A Data Sharing and Annotation Service Infrastructure

Stelios Piperidis, Dimitrios Galanis, Juli Bakagianni, Sokratis Sofianopoulos
Institute for Language and Speech Processing, Athena R.C., Athens, Greece
{spip, galanisd, julibak, s_sofian}@ilsp.gr

Abstract

This paper reports on and demonstrates META-SHARE/QT21, a prototype implementation of a data sharing and annotation service platform, which was based on the META-SHARE infrastructure. META-SHARE, which has been designed for sharing datasets and tools, is enhanced with a processing layer for annotating textual content with appropriate NLP services that are documented with the appropriate metadata. In META-SHARE/QT21 pre-defined processing workflows are offered to the users; each workflow is a pipeline of atomic NLP services/tools (e.g. sentence splitting, part-of-speech tagging). Currently, workflows for annotating monolingual and bilingual resources of various formats are provided (e.g. XCES, TXT, TMX). From the legal framework point of view, a simple operational model is adopted by which only openly licensed datasets can be processed by openly licensed services.

1 Introduction

Language technology research and development relies on the deployment of appropriate resources and processing services more than ever before. However, the resources and services landscape is unorganized and highly fragmented (Soria et al., 2012). Recently, initiatives like CLARIN (Wittenburg et al., 2010), Language Grid (Ishida, 2011), Panacea (Poch and Bel, 2011), LAPPS Grid (Ide et al., 2014) have been launched aiming at improving discoverability and accessibility of resources and services, as well as their lawful re-use and direct deployment in modern computational environments. In this paper, we present META-SHARE/QT21, a prototype implementation of a linguistic data infrastructure enhanced with linguistic processing services, thus bringing language datasets and processing services together in a unified platform. META-SHARE/QT21 builds upon and extends META-SHARE (Piperidis, 2012). Section 2 briefly introduces the basics of META-SHARE, the underlying data model and the supporting software implementation. Section 3 elaborates on the operations of the new language processing layer and Section 4 presents the assumptions of the current implementation. Finally, in section 5 we conclude and present directions of future work.

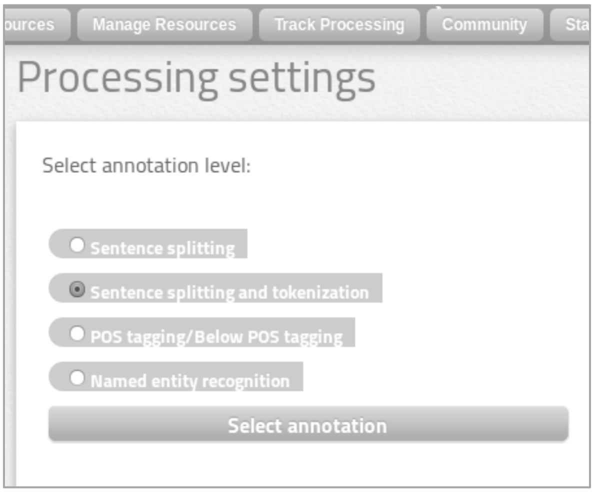
2 META-SHARE design and repository

META-SHARE is designed as a network of distributed repositories of language data, tools and web services, documented with high-quality metadata, aggregated in central inventories allowing for uniform search and access to resources and services. Language resources and services are documented with the META-SHARE metadata schema (Gavrilidou et al., 2012)[1] which builds upon previous initiatives (Broeder et al., 2010), including elements, most of which are linked to ISOCat Data Categories[1], as well as relations (e.g. is_part_of, is_annotation_of) used to describe and link resources that are included in the META-SHARE repository.

Every resource in META-SHARE is primarily assigned to one of the network's repositories, implementing the notion of a master copy of a resource, with the member maintaining the repository undertaking its curation. Metadata records are harvested and stored in the META-SHARE central servers, which maintain an inventory including metadata of all resources available in the distributed network. META-SHARE users, depending on their role, are able to create a user profile, log-in, browse and search

[1] ISO 12620, http://www.isocat.org.

Proceedings of ACL-IJCNLP 2015 System Demonstrations, pages 97–102,
Beijing, China, July 26-31, 2015. ©2015 ACL and AFNLP

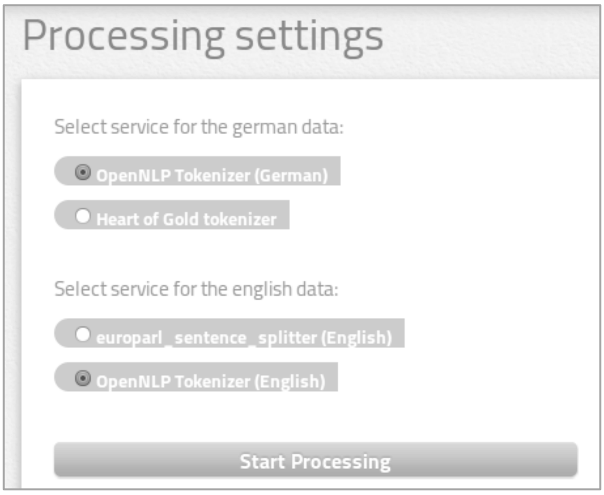


Processing settings

Select annotation level:

○ Sentence splitting
● Sentence splitting and tokenization
○ POS tagging/Below POS tagging
○ Named entity recognition

Figure 1: Dynamically generating annotation levels relevant to a dataset.


Processing settings

Select service for the german data:

● OpenNLP Tokenizer (German)
○ Heart of Gold tokenizer

Select service for the english data:

○ europarl_sentence_splitter (English)
● OpenNLP Tokenizer (English)

Figure 2: Presenting the processing services relevant to the annotation level chosen by the user.

the repository, download resources, upload and document resources etc. All META-SHARE software is open source[2], released under a BSD licence and available at the GitHub repository[3].

3 Enhancing META-SHARE with annotation services

For the purposes of infrastructural projects where META-SHARE was to be used as the language resource sharing platform, notably the CLARIN EL national language infrastructure[4], its functionalities have been extended by providing a mechanism for processing language datasets with appropriate natural language services. The motivation behind this extension is twofold: a) to make language processing services accessible to and usable by a wide range of users (e.g. linguists, lexicographers, social sciences and digital humanities researchers), relieving them from the burden of the technical details of running the tools or services, and b) to bring these tools and services together in a unified platform and facilitate their combination with language datasets, thus paving the way towards the organic growth of the data infrastructure.

Language processing tools are documented with the appropriate metadata in the enhanced repository version (META-SHARE/QT21)[5], and are provided as web services through the language processing (LP) layer. The LP layer has been implemented in Java, based on the Apache Camel framework[6], an open-source project that provides libraries, which enable the easy integration of different technologies and the creation of data processing workflows[7]. For example, Camel offers ready-to-use components/connectors for a) reading the files of a directory b) splitting/aggregating their contents (e.g. txt or XML) into chunks c) forwarding the chunks to data processors d) writing final results to disk.

In the typical scenario that we propose to demonstrate, when a registered META-SHARE/QT21 user selects to process a resource, a list of all available annotation levels (Figure 1) is provided. Then all the available tools/services that correspond to the chosen level are presented (Figure 2). Annotation services can be atomic or composite (a.k.a. workflows) and include: tokenization, sentence splitting, POS tagging, lemmatization, dependency parsing, named entity recognition, and parallel text alignment. As soon as the user selects a service (Figure 2), the META-SHARE/QT21 application consults its database. If the user requests to process a dataset with a specific service, and this dataset has already been processed by the specific service, then the system will forward the user to the processed dataset that has been created and stored in the repository.

[2] The META-SHARE software has been developed using the Django framework, a Python-based web framework, PostgreSQL database and Apache web server. META-SHARE comes with a pre-configured Apache Solr server used to index the META-SHARE database for browsing and searching.

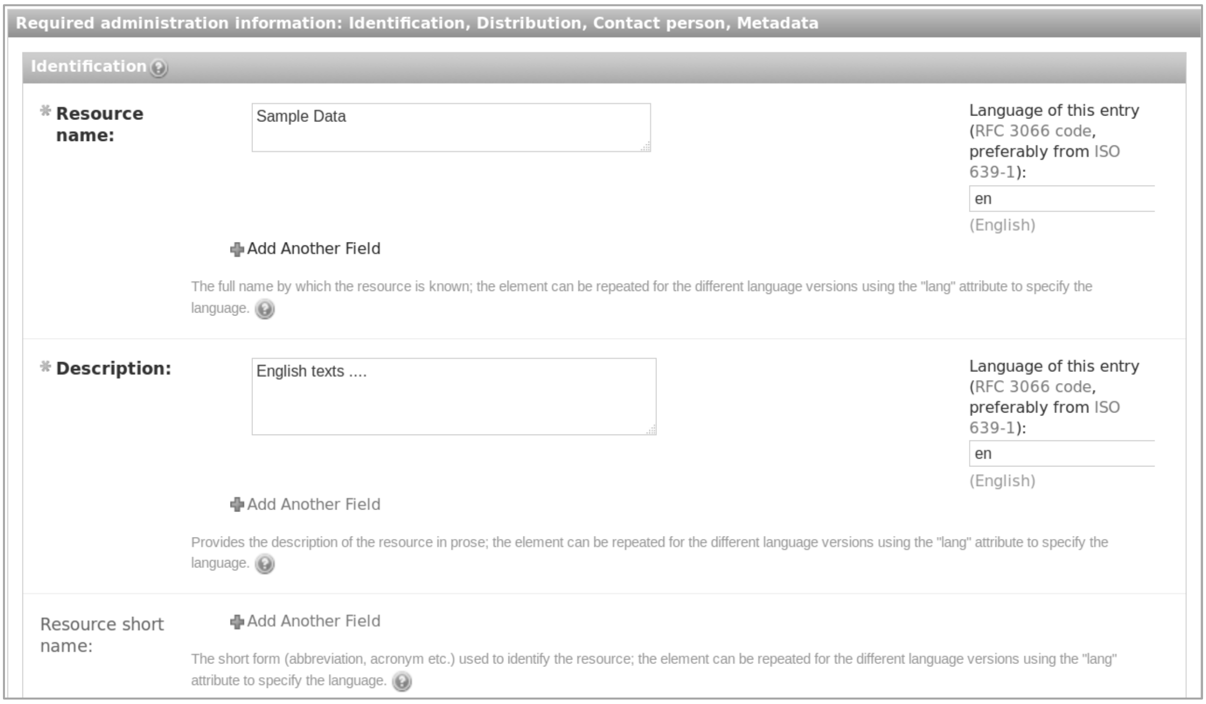
[3] https://github.com/metashare/META-SHARE

[4] http://www.clarin.gr/, http://inventory.clarin.gr

[5] http://qt21.metashare.ilsp.gr/

[6] http://camel.apache.org/

[7] The implemented LP layer is bundled as a web application and can be deployed in a standard java-based web container.



Required administration information: Identification, Distribution, Contact person, Metadata

Identification

* Resource name: Sample Data

Language of this entry (RFC 3066 code, preferably from ISO 639-1): en (English)

Add Another Field

The full name by which the resource is known; the element can be repeated for the different language versions using the "lang" attribute to specify the language.

* Description: English texts ....

Language of this entry (RFC 3066 code, preferably from ISO 639-1): en (English)

Add Another Field

Provides the description of the resource in prose; the element can be repeated for the different language versions using the "lang" attribute to specify the language.

Resource short name:

Add Another Field

The short form (abbreviation, acronym etc.) used to identify the resource; the element can be repeated for the different language versions using the "lang" attribute to specify the language.

Figure 3: Describing/uploading user-owned datasets

Otherwise, META-SHARE/QT21 sends the user request to the LP layer. When the LP gets the request, it starts to process the specified resource by invoking the appropriate tools; when it finishes it notifies the META-SHARE/QT21 application so that the result of the processing is added to the META-SHARE/QT21 repository along with appropriate metadata. Finally, the META-SHARE/QT21 application sends the user an email with the link to the newly created resource. LP's workflows are implemented based on a variety of natural language processing services. These services run either within the LP application environment (loc), or they are accessed via services (rmt). Currently, OpenNLP[8] services (loc) are deployed for English, German and Portuguese, Panacea-DCU[9] services (rmt) for English, LX-Center/University of Lisbon[10] services (rmt) for Portuguese, Heart of Gold (HoG) services[11] (rmt) for German, ILSP NLP[12] services (loc) for Greek, and HunAlign (Varga et al., 2005) text alignment services for aligning parallel corpora at sentence level (loc).

Each set of workflows forms an acyclic directed graph (tree) where each node corresponds to a processing service/tool (e.g. Figure 4). The processing of a data chunk is performed by following a path in such a workflow tree. For example, in case the input is raw text the starting point is the root of the tree. However, LP is also capable of processing already annotated resources thus saving processing time and resources; i.e., if the user requests to process a dataset at a level L (e.g. OpenNLP chunking), and the resource has already been processed at a level A that is a prerequisite for L (e.g. Open NLP Tokenization), then the process will start from the already existing level A annotated resource. Also, the system is aware of what annotation levels make sense and therefore are available for an already processed resource and presents the corresponding choices (e.g. a POS-tagged corpus can be parsed or chunked, but not tokenised) to the user via the web interface (as in Figure 1).

Currently, LP implements services and workflows that can process a) monolingual resources in raw text as well as XCES format and b) bilingual resources in TMX, MOSES, and XCES formats. Bilingual resources, essentially parallel corpora, are split into their language specific parts and monolingual processing services are invoked for each language side.

The resources are stored in the META-SHARE/QT21 repository in a compressed format (e.g. .zip, tar.gz, .gz). Before processing starts, META-SHARE/QT21 decompresses the speci-

[8] https://opennlp.apache.org/

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

[10] http://lxcenter.di.fc.ul.pt/tools/en/

[11] http://heartofgold.dfki.de/

[12] http://nlp.ilsp.gr

fied resource file and then uses an appropriate reader that splits the content of the extracted files in smaller text (data) chunks, so that any file size constraints that a service might have can be met. These chunks are then forwarded to the appropriate processing service/workflow. As soon as the META-SHARE/QT21 has completed the data processing a symmetric procedure collects the resulting (annotated) data chunks and merges them in a single compressed resource.

Additional features of the implemented infrastructure include: a) mechanisms for automatically creating the metadata records of the newly generated datasets, as a result of processing using an annotation service or workflow, b) discoverability of processing services for a particular language and annotation level by simple or faceted search, c) describing and uploading of user-owned datasets up to a size limit (in compressed format) depending on the user's status (Figure 3), d) temporarily storing user-owned processed datasets for 48 hours and deleting them afterwards, unless the user decides to publicly share them, e) checking processed resources for potential errors (e.g. number of files processed as expected), f) monitoring progress of all processing requests and using mechanisms to prevent the application from hanging when waiting for a service response, g) automatically cancelling processing requests that either hang for a long period (e.g. due to network connectivity problems) or are not executed correctly (e.g. when the encoding or the format is not compatible with a service/tool) h) concurrently executing several workflows or parts of a workflow.

3.1 META-SHARE/QT21 user evaluation and scalability tests.

Initially, we conducted a set of user tests which aimed at spotting bugs; then we assessed the stability and usability of META-SHARE/QT21 by asking 15 users to complete a list of 8 annotation tasks for resources of various formats and languages. All testers were researchers and they managed to locate or create the needed resources, submit their requests and receive the annotation results within a few hours without problems. Completion times varied depending on the requested task.

In addition, we assessed the performance and scalability of the LP application by testing it with resources of various lengths depending on the workflow.

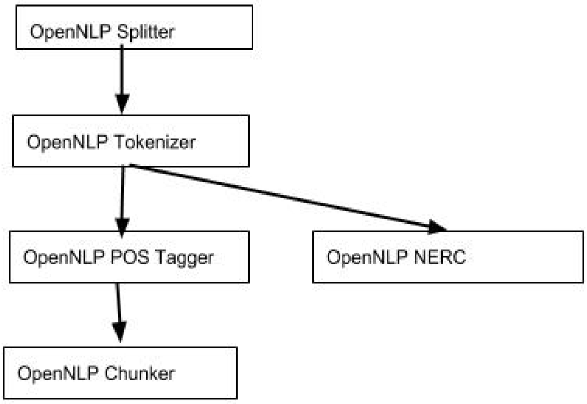


Figure 4: Workflow tree for the English OpenNLP tools.

Locally running services (tools that run within our application) were tested with resources of 1MB, 10MB and 50MB. Remote services were tested with smaller resources of 500KB, 5MB and 10MB. First, each tool/service was tested separately (not concurrently) so as to assess its processing efficiency. Then, we initiated concurrent workflows. All performed tests, concurrent or not, were completed successfully generating the expected output, with the processing times of all growing linearly with resource size; (Figure 5). The tests have also shown that LP application can handle in parallel at least 4 workflows that process ~200MB of data. We plan to handle the processing overload that can be generated by multiple user request for large datasets by using multiple instances of the Camel-based LP in a distributed environment (e.g. Hadoop) in which processing will be carried out in parallel.

4 Assumptions and limitations

Currently, each META-SHARE/QT21 workflow chains together components or services of the same suite of tools, e.g. OpenNLP or the Panacea/DCU services. To accommodate cases where the services deployed belong to different suites, we have developed the appropriate converters. For example, in a UIMA-based tree, where a GATE-based Named Entity Recogniser (NER) is integrated in the respective NER workflow, the UIMA output of the processing services preceding named entity recognition is converted to the GATE format and is fed to the GATE-compatible NER (e.g. Tokenizer → Splitter → POS-Tagger → UIMA-GATE Converter → NER).

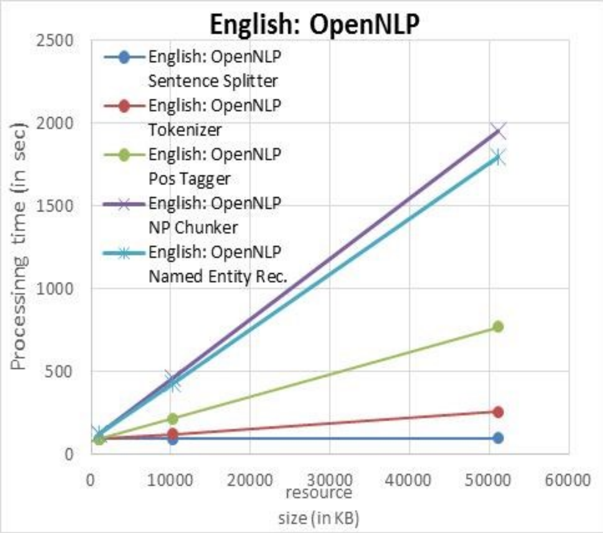


Figure 5: Plot of processing times over resource size for all local English services

Enabling the user to define and deploy custom workflows, cross-suite or not, is on our agenda for the immediate future. The implementation of cross-suite workflows requires the development of several data format converters for each pair of different technologies (e.g. UIMA-GATE, OpenNLP-Panacea/DCU). There are several performance, compatibility and interoperability issues that arise in such cases and have to be investigated and addressed, especially in the light of Language Grid and LAPPS Grid developments (Ide et al., 2014).

Last, but not least, considering the experimental META-SHARE/QT21 repository operations from the legal framework point of view, we have adopted a rather simple operational model by which only openly licensed, with no no-derivatives restriction, datasets can be processed by openly licensed services and workflows. In future versions, in collaboration with other infrastructure providers, we intend to elaborate on a business logic that will allow processing of otherwise licensed datasets and services supporting the appropriate business models.

5 Conclusions and Future Work

The demonstration presented META-SHARE/QT21, a data sharing and annotation service infrastructure. META-SHARE/QT21 is based on META-SHARE, an open-source data infrastructure platform and a language processing layer. The latter is implemented using a widely used integration framework which enables easy creation of data workflows by providing appropriate mechanisms and components for gluing different technologies, services and data sources (XML, txt, TMX). This capability is very useful in a data processing platform, since there is a) an abundance of NLP and machine learning tools implemented (or offered) using different technologies and libraries (e.g. UIMA, GATE, SOAP services, etc.) and b) a variety of data formats (e.g. XCES, TMX). The user evaluation that we conducted has shown that META-SHARE/QT21 can be easily used by NLP researchers for obtaining annotations on a set of resources of various formats. Also a set of stress tests that we conducted revealed that the LP layer can process concurrently a significant amount of data. We are now investigating how data annotation can run on multiple machines in a distributed environment (e.g. Hadoop clusters), thus enabling the processing of large volumes of data.

Acknowledgments

This paper presents work done in the framework of the projects T4ME (GA no. 249119), QTLaunchPad project (GA no. 296347), funded by DG INFSO of the European Commission through the FP7 and ICT-PSP Programmes, and maintained as well as further extended in the framework of the Greek CLARIN Attiki project (MIS 441451), Support for ESFRI/2006 Research Infrastructures, of the Greek Government.

We are grateful to the NLX-Natural Language and Speech Group of the University of Lisbon and its head, Antonio Branco, for kindly providing access to the LX-Center services.

References

Soria, C., Bel, N., Choukri, K., Mariani, J., Monachini, M., Odijk, J., Piperidis, S., Quochi, V., Calzolari, N. (2012). The FLaReNet Strategic Language Resource Agenda. Proceedings of the 8th Language Resources and Evaluation Conference (LREC'12), ELRA.

Gavrilidou, M.; Labropoulou, P.; Desypri, E.; Piperidis, S.; Papageorgiou, H.; Monachini, M.; Frontini, F.; Declerck, T.; Francopoulo, G.; Arranz, V. and Mapelli, V. (2012). The META-SHARE Metadata Schema for the Description of Language Resources. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis (Eds), Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), 23-25 May, Istanbul, Turkey. European Language Resources Association (ELRA).

Wittenburg, P., Bel, N., Borin, L., Budin, G., Calzolari, N. Hajicova, E. Koskenniemi, K., Lemnitzer, L., Maegaard B., Piasecki, M., Pierrel, J.M., Piperidis, S., Skadina, I., Tufis, D., Veenendaal, R.v ., Váradi, T., Wynne, M. (2010). Resource and Service Centres as the Backbone for a Sustainable Service Infrastructure. Proceedings of the 7th Language Resources and Evaluation Conference (LREC'10), ELRA.

Ishida, T. (Ed) (2011). The Language Grid. Service-Oriented Collective Intelligence for Language Resource Interoperability, Springer

Poch, M., Bel, N. (2011) Interoperability and technology for a language resources factory Workshop on Language Resources, Technology and Services in the Sharing Paradigm – IJCNLP 2011.

Piperidis, S. (2012). The META-SHARE language resources sharing infrastructure: Principles, challenges, solutions. In Proceedings of LREC-2012, pages 36–42, Istanbul, Turkey.

Ide, N., Pustejovsky, J., Cieri, C., Nyberg, E., Wang, D., Suderman, K., Verhagen, M., and Wright, J. (2014) The Language Application Grid. Proceedings of the 9th Language Resources and Evaluation Conference (LREC'14), ELRA

Broeder, D., Kemps-Snijders, M., Van Uytvanck, D., Windhouwer, M., Withers, P., Wittenburg, P. and Zinn, C. (2010). A Data Category Registry- and Component-based Metadata Framework. Proceedings of the 7th Language Resources and Evaluation Conference (LREC'10), ELRA.

Varga, D., Németh, L., Halácsy, A., Kornai,, P., Trón, V., Nagy, V. (2005). Parallel corpora for medium density languages. In Proceedings of the RANLP 2005, pages 590-596.