Implementing NLP Projects for Non-Central Languages: Instructions for Funding Bodies, Strategies for Developers *

Oliver Streiter
National University of Kaohsiung

Kevin P. Scannell
Saint Louis University

Mathias Stuflesser
European Academy Bozen Bolzano

February 23, 2006; revised October 27, 2006, revised February 14, 2007

Abstract.

This research begins by distinguishing a small number of *central languages* from the *non-central languages*, where centrality is measured by the extent to which a given language is supported by natural language processing tools and research. We proceed to analyse the conditions under which non-central language projects (NCLPs) and central language projects (CLPs) are conducted. We establish a number of important differences which have far-reaching consequences for NCLPs. In order to overcome the difficulties inherent in NCLPs, traditional research strategies have to be reconsidered. Successful styles of scientific cooperation, such as those found in open-source software development or in the development of the Wikipedia, provide alternative views of how NCLPs might be designed. We elaborate the concepts of free software and software pools and argue that NCLPs, in their own interests, should embrace an open-source approach for the resources they develop and pool these resources together with other similar open-source resources. The expected advantages of this approach are so important that we suggest that funding organizations put it as *condicio sine qua non* into project contracts.

Keywords: Minority languages, open-source, free software, software pools

1. Introduction: Central and Non-Central Language Projects – An Analysis of their Differences

1.1. What are NCLPs?

While NLP systems are continuously making progress in terms of accuracy and speed, this improvement is seen mostly for a handful of languages such as English, Japanese, German, French, Russian and Mandarin Chinese. These are the languages which consume the most research funding in NLP and for which most NLP applications have

* We thank the anonymous reviewers for their constructive criticisms and valuable suggestions.

been developed. As systems for these languages become more and more refined, funds invested in NLP research lead to smaller and smaller gains in processing speed and accuracy. This situation contrasts sharply with the needs of a large number of people around the world. While some researchers might work on fancy topics, such as how to modify your web page while talking on your cell phone, many people have no writing system at all for their mother tongue or their language of daily communication. Even when there is a writing system, there may be no adequate keyboard or input method (see, e.g., (Uchechukwu, 2005)) with which to create electronic texts.

Despite these obstacles, of the estimated 6000–7000 spoken languages in the world, at least 1000 have some presence on the Internet,[1] although some, admittedly, for only a short period (Steven Bird, personal communication). This high number reflects not only the pride of people in their language and culture but also people's willingness and need to use their language for communication, education, documentation, and commerce.

For nearly all of these languages, however, there is no support for manipulating electronic documents beyond mere keyboard input. When using a word processor, for example, there are no proofing tools like spell checkers, hyphenation tools, or grammar checkers. In addition, there is rarely support for information acquisition in a native language context, i.e. information retrieval systems, electronic dictionaries, thesauri, or machine translation systems. In the absence of such resources, it is difficult to develop or maintain a coherent and learnable writing system, and this in turn hinders the development of terminology, the drafting or translation of important legal documents (e.g. the Universal Declaration of Human Rights, texts on conflict resolution, etc.), and localization of software interfaces into the given language. These factors compound the economic obstacles which have placed the blessings of digital culture out of the reach of most language communities.

We view languages as existing in a multidimensional vector space of NLP resources, coordinatized in such a way that the small number of languages with extensive NLP resources occupy the center. These *central languages* have a writing system, Unicode support, fonts, spell checkers, information retrieval systems, corpora, a stemmer, tagger, parser, and machine translation systems. The vast majority of languages are, in contrast, *non-central* and lack most if not all of these resources. Though the terminology "non-central" is a bit clumsy, we prefer it to various other choices with more pejorative connotations,

[1] See *Weaving a Web of linguistic diversity*, `http://www.guardian.co.uk/GWeekly/Story/0,3939,427939,00.html`, 2001-01-25, retrieved 2006-09-12.

e.g. "small" "marginal" or "lesser". "Peripheral" has the advantage of echoing the "center–periphery" dichotomy found in Anglo-American postcolonial discourse, but also suggests being of peripheral importance. In any case, it is important to note that these are not new concepts; in particular, V. Berment's terms *τ-langues* and *π-langues* (Berment, 2004) match our notions of central vs. non-central, as do the *high-density* and *low-density* languages of (Maxwell and Hughes, 2006).[2]

One might make these definitions completely precise by providing coordinates that could be computed, or at least estimated, for any given language. For example, a rough visualization of central vs. non-central languages can be obtained by projecting our hypothetical multidimensional space onto certain small-dimensional subspaces that are easily quantified, by considering such measures as (a) the number of bilingual documents available in the XNLRDF database and (b) the number of Internet portals per language in the XNLRDF database (cf. (Streiter and Stuflesser, 2005)). However such precision is not needed in the present paper.

Note that, by design, this definition does not take into account such factors as the number of speakers of the language, its status as an official language, governmental support, its use in education, the literacy rate, the rate of transmission to children, the average income of its speakers, or the probability that it will still be spoken in the 22nd century. In reality, of course, each of these factors has some impact on the relative centrality of a language as measured by our definition, but we will not concern ourselves with these issues per se.

Note also that these characterizations are dynamic in nature. For example, some languages like Amharic (Ethiopia), Irish (Ireland) and Basque (Northern Spain and Southwestern France) are *centripetal*; despite once occupying the periphery, they have been able to build substantial NLP resources and are now situated closer to the center. On the other hand, languages which are not supported by consistent investment in NLP technology are subject to natural *centrifugal* forces (erosion of previously-developed resources, erosion of the language, erosion of positive language attitudes). Examples are languages like Belorussian (Belarus), Kalmyk (Russia) and many indigenous languages of the Americas and Australia. In some cases, closed-source tools were

[2] Another approach is "Technologically Challenged Languages", used by Justus Roux in his presentation at LREC2004, cf. `http://www.lrec-conf.org/lrec2004/doc/presentation/Roux.pdf`, retrieved 2006-10-26. See also *Language Log: UNESCO International Mother Language Day*, `http://itre.cis.upenn.edu/~myl/languagelog/archives/000481.html`, retrieved 2006-10-26, for a fuller list of options.

developed but then eroded through lack of continuing support. For Ladin (Northern Italy), for example, corpora of more than one million words were built in the TALES project and could be queried via a sophisticated web interface. But after the end of the project, the corpora were no longer available, neither online nor for download, despite the fact that the project received a considerable amount of public funding.

Fortunately, most cultures understand the key role that language plays in their society and therefore try to oppose the centrifugal forces through language development programs, of which NLP projects are just one component. Such NLP projects, and particularly non-central language projects (*NCLPs*, as opposed to central language projects, or *CLPs*) are the main object of our study.

1.2. Why Study NCLPs?

But what is special about NLP projects for non-central languages? Can't they just copy what has been done before in CLPs? Obviously not. They often lack money, infrastructure, an academic environment, commercial interest and suitably trained personnel. But nevertheless these languages try hard to get NLP projects off the ground, and, in doing so, run certain risks. Understanding these risks and finding systematic ways to avoid them seems to us critical for the sustainable success of such projects. Unfortunately little has been done in this regard.

The processing of minority languages and non-central languages has been the subject of a series of workshops in recent years.[3] Most of the papers presented at these workshops discuss specific achievements, e.g. an implementation, or the transfer of a technique from central languages to non-central languages, and only a few articles transcend to higher levels of reflection on how NCLPs might be conducted in general; see in particular (Sarasola, 2000; Agirre et al., 2002; Streiter and De Luca, 2003; Díaz de Ilarraza et al., 2003; Berment, 2004). The papers (Krauwer, 1998) and (Krauwer, 2003) propose a Basic LAnguage Resource Kit (BLARK) as a roadmap of tools to be developed for each language. Although providing valuable insights, these papers do not look into the specific conditions under which linguistic resources are developed and maintained within NCLPs.

In this contribution we will therefore first compare NCLPs and CLPs at a schematic level. This comparison reveals differences which affect, among other things, the status of the researcher, the research paradigm to be chosen, the attractiveness of the research for young

[3] For example (LREC, 1998; LDC, 2000; LREC, 2000; LREC, 2002; TALN, 2003; LREC, 2004; TALN, 2005; LULCL, 2005; LREC, 2006).

researchers, and the persistence and availability of the elaborated data, all to the disadvantage of non-central languages. We propose, as a way of alleviating some of the problems inherent in NCLPs, that developed resources be pooled with similar open-source resources and be made freely available. We will discuss, step-by-step, the possible advantages of this strategy and suggest that it is so promising and so crucial to the survival of the elaborated data that funding organizations should put it as *condicio sine qua non* into their project contracts. But first, we start with a comparison of CLPs and NCLPs.

1.3. Comparing CLPs and NCLPs

Competition: Central languages are generally processed in more than one research center, occasionally by multiple groups at a single research center, each working on a different aspect of the language. The different centers or groups compete for funding and thus strive for scientific recognition via publications, grants, or membership in various decision-making bodies, such as editorial boards of journals, program committees for conferences, or standards-setting committees (LISA, EAGLES, etc.). In contrast, non-central languages are generally worked on by individuals, small research centers, or cultural organizations. Direct competition between groups is unusual as long as funding remains marginal. This situation creates a *niche* which protects the research and the researcher from the pressure to conform to established research paradigms. This, without a doubt, is positive. On the negative side however, methodological decisions, approaches, and evaluations may not be challenged by competitive research. This might lead to a complacency which ignores inspiration coming from successful examples of comparable language projects. We will refer to this negative aspect of the niche as *isolationism*.

Funding opportunities: There are commercial demands for CLPs as can be seen from the large investments that corporations like Google and Microsoft are making in NLP projects. The corresponding lack of commercial demand for NCLPs means that there is little chance that large corporations will help shoulder the financial burden of developing linguistic data or tools for a given non-central language. And even in the case that a particular NCLP is able to provide, say, language recognition data or stemming software to a large search engine company, there is no financial incentive for such a company to absorb the cost of integrating and maintaining such tools. Public sector funding from governmental bodies or charitable foundations has also focused squarely on CLPs. As a consequence, most NCLPs are undertaken

with sorely limited resources in terms of linguistic data, labor, and computing power.

Sharing of data, formats, and programs: Language resources for central languages are produced many times in different variants before they find their way into an application or before they are publicly released. As research centers working on central languages compete for funding and recognition, each center hopes to obtain a relative advantage over its competitors by keeping developed resources inaccessible to others. The same phenomenon occurs, of course, with corporations making investments in NLP technology.[4] For non-central languages such a waste of time and energy is unthinkable and resources which have been built once should be freely available. This allows new projects to build upon earlier work, even if they are conducted elsewhere. Without direct competition, a research center should suffer no disadvantage by making its resources publicly available.

Continuity: CLPs overlap in time and create a continuum of ongoing research. Within this continuum, researchers and resources may develop and adapt to new paradigms (exemplary instances of scientific research, (Kuhn, 1962)) or new research guidelines. Indeed, a large part of many ongoing efforts is concerned with tying the knots between past and future projects; data are re-worked, re-modeled and thus maintained for the future. NCLPs, on the other hand, are discontinuous. Often data have to be created *ex nihilo*. For example, creating legal terminology for a language which has not been official until recently means creating legal terminology without having legal texts. And legal texts are difficult to write without legal terminology. The end of a project may force researchers to leave the research center, and can endanger the persistence of the elaborated data. Data are unlikely to be ported to new platforms or formats, and thereby risk becoming obsolete, unreadable, or uninteresting.[5]

[4] The notion that secretiveness yields long-term advantages can be called into question. Compare, for example, the respective advantages gained by Netscape or Sun from releasing resources to the open-source community. In terms of scientific reputation, some of the most frequently-cited researchers in NLP are those who have made their resources freely available, e.g. Eric Brill (Brill tagger), Henry Kucera and W. Nelson Francis (Brown Corpus), Huang Chu-ren and Chen Keh-jiann (Academia Sinica Corpus), George A. Miller and Christiane Fellbaum (WordNet), Thorsten Brants (TnT tagger), Ted Pedersen (NSP collocation identification) and many others.

[5] Reasons for the physical loss of data include: personal mobility (e.g. after a retirement, nobody knows that the data exist, or how they can be accessed or used); changes in software formats (e.g. changes in the format used by backup programs or changes in the SCSI controller that render the data unreadable); changes in the physical nature of external memories (punch card, soft floppy disk, hard floppy disk,

Data format and encoding: In the beginning of the processing of a language there is a danger of producing data in idiosyncratic or ad hoc formats, with the risk that the data will soon be unusable or difficult to process. Before the advent of the Unicode standard, many non-central languages could not be adequately encoded in standard 8-bit encodings (e.g. the ISO 8859 series) and were forced to rely upon alternative, ad hoc approaches. The digitization of Ladin (Northern Italy) in the 1980s and 1990s serves as a representative example. The first electronic texts and dictionary projects were encoded using special "Ladin fonts" which overwrite certain characters in the Latin-1 encoding. These replacement systems are still in use. The result is that the data can only be shown correctly if the "Ladin fonts" are installed. Since they are not installed on many computers, the average Ladin user writes texts which ignore special characters. Several organizations even produced different Ladin fonts, in which the special characters overwrite different Latin-1 characters. As a result, when processing Ladin texts from different sources, they have to be converted to Unicode, and a script first has to guess which characters are to be replaced.

Specialization: CLPs are generally conducted on a scale that allows them to rely on specialists in programming languages, databases, linguistic theories, parsing, etc. Specialists make the CLP autonomous as project-specific solutions can be produced when needed. Specialization is less likely to be found in NCLPs, where one person has to cover a wider range of activities, theories, and tools in addition to administrative tasks. NCLPs thus cannot operate autonomously, and must rely on toolkits and integrated software packages. Choosing the right toolkit is not an easy task, and a poor choice may cause the project as a whole to fail. In any case, for better or for worse, this choice will influence the course of the research more than any insight of the researcher. If a standard program is chosen simply because the research group is acquainted with it, a rapid start to a project might be bought at the price of future dead ends, producing data which are difficult to port or upgrade, or data which do not match the linguistic reality they are intended to describe. Toolkits that conform to open standards, such as XCES (`http://www.cs.vassar.edu/XCES/`, retrieved 2006-02-15), TEI (`http://www.tei-c.org/`, re-

micro floppy, CD-ROM, magnetic tape, external hard disk, USB stick, etc.) and the devices that can read them; hard disk failure (caused by firmware corruption, electronic or mechanical failure, bad sectors); the limited lifetime of storage devices (two years for tapes, 5–10 for magnetic media, and 10–30 for optical media, depending on the conditions of usage and storage such as temperature, light, and humidity); the absence of an event history that documents the life-cycle and the provenance of a resource, especially its relation to other resources (Caplan and Guenther, 2005).

trieved 2006-02-15), TMX (http://www.lisa.org/standards/tmx/, retrieved 2006-02-15), or that have been shown to produce data that can be ported to one of these standards should be preferred over formats that have been developed without linguistic applications in mind. By no means, however, is a standard format absolutely required. For most languages, collections of raw texts or simple wordlists are the resources most urgently needed. In XNLRDF, for example, embryonic spelling checkers and KWIC tools could be created for 1500 writing systems, just using some raw text corpora, cf. (Liu et al., 2006);

Researchers: A more fundamental problem, also stemming from a lack of funding, is the inability of many organizations working on NCLPs to find researchers with adequate training in NLP. Researchers willing to contribute might not be native speakers, and native speakers willing to contribute might have neither computational nor linguistic training. It is thus important to create a collaborative atmosphere between different experts and native speakers, cf. (Csató and Nathan, 2003; Eisenlohr, 2004).

Research paradigms: CLPs are free to choose their research paradigm and therefore frequently follow the most recent trends. Although different research paradigms offer different solutions and have different constraints, CLPs are not as sensitive to these constraints and can cope successfully with any of them. Even more, CLPs are expected to explore new research paradigms as they have the ability to cope with fruitless attempts, time-consuming explorations, and the small or negative gains of a new research paradigm in its initial phase. Indeed we observe that CLPs frequently turn to the latest research paradigm to gain visibility and reputation, despite the fact that shifts in the research paradigm might make it necessary to re-create language resources in another format or conforming to another logical structure. In contrast, NCLPs depend on the *right research paradigm*: NCLPs do not dispose of rich and manifold resources (dictionaries, tagged corpora, grammars, tagsets, taggers) in the same way that CLPs do. The research paradigm must therefore be chosen according to the nature and quality of the available resources and not according to the latest fashion in research. This might imply the use of example-based methods as they require less annotated data (Streiter and De Luca, 2003), or of unsupervised learning if no annotations at all are available. Hybrid bootstrapping methods are another possibility (Prinsloo and Heid, 2005) though they can be unattractive from a scientific point of view because they are almost impossible to evaluate. Young researchers may experience these restrictions as a conflict. On the one hand they have to promote their research, ideally in the most fashionable research paradigm, but on the other hand they have to find approaches compatible with the available

resources. After the dust of a new research trend has settled,[6] however, new research trends are looked at in a less mystical light and it is perfectly acceptable for NCLPs to stick to an older research paradigm if it conforms to the overall requirements.[7] Another intriguing possibility for NCLPs is the potential for developing entirely new research paradigms tailored specifically to non-central languages; see for example the recent work at Carnegie Mellon on elicitation of language data for machine translation between central and non-central languages (Probst et al., 2002).

Model research: Research on central languages is frequently presented both as research on a particular language and research on Language[8] in general.[9] This is particularly true for English.[10] This leads to an enhanced reputation and better project funding for those engaged

[6] The metaphor is from (Somers, 1998).

[7] Although research centers conducting CLPs are free to choose their research paradigm, they may also be committed to one research paradigm, i.e. the one they have been following for years or the one in which they play a leading role. This specialization of research centers to one research paradigm is partially desirable, as only specialists can advance the respective paradigm. However, when these specialized centers do research on non-central languages, either to extend the scope of the paradigm or to access alternative funding, striking mismatches between the paradigm and the resources may be observed. Such mismatches are of no concern to a central language research center, which after all is doing an academic exercise, but they should be closely watched in NCLPs, where such mismatches would cause the complete failure of the project.

To give one example: Recently, RWTH Aachen University, known for its cutting-edge research in Statistical Machine Translation proposed a statistical approach to sign language translation (Bungeroth and Ney, 2004). One year later Morrissey and Way from Dublin City University, a leading agent in Example-based Machine Translation, proposed "An Example-Based Approach to Translating Sign Languages" (Morrissey and Way, 2005). The fact, however, that parallel corpora involving at least one sign language are extremely rare and extremely small is done away with in both papers as if it would not affect the research. In other words, the research builds on a type of resource which almost does not exist, just to please the paradigm.

[8] We use uppercase to distinguish Language as a general phenomenon from language as referring to a specific language, such as Mongolian.

[9] Note, that this claim is open to empirical validation. One could, for example, compare the percentage of central language linguists doing research on non-central languages with the percentage of non-central linguists doing research on central languages.

[10] In a round table discussion at the 1st SIGHAN Workshop on Chinese language processing, hosted by the ACL in Hong Kong in 2000, a leading researcher in Computational Linguistics vehemently expressed his dissatisfaction at being considered only a specialist in Chinese language processing, while his colleagues working on English are considered specialists in language processing. Working on a non-central language thus offers a niche at the price of a stigma which prevents a researcher from ascending to the Olympus of Science.

in CLPs which in turn makes research on central languages increasingly attractive for young researchers. In addition, central languages tend to be used for illustrating Language in textbooks on syntax, semantics, corpus linguistics and computational linguistics, suggesting implicitly to students that research on these languages is more important or more rewarding. NCLPs, on the other hand, represent applied research – at best! NCLPs are less likely to sell their research as research on Language in general. This perceived lack of generality means that research on NCLPs is less likely to be taught at universities. Students then implicitly learn what valuable research is, namely research on central languages applying recent research paradigms.

To sum up, we have observed that CLPs are conducted in a competitive and sometimes commercialized environment. This competition is the main factor which shapes the way CLPs are conducted. In such an environment it is quite natural for research to overlap and to produce similar resources more than once. Not sharing the developed resources is seen as enhancing the competitiveness of the research center, and is not considered to be an obstacle to the overall advancement of the research field: similar resources are available in other places anyway. Different research paradigms can be freely explored in CLPs with an obvious preference for the latest research paradigm or the one to which the research center is committed. Gaining visibility, funding, and eternal fame are not subordinated to the goal of producing working language resources.

The situation of NCLPs is much more critical. NCLPs have to account for the persistence and portability of their data beyond the lifespan of the project, beyond the involvement of a specific researcher, and beyond the lifespan of a format or specific memory device. This is made especially difficult by the discontinuous nature of NCLPs; if data are not reworked or ported to new platforms they run the risk of becoming obsolete or unusable. These risks must be managed in an environment of limited financial support and limited commercial opportunity; refunding a project because of a shift in research paradigms or because of lost or unreadable data is unthinkable. With few or no external competitors, most inspiration for NCLPs comes from CLPs. However, the reasons underlying the choice of a particular research paradigm by a CLP are not the same as for an analogous NCLP. For talented young researchers, such NCLPs are not attractive. They have been trained on central languages and share with the research community a system of values according to which certain languages and research paradigms are to be preferred.

2. Improving the Situation: Free Software Pools

Let us start with what seems to be the most puzzling question, i.e. how can researchers guarantee the existence of their data beyond what can be directly influenced by the researchers themselves? The answer we are proposing is that the data be pooled together with other data of the same form and function and released as free software.[11]

The notion of free software was introduced by Richard Stallman, founder of the GNU project,[12] and refers to freedom, not price. Specifically, users are guaranteed: 0) the freedom to run the program for any purpose, 1) the freedom to study how the program works and adapt it to their needs, 2) the freedom to redistribute copies, and 3) the freedom to modify the program and release the modified version to the public. Note that freedoms 1) and 3) presuppose access to the program's source code, and because of this free software is sometimes referred to as *open-source* software; strictly speaking, this identification is incorrect, as there is a corresponding formal definition of open-source software[13] which is a bit more inclusive.

One of the principal advantages for NCLPs of integrating your resources in a free software pool is that the community maintaining the pool will take care of the data on your behalf, upgrading it to new formats whenever needed. Of course this begs the question, "Why should someone take care of my data concerning an unimportant and probably dying language?" The answer lies in the pool: Even if those people do not care about your data as such, they care about the pool. When transforming resources for new versions they transform all resources of the pool, knowing well that the attractiveness of the pool comes from the number of different language modules it contains. If all language modules have the same format and function and if one module can be transformed automatically, all others might be automatically trans-

[11] We are certainly not the first to advocate this, even in NLP circles; see, e.g., (Koster and Gradmann, 2004), which argues that all languages, central or non-central, should make their "basic linguistic resources" freely available.

[12] See *The GNU Operating System – Free as in Freedom*, `http://www.gnu.org/`, retrieved 2006-10-26.

[13] See *The Open Source Definition*, `http://www.opensource.org/docs/definition.php`, retrieved 2006-10-26.

formed as well.[14] Thus, the more your data resemble other people's data, the more likely your data are to survive.

In addition, by simply making the source code and data underlying your project freely available, you enable other members of your language community to contribute to the project, or to develop their own projects based on the foundation you have provided. It is important to emphasize a relevant sociological aspect of free software here: freely available source code provides the *means* by which members of the community can contribute, but also provides a strong *motivation*, since there is often a spirit of collective ownership of the resources. We have found this to be particularly true of language processing projects, which simultaneously harness the pride many speakers have in their mother tongue. In any case, contributions from the maintainers of the pool together with contributions from volunteers in your own community offer an effective solution to the "continuity problem" for NCLPs discussed above.

In the previous section we recommended selecting toolkits that conform to open standards (TEI, TMX, etc.). While doing so helps with the continuity problem, if done in an otherwise closed-source context this really becomes only a half-measure, since one is still unable to leverage the help offered by the language community and the pool maintainers.

Guaranteeing the availability of data in conditions of discontinuity is particularly important as many NLP resources build upon each other. For instance, bilingual dictionaries can be built on top of monolingual ones, and for many languages it makes sense to build a grammar checker on top of a spell checker. Allowing others to stand on your shoulders helps to create new resources of greater quality. Keeping existing resources closed, on the other hand, might hinder, or completely prevent the development of the next generation of resources.

Another issue worth mentioning is that open-source programs and data provide an effective way to guarantee the reproducibility of research as reported in journal and conference papers, and are therefore an important contribution to the advancement of language technology as a discipline.

[14] We do not know how much of an idealization this is. The Fink project (*Fink*, `http://fink.sourceforge.net/`, retrieved 2006-09-28), which provides easily-installable software packages for Mac OS X, has one maintainer for each package and not for each pool. As a consequence, not all ISPELL modules are available. In the Linux distribution Debian (*Debian – The Universal Operating System*, `http://www.debian.org/`, retrieved 2006-09-28) we again find one maintainer for each resource, though packages without a maintainer are taken over by the Debian Quality Assurance Group.

2.1. Assessing the Quality of Free Software Pools

As there are no seals of approval for software pools, it is important to check the pools and gauge their capacity to port your data into the next century. The following features are relatively easy to check and, taken together, give a reasonable sense of the quality of a given pool.

- If the different resources within the pool are **uniform**, they are more likely to be collectively upgraded or ported, and it is more likely that these ports can be done semi- or fully automatically. Uniformity can best be achieved with simple dictionaries or raw text corpora. Annotated corpora, treebanks, rich dictionaries and grammars for analysis or generation are unlikely to be uniform across many languages. For the developer this implies that one should try to feather one's nest and place simple resources in pools before embarking on more complex projects.

- The pool should be managed by a **community** of developers and users and not by a single person. A collection of free resources created by one person is not an effective pool. In the free software community, developers are especially prone to losing interest in projects and moving on to greener pastures, and so the existence of an organized community means there is only a limited impact to the pool as a whole as individuals come and go. This helps ensure the survival of the data. Searching for the names of the developers and examining the change logs will help distinguish a one-person-show from a true community. Check to see if discussion fora for developers exist.

- The pool should have the resources **mirrored** on a reasonable number of sites. Debian, for example, has a more than 300 mirrors worldwide[15] and Sourceforge has at least 18 mirrors worldwide in addition to mirrors specific to the Sourceforge project.[16] Data are thus safe even if an earthquake or fire renders one mirror and its backups unusable.

- The pool should be as **paradigm-independent** as possible, so that resources will be preserved even if the the paradigm has fallen out of use, especially if the automatic transformation into another paradigm is difficult. A pool for spellcheckers is thus more likely

[15] See *Debian worldwide mirror sites*, `http://www.debian.org/mirror/list`, retrieved 2006-09-28.

[16] See, for example, *Custom Eclipse Builder*, `http://ceb.sourceforge.net/private-properties.html`, retrieved 2006-09-28.

to be carried over into the 22nd century than a pool of HPSG grammars.

- The pool should be **popular**. Popular pools find volunteers to manage and upgrade the resources more easily. The number of downloads a pool has is a strong indicator of its popularity.
- The pool should be **polychromatic**, shining with many instances of a single data type. Dictionary pools should cover many languages, corpora different genres, etc. This demonstrates their attractiveness to developers and their openness to new developments. In addition, polychromatic resources are more likely to be popular with a wide range of end-users and this leads to the recruitment of new maintainers. It also proves that data formats are widely applicable and highlights the professionalism of the maintainers of the pool.
- The pool should still be **maintained**. Check how frequently updates are made available and when the last update was made.

2.2. Examples of Free Software Pools

To facilitate navigation through the jungle of free resources, we list in Tables I through VI some popular and useful resources which can be considered a pool and which could possibly integrate and maintain your data. Because we are primarily concerned with pools in this paper, *these tables are not intended as a complete survey of free software for NLP*. In particular, many useful and popular open-source resources are omitted since they do not fit our notion of a pool. For example, WordNet[17] does not qualify as a pool as it is just a single resource.[18] Similarly, the Brown Corpus is not a pool; you cannot add sections to it, and so we do not list it here either. Finally, there are many powerful engines for parsing and machine translation (Giza++, Collins' parser, etc.) that are open-source, but are not pools and so will not be found in the tables.

In browsing the tables, be aware that not all pools listed here receive our unconditional approval. Some of the pools are clearly suboptimal, and improving their infrastructure would be of general interest, especially to the extent that NCLPs depend on them.

[17] See *WordNet – Princeton University Cognitive Sciences Laboratory*, `http://wordnet.princeton.edu/`, retrieved 2006-10-26.

[18] One could argue that the Global WordNet project (`http://www.globalwordnet.org/`, retrieved 2007-03-13) approximates a software pool, but it appears to be a somewhat loose confederation and much of the included data is not freely available.

The URLs and numbers of mirrors and supported languages in the tables were accurate as of 14 February 2007.

The most common types of pools are relatively simple structured dictionaries, including word lists and bilingual dictionaries. Here we distinguish dictionaries for Office applications (Table I) from more general dictionaries (Table II).

Pooling of corpora is not as common as the pooling of dictionaries. The main reason for this might be that corpora are very specific and document a particular cultural heritage. Pooling them with corpora of different languages, different subject areas, different registers, etc. is only of limited use. Nevertheless there are some computer-linguistic pools which integrate corpora for computational purposes and which therefore might integrate your corpora and maintain them for you. A description of these (mostly very complex) pools is beyond the scope of this paper, but the interested reader might check the following projects: GATE,[19] Natural Language Toolkit,[20] and XNLRDF.[21]

Machine translation may seem to be a particularly remote goal for languages which have almost no electronic texts or no portable keyboard input method. On the other hand, the advantages that machine translation can offer to non-central languages are too great to ignore. For instance, a system that translates English or another central language into a non-central language could be used to generate a vast amount of content (news, blogs, etc.) very quickly for readers that are unaccustomed to getting such content in a native language context. For endangered languages in particular, this could be an important way to raise the profile of the language on the web, and raise its status in the minds of young speakers. See (Forcada, 2006) for a more detailed discussion of these and many other advantages.

There is no denying that development of a robust MT system is a serious undertaking, but we believe that an open-source approach, using for example one of the engines listed in Table VI and leveraging volunteer contributions from the language communities involved, brings this goal within reach in many cases. This is especially true for language pairs that are linguistically close, e.g. the Romance languages of Spain falling under the Apertium project, or Irish and Scottish Gaelic as in (Scannell, 2006). There is also hope that translation from a central language into a non-central language might be especially tractable since

[19] *GATE, A General Architecture for Text Engineering*, `http://gate.ac.uk/`, retrieved 2006-10-22.

[20] *Natural Language Toolkit*, `http://nltk.sourceforge.net/`, retrieved 2006-10-22.

[21] *Homepage of XNLRDF*, `http://140.127.211.214/xnlrdf`, retrieved 2006-10-22.

Table I. Open-source pools for spelling, office, etc.

name	languages	mirrors	description
ASPELL	> 70	> 300	advanced spell checker, standalone or integrated into smaller applications (emacs, AbiWord, WBOSS); `http://aspell.sourceforge.net/`, retrieved 2007-02-14
HUNSPELL	> 10	> 300	advanced spell checker for morphologically rich languages which can be turned into a morphological analyzer; `http://hunspell.sourceforge.net/`, retrieved 2007-02-14
ISPELL	> 50	> 300	spell checker, standalone or integrated into smaller applications (AbiWord, flyspell, WBOSS); `http://fmg-www.cs.ucla.edu/fmg-members/geoff/ispell.html`, retrieved 2007-02-14
MYSPELL	> 40	> 300	spell checker for OpenOffice.org, now subsumed by HUNSPELL; `http://lingucomponent.openoffice.org/`, retrieved 2007-02-14
OpenOffice.org Grammar	> 5		heterogeneous set of grammar checkers for OpenOffice.org; `http://lingucomponent.openoffice.org/grammar.html`, retrieved 2007-02-14
OpenOffice.org Hyphenation	> 30		hyphenation dictionaries in a common format used by OpenOffice.org, LaTeX, GNU Troff, Scribus, Apache FOP, et al; `http://wiki.services.openoffice.org/wiki/Dictionaries`, retrieved 2007-02-14
OpenOffice.org Thesaurus	> 12		thesaurus for use with OpenOffice.org; `http://wiki.services.openoffice.org/wiki/Dictionaries`, retrieved 2007-02-14
STYLE and DICTION	2		tool to improve wording and readability; `http://www.gnu.org/software/diction/diction.html`, retrieved 2007-02-14

existing open-source tools for parsing, word sense disambiguation, etc. of the source language can be brought to bear.

Table II. Open-source pools for dictionaries

name	languages	mirrors	description
FREEDICT	> 50	> 15	simple bilingual translation dictionaries, optionally with definitions and API as binary and in XML; `http://www.freedict.org/`, retrieved 2007-02-14
FreeLing	5		Morphological dictionaries and libraries for tokenization, morphological analysis, POS tagging, etc.; `http://www.lsi.upc.edu/~nlp/freeling/`, retrieved 2007-03-13
JMDict	> 5		multilingual dictionaries in XML, based on word senses, with Japanese as the pivot language; `http://www.csse.monash.edu.au/~jwb/j_jmdict.html`, retrieved 2007-02-14
Papillon	> 8		multilingual dictionaries constructed according to Mel'cuk's Meaning-Text Theory; `http://www.papillon-dictionary.org/`, retrieved 2007-02-14
Wordgumbo	> 60		multilingual dictionaries in flat simple format, *Wordgumbo*, `http://www.wordgumbo.com/`, retrieved 2007-02-14
dicts.info	> 70		open-source multilingual dictionaries edited by volunteers; `http://www.dicts.info/`, retrieved 2007-02-14

3. Strategies and Recommendations for Developers

3.1. From Pool to Resource

Given that the survival of the data depends in part on the uniformity of the pool, it seems perfectly reasonable to first identify interesting pools and develop resources for them instead of developing idiosyncratic resources and then trying to find matching pools. The pools given in Tables I–VI might also be understood as a kind of checklist of resources that need to be developed for a language to be on par with other languages. Frequently the same resources are available in similar pools, e.g. in ISPELL, ASPELL and MYSPELL. This enlarges the range of applications for a single language resource, increasing its visibility and supporting persistence of the data.

Table III. Corpora

name	languages	mirrors	description
Multext	> 7		parallel corpora of Orwell's 1984 annotated in CES with morpho-syntactic information in 10 Central and Eastern European languages, closed project, but potentially accepts new texts; `http://nl.ijs.si/ME/V2/`, retrieved 2007-02-14
OPUS	> 60		parallel texts harvested from translation compendia of various open-source software projects; `http://logos.uio.no/opus/`, retrieved 2007-02-14
Talk Bank	> 9	> 18	Multimodal database of communicative interactions; `http://talkbank.org/`, retrieved 2007-02-14
UDHR	> 300		Universal Declaration of Human Rights; translated into many languages and can be easily aligned to create parallel corpora, new translations can be submitted; `http://www.unhchr.ch/udhr/navigate/alpha.htm`, retrieved 2007-02-14
Zefania Bibles	> 100	> 18	Bibles with XML markup, easy to align; `http://sourceforge.net/projects/zefania-sharp/`, retrieved 2007-02-14

Table IV. NLP analysis

name	languages	mirrors	description
AGFL	> 4		description of natural languages with context-free grammars; `http://www.cs.ru.nl/agfl/`, retrieved 2007-02-14
CHILDES	> 9		Tagging of corpora in the CHAT format; `http://childes.psy.cmu.edu/morgrams/`, retrieved 2007-02-14
Delphin	> 5		HPSG-Grammars for NLP-applications; in addition, various tools for running and developing HPSG resources; `http://www.delph-in.net/`, retrieved 2007-02-14

Table V. Generation

name	languages mirrors	description
KPML	> 10	Systemic-functional grammars for natural language generation; `http://purl.org/net/kpml`, retrieved 2007-02-14

Table VI. Machine Translation

name	language pairs mirrors	description
Apertium	7	open-source shallow-transfer toolbox, originally designed for the Romance languages of Spain (Armentano-Oller et al., 2005); `http://apertium.sourceforge.net/`, retrieved 2007-02-14
Matxin	1	open-source machine translation engine `http://matxin.sourceforge.net/`, retrieved 2007-02-14
OpenLogos	> 4	open-source version of the Logos MT system, enabling new language pairs to be added; `http://logos-os.dfki.de/`, retrieved 2007-02-14

3.2. From Resource to Pool

If there is no pool of free software data that matches your data you can try one of the following approaches: 1) Modify your data so that they can be pooled with other data. This might involve only a minor change in the format of the data which can be done automatically with a script. 2) Make your data available "as is" under a free software license, thereby increasing the chance that others will copy and take care of your data. 3) Create a community which in the long term will develop its own pool. In general, this requires that you separate the procedural components (tagger, spelling checker, parser, etc.) from the static linguistic data, and that you make the procedural components freely available and describe the format of the static linguistic data.

The *Crúbadán* project [22] serves as a good example of the third approach. The project focuses on the development of NLP tools for non-central languages by using web-crawled corpora and unsupervised statistical methods. Native speakers of more than 50 non-central languages, most with little or no linguistic training, have contributed to the project by editing word lists, helping to tune the language models, and creating simple morphological analyzers. More than two dozen volunteers have helped develop new spell checkers for languages that had little or no language technology before the project began.

3.3. Licensing

In any case, once you decide to make your software and data freely available, you have to think about the license and the format of the data. From the great number of possible licenses you might use for your project,[23] we recommend version 2 of the GNU General Public License[24] as most suitable for typical NCLPs. Through the notion of "Copyleft", it ensures that users of your software have the freedom to redistribute it (with or without changes), while at the same time preventing someone from distributing a modified version without sharing the modifications with you. If the modifications are of general interest, you can integrate them back into your software. The quality of your resources also improves because everyone has access to the source code and can find and point out mistakes or shortcomings. They will report to you as long as you remain the primary developer. Without Copyleft, important language data would already have been lost, e.g. the CEDICT dictionary, after the developer disappeared from the Internet.

The GPL is not the only possibility of course, and any approved open-source license will offer your project benefits in terms of continuity, data preservation, and contributions from the community. There are nuances from license to license regarding the extent to which integration with proprietary software is permitted, the extent to which recognition of authorship is required (which may be an important issue among NLP practitioners in academia), and whether the original author's (or

[22] See *Corpus building for minority languages*, `http://borel.slu.edu/crubadan/`, retrieved 2006-06-26.

[23] See *Various Licenses and Comments about Them – GNU Project*, `http://www.gnu.org/philosophy/license-list.html`, retrieved 2006-10-26, for a commented list of software licenses. A list of "approved" open-source licenses is available from *Open Source Initiative OSI – Licensing*, `http://www.opensource.org/licenses/`, retrieved 2006-10-26.

[24] *GNU General Public License – GNU Project*, `http://www.gnu.org/copyleft/gpl.html`, retrieved 2006-10-26.

sponsoring institution's) name can or cannot be used in advertising the software or derived products.

Generally speaking, when you integrate your language-specific data into a free software pool, your contribution can be licensed completely independently of the pool's code base. The ASPELL source code is available, for example, under the LGPL[25] but the dictionaries are available under a variety of licenses (usually GPL or LGPL). There are, therefore, two decisions to be made; you must be satisfied with the licensing terms for your own software as well as the licensing terms for the pool (or pools) into which you are integrating your resources. We believe that the same arguments in favor of free licenses apply equally well to the pool, and so for example if one must choose between integrating your data into a Microsoft-licensed spell checker that cannot be shared freely and an open-source one than can, we recommend the latter.

The case of Irish language spell checking is illustrative in this regard. K. Scannell developed an Irish spell checker and morphology engine in 2000, integrated it into the ISPELL pool, and released everything under the GPL. Independent work at Microsoft Ireland and Trinity College Dublin led to a Microsoft-licensed Irish spell checker in 2002, but with no source code or word lists made freely available. Now, roughly five years later, the GPL tool has been updated a dozen times thanks to contributions from the community, and the data have been used directly in several advanced NLP tools, including a grammar checker and a machine translation system. The closed-source word list has not, to our knowledge, been updated at all since its initial release. Indeed, a version of the free word list, repackaged for use with Microsoft Word, has all but supplanted use of the Microsoft-licensed tool in the Irish-speaking community.

We mention the possibility of licensing your static linguistic data independently of the pool's code base because it may offer some flexibility in situations where one is *required* to integrate with proprietary software (e.g. if Microsoft or another for-profit company is providing the funding and does not wish to release their intellectual property). In cases like this, the underlying linguistic data should be conceptualized, designed, and developed independently of the service components or algorithmic components, and then one can negotiate an arrangement by which the linguistic data are released freely but the algorithmic components remain closed. Morphological analyzers for some non-central languages (e.g. Sámi) have been developed under this kind of licensing

[25] The so-called "lesser" GPL, which is similar to the GPL but permits your code to be linked with non-free software; see *GNU Lesser General Public License*, `http://www.gnu.org/licenses/lgpl.html`, retrieved 2006-10-26.

scheme: open-source lexica and rule sets combined with the closed-source Xerox Finite State Tools. If none of these arrangements are negotiable, then one must proceed under the imposed conditions, but without any expectation that the data developed will be preserved in the long run.

Dual licensing offers another alternative. Instead of assigning different licenses to the static linguistic data on the one hand and the pool's code on the other, dual licensing makes a single package available under two different licenses. Some projects (OpenLogos, for example) offer their software either under an open-source license, or a "commercial license" that allows integration into proprietary products. A slightly different dual-licensing model suitable for end-user applications is to offer a "professional version" of an otherwise open-source package, which offers advanced features or technical support. This way, a revenue stream can be generated in order to support the NCLP, and at the same time keeping the core resources available for development and maintenance by the community. Although the additional revenues generated through the commercial license might be welcomed by NCLPs, one also risks losing resources if the development of linguistic resources shifts from the open source branch to the commercial branch. Note also that a dual-license approach presupposes the existence of a market for software in the non-central language, which is unrealistic in the majority of cases.

One should not be left with the notion that commercial licenses are the only way to generate revenue for NCLPs. Open-source software development has led to new business models in which the revenue is not created by license fees but instead by software tailoring, customer service, training, etc. These business models are particularly well-suited to adoption by NCLPs, which may be in a unique position to offer localization, native language documentation, and training. It may also be possible to bundle certain resources produced by the NCLP (e.g spelling and grammar checkers, hyphenation patterns, search engines) together with open-source packages.

4. Instructions for Funding Bodies

A sponsoring organization which is not interested in sponsoring a specific researcher or research institute, but which has the goal of promoting a non-central language in electronic applications should insist that the resources developed under its auspices be released under an approved open-source license. Indeed, this condition should be made explicit in all project contracts. This is the only way to guarantee that the resources will continue to be maintained even after the lifetime of

the project. An open-source license allows for the sustainable development of language resources from discontinuous research activities, and guarantees that the most advanced version is available to everybody who might need it. We believe that funding organizations, especially governmental bodies, must work to guarantee that all materials developed with their support be made easily accessible after projects are completed. They might, as an added condition, require that data be bundled with a pool of free software resources to guarantee the physical preservation of the data and its widest accessibility.

Such requirements have rarely been imposed or adhered to in the past, and consequently, far too many resources have been created only to be lost on old computers or tapes, or simply forgotten.[26]

Adding to this invisible pile is a waste of time and money. For those non-central languages which are endangered, this is especially critical. One cannot go back in time when data disappear in order to record or re-record the last speaker of a language after his or her death, bring a spell checker to a generation of students once they graduated from school, or digitize a decomposed manuscript.[27]

Some universities, companies, or research institutes, acting in their own economic interest, might lobby against these contract conditions or try to evade them. They might refer to the intellectual property rights they hold on algorithmic components, or they might stress the value of the service provided to end-users, e.g. a search interface to a corpus or a freely-downloadable spelling checker but without the underlying data made freely available. The fundamental points to keep in mind, however, are (1) that if a public body is providing the funding then they should be able to impose the conditions they see fit in the project contract, (2) preserving the results of the project for the long-term ought to be near the top of the list of conditions, and (3) open-source licensing and software pools are the most effective ways of guaranteeing long-term preservation.

In certain countries, where proprietary software dominates the desktop computing landscape, it might also be argued that funding ought to be provided to private companies as a means to getting language processing tools into the hands of the largest possible number of users. In this situation we suggest, as above, that the linguistic data be sepa-

[26] Interestingly, the American *National Institute of Health* formulates for its research grants Data Sharing Regulations for an *"expedited translation of research results into knowledge, products and procedures to improve human health"*, cf. *NIH Data Sharing Policy*, `http://grants.nih.gov/grants/policy/data_sharing/index.htm`, 2006-08-30, retrieved 2006-09-12.

[27] See: *Digital Race to Save Languages*, `http://news.bbc.co.uk/2/hi/technology/2857041.stm`, 2003-03-20, retrieved 2006-09-12.

rated as much as possible from the proprietary services and algorithms, and that the project contract require that the linguistic data be released under an open-source license. As was illustrated with the Irish spelling example in 3.3, this approach can actually result in a tool being *more* widely accessible than a corresponding fully-proprietary solution, even one that is tightly integrated into widely-used packages such as Microsoft Office.

5. Free Software for NCLPs: Benefits and Unsolved Problems

Admittedly, it would be naive to assume that releasing project results as free software would solve all problems inherent in NCLPs. This step might solve the most important problems of data maintenance and continuity, but can it have more than these positive effects? And which problems remain? Let us return to our original list of critical points for NCLPs and see how they are affected by such a step.

Open-source pools create a platform for research and data maintenance which allows one to overcome the isolationism of NCLPs without having to engage in competition. Data are made freely available for future modifications and improvements. If the data are useful they will be handed over from generation to generation. The physical storage of the data is possible through many of the pools listed above, and therefore does not depend on the survival of the researcher's hard disk. The pools frequently provide specific tools for the production of sophisticated applications, and such tools are the cornerstone of a successful project. In addition, by working with these tools, researchers acquire knowledge and skills which are relevant for the entire area of NLP.

For young researchers, this allows their work on non-central languages to be connected with a wider community for which their research might be relevant. Through the generality of the tools, the content of NCLPs might become more appropriate for university curricula in computational linguistics, terminology, corpus linguistics, etc. Also, a well-designed open-source project can attract a large number of enthusiastic volunteers who are willing to perform heroic amounts of volunteer labor of the kind that might be done by paid research assistants or graduate students for CLPs. The open-source web browser Firefox 2.0, for example, has been localized by volunteers into 39 languages. In contrast, the older commercial browser Internet Explorer 6 is available

in 24 languages only and the new Internet Explorer 7 in five languages only.[28]

The discussion above focuses on the advantages that a specific NCLP can gain from an open-source approach. Perhaps more powerful are the *unforeseen* advantages that a given language stands to gain in terms of its overall NLP infrastructure. For example, by simply releasing an open-source ISPELL spell checker in your language (even a simple word list), it is likely that the following resources will automatically be made available, produced entirely by individuals with no particular interest in your language: (1) a version suitable for use with the free word processor AbiWord[29] (2) a port of your word list to MYSPELL, ASPELL, and HUNSPELL formats, which can then be used with OpenOffice.org (3) a version that can be installed for use with the Mozilla Suite or with the Thunderbird mail handler[30] (4) packages for various Linux distributions (Debian, Gentoo,[31] Mandriva,[32] etc.) (5) a port for Mac OS X (Cocoaspell[33]) (6) free web corpora bootstrapped from your word list (from the *Crúbadán* project cited above) (7) a version of Dasher,[34] a free program for keyboardless text entry, trained for your language using these corpora, etc. etc.

Some problems however remain, for which other solutions have to be found. These are:

- Discontinuous research if research depends on project acquisition.
- Dependence on research paradigm. Corpus-based approaches can be used only when corpora are available, rule-based approaches when formally trained linguists participate in the project. To overcome these limitations, research centers and funding bodies should continuously work on the improvement of the necessary infrastructure for language technology (Sarasola, 2000).
- Attracting and binding researchers. As the success of a project depends to a large extent on the researchers' engagement and skills,

[28] See *Mozilla Firefox – Next Generation Browser*, `http://www.mozilla.com/firefox/all.html`, retrieved 2006-10-26, *Internet Explorer 6: Worldwide Downloads*, `http://www.microsoft.com/windows/ie/ie6/worldwide/default.mspx`, retrieved 2006-09-12, and *Internet Explorer 7: worldwide sites*, `http://www.microsoft.com/windows/ie/worldwide/default.mspx`, retrieved 2006-10-26.

[29] *AbiWord*, `http://www.abisource.com/`, retrieved 2006-10-23.

[30] *Home of the Mozilla Project*, `http://www.mozilla.org/`, retrieved 2006-10-23.

[31] *Gentoo Linux News*, `http://www.gentoo.org/`, retrieved 2006-10-23.

[32] *Welcome/Home – Mandriva Linux*, `http://www.mandriva.com/`, retrieved 2006-10-23.

[33] *cocoAspell*, `http://cocoaspell.leuski.net/`, retrieved 2006-10-23.

[34] *Dasher Project: Home*, `http://www.inference.phy.cam.ac.uk/dasher/`, retrieved 2006-10-23.

attracting and binding researchers is a sensitive topic for which soccer clubs provide an illustrative model. Can NCLPs attract top players or are they just playgrounds for talented young researchers who will sooner or later transfer to CLPs? Can NCLPs count on local players only? A policy of building a home for researchers is thus another sensitive issue for which research centers and funding bodies should try to find a solution.

6. Conclusions

Although the ideas outlined in this paper are very much based on introspection, intuition, a very schematic and simplifying thinking, informal personal communications, and personal experience, we hope to have provided clear and convincing evidence that NCLPs have profited, profit, and will profit from joining the free software community. Most of our claims we have made are open to empirical validation and we invite critics to falsify these claims. For those who want to follow this direction, the first and most fundamental step is to study possible licenses and to understand their implications for the problems of NCLPs, such as the storage and survival of data, their improvement through a large community, etc.

Emotional reactions like "I do not want others fumbling with my data" or "I do not want others to make money from my hard work" should be openly pronounced and discussed. What are the advantages of others having my data? What are the disadvantages? We have attempted to address these questions above in order to put to rest the misconceptions and fears that lead to a rejection of free software principles as often as does rational argument. While this is how humans function, it is not how we advance non-central languages.

References

Agirre, E., I. Aldezabal, I. Alegria, X. Arregi, J. M. Arriola, X. Artola, A. Díaz de Ilarraza, N. Ezeiza, K. Gojenola, K. Sarasola, and A. Soroa: 2002, 'Towards the definition of a basic toolkit for HLT'. In: *Proceedings of the Workshop "Portability issues in Human Language Technologies", LREC'02*. Las Palmas de Gran Canaria, Spain, pp. 42–48.

Armentano-Oller, C., A. M. Corbí-Bellot, M. L. Forcada, M. Ginestí-Rosell, B. Bonev, S. Ortiz-Rojas, J. A. Pérez-Ortiz, G. Ramírez-Sánchez, and F. Sánchez-Martínez: 2005, 'An open-source shallow-transfer machine translation toolbox: consequences of its release and availability'. In: *Proceedings of the Open Source Machine Translation Workshop at MT Summit X*. Pukhet, Thailand, pp. 12–16.

Berment, V.: 2004, 'Méthodes pour informatiser des langues et des groups de langues peu dotées'. Ph.D. thesis, Université Joseph Fourier.

Bungeroth, J. and H. Ney: 2004, 'Statistical Sign Language Translation'. In: O. Streiter and C. Vettori (eds.): *Proceedings of the Workshop on Representation and Processing of Sign Languages, LREC'04*. Lisbon, Portugal, pp. 105–108.

Caplan, P. and R. Guenther: 2005, 'Practical Preservation: The PREMIS Experience'. *Library Trends* **54**(1), 111–124.

Csató, E. and D. Nathan: 2003, 'Multimedia and Documentation of Endangered Languages'. In: P. Austin (ed.): *Language Documentation and Description*, Vol. 1. London: SOAS, pp. 73–84.

Díaz de Ilarraza, A., A. Gurrutxaga, I. Hernaez, N. Lopez de Gereñu, and K. Sarasola: 2003, 'HIZKING21: Integrating language engineering resources and tools into systems with linguistic capabilities'. In: O. Streiter (ed.): *Proceedings of the Workshop "Traitement automatique des langues minoritaires et des petites langues", 10e conférence TALN*. Batz-sur-Mer, France, pp. 243–252.

Eisenlohr, P.: 2004, 'Language Revitalization and New Technologies: Cultures of Electronic Mediation and the Refiguring of Communities'. *Annual Review of Anthropology* **33**, 21–45.

Forcada, M.: 2006, 'Open source machine translation: an opportunity for minor languages'. In: B. Williams (ed.): *Proceedings of the Workshop "Strategies for developing machine translation for minority languages", LREC'06*. Genoa, Italy, pp. 1–6.

Koster, C. H. A. and S. Gradmann: 2004, 'The language belongs to the People!'. In: *Proceedings of LREC'04*. Lisbon, Portugal.

Krauwer, S.: 1998, 'ELSNET and ELRA: Common past, common future'. *ELRA Newsletter* **3**(2).

Krauwer, S.: 2003, 'The Basic Language Resource Kit (BLARK) as the First Milestone for the Language Resources Roadmap'. In: *Proceedings of the International Workshop "Speech and Computer", SPECOM 2003*. Moscow, Russia.

Kuhn, T. S.: 1996/1962, *The Structure of Scientific Revolutions*. University Of Chicago Press, 3 edition.

LDC: 2000, 'New Methods for Creating, Exploring and Disseminating Linguistic Field Data'. Chicago, USA:.

Liu, D. Y.-C., S. C.-F. Su, L. Y.-H. Lai, E. H.-Y. Sung, I. ling Hsu, S. Y.-C. Hsieh, and O. Streiter: 2006, 'From Corpora to Spell Checkers: First Steps in Building an Infrastructure for the Collaborative Development of African Language Resources'. In: J. Roux (ed.): *Proceedings of the LREC wordshop "Networking the development of language resources for African languages"*. Genoa, Italy.

LREC: 1998, 'Workshop on language resources for European minority languages'. Granada, Spain:.

LREC: 2000, 'Developing language resources for minority languages: re-usability and strategic priorities'. Athens, Greece:.

LREC: 2002, 'Portability issues in Human Language Technologies'. Las Palmas de Gran Canaria, Spain:.

LREC: 2004, 'First Steps for Language Documentation of Minority Languages: Computational Linguistic Tools for Morphology, Lexicon and Corpus Compilation'. Lisbon, Portugal:.

LREC: 2006, 'Strategies for developing machine translation for minority languages'. Genoa, Italy:.

LULCL: 2005, 'Lesser Used Languages and Computer Linguistics'. Bozen-Bolzano, Italy:.

Maxwell, M. and B. Hughes: 2006, 'Frontiers in Linguistic Annotation for Lower-Density Languages'. In: *Proceedings of the COLING/ACL2006 workshop "Frontiers in Linguistically Annotated Corpora"*. Sydney, pp. 29–37.

Morrissey, S. and A. Way: 2005, 'An Example-Based Approach to Translating Sign Language'. In: A. Way and M. Carl (eds.): *Proceedings of the Workshop on Example-Based Machine Translation, MT Summit X*. Pukhet, Thailand, pp. 109–116.

Prinsloo, D. J. and U. Heid: 2005, 'Creating word class tagged corpora for Northern Sotho by linguistically informed bootstrapping'. In: I. Ties (ed.): *Proceedings of the conference "Lesser Used Languages and Computer Linguistics"*. Bozen-Bolzano, Italy, pp. 97–115.

Probst, K., L. Levin, E. Peterson, A. Lavie, and J. Carbonell: 2002, 'MT for Minority Languages Using Elicitation-Based Learning of Syntactic Transfer Rules'. *Machine Translation* **17**(4), 245–270.

Sarasola, K.: 2000, 'Strategic priorities for the development of language technology in minority languages'. In: *Proceedings of the Workshop "Developing language resources for minority languages: re-usability and strategic priorities", LREC'00*. Athens, Greece, pp. 106–109.

Scannell, K. P.: 2006, 'Machine translation for closely related language pairs'. In: B. Williams (ed.): *Proceedings of the Workshop "Strategies for developing machine translation for minority languages", LREC'06*. Genoa, Italy, pp. 103–107.

Somers, H.: 1998, '"New paradigms" in MT: the state of play now that the dust has settled'. In: F. Van Eynde (ed.): *10th European Summer School in Logic, Language and Information, Workshop on Machine Translation*. Saarbrücken, pp. 22–33.

Streiter, O. and E. W. De Luca: 2003, 'Example-based NLP for Minority Languages: Tasks, Resources and Tools'. In: O. Streiter (ed.): *Proceedings of the Workshop "Traitement automatique des langues minoritaires et des petites langues", 10e conférence TALN*. Batz-sur-Mer, France, pp. 233–242.

Streiter, O. and M. Stuflesser: 2005, 'XNLRDF, the Open Source Framework for Multilingual Computing'. In: I. Ties (ed.): *Proceedings of the conference "Lesser Used Languages and Computer Linguistics"*. Bozen-Bolzano, Italy, pp. 189–207.

TALN: 2003, 'Traitement Automatique des Langues Minoritaires et des Petites Langues'. Batz-sur-Mer, France:.

TALN: 2005, 'TAL et langues peu dotées'. Dourdan, France:.

Uchechukwu, C.: 2005, 'The Igbo Language and Computer Linguistics: Problems and Prospects'. In: I. Ties (ed.): *Proceedings of the conference "Lesser Used Languages and Computer Linguistics"*. Bozen-Bolzano, Italy, pp. 247–264.