Promoting Interoperability of Resources in META-SHARE

Paul Thompson[1,2], Yoshinobu Kano[3], John McNaught[1,2], Steve Pettifer[1], Teresa Attwood[1,4], John Keane[1] and Sophia Ananiadou[1,2]
[1]Department of Computer Science, University of Manchester, UK
[2]National Centre for Text Mining, University of Manchester, UK
[3]Database Center for Life Science, University of Tokyo, Japan
[4]Faculty of Life Sciences, University of Manchester, UK
{paul.thompson, john.mcnaught, steve.pettifer, teresa.attwood, john.keane, sophia.ananiadou}@manchester.ac.uk
kano@dbcls.rois.ac.jp

Abstract

META-NET is a Network of Excellence aiming to improve significantly on the number of language technologies that can assist European citizens, by enabling enhanced communication and cooperation across languages. A major outcome will be META-SHARE, a searchable network of repositories that collect resources such as language data, tools and related web services, covering a large number of European languages. These resources are intended to facilitate the development and evaluation of a wide range of new language processing applications and services. An important aim of META-SHARE is the promotion of interoperability amongst resources. In this paper, we describe our planned efforts to help to achieve this aim, through the adoption of the UIMA framework and the integration of the U-Compare system within the META-SHARE network. U-Compare facilitates the rapid construction and evaluation of NLP applications that make use of interoperable components, and, as such, can help to speed up the development of a new generation of European language technology applications.

1 Introduction

The two dozen national and many regional languages of Europe present linguistic barriers that can severely limit the free flow of goods, information and services. The META-NET Network of Excellence has been created to respond to this issue. Consisting of 44 research centres from 31 countries, META-NET aims to stimulate a concerted, substantial and continent-wide effort to push forward language technology research and engineering, in order to ensure equal access to information and knowledge for all European citizens.

The success of META-NET is dependent on the ready availability of data, tools and services that can perform natural language processing (NLP) and text mining (TM) on a range of European languages.

These will form the building blocks for constructing language-technology applications that can help European citizens to gain easy access to the information they require. Among these applications will be semantic search systems to provide users with fast and efficient access to precisely the information they require, and voice user interfaces that allow easy access to information and services over the telephone, e.g., booking tickets, etc.

One of the major outcomes of META-NET will be the META-SHARE infrastructure, an open, distributed facility for sharing and exchange of language resources (LRs), consisting of a sustainable network of repositories of language data, tools and related web services for a large number of European languages. LRs will be documented with high-quality metadata and aggregated in central inventories, allowing for uniform search and access to resources. A further aim of META-SHARE is to promote the use of widely acceptable standards for LR building, in order to ensure the greatest possible interoperability of LRs.

META-SHARE shares some goals with related initiatives, such as the Open Language Archives Community (OLAC) (Hughes & Kamat, 2005), which is developing a virtual library of LRs augmented with metadata; the PANACEA project (Bel, 2010), which is creating a library of interoperable web services that automate the stages involved in the production and maintenance of LRs required by MT systems; and the Common Language Resources and Technology Infrastructure

Proceedings of Workshop on Language Resources, Technology and Services in the Sharing Paradigm, pages 50–58,
Chiang Mai, Thailand, November 12, 2011.

(CLARIN) (Váradi et al., 2008), which is establishing an integrated and interoperable research infrastructure of LRs and technology. A memorandum of understanding between META-NET and CLARIN recognizes that they are complementary initiatives with harmonisable goals. Whilst CLARIN is largely oriented towards the social sciences and humanities research community, META-NET aims at supporting Human Language Technology (HLT) development, and thus will target HLT researchers and developers, language professionals (translators, interpreters, etc.), as well as industrial players, with a particular emphasis on cross-lingual technologies.

Advanced language technology applications are usually built from a number of component technologies, which are often common across a large number of different applications. For example, text-based applications frequently make use of tools such as tokenisers, part-of speech taggers, syntactic parsers, named entity recognisers, etc. Through its central inventories and detailed meta-data, META-SHARE will help application developers by facilitating accurate searches to be carried out over a large set of reusable tools, as well as over data on which they can be re-trained and evaluated.

In addition to reusability, a further issue that must be considered is the ease with which component tools can be combined together to create complete applications. Only if this combination can occur with minimal, or no, configuration, can the tools be said to be *interoperable*.

It is often the case that interoperability can be problematic to achieve, especially for resources that have different developers or creators. Reasons for this include the following:

- Use of different programming languages to implement the tools.
- Different input and output formats of the tools (e.g., plain text vs. XML).
- Incompatible data types produced by the tools (e.g., different tag sets).

Having to deal with such issues can be both time-consuming and a source of frustration for the developer, often requiring program code to be rewritten or extra code to be produced in order to ensure that data can pass freely and correctly between the different resources used in the application.

One way to overcome some of the problems of interoperability is to adopt the use of the Unstructured Information Management Architecture (UIMA)[1] (Ferrucci et al., 2006), which aims to facilitate the seamless combination of LRs into workflows that can carry out different natural language processing (NLP) tasks. U-Compare (Kano et al., 2009; Kano et al., 2011), which is built on top of UIMA, provides additional means for ensuring more universal interoperability between resources, as well as providing special facilities that allow the rapid construction and evaluation of natural language-processing/text-mining applications using interoperable UIMA-compliant resources, without the need for any additional programming.

METANET4U is one of a set of projects (together with META-NORD and CESAR), which are preparing LRs that operate on a wide range of different European languages for inclusion within META-SHARE. Part of the contribution of the METANET4U project is to encourage LR providers to make their resources UIMA-compliant. This is partly being achieved through the creation of a pilot version of META-SHARE, in which standard functionality is enhanced through the integration of U-Compare. As an initial step, UIMA-compliant LRs are currently being created for a subset of European languages, based on the resources that will be made available by the METANET4U partners. This will allow us to demonstrate that META-SHARE has the potential to serve not only as a useful tool to locate resources for a range of languages, but also to act as an integrated environment that allows for rapid prototyping and testing of applications that make use of these resources.

2 UIMA

In recent years, the issue of interoperability has been receiving increasing attention, e.g., Copestake et al. (2006); Cunningham et al. (2002); Laprun et al. (2002). UIMA provides a flexible and extensible architecture for implementing interoperability, which is achieved largely by virtue of a standard means of communication between resources when they are combined together into workflows.

2.1 Wrapping resources

At the heart of the UIMA framework is a data structure called the Common Analysis Structure (CAS). During the execution of a workflow, the

[1] http://uima.apache.org/

CAS is accessible by all resources, and stores all annotations, e.g., tokens, part-of-speech tags, syntactic parse trees, etc., that have been produced by the different resources. Each resource to be used within the UIMA framework must be "wrapped" as a UIMA component. This means that it must be specifically configured to obtain its input by reading data from the CAS. As output, UIMA components should add new annotations to the CAS, or update annotations already contained within it. For example, a tokeniser tool may add *Token* annotations to the CAS. A POS tagger may read *Token* annotations, and add a *POS* feature to them.

A standard way of reading, writing and updating the CAS, which must be followed by all UIMA components, means that differences in input/output formats of resources are essentially hidden, once the wrapper has been written. It is this feature that allows flexible and seamless combination of UIMA components into pipelines/workflows.

In order to facilitate such interoperability, a certain amount of overhead is required to create the wrapper code. Given that resources differ in their input/output format and parameters, a specialised wrapper must normally be produced for each different resource, although the general structure of the wrapper code is usually similar. The basic steps are as follows:

1. Read appropriate annotations from the CAS.
2. Convert the UIMA annotations to input format required by the tool (e.g., plain text, XML, standoff annotations, inline annotations, etc.)
3. Execute the tool, passing the correctly formatted input to it.
4. Convert the output of the tool to UIMA annotations.
5. Write or update the CAS with the newly generated UIMA annotations.

An example of a possible workflow for carrying out named entity recognition is the following:
Sentence Splitter →Tokeniser → POS Tagger → Syntactic Parser →Named Entity Recogniser

In combining resources together, it is only necessary to ensure that the types of annotation required as input by a particular component are present in the CAS at the time of execution of that component. For example, tokenisers generally require text that has been split into sentences as input. Thus, if such a tokeniser is to be included in a workflow, one of the components executed earlier in the workflow should produce output corresponding to sentence annotations. The UIMA framework makes this process quite straightforward, since each UIMA component must declare its input/output annotation types in a separate descriptor file.

The UIMA framework also deals with another issue of interoperability, in that after resources are wrapped as UIMA components, the original programming language is hidden and thus becomes irrelevant. Writing the UIMA wrapper is fairly straightforward when the resource is implemented in either Java or C++, or if the tool is available as a web service or as a binary.

2.2 Compatibility of data types

As mentioned above, each UIMA component must declare its input and output annotation types. Annotation types are separately declared in a *type system* descriptor file, and may be hierarchically structured. For example, a type *SemanticAnnotation* may specify *NamedEntity* and *Coreference* as subtypes. Each annotation type may additionally define features, e.g., a *Token* type may have a *PartOfSpeech* feature.

The UIMA framework itself does not impose or recommend the use of a particular type system. Accordingly, the various existing repositories of UIMA components (e.g., the BIONLP UIMA Component Repository (Baumgartner et al., 2008), the CMU UIMA component repository[2] and the UIMA-fr consortium (Hernandez et al., 2010)) generally make use of different type systems. This can be a major barrier to universal interoperability of resources. Although resources chosen from the same repository are likely to be interoperable, the same cannot be said for resources chosen from multiple repositories. This is because the individual type systems may use different package names, different names for annotation types or have different hierarchical structures, even though functionalities of the components across different repositories may be similar.

Ideally, in order to achieve maximum interoperability, a single, common type system would be imposed, to be followed by all developers of UIMA components. However, this is not considered a viable option, as it would be difficult to achieve consensus on exactly which types should be present, given, for example, the various different syntactic and semantic theories on which different tools are based.

[2] http://uima.lti.cs.cmu.edu

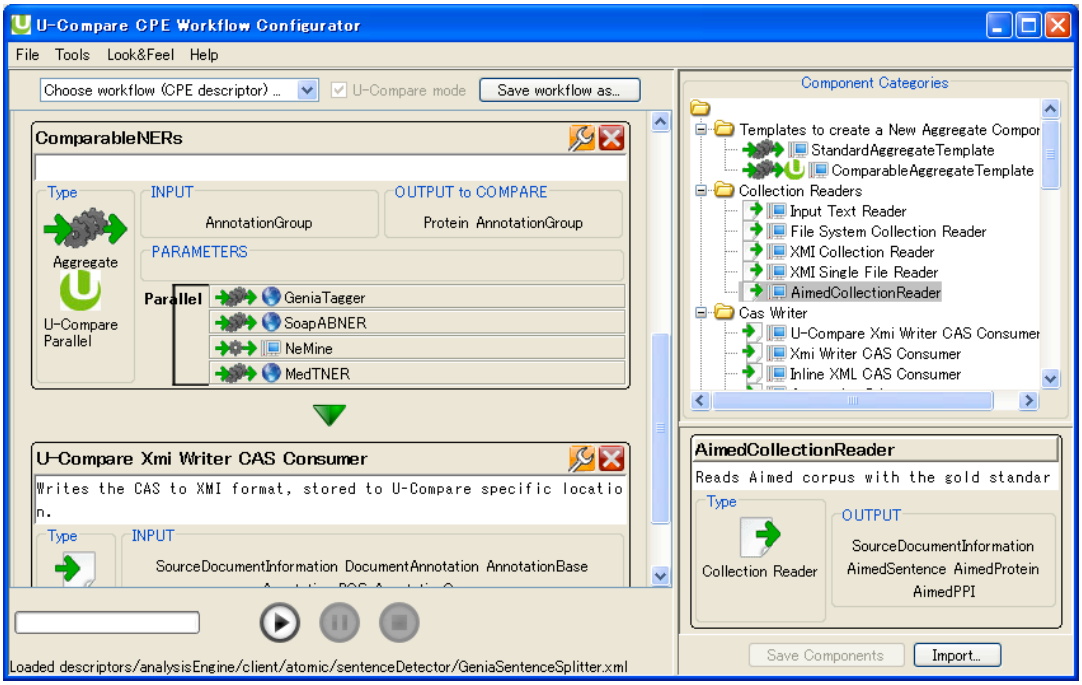

Figure 1: U-Compare interface

3 U-Compare

U-Compare (Kano et al., 2009; Kano et al., 2011) is a system built on top of UIMA. The main goals of U-Compare are to allow rapid and flexible construction of NLP applications and evaluation of these applications against gold-standard annotated data, without the need for any additional programming.

U-Compare builds upon the core elements of UIMA to provide a graphical user interface, which allows users to construct and configure workflows of UIMA components, using simple drag-and-drop actions, and to apply the workflow to a corpus of documents at the click of a button.

U-Compare includes several built-in annotation viewers, making it easy to visualise the various annotations produced by workflows, including more complex annotation types, such as syntactic trees and feature structures. The main U-Compare interface is shown in Figure 1, with the library of available components on the right, and the workflow builder on the left.

The rapid construction of NLP workflows is reliant on the ready availability of component resources. U-Compare is distributed with a library of over 50 UIMA components, constituting the world's largest type-compatible UIMA repository. A particular emphasis on biomedical text processing allows specialised, complex workflows to be constructed, e.g., to disambiguate species of biomedical named entities (Wang et al., 2010).

3.1 Evaluation in U-Compare

U-Compare additionally provides special facilities for evaluating the performance of workflows. For each step of a workflow (e.g., part-of-speech tagging, parsing, etc.) there are often several tools that could be used. U-Compare can compare the performance of each possible combination of tools against a gold standard annotated corpus, i.e., a corpus in which information of the type produced by the tool has been marked-up manually by human annotators. Such a comparison allows the best performing workflow for one's particular task to be determined. Results are reported in terms of performance statistics, precision, recall and F-score. The U-Compare evaluation interface is shown in Figure 2. On the left are the performance statistics and on the right are the annotations produced by the various tools under evaluation.

The power of U-Compare's evaluation framework has recently been demonstrated in the recognition of chemical named entities in scientific texts (Kolluru et al., 2011). A well–established named entity recogniser for the chemistry domain, Oscar3 (Corbett & Murray-Rust, 2006), had a rigid structure, which made it difficult to modularise and to adapt to new and emerging trends in annotation and corpora.

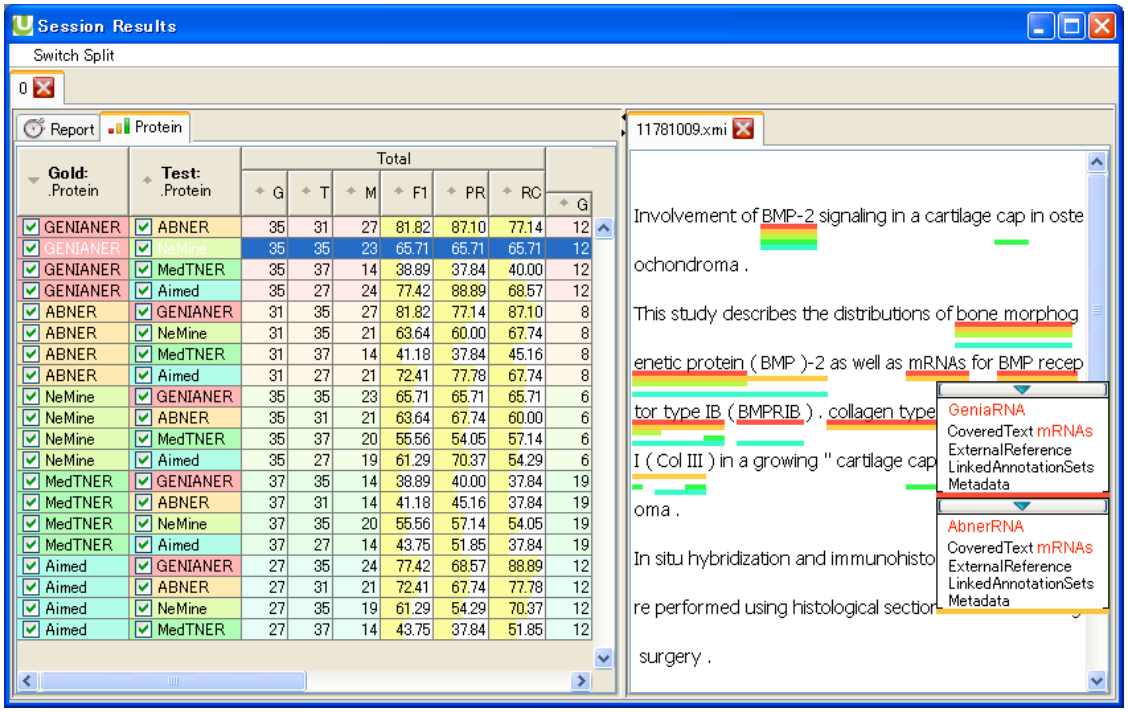

Figure 2: Evaluation in U-Compare

Oscar3 was refactored into a number of separate, reconfigurable U-Compare components, and experiments showed that the substitution of a new tokeniser into the workflow could improve performance over the original system. The new, modularised version of Oscar (OSCAR4[3]) has recently been released.

A similar approach could also be used to improve the performance of other types of applications relevant to language technology, e.g., machine translation systems such as Apertium (Armentano-Oller et al., 2006), which also has a modular architecture.

3.2 U-Compare type system

U-Compare's current inventory of components has been drawn from a number of different sources, including existing UIMA repositories that use their own type systems. This meant that issues of type system compatibility had to be faced. As a partial solution to the type system interoperability problem, U-Compare has defined a *sharable* type system.

The aim of the U-Compare sharable type system is to act as a kind of bridge, to facilitate the construction of workflows containing almost any UIMA components, regardless of their source, or the original type system that they use. Communication between existing UIMA components is made possible by mapping their original input and output types to appropriate types in the U-Compare type system. Newly wrapped components directly use types belonging to the sharable type system. However, such components may define their own type system extensions, as long as any new types defined extend existing types in the hierarchy. It is hoped that the U-Compare type system will eventually be adopted as a standard, which will help to ensure greater interoperability between UIMA components in the future.

As mentioned previously, defining an exhaustive, common type system sufficient for all possible UIMA components would be a virtually impossible task. According to this, the aim of the U-Compare type system is to define a set of types that on the one are hand fairly general, but on the other hand are fine-grained enough to allow the most common types of annotation produced by NLP applications to be represented. The currently defined types correspond to syntactic, semantic and document-level concepts, as illustrated in Figures 3, 4 and 5, respectively.

When mapping between a particular type system and the U-Compare sharable type system, it is inevitable that in certain cases, information loss will occur. This is because the general types of the U-Compare type system cannot encode all the subtleties of information produced by many different components. Therefore, certain aspects of the functionality of a particular resource may

[3] https://bitbucket.org/wwmm/oscar4/

be hidden by the U-Compare type system. However, since one of the aims of U-Compare is to provide as large a library as possible of interoperable NLP components, such a trade-off is sometimes necessary to guarantee such interoperability.

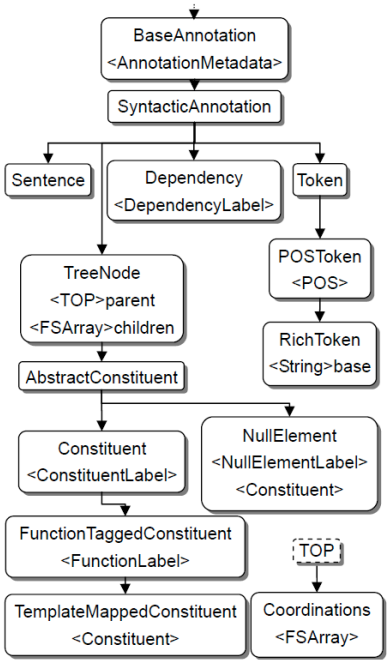

Figure 3: Syntactic types in the U-Compare type system

Despite the possible loss of information when using the U-Compare type system, two important points should be noted. Firstly, the hierarchical nature of the type system aims to minimise information loss as much as possible. Types from exisiting, external systems can be mapped to the most specific type possible in the U-Compare hierarchy. Secondly, since the U-Compare type system is still considered as work in progress, the addition of further well-motivated types will be considered, which could further decrease levels of information loss.

A further advantage of the hierarchical structure of the type system is that it can help to expose clearly the capabilities of a particular resource. Consider, for example, a resource that outputs annotations of type *RichToken* (see Figure 3). These annotations constitute a token whose base form is recorded in the *base* feature. As such, they could be used to store the output of a morphological analyser.

The type system hierarchy tells us that *RichToken* is a subtype of *POSToken,* which stores a token, along with part-of-speech information. Thus, annotations of type *RichToken* will specify not only the base form of the token, but also its part-of-speech. Therefore, if a particular tool requires part-of-speech tagged tokens as input, then it can be executed in a workflow following a tool whose output is *either POSToken* or *RichToken*, since both of these tool types will output token annotations with part-of-speech information. Even though tools outputting *RichToken* information would contain some redundant information in this case, this does not matter, as long as the required information is also present in the CAS.

4 U-Compare and META-SHARE

The utility of U-Compare has already been amply demonstrated through its use in many tasks by both NLP experts and non-expert users, from the individual level to worldwide challenges. These include the BioNLP'09 shared task (Kim et al., 2009) for the extraction of bio-molecular events (bio-events) that appear in biomedical literature, in which U-Compare served as an official support system; the CoNLL-2010 shared task on the detection of speculation in biomedical texts (Farkas et al., 2010); the BioCreative II.5 challenge (Sætre et al., 2009) of text-mining and information-extraction systems applied to the biological domain; and linking with Taverna (Kano et al., 2010), a generic workflow management system.

Mostly, these usages have been limited to the processing of biomedical texts in the English language. Integration within META-SHARE will additionally allow the utility of U-Compare to be demonstrated in a multilingual scenario, where it will help to facilitate the rapid expansion of NLP applications covering a range of European languages. In order to ensure the success of this, a number of different areas have to be addressed.

4.1 Expansion of U-Compare component library

In order to meet with the multilingual and multimodal goals of META-SHARE, the current library of U-Compare components must be expanded. As an initial step, we have identified around 40 resources (both tools and corpora) that concern languages other than English (namely Catalan, French, Maltese, Portuguese, Romanian and Spanish), and which our METANET4U project partners are planning to make available in META-SHARE.

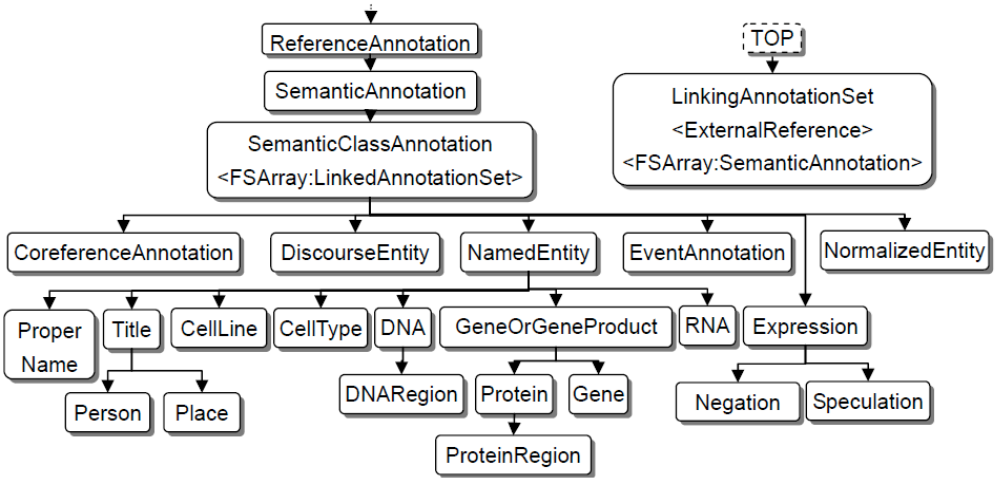

Figure 4: Semantic types in the U-Compare type system

The resources are a mixture of monolingual and multilingual, and concern different modalities (both written and spoken language). As an ongoing task, these resources are being wrapped as UIMA components that comply with the U-Compare type system.

4.2 Evaluation and consolidation of the U-Compare type system

Once completed, the new set of U-Compare compatible UIMA components will almost double the size of the current library, and in creating them, we will be able to consolidate and evaluate the utility of the U-Compare type system in scenarios other than the processing of English biomedical text. This will help us to work towards the goal of defining a sharable-type system that can be applied regardless of language or domain, and which could be promoted as a standard to be followed both in META-SHARE, and beyond.

An initial analysis of the selected resources suggests that, to a large extent, the existing type system is sufficient to describe their inputs and outputs, with no language-specific issues becoming immediately apparent. However, some types of tool that are not currently available in the U-Compare library, such as discourse parsers and semantic role labellers, will motivate a small number of additions to the type system. Since the current version of the type system was created only for written resources, further extensions will need to be made for spoken resources.

4.3 Extending U-Compare functionality

The functionality of the U-Compare software must also be extended to handle the new types of components that will be made available, in particular to provide support for multilingual and speech-based components. As mentioned previously, U-Compare provides annotation viewers that allow annotations produced by workflows to be easily visualised. Since multilingual components will often produce annotations in multiple languages, a new type of viewing component should be developed that allows both source and target language information to be displayed. Viewers for speech-based output will allow speech files to be played and corresponding waveforms to be displayed.

4.4 Specification of workflows

As a final step, we will implement a number of workflows that make use of the newly wrapped components in various ways. Through integration within META-SHARE, these workflows can act as templates for carrying out important language-processing tasks, which may be changed or configured according to the requirements of different types of application.

We have designed workflows for over 20 different tasks, which will be implemented after the appropriate resources have been wrapped. Some of these are fairly simple tasks, which may be considered as building blocks to be used in the construction of more complex workflows (e.g., sentence splitting and POS tagging, etc), whilst others may be considered complete tasks in themselves (e.g., discourse parsing, translation of text, ontology building, etc.), involving 10 or more processing steps.

According to the set of LRs that are currently being wrapped as UIMA components, most of the tasks will be accomplishable in a number of different languages, through the substitution of appropriate alternative components.

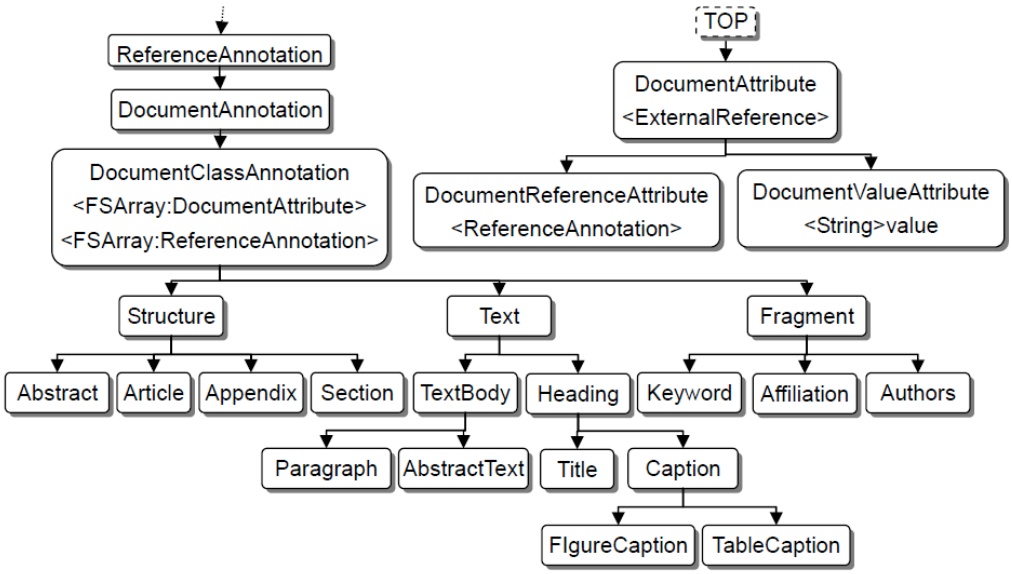

Figure 5: Document-level types in the U-Compare type system

Often, there are several paths that can be taken to complete a given task for each language. For example, some tools perform *both* part-of speech tagging and lemmatization, whilst in other cases, different tools exist to perform each step separately.

Since a number of gold-standard annotated corpora will be made available as U-Compare components, an evaluation of which path produces the best results will often be possible, using U-Compare's evaluation functionalities, as described earlier. By providing facilities for META-SHARE users to make their own workflows available to other users, and to provide feedback about existing workflows, the process of creating new applications could become even easier.

5 Conclusion

The speed and ease with which new applications can be developed using component language resources is heavily dependent on the amount of work that must be performed by system developers to allow such components to communicate with each other in the correct manner. We have described how, by wrapping resources as UIMA components whose annotation types conform to the U-Compare type system, greater interoperability of the resources, and with it, easier reuse and more flexible combination, can be achieved.

It is hoped that the planned integration of the U-Compare system within META-SHARE will contribute to a more rapid and straightforward expansion of the European language technology landscape. The integration will allow users to benefit from running and configuring existing workflows, as well as creating new workflows, with only a few mouse clicks, and without the need to write any new program code.

Acknowledgements

The work described in this paper is being funded by the DG INFSO of the European Commission through the ICT Policy Support Programme, Grant agreement no. 270893 (METANET4U).

References

Armentano-Oller, C., Carrasco, R., Corbí-Bellot, A., Forcada, M., Ginestí-Rosell, M., Ortiz-Rojas, S. (2006). Open-source Portuguese–Spanish machine translation. *Computational Processing of the Portuguese Language*, 50-59.

Baumgartner, W. A., Cohen, K. B., & Hunter, L. (2008). An open-source framework for large-scale, flexible evaluation of biomedical text mining systems. *Journal of Biomedical Discovery and Collaboration, 3*, 1.

Bel, N. (2010). Platform for Automatic, Normalized Annotation and Cost-Effective Acquisition of Language Resources for Human Language Technologies: PANACEA. In *Proceedings of XXVI Congreso de la Sociedad Española para el Procesamiento del Lenguaje Natural (SEPLN-2010).*

Copestake, A., Corbett, P., Murray-Rust, P., Rupp, C. J., Siddharthan, A., Teufel, S. (2006). An architecture for language processing for scientific

texts. In *Proceedings of the UK e-Science All Hands Meeting 2006*.

Corbett, P., & Murray-Rust, P. (2006). High-throughput identification of chemistry in life science texts. *Computational Life Sciences II*, 107-118.

Cunningham, D. H., Maynard, D. D., Bontcheva, D. K., & Tablan, M. V. (2002). GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, pp. 168-175.

Farkas, R., Vincze, V., Móra, G., Csirik, J., & Szarvas, G. (2010). The CoNLL-2010 shared task: learning to detect hedges and their scope in natural language text. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning---Shared Task*, pp. 1-12.

Ferrucci, D., Lally, A., Gruhl, D., Epstein, E., Schor, M., Murdock, J. W. (2006). Towards an Interoperability Standard for Text and Multi-Modal Analytics. *IBM Research Report RC24122*.

Hernandez, N., Poulard, F., Vernier, M., & Rocheteau, J. (2010). Building a French-speaking community around UIMA, gathering research, education and industrial partners, mainly in Natural Language Processing and Speech Recognizing domains. In *LREC 2010 Workshop on New Challenges for NLP Frameworks*, pp. 41-45.

Hughes, B., & Kamat, A. (2005). A metadata search engine for digital language archives. *D-Lib Magazine, 11*(2), 6.

Kano, Y., Baumgartner, W. A., Jr., McCrohon, L., Ananiadou, S., Cohen, K. B., Hunter, L. (2009). U-Compare: share and compare text mining tools with UIMA. *Bioinformatics, 25*(15), 1997-1998.

Kano, Y., Dobson, P., Nakanishi, M., Tsujii, J., & Ananiadou, S. (2010). Text mining meets workflow: linking U-Compare with Taverna. *Bioinformatics, 26*(19), 2486-2487.

Kano, Y., Miwa, M., Cohen, K. B., Hunter, L. E., Ananiadou, S., & Tsujii, J. (2011). U-Compare: A modular NLP workflow construction and evaluation system. *IBM Journal of Research and Development, 55*(3), 11:1-11:10.

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

Kolluru, B., Hawizy, L., Murray-Rust, P., Tsujii, J., & Ananiadou, S. (2011). Using Workflows to Explore and Optimise Named Entity Recognition for Chemistry. *PLoS ONE, 6*(5), e20181.

Laprun, C., Fiscus, J., Garofolo, J., & Pajot, S. (2002). A practical introduction to ATLAS. In *Proceedings of the 3rd LREC Conference*, pp 1928–1932.

Sætre, R., Yoshida, K., Miwa, M., Matsuzaki, T., Kano, Y., & 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*, p 33.

Váradi, T., Krauwer, S., Wittenburg, P., Wynne, M., & Koskenniemi, K. (2008). CLARIN: Common language resources and technology infrastructure. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, pp. 1244-1248.

Wang, X., Tsujii, J., & Ananiadou, S. (2010). Disambiguating the Species of Biomedical Named Entities Using Natural Language Parsers. *Bioinformatics, 26*(5), 661-667.