Infrastructure for Semantic Annotation in the Genomics Domain

Mahmoud El-Haj[1] **Nathan Rutherford**[1] **Matt Coole**[1] **Ignatius Ezeani**[1]
Sheryl Prentice[1] **Nancy Ide**[2] **Jo Knight**[1] **Scott Piao**[1]
John Mariani[1] **Paul Rayson**[1] **and** **Keith Suderman**[3]

[1] Lancaster University, UK
[2], [3] Vassar College, USA
[1] {m.el-haj, n.rutherford, m.coole, i.ezeani, s.r.prentice1, jo.knight, s.piao, j.mariani, p.rayson}@lancaster.ac.uk
[2] ide@vassar.edu, [3] suderman@cs.vassar.edu

Abstract

We describe a novel super-infrastructure for biomedical text mining which incorporates an end-to-end pipeline for the collection, annotation, storage, retrieval and analysis of biomedical and life sciences literature, combining NLP and corpus linguistics methods. The infrastructure permits extreme-scale research on the open access PubMed Central archive. It combines an updatable Gene Ontology Semantic Tagger (GOST) for entity identification and semantic markup in the literature, with a NLP pipeline scheduler (Buster) to collect and process the corpus, and a bespoke columnar corpus database (LexiDB) for indexing. The corpus database is distributed to permit fast indexing, and provides a simple web front-end with corpus linguistics methods for sub-corpus comparison and retrieval. GOST is also connected as a service in the Language Application (LAPPS) Grid, in which context it is interoperable with other NLP tools and data in the Grid and can be combined with them in more complex workflows. In a literature based discovery setting, we have created an annotated corpus of 9,776 papers with 5,481,543 words.

Keywords: BioNLP, Ontology, Semantic Tagger, Corpus, PubMed, Genomics, Infrastructure

1. Introduction

In many fields, academics rely only on full text searching and citation networks to find related research. In the medical domain, much research has been undertaken in literature-based discovery that relies on knowledge and information extraction techniques to perform automated hypothesis generation, in order to find new relationships between existing knowledge. High-level semantic taxonomies and networks can be applied to achieve broad linking and identification of semantic categories, but these fail to identify and disambiguate sub-discipline-specific terminology, or indeed to cope with the continued expansion and development of domain specific terminologies. For example, a search in the main biomedical literature citation database (PubMed) for the term ‘genome wide association study’ results in just five papers from 1995, 141 from 2005 and 3,633 from 2015. At the same time, the domain terminology in genomics has developed, expanded and changed, meaning that broad coverage semantic taxonomies cannot keep in step.

In this paper, we propose a novel combination of Natural Language Processing (NLP) and Corpus Linguistics (CL) methods and tools, connected together via SOAP and REST API calls in a loosely-coupled open infrastructure intended to provide a range of facilities for researchers to collect, annotate, store and retrieve large corpora derived from open access biomedical and life sciences literature. All the tools used were developed by the authors of this paper. NLP annotation tools draw from an existing Gene Ontology (GO) which is updated monthly, and facilitate the automatic identification of genomics terminology, which is comprised largely of multiword expressions. In turn, we combine this with CL methods to permit the large-scale comparison of sub-corpora of literature to uncover how the field has developed over time, or uses different vocabulary in newly developing sub-fields employing keyness, collocation and n-gram tools combined with the semantic annotation. Completing the full cycle, our analysis interface is intended to support the study of the genomics literature corpus, not just for research purposes, but also to improve the quality of supporting resources too e.g. the Gene Ontology, by exposing the usage of terminology in the field and how it has developed over time, akin to the revolution in lexicography via corpus-based dictionary production observed in the 1980s and 1990s.

Our specific contributions in this paper are as follows: 1) entity identification and semantic linking in the genomics domain, 2) a novel open infrastructure for biomedical text mining, 3) a large annotated corpus consisting of open access PubMed Central papers, 4) open platforms to support research reproducibility, and 5) supporting literature based discovery with a novel combination of NLP and CL methods.

2. Related Work

Analysing biomedical data using Natural Language Processing (NLP) and text mining requires a significant amount of domain knowledge (Tan and Lambrix, 2009). Such knowledge is usually found in domain specific ontologies such as the Gene ontology resource[1] which contains important information related to gene products and functions (Kumar et al., 2004).

Over many years, NLP techniques have been widely applied to biomedical text mining to facilitate large-scale information extraction and knowledge discovery from the rapidly increasing body of biomedical literature. Since the begin-

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

ning of biomedical language processing in the late 1990s, the field continued to receive great attention with specialised events and workshops focusing on biomedical NLP, such as the BioNLP Workshop series.

Current biomedical libraries such as MEDLINE[2] by the US National Library of Medicine (NLM)[3] provide searchable databases that are rich with citations and abstracts from the biomedical domain. Tools such as PubMed[4] by NLM can be used to freely search and retrieve abstracts and publications from MEDLINE database. MEDLINE's citations are updated and added to PubMed seven days a week and in 2017 alone more than 800,000 citations were added to MEDLINE[5]. This shows the need for NLP and text mining tools to be able to analyse the constantly growing field.

Among the early researchers who worked on information extraction from MEDLINE was Yakushiji et al. (2000), who implemented an information extraction system using the text of full papers from MEDLINE to investigate the feasibility of text mining, using a general-purpose parser and grammar applied to biomedical domain. Shortly after that, Srinivasan (2001) explored text mining from metadata included in MEDLINE citations. This work introduced MeSHmap, a text mining system that exploits the MeSH[6] indexing accompanying MEDLINE. MeshMap supports searching PubMed using MeSH terms and subheadings, it also allows to compare entities of the same type such as pairs of drugs or pairs of procedures.

The size of the biomedical literature and the variety of citations provide new challenges to text mining. Ananiadou et al. (2006) identified the unmanageable issue of finding useful information manually from the plethora of biomedical scientific literature. Others such as Kann (2007) have also suggested that text mining and analysis approaches are essential for discovering hidden information about given diseases and protein interactions buried within millions of biomedical texts.

Since the recognition of the importance of the biomedical text mining, a variety of NLP tools have been developed and modified to support it. Among the main tools and corpora developed for such purposes include the Genia tagger and corpus (Tsuruoka et al., 2005; Thompson et al., 2017), GOST tagger (El-Haj et al., 2018), and Termine[7]. A related biomedical annotation tool is the Penn BioTagger[8] (Jin et al., 2006), which is capable of tagging gene entities, genomic variations entities and malignancy type entities.

In addition, several infrastructures supporting biomedical text mining have been developed, including U-Compare (Kano et al., 2008) and Argo (Rak et al., 2012). The General Architecture for Text Engineering (GATE) (Cunningham et al., 2011), a broader-based framework for text mining, also includes some tools for handling biomedical texts. These tools and infrastructures are typically self-contained and focused on lexical, syntactic and shallow semantic (named-entity) approaches. More recently, the LAPPS Grid (Ide et al., 2014) has been augmented to support mining biomedical literature (Ide et al., 2018), as well as sophisticated interactive annotation and machine learning tools for domain adaptation to support mining literature in the life sciences.

3. Biomedical Text Mining Infrastructure

In order to fully support the complete cycle of literature-based discovery and symbiotic improvement in language resources in the genomics domain with an existing vast body of work, we need large scale infrastructures. The following subsections discuss our Gene Ontology Semantic Tagger (GOST) (section 3.1.) and its integration with both our new Buster NLP pipeline (section 3.3.) and the LAPPS Grid (section 3.2.) which was previously developed by co-authors from Vassar College. By connecting GOST both with Buster and LAPPS Grid, we are able to provide annotation pipelines to drive full text into our simple web front-end application to support CL style queries (section 3.5.), as well as flexible interoperable NLP workflows.

3.1. GOST

GOST is an updatable Gene Ontology Semantic Tagger (El-Haj et al., 2018). GOST automatically annotates biomedical genomics terms with GO IDs[9] to provide a better coverage via a more fine-grained medical terminology, which helps to include an extra level of annotation by tagging biomedical corpora using the Gene Ontology Consortium's OBO Basic Gene Ontology (go-basic.obo) categories[10]. GOST permits genomics researchers to explore their rapidly growing literature in new ways.

GOST was created by adding a gene ontology dictionary to USAS[11] - a framework for undertaking the automatic semantic analysis of text. This was done by parsing the OBO Basic Gene Ontology. The go-basic.obo is the basic version of the GO ontology, filtered such that the graph is guaranteed to be acyclic paths, and annotations can be propagated up the graph. GOST focuses on the *is_a* relation in order to trace ancestors and children for each entry in the ontology. The *is_a* relationship was chosen in the first instance because it has a more intuitive meaning. Something is only considered *is_a* if an instance of the child process is an instance of the entire parent process. The USAS extended gene ontology dictionary was created following the five steps below:

1. Determine whether a child entry in OBO is single or multi-word expression (e.g. "Cell" vs "Immune System Process").

2. trace the number of paths from a child to the root (i.e. "biological process").

3. extract GO ID entries (i.e. child node's ancestors).

4. determine the level of each ancestor (e.g. appending .1 to the end of that tag refers to the first parent of the node).

[2] https://www.nlm.nih.gov/bsd/medline.html

[3] https://www.nlm.nih.gov/

[4] https://www.ncbi.nlm.nih.gov/pubmed

[5] https://www.nlm.nih.gov/bsd/medline.html

[6] Medical Subject Heading (MeSH) terms are a tree of controlled vocabulary maintained by the Library Operations division of NLM.

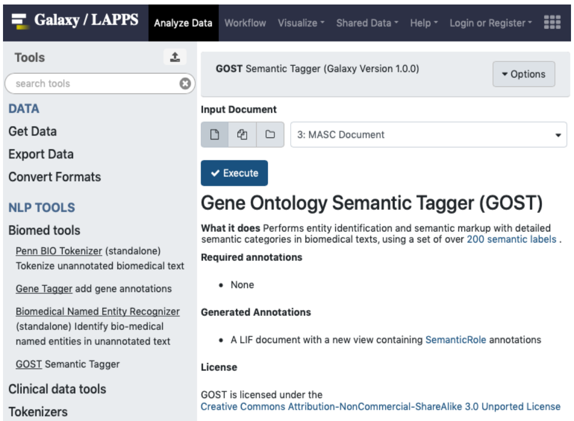
[7] http://www.nactem.ac.uk/software/termine/

[8] http://seas.upenn.edu/∼strctlrn/BioTagger/BioTagger.html

[9] http://geneontology.org

[10] http://current.geneontology.org/ontology/go-basic.obo

[11] http://ucrel.lancs.ac.uk/usas/



Penn BIO Tokenizer (standalone) Tokenize unannotated biomedical text

Gene Tagger add gene annotations

Biomedical Named Entity Recognizer (standalone) Identify bio-medical named entities in unannotated text

GOST Semantic Tagger

GOST Semantic Tagger (Galaxy Version 1.0.0)

Input Document

3: MASC Document

Gene Ontology Semantic Tagger (GOST)

What it does Performs entity identification and semantic markup with detailed semantic categories in biomedical texts, using a set of over 200 semantic labels .

Required annotations

- None

Generated Annotations

- A LIF document with a new view containing SemanticRole annotations

License

GOST is licensed under the
Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License

Figure 1: GOST in the LAPPS Grid Galaxy instance

5. determine whether the path passes through an "Immune System Process", if so the tag will end with a .I to refer to an immune entry otherwise .N referring to a non-immune entry.

The process above resulted with a dictionary of 433 single-word bioterms and 44,180 multiword bioterms, which has been merged into USAS creating a new USAS semantic annotation system, named GOST (Gene Ontology Semantic Tagger).

3.2. LAPPS Grid

The Language Applications (LAPPS) Grid (Ide et al., 2014) provides a large collection of NLP tools exposed as SOAP (Simple Object Access Protocol) web services, together with access to a variety of resources commonly used in the domain. The services are made available to users via a customised instance of the Galaxy web-based workflow development engine[12] (Goecks et al., 2010), directly via SOAP calls, and programmatically through Java and Python interfaces. Crucially, all tools and resources in the LAPPS Grid are rendered mutually interoperable via transduction to the JSON-LD LAPPS Grid Interchange Format (LIF (Verhagen et al., 2016)) and the Web Service Exchange Vocabulary (WSEV (Ide et al., 2015)), both designed to capture fundamental properties of existing annotation models in order to serve as a common pivot among them.

Recently, the LAPPS Grid has been augmented to support mining biomedical literature (Ide et al., 2018) by providing interoperable access to a wide variety of bio-oriented tools, including GOST (Figure 1), the Penn BioTokenizer, Penn BioTagger, the ABNER Biomedical Named Entity Recognizer[13], and cTakes software[14] for analysing clinical texts. Notably, the LAPPS Grid also provides interoperable access to major resources for biomedical publication mining, including several gold standard corpora from several past BioNLP shared tasks as well as the holdings of PubMed and PubMed-Central[15]. The LAPPS Grid has recently incorporated reciprocal access to resources and tools available from PubAnnotation[16], a platform for collaborative annotation of biomedical publications; and an Apache Solr query engine to extract relevant publications from PubMed Central (PMC)[17]. The Grid also incorporates PubAnnotation's TextAE annotation editor, which, coupled with facilities for machine learning, provides an environment for rapid adaptation of trainable named entity recognition (NER) modules to domain-specific vocabularies and development of gold standard data for machine learning and evaluation.

Number of Articles	9,776
Number of Journals	1,178
Words	5,481,543
Download size	72 GB
Corpus Size (Single File)	11 GB
Corpus Size (Tokens)	2.4 GB
Processing Time	Approx 5 days

Table 1: Annotated Corpus

3.3. Buster

Buster is a linear NLP pipeline that downloads full-text, open access papers from PubMed Central (PMC), tags them using GOST, and indexes them in LexiDB (Section 3.4.) to create an annotated corpus (Table 1). Our initial tests of Buster processed almost 10,000 papers from PMC over a five day period.

The pipeline was developed in Python[18] leveraging modern technologies such as docker to provide a modular and scalable system for researchers.

The system comprises of five key components as shown in Figure 2:

Web server: This hosts both the website (web front-end) that can be used to interact with the dataset, and communicates with the NLP pipeline system.

MySQL Databases: There are two auxiliary databases for the system; notifications, and papers. Notifications is used to store status messages passed by the NLP pipeline to the web server to enable progress to be tracked. Papers contains the set of all the papers that have been passed through the pipeline, including metadata.

LexiDB: This is the corpus databases as explained in section 3.4., that is used to store the textual data extracted from all the PMC papers. LexiDB contains two corpora; a tokens corpus containing all of the single word tokens extracted from each paper along with some metadata, and a publications corpus which contains the set of all papers passed that the tokens were extracted from.

Celery: We used celery[19] as a distributed task queue, using RabbitMQ[20] as our message broker, and Redis[21] as the backend. This was used to provide a management system to the pipeline. Each of Buster's components is defined as a celery worker that receives start commands from the celery component. Celery provides reliability by guaranteeing that each worker receives by utilising the redis back-end as a persistent message queue.

[12] https://galaxy.lappsgrid.org
[13] http://pages.cs.wisc.edu/ bsettles/abner/
[14] https://ctakes.apache.org
[15] https://www.ncbi.nlm.nih.gov/pmc/
[16] http://pubannotation.org
[17] https://services.lappsgrid.org/eager/ask
[18] https://delta.lancs.ac.uk/BioTM/BUSTER
[19] http://www.celeryproject.org/
[20] https://www.rabbitmq.com/
[21] https://redis.io/

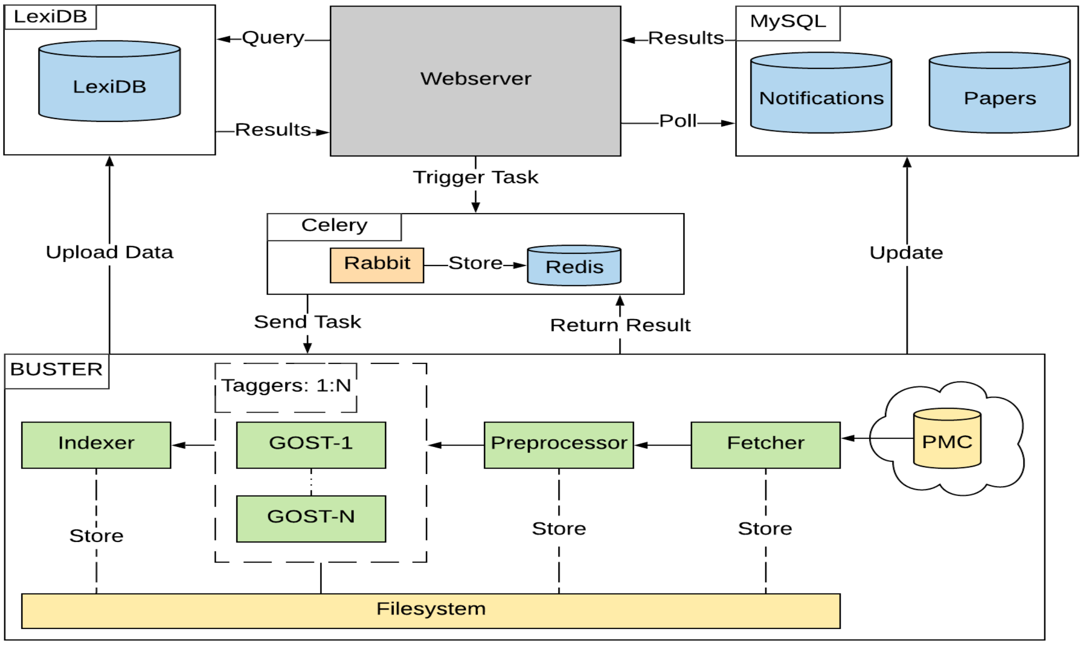


Figure 2: Buster System Architecture Diagram

Buster Pipeline: The Buster pipeline was developed with modularity in mind. Each component is containerised using docker and managed using docker-compose. This enables functionality such as components replication for load-balancing and an overlay network for intra-container communication. Each component takes a predefined file input, and presents a defined file output. As a result of this design, any of the components can be replaced or augmented, so long as the expected input and output are presented.

Figure 2 shows Buster's architecture diagram and the interaction between the components, including GOST as follows:

Fetcher

This component will download all of the open-access files from PMC open-access-subset[22] using their FTP service[23]. Files are downloaded one at a time in compliance with the PMC FTP API regulations[24]. We first download the OAS Non-commercial papers list and download each paper on that list that appears within our PMC query. The set of papers that match our query is acquired using selenium[25] on the PMC search tool and returning the PMCID of each of the results. The fetcher then extracts the .nxml file from the downloaded ZIP folder and places each of these in a folder named by the PMCID of the paper.

Preprocessor

The preprocessor will take the .nxml file extracted by the fetcher and convert it to plain text. It first finds the body of the paper (or the abstract if one cannot be found), proceeding to then remove any of the XML from the text. At this stage, we also extract any meta-data available within the .xml file and insert it into the papers database.

Tagger/GOST

GOST was dockerised along with a python handler for calling the java application, and replicated within the pipeline to increase performance. GOST takes the plain text files composed by the preprocessor and begins to tag each token. The result of this is a CSV file, with each row representing a token in the file, along with its semantic information.

GOST also has the capabilities to be updated periodically with an updated version of the Gene Ontology by generating a list of single words and multiword expressions using a separate tool[26] that can be used to generate a new lexicon resource for GOST.[27]

Indexer

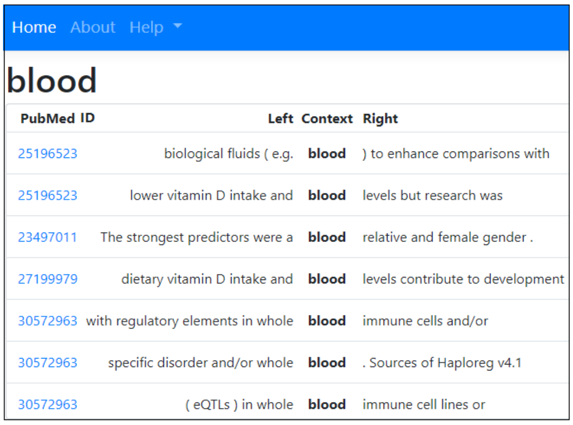
The final component of this pipeline is the indexer. This interacts with LexiDB's REST API to create new corpora and send processed chunks.

3.4. lexiDB

LexiDB is a column centric database management system (DBMS) designed to handle annotated text, similar series data and accompanying metadata. It provides mechanisms for performing complex corpus queries on tagged token streams and can be scaled out across multiple nodes to provide scalability and accommodate corpora consisting of tens of billions of tokens.

Previous work (Coole et al., 2015) has shown whilst existing DBMSs are capable of storing and searching corpora they fall short in both their ability to express corpus queries in a meaningful syntax for linguistic users and their ability to scale up to handle multi-billion word corpora. LexiDB was

[22] https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/
[23] https://www.ncbi.nlm.nih.gov/pmc/tools/ftp/
[24] https://www.ncbi.nlm.nih.gov/pmc/tools/developers/
[25] https://www.seleniumhq.org/
[26] https://github.com/drelhaj/BioTextMining
[27] It is left up to the domain expert end user to decide if they wish to retag existing corpora in the system. It is not clear that new versions of the Gene Ontology should be retrospectively applied to earlier papers.



blood

PubMed ID	Left	Context	Right
25196523	biological fluids (e.g.	**blood**	) to enhance comparisons with
25196523	lower vitamin D intake and	**blood**	levels but research was
23497011	The strongest predictors were a	**blood**	relative and female gender .
27199979	dietary vitamin D intake and	**blood**	levels contribute to development
30572963	with regulatory elements in whole	**blood**	immune cells and/or
30572963	specific disorder and/or whole	**blood**	. Sources of Haploreg v4.1
30572963	(eQTLs) in whole	**blood**	immune cell lines or

Figure 3: Concordances of the query 'blood'

developed with these factors in mind and has been shown (Coole et al., 2016) capable of fulfilling both the corpus query and scalability requirements making it well suited to being the final data-sink for the Buster Pipeline.

The data is stored in two primary tables within the LexiDB database, a token stream table and a publications table. The token stream table leverages the Zipfian column family store for storing and indexing tokens and their associated POS tags and semantic annotations. A continuous column family store is then utilised to provide a means of fast and efficient joins on queries between the token stream and the publications table. These queries are sent by means of a REST API from the corpus interface as described in Section 3.5..

3.5. Corpus interface

The corpus interface is managed by the web-server as noted in section 3.3.. This is a web-app that comes with a graphical user interface (GUI) styled using bootstrap v4[28] for accessibility and responsiveness. This is the entered end-point through which researchers will interact with the corpus.

Interactions with the corpus are facilitated by LexiDB using a REST API that supports querying of the corpus. At present, concordance and keyness queries are supported by the corpus interface. Figure 3 shows concordances returned for the query "blood". Queries are sent to the REST API via asynchronous javascript calls using the fetch[29] API. This approach enables the users to continue to use the website while queries are processed by the backend, which will only show results whenever the query has completed. Presenting a cleaner user-experience than a synchronous alternative that would make the user wait until the query has been processed.

Queries from the corpus interface are generated via javascript based on the input values in the search tool using query templates based on the desired result. Requests are sent to LexiDB as POST requests, sending the table and token users want to query. Results are returned in JSON format, and displayed using a relevant page template. Each displayed result has a reference back to the original PMC paper from which it was extracted (based on the metadata stored in LexiDB). An example is shown in Listing 1.

Listing 1: LexiDB REST API Query Example

```
query =
    {
      "query": {
        "tokens":[
          {"sem": "GO:1904124"}
        ]
      },
      "result": {"type": "kwic"}
    }
```

4. LAPPS Grid and GOST

GOST has been added to the LAPPS Grid as a callable service for annotating texts with semantic tags and GO identifiers. There are several advantages to incorporating GOST into the LAPPS Grid, most notably that incorporation requires that GOST is *interoperable* with all other applications available in the Grid. This is accomplished by mapping GOST's input and output formats to the LAPPS Interchange Format (LIF) (Verhagen et al., 2016) and the LAPPS Grid Web Service Exchange Vocabulary (WSEV) (Ide et al., 2015), an ontology of terms and their properties commonly used in the Natural Language Processing (NLP) field. GOST output can then be processed by tools that may generate additional annotation layers ("views" in LAPPS Grid terminology), with or without using GOST's semantic annotations. The LAPPS Grid also provides a Solr-based query engine for PubMed data that is augmented with ranking rules whose weights can be tweaked as desired by the user, results from which can be used as input to GOST.

A major advantage of incorporating GOST into the LAPPS Grid is the access to PubMed data provided by the newly-established incorporation of the facilities of PubAnnotation (Kim and Wang, 2012) into the Grid. PubAnnotation not only provides access to all PubMed texts, but also, crucially, serves as a repository of annotations that are linked together by common reference (via standoff annotation) to the canonical texts. A common annotation repository enables combining GOST's semantic annotations with annotations generated by other software and/or by human annotators, which in turn can yield insight into linguistic and semantic properties of biomedical terminology and improve our ability to extract meaningful information from biomedical publications. PubAnnotation also provides an annotation editor, now available within the LAPPS Grid, to support "human-in-the-loop" manual correction of automatically generated annotations–in particular, the semantic annotations generated by GOST. Automatically-generated annotations that have been subsequently curated by human experts can be exploited to (re-)train machine learning algorithms in order to gain increased tagging accuracy. Finally, from within the LAPPS Grid one can publish annotations to the PubAnnotation repository, which links them to the canonical text alongside other contributed annotations over the document for use by others.

The potential for mutual exploitation of capabilities between the LAPPS Grid and GOST are considerable. We can, for example, compare results from the respective query engines

[28] https://getbootstrap.com/

[29] https://fetch.spec.whatwg.org/

and potentially exploit both to achieve maximal results. It is also possible to "decompose" GOST into its components so as to allow for alternative part-of-speech taggers to provide GOST input. In future work, we intend to pursue these capabilities as well as explore the potential to use GOST output as input to sophisticated text mining tools.

5. Results and Evaluation

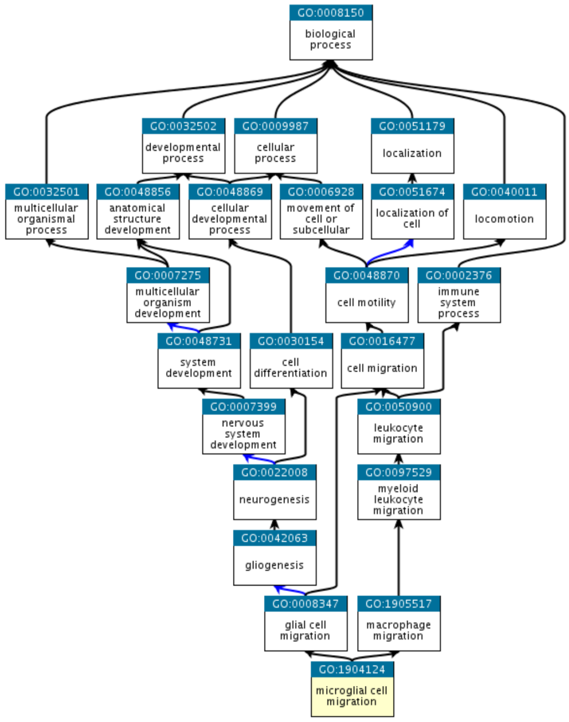


Figure 4: Gene Ontology Term 1 Example

To demonstrate the benefit of using the multiple levels in the GOST annotation as described in Section 3.1., we examined two PubMed searches based on two GO terms to investigate the utility of expanding searches through the relationship tree of a term. Term 1 "Microglial Cell Migration" as in Figure 4, refers to microglial cells that remove cellular debris including dead neurons. This term is a biological process with multiple paths and at least 5 steps before it reaches the root. Term 2 "Synaptic Signalling" as in Figure 5 refers to a specific form of cell signalling involving a structure within neurons.

It is also a biological process with only two nodes back to the root the shortest which has only three steps. Searching PubMed for "Microglial Cell Migration" resulted in only 32 exact match results, whereas "Synaptic Signalling" resulted in 679 – which shows that the topics covered are broader in the later search. As you step up the tree to search for "Macrophage Migration" and "Cell-Cell Signalling" respectively the number of results increases to 5657 and 718. This shows that the level of annotation provides a degree of expansion where the closer you get to the root, the broader the search becomes.

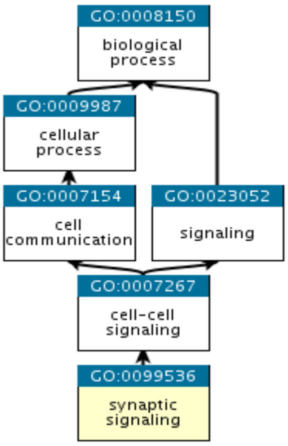
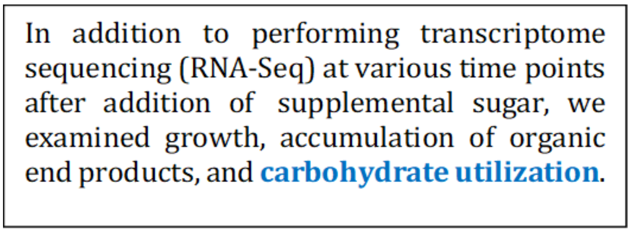


Figure 5: Gene Ontology Term 2 Example



In addition to performing transcriptome sequencing (RNA-Seq) at various time points after addition of supplemental sugar, we examined growth, accumulation of organic end products, and **carbohydrate utilization**.

Figure 6: Example Sentence

To demonstrate the advantage of incorporating GOST into the LAPPS Grid we ran the infrastructure using a sentence extracted from a biomedical article[30] (Figure 6). The GOST tagger extracted the multi-word-expression (MWE) phrase "carbohydrate utilization" as it exists in the OBO dictionary (GO:0009758). Figure 7 shows the OBO Graph of the MWE. Figure 8 shows the LAPPS Interchange Format (LIF) version of the GOST output when used to annotate the sentence in Figure 6. The LIF format of GOST output in Figure 8 can then be processed by tools that may generate additional annotation layers as mentioned in Section 4.. We should note that if the sets of GO IDs returned for MWEs are different in length, any overlaps are dealt with via existing USAS heuristics to prioritise the longest continuous spanning items.

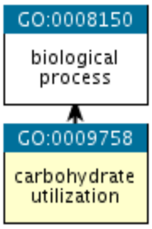
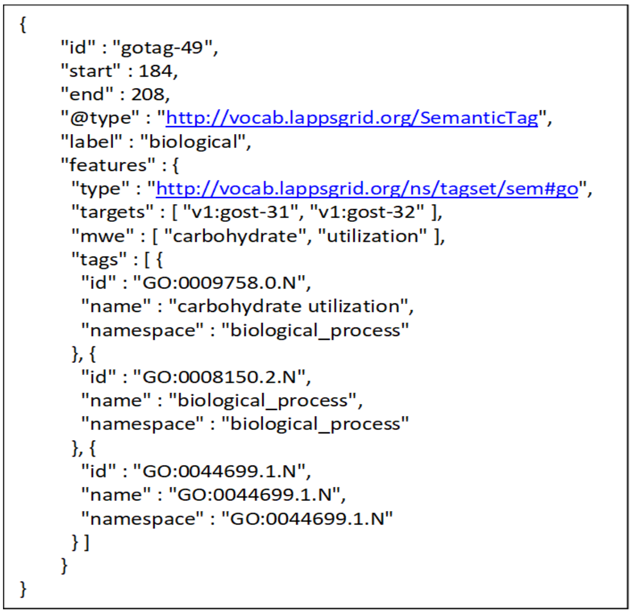


Figure 7: OBO Graph for "carbohydrate utilization"

5.1. Evaluating GOST on the CRAFT Dataset

The Colorado Richly Annotated Full-Text (CRAFT) is an independently annotated corpus. It is a collection of 97 full-length, open-access biomedical journal articles annotated semantically and syntactically to support research in biomedical NLP (Bada et al., 2012; Cohen et al., 2017b). The later version of CRAFT includes co-reference relations to deal with the challenges of false negatives extractions due to the failures in co-reference resolution (Cohen et al., 2017a).

[30] Article PMID: 30374459



```
{
      "id" : "gotag-49",
      "start" : 184,
      "end" : 208,
      "@type" : "http://vocab.lappsgrid.org/SemanticTag",
      "label" : "biological",
      "features" : {
        "type" : "http://vocab.lappsgrid.org/ns/tagset/sem#go",
        "targets" : [ "v1:gost-31", "v1:gost-32" ],
        "mwe" : [ "carbohydrate", "utilization" ],
        "tags" : [ {
          "id" : "GO:0009758.0.N",
          "name" : "carbohydrate utilization",
          "namespace" : "biological_process"
        }, {
          "id" : "GO:0008150.2.N",
          "name" : "biological_process",
          "namespace" : "biological_process"
        }, {
          "id" : "GO:0044699.1.N",
          "name" : "GO:0044699.1.N",
          "namespace" : "GO:0044699.1.N"
        } ]
      }
}
```

Figure 8: GOST LIF output for "carbohydrate utilization"

We downloaded the CRAFT annotated dataset from its GitHub repository[31]. The pre-processing of the data involves extracting the relevant data from the original *XML* format and presenting them in plain text format for the actual evaluation process. Each instance of the data contains a gene ontology ID followed by a word or phrase e.g.:

$$< GO : XXXXXXX >< word|phrase >$$

Item	Counts
No of articles	97
Concept entries	12,962
Concept lexicon	723
Concept types (GO ids)	721

Table 2: Basic statistics from the CRAFT Corpus

Table 2 shows that multiple concepts map to the same type or GO id. However, a closer look at the counts for *Concepts lexicon* and *Concepts types* (i.e. 723 and 721), indicates that there is actually a near 1-to-1 mapping between the concepts and their types. The only exception found is *GO:0051867* corresponding to 3 entries – '*general adaptation syndrome, behavioral process*', '*general adaptation syndrome*', '*behavioral process*' – which are basically the same concepts. The top 20 most common concepts (with their GO ids) are presented in Table 3 showing that the term 'gene expression' constitutes more than 25% of the entire concept entries. Figure 9 also indicates that more than two-thirds of the entire concept types are 2- or 3-word phrased e.g. *embryo development* or *meiotic nuclear division*.

GO ids	Concepts	Count
GO:0010467	gene expression	3704
GO:0065007	biological regulation	823
GO:0007608	sensory perception of smell	462
GO:0007567	parturition	425
GO:0009294	DNA mediated transformation	275
GO:0016265	death	210
GO:0008283	cell proliferation	209
GO:0006915	apoptotic process	176
GO:0009790	embryo development	150
GO:0007613	memory	141
GO:0007565	female pregnancy	136
GO:0008380	RNA splicing	135
GO:0000239	pachytene	129
GO:0007126	meiotic nuclear division	127
GO:0007618	mating	120
GO:0008152	metabolic process	118
GO:0007067	mitotic nuclear division	117
GO:0007612	learning	108
GO:0030154	cell differentiation	100
GO:0006281	DNA repair	97

Table 3: Distribution of Concept Types in CRAFT: Top 20 most common GO ids in the CRAFT dataset

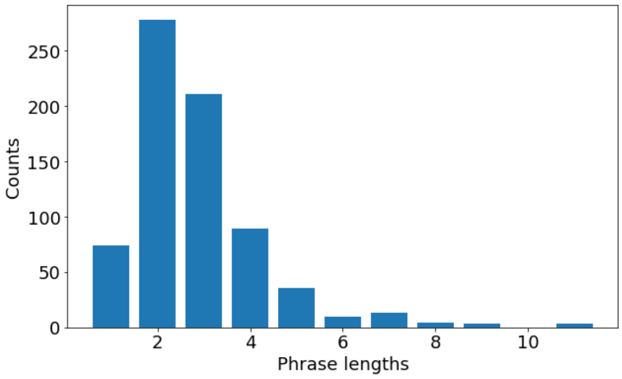


Figure 9: Distribution of the CRAFT Concepts by numbers words used to describe it

The evaluation method passes each instance from the extracted and untagged CRAFT (gold) dataset to the GOST and compares the returned output with the expected output in the tagged version. GOST returns a set of GO ids for each word (non-words or words without GO id tags are excluded) of the text given. For example, if we pass brain to GOST we will get something like:

brain GO:0048856 GO:0048513 GO:0032502 ...

The returned GO ids are ranked according to their likelihood of being predicted by GOST in that context and so '*GO:0048856*' is assumed to be the most likely. Therefore, for evaluation purpose, we implement different schemes:

Top *n*: If the correct GO id is among the top **n** predicted by GOST

Top ALL: If the correct GO id is in the list of all ids predicted by GOST

We used ***n*** values of *1,5,10,15* and *ALL*. For instance, **Top 5** checks whether the correct GO id is among the top **5** predicted by GOST. *Top 1* is the most strict and checks if the *first* GO id is the correct one. Also, if we pass a phrase instead (e.g. *brain development*), GOST returns:
[brain GO:0048856 GO:0048513 GO:0032502 ...]
[development GO:0048856 GO:0048513 GO:0032502 ...]

[31] `https://github.com/UCDenver-ccp/CRAFT`

We have a similar evaluation method but the GO ids in similar positions for each word are grouped together before applying the schemes. For example, we will pre-process the above result to look like:
[brain development (GO:0048856, GO:0048856) (GO:0048513, GO:0048513) ...]

Then we apply the same schemes by checking whether the CRAFT representation of *brain development* is found in the 1st tuple or within the first 5, 10 etc. tuples. Table 4 shows the evaluation scores for the schemes on the key metrics of *Accuracy*, *Precision*, *Recall* and *F1*.

Also, since GOST is expected to return a GO id for any concept given, we had to decide what the 'default' GO id will be when the correct one is not found at the top of the predicted list. Three default approaches were considered:

Default = Top predicted: selects the top predicted GO id when as default

Default = GOST most common selects the most common GO id predicted by GOST

Default = CRAFT most common selects the most common GO id in the CRAFT dataset

Top	Accuracy	Precision	Recall	F1
	Default == Top predicted			
Top 1	22.71	12.07	12.21	12.07
Top 5	62.55	30.51	30.65	30.51
Top 10	68.15	41.05	41.19	41.05
Top 15	68.95	43.97	44.11	43.97
Top All	81.29	62.60	62.60	62.60
	Default == GOST most common (GO:0008150)			
Top 1	22.76	12.07	12.21	12.07
Top 5	62.60	30.51	30.65	30.51
Top 10	68.20	41.05	41.19	41.05
Top 15	68.99	43.97	44.11	43.97
Top All	**81.35**	**62.69**	**62.83**	**62.69**
	Default == CRAFT most common (GO:0010467)			
Top 1	51.28	12.12	12.21	12.14
Top 5	62.54	30.43	30.51	30.46
Top 10	68.15	40.98	41.05	41.00
Top 15	68.94	43.89	43.97	43.92
Top All	81.29	62.64	62.69	62.66

Table 4: Evaluation of scores on *Accuracy*, *Precision*, *Recall* and *F1*

Across the metrics, Table 4 shows that there is a strong similarity between the set of results got from using the top predicted GO id and the most common GOST predicted GO id as the default. Although, the latter got slightly better results especially with the *Top All*, this trend is not entirely surprising given that the top predicted GO id is also produced by applying the GOST tagger.

As expected, the third approach (i.e. using the CRAFT most common GO id as the default tag) that leverages the knowledge of the GO id distribution in the CRAFT dataset gave a better performance for using the first predicted GO id. But the rest of the other schemes gave similar results as the previous schemes. This evaluation assumes a closed-world scenario where the evaluation is done with only the CRAFT dataset. We also used only the concepts in the 'biological process' subset which was closer to the lexicon integrated in GOST.

6. Conclusion

In order to support the large scale application of more advanced computational methods to biomedical and life sciences literature, we have created a novel super-infrastructure from a number of existing tools and developed a new pipeline scheduler in order to integrate the paper downloading, tagging, storage, retrieval and analysis phases. This combined open super-infrastructure permits flexible NLP annotation workflows and corpus linguistics analysis methods such as frequency listing, concordancing, keyness, collocation and n-grams to facilitate exploratory comparative analysis.

We have evaluated our new infrastructure in two ways. First, qualitatively, via performing PubMed searches and observing the effect of query expansion with the Gene Ontology. Second, quantitatively, using information retrieval metrics to compare the annotation performance of GOST itself against a manually annotated corpus.

We have utilised only open access PubMed Central archived papers and have focused on creating an open infrastructure by exploiting SOAP and REST APIs for connectivity across our distributed architecture. A crucial feature of the tagging process is the hierarchical nature of our semantic annotation which derives from the latest update of the Gene Ontology, thus facilitating searching and corpus comparisons with a meaningful domain specific semantic category set.

7. Acknowledgements

We are grateful for support for this research from a combination of sources. For the Biomedical Text Mining project at Lancaster University, we acknowledge support from a Wellcome Trust Seed funding grant (reference 204475/Z/16/Z). For more details about this research, see the project website: http://wp.lancs.ac.uk/btm/. We also acknowledge the support of Lancaster University's Data Science Institute in providing funding to host the workshop on Hypothesis Generating in Genetics and Biomedical Text Mining (HG2BTM) in January 2019, at which the seed was sown for our new more widely connected infrastructure.

References

Ananiadou, S., Kell, D. B., and Tsujii, J. (2006). Text mining and its potential applications in systems biology. *Trends in Biotechnology*, 24(12):571–579.

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner, W. A., Cohen, K. B., Verspoor, K., Blake, J. A., et al. (2012). Concept annotation in the craft corpus. *BMC bioinformatics*, 13(1):161.

Cohen, K. B., Lanfranchi, A., Choi, M. J.-y., Bada, M., Baumgartner, W. A., Panteleyeva, N., Verspoor, K., Palmer, M., and Hunter, L. E. (2017a). Coreference annotation and resolution in the colorado richly annotated full text (craft) corpus of biomedical journal articles. *BMC bioinformatics*, 18(1):372.

Cohen, K. B., Verspoor, K., Fort, K., Funk, C., Bada, M., Palmer, M., and Hunter, L. E. (2017b). The colorado richly annotated full text (craft) corpus: Multi-model annotation in the biomedical domain. In *Handbook of Linguistic Annotation*. Springer, pp. 1379–1394.

Coole, M., Rayson, P., and Mariani, J. (2015). Scaling out for extreme scale corpus data. In 2015 IEEE International Conference on Big Data (Big Data), pages 1643–1649. IEEE.

Coole, M., Rayson, P., and Mariani, J. (2016). lexidb: A scalable corpus database management system. In 2016 IEEE International Conference on Big Data (Big Data), pages 3880–3884. IEEE.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M. A., Saggion, H., Petrak, J., Li, Y., and Peters, W. (2011). Text Processing with GATE (Version 6). GATE.

El-Haj, M., Rayson, P., Piao, S., and Knight, J. (2018). Profiling medical journal articles using a gene ontology semantic tagger. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018).

Goecks, J., Nekrutenko, A., and Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome biology*, 11:R86.

Ide, N., Pustejovsky, J., Cieri, C., Nyberg, E., Wang, D., Suderman, K., Verhagen, M., and Wright, J. (2014). The language applications grid. In Nicoletta Calzolari (Conference Chair), et al., editors, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Ide, N., Suderman, K., Verhagen, M., and Pustejovsky, J. (2015). The Language Applications Grid Web Service Exchange Vocabulary. In Revised Selected Papers of the Second International Workshop on Worldwide Language Service Infrastructure - Volume 9442, WLSI 2015, pages 18–32, New York, NY, USA. Springer-Verlag New York, Inc.

Ide, N., Suderman, K., and Kim, J.-D. (2018). Mining Biomedical Publications With The LAPPS Grid. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), Paris, France, may. European Language Resources Association (ELRA).

Jin, Y., McDonald, R. T., Lerman, K., Mandel, M. A., Carroll, S., Liberman, M. Y., Pereira, F. C., Winters, R. S., and White, P. S. (2006). Automated recognition of malignancy mentions in biomedical literature. *BMC Bioinformatics*, 7(492).

Kann, M. G. (2007). Protein interactions and disease: computational approaches to uncover the etiology of diseases. *Briefings in Bioinformatics*, 8(5):333–346.

Kano, Y., Nguyen, N., Sætre, R., Yoshida, K., Fukamachi, K., Miyao, Y., Tsuruoka, Y., Ananiadou, S., and Tsujii, J. (2008). Towards Data And Goal Oriented Analysis: Tool Inter-Operability And Combinatorial Comparison. In Proceedings of the 3rd International Joint Conference on Natural Language Processing, pages 859–864, Hyderabad, India, January.

Kim, J.-D. and Wang, Y. (2012). Pubannotation - a persistent and sharable corpus and annotation repository. In BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing, pages 202–205, Montréal, Canada, June. Association for Computational Linguistics.

Kumar, A., Smith, B., and Borgelt, C. (2004). Dependence relationships between gene ontology terms based on tigr gene product annotations. In Proceedings of CompuTerm 2004: 3rd International Workshop on Computational Terminology.

Rak, R., Rowley, A., Black, W., and Ananiadou, S. (2012). Argo: An integrative, interactive, text mining-based workbench supporting curation. *Database*, 2012:bas010.

Srinivasan, P. (2001). Meshmap: a text mining tool for medline. In Proceedings of the AMIA Symposium, page 642. American Medical Informatics Association.

Tan, H. and Lambrix, P. (2009). Selecting an ontology for biomedical text mining. In Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing, pages 55–62. Association for Computational Linguistics.

Thompson, P., Ananiadou, S., and Tsujii, J. (2017). Handbook of linguistic annotation. Dordrecht, Netherlands:Springer, chapter The GENIA Corpus: Annotation Levels and Applications.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. In Advances in Informatics - 10th Panhellenic Conference on Informatics (PCI 2005), LNCS 3746, pages 382–392.

Verhagen, M., Suderman, K., Wang, D., Ide, N., Shi, C., Wright, J., and Pustejovsky, J. (2016). The LAPPS Grid Interchange Format. In Revised Selected Papers of the Second International Workshop on Worldwide Language Service Infrastructure - Volume 9442, WLSI 2015, pages 33–47, New York, NY, USA. Springer-Verlag New York, Inc.

Yakushiji, A., Tateisi, Y., Miyao, Y., and Tsujii, J.-i. (2000). Event extraction from biomedical papers using a full parser. In *Biocomputing 2001*. World Scientific, pp. 408–419.