READ-Biomed-Server: A Scalable Annotation Server Using the UIMA Concept Mapper

Ruichen Teng and Karin Verspoor

School of Computing and Information Systems
The University of Melbourne
Melbourne VIC 3010, Australia
ruichen.teng@gmail.com
karin.verspoor@unimelb.edu.au

Abstract. For the BeCalm Technical Interoperability and Performance of annotation Servers (TIPS) task, we produced a fast dictionary lookup tool implemented as a standalone web service. It is based on a web service wrapper of the UIMA ConceptMapper, originally developed for the BioCreative-IV Comparative Toxicogenomics Database web service annotation task. We focused on annotation of *Gene Ontology* terms as the target annotation type for TIPS. We integrated this annotator into a scalable, micro-service architecture to build our annotation server. Message queues and thread pools were used to handle high concurrency and heavy computation. We also addressed handling errors and exceptions.

Key words: gene ontology, concept annotation, dictionary lookup, micro-service

1 Introduction

The BeCalm Technical Interoperability and Performance of annotation Servers (TIPS) task required the development of a web service for document annotation, and specifically annotation of named entities. The Reading, Extraction, and Annotation of Documents Biomedical Server (READ-Biomed-Server) is a web service set up to participate in this task, providing detection and annotation of terms in text from a given vocabulary using flexible dictionary-based matching. The server accepts annotation requests consisting of references to documents, and the types of entities that should be identified and annotated in the response. For the TIPS task, READ-Biomed-Server was set up to return annotations of terms in the Gene Ontology [2, 7], using the UIMA ConceptMapper dictionary-based concept recognition tool [10]. This tool has been shown to be effective for concept recognition of terms from large vocabularies [6], and for the Gene Ontology specifically works well when coupled with synonym generation strategies [5]. For TIPS, we emphasise its deployment in a robust and responsive architecture.

2 Annotator Description

2.1 Annotator methods

For the annotation web service, the implementation of the core annotator was largely based on the code developed for Comparative Toxicogenomics Database (CTD) Annotator [9], submitted for the BioCreative-IV CTD Challenge 2013 [1, 11]. It is a dictionary-lookup system using the UIMA [3, 4] ConceptMapper[1] [10], in which the task of annotating a document is treated as (possibly fuzzy) matching its tokens with terms in the dictionary. For this task we kept the default, strict matching strategy used in the original CTD Annotator [9], namely:

- `OrderIndependentLookup = false` only match if the order of the tokens is the same as in the dictionary.
- `FindAllMatches = false` only find the longest match, ignoring any shorter spans within.
- `SearchStrategy = ContiguousMatch` matched tokens should be adjacent to each other.

2.2 Dictionary

We used the vocabulary of the Gene Ontology[2] from the Gene Ontology project [2, 7]. This project aims at building structured representations of human knowledge related to gene function. We extracted all terms and synonyms, resulting in a dictionary of over 46,000 canonical terms and nearly 170,000 synonyms in ConceptMapper dictionary format. Due to memory limitations on the server used for the TIPS task, we did not load the extended GO synonym set [5].

Below is an entry in the dictionary:

```
<token canonical="ribosomal subunit export from nucleus">
<variant base="ribosomal subunit export from nucleus"/>
<variant base="ribosomal subunit export from cell nucleus"/>
<variant base="ribosomal subunit export out of nucleus"/>
<variant base="ribosomal subunit transport from nucleus to cytoplasm"/>
<variant base="ribosomal subunit-nucleus export"/>
<variant base="ribosome export from nucleus"/>
</token>
```

The annotator will match any synonym variant, and return an annotation to the canonical term.

[1] `http://uima.apache.org/d/uima-addons-current/ConceptMapper/ConceptMapperAnnotatorUserGuide.html`

[2] `http://geneontology.org/page/download-ontology`

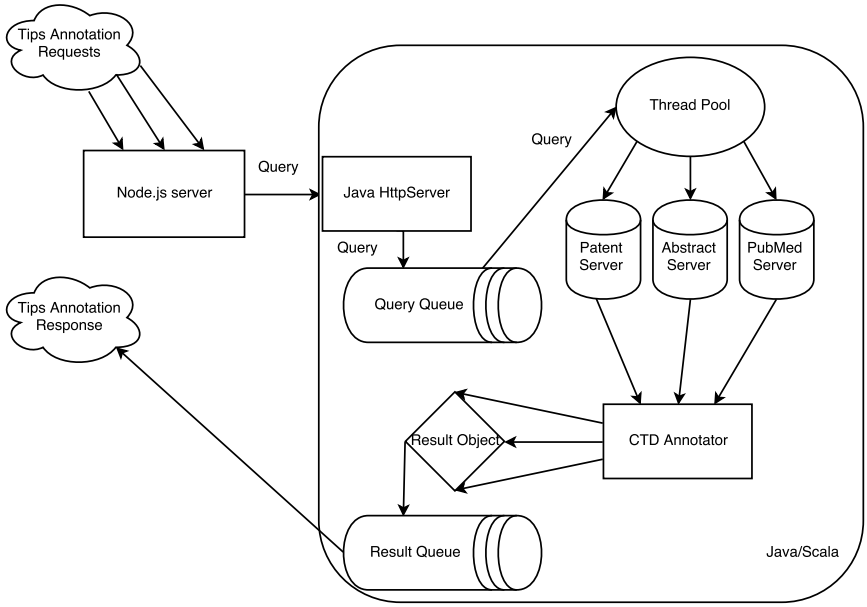

Fig. 1: System Architecture

2.3 Implementation details

As explained in [9], the CTD annotator is implemented as a standalone web service. It has a client-server architecture in which a client sends a message with the text to be annotated, and the server responds with the annotation result. In addition, the dictionary is transformed into the data structures required by the ConceptMapper tool and loaded into memory [10]. The CTD annotator was adapted for use in TIPS, substituting the dictionaries used for the CTD annotation task with the Gene Ontology dictionary, but otherwise using the same REST-based framework for processing annotation requests and translating UIMA annotation data structures into a correctly formatted response.

3 Web Server Description

3.1 System Architecture

The architecture of our approach is illustrated in Figure 1. Our annotation server has two major components. In the forefront, we implemented a "query dispatcher" type of Http server using *Node.js* to listen to requests from BeCalm metaserver, following the API defined at `http://www.becalm.eu/api`. This server does three things:

1. responds to "getState" requests with server state

2. responds to "getAnnotations" requests with ack 200
3. parses "getAnnotations" requests and sends a "query" to the *Java* Http server

Here, a query consists of the communication identifier (ID) and a series of document identifiers and document sources. The query is then passed on to the Java/Scala module to respond to this query, including requesting documents from different servers and annotating each of the documents. Some requests used in the TIPS evaluation included thousands of documents.

Within the Java/Scala module, in order to handle a large number of concurrent requests given that a request can take a long time to process if the number of documents requested is high, we implemented two message queues: one for storing queries, another one for storing annotation results. When the Java Http server receives a query, it adds the query to *Query Queue* and spawns some threads in the thread pool to process the queries. The processing includes:

- retrieving document texts from the source servers (Patent, Abstract, PubMed)
- calling the CTD Annotator to obtain annotation results
- putting results in the *Result Queue*
- sending results to BeCalm metaserver
- handling errors and exceptions

3.2 Document representation

```
1. EMBO J. 1999 Nov 15;18(22):6573-81.

The modified base J is the target for a novel DNA-binding protein in
kinetoplastid protozoans.

Cross M(1), Kieft R, Sabatini R, Wilm M, de Kort M, van der Marel GA, van Boom
JH, van Leeuwen F, Borst P.

Author information:
(1)Division of Molecular Biology and Centre of Biomedical Genetics, The
Netherlands Cancer Institute, 1066 CX Amsterdam, The Netherlands.

DNA from Kinetoplastida contains the unusual modified base
beta-D-glucosyl(hydroxymethyl)uracil, called J. Base J is found predominantly in
repetitive DNA and correlates with epigenetic silencing of telomeric variant
surface glycoprotein genes in Trypanosoma brucei. We have now identified a
protein in nuclear extracts of bloodstream stage T.brucei that binds specifically
to J-containing duplex DNA. J-specific DNA binding was also observed with
extracts from the kinetoplastids Crithidia fasciculata and Leishmania tarentolae.
We purified the 90 kDa C.fasciculata J-binding protein 50 000-fold and cloned the
corresponding gene from C.fasciculata, T.brucei and L.tarentolae. Recombinant
proteins expressed in Escherichia coli demonstrated J-specific DNA binding. The
J-binding proteins show 43-63% identity and are unlike any known protein. The
discovery of a J-binding protein suggests that J, like methylated cytosine in
higher eukaryotes, functions via a protein intermediate.

DOI: 10.1093/emboj/18.22.6573
PMCID: PMC1171720
PMID: 10562569  [Indexed for MEDLINE]
```

Fig. 2: PubMed document Representation

Listing 1 shows the document representation from abstract and patent servers where title and text can be extracted in json values. Figure 2 shows the document representation from PubMed server where it takes more sophisticated parsing to extract title and text.

```
{
   "externalId":"10196197",
   "title":"The subcellular localization of SF2/ASF...",
   "text":"...antibody..."
}
```

Listing 1: Abstract and Patent document representation

3.3 Error and exception handling

An important subtask in this competition is error and exception handling. Below are a few examples of errors and exceptions we have handled in our system:

- Failure to send message between components
- Failure to send annotation results (network error, response code not 200)
- Thread-related errors
- Java ApacheHttpClient related errors (connection pool management)

Particularly, if a result fails to be sent, it will be added back to *Result Queue*, waiting to be resent.

3.4 Multiple annotations over the same span

As shown in Listing 2, it is possible for the CTD annotator to identify multiple different annotations over the same span of text, for instance when the text contains an ambiguous term – a single string that maps to multiple concepts. In the standard ConceptMapper implementation in UIMA, each concept that the string matches is represented as a separate annotation. However, we found that we were unable to maintain this representation in the TIPS framework. Instead, we needed to return a single annotation, with each of the relevant matched canonical terms included in a comma-separated list in the database id field of the annotation response, as shown in Listings 3. To produce this alternative representation, we had to post-process the annotations to identify and combine those that overlapped the same span of text.

```
[
   {
      "matched_text":"antigen binding",
      "dict_type":"Gene Ontology",
      "line_number":0,
      "start_ofset":476,
      "end_offset":484
   },
   {
      "matched_text":"immunoglobulin complex, circulating",
      "dict_type":"Gene Ontology",
      "line_number":0,
      "start_ofset":476,
      "end_offset":484
   },
   {
      "matched_text":"B cell receptor complex",
      "dict_type":"Gene Ontology",
      "line_number":0,
      "start_ofset":476,
      "end_offset":484
   }
]
```


Listing 2: Original annotations from CTD Annotator


```
[
   {
      "document_id":"10196197",
      "section":"A",
      "init":476,
      "end":484,
      "score":0.856016,
      "annotated_text":"antibody",
      "type":"Gene Ontology",
      "database_id":"antigen binding, immunoglobulin complex,
      ↪  circulating, B cell receptor complex"
   }
]
```


Listing 3: Annotation result after post-processing

3.5 Performance

Table 1 summarises the key performance metrics of the READ-Biomed-Server in the TIPS evaluation, returning Gene Ontology term annotations for requested

Category	Performance
ART (average response time)	3.74128s
MAD (mean annotations per document)	8.896
MTSA (Mean time in seconds seek annotations)	0.42042
MTDV (Mean time in seconds per document volume)	0.00307
Mean processing time per document	3.74125
MPDV (total predictions per document volume)	0.0073

Table 1: READ-Biomed-Server System Performance

Time(s)/documents	Time(s)/predictions	#Documents	#Predictions
0.00962	0.00098	4807	47100
0.00986	0.00103	4837	46266
0.00932	0.00101	2900	26757
0.00873	0.00091	2813	26915
0.01222	0.00124	1169	11527

Table 2: Performance for requests with over 1000 documents

documents. As shown in Table 2, our system has very good performance for requests that contain a large number of documents (in thousands). This type of request did not occur during the TIPS evaluation although it was initially proposed as a relevant performance requirement. This was therefore something we had in mind when designing our system.

3.6 Discussion

The micro-service oriented architecture we have employed, illustrated in Figure 1, is highly scalable. Each component (CTD Annotator, Message Queues, Thread Pool) can be easily detached from the module and "scaled out". In production environments, this architecture is beneficial as it has good separation of concerns [8], due to modularity and encapsulation, and no single point of failure.

During the evaluation period, we modified the server several times in order to improve the system performance. The changes we implemented were focused on optimising thread architecture for our CPU and reducing polling message queues to save system resources, but the annotator module remained unchanged.

4 Conclusions

For the TIPS evaluation, we adapted a UIMA ConceptMapper-based annotation server previously developed for BioCreative IV. We primarily focused on implementing our annotation server within a scalable architecture, as well as emphasising robustness and error handling. Since our annotator had already been implemented as a web service, we decided to build loosely coupled components

around it. The end result is an annotation server with a micro-service oriented architecture where different components are implemented as web services.

References

1. Arighi, Cecilia N., Cathy H. Wu, Kevin B. Cohen, Lynette Hirschman, Martin Krallinger, Alfonso Valencia, Zhiyong Lu, John W. Wilbur, and Thomas C. Wiegers. *BioCreative-IV virtual issue.* bau039. (2014)
2. Ashburner et al. Gene ontology: tool for the unification of biology *Nat Genet* **25(1)**:25-9. (2000)
3. Ferrucci, David, and Adam Lally. *UIMA: an architectural approach to unstructured information processing in the corporate research environment.* Natural Language Engineering 10, no. 3-4 : 327-348. (2004)
4. Ferrucci, David, Adam Lally, Karin Verspoor, and Eric Nyberg. *Unstructured Information Management Architecture (UIMA), Version 1.0.* OASIS Standard. (2008).
5. Funk, Christopher S., Kevin Bretonnel Cohen, Lawrence E. Hunter, and Karin M. Verspoor. Gene Ontology synonym generation rules lead to increased performance in biomedical concept recognition. Journal of Biomedical Semantics. 7:52. (2016)
6. Funk, Christopher, Willliam A. Baumgartner, Benjamin Garcia, Christophe Roeder, Michael Bada, Kevin Bretonnel Cohen, Lawrence E. Hunter, and Karin Verspoor. Large-scale biomedical concept recognition: An evaluation of current automatic annotators and their parameters. BMC Bioinformatics 15(1):59. (2014)
7. The Gene Ontology Consortium. Gene Ontology Consortium: going forward. *Nucl Acids Res* **43** Database issue D1049–??D1056. (2015)
8. Phillip A. Laplante. What every Engineer should Know about Software Engineering. CRC Press, Inc., Boca Raton, FL, USA. (2007)
9. MacKinlay, Andrew, and Karin Verspoor. *A web service annotation framework for CTD using the UIMA concept mapper.* In BioCreative challenge evaluation workshop vol, vol. 1. (2013)
10. Tanenblatt, Michael A., Anni Coden, and Igor L. Sominsky. *The ConceptMapper Approach to Named Entity Recognition.* In LREC. (2010)
11. Wiegers, Thomas C., Allan Peter Davis, and Carolyn J. Mattingly. *Web services-based text-mining demonstrates broad impacts for interoperability and process simplification.* Database 2014: bau050. (2014)