WebLicht: Web-based LRT Services in a Distributed eScience Infrastructure

Marie Hinrichs, Thomas Zastrow, Erhard Hinrichs
Seminar für Sprachwissenschaft, University of Tübingen
Wilhelmstr. 19, 72074 Tübingen
E-mail: marie.hinrichs@uni-tuebingen.de, thomas.zastrow@uni-tuebingen.de, erhard.hinrichs@uni-tuebingen.de

Abstract

eScience - enhanced science - is a new paradigm of scientific work and research. In the humanities, eScience environments can be helpful in establishing new workflows and lifecycles of scientific data. WebLicht is such an eScience environment for linguistic analysis, making linguistic tools and resources available network-wide. Today, most digital language resources and tools (LRT) are available by download only. This is inconvenient for someone who wants to use and combine several tools because these tools are usually incompatible with each other. To overcome this restriction, WebLicht makes the functionality of linguistic tools and the resources themselves available via the internet as web services. In WebLicht, several kinds of linguistic tools are available which cover the basic functionality of automatic and incremental creation of annotated text corpora. To make use of the more than 70 tools and resources currently available, the end user needs nothing more than just a common web browser.

1. The eScience Paradigm

With the widespread use of the internet, a new paradigm of scientific work and research has emerged - eScience (enhanced science). It is a framework for a new approach to collaborative organization and work in broadly networked communities of scientific researchers (Gray et al., 2005).

An eScience environment consists of two components (Figure 1). The first is the underlying infrastructure. With the help of modern communication technology, the basis for collaboration and interoperability is created. This includes eLearning platforms and social networks as well as Grid and Cloud Computing for handling large amounts of data and executing computationally intensive tasks.

The second component of eScience covers social aspects and the character of globalization of modern research processes. Groups of researchers can be connected worldwide, communicating and sharing data with the help of the underlying infrastructure.

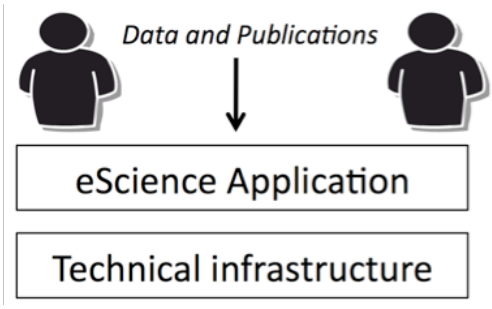

Figure 1: Structure of an eScience environment

This results in a collaborative process of creating data and publishing them together with the results of their analysis. Digitally available primary data and publications about them can then be used as a starting point for new research processes. This includes the use of data and publications in scientific disciplines that are unrelated to the discipline for which they were originally created. For example, algorithms originally developed by computational molecular biologists to analyze DNA strings can be used by computational linguists to analyze written text (Gerdemann, 2010).

In the humanities, eScience environments can be helpful in establishing new workflows and lifecycles of scientific data. WebLicht is such an eScience environment for linguistic analysis, making linguistic tools and resources available network-wide, not only for linguists, but also for researchers from other disciplines who may discover a novel way to apply these tools to their own work.

WebLicht is being developed as part of the D-SPIN[1] project.

2. Availability of Language Resources and Tools

At present, digital language resources and tools (LRT) are available in different forms: some of them can be downloaded from the worldwide web, while others are shipped on CD-ROMs or DVDs. These tools are often restricted to a particular platform or environment, but some can be used independently of operating systems or data formats. In either case, the software must be installed or copied to a local computer or network. This can lead to problems that may be difficult to resolve. For example:

- The users' machines are often not able to handle the needed resources or tools. This includes incompatibilities between libraries and operating systems as well as the fact that even modern PCs are sometimes not powerful enough to fulfil the requirements of linguistic analysis.
- The lack of common data formats makes it

[1] Deutsche SPrachressourcen-INfrastruktur

impossible to create chains of tools or to compare different resources with each other easily.

In recent years, many attempts were made to find solutions to these persistent issues. Standardized text encodings like Unicode and markup languages like XML are addressing the lack of common data formats, while platform independent programming languages like Java and Python provide a promising way to make tools available across platforms. But still, the user is reliant on the capabilities of his local machine.
To overcome this restriction, WebLicht makes the functionality of linguistic tools and the resources themselves available via the internet. To make use of the more than 70 tools and resources currently in WebLicht, the end user needs nothing more than just a common web browser.

3. The Architecture of WebLicht

3.1 Distributed Services

In WebLicht, tools and resources are implemented as distributed services (Tanenbaum & van Steen, 2002). That means that they are not running on one central machine, but instead they are running on many different machines that are distributed across the web. In most cases, the services are running at the institute where the tool was originally developed. This allows the author to keep on developing and updating the tool without interfering with the other services.
Distributing the services in this way has several advantages. The computational workload is spread among various machines, resulting in better performance. It also has the advantage that when a tool is improved, the new version is immediately available. End users do not need to do anything to take advantage of the improved tool.
The services are created as so-called *RESTful web services*, which means that they make use of the HTTP protocol to communicate with each other. To make already existing tools available as RESTful web services, wrappers are created around them to encapsulate HTTPs capabilities of establishing communication between web services and sending data from one location on the web to another (Figure 2). The services must also use a shared

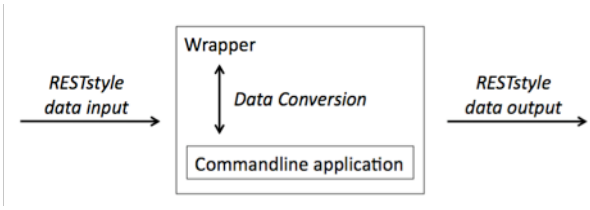

Figure 2: Tool wrapper concept

and standardized data exchange format, which is preferably based on widely accepted formats already in use (UTF-8, XML etc.)
In WebLicht, several kinds of linguistic tools are available. They cover the basic functionality of automatic and incremental annotation of text corpora:

- Tokenization
- Sentence Border Detection
- Part of Speech Tagging
- Constituent Parsing
- Semantic Annotation
- Statistic Analysis

For most of these general tool categories there is more than one tool available. These tools either operate on different languages or use different underlying algorithms, or both.
Additionally, WebLicht offers specific tools for converting the data from one format into another. At the moment, tools are available for German, English, French, Spanish, Italian, Finnish, and Romanian.

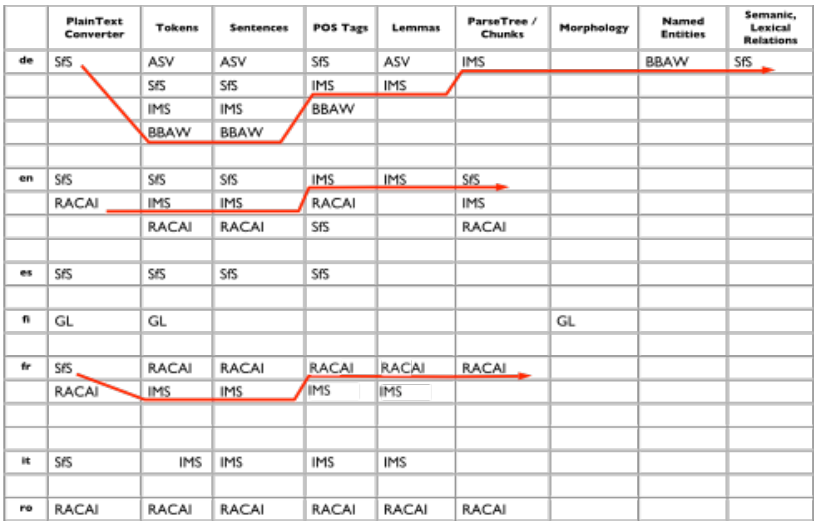

Figure 3: Several sample tool chains

3.2 The Tool Chains

To be able to automatically create linguistically annotated text corpora at different levels of analysis in an incremental fashion, the services can be combined into tool chains. Because of their general structure, many combinations are possible. Each tool that is called requires certain annotations in the input document and adds one or more linguistic annotation layer to the document. As long as the input and output requirements of each service are fulfilled, it is even possible to combine services from different providers into one tool chain. Information about each service is stored in a central repository, which makes it possible to determine which

services are compatible with each other, and therefore which tool chains are possible.
Figure 3 gives an overview of the functionality of WebLicht. The first column identifies the available languages. From left to right are columns of possible annotations. The inner part of the table shows which tools the various partners have produced for the given language. See section 6 (Acknowledgements) for clarification of the abbreviations.
Several sample tool chains that can be executed in WebLicht are shown with bold lines. From top to bottom, a tool chain for German, English and French are drawn. The path of the lines from one location to another depicts the distributed nature of the WebLicht architecture and the possible tool choices the user has.

3.3 The Repository

To be available in WebLicht, a service must be registered in a centralized repository. Figure 4 shows a screenshot of the application used to manage the repository.
The repository stores two different kinds of information about each service:

- Technical metadata: they provide details about what is required in the input document and what output the service generates. This information is needed to be able to compute valid tool chains. For example, a POS tagger needs a tokenizer to be applied to the text before it can be called.
- Descriptive metadata: concerning author, location of the service, legal information, description of the service etc.

The repository not only stores this information, but is able to use it to determine which chains can be formed. This is done by matching the input requirements of each tool in the repository with the annotation layers produced by the chain constructed up to this point. The repository itself offers its functionality as web services, which allows the user interface to integrate new services immediately and seamlessly into the tool chains.

3.4 The User Interface

As mentioned above, accessing Weblicht and its functionality should be as easy and intuitive as possible. Therefore, the user interface was created as a Web 2.0 application. It demands only a common web browser on the users local machine.
In the interface, the user can create chains of linguistic tools and apply them on uploaded texts. WebLicht insures that the user only selects tools that are compatible with each other. After the tool chain has been successfully applied to the input text, the results are presented in several ways:

- The user can view them in XML format with the option of downloading to his or her local machine.
- The integrated visualization in WebLicht makes it possible to view the generated linguistic annotations directly within the WebLicht user interface. The annotations are displayed either in a table (tokens, POS tags, lemmas), in a list (sentences), or as a graphic (parse trees). Which view(s) are presented depends on which annotation layers were produced by the tool chain.

Not only the final result, but also every intermediate step of the tool chain is available for download and visualization. This makes it possible to create a new chain from a different starting point and then compare the results of several tools of the same type. For example, a POS tagger will create different results, depending on which tokenizer was previously applied. These visualization windows can be minimized and later retrieved from a toolbar at the bottom of the screen (see Figure 5).

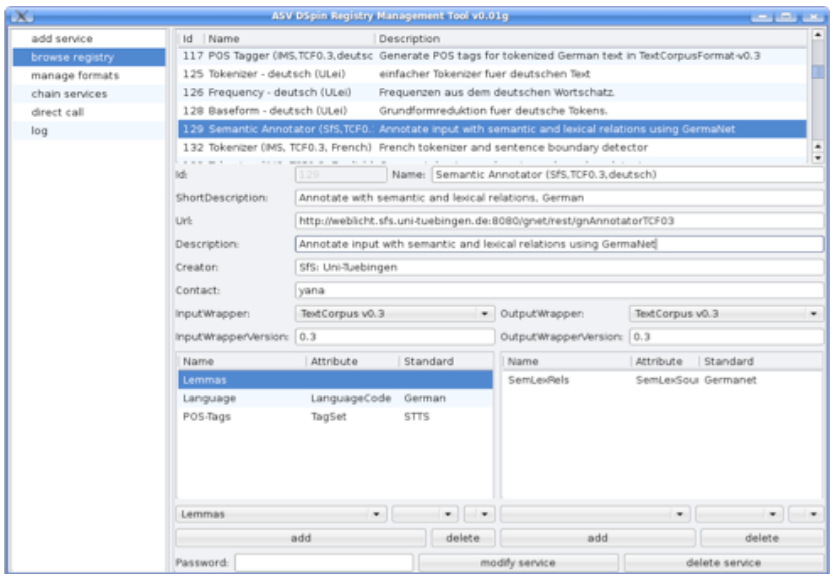

Figure 4: Tool for managing services in the repository

3.5 The Data Format

The data format used by WebLicht is a valid XML format compatible with the *Linguistic Annotation Format* (LAF) and *Graph-based Format for Linguistic Annotations* (GrAF) developed in the ISO/TC37/SC4 technical committee (Ide & Suderman, 2007). All services support compatibility with this format for input and output.

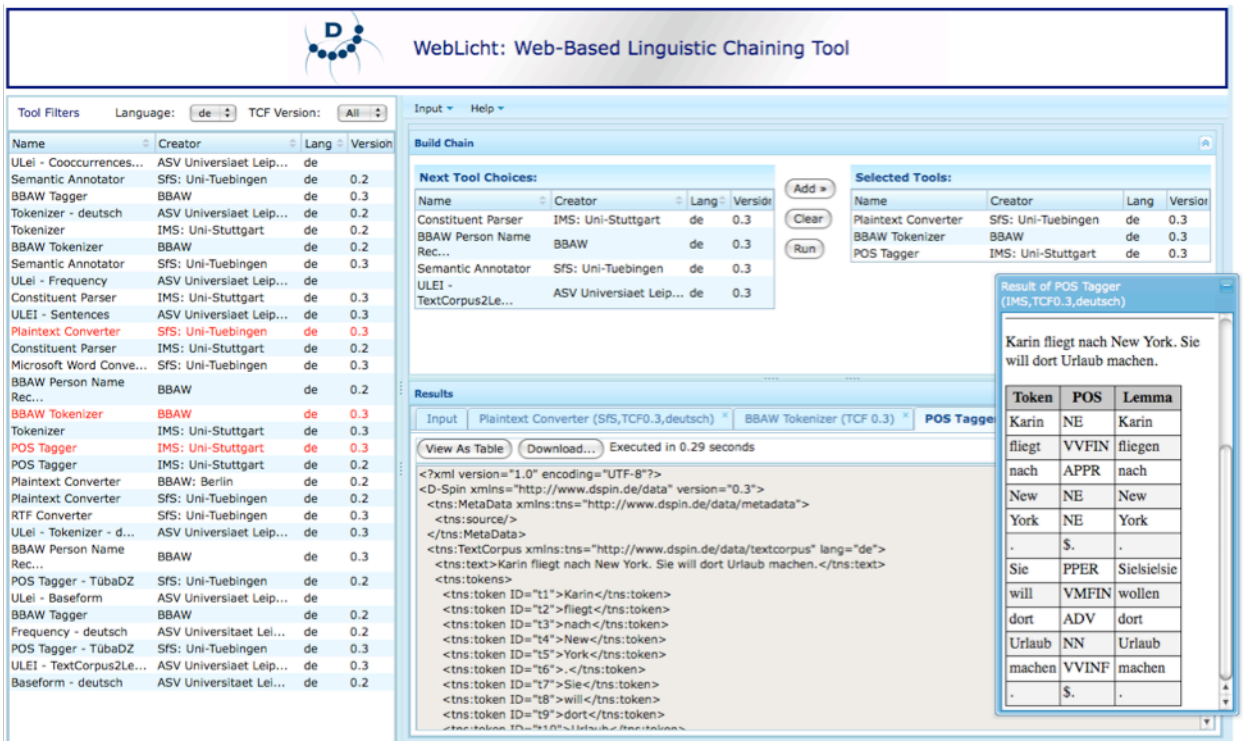

Figure 5: Screenshot of the WebLicht web interface

The data is stored in stand-off manner. This means that every linguistic annotation is stored in its own separate layer within the XML data. For ease of sending the data from one service to the next, the layers are stored in a single file, following the ODD principle ("one document does it all"). Each service may add an arbitrary number of layers, but should never change or remove any existing layers. Each category of linguistic annotation can only appear once in the XML data. For example, it is not possible to add more than one tokenizer to a processing chain. This style of architecture makes it possible to create new branches at every position in the chaining process.

Because of its simple and convenient structure, data in WebLicht's XML format can be converted easily into other data formats. WebLicht will offer converters for some widely known and used linguistic data formats, such as the PennTreeBank and the NEGRA format. XML schemas, NG Relax and Schematron rules are available on the D-SPIN webpage[2].

For a detailed explanation of the data formats used within WebLicht, please see Heid, et al., 2010.

4. How to Participate in WebLicht

The WebLicht team always welcomes new partners who are interested in contributing tools and services. Since WebLicht follows the paradigm of a Service Oriented Architecture (Melzer, 2008), new services can be easily integrated. Normally this can be accomplished in two steps: writing a wrapper around an existing tool so that it can be accessed as a RESTful web service, and then registering it in WebLicht's central repository. Further details, including a tutorial, are available on the D-SPIN homepage (http://www.d-spin.org).

5. Conclusion and Further Work

Services included in WebLicht allow automatic incremental annotation of text corpora. Such annotations are useful not only for researchers in linguistics, but for any humanities disciplines which have to process and analyze large amounts of text. WebLicht combines web services in such a way that the former difficulties and complications of using linguistic tools are avoided. Now they can be accessed easily within a web browser.

In a future version of WebLicht, it will also be possible to store the results of the chaining processes online. This will require the creation of personalized workspaces within WebLicht. Personal workspaces will allow the migration of the scientific workflow into the net, thereby avoiding the restrictions imposed by use of local computer systems. In turn, this will ease the process of maintaining data for long-term preservation and sustainability.

[2] http://www.d-spin.org

6. Acknowledgements

WebLicht is the product of a collaborative effort within the D-SPIN project (www.d-spin.org). At the time of writing, partners include:

- Seminar für Allgemeine Sprachwissenschaft und Computerlinguistik, Universität Tübingen (SfS)
- Abteilung für Automatische Sprachverarbeitung, Universität Leipzig (ASV / ULEI)
- Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart (IMS)
- Berlin Brandenburgische Akademie der Wissenschaften (BBAW)

Additional web services are offered by the University of Helsinki (GL) and the Research Institute for Artificial Intelligence at the Romanian Academy (RACAI).

We thank all collaboration partners for their most valuable contributions.

The D-SPIN (Deutsche SPrachressourcen-INfrastruktur) project is financed by the German Federal Ministry of Education and Research (BMBF); it is a national German complement to the EU-project CLARIN. See the URLs http://www.d-spin.org and http://www.clarin.eu for further details.

7. References

Gerdemann, D. (2010). Gappy Phrase Discovery Using Suffix and Prefix Arrays. To be presented at the First Tübingen Workshop on Machine Learning, Tübingen, Germany.

Gray, J., Liu, D., Nieto-Santisteban, M., Szalay, A., DeWitt, D., Heber, G. (2005). Scientific Data Management in the Coming Decade. Technical Report MSR-TR-2005-10, Microsoft Research.

Heid, U., Schmid, H., Eckart, K., Hinrichs, E. (2010). A Corpus Representation Format for Linguistic Web Services: the D_SPIN Text Corpus Format and its Relationship with ISO Standards. In *Proceedings of LREC 2010*, Malta.

Ide, N., Suderman, K. (2007). GrAF: A Graph-based Format for Linguistic Annotations. In *Linguistic Annotation Workshop* (the LAW), ACL-2007, Prague, Czech Republic.

Melzer, I. (2008). Service-orientierte Architekturen mit Web Services: Konzepte - Standards – Praxis, Spektrum Akademischer Verlag, 3. Auflage.

Tanenbaum, A., van Steen, M. (2002). Distributed Systems, Prentice Hall, Upper Saddle River, NJ, 1st Edition.