TextImager as an interface to BeCalm

Wahed Hemati, Tolga Uslu, and Alexander Mehler

Text Technology Lab,
Goethe Universitt Frankfurt am Main, Germany
{hemati,uslu,amehler}@em.uni-frankfurt.de
http://www.hucompute.org/

Abstract. TIPS (Technical interoperability and performance of annotation servers) is a novel BioCreative task. It focuses on the technical aspects of making NER taggers available as webservices. In this paper, we present the functionality and architecture of BeCalm metaserver integrated into TextImager. We present the integration of in-house developed NER tagger and also the integration and adaption of available NER tagger.

Key words: TextImager, BeCalm, TIPS, NER

1 Introduction

The novel BioCreative task TIPS focuses on making NER taggers available as web servers. We introduced TextImager in [1] where we focused on the architecture and the functions of its backend. In this paper, we present TextImager as an annotation server on the BeCalm platform. TextImager is a UIMA[2]-Based framework that offers a wide range of NLP. It is modular and expendable, due to the underlying UIMA architecture. We made TextImager available online. It can be accessed by BeCalm metaserver requests and is able to answer such requests in the BeCalm TSV format. Currently we trained and implemented 6 NER systems for the BeCalm GPRO and CEMP tasks, namely StanfordNER, MarMot, CRF++, MITIE, Glample and CRFVoter. Each of these tagger can be used in a pipeline together with other NLP-tools that are already integrated in TextImager.

2 Systems description and methods

In this section we will describe the technical implementation of BeCalm metaserver integration into TextImager. We implemented a REST API that provides access and responds to the BeCalm metaserver requests described in http://www.becalm.eu/api

In order to obtain annotation documents the BeCalm metaserver sends a *getAnnotations()* request for a set of documents. The getAnnotations() message is in JSON format, which is exemplified in Listing 1.1. The request contains specification about the documents that need to be processed, authentication information and the *costum_parameter annotator*.

Listing 1.1. Example BeCalm metaserver JSON request

```
{
    "method":"getAnnotations",
    "becalm_key":"141xxx",
    "name":"becalm",
    "custom_parameters":{
        "annotator":"BioGproS"
    },
    "parameters":{
        "expired":"2017-05-02T15:10:00+02:00",
        "documents":[
            {
                "source":"ABSTRACT SERVER",
                "document_id":"12140745"
            },

            {
                "source":"PATENT SERVER",
                "document_id":"EP0959896B1"
            },
            ...
        ],
        "communication_id":"4672794"
    }
}
```

After receiving the *getAnnotations()* request, our server responds to BeCalm with an acknowledge message.

After the acknowledgement, our server executes the internal workflow, which starts by processing the BeCalm JSON request. The retrieved documents are downloaded by the TextImager Rest API from their specific sources, namely *Abstract Server*, *Patent Server* and *PubMed Server*. The documents are then passed to the TextImager backend. TextImagers backend is a UIMA-based framework that offers a range of NLP tools. Every TextImager component is configured as a UIMA-AS service, which may run standalone or in a pipeline (see Figure 2, *Stanford NER service Interface*, *MarMot Service Interface*, etc). All service instances are located on servers. Note that these instances can be distributed among different servers (see Figure 2, *Server 1, Server 2, Server X*). We trained and integrated 6 NER systems for the `BeCalm GPRO` and `CEMP` tasks into TextImager, namely `StanfordNER`[3], `MarMot`[4], `CRF++`[5], `MITIE`[6], `Glample`[7] and `CRFVoter`.

The TextImager Rest API sends the request and the documents to the TextImager Orchestrator. The *custom_parameter: annotator* (see Listing 1.1) is used by the orchestrator to determine which NER service to run. The TextImager Orchestrator selects and acquires components and their resources: it arranges components into pipelines and grants the ability to parallelize them. Each of the 6 integrated NER systems require tokenization, lemmatization and morph tagging as preprocessing steps. Finally after the orchestration step, the documents

are processed by means of the selected service pipeline. After processing is done, the output is passed to the TextImager Rest API, which saves the output to the BeCalm server, by calling the method *saveAnnotations*[1].

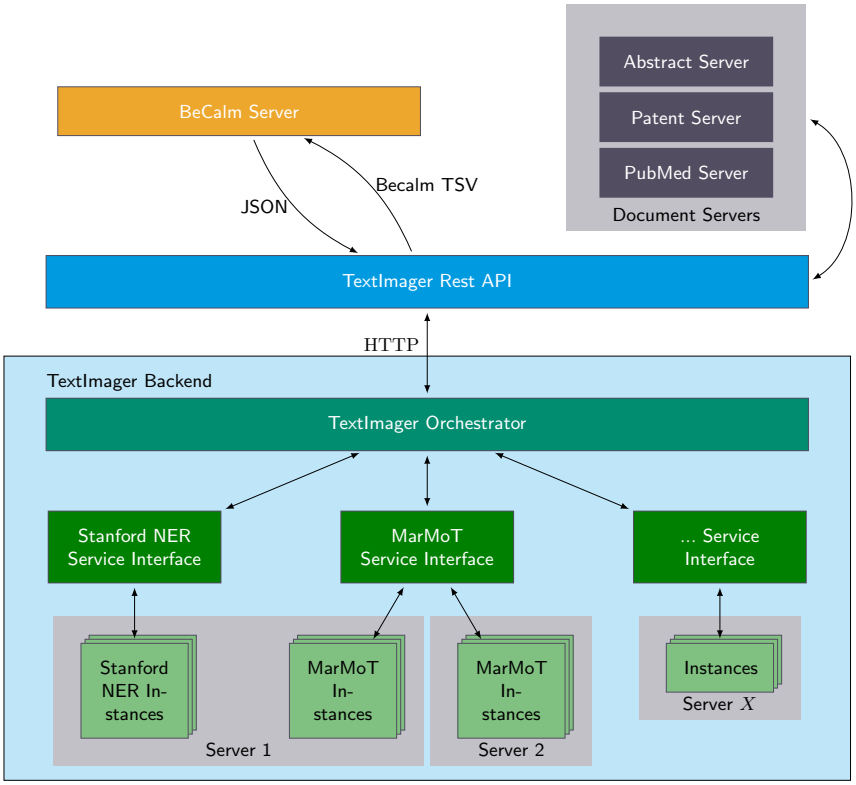

Fig. 1. Architecture of BeCalm metaserver integrated into TextImager

3 Discussion

In this work, we presented TextImager as a Becalm annotation server. Our annotation server is a distributed and modular framework. We currently integrated 6 NER systems for CEMP and GPRO. In future work, we will focus on integrating more NER systems into TextImager and make it available as annotation server. We will also provide more output formats.

[1] `http://www.becalm.eu/api`

Acknowledgments. We gratefully acknowledge support by the *Deutsche Forschungsgemeinschaft* via the Specialised Information Services *Biodiversity Research*.

References

1. Hemati, W., Uslu, T., Mehler, A.: Textimager: a distributed UIMA-based system for nlp. In: Proceedings of the COLING 2016 System Demonstrations, Federated Conference on Computer Science and Information Systems (2016)
2. Ferrucci, D., Lally, A.: UIMA: an architectural approach to unstructured information processing in the corporate research environment. Natural Language Engineering **10**(3-4) (2004) 327–348
3. Finkel, J.R., Grenager, T., Manning, C.: Incorporating non-local information into information extraction systems by gibbs sampling. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Stroudsburg, PA, USA, Association for Computational Linguistics (2005) 363–370
4. Mueller, T., Schmid, H., Schütze, H.: Efficient higher-order CRFs for morphological tagging. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WashinTODOgton, USA, Association for Computational Linguistics (October 2013) 322–332
5. Kudo, T.: CRF++: Yet another CRF toolkit. Software available at https://taku910.github.io/crfpp/ (2005)
6. Geyer, K., Greenfield, K., Mensch, A., Simek, O.: Named entity recognition in 140 characters or less. In: Microposts. (2016)
7. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. CoRR (2016)