Software

Neji: a tool for heterogeneous biomedical concept identification

David Campos[1,*], Sérgio Matos[1] and José Luís Oliveira[1]
[1]IEETA/DETI, University of Aveiro, 3810-193 Aveiro, Portugal

ABSTRACT

Motivation: Concept identification is an essential task in biomedical information extraction, presenting several complex and unsolved challenges. Current solutions are typically performed in an ad-hoc manner or optimized for specific biomedical concepts. Thus, the availability of general and modular solutions is scarce.

Results: This article presents Neji, an open source tool for biomedical concept recognition focused on four key characteristics: modularity, high-performance, speed and usability. It integrates features for biomedical natural language processing, from sentence splitting to chunking and dependency parsing, and supports the most popular input and output formats. Concept recognition and normalization are provided through dictionary matching and machine learning, and the resulting annotations are stored in an innovative concept tree implementation. Neji was evaluated against the CRAFT corpus, achieving high performance F-measure results: species (95%), cell (92%), cellular component (83%), gene and protein (76%), chemical (65%), biological processes and molecular functions (63%). It also provides fast and multi-threaded data processing, annotating up to 1200 sentences/second when using dictionary-based concept identification. Considering the provided features, underlying characteristics and current state-of-the-art methods, Neji constitutes an important contribution to the biomedical community, streamlining the development of complex concept recognition solutions.

Availability and Implementation: Neji is implemented in Java and is available at http://bioinformatics.ua.pt/neji.

Contact: david.campos@ua.pt

1 INTRODUCTION

A growing amount of biomedical data is continuously being produced, resulting largely from the widespread application of high-throughput techniques, such as gene and protein analysis. This growth is accompanied by a corresponding increase of textual information, in the form of articles, books and technical reports. Managing these large amounts of information and knowledge is rapidly becoming a very difficult task, especially when dealing with unstructured information in natural language texts. This has naturally led to the application of text mining (TM) systems to aid in the creation and curation of knowledge bases. An initial and crucial step for this is Named Entity Recognition (NER), aimed at identifying chunks of text that refer to specific entities of interest. However, the identification of such mentions is hindered by the lack of naming standards and the specific characteristics of biomedical entity names (Zhou *et al.*, 2004). Thanks to challenges such as BioCreative (Smith *et al.*, 2008; Morgan *et al.*, 2008; Lu *et al.*, 2011) and JNLPBA (Kim *et al.*, 2004), dozens of new solutions emerged for NER (e.g. Campos *et al.*, 2013) and for normalization (Wermter *et al.*, 2009). However, the resources provided by those challenges are often too specific and focused on the recognition of particular entity types (e.g., gene and protein), generating tailored solutions that provide high performance results on tested corpora. There are also solutions focused on providing annotation of heterogeneous biomedical concepts. For instance, Whatizit (Rebholz-Schuhmann *et al.*, 2008) and Cocoa[1] provide annotations of species, genes and proteins, and disorders, among others concepts. However, since they are provided as web-services, batch processing and application configurations are limited. MetaMap (Aronson, 2001) also provides annotation of heterogeneous concepts, using the Unified Medical Language System (UMLS) Metathesaurus (Bodenreider, 2004) and partial matching for extracting candidate strings with respective scores for concept names. Considering the current tools for the biomedical domain, we believe that there is a lack of solutions that allow batch processing of heterogeneous biomedical concepts, providing the complete set of recognized concepts and an easy to use integrated ecosystem. This document presents Neji, an open source tool optimized for heterogeneous biomedical concept recognition, supporting both machine learning and dictionary-based approaches, and combining the recognized concepts in a structured concept tree.

2 METHODS

2.1 Implementation

The core component of Neji is the processing pipeline, which allows users to submit various modules for execution following a FIFO (First In, First Out) strategy. A pipeline is a list of modules that are executed sequentially, considering specific goals and target chunks of text. **Fig. 1** illustrates the idea of this modular and flexible architecture. Each module is implemented as a custom Deterministic Finite Automaton (DFA), with specific matching rules and actions. The hierarchical text processing features of Monq.jfa[2] are used to support the pipeline infrastructure and module execution.

*To whom correspondence should be addressed.

[1] http://npjoint.com
[2] http://monqjfa.berlios.de

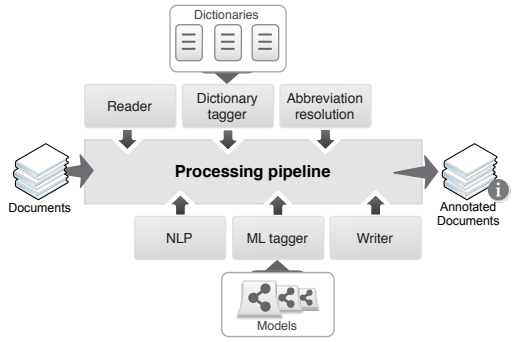

Fig. 1. Illustration of Neji's internal processing architecture.

2.1.1 Data structure Neji provides a flexible and complete data structure to store the generated information, providing easy and fast access to obtained sentences, tokens, concepts and natural language processing output. Since nested and intersected annotations are common in the biomedical domain, it is important to integrate a data structure to support such characteristics in the best and most automated way as possible. In Neji, this is achieved through a tree of annotations, presenting various advantages over typical approaches (e.g., list of annotations), such as the automatic maintenance of structured concept annotations, and easy identification of ambiguity problems. Additionally, each concept may have multiple identifiers associated, where each identifier contains information regarding its source, unique identifier, semantic group and semantic subgroup.

2.1.2 Input formats Both XML and raw text are supported. XML format allows specifying the tags of interest. For instance, considering the Pubmed XML format, if only titles and abstracts have to be processed, only the content of the tags "ArticleTitle" and "AbstractText" are of interest. On the other hand, the raw format considers that all the input text is of interest to be processed.

2.1.3 Natural language processing The sentence splitting module uses the model included in the Lingpipe[3] library, which was trained on biomedical corpora and presents high-performance results (Verspoor *et al.*, 2012). Natural Language Processing (NLP) tasks are performed using GDep (Sagae, 2007), a dependency parser for the biomedical domain built on top of the GENIA tagger, which performs tokenization, lemmatization, part-of-speech (POS) tagging, chunking and NER. Since we are not interested in the named entities provided by GENIA tagger, we removed that feature and its dependencies. Moreover, we adapted the tokenizer behavior in order to make it more consistent, which showed to improve results (Campos *et al.*, 2013). Additionally, since GDep combines all tasks to perform dependency parsing, we decoupled the various processing steps to make it more flexible, obviously respecting all task dependencies and resources (tokenization < POS < lemmatization < chunking < dependency parsing).

2.1.4 Dictionary matching Dictionary matching is offered using a modified version (Gerner *et al.*, 2010) of the dk.brics.automaton[4] library, a DFA implementation for exact and approximate matching. Since a large amount of false positives may be generated when using approximate matching, and considering that we are dealing with a general biomedical solution, we decided to use case insensitive exact matching. Orthographic variants of names can be generated and provided in the dictionary. Even so, it is necessary to pay special attention to terms that are common English words. Thus, a list of non-informative words for the biomedical domain (Kang *et al.*, 2011) is ignored during the matching process. Similarly, tokens with less than three characters are also discarded.

2.1.5 Machine learning The support for ML-based solutions is provided through Gimli (Campos *et al.*, 2013), which uses the Conditional Random Fields (CRFs) implementation from MALLET (McCallum, 2002) to recognize biomedical entity types, and provides high-performance results in two well-known corpora: GENETAG (Tanabe *et al.*, 2005) and JNLPBA (Kim *et al.*, 2004). It also provides a comprehensive set of features, serving as a good starting point to develop NER solutions for the biomedical domain. To complement Gimli, establishing a relation between the entity mentions and unique database identifiers, we developed a simple and general normalization algorithm based on prioritized dictionaries. Following this algorithm, if an identifier is found in the first dictionary, the match is complete and the algorithm finishes. If no match is found in the first dictionary, the second one is used to find a match, and so on. In the end, if no matches are found in the provided dictionaries, the annotation is discarded by default. This configuration works well if the first dictionary is a list of preferred names, and the remaining contain synonyms for each identifier. Moreover, it also provides flexibility to users, which only have to generate the various orthographic variants and prioritize them in the dictionaries. Regarding the matching approach, if a partial match of the annotation is found in the dictionary, it is accepted as a valid identifier for the complete chunk of text. For instance, if "BRCA1 gene" is recognized as an entity mention and only "BRCA1" is present in the dictionary, its identifier is associated with the annotation.

2.1.6 Abbreviation resolution Neji also integrates abbreviation resolution, by adapting a simple but effective abbreviation definition recognizer (Schwartz and Hearst, 2003), which is based on a set of pattern-matching rules to identify abbreviations and their full forms. In this way, we are able to extract both short and long forms of each abbreviation in text. If one of the forms is already provided as a concept, the other one is added as a new concept with the identifiers of the existing one. Additionally, any further occurrences of that entity are also automatically annotated.

2.1.7 Output formats Neji supports various well-known inline and standoff formats used in the biomedical domain, such as IeXML (Rebholz-Schuhmann *et al.*, 2006), A1[5], CoNLL (Tjong Kim Sang and De Meulder, 2003) and JSON. IeXML is an inline annotation format based on XML tags, supporting two levels of detail, i.e. only one annotation nested or intersected in another. Both CoNLL and A1 support ambiguous and intersected concept annotations. The output of the A1 format can be used with brat (Stenetorp *et al.*, 2012) for visualizing and editing the generated annotations. Finally, JSON provides all the information contained in the tree, together with the sentence and respective character positions.

2.1.8 Parallel processing On top of the previously described features, Neji also supports multi-threading processing, automatically duplicating the required resources when necessary. This allows annotating multiple documents at the same time, significantly dropping processing times.

2.2 Usage

Neji is provided as a simple but powerful Command Line Interface (CLI) tool, which provides a complete set of features: *1)* Annotate using dictionaries and/or machine-learning models with respective normalization dictionaries; *2)* Various input and output formats. When the XML input format is used, the XML tags should be indicated; *3)* Parsing level customization. By default, Neji automatically finds the appropriate parsing level considering the ML model characteristics; *4)* Number of threads customization; *5)*

[3] http://alias-i.com/lingpipe

[4] http://www.brics.dk/automaton

[5] http://brat.nlplab.org/standoff.html

Wildcard input filter to properly indicate the files to process; and *6)* Support for compressed and uncompressed files. Such features allow annotating a corpus using a simple bash command, such as:

```
./neji.sh -i input/ -if XML -o output/ -of XML -x AbstractText,ArticleTitle -d resources/dictionaries/ -m resources/models/ -c -t 6
```

In this example, six threads are used to annotate the compressed XML documents in the input folder with the specified dictionaries and machine-learning models, providing the resulting XML documents to the output folder. Note that only the text inside the specified tags is annotated.

3 RESULTS

We evaluated Neji in terms of the quality of the provided concept annotations and the required processing time, given a specific configuration of dictionaries and ML models selected according to the corpus used in the evaluation.

3.1 Corpus

Our analysis was centered on the CRAFT corpus (Bada *et al.*, 2012), one of the largest publicly available gold standard corpora, focused on multiple biomedical concept types with heterogeneous characteristics. The initial release contains a set of 67 full-text articles (more than 21 thousand sentences) manually annotated with concepts from nine biomedical ontologies and terminological resources: Chemical Entities of Biological Interest (ChEBI); Cell Ontology; Entrez Gene; Gene Ontology (biological process, cellular component, and molecular function); NCBI Taxonomy; Protein Ontology and Sequence Ontology. Overall, it contains almost 100 thousand concept annotations.

3.2 Resources

ML models and dictionaries were collected to recognize the biomedical concepts in the CRAFT corpus. Gene and protein names recognition was performed through a ML model trained on GENETAG using a complete and complex set of features, namely lemmas, POS, chunking, orthographic, local context (windows) and morphological features. LexEBI (Thompson *et al.*, 2011), which contains a filtered version of BioThesaurus (Liu *et al.*, 2006), was used to perform normalization. Two different dictionaries were created: the first with preferred names and the second with synonyms for each identifier. For each dictionary a set of orthographic and semantic variants was generated using the Lexical Variants Generation (LVG) tool (Bodenreider, 2004). Chemical recognition was based on a dictionary compiled from the ChEBI database of molecular entities (Degtyarenko *et al.*, 2008). Regarding species, the dictionary provided by LINNAEUS (Gerner *et al.*, 2010) was extended with NCBI Taxonomy entries assigned to taxonomical ranks above "species" and with synonyms from the UMLS Metathesaurus. Cell names were compiled from the "Cell" and "Cell Component" semantic types in the UMLS Metathesaurus. Finally, cellular components, biological processes and molecular functions were obtained from the corresponding sub-ontologies of the Gene Ontology (GO) (Ashburner *et al.*, 2000), and expanded with synonyms from UMLS and with concepts from the semantic types "Physiologic Function", "Organism Function", "Organ or Tissue function", "Cell function", "Molecular function" and "Genetic function". As a filtering step, we rejected names with one or two characters, names starting with a word from a strict list of stopwords (e.g. "the cell"), and also any single word name if that word was included in the list of most frequent words in MEDLINE. Some relevant terms that occur very frequently in MEDLINE, such as GO terms (e.g. "expression", "transcription") and species names (e.g. "human", "Saccharomyces"), were removed from this list to allow identifying them in texts. In the end, our dictionaries contain almost 1 million concept identifiers with 7 million name variants.

3.3 Concept annotation

Results previously presented using CRAFT are focused on text mentions, not evaluating the assigned identifiers. We follow the same approach, considering four matching strategies: exact (annotation is accepted if both left and right sides match); left (annotation is accepted if the left side matches); right (annotation is accepted if the right side matches); and overlap (annotation is accepted if there is any kind of match: exact, nested or intersected). Such matching strategies allow a better understanding of annotation quality, since a non-exact matching does not mean that the correct concept was not recognized. The common evaluation metrics of precision, recall and F-measure are used to analyze the results.

Considering the databases and ontologies used in the annotation of CRAFT, we defined six concept classes: species, cell, cellular component, chemical, gene and protein, and biological processes and molecular functions. Biological processes and molecular functions are grouped into a single class, since annotations are provided in a single file. Moreover, gene and protein annotations are evaluated against Entrez Gene. The comparison was performed against BANNER, the best performing system on (Verspoor *et al.*, 2012), and Cocoa and Whatizit web services, with the provided classes precisely mapped to the corresponding CRAFT concept types.

Fig. 2 presents the results achieved by Neji, Whatizit, Cocoa and BANNER on the CRAFT corpus, considering the various matching strategies. Overall, Neji presents the best results, with significant improvements on various concept types, namely on concepts associated with GO (cellular component, biological process and molecular function), chemical and gene/protein. In more detail, we can see that Neji is the solution that presents overall best recall results without loss in precision. Neji obtained state-of-the-art results on the recognition of species and cell concepts, with overlap F-measure results of 94.7% and 91.5%, respectively. It achieved an F-measure of 83.2% on overlap matching in the recognition of cellular component names, which is significantly better than Cocoa and Whatizit. Regarding gene and protein recognition, Neji ML model with normalization presents better results than Cocoa, BANNER and Whatizit on left and overlap matching. Its performance drop on exact and right matching appears to be a consequence of the different annotation guidelines in CRAFT and GENETAG, which was used to train Neji's ML model. Finally, the results achieved on chemical and biological processes and molecular functions are considerably better than Cocoa and Whatizit.

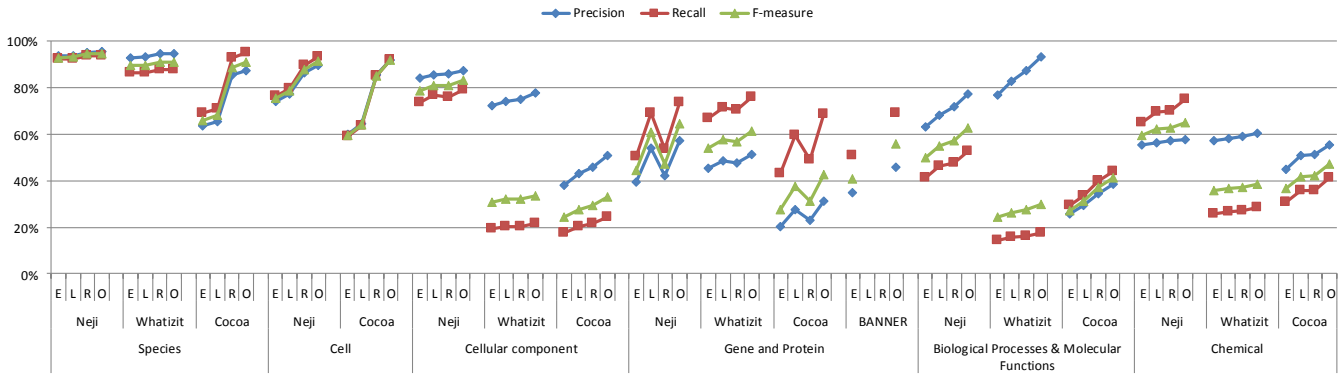

Fig. 2. Comparison of precision, recall, and F-measure results on CRAFT corpus, considering exact (E), left (L), right (R) and overlap (O) matching.

3.4 Speed

To evaluate the annotation speed of Neji, we performed various experiments using the CRAFT corpus, which contains 21749 sentences. The documents were processed on a machine with 8 processing cores @ 2.67 GHz and 16GB of RAM. The annotation process using the dictionaries and ML model previously described and using 5 threads took 124 seconds, corresponding to processing 175 sentences/second or to processing a full text article in 1.8 seconds. Considering that MEDLINE contains 11 million abstracts[6], and that each abstract contains on average 7.2 sentences (Yu, 2006), this configuration may annotate the entire MEDLINE in five days. Since generating complex features for the ML model and collecting POS and chunking features is resource intensive, we also measured the processing speed without using ML, applying only dictionary matching and tokenization from NLP. With this configuration, the CRAFT corpus was processed in 18 seconds, corresponding to 1208 sentences/second.

4 CONCLUSION

This article presents Neji, an open source tool optimized for heterogeneous biomedical concept recognition. It streamlines concept identification, using both dictionary and machine learning-based approaches to extract multiple concept types in an integrated ecosystem with built-in functionalities for natural language processing and concepts management. When evaluated against a manually annotated corpus, it achieved high-end results outperforming existing solutions. Additionally, the presented processing speeds for matching a large amount of concept names are a positive indicator of the solution's scalability. Based on the provided features and inherent characteristics, we believe that Neji is a positive contribution for the biomedical community, enhancing text mining and knowledge discovery processes, and helping researchers in the annotation of millions of documents with dozens of biomedical concepts, in order to infer new biomedical relations and concepts.

ACKNOWLEDGEMENTS

Funding: This research work was funded by FEDER through the COMPETE programme and by national funds through FCT - "Fundação Para a Ciência e a Tecnologia" under the project number PTDC/EIA-CCO/100541/2008. S. Matos is funded by FCT under the Ciência2007 programme.

[6] http://www.nlm.nih.gov/bsd/medline_lang_distr.html

REFERENCES

Aronson,A.R. (2001) Effective mapping of biomedical text to the UMLS Metathesaurus: the MetaMap program. *Proc AMIA Symp*, 17–21.

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics*, **25**, 25.

Bada,M. *et al.* (2012) Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, **13**, 161.

Bodenreider,O. (2004) The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, **32**, D267.

Campos,D. *et al.* (2013) Gimli: open source and high-performance biomedical name recognition. *BMC Bioinformatics*, **14**, 54.

Degtyarenko,K. *et al.* (2008) ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic acids research*, **36**, D344–D350.

Gerner,M. *et al.* (2010) LINNAEUS: a species name identification system for biomedical literature. *BMC Bioinformatics*, **11**, 85.

Kang,N. *et al.* (2011) Comparing and combining chunkers of biomedical text. *Journal of Biomedical Informatics*, **44**, 354–360.

Kim,J.D. *et al.* (2004) Introduction to the bio-entity recognition task at JNLPBA. Association for Computational Linguistics, Geneva, Switzerland, pp. 70–75.

Liu,H. *et al.* (2006) BioThesaurus: a web-based thesaurus of protein and gene names. *Bioinformatics*, **22**, 103–105.

Lu,Z. *et al.* (2011) The gene normalization task in BioCreative III. *BMC Bioinformatics*, **12 Suppl 8**, S2.

McCallum,A.K. (2002) MALLET: A Machine Learning for Language Toolkit.

Morgan,A.A. *et al.* (2008) Overview of BioCreative II gene normalization. *Genome Biol.*, **9 Suppl 2**, S3.

Rebholz-Schuhmann,D. *et al.* (2006) IeXML: towards an annotation framework for biomedical semantic types enabling interoperability of text processing modules. Fortaleza, Brazil.

Rebholz-Schuhmann,D. *et al.* (2008) Text processing through Web services: calling Whatizit. *Bioinformatics*, **24**, 296–298.

Sagae,K. (2007) Dependency parsing and domain adaptation with LR models and parser ensembles. Prague, pp. 1044–1050.

Schwartz,A.S. and Hearst,M.A. (2003) A simple algorithm for identifying abbreviation definitions in biomedical text. *Pac Symp Biocomput*, 451–462.

Smith,L. *et al.* (2008) Overview of BioCreative II gene mention recognition. *Genome Biol.*, **9 Suppl 2**, S2.

Stenetorp,P. *et al.* (2012) BRAT: a Web-based Tool for NLP-Assisted Text Annotation. *EACL 2012*, 102.

Tanabe,L. *et al.* (2005) GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, **6 Suppl 1**, S3.

Thompson,P. *et al.* (2011) The BioLexicon: a large-scale terminological resource for biomedical text mining. *BMC Bioinformatics*, **12**, 397.

Tjong Kim Sang,E.F. and De Meulder,F. (2003) Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. Association for Computational Linguistics, pp. 142–147.

Verspoor,K. *et al.* (2012) 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.

Wermter,J. *et al.* (2009) High-performance gene name normalization with GeNo. *Bioinformatics*, **25**, 815–821.

Yu,H. (2006) Towards answering biological questions with experimental evidence: automatically identifying text that summarize image content in full-text articles. American Medical Informatics Association, pp. 834–838.

Zhou,G. *et al.* (2004) Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, **20**, 1178–1190.