imaginable to get a small fraction of the English speaking world to list all word forms of the English language via a Mechanical Turk type of project, convincing 500 speakers of a morphologically-complex language to do the same for a theoretically and practically much larger list of word forms is impossible. That is, any approach targeting these languages must thus provide an analysis of the words.

1.2. Outline of the article

The remainder of this article is laid out in six sections: The first section discusses the limitations of crowdsourcing especially with respect to community size. The following section looks at the viability of crowdsourcing for a set of languages. The next section describes the two infrastructures, and this is followed by a section describing the crowds who are using these infrastructures. We then describe the end-user tools that are produced within our infrastructures. Finally, we draw some conclusions.

2. Language community size and crowdsourcing

Most of the world's minority languages, and in postcolonial societies even many of the majority ones, receive little or no official support. The exceptions to this generalisation are

[1] https://www.mturk.com/mturk/

typically minorities in Western societies. One example of a minority language for which the majority society practices a positive language policy, is North Sámi. North Sámi has a written tradition dating 250 years back, with the present standard in use only since 1979. Sámi language society consists of approximately 22,000 speakers, it is technologically advanced, literate, well off, online, and eager to see their language in use. Pupils in the core Sámi areas have their whole primary and secondary education with Sámi as the language of instruction, pupils outside these areas typically have Sámi lessons in Sámi, but a large part or even the rest of their education in the majority language. Except for Facebook localisation and an early localisation of the Linux KDE environment, there has so far not been any crowdsourcing projects related to language.

North Sámi is hardly a typical representative of a language of its size. Drawing instead a random equally-sized language from Ethnologue may e.g. give us *Dabarre*, a Cushitic language related to, but not mutually intelligible with Somali. Dabarre is a language without a literary language, and with no online resources. Its speakers are probably not connected to the internet. Dabarre is classified by Ethnologue as VIGOROUS.

3. Investigating crowd-sourcing viability

This section presents the Giellatekno/Divvun and Apertium languages, and compares them with respect to what might be called their crowd-sourcing viability. As a yardstick for such a viability, we use the size of the Wikipedia version for each and every language, and their status according to (Kloss, 1967) concept of *Ausbau* and *Abstand* languages (the former sharing a (recent) origin with the majority language, the latter not).

Wikipedia is the archetypal crowd-sourcing project. Using only open-source software and a web browser, more than 30 million articles have been written in close to 300 languages[2] — all of it by volunteers. The size of a Wikipedia for a given language should thus be a good indicator for whether the language community has the resources and interest to support projects through crowd-sourcing. It is also reasonable to assume that all other projects will have lesser visibility and be lesser known, and thus have a harder time than Wikipedia creating a crowd for their projects. It seems reasonable to assume that if there is no Wikipedia for a language, then it will be very hard to build a crowd for creating important natural-language processing tools.

3.1. Giellatekno/Divvun

The languages being actively developed within the Giellatekno-Divvun (*GTD*) infrastructure are listed in Table 1, together with the Kloss classification (b = Abstand, u = Ausbau, m = Majority), the number of Wikipedia articles, speakers ((Lewis et al., 2013), for the two Mari languages, Moksha and Erzya: (Moseley, 2010)) and articles per speaker for each of them.

Only four languages with a population below 50,000 have any Wikipedia at all. For all four it is true that most of the content has been written by non-native speakers. For the Giellatekno/Divvun languages with a bigger population, none of the Wikipedias has more than 10,000 articles[3]. Looking at the three largest Wikipedias in Table 1, we find the following: Faroese is an Ausbau language with a long literary tradition, an autonomous position and a majority position in its own area. The overwhelming majority of the North Sámi Wikipedia is written by non-native speakers[4]. For Hill Mari, the dominating article genre is articles

Language	Cl.	No. of speakers	WP articles	articles / speaker
Cornish	b	-	2 634	-
Liv	b	15	0	0.00
Pite Sámi	b	20	0	0.00
Northern Haida	b	45	0	0.00
Ingrian	b	120	0	0.00
Nganasan	b	130	0	0.00
Plains Cree	b	160	194	1.21
Inari Sámi	b	300	0	0.00
Skolt Sámi	b	300	0	0.00
Kildin Sámi	b	350	0	0.00
South Sámi	b	600	0	0.00
Lule Sámi	b	2 000	0	0.00
Upper Necaxa Totonac	b	3 400	0	0.00
Veps	b	3 610	0	0.00
Chippewa	b	5 000	0	0.00
Kven Finnish	b	5 000	0	0.00
Inupiaq	b	5 580	168	0.03
Khanty	b	9 580	0	0.00
Chipewyan	b	11 900	0	0.00
North Sámi	b	20 700	7 650	0.37
Nenets	b	21 900	0	0.00
Livvi	b	25 600	0	0.00
Hill Mari	b	36 822	5 119	0.01
Greenlandic	m	50 000	1 602	0.03
Võro	u	60 000	5 141	0.09
Faroese	m	66 000	7 951	0.12
Komi-Zyrian	b	156 000	4 141	0.03
Moksha	b	200 000	1 180	0.00
Buriat (Russia)	b	219 000	907	0.00
Udmurt	b	324 000	3 387	0.01
Erzya	b	400 000	1 636	0.00
Meadow Mari	b	414 211	3 932	0.01

Table 1: Table of the languages under active development supported by the Giellatekno-Divvun infrastructure, and the number of Wikipedia articles and speakers for each of them.

[2] http://meta.wikimedia.org/wiki/List_of_Wikipedias

[3] This is the Wikimedia threshold for getting into the page of number of speakers per article, cf. http://meta.wikimedia.org/wiki/List_of_Wikipedias_by_speakers_per_article

[4] None of the 18 most active writers have North Sámi as their mother tongue, cf. http://stats.wikimedia.org/NN/TablesWikipediaSE.htm

on geographical administrative units[5]. Except for Faroese, the most viable of the Wikipedias in Table 1 thus seem to be Võru, Komi-Zyrian, Meadow Mari and Udmurt, these are also language communities with active language movements. But also these language communities have not been able to make a working-size Wikipedia (cf. footnote 3).

That is, for the core languages of our work, and using Wikipedia as an indicator, it seems to be hard to find a crowd to give substantial input for constructing language-technology resources.

3.2. Apertium

Apertium (Forcada et al., 2011) is a free/open-source machine translation project. Its origin on the Iberian Peninsula is clearly reflected in the language coverage, but apart from that, Apertium is community-driven, and the choice of languages is dependent upon whether there are people willing to put in an effort in order to get them off the ground. It currently has 38 released language pairs, and many more in progress.

In the past, Apertium language pairs have been fully funded — by either governments or companies; partially funded — that is some work done with funding and the remainder voluntary; or totally voluntary.

An example of the latter would be the Spanish–Aragonese language pair. Work on the pair was started by Apertium-developer Jim O'Regan, at the request of Aragonese-speaker Juan Pablo Martínez. After three weeks of initial effort, spread over the course of a year, a final week of concentrated effort lead to the release of the first prototype version, translating from Aragonese to Spanish only. The first bidirectional version was completed after another 6 weeks of work by Juan Pablo, spread over the course of another year. The only available resource at the beginning of this work for Aragonese was the Aragonese edition of Wikipedia and a handful of verb templates on the English edition of Wiktionary. The Aragonese–Spanish dictionary was created by hand, but the Spanish morphological analyser/generator and part-of-speech tagger were taken from the Spanish–Catalan pair. No funding was received from any source towards the creation of the system. However, the main developer did receive a substantial amount of assistance from the Apertium "crowd", and was able to, thanks to the free/open-source nature of Apertium, reuse a non-insignificant amount of previous work on the Spanish side.

Language pairs are often started by an interested speaker of an under-resourced language (such as the case of Aragonese), or by an interested linguist with help from native speakers (as the case of Breton).

It is often the case that crowds overlap. For example, the developers of the resources for Aragonese and Breton are also active in Wikipedia. Given the size of the Wikipedias, it should in principle be possible to find people to work as a crowd on language technology. The Apertium languages can be found in Table 2.

[5]Tests using the "random article" function gave 70% for this type of articles. The article on Marmara Ereğlisi, a town in the Tekirdağ Province in the Marmara region of European Turkey, may serve as a representative example.

Language	Cl.	No. of speakers	WP articles	articles / speaker
Manx	b	-	4 700	-
Aragonese	u	10 000	29 707	2.97
Corsican	u	31 000	6 665	0.22
Scots Gaelic	b	63 130	11 940	0.19
Faroese	m	66 150	7 992	0.12
Nogai	b	87 410	0	0.00
Irish	b	106 210	29 095	0.27
Asturian	u	110 000	19 462	0.18
Breton	b	225 000	47 759	0.21
Icelandic	m	243 840	37 020	0.15
Karakalpak	b	424 000	632	0.00
Kumyk	b	426 550	0	0.00
Maltese	m	429 000	3 045	0.01
Tetum	b	463 500	800	0.00
Welsh	b	536 890	53 627	0.10
Basque	b	657 872	165 988	0.25
Avar	b	761 960	1 124	0.00
Chuvash	b	1 077 420	23 441	0.02
Sardinian	u	1 200 000	3 250	0.00
Bashkir	b	1 221 340	31 714	0.03
Latvian	m	1 272 650	52 746	0.04
Macedonian	m	1 710 670	75 690	0.04
Slovenian	m	1 906 630	139 630	0.07
Occitan	u	2 048 310	86 470	0.04
Mongolian	m	2 373 260	12 001	0.01
Kyrgyz	m	2 941 930	27 093	0.01
Lithuanian	m	3 130 970	163 336	0.05
Galician	u	3 185 000	110 443	0.03
Gilaki	b	3 270 000	6 008	0.00
Afrikaans	u	4 949 410	30 423	0.01
Tatar	b	5 407 550	56 856	0.01
Armenian	m	5 924 320	109 758	0.02
Albanian	m	7 436 990	50 674	0.01
Turkmen	m	7 560 560	4 975	0.00
Belarusian	m	7 818 960	69 359	0.01
Kazakh	m	8 077 770	205 153	0.03
Uzbek	m	21 930 230	127 385	0.01
Indonesian	m	23 200 480	333 536	0.01
Azerbaijani	m	24 237 550	98 359	0.00
Ukrainian	m	36 028 490	485 563	0.01
Bengali	m	193 263 700	28 256	0.00
Arabic	m	223 010 130	260 602	0.00

Table 2: Table of languages under active development supported by the Apertium infrastructure, and the number of Wikipedia articles, speakers and articles per speaker.

3.3. Summing up the crowdsourcing potential of the different languages

As can be seen in Table 1 and Table 2, languages with small or non-existing Wikipedias are either small, or they are Abstand languages. The only instances of Abstand languages among the active Wikipedias in our material are Basque, Tatar, Welsh, Breton and Chuvash, these are all quite large languages. For language communities smaller than hundred thousand speakers, especially for Abstand languages, the normal crowdsourcing effect is unlikely to work.

Whereas Giellatekno-Divvun only has a handful of languages with more than 100k speakers, Apertium has only

a handful of languages with *less than* 100k speakers, and a majority of the Giellatekno-Divvun languages have less than 10k speakers.

4. Infrastructure descriptions

Apertium and Giellatekno-Divvun share a couple of core values: both infrastructures assume a grammar-based approach to language technology to be the primary approach, both rely heavily on the principles of free/open source code, and both focus on non-central languages in the sense of (Streiter et al., 2006). This same paper gives an excellent overview of how to set up a working infrastructure for such languages, and the infrastructures described in this article fit quite nicely with their definition of a «language pool».

In the current Giellatekno/Divvun infrastructure there are about 50 languages. For all of them we can automatically produce the same set of tools, ready to be deployed. The quality of these tools will of course vary with the degree of linguistic development, but from a technical point of view, all languages are equally well supported. In the Apertium infrastructure, the situation is slightly more complicated. Many languages are supported only as part of machine translation pairs. Taking into account these pairs, there are approximately 76 languages supported to some degree. Of these 76 languages, 44 are available as monolingual packages which provide at minimum a morphological analyser for the language, and in the most developed case, also provide a constraint grammar or statistical part-of-speech tagger and an installable spell checker.

The implementation of both the Giellatekno/Divvun and the Apertium infrastructure is quite simple, using a centralised version control system (Subversion[6]) to track changes and handle cooperation and interaction on the file level. To configure and create build files for each language, GNU Autotools[7] are used.

Both offer ready-made templates to linguists and developers of language technology tools, where all the hard technical details are taken care off. They get a boiler-plate template for linguistic resources, and can start off directly working on the grammatical and linguistic issues. They can skip the demanding and time-consuming first stretch of the well-known S curve ((Huchzermeier and Loch, 2001) and (Barraza et al., 2004)), meaning they will immediately see real progress as they work. It also means that there is no need for every language to invent the same wheel over and over again, saving both money, time and frustration.

The shared infrastructure also means that shortcomings within it revealed by the needs of one language, will automatically benefit all languages.

The infrastructures facilitate cooperation across languages as everything is organised the same way. This also encourages cross-lingual cooperation and crowd-sourcing. Several of the projects using these infrastructures cover many languages in parallel.

Being a language pool in the sense of (Streiter et al., 2006) also means that continuity is secured even for languages with too few resources to ensure continuity on their own. A common organisation of files and documentation also means that linguists working on different languages can easily help newcomers getting started on a new language.

4.1. Choice of language technology

Given that the languages described here are morphologically complex, any successful attempt at analysing them must be able to analyse and generate the word forms. In order to do that, we use finite-state transducers. For languages with complex morphophonological processes, we combine the concatenative transducers with morphophonological transducers, thereby making it possible to deal with non-linear phenomena like vowel harmony, consonant gradation (Koskenniemi, 1983).

For syntactic analysis we use *Constraint Grammar* (Karlsson, 1990), a robust bottom-up parser framework that makes it possible to do dependency parsing with precision above 95 % for syntactic function, and above 99 % for part of speech.

4.2. Differences between the infrastructures

From a technical point of view, the Giellatekno/Divvun infrastructure is technology agnostic. For historical and other reasons, it has been built to support the Xerox FST tools (Beesley and Karttunen, 2003), but with parallel support for the free/open-source Helsinki Finite State tools (Lindén et al., 2013) (which are source-code compatible with the Xerox tools). Adding support for a third or fourth type of technology for morphological analysis should be no problem whatsoever, and the same goes for other parts of the language tool set as well as for the end user tools. The differs from the Apertium infrastructure, where only free/open-source tools are supported and relied upon. The agnosticity in the Apertium infrastructure comes from also supporting some statistically-based modules, such as for part-of-speech tagging.

The major difference between the two infrastructure is the number of end user tools supported by them. Whereas Apertium was designed to support one — machine translation — and has been extended to support FST-based spellcheckers, the Giellatekno/Divvun infrastructure has always supported a large number of end user tools. For the Sámi languages, and other languages supported in the Giellatekno/Divvun infrastructure, there is no competition. There is no competition because the language communities are too small for there to be a commercially viable market for any language-technology products. Thus, in order to fully serve the language community, the infrastructure must be able to support all of the tools needed by the community.

To add new features and tools to the languages in the Giellatekno/Divvun infrastructure, it is enough to develop the new feature for one language. When the new feature is ready, it is copied over to a build template, and from there distributed to all languages in one operation. With this system, support for new technologies and new features can easily be added to all languages. This is a variant of what (Streiter et al., 2006) describes as leveraging the pool to get upgrades «for free» even in cases where it would not be motivated for a specific language in itself. This differs

[6] http://subversion.apache.org

[7] http://en.wikipedia.org/wiki/GNU_build_system

from the Apertium method, where each language is developed based on a template, but once the template is copied, changes are only shared by manual copying and merging.

5. Crowds and infrastructure

In this section we try to characterise the groups of people — or the crowds — using the presented infrastructures. The relevant characteristics in this discussion are: paid/unpaid, size (persons/language), and level and type of expertise.

For larger language communities, the crowds consist of a mixture of programmers and language enthusiasts. For all of the languages, and especially for the ones with small language communities, linguists make up an important part of the crowd. One reason for this is that linguists are interested in grammatical analysis of the languages in question, and the linguistic approach makes the projects worthwhile for them.

5.1. The Giellatekno/Divvun crowd

5.1.1. Tromsø

At UiT Norgga árktalaš universitehta the infrastructure and its precursors have been in use from the very beginning of the work on Sámi language technology. It is indeed true that the infrastructure was first developed for the three major Sámi languages in Norway: North, Lule and South Sámi. These language communities vary in size from about 600 to 22,000 native speakers, and none of them have a functioning crowd working on Wikipedia articles — not today, and even much less so when the projects started.

Since the start in the first half of the previous decade, the resources have been developed by native speakers with linguistic education. These have been employed on projects financed through various public funds and institutions and they constitute the first «crowd of experts» using the precursor to the present infrastructure.

Would it have been possible to build a crowd of interested native speakers to help develop these resources? The Giellatekno/Divvun group actually tried a couple of times, and there was genuine interest in both language technology and in our work. But a number of factors caused these attempts to not succeed. One was inexperience, another our lack of understanding of crowd-sourcing and how to make it work in practice. Native speakers often were too occupied with other language-related activities. For several of the candidates the learning curve was too steep, and combined with little to no follow-up afterwards this meant that attendees forgot even the most basic steps in the procedure taught. Often there is also little to no direct feedback (e.g. in the form of seeing your own word available online after the edit). Learning how to master a version control system for submitting changes and edits turned out to be too complex for several of the candidates given the short timeframe.

Nevertheless, a few eager individuals have started to work on other Sámi languages, so that we today cover all the Sámi languages. These individuals are working outside our core group, some at other academic institutions, and some completely on their spare time.

In summary, most of the people working on the Sámi languages are paid, full time workers, native speakers, and except for North Sámi, usually only one person is working actively on any single language.

5.1.2. Nuuk

After a quite expensive — and failed — attempt at making a list-based spellchecker back in 2003, *Oqaasileriffik* (the Greenlandic language secretariat) has since 2005 used the Giellatekno/Divvun infrastructure described here[8]. In 2011 they moved over to the new iteration of the infrastructure, the one presented in this paper. The work has since 2005 involved 7 (mainly 4) people from the Greenlandic language secretariat and 2 people from UiT. Greenlandic was the first language for which we were able to build a spellchecker, *Kuukkinaat* was released in 2006, with the packaging and MS Office integration done by a private company in Finland.

The Greenlandic project has continued using the common infrastructure for the grammatical analysers ever since, but it has chosen other solutions for their practical programs, be it spellchecking, pedagogical programs[9] or online services[10].

This is a perfectly viable way of utilising this infrastructure. The good thing with this solution is that it gives the Greenlandic language secretariat the full control of design and priorities for the end user solution (as for the web services), and that it makes it possible to choose solutions that differ from the other languages when needed (as for the pedagogical programs). Using the common infrastructure for the basic analyser also gives access to the ready-made solutions for them.

The drawback with this solution is that it implies more work for the programmers linked to the Greenlandic project, and that the project is cut off from the synergy effects and possible free rides of the common project.

5.1.3. Pyssyjoki

Kvensk institutt (KI) in Pyssyjoki has since 2012 run a project on Kven language technology, involving 3 employees at KI, two part-time workers at UiT, and one worker at Halti kvenkultursenter.

Kven language technology started out with a 4,000 lemma bidirectional Kven-Norwegian electronic dictionary, written by Terje Aronsen. The dictionary was integrated in the present infrastructure, and paired with a Kven morphological analyser. Still in an initial stage (with a coverage of 71.2 %, measured on a small corpus of 410 words), it is good enough to make the dictionary a reception dictionary[11], allowing the user to click on words in running text cf. also (Haavisto et al., 2013),

The Kven morphological analyser is also the basis for work on interactive pedagogical programs within the *Oahpa* framework (Antonsen et al., 2009). Although not good enough to function as the basis for a spellchecker, the analyser still covers the basic morphological paradigms, and thus make Kven pedagogical programs possible.

[8] http://oqaaserpassualeriffik.org/a-bit-of-history/

[9] http://learngreenlandic.com

[10] http://oqaaserpassualeriffik.org/tools/

[11] http://sanat.oahpa.no/

5.1.4. Helsinki

A Language research funding programme introduced for the years 2012–2016 by the Helsinki-based Kone Foundation is concerned with the retention of a multilingual world. The group at the University of Helsinki has received funding to work on a project of language documentation. The project was initiated to encourage interaction between speakers/users of lesser documented languages and researchers. It involves the construction of morphological parsers for five Uralic languages. The set of languages selected for this project includes Liv (Livonian), Livvi (Olonets-Karelian), Hill Mari, Tundra Nenets, and Moksha Mordvin. The goal was to develop state-of-the-art parsers able to handle extensive inflectional challenges for at least 20,000 lemmas in each of the selected language over a two-year period. At the same time each of the 20,000 lemmas was to be translated into Finnish. With words and ongoing development of both inflection and translation, this small project has been able to utilise several facets of and contribute to the Giellatekno/Divvun infrastructure.

At present (early 2014) the finite-state transducer projects have progressed to the half-way point. Automatically generated reverse-direction dictionaries have also been set up for some of the transducer projects; yet another way to provide access to lesser documented languages.

In Helsinki transducer development coincides with digitisation of 1920–1930 minority Uralic literature at the National Library of Finland[12], and the development of an open-source editor for proof-reading of open-source OCR-ed literature [13]. Transducer descriptions have been used here to enhance text recognition.

5.1.5. Alberta

The cooperation with the University of Alberta in Edmonton is relatively recent, and is thus in a nascent stage. A group of four linguists have started work on Plains Cree, Northern Haida and Dene Suline[14].

For the two first languages, existing dictionaries are being added to the infrastructure, and the grammar is being rewritten in machine-readable form, as a finite-state transducer.

Adding the analysers to the Giellatekno / Divvun infrastructure offers a means of making morphologically-enriched dictionaries[15].

5.2. The Apertium crowd

Members of the Apertium project come from a range of different backgrounds: University researchers in computer science and linguistics, language activists, free-software and language enthusiasts, and students. There is a governing structure in the form of the project management committee,[16] where large decisions are taken democratically, but otherwise this committee takes a largely *laissez faire* approach leaving individual developers to make their own decisions.

The original crowd is based in Alacant in the Valencian Country in Spain. However, the crowd has become increasingly international. Interaction is through fairly low-tech but high productivity tools such as IRC, mailing lists and a Wiki[17]. The project has been working generally with under-resourced languages and communities, rather than endangered-language communities.

As a project, Apertium has participated in the Google Summer of Code and the Google Code-in. The former programme gives students three-month stipends to work on free software during the northern-hemisphere summer. The latter programme offers prizes to school pupils for completing tasks related to the project.[18] These tasks may be programming tasks: implement an algorithm; or linguistic tasks: e.g. lemmatise a wordlist or part-of-speech tag a short text.

5.3. Summary of the crowds and the infrastructure

What we learned form the first attempt at making a Greenlandic spellchecker was that getting a list of word forms from a crowd of language speakers of a morphologically complex language is not going to result in any useful tool. For the languages treated here, the word form is simply not the relevant unit of analysis. What is needed is a system of combining stems, inflectional and derivational affixes, and the set of morphophonological rules to unite them, in short, a grammatical analyser.

Presenting the setup for a morphological analyser to a group of language activists is in itself also not going to result in an analyser. Making grammatical analysers may be achieved by decentralised cooperation, not of language speakers alone, but of different types of experts fulfilling different roles (one of them being the native speakers) in teams working towards a common goal.

6. End-user tools

The Giellatekno/Divvun group have from the beginning focused on proofing tools and language learning. While the type of services and products has been considerably widened, these two are still at the core of the user-oriented activity. For Apertium, the focus has been upon machine translation.

The infrastructures described in this article combine morphological and syntactic parsers with a wide number of end user tools.

6.1. For linguists and researchers

For linguists, the most important tool is the grammatical analysers. Combined with an advanced corpus search interface[19] it is possible to do empirical research, such as distributional studies of syntactic and morphological phenomena.

[12] http://uralica.kansalliskirjasto.fi/

[13] http://ocrui-kk.lib.helsinki.fi/

[14] http://altlab.artsrn.ualberta.ca

[15] http://pikiskwewina.oahpa.no, http://guusaaw.oahpa.no

[16] See for example http://wiki.apertium.org/wiki/Bylaws

[17] http://wiki.apertium.org

[18] Example tasks: http://wiki.apertium.org/wiki/Task_ideas_for_Google_Code-in

[19] http://gtweb.uit.no/korp/

6.2. For language communities

With morphological transducers in place and readily-available bilingual resources, there is a pipeline for creating a wide range of tools: inflecting bilingual dictionaries [20], spellcheckers and morphologically-aware hyphenators[21]. Enriched with syntactic analysis we also are able to make grammar checkers and, with a bilingual dictionary, also machine-translation systems[22].

6.3. For language learners

Most languages dealt with here are inflecting languages. A central part of study is thus mastering the morphological structure of the language. The Oahpa infrastructure (Antonsen et al., 2009) was originally developed for North Sámi, and includes a series of learning programs, including lexical learning, generation of morphological tasks, and open-input dialogue tasks. Oahpa is integrated with the Giellatekno/Divvun infrastructure, so that Oahpa versions for 4 languages are now in use by language learners, and versions for about a dozen additional languages are in the pipeline.

7. Conclusion

We have tried to show that Wikipedia can be a useful indicator of whether it is possible to build a community of crowdsourcing volunteers. We see that for the Apertium languages crowd-sourcing is actually working, whereas it has not been possible for the Giellatekno/Divvun languages. This corresponds quite neatly with the Wikipedia status of those same languages: none of the Giellatekno/Divvun languages have a viable Wikipedia community, whereas most of the Apertium languages do have.

Language technology for morphology-rich languages with few speakers may be done by crowdsourcing of a different kind, by including people fulfilling different roles in a team. With the goal of combining linguistic analysis and functional end-user programs, we have found that finite-state transducers and constraint grammars are effective tools.

For linguists, the possibility of having others write the infrastructure, and themselves concentrate upon linguistic work, while at the same being able to present software to the user community, is clearly an attractive offer. The popularity of the Giellatekno/Divvun infrastructure shows that the possibility of generating a wide range of products while at the same spend the time on working with concrete linguistic problems is attractive enough to really attract linguists to participate in the crowd.

8. Acknowledgements

We would like to thank our colleagues at Giellatekno/Divvun and Apertium, the Norwegian Ministry of Local Government and Modernisation, and all our colleagues from the different cooperating groups.

9. References

Antonsen, L., Huhumarniemi, S., and Trosterud, T. (2009). Interactive pedagogical programs based on constraint grammar. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, number 4 in Nealt Proceedings Series.

Barraza, G., Back, W., and Mata, F. (2004). Probabilistic forecasting of project performance using stochastic s curves. *Journal of Construction Engineering and Management*, 130(1):25–32.

Beesley, K. R. and Karttunen, L. (2003). *Finite State Morphology*. CSLI publications in Computational Linguistics, USA.

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Haavisto, M., Maliniemi, K., Niiranen, L., Paavalniemi, P., Reibo, T., and Trosterud, T. (2013). Kvensk ordbok på nett – hvem har nytte av den? In *Den tolvte konferansen om leksikografi i Norden*. Nordisk konferanse i leksikografi, Oslo.

Howe, J. (2008). *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Publishing Group, New York, NY, USA, 1 edition.

Huchzermeier, A. and Loch, C. H. (2001). Project management under risk: Using the real options approach to evaluate flexibility in r...d. *Management Science*, 47(1):85–101.

Karlsson, F. (1990). Constraint grammar as a framework for parsing running text. In *Proceedings of the 13th Conference on Computational Linguistics - Volume 3*, COLING '90, pages 168–173, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kloss, H. (1967). Abstand languages and ausbau languages. *Anthropological Linguistics*, 7(9):29—41.

Koskenniemi, K. (1983). *Two-level Morphology: A General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, University of Helsinki.

Lewis, M. P., Simons, G. F., and Fennig, C. D. (2013). *Ethnologue: Languages of the World*. SIL International, Dallas, TX, USA, seventeenth edition.

Lindén, K., Axelson, E., Drobac, S., Hardwick, S., Silfverberg, M., and Pirinen, T. A. (2013). Using hfst for creating computational linguistic applications. In *Computational Linguistics*, pages 3–25. Springer Berlin Heidelberg.

Moseley, C., editor. (2010). *Atlas of the World's Languages in Danger*. UNESCO Publishing, Paris, third edition.

Streiter, O., Scannell, K. P., and Stuflesser, M. (2006). Implementing nlp projects for noncentral languages: Instructions for funding bodies, strategies for developers. *Machine Translation*, 20(4):267–289, December.

Surowiecki, J. (2005). *The Wisdom of Crowds*. Anchor.

[20] http://dicts.uit.no

[21] http://divvun.no

[22] http://wiki.apertium.org