WebLicht: Web-based LRT services for German

Erhard Hinrichs, Marie Hinrichs, Thomas Zastrow
Seminar für Sprachwissenschaft, University of Tübingen
firstname.lastname@uni-tuebingen.de

Abstract

This software demonstration presents WebLicht (short for: *Web-Based Linguistic Chaining Tool*), a web-based service environment for the integration and use of language resources and tools (LRT). WebLicht is being developed as part of the D-SPIN project[1]. WebLicht is implemented as a web application so that there is no need for users to install any software on their own computers or to concern themselves with the technical details involved in building tool chains. The integrated web services are part of a prototypical infrastructure that was developed to facilitate chaining of LRT services. WebLicht allows the integration and use of distributed web services with standardized APIs. The nature of these open and standardized APIs makes it possible to access the web services from nearly any programming language, shell script or workflow engine (UIMA, Gate etc.) Additionally, an application for integration of additional services is available, allowing anyone to contribute his own web service.

1 Introduction

Currently, WebLicht offers LRT services that were developed independently at the Institut für Informatik, Abteilung Automatische Sprachverarbeitung at the University of Leipzig (tokenizer, lemmatizer, co-occurrence extraction, and frequency analyzer), at the Institut für Maschinelle Sprachverarbeitung at the University of Stuttgart (tokenizer, tagger/lemmatizer, German morphological analyser SMOR, constituent and dependency parsers), at the Berlin Brandenburgische Akademie der Wissenschaften (conversion of plain text to D-Spin format, tokenizer, taggers, NE recognizer) and at the Seminar für Sprachwissenschaft/Computerlinguistik at the University of Tübingen (conversion of plain text to D-Spin format, GermaNet, Open Thesaurus synonym service, and Treebank browser). They cover a wide range of linguistic applications, like tokenization, co-occurrence extraction, POS Tagging, lexical and semantic analysis, and several laguages (currently German, English, Italian, French, Romanian, Spanish and Finnish). For some of these tasks, more than one web service is available. As a first external partner, the University of Helsinki in Finnland contributed a set of web services to create morphological annotated text corpora in the Finnish language. With the help of the webbased user interface, these individual web services can be combined into a chain of linguistic applications.

2 Service Oriented Architecture

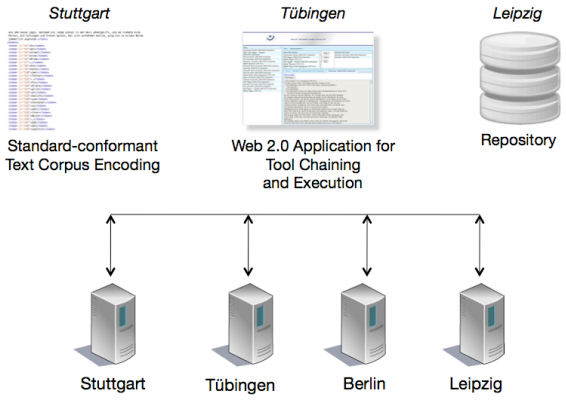

Figure 1: The Overall Structure of WebLicht

WebLicht is a so-called Service Oriented Architecture (Binildas et. al., 2008), which means that distributed and independent services (Tanenbaum et al, 2002) are combined together to a chain of LRT tools. A centralized database, the repository, stores technical and content-related metadata about each service. With the help of

[1] D-SPIN stands for **D**eutsche **SP**rachressourcen **IN**frastruktur; the D-SPIN project is partly financed by the 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 details

Proceedings of the ACL 2010 System Demonstrations, pages 25–29,
Uppsala, Sweden, 13 July 2010.

this repository, the chaining mechanism as described in section 3 is implemented. The WebLicht user interface encapsulates this chaining mechanism in an AJAX driven web application. Since web applications can be invoked from any browser, downloading and installation of individual tools on the user's local computer is avoided. But using WebLicht web services is not restricted to the use of the integrated user interface. It is also possible to access the web services from nearly any programming language, shell script or workflow engine (UIMA, Gate etc.). Figure 1 depicts the overall structure of WebLicht.

An important part of Service Oriented Architectures is ensuring interoperability between the underlying services. Interoperability of web services, as they are implemented in WebLicht, refers to the seamless flow of data between them. To be interoperable, these web services must first agree on protocols defining the interaction between the services (WSDL/SOAP, REST, XML-RPC). They must also use a shared and standardized data exchange format, which is preferably based on widely accepted formats already in use (UTF-8, XML). WebLicht uses the RESTstyle API and its own XML-based data exchange format (Text Corpus Format, TCF).

3 The Service Repository

Every tool included in WebLicht is registered in a central repository, located in Leipzig. Also realized as a web service, it offers metadata and processing information about each registered tool. For example, the metadata includes information about the creator, name and the adress of the service. The input and output specifications of each web service are required in order to determine which processing chains are possible.

Combining the metadata and the processing information, the repository is able to offer functions for the chain building process.

Wrappers: TCF, 0.3 / TCF, 0.3	
Inputs	Outputs
lemmas postags -tagset: stts	sem_lex_rels -source: GermaNet

Table 1: Input and Output Specifications of Tübingen's Semantic Annotator

A specialized tool for registering new web services in the repository is available.

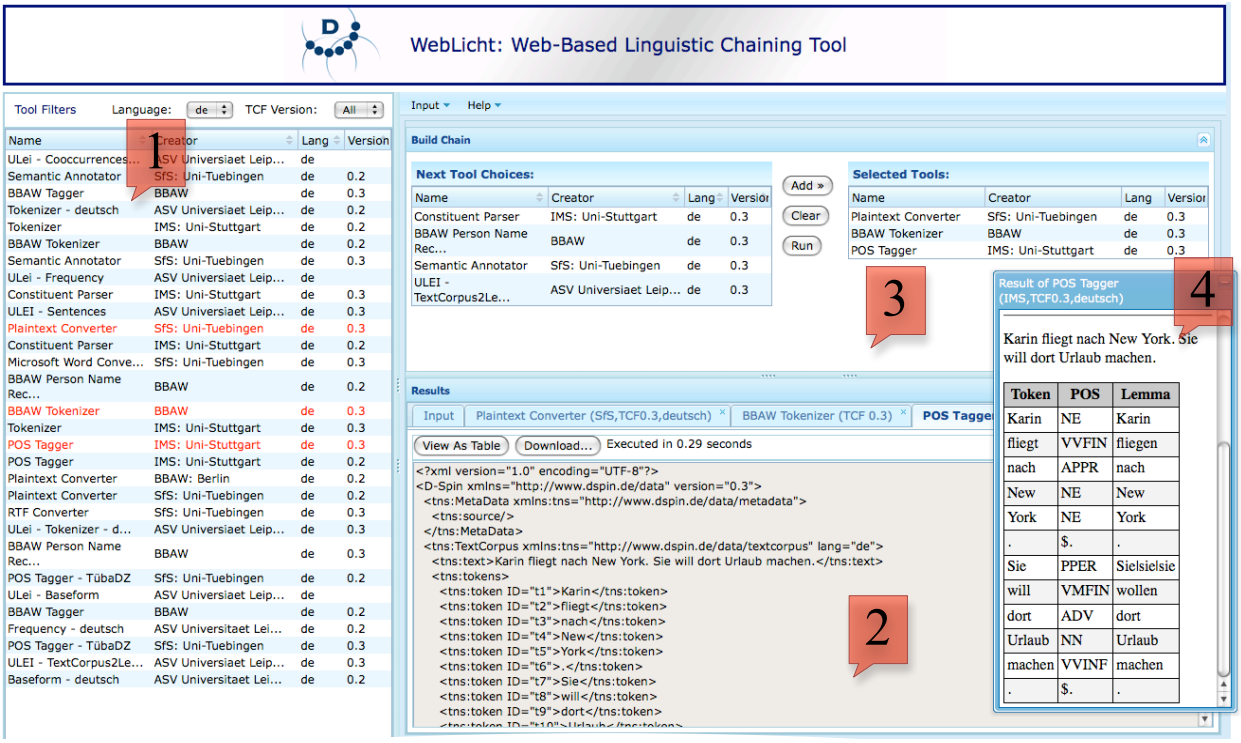

Figure 2: A Screenshot of the WebLicht Webinterface

4 The WebLicht User Interface

Figure 2 shows a screenshot of the WebLicht web interface, developed and hosted in Tübingen. Area 1 shows a list of all WebLicht web services along with a subset of metadata (author, URL, description etc.). This list is extracted on-the-fly from a centralized repository located in Leipzig. This means that after registration in the repository, a web service is immediatley available for inclusion in a processing chain.

The *Language Filter* selection box allows the selection of any language for which tools are available in WebLicht (currently, German, English, Italian, French, Romanian, Spanish or Finnish). The majority of the presently integrated web services operates on German input. The platform, however, is language-independent and supports LRT resources for any language.

Plain text input to the service chain can be specified in one of three ways: a) entered by the user in the *Input tab*, b) file upload from the user's local harddrive or c) selecting one of the sample texts offered by WebLicht (Area 2). Various format converters can be used to convert uploaded files into the data exchange format (TCF) used by WebLicht. Input file formats accepted by WebLicht currently include plain text, Microsoft Word, RTF and PDF.

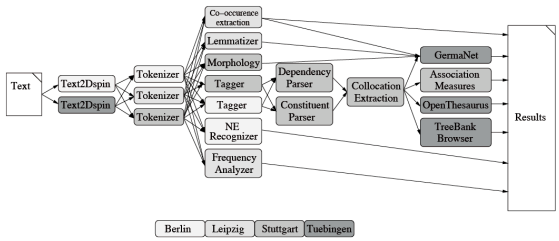

Figure 3: A Choice of Alternative Services

In Area 3, one can assemble the service tool chain and execute it on the input text. The *Selected Tools* list displays all web services that have already been entered into the web service chain. The list under *Next Tool Choices* then offers the set of tools that can be entered as next into the chain. This list is generated by inspecting the metadata of the tools which are already in the chain. The chaining mechanism ensures that this list only contains tools, that are a valid next step in the chain. For example, a Part-of-Speech Tagger can only be added to a chain after a tokenizer has been added. The metadata of each tool contains information about the annotations which are required in the input data and which annotations are added by that tool.

As Figure 3 shows, the user sometimes has a choice of alternative tools - in the example at hand a wide variety of services are offered as candidates. Figure 3 shows a subset of web service workflows currently available in WebLicht. Notice that these workflows can combine tools from various institutions and are not restricted to predefined combinations of tools. This allows users to compare the results of several tool chains and find the best solution for their individual use case.

The final result of running the tool chain as well as each individual step can be visualized in a *Table View* (implemented as a seperate frame, Area 4), or downloaded to the user's local harddrive in WebLicht's own data exchange format TCF.

5 The TCF Format

The D-SPIN *Text Corpus Format* TCF (Heid et al, 2010) is used by WebLicht as an internal data

```
<?xml version="1.0" encoding="UTF-8" ?>
<D-Spin xmlns="http://www.dspin.de/data" version="0.3">
    <tns:MetaData xmlns:tns="http://www.dspin.de/data/metadata">
        <tns:source/>
    </tns:MetaData>
    <tns:TextCorpus xmlns:tns="http://www.dspin.de/data/textcorpus" lang="en">
        <tns:text>Bob went to the zoo.</tns:text>
        <tns:tokens>
            <tns:token ID="t0">Bob</tns:token>
            <tns:token ID="t1">went</tns:token>
            <tns:token ID="t2">to</tns:token>
            <tns:token ID="t3">the</tns:token>
            <tns:token ID="t4">zoo</tns:token>
            <tns:token ID="t5">.</tns:token>
        </tns:tokens>
        <tns:POStags tagset="PennTB">
            <tns:tag tokID="t0">NP</tns:tag>
            <tns:tag tokID="t1">VBD</tns:tag>
            <tns:tag tokID="t2">TO</tns:tag>
            <tns:tag tokID="t3">DT</tns:tag>
            <tns:tag tokID="t4">NN</tns:tag>
            <tns:tag tokID="t5">.</tns:tag>
        </tns:POStags>
        <tns:lemmas>
            <tns:lemma tokID="t0">Bob</tns:lemma>
            <tns:lemma tokID="t1">go</tns:lemma>
            <tns:lemma tokID="t2">to</tns:lemma>
            <tns:lemma tokID="t3">the</tns:lemma>
            <tns:lemma tokID="t4">zoo</tns:lemma>
            <tns:lemma tokID="t5">.</tns:lemma>
        </tns:lemmas>
    </tns:TextCorpus>
</D-Spin>
```

Figure 4: A Short Example of a TCF Document, Containing the Plain Text, Tokens and POS Tags and Lemmas

exchange format. The TCF format allows the combination of the different linguistic annotations produced by the tool chain. It supports incremental enrichment of linguistic annotations at different levels of analysis in a common XML-based format (see Figure 4).

The Text Corpus Format was designed to efficiently enable the seamless flow of data between the individual services of a Service Oriented Architecture.
Figure 4 shows a data sample in the D-SPIN Text Corpus Format. Lexical tokens are identified via token IDs which serve as unique identifiers in different annotation layers. From an organizational point-of-view, tokens can be seen as the central, atomic elements in TCF to which other annotation layers refer. For example, the POS annotations refer to the token IDs in the token annotation layer via the attribute *tokID*.
The annotation layers are rendered in a stand-off annotation format. TCF stores all linguistic annotation layers in one single file. That means that during the chaining process, the file grows (see Figure 5). Each tool is permitted to add an arbitrary number of layers, but it is not allowed to change or delete any existing layer.
Within the D-SPIN project, several other XML based data formats were developed beside the TCF format (for example, an encoding for lexicon based data). In order to avoid any confusion of element names between these different formats, namespaces for the different contextual scopes within each format have been introduced.
At the end of the chaining process, converter services will convert the textcorpora from the

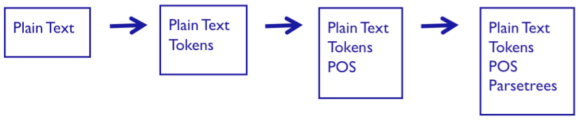

Figure 5: Annotation Layers are Added to the TCF Document by Each Service

TCF format into other common and standardized data formats, for example MAF/SynAF or TEI.

6 Implementation Details

The web services are available in RESTstyle and use the TCF data format for input and output. The concrete implementation can use any combination of programming language and server environment.
The repository is a relational database, offering its content also as RESTstyle web services.
The user interface is a Rich Internet Application (RIA), using an AJAX driven toolkit. It incorporates the Java EE 5 technology and can be deployed in any Java application server.

7 How to Participate in WebLicht

Since WebLicht follows the paradigm of a Service Oriented Architecture, it is easily extendable by adding new services. In order to participate in WebLicht by donating additional tools, one must implement the tool as as RESTful web service using the TCF data format. You can find further information including a tutorial on the D-SPIN homepage[2].

8 Further Work

The WebLicht platform in its current form moves the functionality of LRT tools from the users desktop computer into the net (Gray et al, 2005). At this point, the user must download the results of the chaining process and deal with them on his local machine again. In the future, an online workspace has to be implemented so that annotated textcorpora created with WebLicht can also be stored in and retrieved from the net. For that purpose, an integration of the eSciDoc research environment[3] into Weblicht is planned. The eSciDoc infrastructure enables sustainable and reliable long-term preservation of primary research and analysis data.
To make the use of WebLicht more convenient to the end user, there will be predefined processing chains. These will consist of the most commonly used processing chains and will relieve the user of having to define the chains manually.
In the last year, WebLicht has proven to be a realizable and useful service environment for the humanities. In its current state, WebLicht is still a prototype: due to the restrictions of the underlying hardware, WebLicht cannot yet be made available to the general public.

9 Scope of the Software Demonstration

This demonstration will present the core functionalities of WebLicht as well as related modules and applications. The process of building language-specific processing tool chains will be shown. WebLichts capability of offering only appropriate tools at each step in the chain-building process will be demonstrated.

---

[2] http://weblicht.sfs.uni-tuebingen.de/englisch/weblichttutorial.shtml
[3] For further information about the eSciDoc platform, see https://www.escidoc.org/

The selected tool chain can be applied to any arbitrary uploaded text. The resulting annotated text corpus can be downloaded or visualized using an integrated software module.
All these functions will be shown live using just a webbrowser during the software demonstration.Demo Preview and Hardware Requirements

The call for papers asks submitters of software demonstrations to provide pointers to demo previews and to provide technical details about hardware requirements for the actual demo at the conference.
The WebLicht web application is currently password protected. Access can be granted by requesting an account (weblicht@d-spin.org).
If the software demonstration is accepted, internet access is necessary at the conference, but no special hardware is required. The authors will bring a laptop of their own and if necessary also a beamer.

Acknowledgments

WebLicht is the product of a combined effort within the D-SPIN projects (www.d-spin.org). Currently, partners include: Seminar für Sprachwissenschaft/Computerlinguistik, Universität Tübingen, Abteilung für Automatische Sprachverarbeitung, Universität Leipzig, Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart and Berlin Brandenburgische Akademie der Wissenschaften.

References

Binildas, C.A., Malhar Barai et.al. (2008). *Service Oriented Architectures with Java.* PACKT Publishing, Birmingham – Mumbai

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.

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