The TELRI tool catalogue: structure and prospects

Tomaž Erjavec
Dept. of Intelligent Systems
Institute "Jožef Stefan"
Jamova 39
SI-1000 Ljubljana, Slovenia
tomaz.erjavec@ijs.si

Tamás Váradi
Linguistics Institute
Hungarian Academy of Sciences
P.O.Box 701/518
Budapest H-1399, Hungary
varadi@nytud.hu

Abstract

In the scope of the TELRI concerted action a working group is investigating the formation of a tool catalogue and repository. The idea is similar to that of the ACL Natural Language Software Registry, but the contents should be mostly limited to corpus processing tools available free of cost for research use. The catalogue should also offer a help-line for installing and using the software. The paper reports on the set-up of this catalogue, and concentrates on the technical issues involved in its creation, storage and display. This involves the form interface on the Web, the XML DocBook encoding, and the XSL stylesheets used to present the catalogue either on the Web or in print. The paper lists the current entries in the catalogue and discusses plans for their expansion and maintenance.

1 Introduction

The "Trans-European Language Resources Infrastructure", TELRI (http://www.telri.de/), is a pan-European alliance of focal national language (technology) institutions with the emphasis on Central and Eastern European and NIS countries. Some of the main objectives of TELRI is to collect, promote, and make available monolingual and multilingual language resources and tools for the extraction of language data and linguistic knowledge; to provide a forum where experts from academia and industry share and assess tools and resources; and to make available the expertise of its partner institutions to the research community, to language industry and to the general public.

A number of these goals is being served by the "TELRI Research Archive of Computational Tools and Resources", TRACTOR, (http://www.tractor.de), which features monolingual, bilingual, and multilingual corpora and lexica in a wide variety of languages as well as corpus- and lexicon-related software. While the primary aim is to pool the resources of TELRI partners, TRACTOR also serves other institutions by making the resources and tools available to the wider research and educational community.

While the TRACTOR archives already offer a number of tools, the longer term objective is to offer a more substantial catalogue of corpus and lexicon processing software. Furthermore, the software itself is not necessarily available directly from TRACTOR, which would also have a more formalised structure and a well-defined process of updating and presenting its entries. A closely related initiative and model for this effort is the "The Natural Language Software Registry" of the ACL hosted at DFKI, a new edition of which was released in 2000 (Declerck et al., 2000). While the ACL registry offers a much larger array of tools, the TELRI catalogue should have the advantage that each entry also contains a pointer to the TELRI member who is able to offer advice on installing and using the tool in question.

Other related catalogues on the Web are the CTI's Guide to Digital Resources (http://info.ox.ac.uk/ctitext/resguide/) which has a section on Text Analysis Tools and Techniques. However, it does not seem to be maintained any longer.

The Summer Institute of Linguistics

(http://www.sil.org/computing/catalog/) also hosts a repository containing more than 60 pieces of software developed at SIL. Most of the software is available for free download; the latest update to the pages comes spring 1999.

A view on sharing resources, very much based on latest standardisation initiatives, has been developed by the Open Language Archives Community, OLAC, (Bird and Simons, 2000), http://www.language-archives.org/. OLAC is an international project to construct an infrastructure aimed at opening the whole array of language resources, including texts, recordings, lexicons, annotations, software, protocols, models, and formats. OLAC aims to develop community-specific metadata to link language archives and establish centralized catalogs. It builds directly on two other initiatives, namely the the Open Archives Initiative (developing and promoting interoperability standards for efficient dissemination of content) and the Dublin Core Metadata Initiative (development of interoperable online metadata standards).

In the scope of the TELRI-II concerted action, a working group has been set up to design a catalogue of corpus processing tools, and this paper reports on the preliminary results of the working group. The rest of the paper is structured as follows: Section 2 gives the overall structure of the catalogue and its entries; Section 3 explains the pipeline for updating and displaying the catalogue, i.e. the Web form interface for input, editorial policy, and the stylesheet mechanism for display; Section 4 lists the current contents of the catalogue, while Section 5. gives some conclusions and outlines plans for its expansion and further maintenance.

2 Catalogue Format

The overall encoding chosen for the catalogue was DocBook, an SGML/XML DTD primarily used for encoding computer manuals and other technical documentation. Choosing an SGML/XML framework follows a similar strand of research in annotating linguistic resources, as exemplified in the XML version of the Corpus Encoding Standard (Nancy et al., 2000) and in work on syntactic annotation (Nancy and Romary, 2001). An advantage of XML is the possibility of further standardisation by the use of related recommendations, i.e. the XML Stylesheet Language.

DocBook has a large user base and is well documented: a reference book has been published and is available on-line (Walsh, 1999) for browsing or downloading. There is also an interesting public initiative utilising DocBook, namely the Linux Documentation Project, LDP (http://www.linuxdoc.org/), which is working on developing free, high quality documentation for the GNU/Linux operating system.

Because DocBook is an application of SGML, and, more recently, XML, many freely available tools are available to process it. Most importantly, this includes XSL processors, which can be used to render DocBook documents in, say, HTML or PDF; this issue is further elaborated in Section 4.

The complete catalogue is represented as one `<book>` element, with introductory matter in `<bookinfo>` giving the name, release information and some other general information about the catalogue. The catalogue is then divided (at present) into three `<chapter>` elements, each giving a a certain type of tools we plan to address:

- morpho-syntactic taggers
- concordancers
- aligners

Each catalogue entry is contained in `<sect1>`, the top-level section element. The section, besides containing a `<title>` and being marked with an ID, is composed of two `<sect2>` elements. The first gives the information that is common to all sorts of tools, while the second is tool-type specific.

The information records are encoded as `<formalpara>`, where each such element has a `<title>`, followed by the text of the of the record as a `<para>`. Various other DocBook elements are used to annotate pieces of information, e.g. `<address>`, `<affiliation>` and similar details. Table 1 gives as an example a complete dummy catalogue entry, where variable parts are prefixed by 'this is'.

```
<sect1 id="this_is_name_971886394">
  <title><productname>this is name</productname></title>
  <sect2><title>Common part</title>
    <formalpara><title>Task</title>
      <para>this is task
        <indexterm><primary>this is task</primary></indexterm></para></formalpara>
    <formalpara><title>Author(s)</title>
      <para>this is author</para></formalpara>
    <formalpara><title>Institute/Company</title>
      <para>
        <address>
          <affiliation><orgname>this is affil</orgname></affiliation>
          <street>this is street</street>
          <city>this is city</city>
          <country>this is country</country></address></para></formalpara>
    <formalpara><title>Version</title>
      <para>this is version</para></formalpara>
    <formalpara><title>Interface</title>
      <para>this is interface</para></formalpara>
    <sect3><title>Implementation</title>
      <formalpara><title>Platform</title>
        <para><hardware>this is platform</hardware></para></formalpara>
       <formalpara><title>Operating system</title>
         <para><envar>this is os</envar></para></formalpara>
       <formalpara><title>Language of implementation</title>
         <para>this is impl</para></formalpara></sect3>
    <sect3><title>License</title>
      <formalpara><title>License conditions for research purposes</title>
        <para>this is licres</para></formalpara>
      <formalpara><title>License conditions for commercial purposes</title>
        <para>this is liccom</para></formalpara>
      <formalpara><title>Restrictions</title>
        <para>this is restrict</para></formalpara></sect3>
    <sect3><title>Distribution</title>
      <formalpara><title>Availability of source code</title>
        <para>
          <ulink url="this is source_url">this is source_url</ulink></para></formalpara>
      <formalpara><title>Download possibilities and formats</title>
        <para>
          <ulink url="this is binary_url">this is binary_url</ulink>
        </para></formalpara></sect3>
    <sect3><title>References</title>
      <formalpara><title>Homepage</title>
        <para><ulink url="this is homepage">this is home-
page</ulink></para></formalpara>
      <formalpara><title>Language of documentation</title>
        <para>this is doc_lang</para></formalpara></sect3>
    <sect3><title>TELRI helpline</title>
      <para>this is helpline</para></sect3>
  </sect2>
  <sect2><title>Tool specific part</title>
    <formalpara><title>Description</title>
      <para>this is description</para></formalpara>
  </sect2>
</sect1>
```

Table 1: Example of entry in DocBook produced via the form interace

3 Catalogue input and output

While the initial catalogue was input directly with an SGML editor and then validated, the envisioned additions will be performed via a Web form interface, available at http://gnu.nytud.hu/telri/. Figure 1 displays the top part of the screenshot of the HTML form designed to collect the specification of description of catalogue items.

The definition of the particular information sought about the software tools required some consideration. Obviously, we would like to have as detailed a description of each item as possible. On the other hand, one has to bear in mind that the TELRI Catalogue will appeal for free voluntary contributions. Hence, the form should be maximally easy to fill in with minimal effort in order to avoid possibly deterring people from contributing who might otherwise have done so. The crucial factor to consider was to find the right balance between the set of required and optional items. In the end, the required information fields were confined to the bare minimum of *name*, *task*, *description* and *TELRI helpline*. Table 2 displays the full list of questions used in the HTML form.

The form interface runs a Perl CGI script, which mails the output, encoded as the above described DocBook `<sect1>` element, to the editors of the catalogue. After checking, fresh entries are included in the official release of the catalogue.

The DocBook format is suitable for storage and interchange, but it is, of course, not appropriate for displaying the information. However, one of the benefits of using standardised solutions is that conversion tools and specifications are, to a large extent, already available. For presentation, we have been so far experimenting with the XML Stylesheet Language, XSL, or, more precisely, XSLT, the XSL Transformation Language, (W3C, 2000). XSLT is a recommendation of the W3C and is a language for transforming XML documents into other XML documents. There already exist several freely available XSLT processors, e.g., Xalan (http://xml.apache.org/xalan/), produced by the Apache XML Project.

XSLT is most often used to produce HTML output for viewing on the Web, and so called Formatted Objects, which are then further transformed into print formats, usually PDF. For DocBook XML there exist ready-made stylesheets for both kinds of output, made by Norman Walsh and available at on the Web (http://nwalsh.com/docbook/xsl/). In the current version we have used these 'out of the box' tools to render the catalogue, although some slight modifications would be in order to produce output better tailored to the catalogue application.

Figure 2 contains a sample HTML output of one item in the Catalogue.

In summary, Figure 3 gives a graphical overview of the data processing of the TELRI Catalogue items.

4 Catalogue Contents

The catalogue currently contains only a few sample entries, which, nevertheless, exemplify the kinds of software that are to be most relevant for inclusion into the catalogue:

- tools that at least one TELRI partner has experience in using and that the partner is willing to support for new users
- tools that are available free of cost, at least for academic purposes and, preferably, are open source
- tools that are language independent or adapt easily to new languages
- tools that are primarily meant for corpus processing

At present, the catalogue lists the following tools:

- The morpho-syntactic tagger TnT (Brants, 2000)
  A robust and very efficient statistical part-of-speech tagger that is trainable on different languages and on virtually any tagset. It is available by a license agreement which is free of charge for non-commercial purposes. Distribution is available, in binaries only, for Linux and SunOS/Solaris.

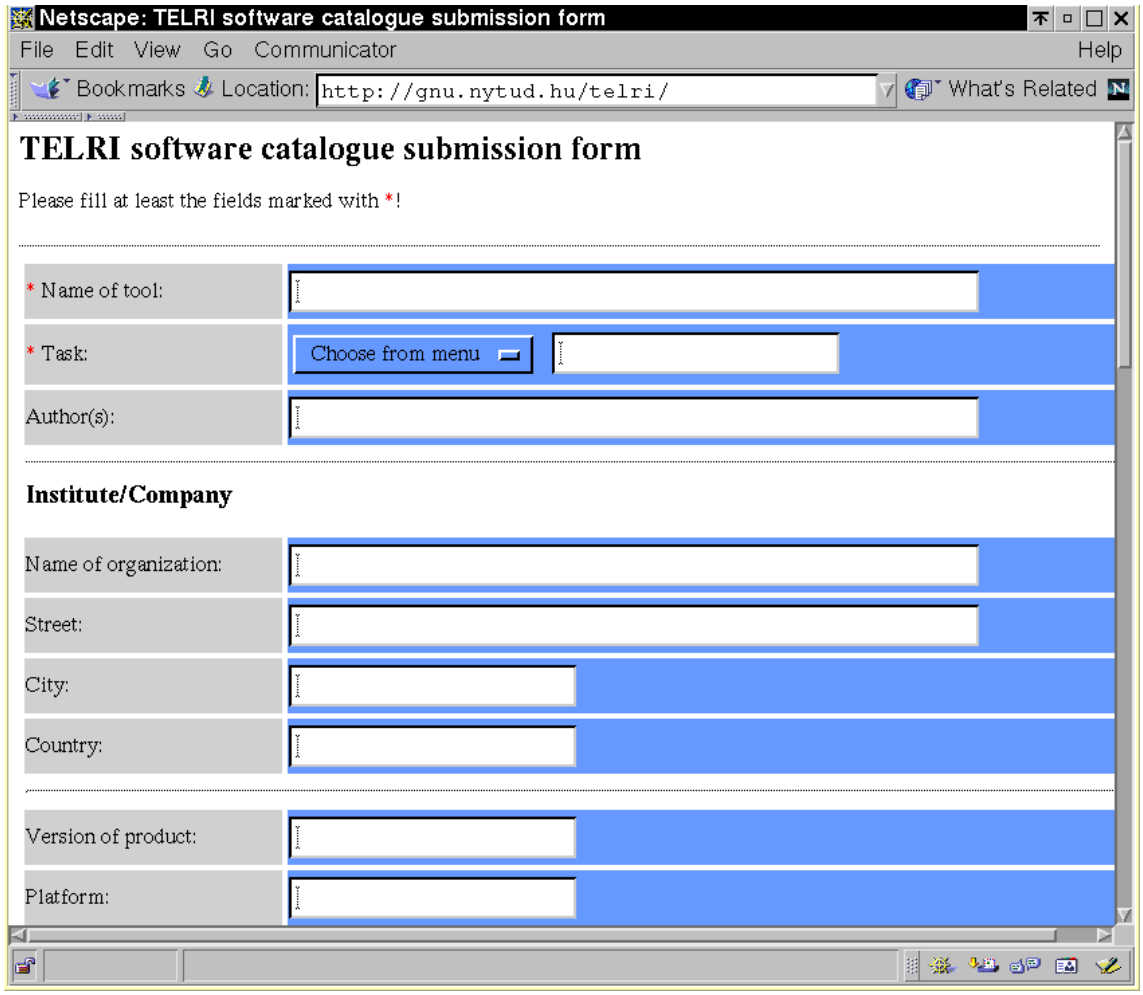

Figure 1: The TELRI Catalogue HTML form

*name	= Name of product
*task	= Task of product
author	= Name(s) of author(s)
affiliation	= Name of company
street	= Address of company
city	
country	
version	= Version number
language	= Language(s)
*description	
licres	= License conditions for research purposes
liccom	= License conditions for commercial purposes
restrict	= License restrictions
source_url	= URL of source code
binary_url	= URL of binary files
platform	= Supported hardware
os	= Supperted operating system(s)
impl	= Language of implementation
interface	= User interface
homepage	= URL of homepage
doc_url	= URL of documentation
doc_lang	= Language of documentation
*helpline	= TELRI helpline

Table 2: Full list of fields of the Catalogue HTML form

IMS Corpus Workbench

Common part

Task: concordancer

Author(s): Oliver Christ, Bruno Schulze

Institute/Company:

Institut für Maschinelle Sprachverarbeitung, Stuttgart University

Interface: command-line and graphical (Motif)

Implementation

Platform: PC, Sun

Operating system: Linux, SunOS / Solaris

Language of implementation: C

License

License conditions for research purposes: licences are available free of charge.

License conditions for commercial purposes: unknown.

References

Homepage: http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/

Language of documentation: English

TELRI helpline

LJU1

Tool specific part

Description:

Distribution package

1. Corpus Query Processor CQP
2. Xkwic graphical user interface
3. tools for the registration and indexing of corpora
4. low-level corpus access tools (decode, lexdecode)
5. sample corpus in CQP-format: Brown corpus
6. documentation

Figure 2: A sample output page of one Catalogue item

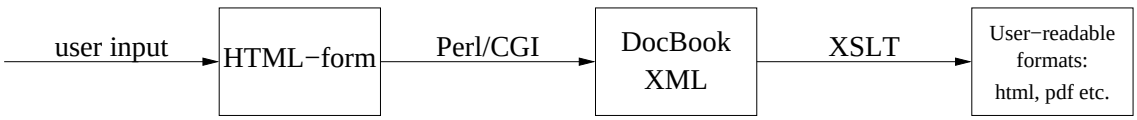

Figure 3: Overview of the catalogue data processing

- The IMS Corpus Workbench concordancer (Christ, 1994)
  Comprises a powerful Corpus Query Processor and a graphical user interface. It is available by a license agreement which is free of charge for non-commercial purposes. Distribution, in binary form only, is available for Linux and SunOS/Solaris.

- The Vanilla sentence aligner (Danielsson and Ridings, 1997)
  A simple but useful program that aligns a parallel corpus by comparing sentence lengths in characters by dynamic time-warping. The program assumes that hard boundaries are correctly aligned and performs alignment on soft boundaries. It is freely available with C source code distribution.

- The Twente Word Aligner (Hiemstra, 1998)
  The program constructs a bilingual lexicon from a parallel sentence aligned corpus. The translations are ranked according to computed confidence. The system uses statistical measures and works for single words (tokens) only. It is available under the GNU General Public License and is written in C.

- PLUG Word Aligner (Ahrenberg et al., 1998)
  The system integrates a set of modules for knowledge-lite approaches to word alignment, with various possibilities to change configuration and to adapt the system to other language pairs and text types. The system takes a parallel sentence aligned corpus as input and produces a list of word and phrase correspondences in the text (link instances) and additionally a bilingual lexicon from these instances (type links). It is available by a license agreement which is free of charge for non-commercial purposes. Distribution is available, in binary form only, for Linux and MS Windows.

5 Conclusions

The paper reported on the set-up of the TELRI corpus-tool catalogue, concentrating on the technical issues involved in its creation (form interface), storage (DocBook) and display (XSLT). At present, the input form is operational and the catalogue contains a few sample entries and has a preliminary (default) rendering of its contents. The current version of the catalogue and templates is available at http://nl.ijs.si/telri/

In the future, we hope to flesh out the catalogue with more tools, and enlist the services of TELRI experts in providing user support for them. The catalogue will, where license permits, also archive a copy of the software, and will continue with a proactive adoption of the GNU license and open standards.

The open (non-profit) nature of the tools we attempt to identify lends them well for pedagocial purposes at the graduate and undergraduate courses in natural language processing, corpus linguistics and language engineering.

The tool catalogue, as well as TRACTOR, could also be made a part of the Open Language Archives Community mentioned in the introduction. To join OLAC a number of changes and mappings would have to be defined, say from the on-line form onto Dublin Core and the OLAC Metadata Set. The choices currently listed in the template could also be changed into a controlled vocabulary to facilitate searching.

The process of catalogue updates is currently manual. To automate the production of the on-line version of the catalogue directly from new form entries would be relatively easy, given sufficient volume to justify this. More challenging would be (semi)automatic tracking of new tools that become available via various (OLAC) archives and announcements.

Acknowledgements

The authors would like to thank Inguna Greitane for her exposition of the catalogue structure vocabulary, Laurent Romary for his invaluable assistance with everything XSLT; and Victor Nagy for his technical assistance in preparing the HTML form and the CGI script.

Thanks also to the anonymous reviewers for their valuable comments on the previous version of the paper; for all remaining errors, only the authors are to blame.

The work report here was supported by the Copernicus TELRI-II concerted action.

References

Lars Ahrenberg, Mikael Andersson, and Magnus Merkel. 1998. A simple hybrid aligner for generating lexical correspondences in parallell texts. In *COLING/ACL*.

Steven Bird and Gary Simons. 2000. Open language archives community. *ElsNews*, 9(4).

Thorsten Brants. 2000. Tnt - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, Seattle, WA. http://www.coli.uni-sb.de/~thorsten/tnt/.

Oliver Christ. 1994. A modular and flexible architecture for an integrated corpus query system. In *Proceedings of COMPLEX '94: 3rd Conference on Computational Lexicography and Text Research*, Budapest, Hungary. CMP-LG archive id 9408005.

Pernilla Danielsson and Daniel Ridings. 1997. Practical presentation of a "vanilla" aligner. In *Presented at the TELRI Workshop on Alignment and Exploitation of Texts*. Institute Jožef Stefan, Ljubljana. http://nl.ijs.si/telri/Vanilla/doc/ljubljana/.

Thierry Declerck, Alexander Werner Jachmann, and Hans Uszkoreit. 2000. The new edition of the natural language software registry (an initiative of acl hosted at dfki). In *Second International Conference on Language Resources and Evaluation, LREC'00*, pages 1129–1132. Paris. ELRA. http://registry.dfki.de/.

Djoerd Hiemstra. 1998. Multilingual domain modeling in Twenty-One: automatic creation of a bi-directoral translation lexicon from a parallel corpus. In *Proceedings Computational Linguistics in the Nederlands*, pages 41–57. Nijmegen.

Ide Nancy and Laurent Romary. 2001. A Common Framework for Syntactic Annotation. In *ACL*, Toulouse.

Ide Nancy, Laurent Romary, and Patrice Bonhomme. 2000. CES/XML : An XML-based Standard for Linguistic Corpora. In *Second International Conference on Language Resources and Evaluation, LREC'00*, pages 825–830. Paris. ELRA.

W3C. 2000. Extensible stylesheet language (XSL) version 1.0. URL. http://www.w3.org/TR/xsl.

Norman Walsh. 1999. *DocBook: The Definitive Guide*. O'Reilly & Associates, Inc. http://docbook.org/.