Bridging the LAPPS Grid and CLARIN

Erhard Hinrichs*, Nancy Ide, James Pustejovsky†, Jan Hajič‡, Marie Hinrichs*, Mohammad Fazleh Elahi*, Keith Suderman**, Marc Verhagen†, Kyeongmin Rim†, Pavel Straňák‡, Jozef Mišutka‡**

*University of Tübingen, **Vassar College, †Brandeis University, ‡Charles University

{erhard.hinrichs, marie.hinrichs, mohammad-fazleh.elahi}@uni-tuebingen.de, {ide, suderman}@cs.vassar.edu, {jamesp, marc}@cs.brandeis.edu, krim@brandeis.edu,{hajic, stranak, misutka}@ufal.mff.cuni.cz

Abstract

The LAPPS-CLARIN project is creating a "trust network" between the Language Applications (LAPPS) Grid and the WebLicht workflow engine hosted by the CLARIN-D Center in Tübingen. The project also includes integration of NLP services available from the LINDAT/CLARIN Center in Prague. The goal is to allow users on one side of the bridge to gain appropriately authenticated access to the other and enable seamless communication among tools and resources in both frameworks. The resulting "meta-framework" provides users across the globe with access to an unprecedented array of language processing facilities that cover multiple languages, tasks, and applications, all of which are fully interoperable.

Keywords: Language Applications Grid, WebLicht, Text Corpus Format (TCF), LAPPS Grid Interchange Format (LIF), syntactic interoperability, semantic interoperability, user identification and authentication

1. Introduction

The Andrew K. Mellon Foundation has funded a project to create a "trust network" between the Language Applications (LAPPS) Grid (Ide et al., 2014), a major framework for composing pipelines of natural language processing (NLP) tools, and the WebLicht workflow engine (Dima et al., 2012) hosted by the CLARIN-D Center in Tübingen. The project also includes integration of NLP services available from the LINDAT/CLARIN Center in Prague[1]. The goal is to allow users on one side of the bridge to gain appropriately authenticated access to the other and enable seamless communication among tools and resources in both frameworks. The resulting "meta-framework" provides users across the globe with access to an unprecedented array of language processing facilities that cover multiple languages, tasks, and applications, all of which are fully interoperable.

The LAPPS Grid/CLARIN Mellon project involves two major tasks: (1) establishing a joint single sign-on user authentication and authorization mechanism; and (2) enabling seamless interoperability at both the syntactic and semantic levels among tools available from both the LAPPS Grid and WebLicht, so that users can mix and match these tools regardless of provenance and without concern for differing I/O requirements. In this paper we describe the work required to accomplish these tasks.

2. Overview

In the LAPPS Grid, language resources and NLP tools are made available as web services through the Galaxy workflow engine and interface (Giardine et al., 2005), as well as programmatic access through the LAPPS Grid API[2]. LAPPS Grid tools consume and produce data in the LAPPS Interchange Format (LIF) (Verhagen et al., 2015), a JSON-LD-based format designed to serve as an internal interchange format for linguistically annotated data. Semantic interoperability among services is accomplished via URI references to the LAPPS Grid Web Service Exchange Vocabulary (WSEV) (Ide et al., 2016). NLP tools are accessed as web services that deliver metadata about the content at a standardized URI and are at present invoked using the SOAP protocol.

WebLicht is an environment for building, executing, and visualizing the results of NLP pipelines, which is integrated into the CLARIN infrastructure (Hinrichs and Krauwer, 2014). WebLicht NLP tools are implemented as web services that consume and produce the Text Corpus Format (TCF)[3] data, an XML format designed for use as an internal data exchange format for WebLicht processing tools. The TCF also ensures semantic interoperability among all WebLicht tools and resources by defining a common vocabulary for linguistic concepts. Metadata descriptions of WebLicht tools are stored in repositories located at the CLARIN center hosting the service. WebLicht web services are invoked using the RESTful protocol.

LINDAT/CLARIN provides various NLP services[4] based on single-purpose tools or pre-configured chains of tools, often for multiple languages, most notably the UDPipe service (Straka et al., 2016). UDPipe produces CoNLL-U[5], the Universal Dependencies (UD) annotation format (Nivre et al., 2016), which is a revised version of the CoNLL-X format (Buchholz and Marsi, 2006) used in the Conference on Natural Language Learning exercises. Sub-chains of UDPipe are being exposed in WebLicht and will be made interoperable with WebLicht's TCF-based tools; conversion to LIF may then be accomplished by converting from TCF to LIF. Access to UDPipe tools are also accessed via WebLicht.

The main challenges to bridging The LAPPS Grid and WebLicht frameworks arise from differences in the architectures of the two systems, in particular the differences in

[1] https://lindat.mff.cuni.cz/en

[2] http://wiki.lappsgrid.org/Developing.html

[3] https://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/The_TCF_Format

[4] http://lindat.mff.cuni.cz/services/

[5] http://universaldependencies.org/format.html

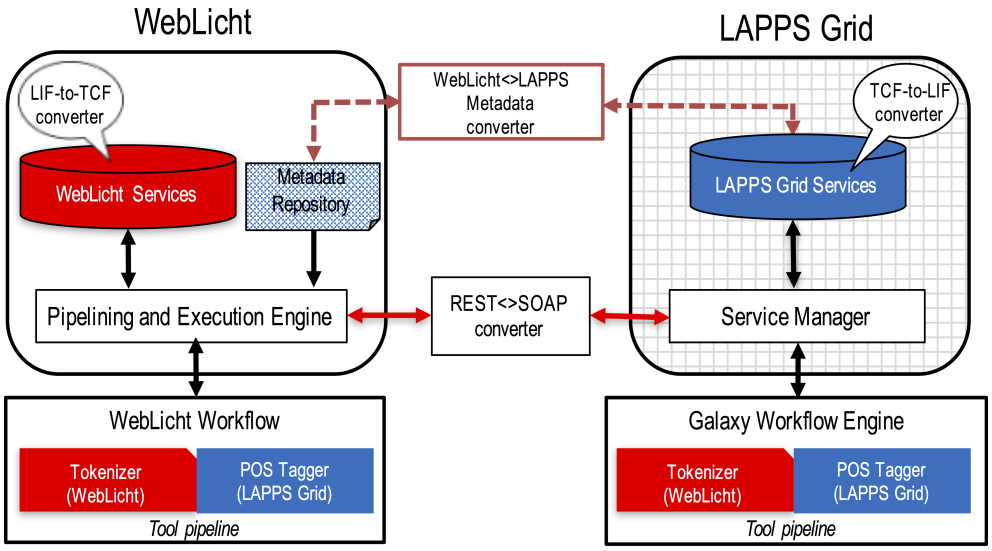

Figure 1: Integration framework

data exchange formats, access to and format of metadata, and the protocols used to invoke web services. In addition, it is necessary to provide support for authentication and authorization mechanisms that allow users to access resources and services provided by each framework as easily and seamlessly as those within the framework they typically use. Each of these tasks is described in the following sections.

3. Communication Protocols and Metadata Conformance

The main challenge in achieving interoperability with regard to communication protocols lies in the way that the services in each framework are implemented. The LAPPS Grid and CLARIN services use different communication protocols. The LAPPS Grid uses the Simple Object Access Protocol (SOAP), whereas the CLARIN tools are implemented as RESTful[6] services. SOAP services send and receive data in SOAP-XML, which is an XML wrapper around a request or response message. RESTful services, on the other hand, send and receive messages directly through HTTP requests or responses. This means that in order to invoke LAPPS services registered in CLARIN, it is necessary to wrap CLARIN's RESTful requests into LAPPS Grid SOAP requests. A SOAP-PROXY service has been implemented to take a REST service request as input, convert it to a SOAP message, invoke the service with the SOAP request, and return the response from the service. WebLicht services can be invoked directly at their entry points via plain HTTP requests from the LAPPS Grid. A similar mechanism enables access to CLARIN RESTful services from the LAPPS Grid.

4. Metadata Conformance

Metadata about the web services available in LAPPS Grid and WebLicht contain information needed to invoke the services from within their respective frameworks. The two frameworks handle web service metadata differently with respect to content, storage location, and fetching.

In the LAPPS Grid, each web service delivers its own metadata on demand, whereas WebLicht web service metadata, which follows the specifications of CLARIN CMDI[7] framework, is retained in CLARIN Center repositories. The WebLicht metadata stored in the repositories includes information corresponding to LAPPS Grid metadata as well as additional details about the format and contents of a service's input and output. In the LAPPS Grid, details about format and contents of a service's input and output are made available as needed by inter-service queries. WebLicht metadata can be converted to LAPPS Grid metadata automatically, but because WebLicht metadata includes additional information beyond that provided by LAPPS Grid services, there is potential information loss; however, because WebLicht metadata is stored in a registry, it can be restored in a WebLicht→LAPPS Grid→Weblicht round trip. LAPPS Grid metadata cannot be automatically converted to WebLicht metadata because of WebLicht's requirement to store the information in the registry; to handle this, it is necessary to store LAPPS Grid metadata in the WebLicht registry manually, and update it when required.

Figure 1 shows the LAPPS Grid-WebLicht integration framework. When one framework calls a service from the other, metadata from the called service is converted and made available to the other, after which it can be processed with the caller's usual handlers. Similarly, data conversion services allow each platform to consume and produce data in its native format. Service calls are tunneled through a proxy, which invokes services using the required protocol.

5. Syntactic Interoperability

The problem of differing data exchange formats has been addressed at the syntactic level by implementing converters between LIF and TCF as web services and registering them in both frameworks. WebLicht is currently

[6] https://www.w3.org/2001/sw/wiki/REST

[7] http://www.clarin.eu/cmdi

in the process of integrating tools that process UDPipe's CoNLL-U format in order to accommodate tools created in LINDAT/CLARIN. Implementation with full interoperability requires the creation of converters between CoNLL-U and TCF; because the LAPPS Grid will access LINDAT/CLARIN tools via the WebLicht portal, conversion between CoNLL-U and LIF will be performed indirectly, with TCF as the liaison. Note that at present we address conversion from CoNLL-U to other formats but not conversions from other formats to CoNLL-U.

Structural differences and the granularity of annotation data in LIF, TCF, and LINDAT/CLARIN's CoNLL-U format imposed several challenges, outlined below.

5.1. Annotation Layers

Between TCF and LIF, the major differences result from the ways in which annotation *layers* (called *views* in LIF) are defined. TCF has a fixed number of annotation layers (shown in the left column of Table 1) and allows only one occurrence of a given annotation layer per document. Each layer has a fixed structure, most consisting of flat lists of XML elements, although some layers are slightly more structured.

In contrast, LIF does not place restrictions on the number and content of its views, and each service can add any number of views[8] or add to an existing view, as long as the metadata in the view provides the relevant information about the view's content. All views in LIF have the same structure, with annotations consisting of a list of elements that follow the same template: an annotation object with a type, an identifier, beginning and end character offsets or a reference to other annotations (in the same view or another) and a dictionary of feature/value pairs.

Figure 3 shows a token layer in LIF and the corresponding token and POSTags layers in TCF. Note that in TCF part-of-speech tags appear in a separate layer referring to token objects in the token layer, while in LIF, part-of-speech is given as the value of the *pos* feature.[9] Therefore, in this case conversion requires either expanding one LIF view into two TCF layers, or collapsing two TCF layers into one LIF view.

Because TCF allows only one occurrence of an annotation type per document, if a LIF document contains multiple LIF annotations for the same phenomenon only one can be chosen for conversion into TCF. This makes it necessary to identify an optimal alternative; more problematically, it means that there is potential information loss (i.e., loss of additional alternative views) when converting from LIF to TCF. Thus a round trip from LIF to TCF back to LIF may not produce the same result. This remains an open problem at this time; currently, the last view for any given annotation type is included in the TCF representation, and the original LIF document may be stored in its entirety in TCF's `textSource` element, to be restored upon its return.

CoNLL-U, an example of which is shown in Figure 2, differs substantially on the surface from both TCF and LIF in terms of physical format. We could regard the set of columns as a fixed set of annotation layers, most consisting of single elements while others have internal structure, and allowing only one column per phenomenon. Conceptually, the information in a CoNLL-U representation corresponds to the token, lemma and part-of-speech layers in TCF (which correspond to the Token layer in LIF) and the dependency parse layer in both TCF and LIF.

5.2. Anchoring to Primary Data

LIF is a stand-off annotation format, which requires that all annotations refer to either character offsets in the primary data or to other annotations that are themselves either directly or indirectly (via a chain of annotations) anchored in primary data. TCF annotations are not grounded in the primary data but instead refer to a single base layer consisting of tokens[10]. Primary data is stored in the TCF `text` element, but in most cases no anchors into the text are provided. Therefore, conversion from LIF to TCF requires mapping character offset anchors to each token element, and conversion in the reverse direction demands recomputing offsets from the primary source.

As a column-based format, CoNLL-U does not provide the primary data source, and therefore conversion from CoNLL-U to TCF first requires reconstruction of a "source text" from the list of surface tokens.[11] To convert to LIF, character offsets into the source text must also be computed, as LIF requires anchoring of at least one annotation in primary data.

5.3. Tree and Graph Structures

TCF represents the tree structure of a constituency parse by exploiting the hierarchical nesting of XML tags, whereas LIF, which is a JSON-based format, represents the tree explicitly by providing the ID of the parent and the IDs of all children for each constituent. Conversion from TCF to LIF therefore requires interpreting and flattening the XML structure.

CoNLL-U provides head-deprel pairs for each token. In TCF and LIF, these relations are represented using references to the IDs of relevant entities as the value of features or attributes such as "governor" and "dependent", as shown in Figure 4. Conversion from CoNLL-U is a straightforward matter of deconstructing the micro-format to generate the corresponding TCF and LIF representations.

5.4. Multi-word Tokens

CoNLL-U includes means to represent multi-word tokens that correspond a single surface token (e.g., *want* and *to* for surface string *wanna* in English, or *in* and *dem* for surface string *im* in German), which are interspersed among surface tokens in the same column. When multi-word tokens are present, CoNLL-U annotations apply to only the word comprising the multi-word token; thus, the surface form is irrelevant for the purposes of processing annotations. Therefore, conversion of CoNLL-U's multi-word tokens into TCF specifies the surface form and its parts tin

[8] In practice, most tools add a single view. [9] Lemmas are handled the same way as part-of-speech tags in both schemes.

[10] An exception is the synonomy layer, which refers to lemma identifiers that in turn reference the token layer. [11] The column labels used in CoNLL-U are given in Table 4, below.

1	They	they	PRON	PRP	Case=Nom|Number=Plur	2	nsubj	2:nsubj|4:nsubj
2	buy	buy	VERB	VBP	Number=Plur|Person=3|Tense=Pres	0	root	0:root
3	and	and	CONJ	CC	_	4	cc	4:cc
4	sell	sell	VERB	VBP	Number=Plur|Person=3|Tense=Pres	2	conj	0:root|2:conj
5	books	book	NOUN	NNS	Number=Plur	2	obj	2:obj|4:obj
6	.	.	PUNCT	.	_	2	punct	2:punct

Figure 2: CoNLL-U representation of "They buy and sell books."

Figure 3: Example LIF (top) and TCF (bottom) formats

Figure 4: Example TCF and LIF representations of dependency relations

attributes on the TCF `token` element. In LIF, multi-word tokens can appear in an additional token layer, which can in turn reference the corresponding surface token in the surface token layer. The multi-word tokens can then be referenced from other annotations.

6. Semantic Interoperability

The three frameworks in this project reference different sets of linguistic objects (with some overlaps), using differing terminology and expressing relations among these objects in differing configurations. To enable semantic interoperability among the services in the three frameworks, we provide means to specify the linguistic objects that a given service or tool requires as input and produces as output, so that other producers and consumers (i.e., other services and/or tools) can determine if its requirements are satisfied. In a pipeline of tools or web services, this information is provided as metadata that must be checked automatically for compatibility. This in turn demands that identical concepts must be identified as such, either by direct match or by reference to a common web-addressable entity. Internally, a given tool may use different terminology; the only necessity is that the tool is wrapped to map the exchange vocabulary into the internal terminology and vice versa. Thus in principle, a single mapping of a tool's specific terminology into and out of the common exchange vocabulary is sufficient to enable information exchange with all others.

At present, TCF concepts are briefly defined in comments to TCF's XML schema. In LIF, concepts are linked to URI-addressable definitions in the Web Service Exchange Vocabulary (WSEV)[12]. The LINDAT/CLARIN tools that are being made available deal primarily with dependency annotation and produce annotations conformant to the Universal Dependencies specifications[13] for treebank annotation. Conversion among TCF, CoNLL-U, and LIF thus indirectly links all three frameworks' vocabularies to the WSEV.

We have developed a mapping and linkage among concepts defined in the the vocabularies of WebLicht, LINDAT/CLARIN, and the WSEV to cover entities contained in the others. All three vocabularies cover similar linguistic phenomena and overlap in many instances, and in fact, semantic mapping has proved to be much more straightforward than originally expected. The most common modification involves extending specifications in the vocabularies to accommodate concepts in the others.

TCF has a fixed number of annotation layers, each of which can be construed as representing a concept in the TCF vocabulary. Table 1 shows the mapping from each of TCF's layers (listed in the left column) to WSEV vocabulary terms *before* modification for conformity of the two schemes, with an indication of required modifications. Table 2 shows the reverse mapping, from WSEV concepts to TCF layers. As noted, there are a number of cases where there is no equivalent for an item in one scheme or another; however,

[12] http://vocab.lappsgrid.org [13] http://universaldependencies.org

TCF layer	WSEV equivalent
token	Simple mapping to Token
sentence	Simple mapping to Sentence
lemma	Map to Token#lemma
POStag	Map to Token#pos
parsing	Simple mapping to PhraseStructure and Constituent. Requires conversion of the XML tree structure to an explicit directed graph.
depparsing	DependencyStructure, Dependency
namedEntities	Simple mapping to NamedEntity
references	Map to Coreference. Add features to correspond to TCF's *heads*, *type*, *rel*, etc.
textstructure	Paragraph and Sentence. TCF's page element has no equivalent in WSEV.
synonymy	No equivalent
matches	No equivalent
wordSplittings	No equivalent
geo	No equivalent
phonetics	No equivalent
orthography	No equivalent
wsd	No equivalent

Table 1: TCF layer mappings to WSEV

LIF view	Mapping to TCF
Token	Token information is distributed over the tokens, lemmas and POStags layers
Sentence	Simple mapping to sentences layer
Paragraph	Map to the textstructure layer
NounChunk VerbChunk	No equivalent in TCF.
NamedEntity	Simple mapping to namedEntities layer
Markable	No equivalent in TCF
Coreference	Map to the references layer. Add Markables to TCF. Move Markable and Token annotations to another TCF layer.
PhraseStructure Constituent	Map to the parsing layer. If the view contains Tokens add to the tokens layer.
DependencyStructure Dependency	Map to the depparsing layer. If the view contains Tokens add to the tokens layer.
Relation	No equivalent in TCF
SemanticRole	No equivalent in TCF

Table 2: WSEV to TCF mapping

for most of these the creation of new items for conformance is straightforward.

In general, the XML elements in TCF layers correspond either to vocabulary items in the WSEV or one of its features, as shown in the detailed mapping from elements in the TCF parsing layer in Table 3. In this table, the elements to the left of the number sign refer to elements defined as tags in the TCF XML schema or to categories in the LAPPS WSEV vocabulary, and the elements on the left are properties. The @ sign indicates that the property is a metadata property in WSEV.

TCF Element	WSEV equivalent
parsing#tagset	PhraseStructure@categorySet
parse	PhraseStructure
parse#ID	PhraseStructure#id
constituent	Constituent
constituent#ID	Constituent#id
constituent#cat	Constituent#category
constituent#edge	No mapping
constituent#tokenIDs	Constituent#children
cref	No mapping
cref#constID	No mapping
cref#edge	No mapping

Table 3: TCF to WSEV mapping for elements of the parsing layer

The LAPPS Grid represents phrase structure with explicit labeled edges between nodes, using `parent` and `child` relations on the `Constituent` annotation type. TCF, on the other hand, represents phrase structure implicitly by embedding XML elements; only when an element's (node's) are all tokens are the children listed explicitly in a `tokenIDs` element. Note that WebLicht's `cref` object allows a secondary edge to another constituent, which in effect turns the tree into a directed graph. However, graphs are not widely used in WebLicht services; if needed for use in a TCF layer other than phrase structure, one of the relation annotation types in the WSEV may be added to the TCF schema for use there.

It is important to note that semantic interoperability involves only the concepts themselves, and not the way they may be structured in a given scheme. So, for example, a TCF concept may be represented as the name of an XML element, whereas it could appear as a feature associated with a primary WSEV concept type. This is the case, for example, for part of speech in the renderings that were shown in Figure 3, where part of speech ("tag") in TCF is not only an element name, but also appears in a different TCF layer, whereas in LIF *pos* is a feature associated with the Token vocabulary item and is included in the token view.

Mapping concepts in CoNLL-U to TCF and the WSEV is relatively straightforward but requires adding several concepts to the target vocabularies. Table 4 shows the CoNLL-U row labels, which correspond to a higher-level set of concepts in the CoNLL-U vocabulary, and most of which exist in or have been added to TCF and the WSEV as concepts or features. Figure 2 shows that a representation in the CoNLL-U format includes additional concepts (e.g., "Number", "Case", "Person") in the FEATS column, which correspond to features on Tokens (and other WSEV vocabulary items) and to items in TCF's morphology layer.

6.1. Discussion

The exercise of making LIF, TCF, and CoNLL-U interoperable showed us that for the schemes we dealt with, the

ID	Word index (range for multiword tokens, decimal number for empty nodes)
FORM	Word form or punctuation symbol
LEMMA	Lemma or stem of word form
UPOSTAG	Universal part-of-speech tag
XPOSTAG	Language-specific part-of-speech tag
FEATS	List of morphological features
HEAD	Head of the word
DEPREL	UD relation to the HEAD
DEPS	List of head-deprel pairs
MISC	Anything else

Table 4: CoNLL-U concepts

greatest obstacles to interoperability were due to variations in representation formats–i.e., at the level of syntax–rather than to variations in semantic categories. Without exception, the semantic categories used in all three schemes were either identical to concepts defined in the other vocabularies or were easily added where missing. This commonality among the three schemes can be largely attributed to the fact that all three schemes deal broadly with basic concepts that have been widely used in the field, such as *token, part-of-speech*, and elements of *phrase structure, dependency structure*, etc. Both LIF and TCF are intended to be general-purpose schemes; only CoNLL-U deals with a specific phenomenon in depth, and even in this case the concepts used are relatively well-established for dependency analysis. Even where *category labels* may differ, the concepts they represent are a part of the common set of objects used in annotation tools.

Another reason why semantic interoperability posed fewer problems in our exercise is that we do not attempt to harmonize what are commonly called *tagsets*, but rather require clear identification of the tagset used in the annotation (e.g., part-of-speech, dependency relation, constituent name etc.) in metadata[14]. There have been attempts to map and/or harmonize such values (e.g., OLiA (Chiarcos, 2008)), which have shown how problematic this kind of mapping can be.

Although we have skirted the issue of tagset interoperability, we argue that any attempt to achieve interoperability at this level would impede our ability to move forward. We focus instead on *tool* interoperability, by requiring metadata identifying the tagset used in a given annotation, and designing our services to check that the tagsets required as input for one tool are satisfied by the tagsets appearing in the output of another. This means that a tool requiring, say, the Penn part-of-speech tags, will effectively "refuse" input from a tool whose output uses another tagset. This obviously places limits on full semantic interoperability; however, in our experience, it is necessary to recognize the distinction between object/feature *names* and their values in order to make progress toward full interoperability–even if in stages.

We do not argue that semantic interoperability for linguistically annotated resources is "solved" or even close to being solved; but our experience suggests that there is evidently a fair degree of commonality among several linguistic annotation schemes, at least in terms of the concepts included. Our experience suggests that to move forward, the quest for semantic interoperability should be further sub-divided into identification of *categories* (or objects) and *values*, and both should be addressed separately. More crucially, it suggests that format differences may pose a far greater obstruction to overall interoperability than assumed. This in turn suggests that in designing annotation schemes and formats as well as attempting mappings among different schemes, it is critical to clearly separate issues of format (syntax) from annotation scheme semantics.

7. Example

Figure 5 presents an example use of tools from both the LAPPS Grid and WebLicht frameworks, accessed via the WebLicht user interface. An input text corpus is converted to LIF format, tokenized and sentence-split by LAPPS services, followed by a LIF-to-TCF format conversion to allow processing to continue using CLARIN services. The lower window in the figure *Input and Chain Selection* shows the tool chain that was selected for execution. After the LAPPS Grid services (Stanford Tokenizer and Stanford Splitter) are executed, the LIF-to-TCF converter is used to return to the WebLicht plaform; the upper window *Next Choices* shows the available WebLicht services. To revert to using LAPPS Grid services again, the user chooses TCF-to-LIF; in this way, a user can alternate between LAPPS Grid services and WebLicht services and vice versa, without ever leaving the WebLicht interface.

Figure 6 shows a portion of the LAPPS Grid Galaxy interface and a workflow in which a tokenizer and part-of-speech tagger from WebLicht are invoked, followed by a named entity recognizer from the LAPPS Grid. The LAPPS Grid Galaxy interface automatically detects and converts formats as needed, without intervention from the user; at the time of this writing detection and conversion for TCF has not been implemented, and therefore the TCF-to-LIF converter is explicitly inserted into the pipeline in order to feed into the LAPPS Grid entity recognize.

8. User Authentication and Identification

Using the LAPPS Grid through the Galaxy interface requires a simple registration in order to provide a uniquely-named workspace for each user, but there are no license requirements or usage restrictions depending on the user type or affiliation. The LAPPS Grid can be used directly via its API[15] without registration or any other restriction. When it is necessary to provide secure access to licensed data and software, the LAPPS Grid employs "click through" licenses that can be accepted in real time as well as verification via timed tokens (Cieri and DiPersio, 2014).

In the CLARIN infrastructure, users must be authenticated via identity providers (IdPs) belonging to EU national academic identity federations. Identity providers are typically

[14] This is true for TCF and LIF; CoNLL-U explicitly specifies values for some categories (UPOSTag and DEPREL, for example) that are to be used.

[15]

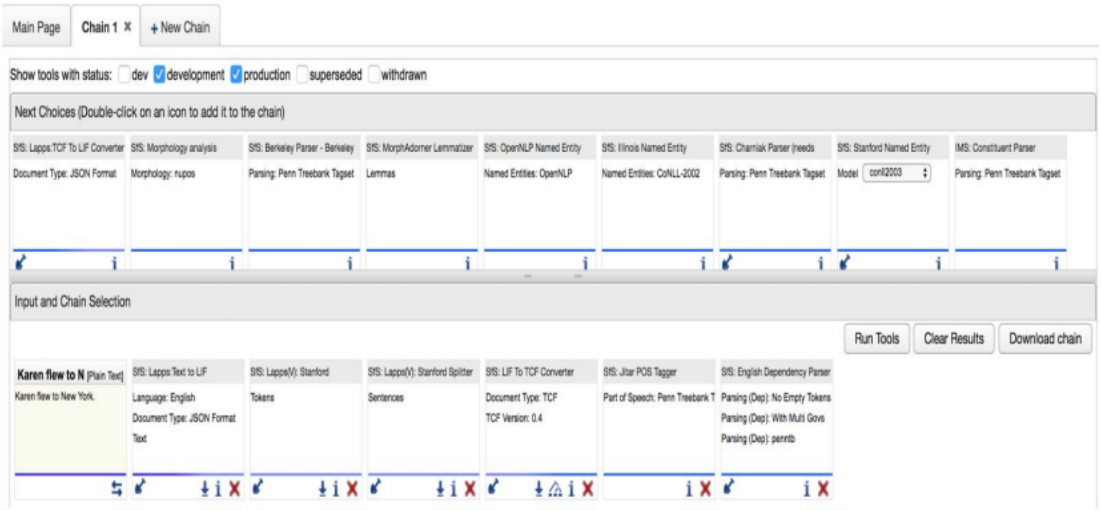

Figure 5: Invoking LAPPS Grid services from WebLicht.

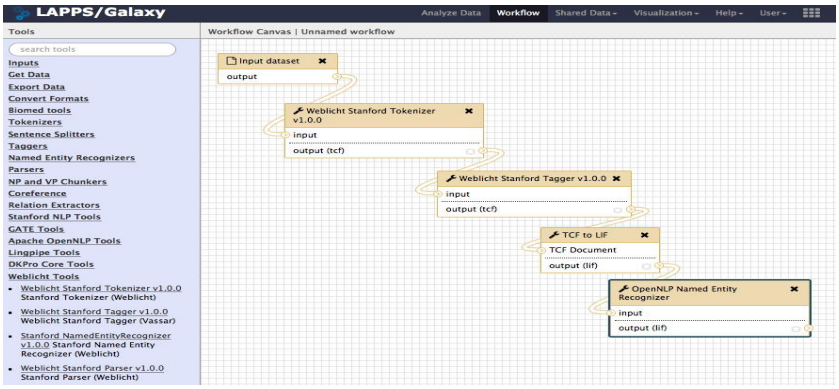

Figure 6: Invoking WebLicht services in a LAPPS Grid/Galaxy pipeline

universities or other academic institutions that have information about users and provide secure authentication (login) services. Service providers (SPs, such as WebLicht) can decide to trust users authenticated via an IdP. This secure system of identifiable user login via IdPs provides reasonable assurance that WebLicht services are being used for academic purposes.

Because of the need for authentication, prior to this project LAPPS Grid users were in general not able to access login-protected CLARIN services. To provide this access, we have devised means for LAPPS Grid users with appropriate credentials to access these services, by registering the LAPPS Grid as both a service and an identity provider with the CLARIN Service Provider Federation (SPF)[16]. The CLARIN SPF includes more than 1700 European institutions, whose members have access to all CLARIN services; academics whose institutions are not part of the inter-federation can be approved for use of CLARIN services through a process of in-person identity verification at their home institutions. Without further authentication, users registered in the LAPPS Grid may access only the publicly available services from Weblicht and other CLARIN centers. However, a LAPPS Grid user can be authenticated for full access to CLARIN services if he or she is a member of an academic institution in the InCommon identity management federation[17], a secure and privacy-preserving trust fabric for research and higher education in the United States that performs a similar function as the national federations in Europe that form the CLARIN SPF federation.

Specific software (e.g., Shibboleth software[18]) has to be installed on the LAPPS Grid and WebLicht servers in order for them to act as identity/service providers. "Trusted" users are then identified either individually or using aggregated feeds of entities such as InCommon CLARIN SPF. The service provider must in turn join these (inter-)federations, to ensure the trust is mutual; otherwise, the service provider would have to negotiate the trust with each and every identity provider. Once the trust has been secured, users from a trusted identity provider can login to a trusted service provider by authenticating via their institution's login page, after which the identity provider sends a confirmation with additional attributes (e.g., id, email, name, affiliation, entitlement) to the service provider via a secure channel.

[16] https://www.clarin.eu/content/service-provider-federation

[17] https://www.incommon.org/

[18] https://www.shibboleth.net/

To summarize, users who can verify an academic affiliation through InCommon, an academic identity provider at their own institution, or the CLARIN SPF identity provider may use all services from either the LAPPS Grid or Weblicht. Others may use only services that are openly available (this includes the majority of services in the LAPPS Grid). In addition, our access control solution supports *single sign-on* in order to minimize the burden of requiring multiple credentials and/or re-entering credentials repeatedly.

9. Conclusion

The meta-framework providing for mutual access between the LAPPS Grid and the two CLARIN frameworks has the potential to transform scholarship and development across multiple disciplines in the sciences, language and social sciences, and digital humanities by providing a transparent interface to a massive range of tools and resources at an unprecedented level.

Bridging the LAPPS Grid and WebLicht frameworks significantly extends the capabilities of each by providing seamless access to services that are currently unavailable in each. For example, the LAPPS Grid will benefit from availability of a more extensive suite of tools for output visualization than currently exists in the LAPPS Grid, and WebLicht will gain access to the sophisticated evaluation services the LAPPS Grid provides.

The potential impact extends even farther than the two frameworks involved, as both the LAPPS Grid and WebLicht are federated with other frameworks to which they provide a gateway. WebLicht is a member of the EU CLARIN network and therefore provides access to multilingual tools and resources from CLARIN Centers hosted throughout Europe. The harmonization will also extend to Asia because the LAPPS Grid is federated with seven other grids[19], including the Language Grid housed at Kyoto University[20]. Like the LAPPS Grid-CLARIN bridge, this federation provides interoperability and seamless access among atomic and composite web services available from any of the grids involved.

A more wide-ranging impact of this project may result from its success in providing interoperable access to services in three major frameworks that were developed entirely independently. Although we acknowledge that universal interoperability for NLP tools is far from a solved problem, we believe this project takes an important step towards its achievement. In particular, the exercise of pursuing semantic interoperability among the three frameworks has yielded new insights into the nature and source of obstacles to interoperability that could significantly impact future progress towards this seemingly elusive goal.

Our solutions to the problems of authentication, authorization, and access to licensed data and tools can serve as a model for other project facing the same issues. Finally, the work performed takes a major step toward the harmonization of software and data developed across the globe that can vastly ameliorate and eventually eliminate the current lack of reusability of resources and tools that thwarts research and development in the field and hampers collaboration. Ultimately, the LAPPS Grid-CLARIN meta-network may lay the groundwork for the eventual creation of a global network of grids and frameworks to serve researchers, developers, and users of NLP technologies.

[19] Federated Grid of Language Services (FGLS) (Ishida et al., 2014). [20] http://langrid.org

Acknowledgments

The work described here is supported by the Mellon Foundation Grant "Transatlantic Collaboration between LAPPS and CLARIN: Semantic, Technical and Infrastructural Interoperability of Services" and by the LINDAT/CLARIN Research Infrastructure, projects No. LM2015071 and CZ.02.1.01/0.0/0.0/16_013/0001781, supported by the Ministry of Education, Youth and Sports of the Czech Republic. CLARIN-D is funded by the German Federal Ministry of Education and Research (BMBF). The LAPPS Grid was developed under funding from the U.S. National Science Foundation SI^2 grants ACI 1147944 and ACI 1147912.

10. Bibliographical References

Buchholz, S. and Marsi, E. (2006). Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06, pages 149–164, Stroudsburg, PA, USA. Association for Computational Linguistics.

Chiarcos, C. (2008). An ontology of linguistic annotations. *LDV Forum*, 23(1):1–16.

Cieri, C. and DiPersio, D. (2014). Intellectual property rights management with web service grids. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, pages 93–100, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Dima, E., Hinrichs, E., Hinrichs, M., Kislev, A., Trippel, T., and Zastrow, T. (2012). Integration of weblicht into the clarin infrastructure. In *Proceedings of the Joint CLARIN-D/DARIAH Workshop at Digital Humanities Conference 2012: Service-oriented Architectures (SOAs) for the Humanities: Solutions and Impacts*, pages 17–23, Hamburg, Germany.

Giardine, B., Riemer, C., Hardison, R. C., Burhans, R., Elnitski, L., Shah, P., Zhang, Y., Blankenberg, D., Albert, I., Taylor, J., Miller, W., Kent, W. J., and Nekrutenko, A. (2005). Galaxy: a platform for interactive large-scale genome analysis. *Genome Research*, 15(10):1451–55.

Hinrichs, E. and Krauwer, S. (2014). The clarin research infrastructure: Resources and tools for ehumanities scholars. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Ide, N., Pustejovsky, J., Cieri, C., Nyberg, E., Wang, D., Suderman, K., Verhagen, M., and Wright, J. (2014). The language applications grid. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Ide, N., Suderman, K., Verhagen, M., and Pustejovsky, J. (2016). The language applications grid web service exchange vocabulary. In *Revised Selected Papers of the Second International Workshop on Worldwide Language Service Infrastructure WLSI (2015)*, pages 18–32, Kyoto, Japan. Springer-Verlag New York, Inc.

Ishida, T., Murakami, Y., Lin, D., Nakaguchi, T., and Otani, M. (2014). Open Language Grid–Towards a Global Language Service Infrastructure. In *The Third ASE International Conference on Social Informatics (SocialInformatics 2014)*, Cambridge, Massachusetts, USA.

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666, Paris, France. European Language Resources Association.

Straka, M., Hajič, J., and Straková, J. (2016). UDPipe: Trainable pipeline for processing coNLL-u files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4290–4297, Paris, France. European Language Resources Association.

Verhagen, M., Suderman, K., Wang, D., Ide, N., Shi, C., Wright, J., and Pustejovsky, J. (2015). The lapps interchange format. In *Revised Selected Papers of the Second International Workshop on Worldwide Language Service Infrastructure WLSI (2015)*, pages 33–47, Kyoto, Japan. Springer International Publishing.