Transatlantic Collaboration between LAPPS and CLARIN
M6 Milestone Report (1-6 months)

WP2, Tasks T2.1 and T2.2

Data Converters, Webservice Metadata

Introduction

This document reports on the progress in WP2 in the first 6 months of the project. The goals were to implement conversion software between the data formats used by the two systems (task T2.1), and to review options for representing tool metadata (task T2.2). Software for converting documents in both directions between the two formats used by the systems was implemented and made available on GitHub. This is an important first step in making tools in each system available to users of the other system, due to their differing underlying architectures and internal data exchange formats. Data sent from one system to the other for processing will first be converted to the appropriate format, allowing the underlying tools to operate on data in their native format. In order to make the tools in each system visible to the other, metadata describing the tools needs to be made available. Options for representing tool metadata were also reviewed.

Background

The Language Application (LAPPS)[1] Grid is a framework that provides access to basic NLP processing tools and resources and enables pipelining these tools to create custom NLP applications, as well as access to language resources such as mono- and multilingual corpora and lexicons that support NLP. It orchestrates access to and deployment of language resources and processing functions available from servers around the globe and enables users to add their own language resources. It is using as a front-end the Galaxy[2] workflow infrastructure that provides significant advantages for deploying pipelines. Galaxy's functionality includes means to create and deploy locally-run and even customized versions of the LAPPS Grid. Galaxy also gives access to a huge array of statistical and visualization tools that have been developed for use in genomics research. Recently, functionality has been added to make it easy to deploy a LAPPS Grid in the cloud using Docker images.[3] The LAPPS Interchange Format (LIF)[4],–a JSON-LD based data exchange format, is designed to represent linguistically annotated data for the purpose of web service exchange. The LIF Format enables syntactic and semantic interoperability of language services by providing a uniform syntax for common linguistic data

[1] http://www.lappsgrid.org/
[2] http://galaxyproject.org
[3] https://www.docker.com/
[4] http://wiki.lappsgrid.org/interchange/

and by using the Linked Data aspect of JSON-LD to refer to external definitions of linguistic categories. It is tightly integrated with the Web Services Exchange Vocabulary (WSEV)[5], which specifies a terminology for a core of linguistic objects and features exchanged by services. To access the language services and resources of LAPPS from another application, the input data needs to be wrapped as LIF.

WebLicht[6] is an environment for building and executing natural language processing pipelines, integrated into the CLARIN[7] infrastructure, which provides easy access to a wide range of text processing tools to researchers in the humanities and social sciences. It is built upon Service Oriented Architecture (SOA) principles, which means that processing tools are implemented as web services that are hosted on servers distributed across the web, in this case at CLARIN centers. The tools are exposed to WebLicht via metadata descriptions which are harvested from the repositories of CLARIN centers that provide WebLicht-compatible web-services. The validity of workflows constructed in WebLicht is ensured through use of this metadata, in particular the input and output requirement descriptions in the metadata. Each time a service is added to a workflow, the cumulative output of the workflow is calculated by inspecting the output descriptions in the metadata for each tool in the workflow. This cumulative output is used to exclude some services as candidates for inclusion in the workflow at the next step, such as services which require input that is not provided by the workflow, or which produce output that is already supplied by the workflow. A workflow is executed by sequentially invoking each service in the pipeline, passing the output of one service as input to the next. To enable the services to interpret the input data and to produce valid output data, the XML format Text Corpus Format (TCF)[8] was developed for use as an internal data exchange format. Annotation tools are wrapped as web services that receive and return TCF. Use of the TCF format allows the various linguistic annotations produced by the tools to be stored in one document, where each tool in a workflow adds linguistic annotations to the document.

As described above, the LAPPS Grid and WebLicht have different underlying architectures and internal data exchange formats. To build a bridge which allows tools from one architecture to be called from the other architecture, while at the same time allowing tools in each system to operate on data in their native format, data converters (LIF ↔ TCF) and distribution of service metadata in each direction are necessary. The following two sections report on the development carried out on each of the two tasks described in the funding proposal: T2.1, Implementation of data converters, and T2.2, Inspection of the current state of metadata descriptions for LAPPS and WebLicht tools.

[5] http://vocab.lappsgrid.org/
[6] https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Main_Page
[7] https://www.clarin.eu/
[8] https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/The_TCF_Format

T2.1 Data Converters

This section discusses various aspects of the data converters that convert between the LAPPS Interchange Format (LIF), which is used by LAPPS services, and TCF, used by WebLicht services. The section starts with outlining the underlying structure of LIF and TCF along with JSON-LD and XML. Following this, the structural differences in relation to the conversion are detailed. To reveal the structural differences an example of conversion of token-level-information is presented. The section ends with the discussion of the development of the converter software and its functionalities.

In order to convert from one data format to another, it is necessary to understand the vocabulary terms used in each format and map them to those used in the other format. LIF is tightly integrated with the Web Services Exchange Vocabulary (WSEV), which specifies a terminology for a core of linguistic objects and features exchanged by services. TCF has a similar, but not identical, vocabulary of terms to represent linguistic annotations. The vocabulary terms used for annotations in each format were identified. The detail of inventory of terms and features, which is out of scope of this report, is detailed in the report on objects and features (task T3.1).
LIF and TCF are stored in different file formats. A LIF document is an instantiation of JSON-LD[9] (JavaScript Object Notation for Linked Data), a method for transporting Linked Data using JSON. JSON[10] is a lightweight, text-based, language-independent data interchange format that defines a small set of formatting rules for the portable representation of structured data. In a JSON file, the data is stored as a collection of *name*/*value* pairs. JSON-LD extends JSON by enabling services to reference categories and definitions in web-based repositories and ontologies. Following the structure of JSON-LD, LIF stores data as LIF object which is a collection of *name*/*value* pairs of JSON. The *name* field refers to annotation tag and the *value* field refers to either content or tag. The syntactic shape of the object is described in JSON schema for LIF[11] where the specifications are taken from the elements of Web Services Exchange Vocabulary (WSEV). A TCF document is an instantiation of XML (eXtensible Markup Language), which is widely used for representing data for transport over the internet. In XML, data is represented in nested elements. TCF stores each annotation type in a separate element, and uses XML attributes to give additional information about a given element. The structure of a TCF document is described in its schema, which is available on github[12].

The structural differences and granularity of annotation data in LIF and TCF impose several challenges to the conversion task. Some of these challenges can be explained in the context of token-level-information conversion in both directions.

One challenge faced in converting token-level annotations from TCF to LIF stems from the fact that LIF requires start and end character offsets into the primary text to be part of the tokens-level annotation, but TCF token annotations normally do not contain this character offset

[9] http://json-ld.org
[10] http://json.org
[11] http://vocab.lappsgrid.org/schema/lif-schema.json
[12] https://github.com/weblicht/tcf-spec

information. Figure 1 shows token-level annotations in LIF (on the left) and TCF (on the right). Here, the token ‘Karen’ has a start offset of 0 and an end offset of 5 in LIF. To solve this problem, it was necessary to calculate the start and end character offsets for each token when converting from TCF to LIF.

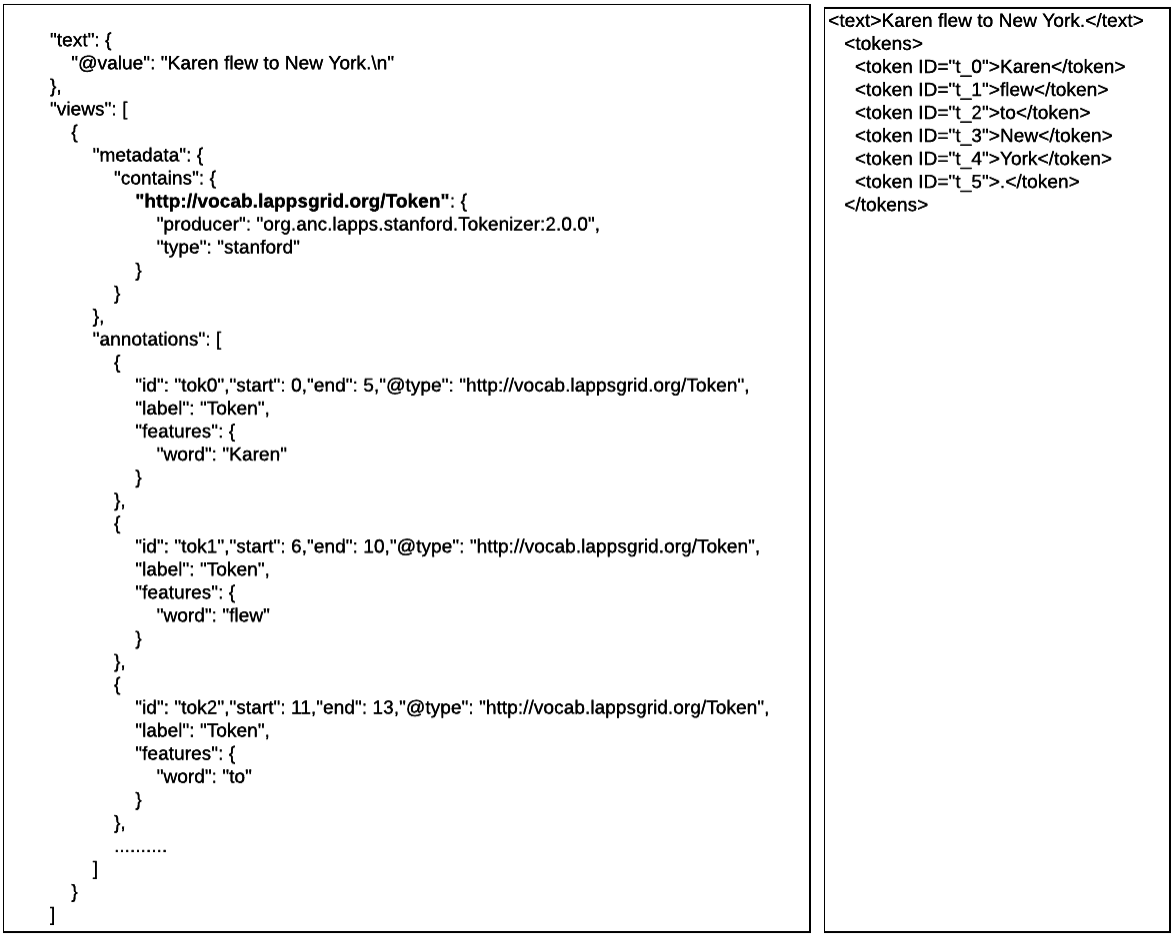

Figure 1 shows token-level annotations in LIF (on the left) and TCF (on the right).

Another challenge was encountered in converting LIF to TCF when the structures of the two document types were considered. In TCF, each annotation type (e.g. token, part-of-speech, etc) is stored in a separate XML element, whereas in LIF multiple annotation types can be present in a single JSON object (we are referring to LIF views here). Furthermore, LIF allows multiple occurrences of the same annotation type within a document, whereas TCF allows only one occurrence of an annotation type per document. This can be seen in Figure 2, where tokens and part-of-speech annotations are contained in separate elements in TCF, but are grouped together in LIF. It can also be seen that the LIF document in Figure 2 contains multiple part-of-speech annotations. A problem arises when multiple annotations of the same type are present in a LIF document. Only one set of annotations can be chosen for conversion into TCF. The strategy used in the LIF to TCF converter was to choose the last set of annotations within the

document, since tools add annotations to the end of a document, and the last set of annotations is often the most informative.

These challenges, along with others, and technical implementation details were discussed at the kick-off meeting in Prague in March. Developers from each site remained in Prague for several days after the main kick-off meeting and implemented prototypes for the data converters. Two packages were created, one for each conversion direction. The software was further developed and improved after the meeting and made available on GitHub (LIF to TCF[13] and TCF to LIF[14]). The converters are now capable of transforming, in their respective format directions, most of the core annotations used in natural language processing, including Text, Tokens, Part-of-Speech tags, Lemmas, Sentence, Named Entities, Dependency Parsing, and Constituent Parsing.

[13] https://github.com/SfS-ASCL/service-lapps-converter
[14] https://github.com/lappsgrid-incubator/tcf-converter-prototype

```
  "text": {
     "@value": "Karen flew to New York.\n"
  },
  "views": [
     {
        "metadata": {
           "contains": {
              "http://vocab.lappsgrid.org/Token": {
                 "producer": "org.anc.lapps.stanford.Tokenizer:2.0.0","type": "stanford"
              }
           }
        },
        "annotations": [
           {
              "id": "tok0","start": 0,"end": 5,"@type": "http://vocab.lappsgrid.org/Token","label": "Token",
              "features": {
                 "word": "Karen"
              }
           },
           ....................
        ]
     },
     {
        "metadata": {
           "contains": {
              "http://vocab.lappsgrid.org/Token#pos": {
                 "producer": "edu.brandeis.cs.lappsgrid.opennlp.POSTagger:2.0.2","type": "tagger:opennlp"
              }
           }
        },
        "annotations": [
           {
              "id": "pos_0","start": 0,"end": 5,"@type": "http://vocab.lappsgrid.org/Token#pos",
              "features": {
                 "pos": "NNP"
              }
           },
           ....................
        ]
     },
     {
        "metadata": {
           "contains": {
              "http://vocab.lappsgrid.org/Token#pos": {
                 "producer": "org.anc.lapps.stanford.Tagger:2.0.0","type": "tagset:penn"
              }
           }
        },
        "annotations": [
           {
              "id": "tok0","start": 0,"end": 5,"@type": "http://vocab.lappsgrid.org/Token","label": "Token",
              "features": {
                 "pos": "NNP","word": "Karen"
              }
           },
           ....................
        ]
     }
  ]
}
```



```
<text>Karen flew to New York.
</text>

  <tokens>
    <token ID="t_0">Karen</token>
    <token ID="t_1">flew</token>
    <token ID="t_2">to</token>
    <token ID="t_3">New</token>
    <token ID="t_4">York</token>
    <token ID="t_5">.</token>
  </tokens>

  <POStags tagset="pennTB">
    <tag tokenIDs="t_0">NNP</tag>
    <tag tokenIDs="t_1">VBD</tag>
    <tag tokenIDs="t_2">TO</tag>
    <tag tokenIDs="t_3">NNP</tag>
    <tag tokenIDs="t_4">NNP</tag>
    <tag tokenIDs="t_5">.</tag>
  </POStags>
```


Figure 2: An example of multiple parts-of-speech layers (OpenNLP and Stanford) in LIF (on the left). As can be seen from the figure, only one set of annotations (the last one) is chosen for conversion into TCF (on the right).

T2.2 Webservice Metadata

The metadata of a web service provides all information needed in order to interact with it. Such metadata includes a description of functionalities offered by a service, pre and postconditions, and specifications of data that is consumed and produced by a service. This information makes it possible to invoke a service successfully, providing it with the data it needs for processing, and enabling interpretation of the results. WebLicht and LAPPS Grid use different methods for storing and fetching service metadata. In order to integrate the two systems, making it possible for both systems to invoke their combined set of tools, it is necessary for each system to have access to the web service metadata of the other. In this reporting period, the current state of tool metadata in both frameworks was reviewed.

WebLicht webservice metadata is based on the Component Metadata Infrastructure (CMDI)[15], a framework developed within CLARIN that provides a way to describe and reuse metadata components. Components are building blocks of information (e.g. name or email) which can be grouped to form profiles (e.g. contact-person). Both the components themselves and the profiles that are built with them are stored in the CLARIN Component Registry. Metadata for WebLicht tools are stored in repositories at each CLARIN center, in the CMDI WebLichtWebService[16] profile format, which includes:

- General Information about the tool (name, description, PID, etc)
- Expected Input (data type and annotations required to be in the input)
- Output Produced (data type and list of annotation layers added)
- URL + query parameters

After creating a CMDI description of a service and entering it in a CLARIN repository, the metadata will be fetched by the WebLicht harvester. The harvester periodically checks all the center repositories for WebLicht-compatible web service metadata and fetches them via the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH[17]), which is implemented in all CLARIN center repositories.

LAPPS Grid uses the Simple Object Access Protocol (SOAP)[18], a messaging protocol for exchanging information via the internet, for invoking web services. In the SOAP protocol, each web service provides its own metadata. Brandeis University and Vassar College maintain repositories of LAPPS Grid web services and provide service discovery functionalities to users and applications. LAPPS Grid services provide the following metadata information:

- General information about the tool (name, description, vendor, licensing)
- Input requirements (data type, language and encoding, required previous annotations)
- Output produced (data type, language and encoding, output annotations)

[15] http://www.clarin.eu/cmdi
[16] https://catalog.clarin.eu/ds/ComponentRegistry/#/?_k=96wfwx
[17] https://www.openarchives.org/pmh/
[18] https://www.w3.org/TR/soap12/

On the basis of the review, it was concluded that tool metadata in both systems contains sufficient information to enable invocation of the web services from the other system. Initially, metadata for LAPPS tools will be added to the Tübingen repository. To access WebLicht services from LAPPS, tool metadata will be converted to the required format and hosted by the LAPPS Grid.

In the next phase of the project, we will consider the feasibility and necessity of additional metadata repositories. We will also further explore the technical aspects of invoking the web services for which metadata has been fetched.