Neji: DIY web services for biomedical concept recognition

André Santos and Sérgio Matos*

DETI/IEETA, University of Aveiro,
Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
{andre.jeronimo,aleixomatos}@ua.pt
http://bioinformatics.ua.pt/

Abstract. The BioCreative V.5 task on technical interoperability and performance of annotation servers evaluated the provision of named entity annotations through web services.

This paper describes Neji, a web-services ready text processing and annotation framework, and shows how its modular and flexible architecture allowed simple adaptation to the requirements of the task. The configured service offers the annotation of eight concept types through five dictionaries and three machine-learning models, and has support for a variety of input and output formats.

Key words: Named entity recognition, biomedical text mining, web-services

1 Introduction

The BioCreative[1] community has promoted the development and evaluation of biomedical information retrieval and extraction tools, through the organization of various shared tasks focused on document triage, entity recognition (e.g. genes, chemicals) and relation extraction (e.g. protein-protein interactions, chemical-disease associations).

The technical interoperability and performance of annotation servers (TIPS) task, part of BioCreative V.5, focused on evaluating the technical aspects of providing inter-operable web services for named entity recognition [1]. We describe the latest developments of Neji, a modular framework for biomedical text processing and concept recognition, including the in-built support for REST web-services. Neji web server was used for participation in the TIPS task with a concept recognition service configured for annotating eight concept types through five dictionaries and three machine-learning models.

* Corresponding author

[1] http://www.biocreative.org/

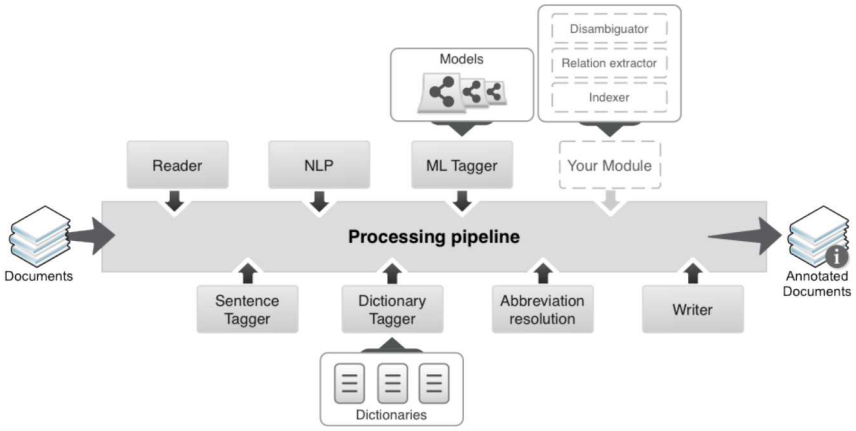

Fig. 1. Neji text processing pipeline.

2 System description

Neji[2] is a flexible and extensible concept recognition framework specially optimized for biomedical text [2]. As illustrated in Figure 1, the main component in Neji's architecture is the processing pipeline, which manages and sequentially executes a series of independent modules, each responsible for a specific processing task. Each module is implemented as a custom deterministic finite automaton (DFA) using Monq.jfa [3], a library for fast and flexible text filtering with regular expressions.

Neji includes natural language processing modules, based on GDep [3] and Apache OpenNLP [4], for machine learning (ML) and dictionary-based concept recognition, and for post-processing, including parentheses correction, abbreviation resolution, and false positives filtering. The machine learning component is based on Gimli [4], and makes use of MALLET [5] for providing simple training and application of Conditional Random Fields (CRFs) models [6].

Neji's modular architecture allows end users to configure the processing of documents according to their specific requirements, by simply combining existing modules for reading from and writing results to a variety of supported formats, and by using the appropriate dictionaries and machine learning models according to the concept types of interest, all of which can be achieved through the simple command line interface or through the provided API. Additionally, Neji can be extended by creating custom modules for reading from or writing to specific formats, or by adding new processing modules.

[2] Available from `https://github.com/BMDSoftware/neji`
[3] `https://github.com/HaraldKi/monqjfa`
[4] `https://opennlp.apache.org/`

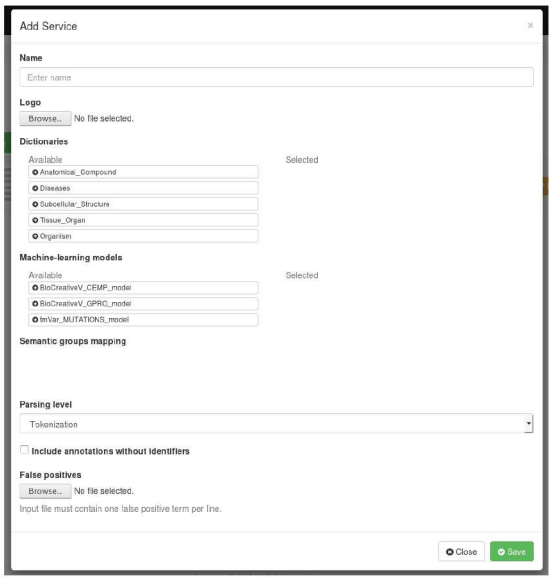

Fig. 2. Simple definition of an annotation service using dictionaries and ML models.

Neji web server is built on top of the Neji framework, proving a straightforward way of defining and managing annotation services, each accessible through a REST API end-point. The server also provides simple interfaces for managing annotation resources (dictionaries, or ML models previously trained with Neji) and for creating annotation services based on those resources, as shown in Figure 2. A web page with interactive annotation is also created for each service, allowing inspection of the annotation results and offering several exporting options to different formats (Figure 3). Neji server was developed in Java and uses a Jetty server and a SQLite database for storing the service configurations. The client side interfaces are based on HTML5, CSS3, JavaScript and Bootstrap, offering support on all modern browsers and platforms.

For the TIPS task, we developed four new writer modules to support all the output formats proposed in the task, namely TSV, JSON, BioC and BioC JSON. Additionally, the REST API was extended and adapted according to the task requirements. An annotation service was configured that allows annotating the following concept types: Anatomic Component, Diseases, Subcellular structure, Tissue and Organ, and Organism, through dictionaries compiled from the UMLS Metathesaurus, and Chemicals, Genes and Proteins, through machine learning models trained on the BioCreative V CHEMDNER corpus [7], and Mutations, using an ML model trained on the tmVar corpus [8]. The server accepts raw text as input, as well as PubMed and PubMedCentral identifiers, which are used for obtaining the documents to be processed. The output format and annotated

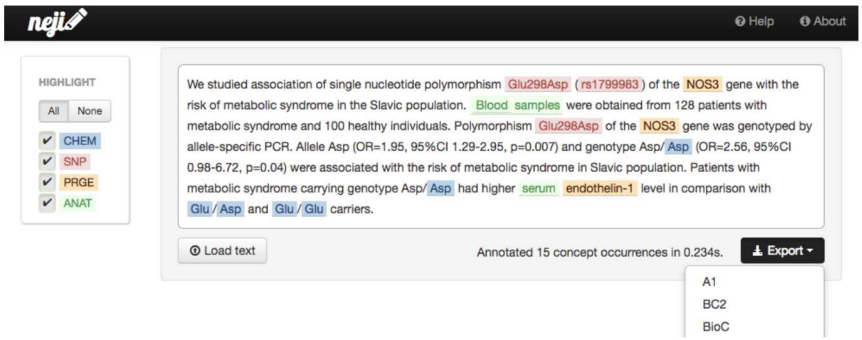

Fig. 3. A user interface is provided for each annotation service.

concept types can be configured by using the custom API parameters, as shown below. By default, all concept types are returned.

```
{
  "format": "BECALM_JSON",
  "groups": {ANAT, DISO}
}
```

3 Results

The annotation service for participating in the TIPS task was configured to run with 23 threads and was deployed on a Docker container with 32GB of memory over a server with 24 processing cores.

We performed a simple evaluation in terms of processing times by submitting several requests to the server, with different number of documents. We followed the procedure defined for the TIPS task, in which the document text is obtained from the BeCalm abstract and patent servers, and measured the time since the request was submitted to the Neji annotation service until the annotation results were returned. We observed average processing times ranging from 11.5 seconds for abstracts and 9.35 seconds for patents when annotating a single document, to 0.347 seconds per abstract and 0.173 seconds per patent when annotating sets of 1000 documents (Table 1).

We also measured the processing time for documents sent directly to the annotation server, that is, without request to the BeCalm document servers. In these tests, the full Craft corpus [9], composed of 67 full text documents containing more than 560000 tokens in total, was annotated in 15 minutes, which corresponds to an average processing time of 13.55 seconds per document and a processing speed over 600 tokens per second. Documents were sent to the annotation service one at a time and as raw text.

Table 1. Average processing times, in seconds, for documents obtained from the BeCalm document servers.

No. documents	abstracts	patents
1	11.5	9.35
100	0.421	0.236
1000	0.347	0.173

Acknowledgments This work was supported by Portuguese National Funds through FCT - Foundation for Science and Technology, in the context of the project IF/01694/2013. Sérgio Matos is funded under the FCT Investigator programme.

References

1. Pérez-Pérez, M., Pérez-Rodríguez, G., Blanco-Míguez, et al.: Benchmarking biomedical text mining web servers at BioCreative V.5: the technical Interoperability and Performance of annotation Servers - TIPS track. Proceedings of the BioCreative V.5 Challenge Evaluation Workshop., p. 12-21 (2017)
2. Campos, D., Matos, S., Oliveira, J.L.: A modular framework for biomedical concept recognition. BMC bioinformatics 14(281) (2013)
3. Sagae, K.: Dependency parsing and domain adaptation with LR models and parser ensembles. In: Eleventh Conference on Computational Natural Language Learning. pp. 1044–1050. Association for Computational Linguistics, Prague, Czech Republic (2007)
4. Campos, D., Matos, S., Oliveira, J.L.: Gimli: open source and high-performance biomedical name recognition. BMC bioinformatics 14(1), 54 (2013)
5. McCallum, A.K.: MALLET: A Machine Learning for Language Toolkit (2002)
6. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data (2001)
7. Krallinger, M., Rabal, O., Leitner, F., et al.: The CHEMDNER corpus of chemicals and drugs and its annotation principles. Journal of Cheminformatics 7(Suppl 1):S2 (2015)
8. Wei, C.H., Harris, B.R., Kao, H.Y., Lu Z.: tmVar: A text mining approach for extracting sequence variants in biomedical literature. Bioinformatics 29(11):1433-1439 (2013)
9. Verspoor, K., Cohen, K. B., Lanfranchi, A., et al.: A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. BMC Bioinformatics, 13:207 (2012).