Data and Text Mining

NLIMED: Natural Language Interface for Model Entity Discovery in Biosimulation Model Repositories

Yuda Munarko [1,∗], Dewan M. Sarwar [1], Koray Atalag,[1] and David P. Nickerson [1,∗]

[1] Auckland Bioengineering Institute, University of Auckland, Auckland, New Zealand

∗To whom correspondence should be addressed.

Abstract

Motivation: Semantic annotation is a crucial step to assure reusability and reproducibility of biosimulation models in biology and physiology. For this purpose, the COmputational Modeling in BIology NEtwork (COMBINE) community recommend the use of the Resource Description Framework (RDF). The RDF implementation provides the flexibility of model entity searching (e.g. flux of sodium across apical plasma membrane) by utilising SPARQL. However, the rigidity and complexity of SPARQL syntax and the nature of semantic annotation which is not merely as a simple triple yet forming a tree-like structure may cause a difficulty. Therefore, the availability of an interface to convert a natural language query to SPARQL is beneficial.

Results: We propose NLIMED, a natural language query to SPARQL interface to retrieve model entities from biosimulation models. Our interface can be applied to various repositories utilising RDF such as the PMR and Biomodels. We evaluate our interface by collecting RDF in the biosimulation models coded using CellML in PMR. First, we extract RDF as a tree structure and then store each subtree of a model entity as a modified triple of a model entity name, path, and class ontology into the RDF Graph Index. We also extract class ontology's textual metadata from the BioPortal and CellML and manage it in the Text Feature Index. With the Text Feature Index, we annotate phrases resulted by the NLQ Parser (Stanford parser or NLTK parser) into class ontologies. Finally, the detected class ontologies then are composed as SPARQL by incorporating the RDF Graph Index. Our annotator performance is far more powerful compared to the available service provided by BioPortal with F-measure of 0.756 and our SPARQL composer can find all possible SPARQL in the collection based on the annotation results. Currently, we already implement our interface in Epithelial Modelling Platform tool.

Availability: https://github.com/napakalas/NLIMED

Contact: ymun794@aucklanduni.ac.nz, d.nickerson@auckland.ac.nz

1 Introduction

The Resource Description Network (RDF) is a standard data model used in almost all semantically annotated biosimulation models in the Physiome Repository Model (PMR) (Yu *et al.*, 2011) and BioModels (Chelliah *et al.*, 2015). These RDF annotated models, then, can be discovered for its model entities such as variables, components, mathematical formula, reactions, compartments, species, and events. Thus, this discoverability supports communities in biology and physiology through reusability and reproducibility. Currently, the utilisation of the RDF in biosimulation models semantic annotation has been formalised as the only standard by the COmputational Modeling in BIology NEtwork (COMBINE) community (Neal *et al.*, 2019).

We can now leverage SPARQL, a standard query language to retrieve information from data encoded using RDF, to discover model entities. SPARQL is a simple, easy-to-understand and good-performing language, especially for triple (subject, predicate, and object) searches in the RDF (Pérez *et al.*, 2009). However, for our purpose, SPARQL becomes complicated, since model entities can be annotated compositely by several class ontologies connected by a series of predicates and objects, creating a tree structure. Hence, knowledge in the class ontologies and tree structure related to the model itself is critical, so maybe only experts can create SPARQL that suits their information needs. Therefore, a simple interface translating information needs in Natural Language Query (NLQ) to SPARQL is required.

To address the above challenge generally, NLQ to SPARQL translation applies two processes: NLQ pre-processing and SPARQL generation (Hamon *et al.*, 2014; Zou *et al.*, 2014; Yahya *et al.*, 2012; Lopez *et al.*, 2013). The aim of NLQ pre-processing is to identify subjects or predicates or objects and the intention of the question, while SPARQL generation is to construct SPARQL based on NLQ pre-processing result. Most of NLQ pre-processing has utilised Natural Language Processing (NLP) such as in several works in Question Answering over Linked Data (QALD-4) challenge focusing on biomedical, e.g. RO_FII (Unger *et al.*, 2014), GFMed (Marginean, 2014), and POMELO (Hamon *et al.*, 2016) and Multilanguage data, e.g. Xser (Xu *et al.*, 2014). GFMed was based on a set of self-determined grammar rules for a group of natural language patterns, making it less suitable for NLQ that do not fit the patterns. POMELO and Xser have provided a more flexible approach by exploiting information from target RDF databases to abstract NLQ so that it can handle a broader range of NLQ types. All these works are useful to querying RDF databases consisting of simple triple or interlinked structure such as DBPedia, Drugbank, Sider, Diseasome. For complex structure such as in biosimulation models, we adapt NLP to recognised phrases associated with class ontologies, rather than with subjects, predicates, and objects.

In biosimulation model communities, SemGen, a model annotation tool (Neal *et al.*, 2018), suggested the BioPortal service NCBO Annotator (Jonquet *et al.*, 2009), for easy and fast searching of class ontologies. This approach works well, but by using NCBO Annotator only we do not consider local data in biosimulation models that are likely to cause misidentification. Another tool, Epithelial Modelling Platform (EMP), a visual web interface to create a new epithelial transport model based on the available models (Sarwar *et al.*, 2019), utilised a template-based search interface (Sarwar *et al.*, 2018) to find the appropriate model entities. For this specific type of searching, the template-based interface can perform properly and is easy to deploy; however, for our purpose, it needs the development of numerous templates for different biosimulation model types.

Hence, we introduce here NLIMED (Natural Language Interface for Model Entity Discovery), a natural language interface for searching semantically annotated model entities from biosimulation models. Initially, we develop NLQ Annotator for annotating NLQ to a set of class ontologies by utilising NLQ parser and key information in BioPortal (Whetzel *et al.*, 2011) and biosimulation model. For fast annotation, we organise the key information as global and local features in the Text Feature Index. Then, we develop SPARQL Generator for generating SPARQL from a set of class ontologies. The vital part of SPARQL Generator is the RDF Graph Index storing the relationship of model entity's predicates, objects, class ontologies, and textual information. By the RDF Graph Index, we can construct all possible SPARQL based on provided class ontologies.

We evaluate NLIMED on the PMR collection (Yu *et al.*, 2011) by collecting all CellML files annotated using the RDF and then creating the RDF Graph Index and the Text Feature Index. We demonstrate that NLIMED translates NLQ to SPARQL that can be superior compared to the existing approach and has faster execution time. NLIMED is ready to be implemented and can accommodate different biosimulation model platforms, such as CellML (Cuellar *et al.*, 2003), and SBML (Hucka *et al.*, 2003) so it is possible to create generic model entities discovery tool. Currently, we have implemented NLIMED in EMP to optimise its user experience when searching for model entities (Sarwar *et al.*, 2019). We provide our implementation and experiment setup freely accessed at https://github.com/napakalas/NLIMED.

2 Materials and methods

Our interface consists of two primary modules, NLQ Annotator and SPARQL Generator (Figure 1). Both modules are based on data collected from the PMR (Yu *et al.*, 2011) and BioPortal (Whetzel *et al.*, 2011). We also utilise natural language parser provided by Stanford CoreNLP (Manning *et al.*, 2014) and NLTK (Bird *et al.*, 2009).

2.1 The Physiome Model Repository and BioPortal

The PMR contains more than 800 CellML biosimulation models in which around 20% have been semantically annotated with RDF. Within these annotated models, there are 4,671 model entities, where each model entity forms a tree structure with its name as root and description and class ontologies as leaves (Figure 2). Further, we develop RDF Graph Index (RGI) described at 2.3 based on model entity structure and Text Feature Index (TFI) explained at 2.2 based on the description inside model entities and information inside class ontologies.

The number of distinct leaves is 3,472, and the number of paths between roots and leaves is 29,755. For each class ontology leaf, we extract textual features from BioPortal. We choose BioPortal among other ontology service providers because of its completeness covering all types of ontology found in the PMR, e.g. Chemical Entities of Biological Interest (ChEBI), Ontology of Physics for Biology (OPB), Foundational Model of Anatomy (FMA), Protein Ontology (PR), Gene Ontology (GO), and Uber Anatomy Ontology (UBERON). Moreover, BioPortal provides NCBO Annotator (Jonquet *et al.*, 2009), which is useful for NLIMED performance comparator.

2.2 Natural Language Query (NLQ) Annotator

NLQ Annotator module comprises TFI, Natural Language Query (NLQ) Parser, and Phrase Annotator. TFI holds a vector space model (Salton *et al.*, 1975) describing class ontologies such as http://purl.obolibrary.org/obo/CHEBI_29103 (potassium(1+)) and http://purl.obolibrary.org/obo/FMA_84666 (apical plasma membrane). We apply terms in global metadata in BioPortal and local metadata in the related model entity to represent every class ontology. From global metadata, we differentiate terms into features of preferred label, synonym, and definition, while from local metadata, we only have description feature. These features, then, will be used to compute the weight of a term associated with a class ontology. For fast retrieval, TFI adapts an inverted index concept (Harman *et al.*, 1992) to manage terms and its relationship to class ontologies, and features.

The Natural Language Query (NLQ) annotation process is started by parsing of an NLQ using NLQ Parser. In this work, we utilise and compare the performance of Stanford parser (Manning *et al.*, 2014) and NLTK parser (Bird *et al.*, 2009). NLQ Parser works to get candidate phrases in NLQ; all of which are possible noun phrases. For example, the NLQ 'the concentration of potassium in the portion of tissue fluid', if parsed using Stanford parser, the candidate phrases will be CP_1 = 'concentration potassium portion tissue fluid', CP_2 = 'concentration', CP_3 = 'potassium', CP_4 = 'portion tissue fluid', CP_5 = 'portion', and CP_6 = 'tissue fluid'. The candidate phrases may overlap, whereas, we are interested in finding a set of phrases associated with class ontologies with the highest weight and most extended terms without overlapping terms and fully cover the NLQ. Using Phrase Annotator, we calculate the association weight of each candidate phrase to each candidate class ontology. First, we have Equation (1) to get the global association weight between a phrase and a class ontology, W_{global},

$$W_{global} = \sum_{i=term\in phrase}^{n} \alpha\frac{p_i}{lp_i+nt} + \beta\frac{s_i}{ls_i+nt} + \gamma\frac{d_i}{(ld_i+nt)N} \quad (1)$$

where p_i, s_i, and d_i are the appearance of the term in features of preferred label, synonym, and definition consecutively. The values of these variables are 1 or 0 for present and absent, respectively. $lp_i + nt$, $ls_i + nt$, and $(ld_i + nt)N$ are smoothing denominator to normalise the contribution of occurred terms, lp_i, ls_i, and ld_i are the number of terms in features of preferred label, synonym, and definition, nt is the number of terms in the phrase, and N is the number of class ontologies having the term. Then, we compute local association weight, W_{local}, with Equation (2),

$$W_{local} = \sum_{i=term\in phrase}^{n} \delta\frac{f_i}{lf_i+nt} . \ln\frac{S}{S-ts_i} \quad (2)$$

where f_i is indicating the appearance of the term in description feature, $lf_i + nt$ is smoothing denominator, and lf_i is the number of terms in description feature. We implement Inverse Document Frequency (IDF) $\ln\frac{S}{S-ts_i}$, where S is the number of model entities in the collection and ts_i is the number of model entities having the term. The logic of IDF is that a term occurring in many model entities is less important than other terms with less occurrence (Salton and Yang, 1973). Finally, we total W_{global} and W_{local} to get the association weight of a candidate phrase to a candidate class ontology, W_{co}, as Equation (3).

$$W_{co} = W_{global} + W_{local} \quad (3)$$

We apply multiple weighting scenario based on features (Ogilvie *et al.*, 2003; Robertson *et al.*, 2004), so we have multipliers α, β, γ, and δ in Equation (1) and Equation (2) to determine the level of importance of the term in preferred label, synonym, definition, and description. The values of these multipliers are decided empirically to get the best performance. In our experiment, one of the best configuration for Stanford parser is $\alpha = 4, \beta = 0.7, \gamma = 0.5$, and $\delta = 0.8$, while for NLTK parser is $\alpha = 3$, $\beta = 0.1$, $\gamma = 0.1$, and $\delta = 0.1$. The number of class ontologies for each

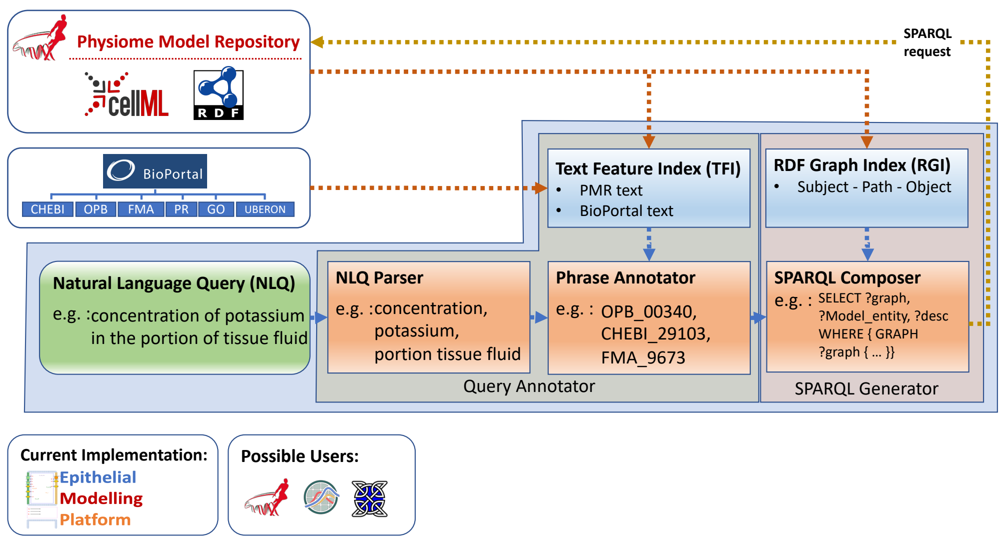
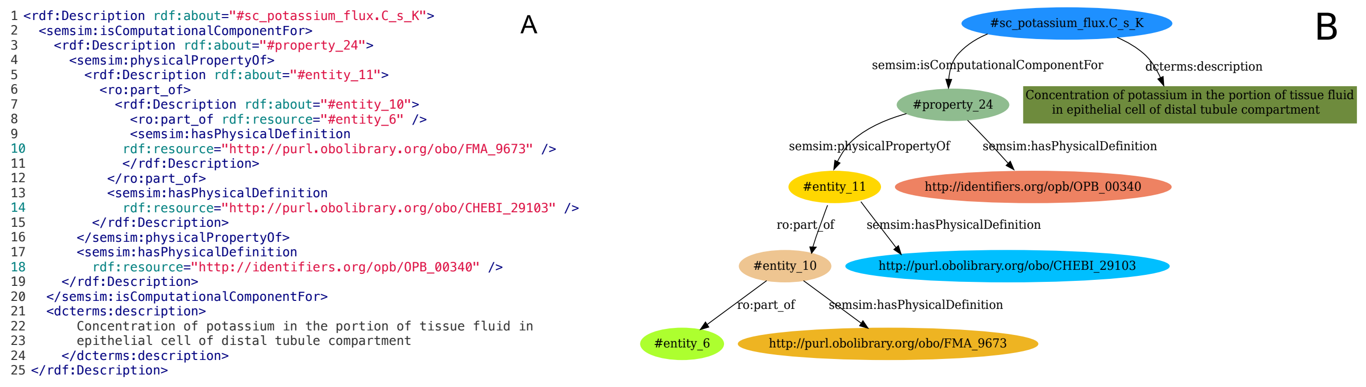

Physiome Model Repository
cellML
RDF
SPARQL request
BioPortal
CHEBI
OPB
FMA
PR
GO
UBERON
Text Feature Index (TFI)
• PMR text
• BioPortal text
RDF Graph Index (RGI)
• Subject - Path - Object
Natural Language Query (NLQ)
e.g. :concentration of potassium in the portion of tissue fluid
NLQ Parser
e.g. :concentration, potassium, portion tissue fluid
Phrase Annotator
e.g. : OPB_00340, CHEBI_29103, FMA_9673
SPARQL Composer
e.g. : SELECT ?graph, ?Model_entity, ?desc WHERE { GRAPH ?graph { ... }}
Query Annotator
SPARQL Generator
Current Implementation:
Epithelial Modelling Platform
Possible Users:

Fig. 1. NLIMED workflow. First, we create a Text Feature Index (TFI) and an RDF Graph Index (RGI) developed based on data on the PMR and BioPortal. Natural Language Query (NLQ), initially, is annotated into class ontologies in Query Annotator module, then, translated into SPARQL in SPARQL Generator module.



```
<rdf:Description rdf:about="#sc_potassium_flux.C_s_K">
  <semsim:isComputationalComponentFor>
    <rdf:Description rdf:about="#property_24">
      <semsim:physicalPropertyOf>
        <rdf:Description rdf:about="#entity_11">
          <ro:part_of>
            <rdf:Description rdf:about="#entity_10">
              <ro:part_of rdf:resource="#entity_6" />
              <semsim:hasPhysicalDefinition
             rdf:resource="http://purl.obolibrary.org/obo/FMA_9673" />
             </rdf:Description>
           </ro:part_of>
           <semsim:hasPhysicalDefinition
             rdf:resource="http://purl.obolibrary.org/obo/CHEBI_29103" />
         </rdf:Description>
       </semsim:physicalPropertyOf>
       <semsim:hasPhysicalDefinition
         rdf:resource="http://identifiers.org/opb/OPB_00340" />
     </rdf:Description>
   </semsim:isComputationalComponentFor>
   <dcterms:description>
      Concentration of potassium in the portion of tissue fluid in
      epithelial cell of distal tubule compartment
     </dcterms:description>
 </rdf:Description>
```

B
#sc_potassium_flux.C_s_K
semsim:isComputationalComponentFor
dcterms:description
#property_24
Concentration of potassium in the portion of tissue fluid in epithelial cell of distal tubule compartment
semsim:physicalPropertyOf
semsim:hasPhysicalDefinition
#entity_11
http://identifiers.org/opb/OPB_00340
ro:part_of
semsim:hasPhysicalDefinition
#entity_10
http://purl.obolibrary.org/obo/CHEBI_29103
ro:part_of
semsim:hasPhysicalDefinition
#entity_6
http://purl.obolibrary.org/obo/FMA_9673

Fig. 2. An annotation of model entity of concentration of potassium located in a CellML describing a mathematical model of the renal distal tubule (Chang and Fujita, 1999). (A) An RDF/XML code describing the model entity. (B) A tree structure representing the RDF code describing the model entity. The relevant information in this RDF code is the description and all class ontologies.

candidate phrase can be more than one, but in this work, we choose the highest association weight, although we can also consider all classes.

Using the specified weighing formula, the candidate phrases of the example NLQ are annotated into class ontologies CO in http://purl.obolibrary.org/obo/ and weighed into W_{co} as $(CO_1, W_{co_1}) = ('FMA\_9673', 0.168)$, $(CO_2, W_{co_2}) = ('OPB\_00340', 0.137)$, $(CO_3, W_{co_3}) = ('CHEBI\_29103', 0.145)$, $(CO_4, W_{co_4}) = ('FMA\_9673', 0.223)$, $(CO_5, W_{co_5}) = ('FMA\_66836', 0.133)$, $(CO_6, W_{co_6}) = ('FMA\_9673', 0.188)$. Considering the highest association weight, the longest phrase, and the completeness, $CO_2 =' OPB\_00340'$ (concentration), $CO_3 =' CHEBI\_29103'$ (potassium), and $CO_4 =' FMA\_9673'$ (portion tissue fluid) are selected to be composed as SPARQL.

2.3 SPARQL Generator

Along with the result of NLQ Annotator, SPARQL Generator constructs SPARQL utilising RGI and SPARQL Composer. RGI is a representation of biologically annotated model entities using the RDF. As can be seen in Figure 2, an RDF annotated model entity has a model entity name as root and class ontologies or description objects as leaves. Between root and leaves, there are other objects, e.g. '#property_24' and 'entity_11, and predicates, e.g. 'semsim:hasPhysicalDefinition' and 'semsim:physicalPropertyOf'. However, to construct SPARQL for model entity discovery, objects other than leaves are not necessarily important; hence, we only consider a triple of a model entity name, set of predicates assembling a path, and class ontology or description which further named as root, path, and leaf consecutively. In general, RGI provides two map structures, $h_1 : L \to R$, where L is a set of leaves and R is a set of roots, and $h_2 : (L, R) \to P$, where (L, R) is a set of leaf and root pairs and P is a set of paths. We find that there is a large amount of repetition of data in paths and leaves such as namespaces, class ontologies, and predicates. Therefore, we decided to fully normalize the maps by dividing it into several maps to ensure data consistency and reduce space requirements.

Utilising RGI's maps, SPARQL Composer compiles the class ontologies resulted by NLQ Annotator, generating SPARQL. First, we look for candidate roots $\bar{R}$ having same class ontologies $\bar{L}$, $\bar{R} = \bigcap_{l \in \bar{L}} h_1(l)$. Then, using the pairs of $\bar{L}$ and $\bar{R}$, we extract paths $\bar{P}$ connecting $\bar{L}$ and $\bar{R}$, $\bar{P} = h_2(\bar{L}, \bar{R})$. Next, we construct the SPARQL syntax for each $p \in \bar{P}$, and finally, combine the constructed syntax with the same r into a new SPARQL. Hence, this approach may result in zero or more SPARQL based on the number of $\bar{R}$.

3 Experiments and results

We ran experiments to measure the performance of NLIMED in translating NLQ into SPARQL. We prepare data test examined by experts consisting of 51 NLQs with different complexity, having one to 28 terms and one to six phrases. All NLQs have been separated into phrases and annotated into class ontologies. We aim to provide more NLQ with the number of terms between one and five and the number of phrases between one and three; this is because, naturally, queries with natural language tend to be short (Jansen *et al.*, 2000; Yi *et al.*, 2006). SPARQL Generator is able to generate all possible SPARQL based on class ontologies provided by NLQ Annotator as long as all class ontologies found in at least one model entity.

3.1 NLQ Annotator performance

We measure the performance of NLQ Annotator using precision, recall, and F-Measure. Precision shows the proportion of correct annotation to the entire annotated phrases, while recall presents the proportion of correct annotation to the whole phrases should be annotated. F-measure, on the other hand, harmonises precision and recall by its mean. Another measurement is query accuracy, where NLQ annotation is considered correct if all related class ontologies are found. The final measurement is execution time averaging from ten runs of all NLQs in data test. To get the best results from NCBO Annotator, we limit this annotator to recognise only seven ontologies at the PMR and prioritise the most prolonged- phrase.

Table 1. The values of precision, recall, F-Measure, query accuracy, and rate of execution time for data test of 51 NLQ in second of NLQ Annotator using Stanford parser and NLTK parser, and NCBO Annotator.

Method	Precision	Recall	F-measure	Query accuracy	Exec time
NLQ Annotator + Stanford parser	0.744	0.768	0.756	0.549	0.532
NLQ Annotator + NLTK parser	0.591	0.728	0.652	0.333	0.101
NCBO Annotator	0.402	0.376	0.388	0.196	36.697

From Table 1 we see that NLQ Annotator has a better performance than NCBO Annotator in all measurement types. The use of the Stanford parser is superior to the use of the NLTK parser, which shows that our approach is fully working, although it depends on the accuracy of the parser. However, the query accuracy value is still relatively low, 0.333 and 0.549 for the use of NLTK parsers and Stanford parser, respectively. We find that this imperfection happens for a complex or long NLQ containing more than three phrases. However, annotation imperfection is not directly related to the failure of SPARQL compilation, the shortcomings and the excess number of recognised class ontologies only cause changes in the restrictiveness of SPARQL. In terms of execution time, the use of Stanford parser requires five times more time than NLTK Parser. Notwithstanding, the use of Stanford parser is still relatively fast because for one NLQ it takes about 0.01 seconds. NCBO Annotator's slow performance is due to the need to access the BioPortal server using the Internet.

Considering the NLQ complexity, we present F-measure graphs for NLQ Annotator and NCBO Annotator in Figure 3. Generally, F-measure of NLQ Annotator is higher than NCBO Annotator except when the number of terms is five and twelve. It is because NCBO Annotator works well for NLQ with one phrase having many terms, whereas, NLQ Annotator tends to divide this NLQ into several phrases. Nevertheless, the use of high-performance parsers, such as Stanford parser, may overcome this problem. As many studies have shown that queries with natural language are generally short (Jansen *et al.*, 2000; Yi *et al.*, 2006), we believe that this tendency also applies to the search for model entities, therefore, paying attention to this NLQ type is a practical choice. Furthermore, NLQ Annotator performance declines when implementing NLTK parser and annotating NLQ with six phrases while NCBO Annotator suffers deficient performance when annotating NLQ with two phrases or two until three terms.

3.2 The role of features

There are four features used to measure the weight of a phrase to a class ontology, including preferred label, synonym, definition, and description found inside the PMR. We find that preferred label is the most important feature; its use only could reach F-measure of 0.466 and 0.365 for the NLIMED with Stanford parser or NLTK parser, respectively. Hence, our attention shift to the role of other features used together with preferred label ($\alpha = 4$) with Stanford parser, which shows that features of description, synonym, and description increase F-measure to more than 0.7, 0.519, and 0.566 respectively (Figure 4). Further, the use of all features might increase the F-measure to 0.756 when the synonym multiplier (δ) is between 0.6 and

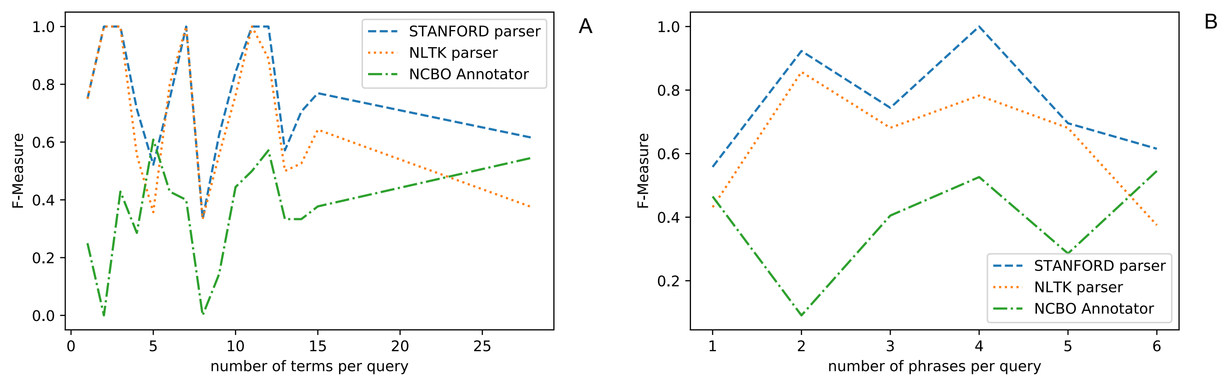


Fig. 3. The comparison of F-Measure of the use of Stanford parser and NLTK parser, and NCBO Annotator based on NLQ complexity. (A) F-measure based on the number of terms inside NLQ. (B) F-measure based on the number of phrases inside NLQ.

1.0, the definition multiplier (γ) is between 0.0 and 0.4, and the description multiplier (δ) is between 0.6 and 1.0 (Figure 4).

4 Discussion

4.1 NLQ to SPARQL evaluation

Although NLIMED can translate NLQ to SPARQL with excellent performance, there are some conditions where NLIMED does not work optimally and needs improvement. In the other side, the use of NCBO Annotator is promising for very long NLQ; however, it lacks the contribution of biosimulation model repositories, so it solely use is insufficient for model entity discovery. Further, we discuss the NLIMED based on the NLQ complexity.

Short NLQ (1-3 terms, 1-2 phrases): NLIMED can reach high F-measure. NLQ Annotator correctly annotates NLQs such as ’potassium’, ’apical plasma membrane’, ’basolateral membrane’, and ’flux of sodium’, ’van’t Hoff law’. However, it failed to annotate ’sodium flux’ which is similar to ’flux of sodium’. The problem is that while ’flux of sodium’ can be divided into two phrases, ’flux’ and ’sodium’, ’sodium flux’ is recognised as a single phrase. The use of NCBO Annotator even worse because it tends to ignore phrases related to OPB ontology such as ’flux’ and ’concentration’. Furthermore, for phrases with monoatomic monocation meaning such as sodium (Na+) or potassium (K+) related to ChEBI, NCBO Annotator more often associates it with atomic meaning such as sodium atom or potassium atom.

Complex NLQ (4-6 terms, 1 phrase): the F-measure of NLQ Annotator cannot compete NCBO Annotator. The low performance of NLQ Annotator is mostly due to the absence of NLQ’s phrase in the PMR. For example, ’sodium/glucose cotransporter 4 (mouse)’ which is not contained by the PMR is annotated incorrectly to ’Protein Ontology: PR_000015165’ describing ’sodium/glucose cotransporter 1’ while NCBO Annotator can annotate correctly to ’Protein Ontology: PR_000015175’. However, this inaccuracy is not a weakness; instead, this is an advantage that shows that NLQ Annotator can adjust to the availability of data, whereas, the accuracy of NCBO Annotator can cause the inability of SPARQL Generator to find the appropriate SPARQL.

Long NLQ (> 2 phrases): NLQ Annotator’s performance is far better than NCBO Annotator for NLQ with three to five phrases, but it is almost similar for NLQ with six phrases. Generally, long NLQ can be annotated appropriately, e.g. ’concentration of sodium in the portion of tissue fluid in epithelial cell of distal tubule’ and ’flux of IP3 receptor through P2Y2 purinoceptor and apical plasma membrane’. The incorrect annotation usually is partial for one or two phrases by the division of a phrase into two phrases or the ignoration of a phrase. While the division of a phrase implicates to the generation of more specific or inaccurate SPARQL, the ignoration of a phrase leads to the generation of more general SPARQL.

Question type NLQ: NLQs such as ’I want a model of SGLT1 in apical membrane’ and ’give me a model of glucose transporter’ are correctly annotated, but there is an additional class ontology related to ’a model’ phrase. The ’a model’ is the intention of the question, so by applying another NLP approach, this can be handled appropriately and implemented for question and answer system. Nevertheless, for our purpose, since ’a model’ is frequently found in biosimulation models, we may consider terms in this phrase as stop words. Alternatively, preventing the same problem with different phrases, we may apply an IDF threshold to determine whether or not to annotate a phrase.

We find that the Phrase Annotator can choose the correct class ontology almost excellently when provided with the correct phrases. It means that using the features of preferred label, synonym, definition, and description together with our weighing scenario can work satisfactorily. We also find that SPARQL Generator may generate all possible SPARQL based on the available class ontologies. Besides, to improve NLIMED performance, it is necessary to create a special parser for the discovery of biosimulation models and model entities with specific data models.

4.2 Possible implementation

Currently, we develop NLIMED over CellML in the PMR (Yu et al., 2011) and implement it in EMP (Sarwar *et al.*, 2019) as an additional interface discovering model entities and being able to meet EMP requirements. We are confident that NLIMED may also be developed over BioModels (Chelliah *et al.*, 2015) since it contains models richly annotated with the RDF utilising class ontologies. NLIMED is potential to applied on EMP like tools such as the Cardiac Electrophysiology Web Lab (Cooper *et al.*, 2016), eSolv (de Boer *et al.*, 2017), and PhysioMaps (Cook *et al.*, 2013). Present annotation tools, e.g. SemGen (Neal *et al.*, 2018), OpenCOR (Garny and Hunter, 2015), and Saint (Lister *et al.*, 2009), which can provide class ontologies suggestion based on the available ontology databases may take advantage of NLIMED. Following the COMBINE recommendation about standardisation of biosimulation model annotation (Neal *et al.*, 2019), this work can be directed to provide a comprehensive search interface to discover model entities from various biosimulation model repositories .

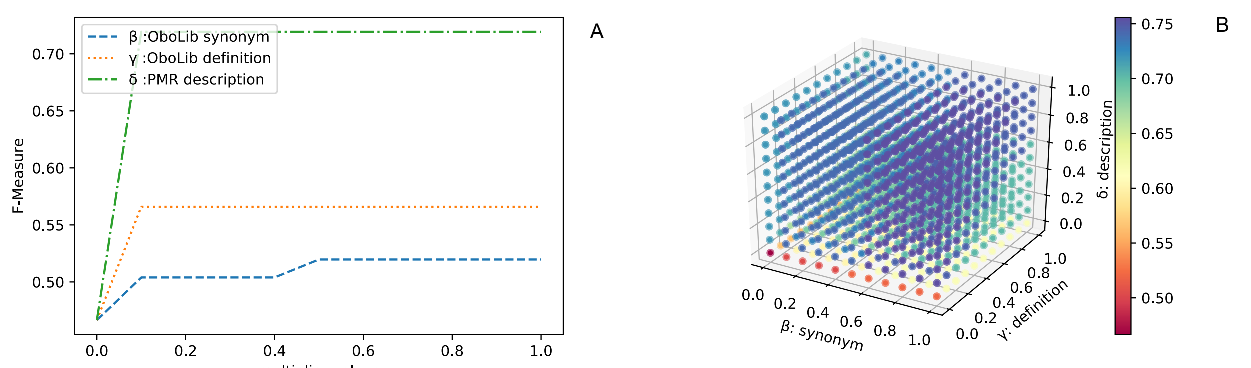


Fig. 4. The role of features of synonym (β), definition (γ), and description (δ) when used with preferred label ($\alpha = 4$) and Stanford Parser in NLQ Annotator measured using F-Measure. (A) NLIMED performance when uses two features, pair of preferred label and another feature. (B) NLIMED performance when uses all features.

4.3 Limitations and future directions

There are several limitations of NLIMED, some of which will direct our future work. While NLIMED can identify the logical-semantics in NLQ and then classify into class ontologies, we are not yet exploring the lexical-semantics; consequently, we cannot differentiate between NLQ such as 'flux of potassium to the portion of cytosol' and 'flux of potassium from the portion of cytosol'. We suppose that the lexical-semantic in NLQ is closely related to the model entity's tree structure, i.e. paths connecting root and leaves. Therefore, it will be beneficial to understand semantic concepts related to paths and use them with the lexical-semantic of NLQ.

Another limitation is that SPARQL Generator produces all possible SPARQL without any selection criteria, so NLIMED presents all model entities retrieved by all generated SPARQL ignoring relevancy order. We may adapt text-based ranking approaches such as weighing in Phrase Annotator, BM25 (Robertson *et al.*, 2004), and term-weighting (Salton and Buckley, 1988) for SPARQL selection and model entities ranking. However, for this purpose, the ranking approaches should consider non-textual features, such as class ontologies and semantic concepts.

In the future, we anticipate improving NLIMED performance by accommodating lexical-semantic of NLQ along with model entitiesâĽ™ semantic concepts.

5 Conclusion

We demonstrated NLIMED, an interface to translate NLQ into SPARQL consisting of NLQ Annotator and SPARQL Generator, for model entities discovery. NLQ Annotator can identify class ontologies in NLQ utilising global features extracted from BioPortal (preferred label, synonym, and definition) and local feature extracted from biosimulation models' RDF (definition). The class ontologies identification performance is quite high, reaching F-measure of 0.756, but it can be further improved by applying better NLQ Parser. We also showed that NLIMED could handle a wide range of NLQ types containing one or many terms with one or many phrases. Our SPARQL Generator storing RDF graph as indexes can generate all possible SPARQL based on provided class ontologies. The execution time of NLIMED is reasonably fast, takes around 0.002 to 0.01 second for a single NLQ depend on the type of parser (NLTK parser or Stanford parser). NLIMED presently has been implemented in EMP for model entities searching from the PMR and is possibly applied as a generic search interface exploring model entities from numerous biosimulation model repositories, e.g. the PMR and BioModels. Further, we interest to explore lexical-semantic inside NLQ and semantic concept inside model entities to increase NLIMED performance and its use for question and answer system. We believe that NLIMED will be useful for biosimulation model communities and support reusability and reproducibility of biosimulation models.

Acknowledgements

Funding

Aotearoa Foundation Physiome Fellowship to DPN.

References

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition.

Chang, H. and Fujita, T. (1999). A numerical model of the renal distal tubule. *Am. J. Physiol.*, **276**(6), F931–951.

Chelliah, V., Juty, N., Ajmera, I., Ali, R., Dumousseau, M., Glont, M., Hucka, M., Jalowicki, G., Keating, S., Knight-Schrijver, V., Lloret-Villas, A., Natarajan, K. N., Pettit, J.-B., Rodriguez, N., Schubert, M., Wimalaratne, S. M., Zhao, Y., Hermjakob, H., Le NovÃ¨re, N., and Laibe, C. (2015). BioModels: ten-year anniversary. *Nucleic Acids Res.*, **43**(Database issue), D542–548.

Cook, D. L., Neal, M. L., Hoehndorf, R., Gkoutos, G. V., and Gennari, J. H. (2013). Representing physiological processes and their participants with PhysioMaps. *J Biomed Semantics*, **4 Suppl 1**, S2.

Cooper, J., Scharm, M., and Mirams, G. R. (2016). The Cardiac Electrophysiology Web Lab. *Biophys. J.*, **110**(2), 292–300.

Cuellar, A. A., Lloyd, C. M., Nielsen, P. F., Bullivant, D. P., Nickerson, D. P., and Hunter, P. J. (2003). An Overview of CellML 1.1, a Biological Model Description Language. *SIMULATION*, **79**(12), 740–747.

de Boer, T. P., van der Werf, S., Hennekam, B., Nickerson, D. P., Garny, A., Gerbrands, M., Bouwmeester, R. A. M., Rozendal, A.-P., Torfs, E., and van Rijen, H. V. M. (2017). eSolv, a CellML-based simulation front-end for online teaching. *Advances in Physiology Education*, **41**(3), 425–427.

Garny, A. and Hunter, P. J. (2015). OpenCOR: a modular and interoperable approach to computational biology. *Front. Physiol.*, **6**.

Hamon, T., Grabar, N., Mougin, F., and Thiessard, F. (2014). Description of the POMELO System for the Task 2 of QALD-2014. In *CLEF*.

Hamon, T., Grabar, N., and Mougin, F. (2016). Querying biomedical Linked Data with natural language questions. *Open Journal Of Semantic Web*, **0**, 1 – 19.

Harman, D., Fox, E. A., Baeza-Yates, R. A., and Lee, W. C. (1992). *Inverted Files*.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., Arkin, A. P., Bornstein, B. J., Bray, D., Cornish-Bowden, A., Cuellar, A. A., Dronov, S., Gilles, E. D., Ginkel, M., Gor, V., Goryanin, I. I., Hedley, W. J., Hodgman, T. C., Hofmeyr, J.-H., Hunter, P. J., Juty, N. S., Kasberger, J. L., Kremling, A., Kummer, U., Le NovÃ¨re, N., Loew, L. M., Lucio, D., Mendes, P., Minch, E., Mjolsness, E. D., Nakayama, Y., Nelson, M. R., Nielsen, P. F., Sakurada, T., Schaff, J. C., Shapiro, B. E., Shimizu, T. S., Spence, H. D., Stelling, J., Takahashi, K., Tomita, M., Wagner, J., and Wang, J. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**(4), 524–531.

Jansen, B. J., Spink, A., and Saracevic, T. (2000). Real life, real users, and real needs: a study and analysis of user queries on the web. *Information Processing & Management*, **36**(2), 207–227.

Jonquet, C., Shah, N., and Musen, M. A. (2009). The Open Biomedical Annotator. In *AMIA Summit on Translational Bioinformatics*, pages 56–60, San Francisco, CA, United States.

Lister, A. L., Pocock, M., Taschuk, M., and Wipat, A. (2009). Saint: a lightweight integration environment for model annotation. *Bioinformatics*, **25**(22), 3026–3027.

Lopez, V., Unger, C., Cimiano, P., and Motta, E. (2013). Evaluating question answering over linked data. *Journal of Web Semantics*, **21**, 3–13.

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Marginean, A. (2014). GFMed: Question Answering over BioMedical Linked Data with Grammatical Framework. In *CLEF*.

Neal, M. L., Thompson, C. T., Kim, K. G., James, R. C., Cook, D. L., Carlson, B. E., and Gennari, J. H. (2018). SemGen: a tool for semantics-based annotation and composition of biosimulation models. *Bioinformatics*, **35**(9), 1600–1602.

Neal, M. L., König, M., Nickerson, D., Mısırlı, G., Kalbasi, R., DrÃ¤ger, A., Atalag, K., Chelliah, V., Cooling, M. T., Cook, D. L., Crook, S., de Alba, M., Friedman, S. H., Garny, A., Gennari, J. H., Gleeson, P., Golebiewski, M., Hucka, M., Juty, N., Myers, C., Olivier, B. G., Sauro, H. M., Scharm, M., Snoep, J. L., TourÃ©, V., Wipat, A., Wolkenhauer, O., and Waltemath, D. (2019). Harmonizing semantic annotations for computational models in biology. *Brief Bioinform*, **20**(2), 540–550.

Ogilvie, P., Callan, J., and Callan, J. (2003). Combining Document Representations for Known-item Search. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, SIGIR '03, pages 143–150, New York, NY, USA. ACM. event-place: Toronto, Canada.

Pérez, J., Arenas, M., and Gutierrez, C. (2009). Semantics and Complexity of SPARQL. *ACM Trans. Database Syst.*, **34**(3), 16:1–16:45.

Robertson, S., Zaragoza, H., and Taylor, M. (2004). Simple BM25 Extension to Multiple Weighted Fields. In *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*, CIKM '04, pages 42–49, New York, NY, USA. ACM. event-place: Washington, D.C., USA.

Salton, G. and Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, **24**(5), 513–523.

Salton, G. and Yang, C. S. (1973). On the specification of term values in automatic indexing. *Journal of Documentation*.

Salton, G., Wong, A., and Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Commun. ACM*, **18**(11), 613–620.

Sarwar, D. M., Kalbasi, R., Gennari, J. H., Carlson, B. E., Neal, M. L., Bono, B. d., Atalag, K., Hunter, P. J., and Nickerson, D. P. (2018). Model Annotation and Discovery with the Physiome Model Repository. *bioRxiv*, page 498501.

Sarwar, D. M., Munarko, Y., and Nickerson, D. P. (2019). Epithelial Modelling Platform: A Tool for Model Discovery and Assembly with the Physiome Model Repository. *bioRxiv*, page 631465.

Unger, C., Forascu, C., Lopez, V., Ngomo, A.-C. N., Cabrio, E., Cimiano, P., and Walter, S. (2014). Question Answering over Linked Data (QALD-4).

Whetzel, P. L., Noy, N. F., Shah, N. H., Alexander, P. R., Nyulas, C., Tudorache, T., and Musen, M. A. (2011). BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. *Nucleic Acids Res*, **39**(Web Server issue), W541–W545.

Xu, K., Zhang, S., Feng, Y., and Zhao, D. (2014). Answering Natural Language Questions via Phrasal Semantic Parsing. In C. Zong, J.-Y. Nie, D. Zhao, and Y. Feng, editors, *Natural Language Processing and Chinese Computing*, Communications in Computer and Information Science, pages 333–344. Springer Berlin Heidelberg.

Yahya, M., Berberich, K., Elbassuoni, S., Ramanath, M., Tresp, V., and Weikum, G. (2012). Natural Language Questions for the Web of Data. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 379–390, Stroudsburg, PA, USA. Association for Computational Linguistics. event-place: Jeju Island, Korea.

Yi, K., Beheshti, J., Cole, C., Leide, J. E., and Large, A. (2006). User search behavior of domain-specific information retrieval systems: An analysis of the query logs from PsycINFO and ABC-Clio's Historical Abstracts/America: History and Life. *Journal of the American Society for Information Science and Technology*, **57**(9), 1208–1220.

Yu, T., Lloyd, C. M., Nickerson, D. P., Cooling, M. T., Miller, A. K., Garny, A., Terkildsen, J. R., Lawson, J., Britten, R. D., Hunter, P. J., and Nielsen, P. M. F. (2011). The Physiome Model Repository 2. *Bioinformatics*, **27**(5), 743–744.

Zou, L., Huang, R., Wang, H., Yu, J. X., He, W., and Zhao, D. (2014). Natural Language Question Answering over RDF: A Graph Data Driven Approach. In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, SIGMOD '14, pages 313–324, New York, NY, USA. ACM. event-place: Snowbird, Utah, USA.