The Language Application Grid Web Service Exchange Vocabulary

Nancy Ide
Department of Computer Science
Vassar College
Poughkeepsie, New York USA
ide@cs.vassar.edu

James Pustejovsky
Department of Computer Science
Brandeis University
Waltham, Massachusetts USA
jamesp@cs.brandeis.edu

Keith Suderman
Department of Computer Science
Vassar College
Poughkeepsie, New York USA
suderman@anc.org

Marc Verhagen
Department of Computer Science
Brandeis University
Waltham, Massachusetts USA
marc@cs.brandeis.edu

Abstract

In the context of the Linguistic Applications (LAPPS) Grid project, we have undertaken the definition of a Web Service Exchange Vocabulary (WS-EV) specifying a terminology for a core of linguistic objects and features exchanged among NLP tools that consume and produce linguistically annotated data. The goal is not to define a new set of terms, but rather to provide a single web location where terms relevant for exchange among NLP tools are defined and provide a "sameAs" link to all known web-based definitions that correspond to them. The WS-EV is intended to be used by a federation of six grids currently being formed but is usable by any web service platform. Ultimately, the WS-EV could be used for data exchange among tools in general, in addition to web services.

1 Introduction

There is clearly a demand within the community for some sort of standard for exchanging annotated language data among tools.[1] This has become particularly urgent with the emergence of web services, which has enabled the availability of language processing tools that can and should interact with one another, in particular, by forming pipelines that can branch off in multiple directions to accomplish application-specific processing. While some progress has been made toward enabling *syntactic interoperability* via the development of standard representation formats (e.g., ISO LAF/GrAF (Ide and Suderman, 2014; ISO-24612, 2012), NLP Interchange Format (NIF) (Hellmann et al., 2013), UIMA[2] Common Analysis System (CAS)) which, if not identical, can be trivially mapped to one another, *semantic interoperability* among NLP tools remains problematic (Ide and Pustejovsky, 2010). A few efforts to create repositories, type systems, and ontologies of linguistic terms (e.g., ISOCat[3], OLiA[4], various repositories for UIMA type systems[5], GOLD[6], NIF Core Ontology[7]) have been undertaken to enable (or provide) a mapping among linguistic terms, but none has yet proven to include all requisite terms and relations or be easy to use and reference. General repositories such as Dublin Core[8], schema.org, and the Friend of a Friend

[1] See, for example, proceedings of the recent LREC workshop on "Language Technology Service Platforms: Synergies, Standards, Sharing" (http://www.ilc.cnr.it/ltsp2014/).

[2] https://uima.apache.org/

[3] http://www.isocat.org

[4] http://nachhalt.sfb632.uni-potsdam.de/owl/

[5] E.g., http://www.julielab.de/Resources/Software/UIMA+type+system-p-91.html

[6] http://linguistics-ontology.org

[7] http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core/nif-core

[8] http://dublincore.org

Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT, pages 34–43,
Dublin, Ireland, August 23rd 2014.

project[9] include some relevant terms, but they are obviously not designed to fully cover the kinds of information found in linguistically annotated data.

In the context of the Linguistic Applications (LAPPS) Grid project (Ide et al., 2014), we have undertaken the definition of a Web Service Exchange Vocabulary (WS-EV) specifying a terminology for a core of linguistic objects and features exchanged among NLP tools that consume and produce linguistically annotated data. The work is being done in collaboration with ISO TC37 SC4 WG1 in order to ensure full community engagement and input. The goal is not to define a new set of terms, but rather to provide a single web location where terms relevant for exchange among NLP tools are defined and provide a "sameAs" link to all known web-based definitions that correspond to them. A second goal is to define relations among the terms that can be used when linguistic data are exchanged. The WS-EV is intended to be used by a federation of grids currently being formed, including the Kyoto Language Grid[10], the Language Grid Jakarta Operation Center[11], the Xinjiang Language Grid, the Language Grid Bangkok Operation Center[12], LinguaGrid[13], MetaNET/Panacea[14], and LAPPS, but is usable by any web service platform. Ultimately, the WS-EV could be used for data exchange among tools in general, in addition to web services.

This paper describes the LAPPS WS-EV, which is currently under construction. We first describe the LAPPS project and then overview the motivations and principles for developing the WS-EV. Because our goal is to coordinate with as many similar projects and efforts as possible to avoid duplication, we also describe existing collaborations and invite other interested groups to provide input.

2 The Language Application Grid Project

The Language Application (LAPPS) Grid project is in the process of establishing a framework that enables language service discovery, composition, and reuse, in order to promote sustainability, manageability, usability, and interoperability of natural language Processing (NLP) components. It is based on the service-oriented architecture (SOA), a more recent, web- oriented version of the pipeline architecture that has long been used in NLP for sequencing loosely-coupled linguistic analyses. The LAPPS Grid provides a critical missing layer of functionality for NLP: although existing frameworks such as UIMA and GATE provide the capability to wrap, integrate, and deploy language services, they do not provide general support for service discovery, composition, and reuse.

The LAPPS Grid is a collaborative effort among US partners Brandeis University, Vassar College, Carnegie-Mellon University, and the Linguistic Data Consortium at the University of Pennsylvania, and is funded by the US National Science Foundation (NSF). The project builds on the foundation laid in the NSF-funded project SILT (Ide et al., 2009), which established a set of needs for interoperability and developed standards and best practice guidelines to implement them. LAPPS is similar in its scope and goals to ongoing projects such as The Language Grid[15], PANACEA/MetaNET[16], LinguaGrid[17], and CLARIN[18], which also provide web service access to basic NLP processing tools and resources and enable pipelining these tools to create custom NLP applications and composite services such as question answering and machine translation, as well as access to language resources such as mono- and multilingual corpora and lexicons that support NLP. The transformative aspect of the LAPPS Grid is therefore not the provision of a suite of web services, but rather that it orchestrates access to and deployment of language resources and processing functions available from servers around the globe, and enables users to easily add their own language resources, services, and even service grids to satisfy their particular needs.

[9] http://www.foaf-project.org

[10] http://langrid.nict

[11] http://langrid.portal.cs.ui.ac.id/langrid/

[12] http://langrid.servicegrid-bangkok.org

[13] http://www.linguagrid.org/

[14] http://www.panacea-lr.eu

[15] http://langrid.nict

[16] http://panacea-lr.eu/

[17] http://www.linguagrid.org/

[18] http://www.clarin.eu/

The most distinctive innovation in the LAPPS Grid that is not included in other projects is the provision of an open advancement (OA) framework (Ferrucci et al., 2009a) for component- and application-based evaluation of NLP tools and pipelines. The availability of this type of evaluation service will provide an unprecedented tool for NLP development that could, in itself, take the field to a new level of productivity. OA involves evaluating *multiple possible solutions* to a problem, consisting of different configurations of component tools, resources, and evaluation data, to find the optimal solution among them, and enabling rapid identification of frequent error categories, together with an indication of which module(s) and error type(s) have the greatest impact on overall performance. On this basis, enhancements and/or modifications can be introduced with an eye toward achieving the largest possible reduction in error rate (Ferrucci et al., 2009; Yang et al., 2013). OA was used in the development of IBM's Watson to achieve steady performance gains over the four years of its development (Ferrucci et al., 2010); more recently, the open-source OAQA project has released software frameworks which provide general support for open advancement (Garduno et al., 2013; Yang et al., 2013), which has been used to rapidly develop information retrieval and question answering systems for bioinformatics (Yang et al., 2013; Patel et al., 2013).

The fundamental system architecture of the LAPPS Grid is based on the Open Service Grid Initiative's Service Grid Server Software[19] developed by the National Institute of Information and Communications Technology (NICT) in Japan and used to implement Kyoto University's Language Grid, a service grid that supports multilingual communication and collaboration. Like the Language Grid, the LAPPS Grid provides three main functions: language service registration and deployment, language service search, and language service composition and execution. As noted above, the LAPPS Grid is instrumented to provide relevant component-level measures for standard metrics, given gold-standard test data; new applications automatically include instrumentation for component-level and end-to-end measurement, and intermediate (component-level) I/O is logged to support effective error analysis.[20] The LAPPS Grid also implements a dynamic licensing system for handling license agreements on the fly[21], provides the option to run services locally with high-security technology to protect sensitive information where required, and enables access to grids other than those based on the Service Grid technology.

We have adopted the JSON-based serialization for Linked Data (JSON-LD) to represent linguistically annotated data for the purposes of web service exchange. The JavaScript Object Notation (JSON) 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. Because it is based on the W3C Resource Definition Framework (RDF), JSON-LD is trivially mappable to and from other graph-based formats such as ISO LAF/GrAF and UIMA CAS, as well as a growing number of formats implementing the same data model. Most importantly, JSON- LD enables services to reference categories and definitions in web-based repositories and ontologies or any suitably defined concept at a given URI.

The LAPPS Grid currently supports SOAP services, with plans to support REST services in the near future. We provide two APIs: `org.lappsgrid.api.DataSource`, which provides data to other services, and `org.lappsgrid.api.WebService`, for tools that annotate, transform, or otherwise manipulate data from a datasource or another web service. All LAPPS services exchange `org.lappsgrid.api.Data` objects consisting of a discriminator (type) that indicates how to interpret the payload, and a payload (typically a utf-8 string) that consists of the JSON-LD representation. Data converters included in the LAPPS Grid Service Engines map from commonly used formats to the JSON-LD interchange format; converters are automatically invoked as needed to meet the I/O requirements of pipelined services. Some LAPPS services are pre-wrapped to produce and consume JSON-LD. Thus, JSON-LD provides *syntactic interoperability* among services in the LAPPS Grid; *semantic inter-*

[19] http://servicegrid.net

[20] Our current user interface provides easy (re-)configuration of single pipelines; we are currently extending the interface to allow the user to specify an entire range of pipeline configurations using configuration descriptors (ECD; (Yang et al., 2013) to define a space of possible pipelines, where each step might be achieved by multiple components or services and each component or service may have configuration parameters with more than one possible value to be tested. The system will then automatically generate metrics measurements plus variance and statistical significance calculations for each possible pipeline, using a service-oriented version of the Configuration Space Exploration (CSE) algorithm (Yang et al., 2013).

[21] See (Cieri et al., 2014) for a description of how licensing issues are handled in the LAPPS Grid.

operability is provided by the LAPPS Web Service Exchange Vocabulary, described in the next section.

3 LAPPS Web Service Exchange Vocabulary

3.1 Motivation

The WS-EV addresses a relatively small but critical piece of the overall LAPPS architecture: it allows web services to communicate about the content they deliver, such that the *meaning*–i.e., exactly what to do with and/or how to process the data–is understood by the receiver. As such it performs the same function as a UIMA type system performs for tools in a UIMA pipeline that utilize that type system, or the common annotation labels (e.g., ”Token”, ”Sentence”, etc.) required for communication among pipelined tools in GATE: these mechanisms provide semantic interoperability among tools as long as one remains in either the UIMA or GATE world. To pipeline a tool whose output follows GATE conventions with a tool that expects input that complies with a given UIMA type system, some mapping of terms and structures is likely to be required.[22] This is what the WS-EV is intended to enable; effectively, it is a *meta-type-system* for mapping labels assigned to linguistically annotated data so that they are understood and treated consistently by tools that exchange them in the course of executing a pipeline or workflow. Since web services included in LAPPS and federated grids may use any i/o semantic conventions, the WS-EV allows for communication among any of them–including, for example, between GATE and UIMA services[23]

The ability to pipeline components from diverse sources is critical to the implementation of the OA development approach described in the previous section, it must be possible for the developer to “plug and play” individual tools, modules, and resources in order to rapidly re-configure and evaluate new pipelines. These components may exist on any server across the globe, consist of modules developed within frameworks such as UIMA and GATE, and or be user-defined services existing on a local machine.

3.2 WS-EV Design

The WS-EV was built around the following design principles, which were compiled based on input from the community:

1. The WS-EV will not reinvent the wheel. Objects and features defined in the WS-EV will be linked to definitions in existing repositories and ontologies wherever possible.

2. The WS-EV will be designed so as to allow for easy, one-to-one mapping from terms designating linguistic objects and features commonly produced and consumed by NLP tools that are wrapped as web services. It is not necessary for the mapping to be object-to-object or feature-to-feature.

3. The WS-EV will provide a *core* set of objects and features, on the principle that “simpler is better”, and provide for (principled) definition of additional objects and features beyond the core to represent more specialized tool input and output.

4. The WS-EV is not LAPPS-specific; it will not be governed by the processing requirements or preferences of particular tools, systems, or frameworks.

5. The WS-EV is intended to be used *only* for interchange among web services performing NLP tasks. As such it can serve as a “pivot” format to which user and tool-specific formats can be mapped.

6. The web service provider is responsible for providing wrappers that perform the mapping from internally-used formats to and/or from the WS-EV.

7. The WS-EV format should be compact to facilitate the transfer of large datasets.

[22] Within UIMA, the output of tools conforming to different type systems may themselves require conversion in order to be used together.

[23] Figure 5 shows a pipeline in which both GATE and UIMA services are called; GATE-to-GATE and UIMA-to-UIMA communication does not use the WS-EV, but it is used for communication between GATE and UIMA services, as well as other services.

8. The WS-EV format will be chosen to take advantage, to the extent possible, of existing technological infrastructures and standards.

As noted in the first principle, where possible the objects and features in the WS-EV are drawn from existing repositories such as ISOCat and the NIF Core Ontology and linked to them via the **owl:sameAs** property[24] or, where appropriate, **rdfs:subClassOf**[25]. However, many repositories do not include some categories and objects relevant for web service exchange (e.g., "token" and other segment descriptors), do include multiple (often very similar) definitions for the same concept, and/or do not specify relations among terms. We therefore attempted to identify a set of (more or less) "universal" concepts by surveying existing type systems and schemas – for example, the Julie Lab and DARPA GALE UIMA type systems and the GATE schemas for linguistic phenomena – together with the I/O requirements of commonly used NLP software (e.g., the Stanford NLP tools, OpenNLP, etc.). Results of the survey for token and sentence identification and part-of-speech labeling[26] showed that even for these basic categories, no existing repository provides a suitable set of categories and relations.

Perhaps more problematically, sources that do specify relations among concepts, such as the various UIMA type systems and GATE's schemas, vary widely in their choices of what is an object and what is a feature; for example, some treat "token" as an object (label) and "lemma" and "POStag" as associated features, while others regard "lemma" and/or "POStag" as objects in their own right. Decisions concerning what is an object and what is a feature are for the most part arbitrary; no one scheme is right or wrong, but a consistent organization is required for effective web service interchange. The WS-EV therefore defines an organization of objects and features for the purposes of interchange only. Where possible, the choices are principled, but they are otherwise arbitrary. The WS-EV includes *sameAs* and *similarTo* mappings that link to like concepts in other repositories where possible, thus serving primarily to group the terms and impose a structure of relations required for web service exchange in one web-based location.

In addition to the principles above, the WS-EV is built on the principle of orthogonal design, such that there is one and only one definition for each concept. It is also designed to be very lightweight and easy to find and reference on the web. To that end we have established a straightforward web site (the Web Service Exchange Vocabulary Repository[27]), similar to *schema.org*, in order to provide web-addressable terms and definitions for reference from annotations exchanged among web services. Our approach is bottom-up: we have adopted a minimalist strategy of adding objects and features to the repository only as they are needed as services are added to the LAPPS Grid. Terms are organized in a shallow ontology, with inheritance of properties, as shown in Figure 1.

4 WS-EV and JSON-LD

References in the JSON-LD representation used for interchange among LAPPS Grid web services point to URIs providing definitions for specific linguistic categories in the WS-EV. They also reference documentation for processing software and rules for processes such as tokenization, entity recognition, etc. used to produce a set of annotations, which are often left unspecified in annotated resources (see for example (Fokkens et al., 2013)). While not required for web service exchange in the LAPPS Grid, the inclusion of such references can contribute to the better replication and evaluation of results in the field. Figure 3 shows the information for Token, which defines the concept, identifies application types that produce objects of this type, cross-references a similar concept in ISOCat, and provides the URI for use in the JSON-LD representation. It also specifies the common properties that can be specified for a set of Token objects, and the individual properties that can be associated with a Token object. There is no requirement to use any or all of the properties in the JSON-LD representation, and we foresee that many web services will require definition of objects and properties not included in the WS-EVR or elsewhere.

[24] http://www.w3.org/TR/2004/REC-owl-semantics-20040210/#owl_sameAs

[25] http://www.w3.org/TR/owl-ref/#subClassOf-def

[26] Available at `http://www.anc.org/LAPPS/EP/Meeting-2013-09-26-Pisa/ep-draft.pdf`

[27] http://vocab.lappsgrid.org

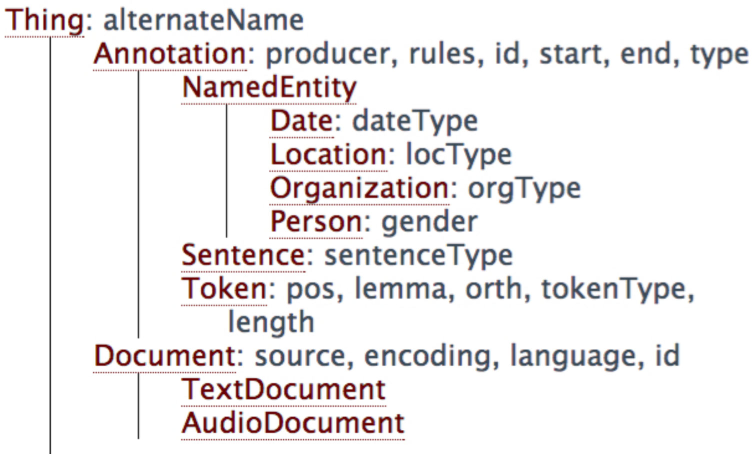


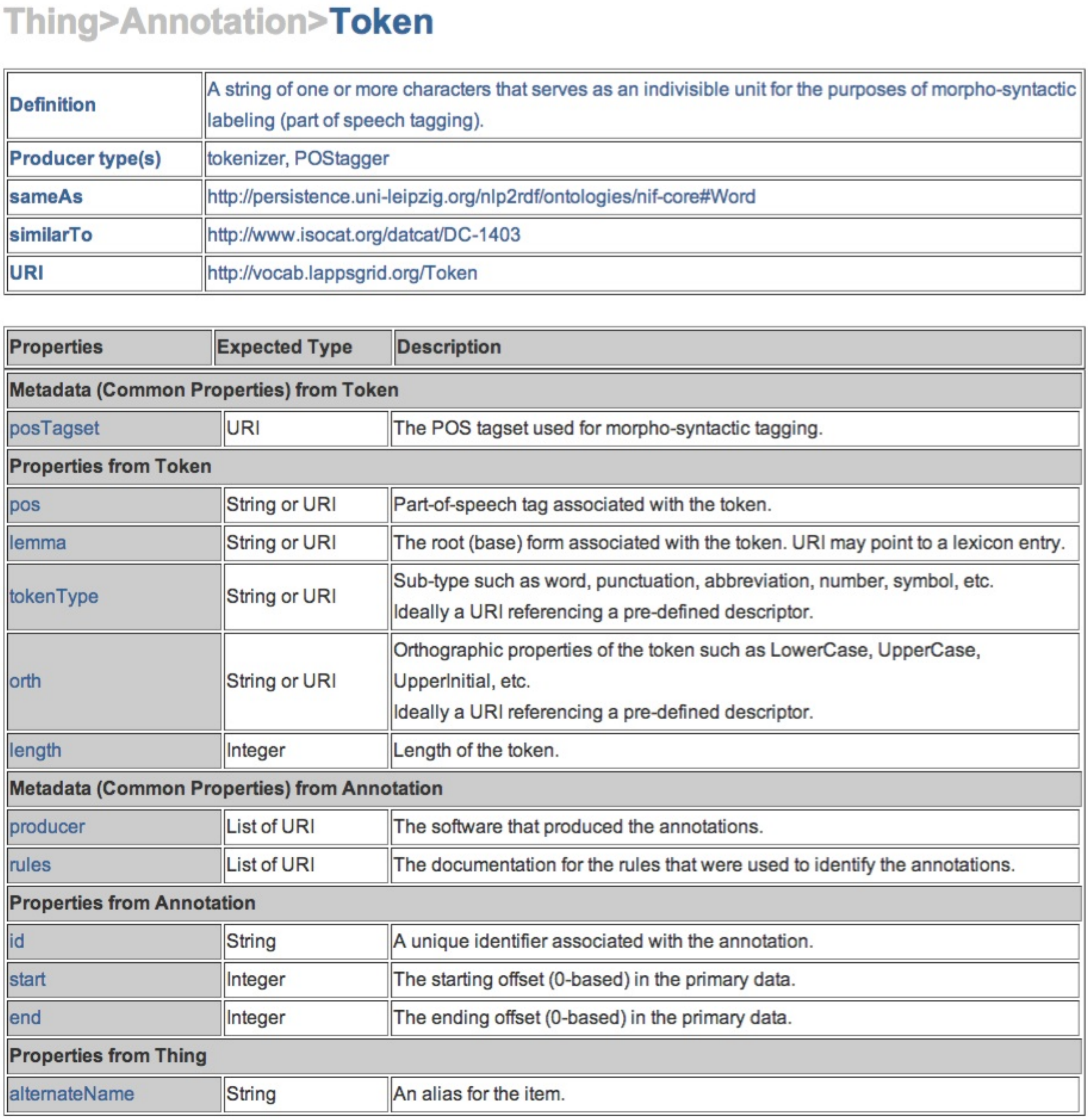
Figure 1: Fragment of the WS-EV ontology (associated properties in gray)

We therefore provide mechanisms for (principled) definition of objects and features beyond the WS-EVR. Two options exist: users can provide a URI where a new term or other documentation is defined, or users may add a definition to the WS-EVR. In the latter case, service providers use the name space automatically assigned to them at the time of registration, thereby avoiding name clashes and providing a distinction between general categories used across services and more idiosyncratic categories.

Figure 2 shows a fragment of the JSON-LD representation that references terms in the WS-EV. The *context* statement at the top identifies the URI that is to be prefixed to any unknown name in order to identify the location of its definition. For the purposes of the example, the text to be processed is given inline. Our current implementation includes results from each step in a pipeline, where applicable, together with metadata describing the service applied in each step (here, org.anc.lapps.stanford.SATokenizer:1.4.0) and identified by an internally-defined type (stanford). The annotations include references to the objects defined in the WS-EV, in this example, Token (defined at http://vocab.lappsgrid.org/Token) with (inherited) features *id, start, end* and specific feature *string*, defined at http://vocab.lappsgrid.org/Token#id, http://vocab.lappsgrid.org/Token#start, http://vocab.lappsgrid.org/Token#end, and http://vocab.lappsgrid.orgToken/#string, respectively. The web page defining these terms is shown in Figure 3.

```
"@context" : "http://vocab.lappsgrid.org/",
"metadata" : { },
"text" : {
    "@value" : "Some of the strongest critics of our welfare system..." }
"steps" : [ {
    "metadata" : {
          "contains" : {
          "Token" : {
          "producer" : "org.anc.lapps.stanford.SATokenizer:1.4.0",
          "type" : "stanford"
        }
      }
    },
    "annotations" : [ {
      "@type" : "Token",
      "id" : "tok0",
      "start" : 18,
      "end" : 22,
      "features" : {
        string" : "Some" }
    },
```

Figure 2: JSON-LD fragment referencing the LAPPS Grid WS-EV



Thing>Annotation>Token

Definition	A string of one or more characters that serves as an indivisible unit for the purposes of morpho-syntactic labeling (part of speech tagging).
Producer type(s)	tokenizer, POStagger
sameAs	http://persistence.uni-leipzig.org/nlp2rdf/ontologies/nif-core#Word
similarTo	http://www.isocat.org/datcat/DC-1403
URI	http://vocab.lappsgrid.org/Token

Properties	Expected Type	Description
Metadata (Common Properties) from Token		
posTagset	URI	The POS tagset used for morpho-syntactic tagging.
Properties from Token		
pos	String or URI	Part-of-speech tag associated with the token.
lemma	String or URI	The root (base) form associated with the token. URI may point to a lexicon entry.
tokenType	String or URI	Sub-type such as word, punctuation, abbreviation, number, symbol, etc. Ideally a URI referencing a pre-defined descriptor.
orth	String or URI	Orthographic properties of the token such as LowerCase, UpperCase, UpperInitial, etc. Ideally a URI referencing a pre-defined descriptor.
length	Integer	Length of the token.
Metadata (Common Properties) from Annotation		
producer	List of URI	The software that produced the annotations.
rules	List of URI	The documentation for the rules that were used to identify the annotations.
Properties from Annotation		
id	String	A unique identifier associated with the annotation.
start	Integer	The starting offset (0-based) in the primary data.
end	Integer	The ending offset (0-based) in the primary data.
Properties from Thing		
alternateName	String	An alias for the item.

Figure 3: Token definition in the LAPPS WS-EVR

4.1 Mapping to JSON-LD

As noted above in Section 1, existing schemes and systems for organizing linguistic information exchanged by NLP tools vary considerably. Figure 4 shows some variants for a few commonly used NLP tools, which differ in terminology, structure, and physical format. To be used in the LAPPS Grid, tools such as those in the list are wrapped so that their output is in JSON-LD format, which provides syntactic interoperability, terms are mapped to corresponding objects in the WS-EV, and the object-feature relations reflect those defined in the WS-EV. Correspondingly, wrappers transduce the JSON-LD/WS-EV representation to the format used internally by the tool on input. This way, the tools use their internal format as usual and map to JSON-LD/WS-EV for exchange only.

Name	Input	Form	Output	Form	Example
Stanford tagger	pt	n/a	word_pos	opl	box_NN1
	XML	n/a	XML	inline	<word id=”0” pos=”VB”>Let</word>
NaCTeM tagger	pt	n/a	word/pos	inline	box/NN1
CLAWS (1)	pt	n/a	word_pos	inline	box_NN1
CLAWS (2)	pt	n/a	XML	inline	<w id=”2” pos=”NN1”>Type</w>
CST Copenhagen	pt	n/a	word/pos	inline	box/NN1
TreeTagger	pt?	n/a	word pos lem	opl	The DT the
TnT	token	opl	word pos	opl	der ART
			word (pos pr)+	opl	Falkenstein NE 8.00 NN 1.99
Twitter NLP	pt	opl	word pos conf	opl	smh G 0.9406
NLTK	pt	s, bls	[(’word’, ’pos’)]	inline	[(’At’, ’IN’), (’eight’, ’CD’),]
OpenNLP splitter	pt	n/a	sentences	ospl	I can’t tell you if he’s here.
OpenNLP tokenizer	sent	ospl	tokens	wss, ospl	I can ’t tell you if he ’s here .
OpenNLP tagger	token	wss, ospl	word_pos	ospl	At_IN eight_CD o’clock_JJ on_IN

pt = plain text	opl = one per line	wss = white space separated
	ospl = one sentence per line	bps = blank line separated

Figure 4: I/O variants for common splitters, tokenizers, and POS taggers

For example, the Stanford POS tagger XML output format produces output like this:

```
<word id="0" pos="VB">Let</word>
```

This maps to the following JSON-LD/WS-EV representation:

```
{
   "@type" : "Token",
   "id" : 0",
   "start" : 18,
   "end" : 21,
   "features" : {
     "string" : "Let",
     "pos"    : "VB"
   }
}
```

The Stanford representation uses the term “word” as an XML element name, gives an id and pos as attribute-value pairs, and includes the string being annotated as element content. For conversion to JSON-LD/WS-EV, “word” is mapped to “Token”, the attributes *id* and *pos* map to features of the Token object with the same names, and the element content becomes the value of the *string* feature. Because the JSON-LD representation uses standoff annotation, the attributes *start* and *end* are added in order to provide the offset location of the string in the original data.

Services that share a format other than JSON-LD need not map into and out of JSON-LD/WS-EV when pipelined in the LAPPS Grid. For example, two GATE services would exchange GATE XML documents, and two UIMA services would exchange UIMA CAS, as usual. This avoids unnecessary conversion and at the same time allows including services (consisting of individual tools or composite workflows) from other frameworks. Figure 5 gives an example of the logical flow in the LAPPS Grid, showing conversions into and out of JSON-LD/WS-EV where needed.

Each service in the LAPPS Grid is required to provide metadata that specifies what kind of input is required and what kind of output is produced. For example, any service as depicted in the flow diagram in Figure 5 can require input of a particular format (gate, uima, json-ld) with specific content (tokens, sentences, etc.). The LAPPS Grid uses the notion of *discriminators* to encode these requirements, and the pipeline composer can use these discriminators to determine if conversions are needed and/or input requirements are met. The discriminators refer to elements of the vocabulary.

5 Collaborations

The LAPPS Grid project is collaborating with several other projects in an attempt to harmonize the development of web service platforms, and ultimately to participate in a federation of grids and service platforms throughout the world. Existing and potential projects across the globe are beginning to

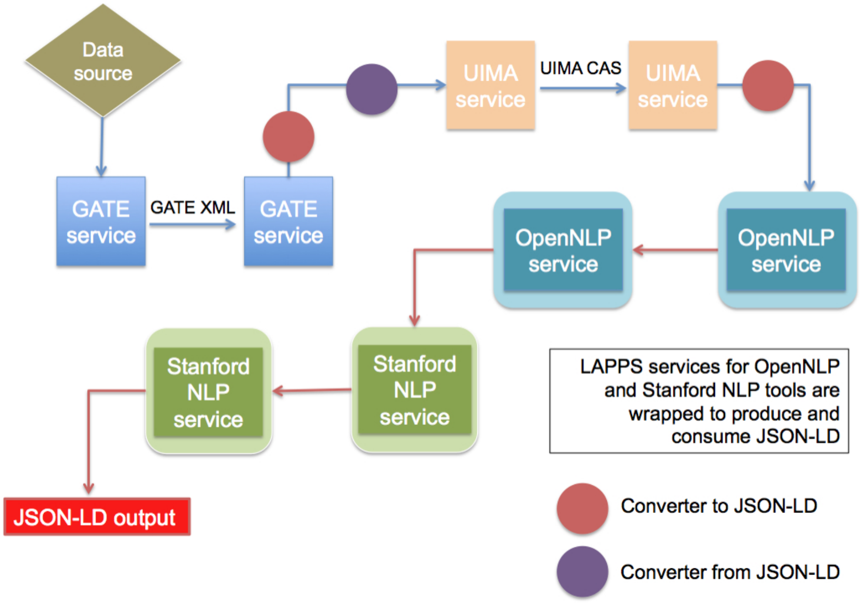


Figure 5: Logical flow through the LAPPS Grid (client-server communication not represented)

converge on common data models, best practices, and standards, and the vision of a comprehensive infrastructure supporting discovery and deployment of web services that deliver language resources and processing components is an increasingly achievable goal. Our vision is therefore not for a monolithic grid, but rather a heterogeneous configuration of federated grids that implement common strategies for managing and inter-changing linguistic information, so that services on all of these grids are mutually accessible.

To this end, the LAPPS Grid project has established a multi-way international collaboration among the US partners and institutions in Asia, Australia, and Europe. The basis is a formal federation among the LAPPS Grid, the Language Grid (Kyoto University, Japan), NECTEC (Thailand), grids operated by the University of Indonesia and Xinjiang University (China), and LinguaGrid[28], scheduled for implementation in January 2015. The connection of these six grids into a single federated entity will enable access to all services and resources on any of these grids by users of any one of them and, perhaps most importantly, facilitate adding additional grids and service platforms to the federation. Currently, the European META-NET initiative is committed to joining the federation in the near future.

In addition to the projects listed above, we are also collaborating with several groups on technical solutions to achieve interoperability and in particular, on development of the WS-EV, the JSON-LD format, and a corollary development of an ontology of web service types. These collaborators include the Alveo Project (Macquarie University, Australia) (Cassidy et al., 2014), the Language Grid project, and the Lider project[29]. We actively seek collaboration with others in order to move closer to achieving a “global laboratory” for language applications.

6 Conclusion

In this paper, we have given a brief overview of the LAPPS Web Service Exchange Vocabulary (WS-EV), which provides a terminology for a core of linguistic objects and features exchanged among NLP tools that consume and produce linguistically annotated data. The goal is to bring the field closer to achieving semantic interoperability among NLP data, tools, and services. We are actively working to both engage with existing projects and teams and leverage available resources to move toward convergence of terminology in the field for the purposes of exchange, as well as promote an environment (the LAPPS Grid) within which the WS-EV can help achieve these goals.

[28] http://www.linguagrid.org/

[29] http://www.lider-project.eu

Acknowledgements

This work was supported by National Science Foundation grants NSF-ACI 1147944 and NSF-ACI 1147912.

References

Steve Cassidy, Dominique Estival, Timothy Jones, Denis Burnham, and Jared Burghold. 2014. The Alveo Virtual Laboratory: A Web based Repository API. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

Christopher Cieri, Denise DiPersio, , and Jonathan Wright. 2014. Intellectual property rights management with web services. In *Proceedings of the Workshop on Open Infrastructures and Analysis Frameworks for HLT*, Dublin, Ireland, August.

David Ferrucci, Eric Nyberg, James Allan, Ken Barker, Eric Brown, Jennifer Chu-Carroll, Arthur Ciccolo, Pablo Duboue, James Fan, David Gondek, Eduard Hovy, Boris Katz, Adam Lally, Michael McCord, Paul Morarescu, Bill Murdock, Bruce Porter, John Prager, Tomek Strzalkowski, Chris Welty, and Wlodek Zadrozny. 2009. Towards the Open Advancement of Question Answering Systems. Technical report, IBM Research, Armonk, New York.

David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. 2010. Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79.

Antske Fokkens, Marieke van Erp, Marten Postma, Ted Pedersen, Piek Vossen, and Nuno Freire. 2013. Offspring from reproduction problems: What replication failure teaches us. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1691–1701, Sofia, Bulgaria, August. Association for Computational Linguistics.

Elmer Garduno, Zi Yang, Avner Maiberg, Collin McCormack, Yan Fang, and Eric Nyberg. 2013. CSE Framework: A UIMA-based Distributed System for Configuration Space Exploration Unstructured Information Management Architecture. In Peter Klgl, Richard Eckart de Castilho, and Katrin Tomanek, editors, *UIMA@GSCL*, CEUR Workshop Proceedings, pages 14–17. CEUR-WS.org.

Sebastian Hellmann, Jens Lehmann, Sören Auer, and Martin Brümmer. 2013. Integrating nlp using linked data. In *12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia*.

Nancy Ide and James Pustejovsky. 2010. What Does Interoperability Mean, Anyway? Toward an Operational Definition of Interoperability. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources*. ICGL.

Nancy Ide and Keith Suderman. 2014. The Linguistic Annotation Framework: A Standard for Annotation Interchange and Merging. *Language Resources and Evaluation*.

Nancy Ide, James Pustejovsky, Nicoletta Calzolari, and Claudia Soria. 2009. The SILT and FlaReNet international collaboration for interoperability. In *Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP*, August.

Nancy Ide, James Pustejovsky, Christopher Cieri, Eric Nyberg, Di Wang, Keith Suderman, Marc Verhagen, and Jonathan Wright. 2014. The language application grid. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland, may. European Language Resources Association (ELRA).

ISO-24612. 2012. Language Resource Management - Linguistic Annotation Framework. ISO 24612.

Alkesh Patel, Zi Yang, Eric Nyberg, and Teruko Mitamura. 2013. Building an optimal QA system automatically using configuration space exploration for QA4MRE'13 tasks. In *Proceedings of CLEF 2013*.

Zi Yang, Elmer Garduno, Yan Fang, Avner Maiberg, Collin McCormack, and Eric Nyberg. 2013. Building optimal information systems automatically: Configuration space exploration for biomedical information systems. In *Proceedings of the CIKM'13*.