A Multi-purpose Online Toolset for NLP Applications*

Maciej Ogrodniczuk and Michał Lenart

Institute of Computer Science, Polish Academy of Sciences

Abstract. This paper presents a new implementation of the multi-purpose set of NLP tools for Polish, made available online in a common web service framework. The tool set comprises a morphological analyzer, a tagger, a named entity recognizer, a dependency parser, a constituency parser and a coreference resolver. Additionally, a web application offering chaining capabilities and a common BRAT-based presentation framework is presented.

1 Introduction

The idea of making a linguistic toolset available online is not new; among other initiatives, it has been promoted by CLARIN[1], following its aspirations for gathering Web services offering language processing tools [3] or by related initiatives such as WebLicht.

The first version of a toolset for Polish made available in the Web service framework has been proposed in 2011, and called *the Multiservice* [2]. Its main purpose was to provide a consistent set of mature annotation tools — previously tested in many offline contexts, following the open-source paradigm and under active maintenance — offering basic analytical capabilities for Polish.

Since then, the Multiservice has been thoroughly restructured and new linguistic tools have been added. The framework currently features a morphological analyzer *Morfeusz PoliMorf*, two disambiguating taggers *Pantera* and *Concraft*, a shallow parser *Spejd*, the *Polish Dependency Parser*, a named entity recognizer *Nerf* and a coreference resolver *Ruler*.

2 Architecture

The Multiservice allows for chaining requests involving integrated language tools: requests to the Web service are enqueued and processed asynchronously, which allows for processing larger amounts of text. Each call returns a token used to check the status of the request and retrieve the result when processing completes.

* The work reported here was partially funded by the *Computer-based methods for coreference resolution in Polish texts (CORE)* project financed by the Polish National Science Centre (contract number 6505/B/T02/2011/40).

[1] Common Language Resources and Technology Infrastructure, see `www.clarin.eu`

E. Métais et al. (Eds.): NLDB 2013, LNCS 7934, pp. 392–395, 2013.

One of the major changes in the current release is a redesign of the internal architecture with the Apache Thrift framework (see `thrift.apache.org`, [1]), used for internal communication across the service. It features a unified API for data exchange and RPC, with automatically generated code for the most popular modern programming languages (including C++, Java, Python, and Haskell), the ability to create a TCP server implementing such an API in just a few lines of code, no requirement of using JNI for communication across various languages (unlike in UIMA), and much better performance than XML-based solutions.

The most important service in the infrastructure is the Request Manager, using a Web Service-like interface with the Thrift binary protocol instead of SOAP messages. It accepts new requests, saves them to the database as Thrift objects, keeps track of associated language tools, selects the appropriate ones for completing the request, and finally invokes each of them as specified in the request and saves the result to the database.

Since the Request Manager service runs as a separate process (or even a separate machine), it can potentially be distributed across multiple machines or use a different DBMS without significant changes to other components. The service can easily be extended to support communication APIs other than SOAP or Thrift and the operation does not create significant overhead (sending data using Apache Thrift binary format is much less time-consuming than sending XMLs or doing actual linguistic analysis of texts).

Requests are stored in `db4o` — an object oriented database management system which integrates smoothly with regular Java classes. Each arriving request is stored directly in the database, without any object-relational mapping code.

Language tools run as servers listening to dedicated TCP ports and may be distributed across multiple machines. There are several advantages of such architecture, the first of which is its scalability — when the system is under heavy load, it is relatively easy to run new service instances. Test versions of the services can be used in a request chain without any configuration — there is simply an optional request parameter that tells the address and port of the service. Plugging-in new language tools is equally easy — Apache Thrift makes it possible to create a TCP server implementing a given RPC API in just a few lines of code.

3 Usage and Presentation

The tools offer two interchangeable formats, supporting chaining and uniform presentation of linguistic results: TEI P5 XML and its JSON equivalent. The TEI P5 format is a packaged version of the stand-off annotation used by the National Corpus of Polish (NKJP [4]), extended with new annotation layers originally not available in NKJP.

Sample Python and Java clients for accessing the service have been implemented. To facilitate non-programming experiments with the toolset, a simple Django-based Web interface (see Fig. 1) is offered to allow users to create toolchains and enter texts to be processed.

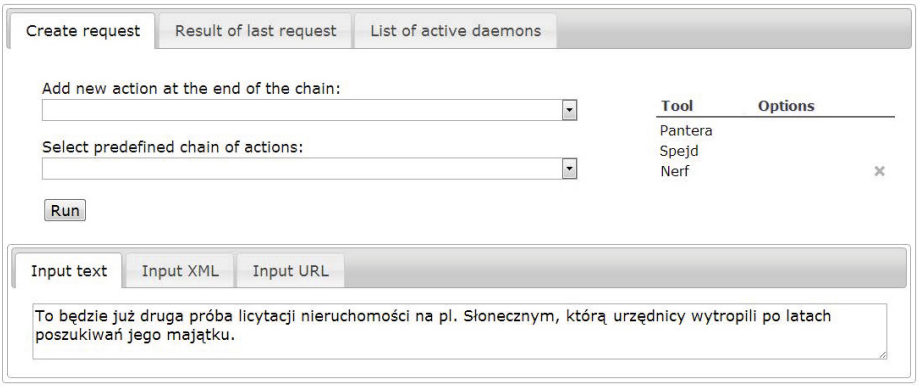

Fig. 1. The Multiservice Web interface

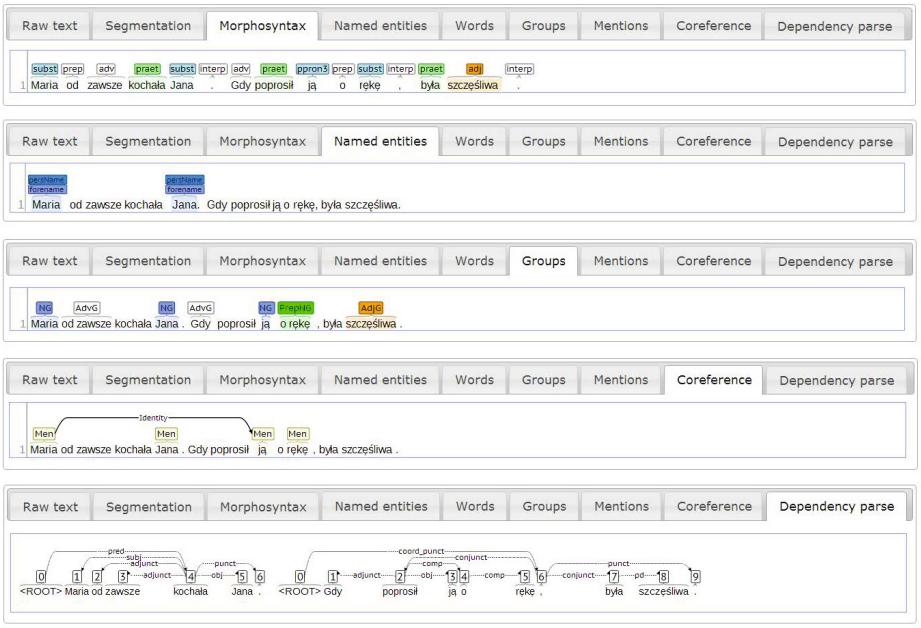

Fig. 2. Different levels of linguistic annotation displayed with BRAT

The application allows for triggering a processing request and periodically checking its status. Upon completion, the result is retrieved and displayed to the user. In the case of failure, an appropriate error message is presented.

The Web Interface[2] features consistent visualization of linguistic information produced with the BRAT tool [5] for all layers made available by integrated

[2] Available at `http://glass.ipipan.waw.pl/multiservice/`

annotators. See Fig. 2 for a selection of annotations produced for an example sentence (*Maria od zawsze kochała Jana. Gdy poprosił ją o rękę, była szczęśliwa.* *'Maria has always loved John. When he asked her to marry him, she was happy.'*). Additional context-dependent linguistic properties of the annotation items (e.g. all morphosyntactic interpretations of a word, not just a disambiguated one) are available at mouseover. A unified framework for linking visualization to other levels of linguistic annotation is also provided and the only necessary implementation step is a conversion of JSON-encoded request results into BRAT internal format.

4 Conclusions

As compared to its offline installable equivalents, the toolset provides users with access to the most recent versions of tools in a platform-independent manner and without any configuration. At the same time, it offers developers a useful and extensible demonstration platform, prepared for easy integration of new tools within a common programming and linguistic infrastructure. We believe that the online toolset will find its use as a common linguistic annotation platform for Polish, similar to positions taken by suites such as Apache OpenNLP or Stanford CoreNLP for English.

References

1. Agarwal, A., Slee, M., Kwiatkowski, M.: Thrift: Scalable cross-language services implementation. Tech. rep., Facebook (April 2007), `http://thrift.apache.org/static/files/thrift-20070401.pdf`
2. Ogrodniczuk, M., Lenart, M.: Web Service integration platform for Polish linguistic resources. In: Proceedings of the 8th International Conference on Language Resources and Evaluation, LREC 2012, pp. 1164–1168. ELRA, Istanbul (2012)
3. Ogrodniczuk, M., Przepiórkowski, A.: Linguistic Processing Chains as Web Services: Initial Linguistic Considerations. In: Proceedings of the Workshop on Web Services and Processing Pipelines in HLT: Tool Evaluation, LR Production and Validation (WSPP 2010) at the 7th Language Resources and Evaluation Conference (LREC 2010), pp. 1–7. ELRA, Valletta (2010)
4. Przepiórkowski, A., Bańko, M., Górski, R.L., Lewandowska-Tomaszczyk, B. (eds.): Narodowy Korpus Języka Polskiego. Wydawnictwo Naukowe PWN, Warsaw (2012)
5. Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., Tsujii, J.: BRAT: a web-based tool for NLP-assisted text annotation. In: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2012, pp. 102–107. Association for Computational Linguistics, Stroudsburg (2012)