Description of the POMELO system for the task 2 of QALD-2014

Thierry Hamon[1,2], Natalia Grabar[3], Fleur Mougin[4], and Frantz Thiessard[4]

[1] LIMSI-CNRS, Orsay, France
thierry.hamon@limsi.fr,
[2] Université Paris 13, Sorbonne Paris Cité, France
[3] STL UMR8163 CNRS, Université Lille 3, France
natalia.grabar@univ-lille3.fr,
[4] Université Bordeaux, ISPED, Centre INSERM U897, ERIAS, France
firstname.lastname@isped.u-bordeaux2.fr

Abstract. In this paper, we present the POMELO system we develop for participating to the task 2 of the QALD challenge. Our approach for translating natural language questions in SPARQL queries is based on natural language processing methods, semantic resources and RDF triple description. We design a four-step method which pre-process the question, generation an abstraction of the question, then build a representation of the SPARQL query and finally generate the query. The system was ranked 2nd out of 3 submissions. It achieves good performance with a F-measure of 0.85 on the set of 25 questions.

Keywords: Natural Language Processing, SPARQL, biomedical domain, semantic resources

1 Introduction

Biomedical knowledge is disseminated in databases which become increasingly available on the Web. These databases usually focus on a given type of information: chemical, pharmacological and target information on drugs in Drugbank [11], clinical studies in ClinicalTrials.gov[5], drugs and their side effects in Sider [8], etc. Connecting such Life science databases is crucial for obtaining more global and comprehensive view on links that may exist between different components, factors and actors of the biomedical knowledge. Moreover, this allows inducing and producing new knowledge from the already available data. Particularly, creation of fine-grained links between the existing databases related to drugs is a great challenge that is being addressed by the project Linked Open Drug Data (LODD) for instance[6]. In this project, the knowledge recorded in the databases and dataset interlinks is represented as RDF triples, on the basis of which the linked data can then be queried through the SPARQL end-point. However, typical users of this knowledge, such as physicians, life science researchers

[5] http://clinicaltrials.gov/
[6] http://www.w3.org/wiki/HCLSIG/LODD

or even patients, cannot manage the syntactic and semantic requirements of the SPARQL language neither can they manage the structure of various databases. This situation impedes the efficient use of databases and the retrieval of useful information. Therefore it is important to design friendly interfaces that mediate the technical ans semantic complexity of the task and provide simple approaches for querying the databases.

For instance, an existing work shows that, for querying the databases and the Semantic Web data, the use of full and standard sentences is preferred to the use of keywords, menus or graphs [6]. While this study is conducted on general knowledge data, we assume this observation is also relevant for the users of biomedical databases. Such question is addressed in general database to provide friendly user interface [7, 3]. We can also mention another work which aims at translating medical questions issued from a journal into SPARQL queries [1].

In relation with such research problems, the Question Answering over Linked Data (QALD-4) campaign proposes the task dedicated to the retrieval of precise biomedical information in linked databases with questions in natural language. We present in this paper, the methodology we propose to translate natural language questions in SPARQL queries and the system we developed for our participation to the challenge.

We start with the definitions of the main terms used in the proposed presentation (section 2). Then, we describe the semantic resources available and developed to enrich the questions in section 3. The methodology and the system are described in section 4. The evaluation of the system on the QALD-4 queries is presented in section 5.

2 Terminology

The main terms are used with the following meaning:

Question The questions are the natural language expressions uttered by human users in order to formulate their information need.

Query The queries respect the SPARQL syntax and semantics. They are created automatically on the basis of natural language questions.

The main challenge of the proposed work is to design the optimal methodology for an easy and reproducible rewriting of natural language questions in SPARQL queries.

3 Definition of the semantic resources

Some of the resources used are provided by the challenge organizers (section 3.1), others are collected and built speciafically for the challenge (section 3.2) in order to support the method. These resources are used in a way that allows rewriting the questions in queries.

3.1 Resources provided by the QALD challenge

Three datasets are provided by the QALD challenge:

- Drugbank[7] is dedicated to drugs [11]. It merges chemical, pharmacologic and pharmaceutical information from other available databases. We exploited the documentation[8] of this resource to define rewriting rules[9] and regular expression in our named entity recognizer.
- Diseasome[10] is dedicated to disease and genes linked among them by known disorder/gene associations [5]. It provides single framework with all known phenotype and disease gene associations, indicating the common genetic origin of many diseases. We exploit the RDF triples and the documentation of the resource to define the rewriting rules[11].
- Sider[12] is dedicated to adverse drug effects [8]. It contains information on marketed medicines and their recorded adverse drug reactions. The information is extracted from public documents and package inserts. The available information include side effect frequency, drug and side effect classifications as well as links to further information, for example drug-target relations.[13]

The content of each resource is provided in specific format: RDF triples *subject predicate object*, so that they encode the useful and usable core frame elements of the frames (or predicates).

3.2 Resources collected and built for the QALD challenge

On the basis of the RDF triples, we build frames from the RDF schema where the RDF predicate is the frame predicate, and the subject and the object of the RDF triples are the core frame elements. This also includes the OWL sameAs triples. Several types of entities are isolated:

- As indicated, subject, object and predicate become semantic entities. They may occur in questions: in this way, the frames are the main resource for the rewriting of questions in queries;
- Vocabulary specific to the questions is also built. It covers for instance aggregation operators, negation, types of requests, etc.;
- RDF literals, issued from named entity recogniser or term extractor, complete the resources. The RDF literals are detected with specifically designed automata that may rely on the source database documentation.

[7] http://www.drugbank.ca
[8] http://www.drugbank.ca/documentation
[9] (example: approved)
[10] http://diseasome.eu
[11] (example: Connective_tissue_disorder)
[12] http://sideeffects.embl.de
[13] exemple?

These entities are associated with the expected semantic type, which allows creating the queries and rewriting the RDF triples in the SPARQL queries. In that respect, we can consider IRI as well as strings, common datatype or regular expressions when literal are expected.

Most of the entities are considered and processed through their semantic type, although some ambiguous entities are considered atomically. For these, the rewriting rules will be applied contextually to generate the semantic entities corresponding to the frames (see section 4.2). When using the queries, the semantic types are variables and are used for connecting the edges of queries.

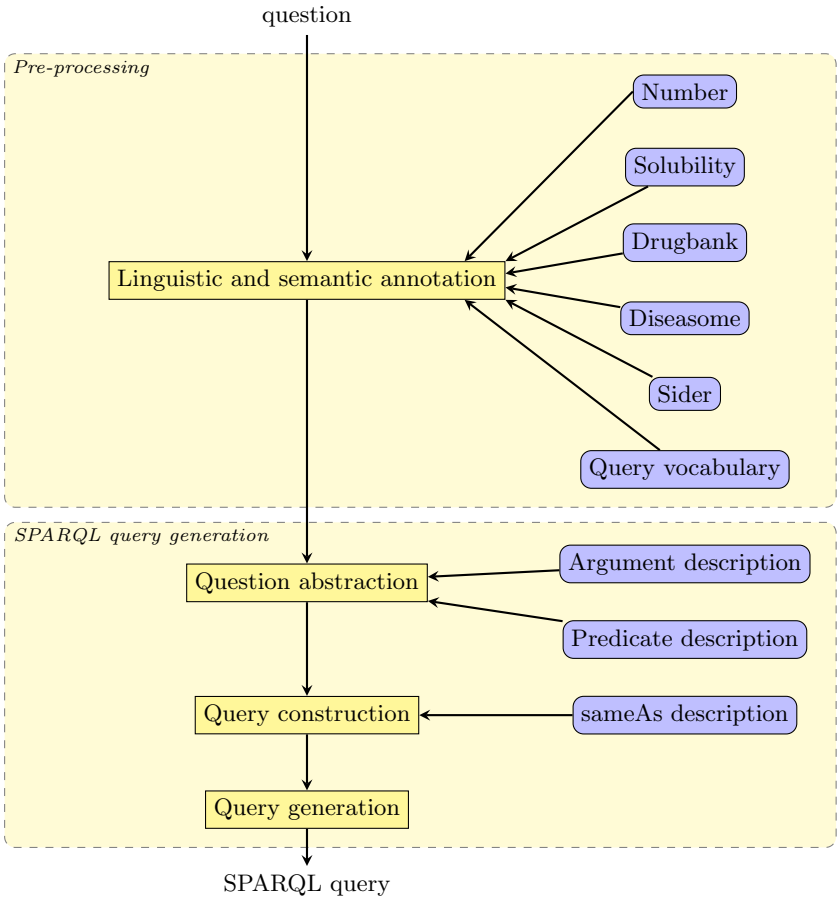


Fig. 1. Architecture of the POMELO system (processing steps are in yellow, and resources in blue)

4 System description

We design a four-step process based on natural language processing (NLP) methods, semantic resources and RDF triple description (see Figure 1):

1. we pre-processing the queries in order to enrich them with linguistic and semantic information (section 4.1);
2. we perform a question abstraction (section 4.2);
3. we use the abstracted question to construct the corresponding SPARQL query representation (section 4.3);
4. we generate the SPARQL query (section 4.4).

The process is implemented as module within the NLP platform Ogmios [4].

4.1 Pre-processing

The annotation of the questions consists in recognition of numerical values (such as numbers and solubility values), word segmentation, part-of-speech tagging and lemmatization of the words with TreeTagger [9]. Then, we apply the TermTagger Perl module[14] for identification of semantic entities, i.e. terms with associated semantic categories. TermTagger exploits the semantic resources (see section 3) to recognize semantic entities such as disease names, side effects, etc.

However, as we realized during our preliminary experiments on the training set, the coverage of the terminological entities that appear in questions is not sufficient. We apply the term extractor YaTeA[15] [2] to improve the coverage. The term extractor performs shallow parsing of the POS-tagged and lemmatized text by chunking it according to syntactic frontiers (pronouns, conjugated verbs, typographic marks, etc.) to identify noun phrases. Then, parsing patterns taking into account the morpho-syntactic variation, are recursively applied and provide parsed terminological entities, usually noun phrases relevant for the targeted domain. Each term is represented as a syntactic tree, and sub-terms are also considered as terms in the current configuration. No semantic categories are associated to the terms extracted by YaTeA.

Moreover, we have observed on the training step that Genia Tagger [10] performs a better lemmatization. However, we didn't use it on the test set because it is time consuming.

4.2 Question abstraction

This step aims at identifying the relevant elements within the questions and building a representation of these elements. At this step, we use the linguistic and semantic annotations associated to the question words in the previous step.

Before the identification of relevant elements, a post-processing is carried out for disambiguating the annotations if necessary. Indeed, the annotated semantic

[14] http://search.cpan.org/ thhamon/Alvis-TermTagger/
[15] http://search.cpan.org/ thhamon/Lingua-YaTeA/

entities may receive conflicting or redundant semantic types, while the post-processing permits to select those entities and semantic types that may useful for the next steps. For instance, we keep larger terms which do not include semantic entities. Besides, the rewriting rules are defined so that they can modify or delete the semantic type associated to an entity according to the context. Other rules may also modify or delete the entity in context. As an example, the semantic entity *interaction* has to be rewritten in `interactionDrug1` if a drug occurs in its context, but it has to be rewritten in `foodInteraction` if a term with the semantic category *food* occurs in its context. On the whole, we define 44 contextual rewriting rules based on the vocabulary used in the questions and the documentation of databases, mainly the one from Drugbank[16].

To abstract the question in a data representation, we identify information basically related to the query structure:

1. Result form definition: the question is scanned for identifying words expressing the negation, e.g. *no*, and its scope, the aggregation operation on the results, e.g. *number* for `count`, *mean* for `avg` or *higher* for `max`, and specific result form such as boolean query (`ASK`). Information concerning the presence of the negation and aggregation operators or of specific result form is recorded in data structures to be used at the end of the *query construction* step or during the *query generation* step.
2. Question topic identification: we consider the first semantic entity with a given expected semantic type to be the question topic. The expected semantic types are those provided by the RDF subjects and objects in the Drugbank, Diseasome and Sider. This information will be used during the *query construction* step.
3. Predicate and Argument identification: according to our internal frame-based representation of the three resources, the potential predicates, subjects and objects are identified among the semantic entities and described into a symbol table. At this step, the subjects and objects are fully described in the symbol table. Concerning the predicates, only the semantic types of their arguments are instantiated in the symbol table with the the RDF schema. The subjects and objects can be URI, RDF typed literals (numerical values or strings) and extracted terms (these are considered as elements of regular expressions).

4.3 Query construction

The objective of the *query construction* step is to connect previously identified elements and to build the SPARQL graph pattern (introduced by the keyword `WHERE`). Thus, the symbols of the predicate arguments are instantiated by either URI associated to the subjects and objects, variables, numerical values or strings.

For each question, we perform several connections:

[16] http://www.drugbank.ca/documentation

1. the question topic to the predicate(s). A variable is associated to the question topic and the predicate arguments that matched the semantic type of the question topic. Note that at the end of this stage, the question topic may remain not associated to any predicate;
2. the predicate arguments to the subjects and objects identified during the question abstraction. It concerns elements referring to URI;
3. the predicates between them through their subjects and objects. The connection between two predicates is then represented by a variable;
4. the predicates from different datasets. We use the `sameAs` description to identify URI referring to the same element. New variables are defined to connect two predicates;
5. the remaining question topic to arguments of the `sameAs` predicate;
6. the arguments with the `string` type to extracted terms annotated in the question. We assume these arguments will be related to string matching operator `REGEX`. Thus, terms are considered as string expressions.

At this point, the predicate arguments which remain unassociated are replaced by new variables in order to avoid empty literals. Finally, the negation operator is taken into account: the predicates are marked as negated and the arguments within the negation scope are included in a new predicate `rdf:type` if required.

At this stage, the question is fully translated in data structures representing the SPARQL query.

4.4 Query generation

This final step aims at generating the SPARQL query string based on the data structures built during the *query construction* step. The output of this step is the string corresponding to the SPARQL query. It is composed in two parts:

1. the generation of the results form which take into account the expected type of result form (`ASK` or `SELECT`), the presence of aggregation operators and the variable associated to the question topic;
2. the generation of the graph pattern. Basically, the part of the query generation consists in generating the strings representing each RDF triples and filters if predicates are negated. But when aggregation operators are used, we also need to recursively generate sub-queries computing the subsets of expressions, before their aggregation.

The SPARQL queries have been submitted without retrieving the answers. We let this task to the evaluation tool.

5 Results

5.1 Evaluation metrics

The automatically generated SPARQL queries are evaluated with the online evaluation tool[17]. The answer of each query q is compared with the gold stan-

[17] http://greententacle.techfak.uni-bielefeld.de/ cunger/qald/index.php?x=evaltool&q=4

dard. The evaluation measures (F-measure, precision and recall) are computed and provided. The system results are evaluated with macro-measures.

The challenge provides 25 questions for training and 25 questions for the tests (see Tables 2 and 3).

5.2 Global results

In table 1, we presents the overall results on the training set and the test set. Our system was ranked 2nd out of 3 submissions. We can observe that our results are similar on these two sets of queries.

query set	Recall	Precision	F-measure
Training	0.87	0.83	0.85
Test	0.87	0.82	0.85

Table 1. Results on the training set and the test set

The results per query are presented in Table 2 for the training set and Table 3 for the test set. We can observe that in both cases, for 19 questions, the system exactly provides the expected answers. For 4 questions from the training set and 3 questions from the test set, we obtain partial answers. In the training set, 2 questions receive no answer while in the test set, we have 3 such questions. To our opinion, the reference SPARQL query for the question 19 of the training set is not correct: the expected result of the question is a list of drugs, while the reference SPARQL query returns a list of diseases. On the test set, we can propose two observations on the limitations of our system:

- in question 1, contextual rewriting rules can not be correctly applied because the semantic entity (`gene ... associated`) is discontinued;
- in question 18, the system correctly detects the semantic entities and the predicates, including the `sameAs` predicate. The problem is that the system assumes the `sameAs` predicate is reflexive while in the resources, the instances of this predicate don't encode the reflexivity of the relation.

5.3 System performance

We analyze the system performance on a standard computer. The figure 2 presents the running time for each query according to the pre-processing sub-steps (named entity recognition, word and sentence segmentation, POS tagging, semantic entity tagging and term extraction) and the question translation in SPARQL query (Question2SPARQLQuery). We can observe that each question is processed in 1.94 seconds on the average on the training set and 1.97 seconds on the average on the test set. The most of the processing time is dedicated to the TermTagger which aims at recognizing semantic entities. The figure 3 shows

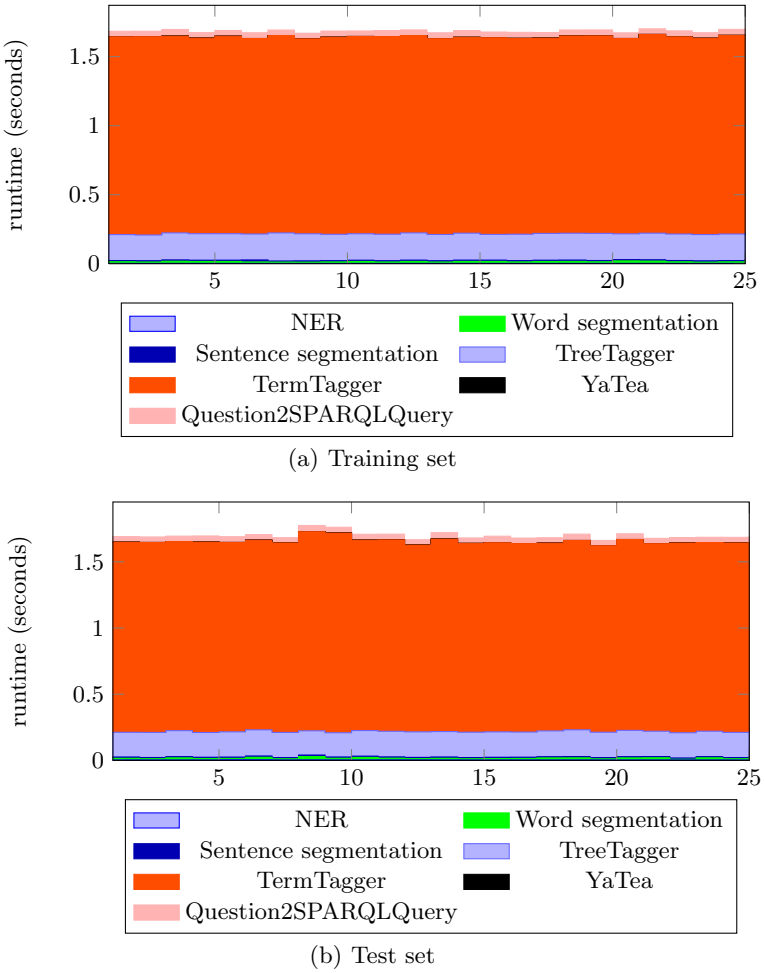
id	question string	Recall	Precision	F-measure
1	Which diseases is Cetuximab used for?	1.0	1.0	1.0
2	What are the diseases caused by Valdecoxib?	0	0	0
3	What is the side effects of drugs used for Tuberculosis?	1.0	1.0	1.0
4	What are the side effects of Valdecoxib?	1.0	1.0	1.0
5	Which genes are associated with breast cancer?	0.5	0.05	0.10
6	Which drugs have fever as a side effect?	1.0	1.0	1.0
7	Give me diseases treated by tetracycline	1.0	1.0	1.0
8	Which drugs interact with allopurinol?	1.0	1.0	1.0
9	What are side effects of drugs used for asthma?	1.0	0.59	0.74
10	Which foods does allopurinol interact with?	1.0	1.0	1.0
11	What are enzymes of drugs used for anemia?	0.25	1.0	0.40
12	What is the target drug of Vidarabine?	1.0	1.0	1.0
13	Which drugs target Multidrug resistance protein 1?	1.0	1.0	1.0
14	Give me drug references of drugs targeting Prothrombin.	1.0	0.032	0.062
15	Which genes are associated with diseases treated with Cetuximab?	1.0	1.0	1.0
16	Which drugs have hypertension and vomiting as side-effects?	1.0	1.0	1.0
17	Which are possible drugs against rickets?	1.0	1.0	1.0
18	What are the common side effects of Doxil and Bextra?	1.0	1.0	1.0
19	Which are the drugs whose side effects are associated with the gene TRPM6?	0	0	0
20	Which are the side effects of Penicillin G?	1.0	1.0	1.0
21	Which diseases are associated with the gene FOXP2?	1.0	1.0	1.0
22	Which are possible drugs for diseases associated with the gene ALD?	1.0	1.0	1.0
23	Which are targets of Hydroxocobalamin?	1.0	1.0	1.0
24	Which are targets for possible drugs for diseases associated with the gene ALD?	1.0	1.0	1.0
25	Which genes are associated with diseases whose possible drugs target Cubilin?	1.0	1.0	1.0
	Overall results	0.87	0.83	0.85

Table 2. Results on the training set

id	question string	Recall	Precision	F-measure
1	Which genes are associated with Endothelin receptor type B?	0	0	0
2	Which genes are associated with subtypes of rickets?	1.0	1.0	1.0
3	Which drug has the highest number of side-effects?	1.0	1.0	1.0
4	List drugs that lead to strokes and arthrosis.	1.0	1.0	1.0
5	Which drugs have a water solubility of 2.78e-01 mg/mL?	1.0	1.0	1.0
6	Give me the side-effects drugs with a solubility of 3.24e-02 mg/mL.	1.0	0.59	0.74
7	Which diseases are associated with SAR1B?	1.0	1.0	1.0
8	Which experimental drugs interact with food?	1.0	1.0	1.0
9	Which approved drugs interact with fibers?	1.0	1.0	1.0
10	Which drugs interact with food and have HIV infections as side-effects?	0	0	0
11	Give me diseases whose possible drugs target the elongation factor 2.	1.0	1.0	1.0
12	Which drugs achieve a protein binding of 100%?	1.0	1.0	1.0
13	List illnesses that are treated by drugs whose mechanism of action involves norepinephrine and serotonin.	1.0	1.0	1.0
14	Which is the least common chromosome location?	1.0	1.0	1.0
15	Are there drugs that target the Protein kinase C beta type?	1.0	1.0	1.0
16	Give me all diseases of the connective tissue class.	1.0	1.0	1.0
17	Which targets are involved in blood clotting?	1.0	1.0	1.0
18	List the number of distinct side-effects of drugs which target genes whose general function involves cell division.	0	0	0
19	Which drugs have no side-effects?	1.0	1.0	1.0
20	List diseases whose possible drugs have no side effects.	1.0	0.02	0.032
21	Give me the drug categories of Desoxyn.	0.86	1.0	0.92
22	Give me drugs in the gaseous state.	1.0	1.0	1.0
23	Which disease has the largest size?	1.0	1.0	1.0
24	Which drugs have bipolar disorder as indication?	1.0	1.0	1.0
25	Which diseases have a class degree of 11?	1.0	1.0	1.0
	Overall results	0.87	0.82	0.85

Table 3. Results on the test set

the overall system performance according the number of questions to process. The time needed for processing all the questions is higher on the test set. The variation of running time between processing one question and the whole set of questions is less than one second on the training set and a little more than one second on the test set. We assume the difference is due to the higher complexity of the questions in the test set.



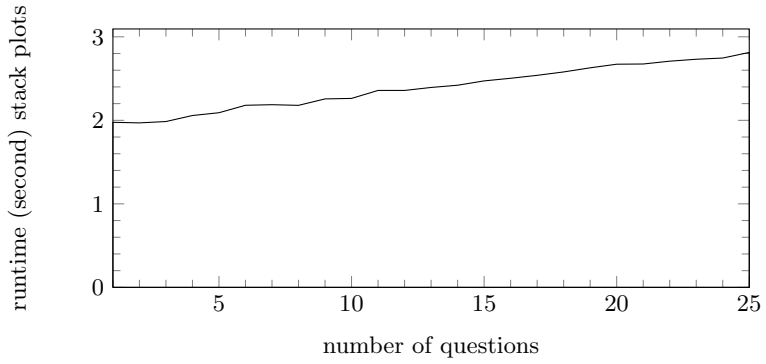
(a) Training set

(b) Test set

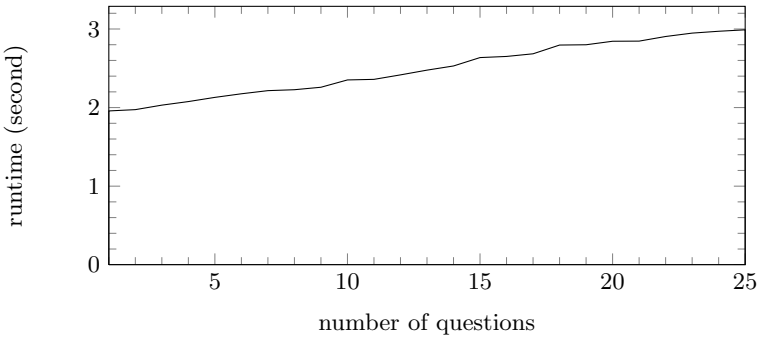
Fig. 2. System performance for each question

6 Conclusion

We proposed a four-step process based on natural language processing methods, semantic resources and RDF triple description. The system achieves good performance with a F-measure of 0.85 on the set of 25 questions. It is ranked 2nd out of 3 submissions. Further work includes optimization of the running time for the question processing and the extension of the SPARQL syntax by taking into account the operators on collection of sets. We also investigate the integration of other biomedical resources as Dailymed or RxNorm, and the use of such system in text mining applications.



(a) Training set



(b) Test set

Fig. 3. System performance for an increasing number of questions

References

1. Abacha, A.B., Zweigenbaum, P.: Medical question answering: Translating medical questions into sparql queries. In: ACM SIGHIT International Health Informatics

Symposium (IHI 2012) (2012)
2. Aubin, S., Hamon, T.: Improving term extraction with terminological resources. In: Salakoski, T., Ginter, F., Pyysalo, S., Pahikkala, T. (eds.) Advances in Natural Language Processing (5th International Conference on NLP, FinTAL 2006). pp. 380–387. No. 4139 in LNAI, Springer (August 2006)
3. Damljanovic, D., Agatonovic, M., Cunningham, H.: Natural language interfaces to ontologies: Combining syntactic analysis and ontology-based lookup through the user interaction. In: Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part I. pp. 106–120. ESWC'10, Springer-Verlag, Berlin, Heidelberg (2010)
4. Hamon, T., Nazarenko, A., Poibeau, T., Aubin, S., Derivire, J.: A robust linguistic platform for efficient and domain specific web content analysis. In: Proceedings of RIAO 2007. Pittsburgh, USA (2007), http://riao.free.fr/papers/64.pdf, 15 pages
5. Janji, V., Prulj, N.: The core diseasome. Mol Biosyst 8(10), 2614–2625 (Aug 2012), http://dx.doi.org/10.1039/c2mb25230a
6. Kaufmann, E., Bernstein, A.: How useful are natural language interfaces to the semantic web for casual end-users? In: Proceedings of the Forth European Semantic Web Conference (ESWC 2007). Innsbruck, Austria (June 2007)
7. Kuchmann-Beauger, N., Aufaure, M.A.: Natural language interfaces for datawarehouses. In: 8mes journes francophones sur les Entrepts de Donnes et l'Analyse en ligne (EDA 2012), Bordeaux. RNTI, vol. B-8, pp. 83–92. Hermann, Paris (Juin 2012)
8. Kuhn, M., Campillos, M., Letunic, I., Jensen, L.J., Bork, P.: A side effect resource to capture phenotypic effects of drugs. Molecular Systems Biology 6(1) (2010), http://msb.embopress.org/content/6/1/343
9. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Jones, D., Somers, H. (eds.) New Methods in Language Processing Studies in Computational Linguistics (1997)
10. Tsuruoka, Y., Tateishi, Y., Kim, J.D., Ohta, T., McNaught, J., Ananiadou, S., Tsujii, J.: Developing a robust part-of-speech tagger for biomedical text. In: Proceedings of Advances in Informatics - 10th Panhellenic Conference on Informatics. pp. 382–392. LNCS 3746 (2005)
11. Wishart, D.S., Knox, C., Guo, A.C., Shrivastava, S., Hassanali, M., Stothard, P., Chang, Z., Woolsey, J.: Drugbank: a comprehensive resource for in silico drug discovery and exploration. Nucleic Acids Research 34, D668D672 (2006), database issue