Implementing NLP Projects for Non-Central Languages: Instructions for Funding Bodies, Strategies for Developers. ABRIDGED VERSION.

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; abridged February 3, 2010

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 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 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, 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, 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* match our notions of central vs. non-central, as do the *high-density* and *low-density* languages in Hughes and Maxwell....

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....

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 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....

An extended comparison between CLPs and NCLPs follows, treating factors such as: competition among researchers or research groups, funding opportunities, research specialization, project staffing, research paradigms, project continuity, and sharing of data. Only the last two discussions are included below, as the most relevant to the topic of Open Linguistic Data.

1.3. Comparing CLPs and NCLPs

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.[1] For non-central languages

[1] 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. Netscape-based browsers like Firefox outperform their competitors such as Internet Explorer and Opera, and an open, XML-based file format such as one finds in OpenOffice.org is going to be adopted in the next release of Microsoft Office. In terms of scientific reputation, some of the most frequently-cited researchers in NLP are those who have made their resources freely available, including dictionaries and corpora, e.g. Eric Brill (Brill tagger), Henry Kucera and W. Nelson Francis (Brown Corpus),

such a waste of time and energy is unthinkable and resources which have been built once should be made freely available. This allows new projects to be built 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, 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, remodeled and thus maintained for the future. NCLPs, on the other hand, are discontinuous. This threatens the continuity of the research, forces 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.

...

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

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.

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 CLPs 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.

The notion of free software was introduced by Richard Stallman, founder of the GNU project, 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 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 transformed as well. 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 also 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.

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 an **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-man-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 and Sourceforge has at least 18 mirrors worldwide in addition to mirrors specific to the Sourceforge project. 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 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 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-VI some popular and useful pools which could possibly integrate and maintain your data... *tables omitted in abridged version.*

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.

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 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, we recommend version 2 of the GNU General Public License[2] 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 im-

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

proves 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.

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 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. The second author 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 scheme: open-source lexica and rule sets combined with the closed-source Xerox Finite State Tools. If such an arrangement is not negotiable, then one must proceed under the imposed conditions, but without any expectation that the data developed will be preserved in the long run.

4. Instructions for Funding Organizations

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. Only this will 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. 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 the last speaker of a language, bring a spell checker to a generation of schoolchildren, or digitize a decomposing manuscript.

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 separated 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.

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 (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 (4) packages for various Linux distributions (Debian, Gentoo, Mandriva, etc.) (5) a port for Mac OS X (Cocoaspell) (6) free web corpora bootstrapped from your word list (from the *Crúbadán* project cited above) (7) a version of Dasher, a free program for keyboardless text entry, trained for your language using these corpora, etc. etc....