U-Compare: an integrated language resource evaluation platform including a comprehensive UIMA resource library

Yoshinobu Kano[1] Ruben Dorado[1] Luke McCrohon[1] Sophia Ananiadou[2] Jun'ichi Tsujii[1,2]

[1] Department of Computer Science, University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo 113-0033 Tokyo

[2]School of Computer Science, University of Manchester and National Centre for Text Mining, 131 Princess St, M1 7DN, UK

E-mail: [kano,rdorado,tsujii]@is.s.u-tokyo.ac.jp, luke.mccrohon@gmail.com, sophia.ananiadou@manchester.ac.uk

Abstract

Language resources, including corpus and tools, are normally required to be combined in order to achieve a user's specific task. However, resources tend to be developed independently in different, incompatible formats. In this paper we describe about U-Compare, which consists of the U-Compare component repository and the U-Compare platform. We have been building a highly interoperable resource library, providing the world largest ready-to-use UIMA component repository including wide variety of corpus readers and state-of-the-art language tools. These resources can be deployed as local services or web services, even possible to be hosted in clustered machines to increase the performance, while users do not need to be aware of such differences. In addition to the resource library, an integrated language processing platform is provided, allowing workflow creation, comparison, evaluation and visualization, using the resources in the library or any UIMA component, without any programming via graphical user interfaces, while a command line launcher is also available without GUIs. The evaluation itself is processed in a UIMA component, users can create and plug their own evaluation metrics in addition to the predefined metrics. U-Compare has been successfully used in many projects including BioCreative, Conll and the BioNLP shared task.

1. Introduction

Language resources have been increasing year by year, not just in their quantities but also in their varieties, even for the same sort of resources. From the users' point of view, it is getting a difficult issue how to select the most suitable set of the resources, in order to achieve the users' specific goal, from among huge number of possible combinations of the resources.

Simply collecting links to resources is not enough, in order to satisfy such needs of users. We should provide the language resources in a way which allows users to compare and evaluate the resources for any corpus of any required domain, by as less human work as possible.

The interoperability is the key issue for such actual use cases of the resources. UIMA (Ferrucci, et al., 2006), Unstructured Information Management Architecture, is an open framework for the interoperability, which is an OASIS standard and an open source project in Apache, getting widely used in the community, e.g. CMU component repository, JCoRe (Hahn, et al., 2008) BioNLP Component Repository (Baumgartner, et al., 2008).

However, since UIMA is a generic framework and APIs, it is still not enough to be truly interoperable to make the human work decreased. We have been developing U-Compare (Kano, et al., 2009), an integrated language resource platform based on the UIMA framework. U-Compare largely consists of two parts: a comprehensive language resource kit as the world largest UIMA component repository collected from the world's famous resources, and an integrated text mining platform for any UIMA component to be very easily combined, run, compared, evaluated and the results visualized.

The U-Compare system is publicly available (http://u-compare.org/), users can run and evaluate any workflow of components for any corpus without any programming. U-Compare initiative is a joint project between the University of Tokyo, UK National Centre for Text Mining, and the University of Colorado School of Medicine.

In this paper we describe details of the U-Compare system, focusing on the language resources available through U-Compare.

2. UIMA

UIMA is an open framework specified by OASIS[1]. Apache UIMA[2] provides a reference implementation as an open source project, with both a pure java API and a C++ development kit. UIMA itself is intended to be purely a framework, i.e. it does not intend to provide specific tools or type definitions. Users should develop such resources themselves.

The UIMA framework uses the "stand-off annotation" style (Ferrucci et al., 2006). The underling raw text of a document is generally kept unchanged during analysis, and the results of processing the text are added as new stand-off annotations with references to their positions in the raw text. A *Common Analysis Structure* (*CAS*) holds a set of such annotations. Each of which is of a given *type* as defined in a specified hierarchical *type system*. Annotation[3] types may define features, which are

[1] http://www.oasis-open.org/committees/uima/

[2] http://incubator.apache.org/uima/

3 In the UIMA framework, Annotation is a base *type* which

Type	Name
Corpus and File Readers	Bio1, BioIE, Texas, Yapex Reference/Test, NLPBA, BioCreative1a, AImed, BioNLP '09 Shared Task, Input Text, Plain Text Files, XMI, BIO
Sentence Detectors	GENIA, LingPipe, NaCTeM, OpenNLP, UIMA
Tokenizers	GENIA, OpenNLP, UIMA, PennBio
POS Taggers	GENIA, LingPipe, OpenNLP, Stepp
Lemmatizers	morpha, GENIA, Enju
Syntactic Parsers	Enju, mogura (HPSG); OpenNLP (CFG); Stanford (Dependency)
Named Entity Recognizers	ABNER (NLPBA/BioCreative/User Model), GENIA Tagger, NaCTeM Species Word Detector, NeMine,, MedTNER-M, Moara CBR-Tagger, LingPipe Entity Tagger (Genia, Genia-NLPBA, GeneTag), OpenNLP
Named Entity Normalizers	NaCTeM Species Disambiguator, MedTNER
Biological Event Detectors	(release planned in early 2010)
Abbreviation Detectors	Extractabbrev
Visualizers and Integrated Tools	Annotation Viewer, MoriV (HPSG feature and tree structure viewer), U-Compare Parallel Component, etc.
Evaluation Components	Boundary, BioNLP Strict/Approximate

Table 1. List of publicly available U-Compare components

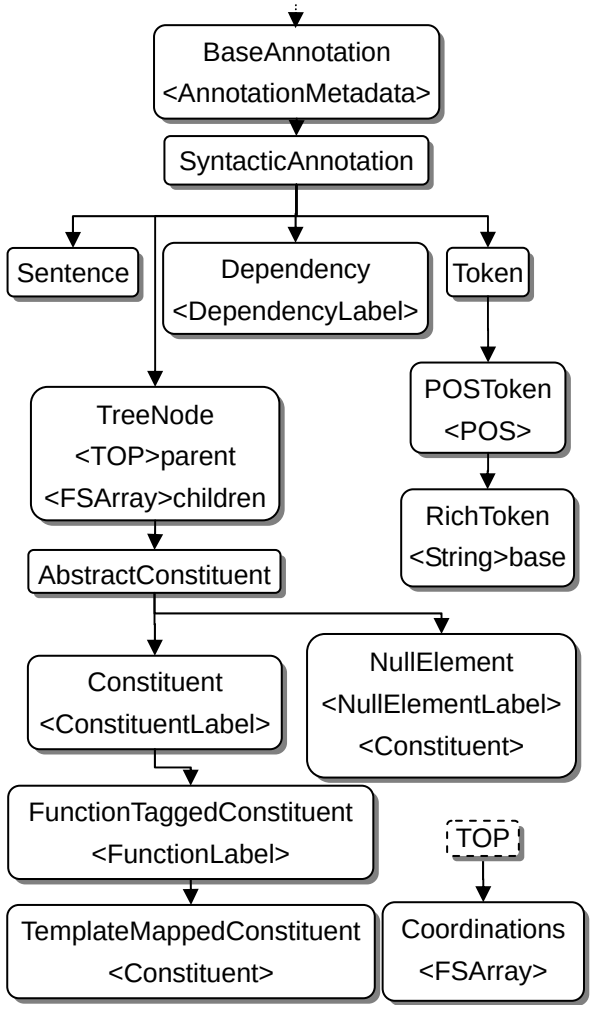

Figure 1. Syntactic Types in U-Compare.

has *begin* and *end* offset values. In this paper we call any objects (any subtype of TOP) as *annotations.*

themselves typed. Apache UIMA provides definitions of a range of built in primitive types, but a more complete type system should be specified by developers. The top level Apache UIMA type is referred to as TOP, other priitive types include. int, String, Annotation and FSArray (an array of any annotations).

UIMA also standardizes component metadata, web service component, workflow metadata, and programmable workflow order controller, possible to represent any workflow configuration.

3. U-Compare Type System

Although UIMA is an excellent framework which provides the interoperability, there is no standardized official type system. Even if the components are UIMA compliant, incompatibility of the type system makes the components incompatible.

We have designed our U-Compare type system considering possible requirements for a type system to be shared among many components (Kano, et al., 2008a).

The basic requirement is that the given information should be able to be fully represented by the type system. However, if we properly encode the information in a single string value, there is no need to use the types. Thus this requirement does not pose any restriction on the type system design.

The main usage of the type system is the I/O capability, which characterizes a component by the supposed input and output types. When we determine which component can be connected with another component, the I/O capabilities are the only clue which is exposed as machine readable information. This requires the type system to be

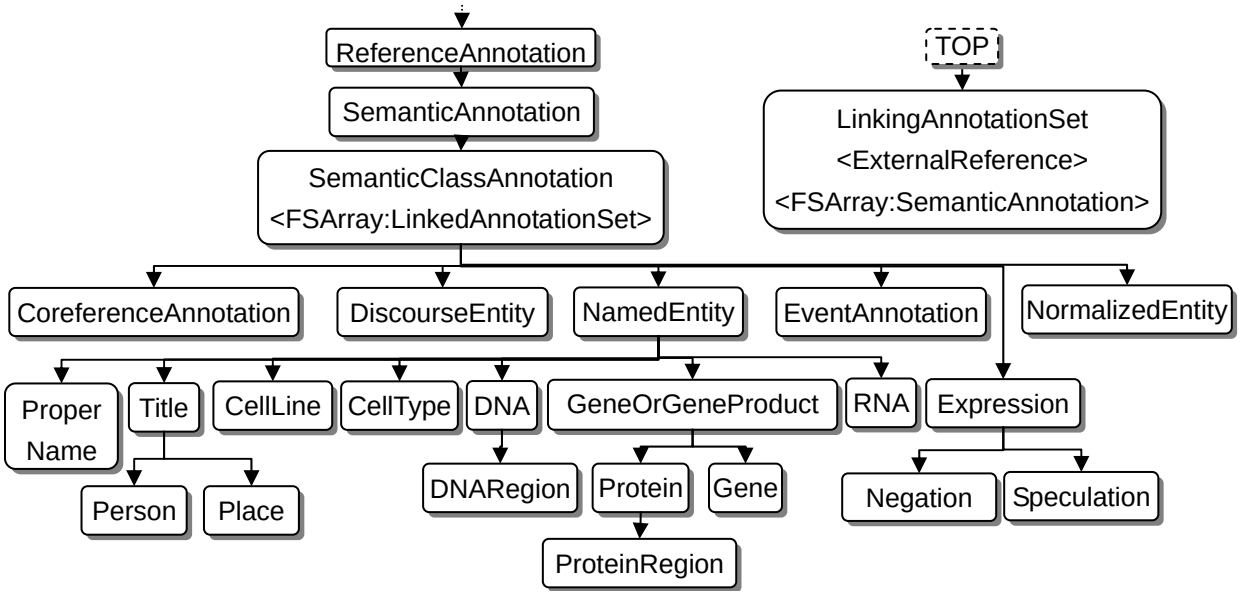

Figure 2. Semantic types in the U-Compare type system.

hieratical enough. For example, many part-of-speech taggers can perform tokenization at the same time. Then a type of token with a part-of-speech field (POSToken) can be the output capability type of a part-of-speech tagger. This POSToken type can also be used to represent the output capability type of simple tokenizers. However, using POSToken as the output capability for both part-of-speech taggers and simple tokenizers makes it impossible to distinguish the differences of these two types of tools. Therefore, we need to define a Token type as a parent type of the POSToken type without the part-of-speech field.

Another usage of the type system is to retain the uniqueness of the data. If there is already a finite set of well defined tags, including these tags as UIMA types will retain the uniqueness. For example, the Penn Treebank tagset is such a well defined tagset, while there are potential ambiguities when a tag is represented as a string value (e.g. upper/lower cases).

These criterions are not enough to determine which concept should be defined as a type. Ontology based entities are frequently seen in the named entity recognition, but mapping all of the potential ontology entities to the type system is not a realistic solution because the number of the entities is essentially infinite and the ontology itself tends to be updated version to

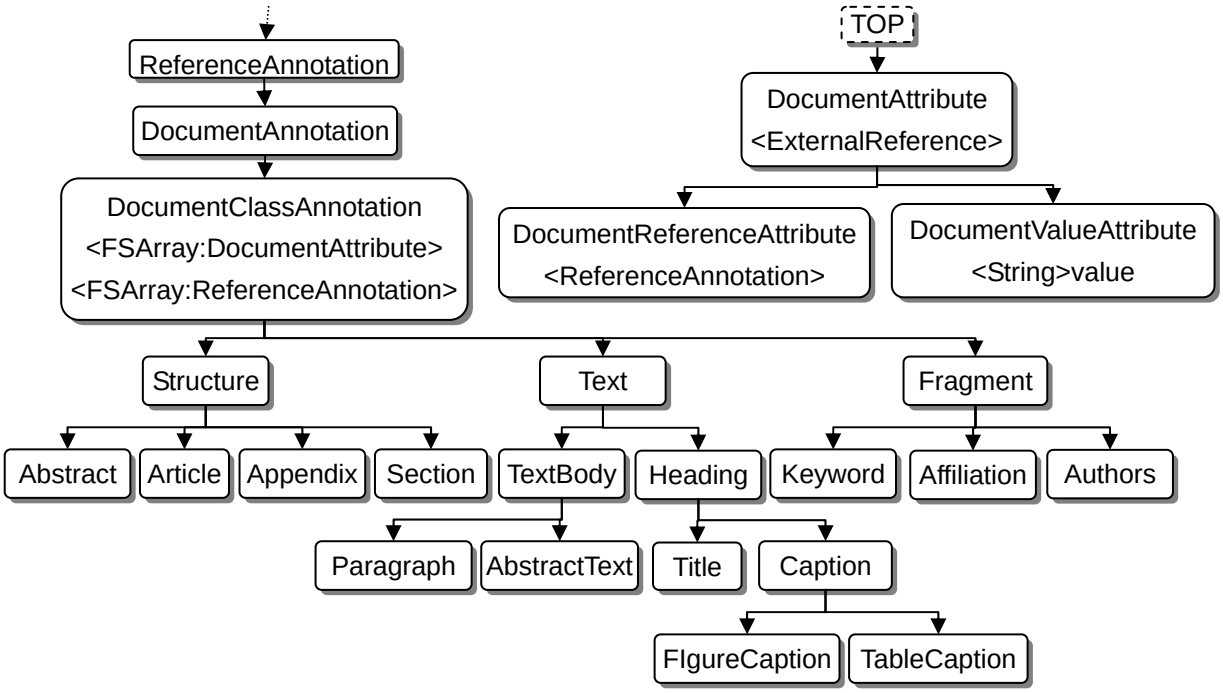

Figure 3. Document types in the U-Compare type system.

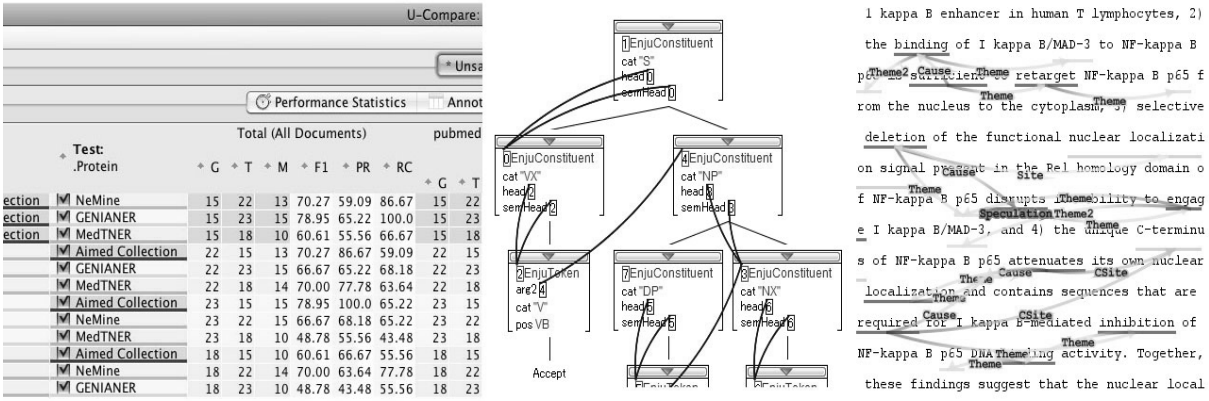

Figure 4. Screenshots of a) U-Compare Statistics Viewer showing comparison between AImed corpus and three NERs, b) U-Compare Tree and Feature Structure Visualizer showing an HPSG syntactic tree, and c) U-Compare Graphical Annotation Viewer showing biological event annotations.

version. In such cases, our criterion whether to define a concept as a type or not is based on the actual use cases of the resources. For example, Protein, DNA, RNA, CellLine, and CellTypes are defined as types because these types are used in the JNPBA shared task evaluation (Kim, et al., 2004).

Based on these criterions, we have designed the U-Compare type system which covers syntactic (Figure 1), semantic (Figure 2), and document related concepts (Figure 3).

4. U-Compare Component Repository

We have been collecting and providing world famous language resources as UIMA components, the U-Compare component repository. U-Compare UIMA components are not just UIMA compliant, but fully compatible with the U-Compare type system. Since some components are originally developed in another type system, we made a type system converter to make it compatible in such a case. Due to this higher level of the interoperability, users only need to be aware of the input and output data types of each component when connecting resources to a workflow.

U-Compare covers broad range of language resources, including annotated corpora, sentence detectors, tokenizers, part-of-speech taggers, dependency parsers, syntactic parsers, named entity taggers, named entity normalizers, relation extractors, etc.

Table 1 shows the current list of the public available language resources. Some of the resources are provided as web services hosted worldwide. Users can mix local and web services transparently when creating their workflows.

5. U-Compare Integrated Platform

U-Compare provides an integrated natural language processing/text mining platform for any UIMA component, which supports users' development cycle: workflow creation, execution, analysis of the result (including evaluation, comparison and visualizations). We provide an easy Graphical User Interface (GUI) based system for such tasks but the command-line based way is also available.

5.1 Launcher Systems

The U-Compare integrated platform, both GUI based and command-line version, can be launched with a single click or a single line of a command, using our UCLoader launcher system. UCLoader only requires Java 6 in the machine, installation and updates (if any) are automatic via Internet. Once launched, offline execution is also available.

5.2 Workflow Creation GUI

The workflow creation GUI consists of two panes, the component library pane and the workflow pane. Users can register any UIMA component into the library while there are many U-Compare components ready to use. With a simple drag and drop motion from the library pane, users can create any workflow easily. Created workflow is a UIMA CPE (Collection Processing Engine) workflow which is still completely compliant to UIMA.

5.3 Combinatorial Comparison

Comparison between components and evaluation with the gold standard data are normally the key process of the NLP/TM development. U-Compare supports these tasks in a very simple manner; when creating a workflow, just to specify which components to compare and which corpus to use for the evaluation. This function is currently available only in U-Compare among many academic UIMA based systems,

Further, U-Compare has an automatic combinatorial comparison system. An NLP/TM workflow tends to consist of a serial pipeline of components, where some of the components could be replaced with similar other components (e.g. a part-of-speech tagger could be replaced with another part-of-speech tagger). If the user specifies such candidate components in the workflow,

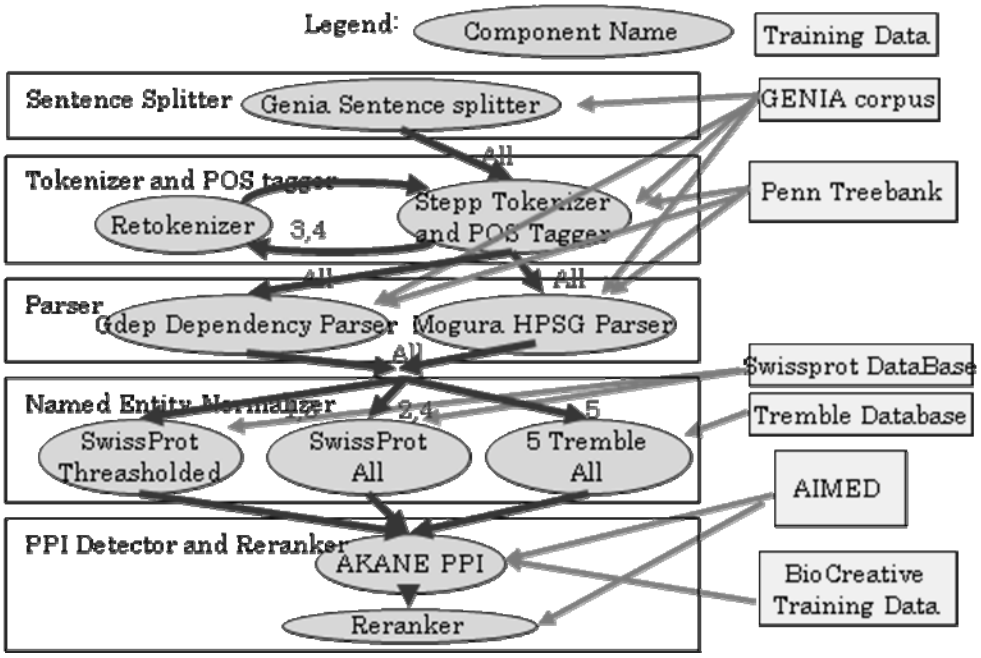

Figure 5. Workflow diagram of the BioCreative II.5 challenge. We provided five workflows sharing some components, "All", "3,4", "5", etc. means which workflow used which component.

U-Compare automatically calculates possible combinations of the components and compares the generated combination workflows.

5.4 Pluggable Evaluation System

Comparison and evaluation need evaluation metrics how to compare the generated annotations. U-Compare provides basic evaluation metrics, but users can plug their own evaluation metrics as a UIMA component (Kano, et al., 2009). Using the evaluation metrics is also very simple without any programming effort.

5.5 Statistics and Evaluation

Based on such comparison and evaluation results, U-Compare shows statistics and instance based visualizations (Figure 1). Most of the TM/NLP researches only show the direct evaluation of the tool. However, it is often discussed how much the scores are improved versus the baseline, then not just the evaluation score between the gold standard data but also the differences between the tool and baseline could be useful. U-Compare shows such statistics as well.

Visualization is another key feature to improve the TM/NLP development process, together with the interoperability and utility features. U-Compare provides a couple of visualization tools specialized on the characteristics of the language resource data structures.

5.6 Command-Line Mode

Although the GUI based system described above is useful and decrease the human works, there are certain requirements from the users to access the resources without GUIs. For example, a user might wish to use their headless server. U-Compare provides a command-line launcher which launches a specified workflow.

Since U-Compare components are UIMA compliant, it is also very easy to embed these components into existing UIMA workflows or UIMA based systems. The only issue is the type system compatibility.

The UIMA framework is provided both in pure Java and C++, it is straightforward to call a UIMA/U-Compare workflow in a normal Java/C++ applications, using the official UIMA APIs.

Further, U-Compare provides a simple stand-off format I/O via the standard input/output streams. Developers using scripting languages can easily wrap their tools to be UIM A compliant.

5.7 Remote Deployments

NLP components are sometimes very heavy, consuming large computational resources. Such a heavy component tends to occupy large amount of the process time in a workflow (e.g. a syntactic parser tends to be a heavier process). Due to the independency between documents processed in most of the TM/NLP workflows, we can increase the total performance by making the heaviest component parallelized, processing many documents at the same time. We have developed an automatic cluster deployment system which parallelizes any UIMA component to the cluster system, as a whole pretends like a single very fast component via a load-balancing gateway machine.

6. Use Cases

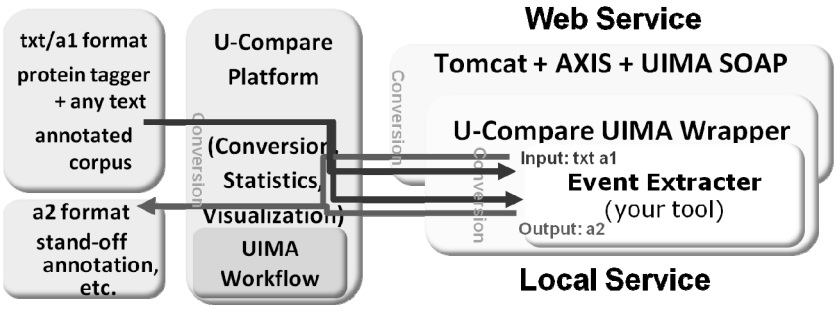

Figure 6. A conceptual diagram of the U-Compare event service. The "Event Extracter" should be provided by the developer, while all other parts are provided by the U-Compare wrapper package.

U-Compare has been successfully used in many projects. In this section, we describe about various actual use cases from end-user application to service provider.

6.1 BioCreative II.5

We participated the BioCreative II.5 task using the U-Compare system as a base platform to create and run the required services, achieved one of the best results among 45 submitted result sets in the Interacting Pairs Task (IPT) (Sætre, et al., 2009). We prepared several U-Compare compatible components and tried many possible workflows, provided five workflows as web services which were considered to perform better among them (Figure 5). Since some components e.g. the named entity normalizers consumed huge computational resources and required long initialization time, such components were internally deployed as web services, shared among the workflows.

6.2 Shared Task Supports

Since the U-Compare component repository includes most of the commonly used types of tools and the evaluation system as well, U-Compare is very useful for the participants of shared task challenges when shared task specific components are provided. U-Compare supported the BioNLP '09 Shared Task on Event Extraction as an official support system (Kim, et al., 2009), provided a shared task corpus reader and evaluation components in addition to the original U-Compare components. Further, we have aggregated the participants' results by majority voting using U-Compare, achieved the world best score which is four points better than the best participant result. The CoNLL 2010 shared task also provided corpus reader/writer as U-Compare compatible components.

6.3 U-Compare Bio-Event Server

We have wrapped nine state-of-the-art event extraction tools to be U-Compare compatible Bio-Event Servers, collaborating with some of the BioNLP '09 Shared Task participants (Kano, et al., to appear). While most of the original I/O formats are as same as the shared task corpus, the wrapped services are completely U-Compare compatible. This compatibility allows combinations with many U-Compare compatible protein mention taggers or protein mention annotated corpora without any programming, which were previously pointed to be difficult for end users (Kabiljo, et al., 2009). From the developer's point of view, the wrapping process itself is also very simple if the original I/O format is as same as the shared task corpus (Figure 6).

7. Summary and Future Directions

U-Compare is a UIMA based all-in-one TM/NLP system with the world largest UIMA component repository. Our aim is not only collecting interoperable language resources, but to provide a total solution for the users decreasing the human works as much as possible, in order for the users to concentrate on their essential tasks.

Future directions include addition of more language resources (e.g. non-English) and evaluation metrics.

Acknowledgments

We wish to thank Dr. Lawrence Hunter's text mining group at Center for Computational Pharmacology, University of Colorado School of Medicine, for helping build the type system and for making their tools available for this research. This work was partially supported by Grant-in-Aid for Specially Promoted Research and Grant-in-Aids for Scientific Research (C) (MEXT, Japan). The National Centre for Text Mining is funded by JISC.

References

Baumgartner, W.A., Jr., Cohen, K.B. and Hunter, L. (2008) An open-source framework for large-scale, flexible evaluation of biomedical text mining systems, *J Biomed Discov Collab*, **3**, 1.

Ferrucci, D., Lally, A., Gruhl, D., Epstein, E., Schor, M., Murdock, J.W., Frenkiel, A., Brown, E.W., Hampp, T., Doganata, Y., Welty, C., Amini, L., Kofman, G., Kozakov, L. and Mass, Y. (2006) Towards an Interoperability Standard for Text and Multi-Modal Analytics. IBM Research Report.

Hahn, U., Buyko, E., Landefeld, R., Mühlhausen, M., Poprat, M., Tomanek, K. and Wermter, J. (2008) An Overview of JCoRe, the JULIE Lab UIMA Component Repository, In *Proceedings of LREC'08 Workshop, Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP*, 1-8.

Kabiljo, R., Clegg, A.B. and Shepherd, A.J. (2009) A realistic assessment of methods for extracting gene/protein interactions from free text, *BMC Bioinformatics*, **10**, 233.

Kano, Y., Baumgartner, W.A., McCrohon, L., Ananiadou, S., Cohen, K.B., Hunter, L. and Tsujii, J. (2009) U-Compare: share and compare text mining tools with UIMA, *Bioinformatics*, **25**, 1997-1998.

Kano, Y., McCrohon, L., Ananiadou, S. and Tsujii, J. (2009) Integrated NLP Evaluation System for Pluggable Evaluation Metrics with Extensive Interoperable Toolkit, In *Proceedings of Software engineering, testing, and quality assurance for natural language processing workshop (SETQA-NLP), NAACL-HLT*, 22-30.

Kano, Y., Nguyen, N., Sætre, R., Yoshida, K., Miyao, Y., Tsuruoka, Y., Matsubayashi, Y., Ananiadou, S. and Tsujii, J. (2008a) Filling the gaps between tools and users: a tool comparator, using protein-protein interaction as an example, In *Proceedings of Pacific Symposium on Biocomputing (PSB)*, **13**, 616-627.

Kim, J.D., Ohta, T., Pyysalo, S., Kano, Y. and Tsujii, J. (2009) Overview of BioNLP'09 Shared Task on Event Extraction, In *Proceedings of BioNLP 2009 Workshop Companion Volume for Shared Task*, 1-9.

Kim, J.D., Ohta, T., Tsuruoka, Y., Tateisi, Y. and Collier, N. (2004) Introduction to the Bio-Entity Recognition Task at JNLPBA, In *Proceedings of International Workshop on Natural Language Processing in Biomedicine and its Applications (JNLPBA-04)*, 70--75.

Sætre, R., Yoshida, K., Miwa, M., Matsuzaki, T., Kano, Y. and Tsujii, J. (2009) AkaneRE Relation Extraction: Protein Interaction and Normalization in the BioCreAtIvE II.5 Challenge, In *Proceedings of BioCreative II.5 Workshop 2009 special session | Digital Annotations*, 33.