The Language Application Grid

Nancy Ide*, James Pustejovsky†, Christopher Cieri‡, Eric Nyberg, Denise DiPersio‡, Chunqi Shi†,**

Keith Suderman*, Marc Verhagen†, Di Wang, Jonathan Wright‡**

*Vassar College, Poughkeepsie, New York USA, {ide,suderman}@cs.vassar.edu
†Brandeis University, Waltham, Massachusetts USA, {jamesp,shicq, marc}@cs.brandeis.edu
‡Linguistic Data Consortium, Philadelphia, Pennsylvania USA, {ccieri, dipersio, jdwright}@ldc.upenn.edu
**Carnegie-Mellon University, Pittsburgh, Pennsylvania USA, {ehn, diwang}@cs.cmu.edu

Abstract

The Language Application (LAPPS) Grid project is establishing a framework that enables language service discovery, composition, and reuse and promotes 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 access to basic NLP processing tools and resources and enables pipelining such tools to create custom NLP applications, as well as composite services such as question answering and machine translation together with language resources such as mono- and multi-lingual corpora and lexicons that support NLP. The transformative aspect of the LAPPS Grid is that 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, services, and even service grids to satisfy their particular needs.

Keywords: NLP frameworks, web services, service grids, open advancement, resource licensing

1. Introduction

The Language Application (LAPPS) Grid project is establishing a framework that enables language service discovery, composition, and reuse and promotes 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. The project builds on the foundation laid in projects such as SILT (Ide et al., 2009), The Language Grid[1], PANACEA[2], LinguaGrid[3] and CLARIN[4], as well as the momentum toward a comprehensive network of web services and resources within the NLP community. The goals of the project are to: (1) design, develop, and promote a *Language Application Grid* (LAPPS Grid) based on Service Grid Software[5] to support the development and deployment of integrated natural language applications and enable federation of grids and services throughout the world; (2) provide an *open advancement (OA) framework* (Ferrucci et al., 2009a) for component- and application-based evaluation; (3) provide access to language resources for members of the NLP community as well as researchers in a wide range of social science and humanities disciplines, (4) enable easy navigation through licensing issues; and (5) actively promote adoption, use, and community involvement with the LAPPS Grid.

The LAPPS Grid provides access to basic NLP processing tools and resources and enables 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 multi-lingual corpora and lexicons that support NLP. However, the transformative aspect of the LAPPS Grid is 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 add their own language resources, services, and even service grids to satisfy their particular needs. As such, the LAPPS Grid is ultimately a community-based project, to which services will be contributed by members of the community and existing service repositories and grids can be federated to enable universal access.

In this paper we provide an overview of the LAPPS Grid and the technologies we are developing to support its use. Section 2 describes the overall architecture of the LAPPS Grid. In Section 3, the development of the LAPPS Web Service Exchange Vocabulary, which enables interoperability among services in the Grid, is described. Section 4 introduces the Composer interface for accessing and constructing atomic and composite web services, and in Section 5 we overview the open advancement evaluation capabilities

[1] http://langrid.nict.go.jp.
[2] http://panacea-lr.eu/.
[3] http://www.linguagrid.org/.
[4] http://www.clarin.eu/.
[5] http://servicegrid.net.

that are being provided in the Grid. Section 6 discusses our approach to handling potentially divergent licensing constraints in web service pipelines. Finally, Sections 7 and 8 discuss user-provided evaluation of the LAPPS Grid and the relation of this project to similar projects in Asia, Australia, and the European Union.

2. LAPPS Grid Design

The fundamental system architecture of the LAPPS Grid is based on the Open Service Grid Initiative's Service Grid Server Software 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. From the perspective of application developers, one of the intended audiences for the LAPPS Grid, several aspects of service deployment are important:

1. *Service Discovery*. An application designer can query for existing components and services that provide some desired functionality, and quickly identify elements in the repository that are suited to the task.

2. *Service Adaptation*. The LAPPS Grid supports straightforward customization and adaptation of each component or service (e.g., by exposing parameters, options, etc.).

3. *Service Composition*. New applications can be built from existing elements and tested on client data with a minimum amount of programming.

4. *Metrics and Measurement*. 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; intermediate (component-level) I/O is logged to support effective error analysis.

To support these four aspects, the LAPPS Grid extends the core functionality of the Service Grid Software by further enabling composition of tool and resource chains as well as by providing sophisticated evaluation services. In addition, the LAPPS Grid implements a *dynamic licensing* system for handling license agreements on the fly; provide the option to run services locally, with high-security technology to protect sensitive information where required; improve data delivery services; and enable access to grids other than those based on the Service Grid technology. Also, because the LAPPS Grid is a community-based resource to which members of the community will increasingly contribute as well as use, we provide user-friendly, transparent facilities for wrapping user-provided services.

Some of these extensions are available in the current LAPPS Grid as prototypes, most notably (1) modules for composing services in a straightforward way, (2) an exchange vocabulary for facilitating input/output interchange and reuse of components, (3) more user-friendly ways to wrap and invoke services, (4) an online service composer, (5) conversion modules to increase interoperability, (6) initial modules for evaluation services, (7) data services that interface to ANC data at Vassar and various data at the Linguistics Data Consortium, and (8) extended licensing schema.

The basic components of the LAPPS Grid are presented in Figure 1. The main LAPPS server maintains a workflow repository for composite linguistic services and is equipped with a workflow engine to enable users to develop their own composite (pipelined) services. It also contains various modules for discovery, wrapping and conversion. LAPPS Grid nodes housed at Brandeis University and Vassar College maintain repositories of known atomic linguistic services and provides service discovery functionality to users and applications. The LDC node houses various data services and the node at CMU provides services for automatic instrumentation and measurement of LAPP performance, error analysis at the component and end-to-end application level, as well as a service for running LAPPS pipelines augmented with measurement and analysis components.

Each web service in the LAPPS Grid publishes metadata describing what it requires for input and what it produces as output. Any service registered in the LAPPS Grid must provide this information. A process that is constructing a service pipeline can then query each service to determine compatibility.

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)[6] 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 (Ide and Suderman, 2014; ISO-24612, 2012) and UIMA CAS[7], as well as a growing number of formats implementing the same data model. JSON-LD enables services to reference categories and definitions in web-based repositories and ontologies (e.g., ISOCat[8], GOLD[9], Dublin Core[10], OLiA[11]) or any suitably defined concept at a given URI.

The data converters included in the Language Application Service Engines (see Figure 1) 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.

3. Exchange vocabulary

Although the pipeline architecture has been implemented in several NLP frameworks over the past two decades, includ-

[6] http://www.json.org and http://www.ietf.org/rfc/rfc4627.txt.

[7] The *Common Analysis Structure* (CAS) is the internal format for exchange among modules in the UIMA framework.

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

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

[10] http://dublincore.org

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

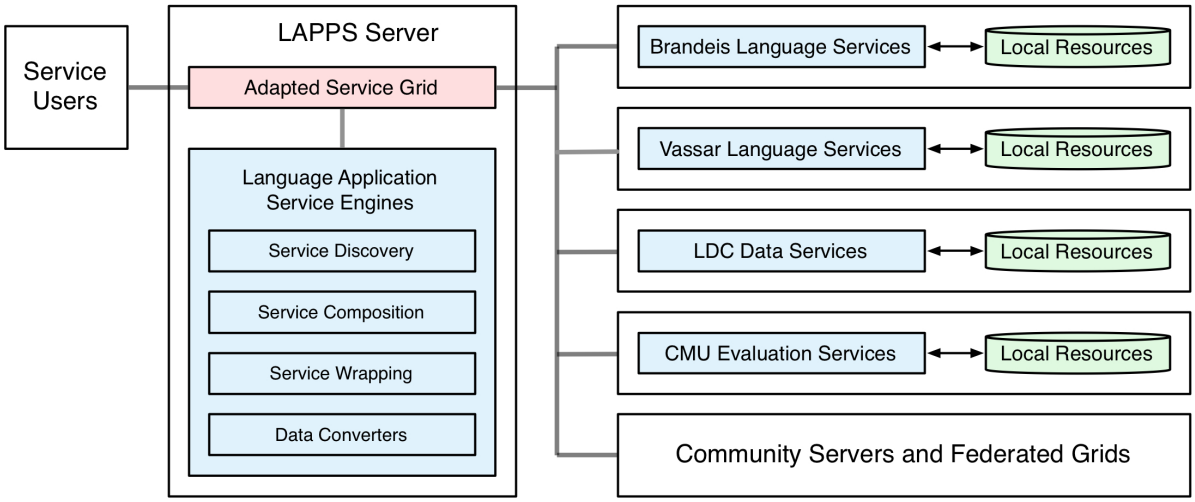

Figure 1: LAPPS Grid Architecture

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	
almostSameAs	http://www.isocat.org/datcat/DC-1403
URI	http://vocab.lappsgrid.org/Token

Properties	Expected Type	Description
Common Properties		
Producer	URI	The software that produced the annotation.
Rules	URI	The documentation for the rules that were used to identify the annotated items.
POSTagset	URI	The POS tagset used for morpho-syntactic tagging.
Individual Properties		
id	String	A unique identifier associated with an annotation.
start	Integer	The starting offset (0-based) in the primary data.
end	Integer	The ending offset (0-based) in the primary data.
posTag	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.
type	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.

Figure 2: Token definition in the LAPPS WS-EVR

ing self-contained (non-service) frameworks such as GATE and UIMA, no accepted standard for module description or input/output interchange to support service discovery, composition, and reuse in the language application domain exists. To address this, we have defined a Web Service Exchange Vocabulary (WS-EV) that specifies a terminology for a core of linguistic objects and features exchanged among NLP tools that consume and produce linguistically annotated data. As such, it addresses a need within the community to not only identify a standard terminology, but also indicate the relations among them.

Because of the well known difficulties of devising such standards, our approach is "bottom-up", avoiding *a priori* development of a comprehensive standard linguistic type system. To that end, we have adopted a "minimalist" strategy of providing a simple core set of objects and features. Where possible, the core is drawn from existing repositories such as ISOCat; however, because many categories and objects relevant for web service exchange are not included in such repositories, we have 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.).[12]

We have established a Web Service Exchange Vocabu-

[12]The survey of basic linguistic objects was undertaken within a Working Group of ISO TC37 SC4. A working draft and an inventory of type systems are available at http://vocab.lappsgrid.org/EV/ev-draft.pdf and http://vocab.lappsgrid.org/EV/materials/.

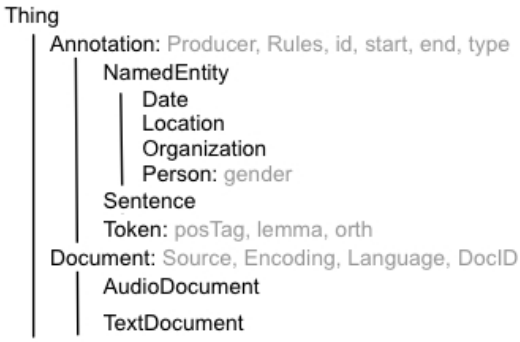

Figure 3: Fragment of the WS-EVR ontology (associated properties in gray)

lary Repository (LAPPS WS-EVR)[13] for NLP web services, similar to `schema.org`, in order to provide web-addressable terms and definitions for reference from annotations exchanged among web services. Wherever possible, terms in the vocabulary are mapped to categories defined in other repositories, ontologies, registries, etc. (including mapping to multiple repositories when appropriate). For this purpose we utilize the taxonomy of relation types defined in RELcat (Windhouwer, 2012), which accommodates multiple vocabularies for relation predicates including those from the Web Ontology Language (OWL) (W3C OWL Working Group, 2012) and the Simple Knowledge Organization System (SKOS) (W3C SKOS Working Group, 2009). Terms are organized in a shallow ontology, with inheritance of properties, as shown in Figure 3. WS-EVR development is undertaken in collaboration with a Working Group within ISO TC37 SC4, to guarantee substantial community involvement and so that our results may ultimately become a part of the larger set of ISO standards for language resource management.

References in the JSON-LD representation point not only to definitions for specific linguistic categories, but also to 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 (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 2 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. 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.

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

4. LAPPS Web Composer

The LAPPS Composer provides a web user interface (see Figure 4) that currently supports the following: (1) registration of tools and resources, making them accessible through the LAPPS Grid; (2) browsing available resources and services ; and (3) searching available atomic web services to identify components of interest. The Composer also helps LAPPS users to rapidly compose a service workflow, run experiments, and display results. Composing service workflow is simply a matter of dragging service names from the available service tab and dropping them into the selected services tab in the interface. The user can either select a data source service or upload text in ad-hoc document field as the input source. An experiment can be initiated then by applying the composed service workflow on the selected input source. When a component requires a type of annotation that not provided by any previous steps, the Composer will pop out a notification to the user with what is missing. If a tool does not directly produce its results or consume its input in the form of JSON-LD, the encapsulating service provides a mapping to and from the input and output JSON-LD realizations that can be used internally by the tool.

After an experiment has been successfully finished, both the final result and every intermediate step output are presented in different tabs inside the interface. The Composer also integrates with an evaluation component and the user can configure which steps generate the gold and predicted annotations. It outputs evaluation results with metrics such as precision, recall, and f measures, and also shows highlighted tables comparing two sets of selected annotations. All intermediate and final results are persisted into a centralized database in JSON format for comparison of multiple experiments later, which also make it possible to connect sophisticated evaluation services (see Section 5.) that enable the user to rapidly assess the quality of each components' contribution to the overall results and experiment with substitute components to achieve the best possible performance.

The current prototype is implemented using Java Server Faces with a backend that checks for compatibility of inputs and outputs between services (see Section 2.).

5. Open Advancement

CMU is providing the tooling and infrastructure for two major services, based in part on the existing OAQA framework developed at CMU and deployed on a service node housed at CMU. The availability of this type of evaluation service, which implements state-of-the-art Open Advancement techniques, will provide an unprecedented tool for NLP development that could, in itself, take the field to a new level of productivity. The open advancement (OA)

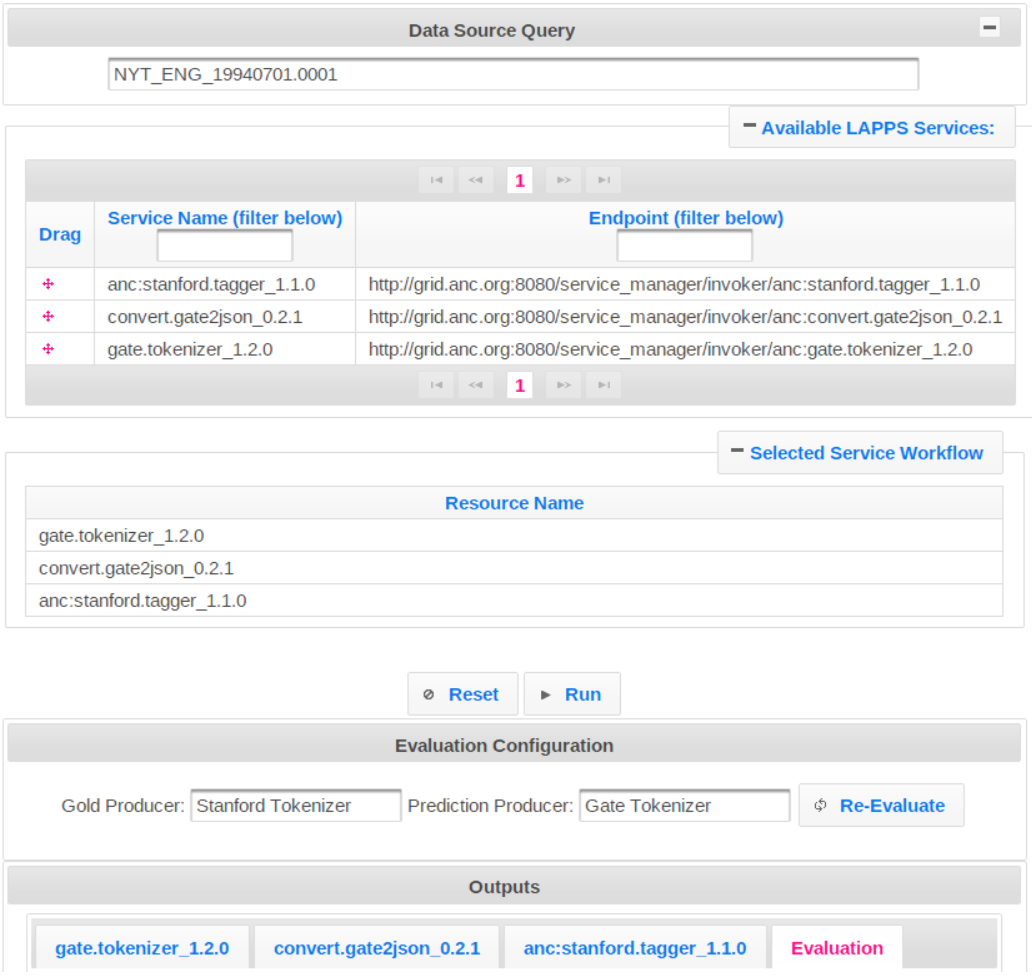

Figure 4: A Screenshot of the LAPPS Composer Interface

approach for component- and application-based evaluation has been successful in enabling rapid identification of frequent error categories within modules and documents, together with an indication of which module(s) and error type(s) have the greatest impact on overall performance, thus contributing to more effective investment of resources in both research and application assembly (Ferrucci et al., 2009b; Yang et al., 2013). The OA approach 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 of information systems (Garduno et al., 2013; Yang et al., 2013); the OAQA software has been used to rapidly develop information retrieval and question answering systems for bioinformatics (Yang et al., 2013; Patel et al., 2013).

A fundamental element of open advancement involves evaluating multiple possible solutions to a given problem, to find the optimal solution available using given components, resources and evaluation data. The output of the optimal solution is then subjected to error analysis, to identify the most frequent errors with the highest impact on system output quality. Possible enhancements to the system are then considered, with an eye toward achieving the largest possible reduction in error rate by addressing the most frequent error types. The performance of each new configuration is evaluated to determine whether a significant improvement has been achieved in comparison with prior baselines or best known configurations. When multiple teams collaborate to implement this process across several sites, types of components, etc. it is possible to make rapid progress in improving solution quality, as measured by the chosen metrics and evaluation dataset (**?**; Ferrucci et al., 2009b). To support rapid, open advancement, it should be possible for a developer to add new components to the system and test them in the context of existing pipelines by "plugging them in" to existing solutions.

The Composer module described in the previous section provides easy (re-)configuration of pipelines, and represents our first step in supporting open advancement by allowing users to rapidly configure and evaluate a new, single pipeline on a chosen dataset and metrics. Ideally, it should be possible for the user to specify an entire range of pipeline configurations for comparative evaluation; the system will then evaluate each possible pipeline configuration and generate metrics measurements, plus variance and statistical significance calculations. To achieve this goal, we are working to extend the Composer to allow easy specification of configuration descriptors (ECD; (Yang et al., 2013) that define a space of possible pipelines, where each step in the pipeline might be achieved by multiple compo-

nents or services; each component or service may also have configuration parameters with more than one possible value to be tested. We also plan to extend the system to support automatic evaluation of each configuration so specified, by implementing a service-oriented version of the Configuration Space Exploration (CSE) algorithm (Yang et al., 2013).

6. Resource Access

LDC's contributions to the multi-site LAPPS Grid focus naturally on data. LDC is creating services that provide grid access to the contents of its LDC Online service: multilingual newswire and transcribed conversational telephone speech in English, as well as to lexical databases. The challenges of this work lie in developing useful and efficient service interfaces to these data. In each case, we envision the interface as containing a number of simple operations: requests to retrieve the features of the supplied data, queries into the data using those features that return identifiers and requests to fetch data elements by identifier, via iteration or randomly. LDC already deploys data services, both internal and external, so our Grid work emphasizes enclosing those services in a thin wrapper within a Grid node that we host. Using the data source API developed by the LAPPS project, we pass on Grid requests to LDC services. Some LDC services, including the Grid node, run on virtual machines, allowing us to easily adjust system resources to match changing demand. LDC's infrastructure also includes a Solr[14] server for searching text, including some of the content available to the Grid.

Along with the flexibility the LAPPS Grid offers to users seeking to create service pipelines comes an increase in the complexity of intellectual property arrangements. We anticipate two major pipeline types. In the first, users request language resources from a given source (or supply their own) and route them through a workflow of multiple grid services with the final result returned to the user. In the second type, language resources are routed through a single service and then back to the user before being routed along to the next service. The difference between these user case types has implications for licensing and constraints imposed on grid users, services and operators. Moreover, within those cases, one must consider constraints imposed by the language resources, data and software enabling the web services.

At each point in either pipeline above, constraints depend upon the language resources or resulting services, processing and user. Resources may be constrained or unconstrained. Constraints may be imposed by legal principles such as copyright or by contract. Constraints may prohibit commercial use, derivative works or re-distribution or insist upon attribution or in-kind sharing of the user's intellectual products. Resources may be constrained as to user, typically forbidding use by commercial organizations, or as to use, whether for education, basic research, applied research, technology development, evaluation and deployment or resale. Processing may also be constrained, for example, ruling out derivative works and only permitting so-called transformative works. Users may be licensed or not. Their licensing may be defined by enumeration or by user features, for example whether they work in an academic, non-academic, not-for-profit, government, pre-commercial or commercial environments.

We manage this complexity by identifying the licenses associated with each Grid service and analyzing them into their component constraints. Then at each stage of the workflow we check for compatibility among the constraints imposed so far and the uses planned. Where these are compatible, flow continues; otherwise flow is blocked. Providers of resources, the first step in many pipelines only impose constraints on their output. At each intermediate step, another grid service may request rights of its input but more commonly imposes constraints on its output. The final step, delivery to user, only requests rights to use. In our model, the constraints imposed during the pipeline are cumulative and the user must satisfy the superset. In this way, the work of determining whether a specific grid workflow is allowable amounts to traversing that workflow to identify, at each stage, what rights are requested and what constraints are imposed, confirming that the constraints are satisfied and, if so, accumulating any additional constraints imposed at that stage. A failure to satisfy the constraints at any stage indicates that the workflow is forbidden. Thus the legal status of the pipeline is controlled by two opposing forces, the imposition of constraints and the impetus for rights to use. Figure 5 provides some examples of possible licensing pipelines.

Variation in license terms notwithstanding, the human language technology community has for some time envisioned open source-based models for language resource development and distribution. Most recently, META-SHARE proposes a network of distributed repositories that license resources from a single platform via open source agreements (META-SHARE Commons licenses) as well as more restrictive arrangements (Piperdis, 2012). Although all levels of licensing complexity are acknowledged in the LAPPS Grid, the LAPPS license scheme depends on the utilization of open source software and resource licenses to the greatest extent possible. By limiting distribution and processing constraints, we aim to promote the project goal of community engagement through sharing, federation and other means. By developing a comprehensive model for addressing constraints on the intellectual property used in the Grid we hope to create a resource that is maximally open to users ranging from open source developers to commercial users of languages services.

7. User evaluation

To a large extent, the measure of success for LAPPS is a matter of the ease with which the user community–both NLP researchers and developers and those with little knowledge of the field–can use the infrastructure to serve their needs. The project therefore includes an on-going user-evaluation component involving a range of user types, including those whose computational expertise may be limited, who provide periodic feedback concerning Grid access, adding applications to the Grid, using external applications or services in combination with the Grid, etc. In the spirit of open advancement, we measure the total time

[14] https://lucene.apache.org/solr/

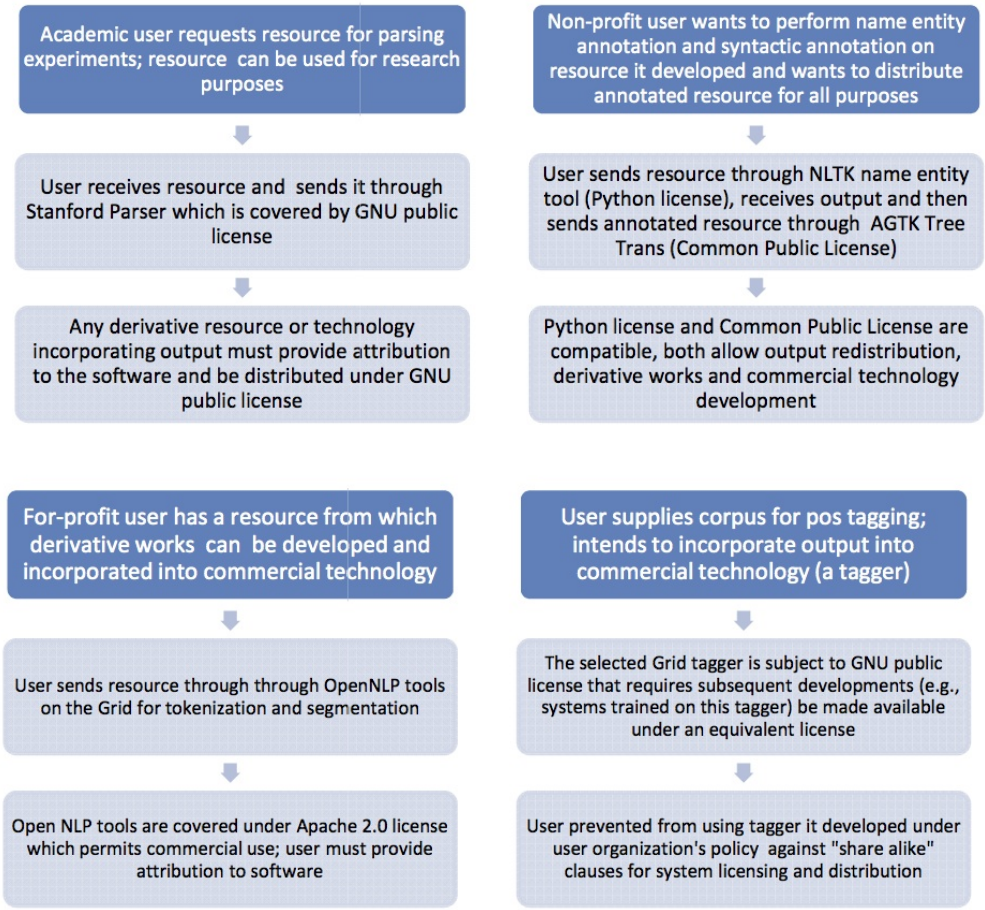

Figure 5: Four licensing pipeline scenarios

and effort required to determine the optimal configuration of existing components for a given problem and use these measures to improve the system's design.

To support community use, we regularly offer tutorials and training workshops on LAPPS Grid use at major conferences in the field[15], including venues associated with other disciplines, with the goal of introducing scientists and engineers from diverse disciplines to a broad-based and integrated set of NLP services that has the potential to impact their research and development needs. We envision that research from sociology, psychology, economics, education, linguistics, digital media, as well as engineering, can be impacted by the ability to manipulate and process diverse data sources in multiple languages.

8. Relation to other projects

The LAPPS Grid effort builds on the foundation laid in several recent U.S., European, and Asian projects, including the NSF-funded Sustainable Interoperability for Language Technology (SILT) project (Ide et al., 2009) and the EU-funded Fostering Language Resources Network (FLaReNet) project (Calzolari et al., 2009). At the same time, the International Standards Organization (ISO) committee for Language Resource Management (ISO TC37 SC4)[16] has addressed the need for standards for linguistic data. Through these and other projects and parallel efforts in Asia and Australia, substantial groundwork—in terms of standards development, raising community awareness and buy-in, and proof-of-concept implementation—has been laid to turn existing, fragmented NLP technologies and data into web-accessible, stable, and interoperable resources that can be readily reused across several fields. As a result, existing and potential projects across the globe are beginning to 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 a set of best practices for managing and interchanging linguistic information, so that services on all of these grids are mutually accessible. To that end, the LAPPS Grid project has established a multi-way international collaboration among the US partners and institu-

[15] E.g., *Web Services for Effective NLP Application Development and Evaluation: Using and Contributing to the Language Application (LAPPS) Grid*, offered at LREC 2014.

[16] ISO/TC 37/SC4, Language Resources Management, http://www.tc37sc4.org.

tions in Asia (The National Institute of Information and Communications Technology (NICT), and Kyoto University, Japan), Australia (Macquarie University, Sydney), and Europe (Universitat Pompeu Fabra, Barcelona; Istituto di Linguistica Computazionale-Consiglio Nazionale Ricerche (CNR), Pisa; University of Trento; University of Torino; and CELI Research, Rome). This collaboration brings together several relevant individuals and projects involved with language resource development and distribution, including individual researchers, resource developers, major resource providers, developers of major frameworks and systems for language resource creation and use, and appropriate representatives of standards-making groups such as ISO TC37 SC4.

The goal of this collaboration is to ensure that all relevant parties can provide input to the development and/or refinement of standards and practices that promote increased interoperability among web service platforms. Therefore, we continue to reach out to other projects to join the collaboration and, where appropriate, grid federation, including EU projects such as MetaNet/Meta-Share[17], CLARIN[18], and KYOTO[19], with which we have close ties and which are developing their own services and service grids, as well as large projects developing NLP components and data such as the Global WordNet Grid[20] and U-Compare[21], which provides an interface to UIMA-based components primarily for the Biomedical domain. We are also pursuing potentially fruitful uni-directional federations, in which other grids and service nodes are one-way users of the LAPPS Grid; for example, users of an e-Learning Grid could be users of the LAPPS Grid in order to develop e-learning resources, but the LAPPS Grid need not be a user of the e-Learning Grid.

9. Conclusion

The LAPPS Grid project is currently in its second year and has so far provided the basic functionality of the framework. The next steps include expanding the range of services offered, enhancing the Composer interface, and fully implementing the mechanisms to handle licensing. As our intention is to provide one piece of what is envisioned to become a global network of federated grids and services for NLP, another important activity is to pursue additional collaborations with similar projects around the world, and to work to ensure the maximal involvement of the community in the development of exchange mechanisms. We are also seeking means to incorporate individual services and composite service pipelines into the LAPPS Grid (either via direct inclusion or federation with grids that provide these services) for tasks relevant for research in areas such as digital humanities and bioinformatics, and in general to better accommodate the non-technical user.

[17] http://www.meta-net.eu/.

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

[19] http://www.kyoto-project.eu/.

[20] http://www.globalwordnet.org/gwa/gwa_grid.htm

[21] http://u-compare.org/

10. Acknowledgements

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

11. References

Calzolari, N., Baroni, P., Bel, N., Budin, G., Choukri, K., Goggi, S., Mariani, J., Monachini, M., Odijk, J., Piperidis, S., Quochi, V., Soria, C., and Toral, A., editors. (2009). *Proceedings of "The European Language Resources and Technologies Forum: Shaping the Future of the Multilingual Digital Europe"*. ILC-CNR.

Ferrucci, D., Nyberg, E., Allan, J., Barker, K., Brown, E., Chu-Carroll, J., Ciccolo, A., Duboue, P., Fan, J., and Gondek, D. (2009a). Towards the open advancement of question answering systems. *Science*, 24789:RC24789.

Ferrucci, D., Nyberg, E., Allan, J., Barker, K., Brown, E., Chu-Carroll, J., Ciccolo, A., Duboue, P., Fan, J., Gondek, D., Hovy, E., Katz, B., Lally, A., McCord, M., Morarescu, P., Murdock, B., Porter, B., Prager, J., Strzalkowski, T., Welty, C., and Zadrozny, W. (2009b). Towards the Open Advancement of Question Answering Systems. Technical report, IBM Research, Armonk, New York.

Ferrucci, D. A., Brown, E. W., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A., Lally, A., Murdock, J. W., Nyberg, E., Prager, J. M., Schlaefer, N., and Welty, C. A. (2010). Building Watson: An overview of the DeepQA project. *AI Magazine*, 31(3):59–79.

Fokkens, A., van Erp, M., Postma, M., Pedersen, T., Vossen, P., and Freire, N. (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.

Garduno, E., Yang, Z., Maiberg, A., McCormack, C., Fang, Y., and Nyberg, E. (2013). CSE Framework: A UIMA-based Distributed System for Configuration Space Exploration Unstructured Information Management Architecture. In Klgl, P., de Castilho, R. E., and Tomanek, K., editors, *UIMA@GSCL*, CEUR Workshop Proceedings, pages 14–17. CEUR-WS.org.

Ide, N. and Suderman, K. (2014). The Linguistic Annotation Framework: A Standard for Annotation Interchange and Merging. *Language Resources and Evaluation*.

Ide, N., Pustejovsky, J., Calzolari, N., and Soria, C. (2009). The SILT and FlaReNet international collaboration for interoperability. In *Proceedings of the Third Linguistic Annotation Workshop, ACL-IJCNLP*, August.

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

Patel, A., Yang, Z., Nyberg, E., and Mitamura, T. (2013). Building an optimal QA system automatically using configuration space exploration for QA4MRE'13 tasks. In *Proceedings of CLEF 2013*.

Piperdis, S. (2012). The META-SHARE Language Resources Sharing Infrastructure: Principles, Challenges, Solutions. In *Proceedings of the Eighth International Language Resources and Evaluation (LREC12)*, Istan-

bul, Turkey. European Language Resources Association (ELRA).

W3C OWL Working Group. (2012). *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation.

W3C SKOS Working Group. (2009). *SKOS Simple Knowledge Organization System Reference*. W3C Recommendation.

Windhouwer, M. (2012). RELcat: a Relation Registry for ISOcat data categories. In Calzolari, N., Choukri, K., Declerck, T., Dogan, M. U., Maegaard, B., Mariani, J., Odijk, J., and Piperidis, S., editors, *LREC*, pages 3661–3664. European Language Resources Association (ELRA).

Yang, Z., Garduno, E., Fang, Y., Maiberg, A., McCormack, C., and Nyberg, E. (2013). Building optimal information systems automatically: Configuration space exploration for biomedical information systems. In *Proceedings of the CIKM'13*.