The CLARIN Language Resource Switchboard

Claus Zinn
Seminar für Sprachwissenschaft
Universität Tübingen
claus.zinn@uni-tuebingen.de

Abstract

The CLARIN infrastructure gives users access to an increasingly rich set of language-related resources, using the Virtual Language Observatory, the Federated Content Search, and the Virtual Collection Registry. While there is ample support for searching resources using metadata-based search, or full-text search, or aggregating resources into virtual collections, there is little support for users to help them processing resources in one way or another. While there is a considerable number of processing software in the CLARIN world, there is no single point of access where users can find tools to fit their needs and the resource they have. In this paper, we present the CLARIN Language Resource Switchboard (LRS), which aims at helping users to connect resources with the tools that can process them. The LRS lists all applicable tools for a given resource, lists the tasks the tools can achieve, and invokes the selected tool in such a way so that processing can start immediately without any or little prior tool parameterization.

1 Introduction

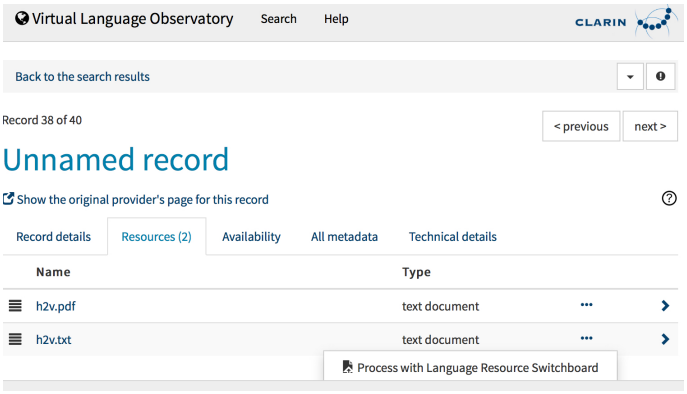
The pan-European CLARIN project is an e-Humanities project that is building an eScience infrastructure for language-related resources. Among the pillars of the infrastructure is the *Virtual Language Observatory* (VLO) that gives users a metadata-based access to language resources [1], the *Federated Content Search* (FCS) that gives users a full-text search across resources [2], and the *Virtual Collection Registry* (VCR), where users can collect resources in a virtual set [3]. CLARIN makes use of the Component MetaData infrastructure (CMDI) to describe resources in a common, flexible language. Persistent identifiers based on the Handle system ensure a persistent URL addressing of resources.

In the CLARIN world, there exists also an increasing number of tools to process language-related resources in a manifold manner. While a large part of the tools must be installed locally on users' desktop machines, there is available a good number of browser-based tools and web services.

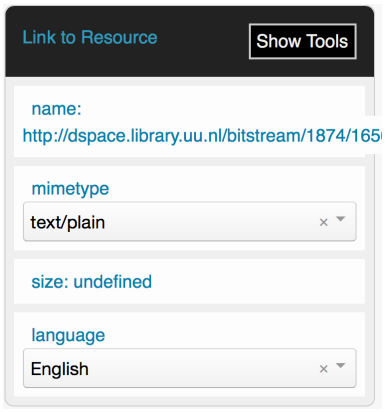
The CLARIN Language Resource Switchboard (LRS) aims at bridging the gap between resources (as identified in the VLO, FCS, and VCR) and tools that can process these resources in one way or another. The LRS can be seen as a Virtual Tool Registry. For a given resource in question, it identifies all tools that can process the resource. It then sorts the tools in terms of the tasks they perform, and presents a task-oriented list to the user. Users can then select and invoke the tool of their choosing. By invoking the tool, all relevant information about the resource in question is passed onto the tool, and the tool opens with most information gathered by the switchboard. This makes it easy for users to identify the right tools for their resource, but also to use the chosen tool in the most effective way possible.

2 The Language Resource Switchboard

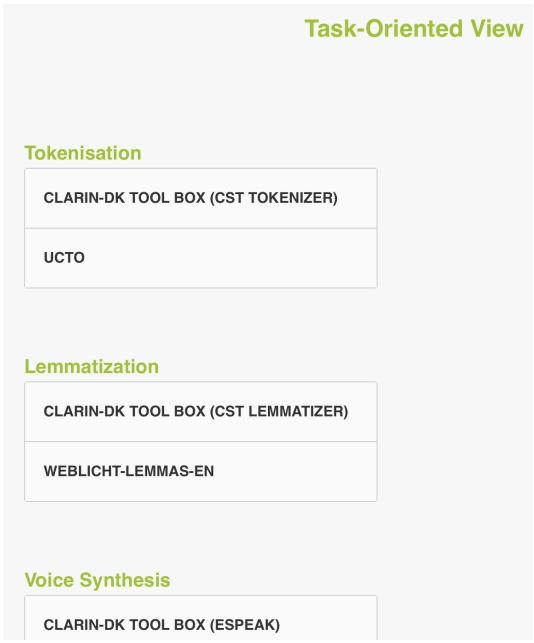
Consider the scenario where a linguist uses the VLO to find an English text which she then would like to investigate further. On the VLO search results page, the user can now click on the $\cdots$ area to invoke the LRS with this resource, see Fig. 1(a). In a new browser tab, the LRS opens and shows a resource pane that depicts all relevant information about the resource, see Fig. 1(b). The user is free to correct this metadata, before clicking on 'Show Tools' to get to the task-oriented view, shown in Fig. 1(c). If the user, say is interested in the lemmatization task, she may wish to get more information about the



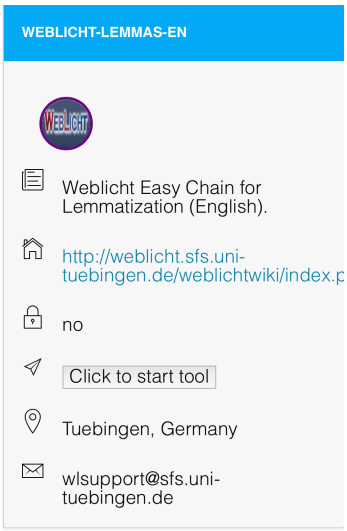
(a) The VLO – LRS Interface.



(b) The LRS Resource Pane.



(c) The LRS Task Oriented View.



(d) The LRS Tool Detail View.

Figure 1: The LRS in Action.

two tools offered, in which case more detailed information about the chosen tool is given, see Fig. 1(d). When the user then clicks on 'Click to start tool', the chosen tool, here Weblicht, opens in a new browser tab. Weblicht obtains from the LRS a reference to the resource, the resource's mimetype and language as well as the chosen task. Weblicht opens with the predefined easy chain for lemmatization, loads itself the resource, and sets all relevant parameters so that the user is left to click on Weblicht's RUN command to start the processing chain. No further user action is required to parameterize Weblicht for this.

2.1 Architecture

Fig. 2 depicts the architecture of the switchboard. Its back-end consists of three main components: a *profiler*, an *app registry*, and a *matcher*.

The Resource Profiler is responsible for identifying those resource characteristics that support listing applicable tools that can process the resource. The current version of the LRS takes into account the resource's mimetype and language, whose values are transferred from the VLO to the LRS. The profiler then makes use of the REST-based services of the Apache TIKA software to double-check the given

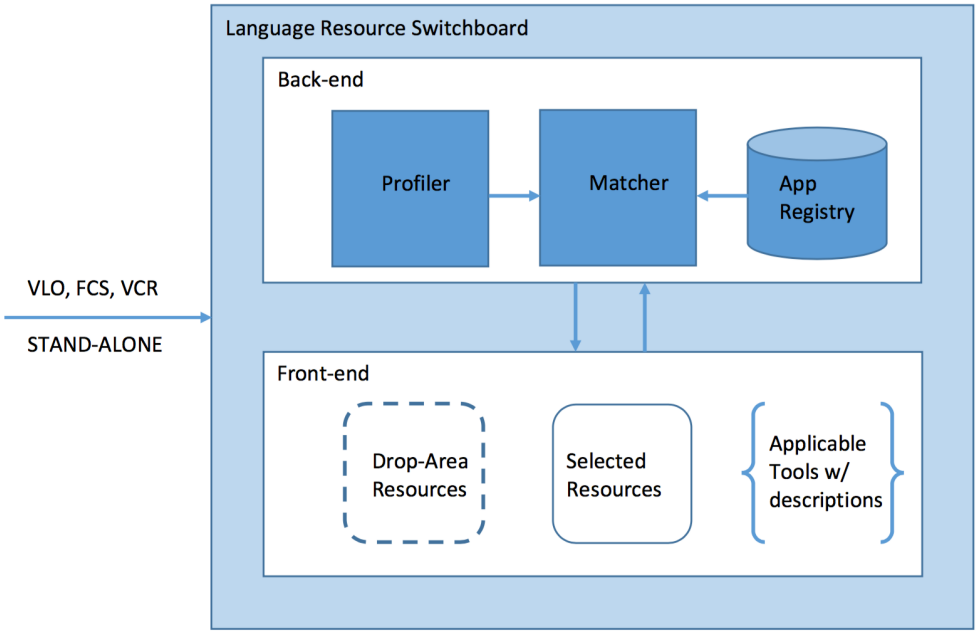


Figure 2: The LRS Architecture.

mimetype and language information, see `http://tika.apache.org`.

In a future version, the LRS profiler may take into account the entire CMDI-based description for a resource, but also the users' VLO browser history; the facets that users selected to identify some resource encode, at least to some extent, the user's search intent.

The Application Registry manages a set of metadata descriptions. Fig. 3 depicts the metadata entry for the `CLARIN-DK` toolbox from the Radboud University Nijmegen, offering a named entity analysis. For the CLARIN user, relevant metadata include the title of the application, an English description about its capabilities, and contact information about the tool provider. For the LRS, the relevant parts are the tool's task description (using a controlled vocabulary, e.g., "tokenization", "part-of-speech-tagging", "optical character recognition"), an ISO 639-3 based identification of the languages the tool can handle, and the mimetypes it can process. For tool invocation, the metadata holds the tool's web address and a list of parameters the tool understands. With this information, a URL can be constructed where all relevant information is URL-encoded.

The Matcher uses the resource' metadata from the profiler, and the tools' metadata from the application registry to find matches. For the given resource profile, it computes a list of all applicable tools and the analyses they offer. For the time being, only the resource' mimetype and language are taken into account. A future version of the LRS might complement the tool profile with a user profile that holds information about a user's access rights. Here, the user can ask the switchboard to only process resources and only list tools that the user has access to.

The LRS User Interface can be invoked from the VLO (see Fig.1(a)), but there is also a LRS standalone version [4]. Here, users can upload their resources via a simple file drag and drop mechanism. The LRS then determines the resource's mimetype and language (using Apache TIKA), as shown in Fig.1(b), which the users can correct if necessary. In the future, the LRS may also offer a REST-based interface where a JSON structure of applicable tools will be returned given the resource or its metadata is being provided.

```
{ task: "Named Entity Recognition",
  name: "CLARIN-DK Tool Box (CST's name recognizer)",
  logo: "YourLogoComesHere.png",
  homepage: "https://clarin.dk/clarindk/forside.jsp",
  location: "Copenhagen, Denmark (CLAM Webservices)",
  creators: ["Bart Jongejan et al."],
  contact: {
      person: "Bart Jongejan",
      email: "bartj@hum.ku.dk",
  },
  version: "0.8.3",
  license: "public",
  authentification: "no",
  shortDescription: "CLARIN-DK Tool Box (CST's name recognizer)",
  longDescription:  "CLARIN-DK Tool Box (CST). CST's name recogniser classifies names as
         proper names, locations (with sub-classes of street, city, land and other types
         of locations), and other names (called MISC)",
  lang_encoding: "639-1",
  languages: ["dan"],
  mimetypes: ["text/plain"
              ],
  url: ["https://www.clarin.dk/tools/createByGoalChoice"],
  parameter: {  input    : "self.linkToResource",
                lang     : "self.linkToResourceLanguage",
                analysis: "ner",
                UIlanguage: "en"
             },

  // CLARIN-DK calls those parameters differently, namely:
  mapping:   { input        : "URL",
               lang         : "language"
             }
},
```

Figure 3: The metadata entry for the CLARIN-DK toolbox.

3 Discussion and Conclusion

Two big issues that affect the usability of the LR switchboard is the quality of the metadata in the VLO, and access restrictions to resources and tools. The VLO lists a considerable number of resources with incorrect or incomplete mimetypes. Searching, for instance, for resources of type "text/plain" might yield textual resources that are not plain, but enriched with other annotations. When users load such a resource in the LRS, and subsequently process the resource with a tool of their choice, they may find out that the resource's content is not what they expected. Similarly, accessibility and authentification issues affect the usability of the LR switchboard. When the user invokes an applicable tool in the switchboard, the tool might not be able to download the resource from the resource provider. Also, some tools require authentification, and some users may not have the proper access rights to make use of the tools. For this, a number of user delegation issues need to be tackled, with many technical intricacies involved [7]. Also, public resources should be marked accordingly in the resource metadata to prevent cases where users have access to the tools but not to the resources, or *vice versa*.

In the near future, the LRS will need to address a number of issues.

Interoperability Aspects. At the time of writing, the tools that feature in the LRS can hardly be used together, or chained one after the other. As a result, the LRS was never designed to be a workflow engine such as WebLicht. The task of the LRS is rather simple. Given some information about a resource, help users to find and start a tool that can process the resource in one way or another. Once the user has been directed to the tool (the chosen tool is started in a new browser tab), LRS's engagement ends; LRS is unaware of the processing, and whether the processing succeeded or failed. Of course, users may manually save the output of the tool to a file, and then upload it to the LRS to find post-processing tools. But such tools are rarely available as most tools have their proprietary output format that other tools cannot read. In this respect, note that the integration of tools into the WebLicht workflow engine is very labour-intensive as tools must be adapted to become capable of reading and writing TCF conform data. Tools that were connected to the LRS required relatively little adaptation; they were modified so that tools were capable of processing a number of input parameters, no data formats were changed.

Management Aspects. So far, the process of adding a tool to the LRS is not mechanised. In the future, the LRS may offer an extra web form, where tool owners can register their tool with the provision of

metadata (see Fig. 3), or where they can update such data. In an update step, tool owners may advertise a new version of their tool, an alternative web address where it can be reached, or even an alternative mirroring site. We anticipate the need for such a tool-supported management software once the LRS is more widely known and used.

Legal Aspects. In the LRS stand-alone version, an uploaded resource is temporarily stored at the Max Planck Computing and Data Facility. When the user chooses a tool to process the resource, the tool downloads the resource from there. Resources are hence passed back and forth to different servers, and at the time of writing, legal aspects have not been taken into account. Such aspects need to be addressed in due time, in particular, for resources that do not fall under open access policies. Note that this issue relates to the authentication and authorization aspect mentioned earlier.

Scalability Aspects. The processing of large files is problematic as some tools may fail to process them in a timely manner. WebLicht, for instance, imposes a size limit of 3 megabytes (independent of the chosen workflow), but other tools are less explicit about this. Moreover, the size of a resource is sometimes not visible in the VLO; in this case, the size is only detectable once the resource has been downloaded to a hard disk or to the temporary file storage server. In any case, the LRS is currently not using file size information, and hence will not hide tools that may face problems with large files. This issue will need to be addressed.

At the time of writing, the LRS, and hence its associated tools, offer no "batch mode", where users can pack (zip) files of a common type together to have them processed in sequential order. For this task, users should directly use services amenable to such processing such as *WebLicht as a Service*, see [6].

Some tools need more than a single file to offer their services. The WebMAUS Basic software, *e.g.*, requires both a waveform audio file and a plain text file for phonetic segmentation [5]. The stand-alone LRS will address the issue of processing file pairs or triples by adapting the upload mechanism in the graphical user interface accordingly. The solution is less clear when looking from the VLO side.

Conclusion. The LR switchboard has received positive feedback during the last CLARIN centre meeting in Utrecht. Participants liked the stand-alone version of the LRS, and asked for the version to remain available once the LR switchboard is integrated with the VLO, FCS, and the VCR. It showed that the LR switchboard is indeed perceived as the missing link between language-related resources and the tools that can process them. We would like to encourage researchers to use the LRS for their work, and to report their feedback to steadily improve the switchboard service.

Acknowledgments. Thanks to Marie Hinrichs and Wei Qui for making available the URL parameter passing style to WebLicht, thanks to Maarten van Gompel for adding a similar construction to the LST Webservice Portal, and to Bart Jongehan for adding such support to the CLARIN-DK Tool Box. Thanks to Twan Goosen for linking the VLO with the LRS. In addition, thanks to Dieter Van Uytvanck for valuable feedback on usability aspects, and for promoting the LRS in the CLARIN community. We would like to thank the anonymous referees for their comments.

References (All links were accessed on September 28, 2016.)

[1] The CLARIN Virtual Language Observatory, see `https://vlo.clarin.eu/`.
[2] The CLARIN Federated Content Search, see `http://weblicht.sfs.uni-tuebingen.de/Aggregator/`.
[3] The CLARIN Virtual Collection Registry, see `http://clarin.ids-mannheim.de/vcr/app/public/`.
[4] The CLARIN Language Resource Switchboard, see `http://weblicht.sfs.uni-tuebingen.de/clrs/`.
[6] WebLicht as a Service, see `https://weblicht.sfs.uni-tuebingen.de/WaaS/`.
[6] WebMaus Basic, see `https://clarin.phonetik.uni-muenchen.de/BASWebServices/#/services/WebMAUSBasic`.
[7] J. Blumtritt, W. Elbers, T. Goosen, M. Hinrichs, W. Qiu, M. Salle, M. Windhouwer. *User Delegation in the CLARIN Infrastructure.* CLARIN 2014 Selected Papers; Linkoping Electronic Conference Proceedings, 2014.