Processing, Analysing and Visualising Language Data using solutions prepared in CLARIN-PL LTC

Tomasz Walkowiak
CLARIN-PL
Wroclaw University of Science and Technology, Poland
tomasz.walkowiak@pwr.edu.pl

Marcin Pol
CLARIN-PL
Wroclaw University of Science and Technology, Poland
marcin.pol@pwr.edu.pl

Maciej Piasecki
CLARIN-PL
Wroclaw University of Science and Technology, Poland
maciej.piasecki@pwr.edu.pl

Abstract

The paper presents a functionalities of CLARIN-PL Language Technology Centre (LTC). LTC uses technology infrastructure developed by CLARIN project which is a complex computer system that enables combining language tools with language resources into processing workflows. The processing tools are developed as microservices connected to each other using solutions like RabbitMQ broker[1] and Citrix Server Virtualization. Researchers use complete workflow submitting data using modified version of DSpace[2] or NextCloud[3] repository system.

Keywords: language technology infrastructure, natural language processing, microservice, CLARIN-PL

1. Introduction

CLARIN-PL Language Technology Centre (henceforth LTC, http://clarin-pl.eu) is the Polish node of the CLARIN ERIC (http://clarin.eu) Language Technology (LT) research infrastructure and a typical CLARIN B-type centre. We have started gradual expansion of the basic blue-print of the B-type centre towards supporting a digital research paradigm in broadly understood language data analysis. That follows the proposed bi-directional model for the development of the LT research infrastructure (Piasecki, 2014) in way which combines integration of the available LT elements with the construction of research applications according to the user's requirements. CLARIN offers easy access to ready-to-use language resources and tools via Internet (e.g. as Web Services and simple Web Applications), but also develops web based research applications (Wittenburg et al., 2010). They allow users to process data online without the need (or at least very reduced) to bother with technicalities. As a result, the constructed CLARIN-PL research infrastructure (as a part of CLARIN ERIC distributed system) promotes open, centralized workflows that support and encompass different aspects and products of the research lifecycle, including developing research ideas, designing a study, storing and analysing collected data. The paper is structured as follows. We start with the overview of the construction of LTC. It is followed by a description of Natural Language Processing (NLP) microservices that can be orchestrated into complex processing workflows that allow constructing flexible research applications – shortly described in chapter 4. Next, selected LTC infrastructure elements, including research web applications are presented.

2. Functionality according to user's Requirements

The CLARIN-PL research infrastructure is built in response to the users demands to help researchers from the areas of Humanities and Social Sciences in their work with tools originating from Natural Language Engineering, e.g. for corpus analysis, lexicography, stylometry, text mining, information extraction, statistical semantic analysis, etc. LTC provides web based software which helps users in their work on the previously developed digital archives and corpora, but also to build new resources from raw texts or textual data in other formats.

We offer solutions which do not require installation on the user's computer and do not require from the user skills in programming. Moreover, we also try to minimise the amount of knowledge from NLE expected from the user. Researchers can upload their files to a digital repository which is based on an adapted and expanded version of the D-Space system to make their resource and and projects publicly available (at least on the level of meta-data, as the data can be exceptionally kept private. We also introduced recently a possibility of storing the data in an instance of the NextCloud system to make the users' projects private, e.g. during their developemnt. All resources stored in D-Space receive a unique, persistent identifier (PID). Next, the resources can be directly processed by applications built by CLARIN-PL or by selected open solutions installed in LTC like Kontext which is NoSketch Engine. Tools or applications that are appropriate for a resource are chosen by the user

[1] https://www.rabbitmq.com/
[2] http://www.dspace.org/
[3] https://nextcloud.com/

depending on his needs, but many corpora, especially Polish, can be processed by the built-in workflows.

2.1. CLARIN-PL Language Technology Centre

With the core functionality of the LTC researchers have the possibility to create and develop NLP projects. All tools and applications are inter-connected to give users ability to set up a project for a particular research task or a specific experiment.

To use resources and tools of LTC users have to set up a free account in our D-Space repository or login via federation identity using *shibboleth* – a system of distributed authorisation. Once logged, users can upload their files to one of the storage systems: NextCloud or D-Space. After that the data can be processed by NLP services described in next chapter and pushed to different CLARIN-PL applications. They have been primarily developed to support researchers and students from H&SSs, but also can be found in other areas of science and technology.

Applications like *WebSty* (a web-based system for stylometric analysis, e.g. text similarity and clustering) or *NoSketch Engine* instance (language corpus management and query system) show the processed data to researchers in a convenient form. Applications like *Inforex* which is a system designed for managing and annotating text corpora or *MeWeX* (a system for collocation extraction) enable upload the produced results of the analysis back to D-Space or NextCloud installations.

Fig. 1. presents the data flow between several selected NLP Services, applications and storage systems.

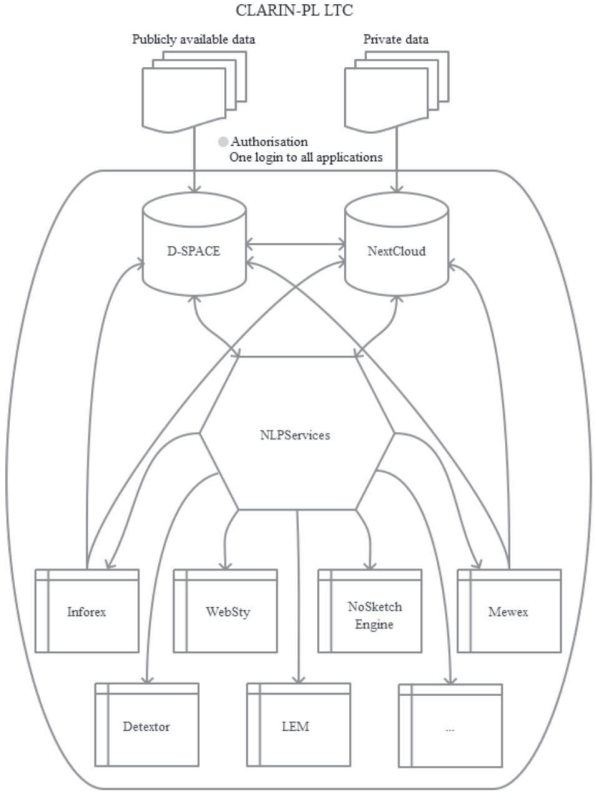

Fig. 1. CLARIN-PL data flow between selected NLP Services, applications and storage systems.

3. NLP Micro-services

Construction of a multi-user, web system that runs natural language processing (NLP) and machine learning (ML) tools poses several problems related with the system availability and performance. The system should be scalable, responsive and available all the time. Language tools have excessive CPU and memory consumption. In addition, a number of users or tasks at a given time is unpredictable. Moreover, tools are developed in different technologies (C++, Python, Java, R, Julia). They use a variety of different data formats that are very often not compatible with each other.

Many of NLP tools (like a Named Entity Recognizer or a Word Sense Disambiguation tool) have knowledge models of large size. In such cases, the time required for loading a model into operating memory is much longer than processing of a single text file. This can be avoided by running a tool as a service with pre-loaded data models kept later in memory. Each service is run as a separate process. The usage of services communicating with others by lightweight mechanisms solves also a problem of the variety of the programming languages used by NLP and ML tools, as there is no need for tight integration.

Therefore, we have built the LTC architecture in way following microservices architecture proposed, e.g., by (Bell, 2010). Each NLP and ML tool is run as a microservice. The AMQP[4] protocol was used for lightweight communication. Scalability is achieved by using a central broker with a queuing system (open source RabbitMQ was used). Each NLP and ML tool has its own queue. The NLP and ML microservices collect tasks from the given queue, process data and send back messages when results are available. A task is defined as a triple: path of the input file (or directory), parameters (in JSON format) and output path. All microservices have access to the same network file system used for storing input/output files. The most required or most frequently used NLP and ML microservices have to be run in several instances since a queuing system acts as an effective load balancer.

The research web applications expect NLP and ML tools to be run in a specific order. Very often this not a simple chain of tools but a workflow of tools (Walkowiak and Piasecki 2015). Therefore, we have developed (Walkowiak, 2018) a human readable orchestration language, cf (Peltz, 2003), that allows for describing different text processing tasks. It is called Language Processing Modelling Notation (LPMN). An exemplar LPMN statement is presented in Fig. 2.

Each microservice is defined by its name (for example: any2txt, spacy, fextor in Fig. 2). Data could be loaded from different sources, like local files (downloaded be a sparate service), zip files, urls, dSpace and Next Cloud repository defined by separate LPMN statements (for example urlzip in Fig. 2).

[4] https://www.amqp.org

```
urlzip("http://ws.CLARIN-pl.eu/public/teksty/en.zip")|any2txt|spacy({"annotate_entities":true})|fextor({"features":"base pron_count aux_count conj_count det_count intj_count num_count"})|dir|featfilt({"similarity":"ratio","weighting":"all:sm-mi_simple"|cluto({"no_clusters":2,"analysis_type":"plottree"})
```

Fig. 2. Exemplar LPMN

In the proposed architecture (Fig. 3) additional server grants the access from the Internet. It works as a proxy for the core system delivering REST API. Such an approach allows for easy integration with almost any kind of applications. In addition, the engine for running workflows described in LPMN was developed. It allows to process large corpus of text in a batch like mode. It is developed in asynchronous architecture. For each LPMN statement it has a single thread running an event loop and waiting for incoming events. Events are emitted by RabbitMQ broker that facilitates communication between all system components. The LPMN engine has a priority mechanism that prevents the large corpora from blocking the system. Files with a large size and from corpora consisting of a large number of texts have much lower priority then other files. It allows to effectively use all instances of a given tool but does not block the system for other users.

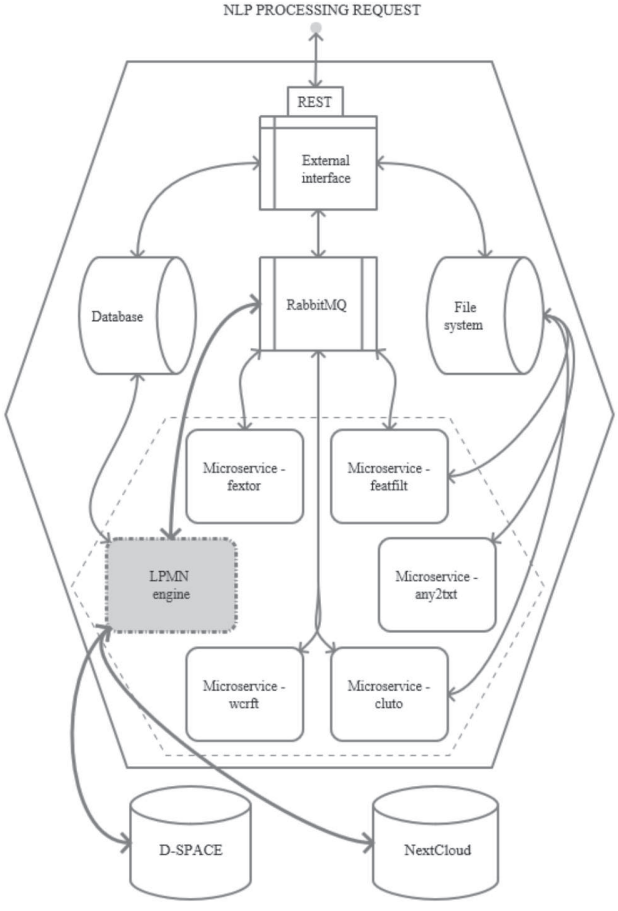

Fig. 3. System architecture

There are more than 70 mircoservices available in the current system[5]. The number of tools grows instantly. The most used tools includes:

- `any2txt` – converting documents (doc, docx, rtf, pdf, etc.) to UTF-8 texts,
- `makezip` – compressing files,
- `convert` – conversion of NLP formats,
- `wcrft2` – tagger for Polish ,
- `spacy` – tagger and name entity recognizer for English and German[6],
- `fextor` – tool for feature extracting (calculating frequency of morphological and grammatical features),
- `featfilt` – tool for feature weighting and calculation of similarities,
- `cluto` – clustering[7] of multidimensional vectors,
- `mail` – sending e-mails to users,
- `spejd` – partial parsing and rule-based morphosyntactic disambiguation[8]
- `maltparser` – a tool for data-driven dependency parsing[9]

Moreover, the system includes a database, accessed by a separate microservice in the asynchronous mode. It is used for logging processing tasks and synchronization with external applications.

The solution is very flexible. It allows a fast development of a new web based application and a simple addition of a new NLP or ML tool. A typical web based application is built in a Single Page Application style. It consists of a web GUI that allows a user to select the source of corpus and data processing options specific for each tool. Selected options are serialized to the LPMN statement which is stored in the LTC. Next, the id of the results is sent back to the application. The application downloads results and visualize them. Adding a new tool requires a construction of a new microservice and naming it. After the first run in CLARIN-PL cloud the new tool is automatically available in LPMN statements. Construction of a new mircoservice is simplified by existence of ready-to-use skeleton code (Java, C++ and Python) and set of examples. The developer has to build a new class that implements three virtual methods: two for initialization and one for the processing. The libraries implement multithreading (Java, C++) or multiprocessing (Python). It allows (if it is possible for a given tool) to run several instances of a tools with shared models. Inside LTC application workflows (described in LPMN) are set by system administrators, but users with programing skills can create them by themselves.

[5] http://ws.clarin-pl.eu
[6] https://spacy.io
[7] http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview
[8] http://zil.ipipan.waw.pl/Spejd
[9] http://zil.ipipan.waw.pl/PolishDependencyParser

4. Applications

The CLARIN-PL LTC provides a large set of online applications that can be accessed from the main web page: http://clarin-pl.eu. The most used and sophisticated research applications are WebSty, LEM, Inforex, MeWeX and WoSeDon.

WebSty is an open, stylometric system with web-based user interface. It allows for analysing similarities between texts based on automatically extracted language features. It is used to identify groups of texts that exhibit subtle similarities hidden to the naked eye but traceable by multidimensional statistical techniques. A typical use cases are: authorship attribution, literary style analyses or author's characteristic analysis (Piasecki et al. 2017a). The tool offers several visualisation methods (see Fig. 4, 5) and techniques for the extraction of characteristic features. WebSty was designed for Polish texts, next it was extended to English and now a multilingual version is under development.

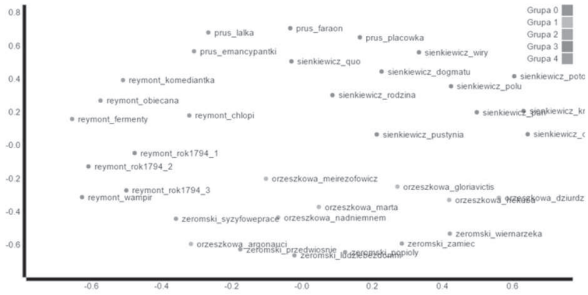

Fig. 4. WebSty results. Text similarities in the form of 2D plot with a usage of multidimensional scaling technique

Literary Exploration Machine (LEM) [10] provides a virtual research environment for textual scholars, allowing them to upload texts in Polish and either explore them with a suite of dedicated tools or transform them into another format (text, table, list). This latter functionality allows for further processing with other tools (including those built for the English language). LEM brings together the already existing applications developed by CLARIN-PL and supplements them with new functionalities. The main features include (Fig. 6): lemmatization, part-of-speech tagging, generating custom wordlists and lemmatized texts. (Piasecki. et al. 2017b)

Fig. 5. WebSty results. Text similarities in the form of a heatmap

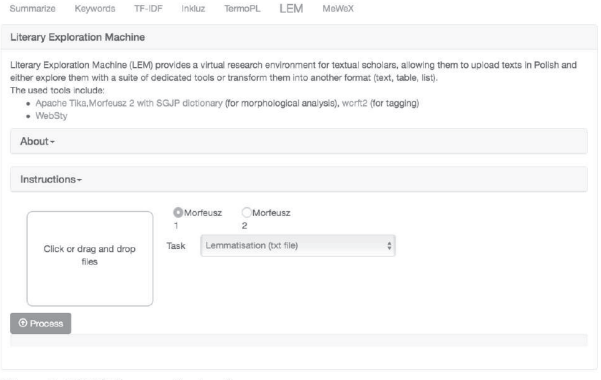

Fig. 6. LEM user interface

Inforex[11] is a system for collaborative text corpora annotation and analysis. It was developed to construct corpus-based linguistic resources for various tasks in the field of natural language processing, but it is also used by the scientist for other purposes. It is integrated with a digital repository for storing and publishing language resources (D-Space). Inforex supports manual text annotation on the semantic level, e.g. annotation of Named Entities (NE), anaphora, Word Sense Disambiguation (WSD) and relations between named entities. The system also supports manual text clean-up and automatic text pre-processing including text segmentation, morphosyntactic analysis and word selection for WSD annotation.

MeWeX[12] is web based system for extraction of multiword expressions (collocations).

WoSeDon[13] is a tool for Word Sense Disambiguation. It works for polish texts and as a source of possible senses using plWordNet[14].

[10] ws.clarin-pl.eu/lem.shtml

[11] inforex.clarin-pl.eu

[12] https://mewex.clarin-pl.eu/login

[13] http://wosedon.clarin-pl.eu/

All applications are being gradually developed thanks to a constructive feedback from and often collaboration with the researchers in Humanities and Social Sciences.

5. Infrastructure

The LTC have to be scalable, responsive and available all the time. Language tools have excessive CPU and memory consumption and a number of users or tasks at a given time is unpredictable.

According to above demands CLARIN-PL LTC hardware consists of nine inter-connected servers in a mixed rack/blade architecture. To allow big data analysis servers have from 192 to 224 GB of RAM, which gives a total of almost 2 TB. The power of 324 processes in parallel are supported by Intel (R) Xeon (R) CPUs E5-2665@2.40GHz, as there are up to 16 threads per processor. Data are submitted to data storage subsystem which is using RAID10 volumes based on IBM Storwize V7000. High availability access to data is provided by redundant dual-active intelligent FC 8Gb controllers and dual-active iSCSI controllers. All data is protected by backup with deduplication mode.

Computation processes and data are protected by uninterruptible power supply which allows to run all component systems without external power source for almost 30minutes. To make the LTC system easily scalable servers are managed by XENServer[15] that allows to run and manage a large number of virtual machines. Virtualization makes the LTC management more convenient and efficient. Operating systems are independent from the hardware in the virtual environment, so they can be easily moved to another server as a reaction to any failure or resource shortage. Moreover, virtualization give us possibilities to run different operating systems in parallel that is important, as LT tools are developed in different technologies and often require different versions of the systems.

Using XENServer virtual machines solution provides a disaster recovery mechanism ensuring that when a virtualized system crashes, it will be restored as quickly as possible. The resources (memory, CPU, disk) could be attached to any machine on demand and changed according to needs.

5. Conclusions

CLARIN-PL LTC focuses on research tools based on LT. The achieved high degree of flexibility means that it will be possible to easily customize projects to fit a variety of needs, from small ones to large research collaborations. Researchers can process large corpora of text and easily publish or share results. So far, CLARIN-PL is focused mainly on Polish, but most of the applications are able to work with English and German. We will continue expanding our tools to support multi-linguality to much broader extent.

[14] http://plwordnet.pwr.wroc.pl/wordnet/ - a large network of words

[15] https://www.citrix.com.pl/products/xenserver/

Acknowledgements

Works funded by the Polish Ministry of Science and Higher Education within CLARIN-PL Research Infrastructure.

References

Piasecki, M. (2017a) Open Stylometric System WebSty: Towards Multilingual and Muiltipurpose Workbench

Piasecki, M. (2014) User-driven Language Technology Infrastructure -- the Case of CLARIN-PL. In Proceedings of the Ninth Language Technologies Conference, Information Society - IS 2014, Ljubljana, Slovenia, 2014. URL: http://nl.ijs.si/isjt14/proceedings/isjt2014_01.pdf

Hinrichs, E., Hinrichs, M., Zastrow, T. (2010). WebLicht: Web-based lrt services for german. In Proceedings of the ACL 2010 System Demonstrations, ACL Demos '10, pages 25–29, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sedlák, M. (2014). Treex::web. In LINDAT/CLARIN. http://hdl.handle.net/11858/00-097C-0000-0023-44AFC. 3.2.

Ogrodniczuk, M., Lenart, M. (2013). A multi-purpose online toolset for NLP applications. LNCS, vol. 7934, pp. 392–395. Springer-Verlag, Berlin, Heidelberg

Walkowiak, T. (2018). Language Processing Modelling Notation – orchestration of NLP microservices. In: Advances in Dependability Engineering of Complex Systems: Proceedings of the Twelfth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, July 2 - 6, 2017, Brunów, Poland, Springer, pp. 464–473

Wittenburg, P. et al. (2010) Resource and service centres as the backbone for a sustainable service infrastructure. In N. Calzolari et al., Proceedings of LREC 2010, Malta, pp. 60–63. ELRA.

Walkowiak T. (2015) Web Based Engine for Processing and Clustering of Polish Texts. In: Zamojski W., Mazurkiewicz J., Sugier J., Walkowiak T., Kacprzyk J. (eds) Theory and Engineering of Complex Systems and Dependability. Advances in Intelligent Systems and Computing, vol 365. Springer pp. 515–522

Walkowiak, T., Pol, M. (2017). The impact of administrator working hours on the reliability of the Centre of Language Technology. Journal of Polish Safety and Reliability Association, 8(1): 167-173

Dragoni N., Giallorenzo S., Lluch-Lafuente A., Mazzara M., Montesi F., Mustafin R., Safina L.(2016): Microservices: yesterday, today, and tomorrow, CoRR, vol abs/1606.04036

Peltz, Ch. (2013). Web services orchestration and choreography. Computer, vol. 36, no. 10, pp. 46–52