Real-time tagging of biomedical entities

Evangelos Pafilis¹* and Lars Juhl Jensen²*

¹ Institute of Marine Biology, Biotechnology and Aquaculture, Hellenic Centre for Marine Research, Heraklion, Crete, Greece.
² Disease Systems Biology Program, Novo Nordisk Foundation Center for Protein Research, Faculty of Health and Medical Sciences, University of Copenhagen, Copenhagen, Denmark.
* Contact pafilis@hcmr.gr or lars.juhl.jensen@cpr.ku.dk

Automatic annotation of text is an important complement to manual annotation, because the latter is highly labor intensive. We have developed a fast dictionary-based named entity recognition system, which is used for both real-time and bulk processing of text in a variety of biomedical web resources. We propose to adapt the system to make it interoperable with the PubAnnotation and Open Annotation standards.

Software implementation

The core of our named entity recognition server is a highly optimized dictionary-based tagging engine implemented in C++. The core tagger is able to process in the order of thousands of PubMed abstracts per second with a single CPU thread and is inherently thread-safe, allowing for perfect scalability in multi-threaded use. It is also available as a Python module, which has been generated in part by the Simplified Wrapper and Interface Generator (SWIG).

To make the tagger available as a web service, we have developed a multi-threaded Python HTTP server that utilizes the tagger Python module. To ensure efficiency and robustness, tagging requests are processed by a thread pool via a priority queue. If a user already has many requests in the queue, further requests are rejected with an error code to prevent a single user from blocking the service. Requests are also rejected if the document size exceeds 10 MB.

Applications of the tagger

The tagger has already been applied in a variety of different ways. Its high speed makes it well suited for real-time text mining of web pages, which we have utilized in the augmented browsing tool Reflect (1,2) and the interactive annotation tool EXTRACT (3).

The same tagging engine forms the basis for several command-line tools. We have previously published open-source taggers for named entity recognition of organisms (4) and environments (5) in large text corpora. Such taggers, combined with other dictionaries and a comention-based scoring scheme, also form the basis for extraction of associations among genes/proteins (6), small molecule compounds (7), cellular components (8), tissues (9), and diseases (10). These text-mining results are normalized to identifiers from suitable databases (11–14) and ontologies (15–18), integrated with associations from other data sources, and made available as a suite of web resources (4–10). The tagging results are also available as JSON-based Linked Data (JSON-LD) via a RESTful Open Annotation API described below (19).
RESTful web services

The real-time tagger can be accessed via standard HTTP requests with the following syntax:

http://tagger.jensenlab.org/{method}?document={text}&entity_types={types}&...

where {method} is either GetEntities or GetHTML, {text} is the plain or HTML-formatted text to be processed, and {types} specifies the types of entities to be tagged. These and additional optional parameters are explained in more detail in Table 1.

The GetEntities and GetHTML methods both tag the specified types of entities within the provided text document. However, they differ in the results that are returned. GetEntities returns the unique list of the entities identified in the document, either in tab-delimited or in XML format. By contrast, GetHTML expects the input document to be in HTML format and returns a modified HTML document in which the recognized terms have been marked up by HTML tags.

By default the tagger will auto detect the organisms mentioned in a document and subsequently tag their proteins/gene. If a user is interested in genes/proteins of specific organisms, tagging of them can be forced via the entity_types parameter (e.g. entity_types=9606 will force tagging of human proteins). Unless auto_detect is disabled (auto_detect=0), the tagger will also tag genes/proteins from any organisms that are explicitly mentioned in the text. If a user is not interested in the identification of genes/proteins, auto detection should be disabled.

Table 1. Parameters for the GetEntities and GetHTML requests.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>document</td>
<td>required</td>
<td>the plain or html-formatted text to be tagged</td>
</tr>
<tr>
<td>entity_types</td>
<td>required</td>
<td>the supported entity types are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0&gt;: Genes/proteins from the specified organism</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-1: PubChem Compound identifiers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-2: NCBI Taxonomy entries</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-21: Gene Ontology biological process terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-22: Gene Ontology cellular component terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-23: Gene Ontology molecular function terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-25: BRENDA Tissue Ontology terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-26: Disease Ontology terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-27: Environment Ontology terms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(concatenate with &quot;+&quot; to use multiple)</td>
</tr>
<tr>
<td>format</td>
<td>optional</td>
<td>tsv or xml (default)</td>
</tr>
<tr>
<td>auto_detect</td>
<td>optional</td>
<td>0 or 1 (default)</td>
</tr>
</tbody>
</table>

For further documentation on the API of the real-time tagger, please refer to O'Donoghue et al. (2) and http://extract.hcmr.gr/ (purple section of the help page).

The pre-tagged PubMed abstracts can also be queried via HTTP requests, using this syntax:

http://{resource}.jensenlab.org/document/{pmid}/annotations/, where {resource}
is either compartments, tissues, diseases, or organisms and {pmid} is the PubMed ID of the abstract of interest. For more general information on the RESTful Open Annotation API, please refer to Pyysalo et al. (19) and https://github.com/restful-open-annotation/.

Future plans

The core tagger obviously has the full information about which entities were found where in the text. However, this level of detail is not currently exposed by the REST API. Our plan for the hackathon is to address this shortcoming in a manner that makes the real-time tagger provide automatic annotations according to the PubAnnotation (20,21) standard and, if time permits, also the the Open Annotation (19) standard. In the case of the pre-tagged corpora, we plan to make these available also in PubAnnotation format.

Acknowledgments

This work was funded by the LifeWatchGreece Research Infrastructure (384676-94/GSRT/NSRF(C&E)) and the Novo Nordisk Foundation (NNF14CC0001).

References


