Semantic Web 0 (0) 1
IOS Press

Querying Biomedical Linked Data with Natural Language Questions

Thierry Hamon [a,b,*], Natalia Grabar [c], Fleur Mougin [d]
[a] *LIMSI-CNRS, Orsay, France*
E-mail: hamon@limsi.fr
[b] *Université Paris 13, Sorbonne Paris Cité, France*
[c] *STL UMR8163 CNRS, Université Lille 3, France*
E-mail: natalia.grabar@univ-lille3.fr
[d] *Université Bordeaux, ERIAS, Centre INSERM U897, ISPED, France*
E-mail: {firstname.lastname}@u-bordeaux.fr

Abstract. A recent and intensive research in the biomedical area enabled to accumulate and disseminate biomedical knowledge through various knowledge bases increasingly available on the Web. The exploitation of this knowledge requires to create links between these bases and to use them jointly. Linked Data, SPARQL language and interfaces in Natural Language question-answering provide interesting solutions for querying such knowledge bases. However, while using biomedical Linked Data is crucial, life-science researchers may have difficulties using SPARQL language. Interfaces based on Natural Language question-answering are recognized to be suitable for querying knowledge bases. In this paper, we propose a method for translating natural language questions into SPARQL queries. We use Natural Language Processing tools, semantic resources and the RDF triples description. We designed a four-step method which linguistically and semantically annotates the question, performs an abstraction of the question, then builds a representation of the SPARQL query and finally generates the query. The method is designed on 50 questions over 3 biomedical knowledge bases used in the task 2 of the QALD-4 challenge framework and evaluated on 27 new questions. It achieves good performance with 0.78 F-measure on the test set. The method for translating questions into SPARQL queries is implemented as a Perl module and is available at `http://search.cpan.org/~thhamon/RDF-NLP-SPARQLQuery/`.

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

1. Introduction

A recent and intensive research in the biomedical area enabled to accumulate and disseminate biomedical knowledge through various knowledge bases (KBs) increasingly available on the Web. Such life-science bases usually focus on a specific type of biomedical information: clinical studies in ClinicalTrials.gov[1]; drugs and their side effects in Sider [14]; chemical, pharmacological and target information on drugs in Drugbank [28], etc.

Nowadays, creating connections between these knowledge bases is crucial for obtaining a more global and comprehensive view on the links between different biomedical components. Such links are also required for inducing and producing new knowledge from the already available data. There is a great endeavour in the definition of Open Linked Data to connect such knowledge and in taking advantage of the SPARQL language to query multiple knowledge bases jointly. Particularly, the creation of fine-grained links between

*Corresponding author. E-mail: hamon@limsi.fr.
[1] `http://clinicaltrials.gov/`

the existing KBs related to drugs is a great challenge that is being addressed, for instance, by the project Linked Open Drug Data (LODD)[2]. The knowledge recorded in the knowledge bases and dataset interlinks are represented as RDF triples, on the basis of which the linked data can then be queried through a SPARQL end-point. However, typical users of this knowledge, such as physicians, life-science researchers or even patients, cannot manage the syntactic and semantic requirements of the SPARQL language neither can they manage the structure of various knowledge bases. This situation impedes an efficient use of knowledge bases and retrieval of useful information. Therefore, it is important to design friendly interfaces that mediate a technical and semantic complexity of the task and provide simple approaches for querying the knowledge bases. The main challenge is then to design an optimal methodology for an easy and reproducible rewriting of natural language questions into SPARQL queries. In the following, the term *question* means the natural language expressions uttered by human users to formulate their information need, and the term *query* designates the same expression formalised in the SPARQL syntax and semantics.

We present in this paper a novel method to translate natural language questions into SPARQL queries over biomedical Linked Data. The method is based on the use of Natural Language Processing (NLP) tools and resources, the both used to enrich question with linguistic and semantic information. Questions are then translated into SPARQL with a rule-based approach. We design our approach on the 50 questions proposed by the task 2, *Biomedical question answering over interlinked data*,[3] of the QALD-4 challenge, and evaluate it on 27 newly defined questions[4]. A sample of the new test set is given at Sect. 5.1. We work with three knowledge bases: Drugbank, Diseasome, and Sider described in Sect. 4.

The paper is structured as follows. Section 2 presents the related work. We describe the proposed method in Sect. 3 and then the semantic resources available and developed for enriching the questions in Sect. 4. The evaluation of the method is presented in Sect. 5.

[2] `http://www.w3.org/wiki/HCLSIG/LODD`

[3] `http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=task2\&q=4`

[4] The new set with 27 questions is available at the following URL: `http://perso.limsi.fr/hamon/Files/QALD/qald-4\_biomedical\_additional\_test.xml`

2. Related Work

Querying Linked Data requires the definition of the end user interfaces which hide the underlying structure of the knowledge base as well as the SPARQL syntax. Three ways are identified for querying Linked Data: Knowledge-Based Specific Interface, Graphical Query Builder and Question-Answering System [10]. It has also been recognized that Natural Language interfaces are the most suitable: the authors have demonstrated that for querying the knowledge bases and the Semantic Web data, the use of full and standard sentences is preferred to the use of keywords, menus or graphs [10].

Another possible distinction is related to the types of linked data processed and the purpose of this processing. Concerning the types of the linked data processed, these can be provided from the general language [5][13][27] or specialized languages [1] knowledge bases. In relation to the purpose of this processing, two kinds of work can be distinguished: (1) transformation of natural language questions in SPARQL queries in order to get the best coverage of the queries generated (Sect. 2.1); (2) transformation of natural language questions in SPARQL queries and questionning knowledge bases in order to get the best results when querying the linked data (Sect. 2.2). Our work is related to this latter purpose. In what follows, we also present other related research questions addressed in the existing work (Sect. 2.3).

2.1. Transformation of Questions in Queries

The main objective is to propose methods for a more efficient transformation of natural language questions in SPARQL queries. Most of the existing approaches rely on patterns or templates.

One system, called Question-Answering system (AutoSPARQL), is based on active supervised machine learning independent from the knowledge base [16]. The SPARQL query model is learnt from Natural Language questions. The authors report

that 50 questions are successfully processed with the system.

A method based on modular patterns is designed to parse questions [21]. The main purpose is to model questions for the SPARQL query generation. In this method, semantic relations are identified for the first keywords detected. The method is tested on 160 movie-related questions. An advantage of this method is that it requires only four query patterns, while in a previous work of the authors twelve patterns were necessary.

Another existing system relies on resources automatically derived from ontologies or knowledge bases for transforming questions in SeRQL queries [26]. Questions undergo a set of treatments (e.g. linguistic analysis, string similarity). From the set of 22 questions, 15 (68%) of them can be interpreted and transformed correctly.

An existing multilingual toolkit for 36 languages [22] is also used for the transformation of questions in queries [6]. Correspondences between linguistic units and SPARQL elements are established: common noun (*Kind*), noun phrase (*Entity*), verb phrase (*Property*) and verb phrase with a higher arity (*Relation*). An evaluation is performed on seven languages. The results indicate that 112 basic query patterns are used and can be combined with several logical operators.

Another system is based on a manually-written grammar and ontological knowledge [7]. It allows processing 145 questions from the set of 164 (88% coverage).

2.2. Query Generation and Questionning over Linked Data

The main objective of the related set of work is more complex: first, the system has to transform the natural language questions in SPARQL queries; and second, it has to question the knowledge bases in order to get the best results when querying the linked data. The main advantage of this kind of works is that they cover the entire querying process. Moreover, they allow to evaluate the final results (answers extracted from the knowledge bases) and to provide precise evaluation figures. Often, NLP tools and methods are used for the transformation of questions in queries.

A template-based system relying on NLP tools and semantic resources is proposed for processing natural language questions [27]. The application of the system on 50 questions from DBpedia proposed by the QALD-2 challenge gives competitive results with 0.62 average F-measure obtained with 39 questions (average recall is 0.63 and average precision is 0.61), but shows a low coverage as 11 questions are not covered by the templates.

Translation of medical questions issued from a journal into SPARQL queries relies on a hybrid approach [1]: an SVM machine learning-based approach used to extract the characteristics of the questions (named entities, relations) is combined with patterns to generate the SPARQL queries. The evaluation is carried out on 100 questions and the corresponding queries are tested on clinical documents. The method achieves a 0.62 precision when querying the documents.

An approach based on knowledge-driven disambiguation of questions and on coloured activation of query graph [18] has been tested on 100 questions from the QALD-2013 dataset. According to settings tested, the system F-measure varies from 0.4 (QALD 2013 dataset), to 0.6 with the entity search, and up to 0.8 with a boolean setting. Among the difficulties observed, the authors notice errors due to the relation interpretation, missing lexical knowledge, parsing of complex questions and remaining difficulties with the ambiguities.

Recently, the Question Answering over Linked Data (QALD-4) challenge proposes a task[5] dedicated to the retrieval of precise biomedical information in linked knowledge bases with questions expressed in natural language.

2.3. Other Related Research Questions

Several research questions can be related to the querying of linked data through the natural language interfaces. Usually this kind of work aims at improving specific points: identification of different types of SPARQL queries (select, construct, ask, describe) [17], detection of named entities [12], generation of SPARQL templates [11], classification of the semantic correspondence between question units and query elements [8], design of a SPARQL solver based on contraint programming to query RDF documents [15], processing of

[5]Biomedical question answering over interlinked data, `http://greententacle.techfak.uni-bielefeld.de/~cunger/qald/index.php?x=task2\&q=4`

complex queries and their decomposition in subqueries [20].

3. Question translation into SPARQL Query

To translate natural language questions into SPARQL queries, we design a four-step rule-based method, that relies on NLP methods, semantic resources and the RDF triple description (see Fig. 1). Natural language questions are enriched with linguistic and semantic information (Sect. 3.1). This information is used for abstracting the questions and for identifying relevant information (Sect. 3.2), and then for building the corresponding SPARQL query representations (Sect. 3.3) and for generating the SPARQL queries (Sect. 3.4). The same process is used for translating questions provided by training and test sets. The method for translating questions into SPARQL queries is implemented as a Perl module and is available at `http://search.cpan.org/~thhamon/RDF-NLP-SPARQLQuery/`. Our approach is close to the one proposed by [27]: we use NLP tools in order to linguistically enrich the questions. The main difference is that we use information issued from the Linked Data resources to semantically annotate the questions and to define the frames (i.e., linguistic representations of the RDF schema) in order to model and to build the SPARQL queries. We exemplify our approach on the question *What is the side effects of drugs used for Tuberculosis?*

3.1. Linguistic and Semantic Annotation of Questions

The annotation step aims at associating a linguistic and semantic description to the words and terms of the questions (see Fig. 2).

First, numerical values (such as numbers and solubility values) are identified with a named entity recognizer. Then, the parsing of the questions consists of: word segmentation, part-of-speech tagging and lemmatization of words with TreeTagger [24]. Semantic entities, i.e. terms with associated semantic types representing their meaning, are identified with the TermTagger Perl module[6]. This term recognition relies on semantic resources (see Sect. 4) to recognize semantic entities such as disease names, side effects, etc. However, because the semantic resources often suffer from low coverage [3,19], we also use the term extractor YaTeA[7] [2] to improve the coverage of our method. YaTeA performs shallow parsing of the POS-tagged and lemmatized text by chunking it according to syntactic frontiers (pronouns, conjugated verbs, typographic marks, etc.) in order to identify noun phrases. Then, parsing patterns are recursively applied and provide parsed terminological entities, usually noun phrases relevant for the targeted domain. These parsing patterns have been manually defined during previous work [2]. They take into account the morpho-syntactic variation and reflect basic syntactic dependencies within terminological entities. Each term is represented as a syntactic tree, and sub-terms are also considered as terms in the current configuration. No semantic types are associated with the terms extracted by YaTeA. Negated terms are identified with NegEx algorithm[8] [4]: words expressing the negation e.g. *no*, are identified through regular expressions and then their scope is calculated to detect terms that are negated in the questions. Figure 3 illustrates the linguistic and semantic annotation of questions.

3.2. Question Abstraction

The question abstraction step aims at identifying the relevant elements within questions and at building the representation of these elements (see Fig. 4). It relies on a linguistic and semantic annotation associated with the question words detected in the previous step.

Before the identification of relevant elements, the annotations are post-processed in order to disambiguate the generated semantic annotations. Indeed, the annotated semantic entities may receive conflicting or concurrent semantic types, while the post-processing enables to select those entities and semantic types that may be useful for the next steps. As part of this post-processing, we keep larger terms which do not include other semantic entities. Also, we manually defined rewriting rules on the training set in order to adjust (modify or delete) the semantic types associated

[6] `http://search.cpan.org/~thhamon/Alvis-TermTagger/`

[7] `http://search.cpan.org/~thhamon/Lingua-YaTeA/`

[8] `http://search.cpan.org/~osler/Lingua-NegEx/`

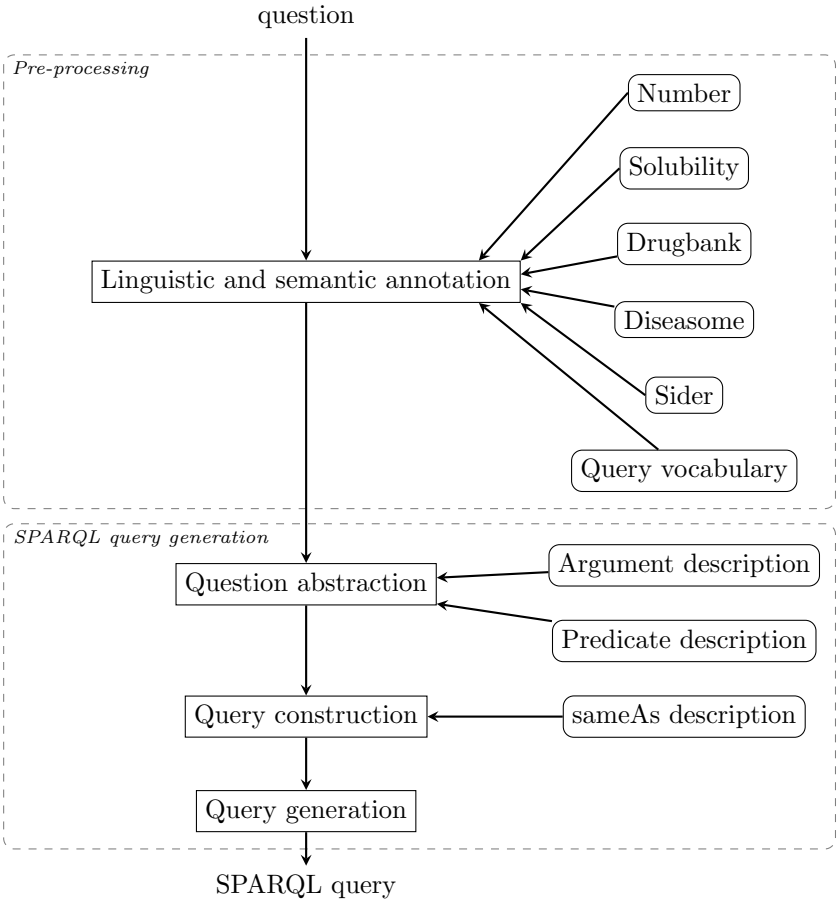

Fig. 1. The global architecture of the system (the processing steps are in yellow, the resources in blue). Square boxes represent the processing steps which are further detailed in figures 2 to 6. Rounded boxes describe the resource used for the processing of the questions and queries.

with a given entity according to the context. Other rules may also modify or delete the entity itself according to its context. For instance, the semantic entity *interaction* has to be rewritten into `interactionDrug1` if its context contains mention of drugs, but it has to be rewritten into `foodInteraction` if its context contains a term with the semantic type *food*. On the whole, we defined 44 contextual rewriting rules based on the vocabulary used in questions from the training set and on the documentation from the exploited knowledge bases, mainly the one from Drugbank[9]. In addition to the rewriting rules, an additional disambiguation of the annotations is also performed during a *SPARQL query construction* step when the arguments of the predicate or the question topic are connected because they share the same semantics.

For performing the question abstraction, we identify information related to the query structure:

1. Definition of the Result form: the question is scanned for identifying the negated terms but also for identifying the aggregation operation on the results, e.g. *number* for `count`, *mean* for `avg` or *higher* for `max`, and specific result forms such as boolean queries (`ASK`). Negated terms and information related to conjunction marks, aggregation operators, and requirement on specific result forms are recorded and will be used at the end of the *query construction* step or during the *query generation*

[9] `http://www.drugbank.ca/documentation`

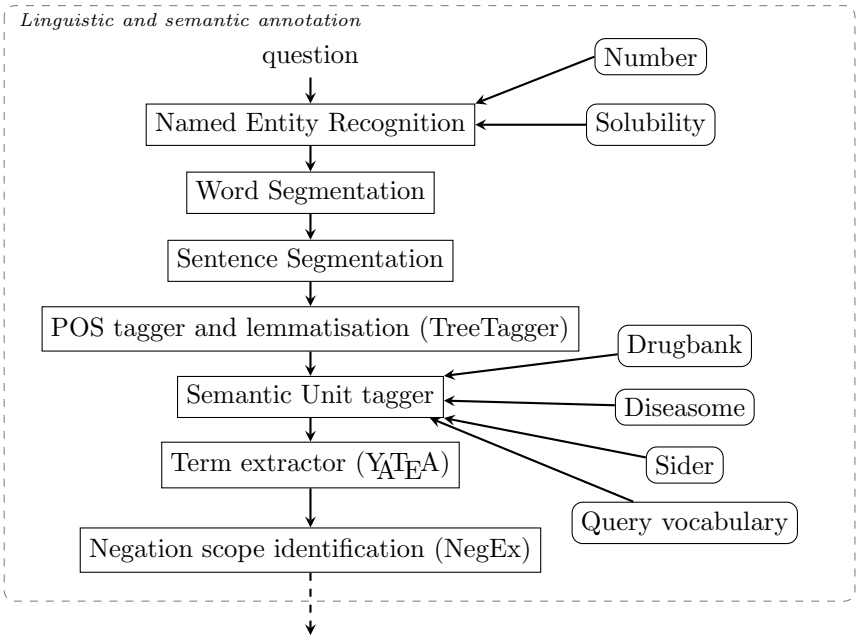

Fig. 2. Linguistic and semantic annotation process. Square boxes represent the steps of the linguistic and semantic analysis of questions. Rounded boxes indicate the resources used for the semantic annotation.

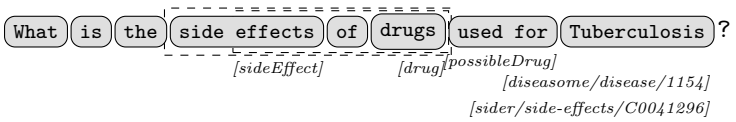

Fig. 3. Example of question pre-processing. The gray rounded boxes represent the word and semantic entities. The subscript texts are the Part-of-Speech tags and the bracketed subscript texts are the semantic types associated with the semantic entities.

step. In the example from Figure 3, no such information is found.

2. Identification of the Question topic: the first semantic entity with a given expected semantic type is considered to be the question topic. The expected semantic types are those provided by the RDF subjects and objects issued from the resources. This information will be used during the *query construction* step. The question topic of our example is identified as **drug**.
3. Identification of Predicate and Argument: we use linguistic representations of the RDF schema i.e. frames which contain one predicate and at least two elements with associated semantic types. In that respect, the potential predicates, subjects and objects of frames are identified among the semantic entities and recorded in a table: entries are the semantic types of the elements and refer to linguistic, semantic and SPARQL information associated with these elements. Subjects and objects are fully described in the table with the inflected and lemmatized forms of words or terms, the corresponding SPARQL types and indicators on their use as object or subject of a predicate. Concerning the predicates, only the semantic types of their arguments are instantiated. Subjects and objects can be URI, RDF typed literals (numerical values or strings) and extracted terms (these are considered as elements of regular expressions). In the example from Figure 3, the predicate **state** with the expected arguments **drugbank/drugs** and **Gas/String** is recognized.
4. Scope of conjunctions: arguments and the predicate in the neighbourhood of conjunc-

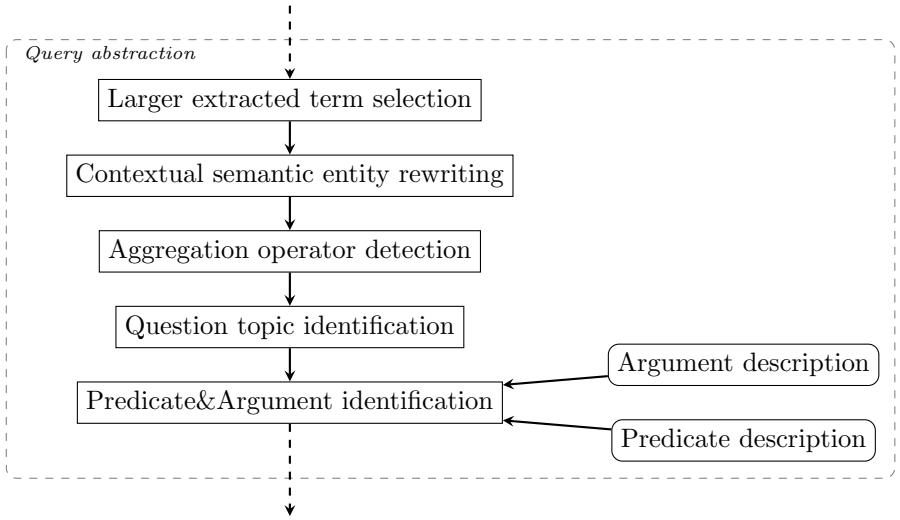

Fig. 4. Question abstraction process. Square boxes represent the abstraction steps of questions. Rounded boxes indicate the resources used.

tions are identified. These elements are recorded as coordinated.

Figure 5 presents a graphical representation and abstraction of the example question.

3.3. Query Construction

The objective of the *query construction* step is to connect previously identified elements and to build a representation of the SPARQL graph pattern (introduced by the keyword `WHERE`). Figure 6 presents the architecture of the query construction. Thus, the predicate arguments are instantiated by URIs associated with the subjects, objects, variables, and numerical values or by strings. For each question, we perform several connections:

1. The question topic is connected to the predicate(s). A variable is associated with the question topic and the predicate arguments that match the semantic type of the question topic. Note that at the end of this stage, the question topic may remain non-associated to any predicate. In the example from Figure 3, the variable `?v0` represents the association between the question topic and the subject (with the expected type **drugbank/drugs**) of the predicate **state**;
2. The predicate arguments are connected to subjects and objects identified during the question abstraction: they concern elements referring to URIs. Moreover, each predicate within the conjunction scope is duplicated and its arguments are also connected to the subject and objects if needed;
3. The predicates are connected between them through their subjects and objects. The connection between two predicates is then represented by a variable;
4. The predicates from different datasets are connected. We use the `sameAs` description to identify URIs referring to the same element. New variables are defined to connect two predicates;
5. The remaining question topic is connected to arguments of the `sameAs` predicate;
6. The arguments corresponding to the `string` type are connected with the extracted terms. We assume these arguments will be related to the string matching operator `REGEX`. Thus, the 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 processed: the predicates are marked as negated and the arguments corresponding to negated terms are included in a new predicate `rdf:type` if required.

At this stage, each question is fully translated into a representation of the SPARQL query. Fig-

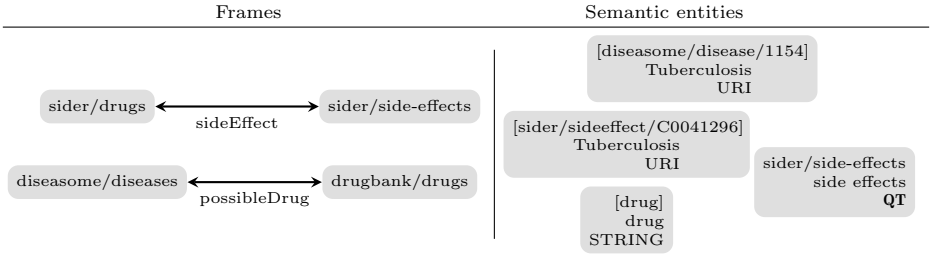

Fig. 5. Example of question abstraction (question#22 of the QALD-4 test set)

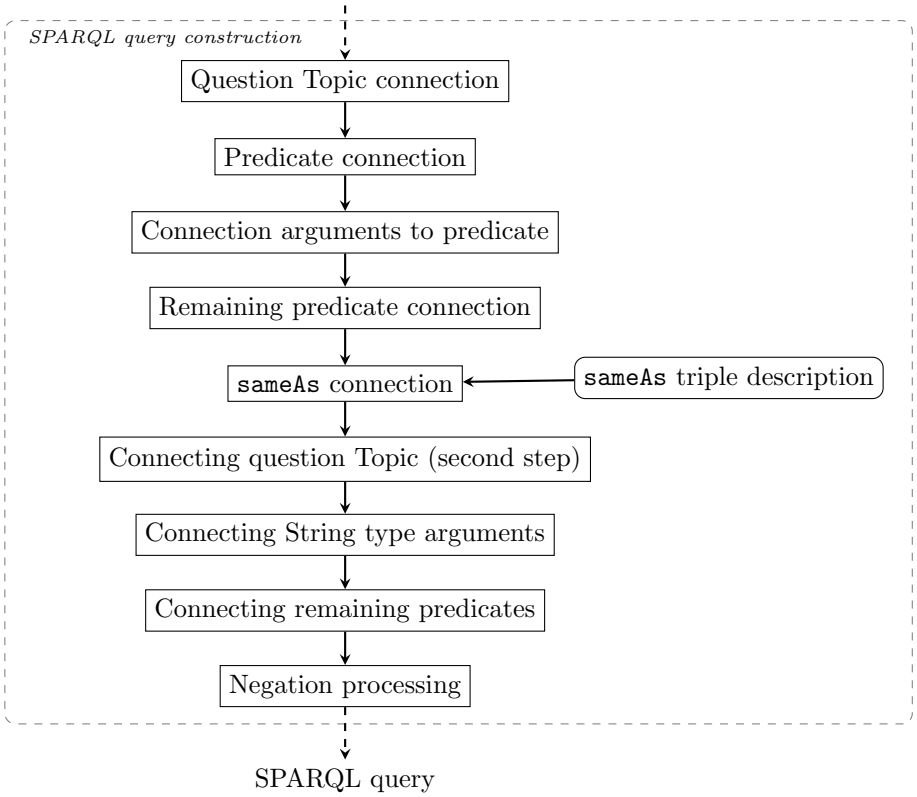

Fig. 6. Query construction process. Square boxes represent the construction steps of queries. Rounded boxes indicate the resource used.

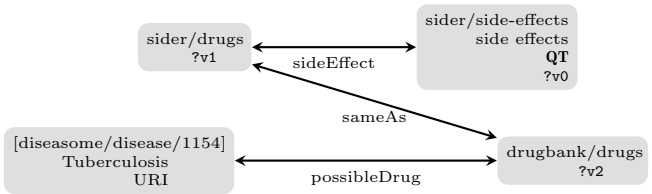

Fig. 7. Example of query construction (question#22 of the QALD-4 test set)

ure 7 illustrates the construction of the query corresponding to the example question.

3.4. Query Generation

The SPARQL query representation built during the *query construction* step is used to generate the

SPARQL query string. The process is composed of two parts:

1. The generation of the result form which takes into account the expected type of the result form (`ASK` or `SELECT`), the presence of aggregation operators and the variable associated with the question topic;
2. The generation of the graph pattern. This part consists of the generation of strings for representing each RDF triple and the filtering if the predicates are negated terms. But when aggregation operators are used, we also need to recursively generate sub-queries for computing the subsets of expressions, before their aggregation. In the example from Figure 3, the predicate **state** is replaced by the corresponding URI and its object is replaced by the string **gas**.

The SPARQL queries have been submitted to a SPARQL end-point[10] and answers are collected for the evaluation. Figure 8 presents the generated query which corresponds to the example question.

4. Definition of the Semantic Resources

The method described above relies on: (1) existing biomedical resources that provide information on the semantic entities (Sect. 4.1); (2) additional resources specifically collected and built to support the method (Sect. 4.2).

4.1. Domain-specific Resources

To process the set of questions treated, we used three biomedical resources:

- Drugbank[11] knowledge base is dedicated to drugs [28]. It merges chemical, pharmacological and pharmaceutical information from other available knowledge bases. We exploited the documentation[12] of this resource to define the rewriting rules and regular expressions for the named entity recognition;
- Diseasome[13] is dedicated to diseases and genes linked among them by known disorder/gene associations [9]. It provides a single framework with all known phenotypes and disease gene associations, indicating the common genetic origin of many diseases. We exploited the RDF triples and the documentation of the resource to define the rewriting rules;
- Sider[14] is dedicated to adverse drug effects of drugs [14]. It contains information on marketed medicines and their recorded adverse drug reactions. Information is extracted from public documents and package inserts. The available information includes side effect frequency, drug and side effect classifications, as well as links to other data, such as drug-target relations.

The content of each resource is provided in specific format: RDF triples *subject predicate object*. In that respect, we also exploit their RDF schema to define frames (see Sect. 4.2).

4.2. Additional Resources for the Question Annotation

On the basis of the RDF triples, frames are built from the RDF schemas in which the RDF predicate is the frame predicate, and subject and object of the RDF triples are the frame elements. This also includes the OWL `sameAs` triples. Several types of frame entities are isolated:

- As indicated, subject, object and predicate become semantic entities. At least one of them must occur in questions: in this way, the frames are the main resources for rewriting questions in queries;
- The vocabulary specific to questions is also built. It covers for instance the aggregation operators and the types of questions;
- RDF literals, issued from named entity recognizer or term extractor, complete the resources. The RDF literals are detected with specifically designed automata that may rely on the source knowledge base documentation.

These entities are associated with the expected semantic types which allow creating the queries and

[10] For our experiments, we use the SPARQL end-point provided by the QALD-4 challenge `http://vtentacle.techfak.uni-bielefeld.de:443/sparql`

[11] `http://www.drugbank.ca`

[12] `http://www.drugbank.ca/documentation`

[13] `http://diseasome.eu`

[14] `http://sideeffects.embl.de`

```
SELECT DISTINCT ?v0
WHERE {
    ?v1 <http://www4.wiwiss.fu-berlin.de/sider/resource/sider/sideEffect> ?v0.
    <http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseases/1154>
            <http://www4.wiwiss.fu-berlin.de/diseasome/resource/diseasome/possibleDrug> ?v2.
    ?v1 <http://www.w3.org/2002/07/owl#sameAs> ?v2.
}
```

Fig. 8. Example of the query generation (question#22 of the QALD-4 test set)

rewriting the RDF triples into SPARQL queries. In that respect, we can process several types of data (IRIs, strings, common datatypes or regular expressions) when literals are expected.

Most of the entities are considered and processed through their semantic types, although some ambiguous entities (e.g. interaction or class) are considered atomically. For these, the rewriting rules are applied contextually to generate the semantic entities corresponding to the frames (see Sect. 3.2). When using the queries, the semantic types become variables and are used for connecting the edges of queries.

5. Experiments and Results

5.1. Training and Test question set

Because the complexity of the QALD-4 training and test sets is unbalanced (e.g. the QALD-4 training set does not contain questions with aggregation operators), we use the 50 questions of the QALD-4 training and test sets for fitting our system. The evaluation is performed on 27 new questions. The questions of this new test set are similar to the QALD-4 questions but may differ according to the semantic entities or the involved predicates. Our method is applied to this new test set without additional adaptations. Figure 9 presents a sample of questions from the new test set.

5.2. Evaluation Metrics

The generated SPARQL queries are evaluated through the answers they generate. The evaluation is performed with the following macro-measures [25]:

$$M_{\text{precision}} = \frac{\sum_{i=1}^{|q|} \frac{TP(q)}{TP(q)+FP(q)}}{|q|} \quad (1)$$

$$M_{\text{recall}} = \frac{\sum_{i=1}^{|q|} \frac{TP(q)}{TP(q)+FN(q)}}{|q|} \quad (2)$$

$$M_{\text{F-measure}} = \frac{2 \times M_{\text{precision}} \times M_{\text{recall}}}{M_{\text{precision}} + M_{\text{recall}}} \quad (3)$$

where $TP(q)$ are the correct answers, $FP(q)$ are the wrong answers and $FN(q)$ are the missing answers for the question q.

The use of macro-measures equally considers all the questions independently of the number of expected answers to the SPARQL queries.

5.3. Global Results

Table 1 presents the overall results obtained on the training and test sets. On the test set, the macro-F-measure is 0.78 with 0.81 precision and 0.76 recall, while on the training set, the macro-F-measure is 0.86 with 0.84 precision and 0.87 recall. Each question is processed in less than 2 seconds on a standard computer (2.7GHz dual-core CPU and 4 Gb of memory). Most of the computing time is spent for the linguistic and semantic annotation of the questions.

Table 1
Results obtained with the training and test sets

Query set	Training (50 Q)	Test (27 Q)
Correct Queries	39	20
M-precision	0.84	0.81
M-recall	0.87	0.76
M-F-measure	0.86	0.78

Our method always proposes syntactically correct SPARQL queries for all natural language questions. On the test set, concerning the answers generated over linked data, 20 questions provide the

Which foods does fluvoxamine interact with?
Are there drugs that target the Probable arabinosyltransferase A?
Which genes are associated with subtypes of rheumatoid arthritis?
Which disease has the highest degree?
Which targets are involved in immune function?

Fig. 9. Example of natural language questions from the new test set

exact expected answers, two questions return partial answers, and five questions return erroneous answers. On the training set, 39 SPARQL queries (out of the 50 questions) provide the expected answers, six questions return partial answers, and five questions return no answers.

5.4. Error Analysis

An analysis of the erroneous or partial answers shows that most of the errors are due to (i) the encoding of the `sameAs` predicate in the resources and (ii) the management of ambiguities in the questions. This last point has also been noticed in a previous work [18].

Regarding the former type of errors, we observe that, although our method generates the correct SPARQL query, the SPARQL end-point does not return the expected answers. We can observe that by switching the arguments of the `sameAs` predicate in the queries, the expected answers are returned. In that respect, it appears that the instances of this predicate do not encode the expected reflexivity of this relation in the source knowledge bases while our method assumes that the `sameAs` predicate is symmetric by definition.

As for the errors due to ambiguity, they mainly deal with:

- The annotation of semantic entities. For instance, in the question *Which genes are associated with breast cancer?*, *breast cancer* is correctly annotated, while the reference assumes it concerns the semantic entity *Breast cancer-1*.
- The intended meaning of the terms in the questions. Semantic entities mentioned in some questions may refer to specific entities while in other questions they refer to the general entities. For instance, in the question *What are enzymes of drugs used for anemia?*, the semantic entity *anemia* refers to all types of anemia (*Hypercholanemia*, *Hemolytic anemia*, *Aplastic anemia*, etc.), and not specifically to elements that contain the label *anemia*.

These two main problems can be solved by using regular expressions in SPARQL graphs rather than URIs. However, we must test the influence of this modification on each query.

Other erroneous answers happen during the question abstraction step when the question topics are wrongly identified or when the contextual rewriting rules are not applied. Errors also occur during the query construction step: the method may abusively connect predicate arguments and semantic entities or, on contrary, it may not consider all the identified semantic entities. Further investigations have to be carried out to solve these limitations.

Besides, during the design of queries, we had difficulties to express some constraints in SPARQL. For instance, the question *Which approved drugs interact with calcium supplements?* requires to define a regular expression with the term *calcium supplement* while this term is only mentioned in conjunction with other supplements in the exploited knowledge bases (e.g. *Do not take calcium, aluminum, magnesium or Iron supplements within 2 hours of taking this medication.*). We assume that solving this difficulty requires a more sophisticated NLP processing of the textual elements of the RDF triples, such as parsing of the RDF textual elements, named entity and term recognition, identification of discontinuous terms and term variants.

Other limitations are related to the updating of the knowledge bases and the change of their structure. In the former case, it is only required to rebuild the semantic resources used for identifying the semantic entities. In the latter case, entire frames must be regenerated. This is our ongoing research work. Moreover, the addition of new resources such as Dailymed[15] is also related to these two problems.

[15] `http://dailymed.nlm.nih.gov/`

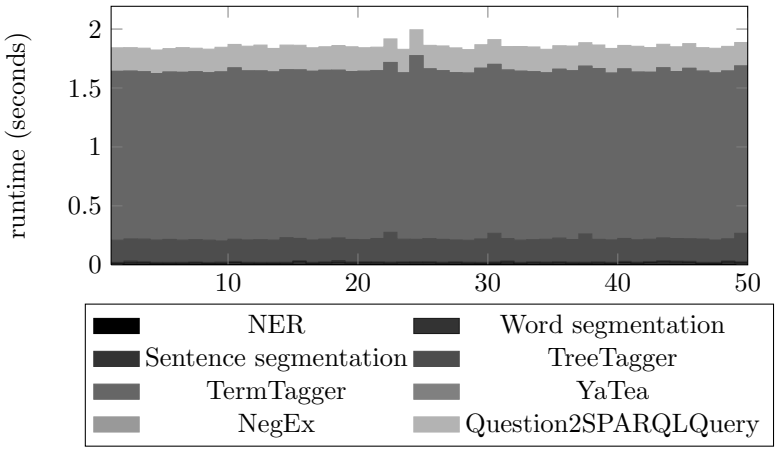

Fig. 10. Performance per sub-step for each question of the 50 questions of the training set.

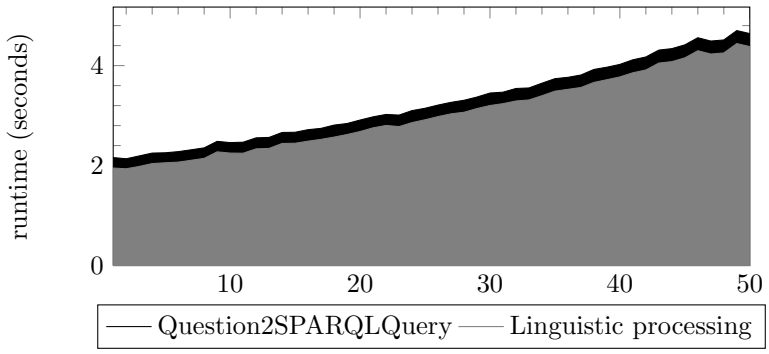

Fig. 11. System performance for an increasing number of 50 questions issued from the training set.

5.5. System performance

We analyzed the system performance for translating the 50 questions of the training set, on a computer with 4 Gb of memory and a 2.7GHz dual-core CPU. Figure 10 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, term extraction with YaTeA and negation scope identification with NegEx) and the question translation into SPARQL queries (Question2SPARQLQuery). Most of the processing time is dedicated to the TermTagger which aim is to recognize the semantic entities. With the internal Ogmios processing (i.e. mainly the control of inputs and outputs), each question is processed in 2 seconds on the average on the training set.

Figure 11 shows the overall system performance according to the number of questions to process. The variation of running time between processing one question and the whole set of questions is less than two second on the training set.

6. Conclusion

We proposed a rule-based method to translate natural language questions into SPARQL queries. The method relies on linguistic and semantic annotations of questions with NLP methods, semantic resources and the RDF triples description. We designed our approach on 50 biomedical questions proposed by the QALD-4 challenge, and tested it on 27 newly created questions. The method achieves good performance with 0.78 F-measure on the set of 27 questions.

Further work aims at addressing the limitations of our current method including the management of term ambiguity, the question abstraction, and the query construction. Moreover, to avoid the man-

ual definition of the dedicated resources required by our approach (frames, specific vocabulary and rewriting rules), we plan to investigate how to automatically build these dedicated resources from the RDF schemas of the Linked Data set. It will also facilitate the integration of other biomedical resources such as Dailymed or RxNorm [23], and the use of our method in text mining applications.

Acknowledgments

This work was partly funded through the project POMELO (*PathOlogies, MEdicaments, aLimentatiOn*) funded by the MESHS (Maison Européenne des Sciences de l'Homme et de la Société) under the framework *Projets Émergents*. We thank Arthur Plesak for his editorial assistance.

References

[1] A. B. Abacha and P. Zweigenbaum. Medical question answering: Translating medical questions into sparql queries. In *ACM SIGHIT International Health Informatics Symposium (IHI 2012)*, 2012.

[2] S. Aubin and T. Hamon. Improving term extraction with terminological resources. In T. Salakoski, F. Ginter, S. Pyysalo, and T. Pahikkala, editors, *Advances in Natural Language Processing (5th International Conference on NLP, FinTAL 2006)*, number 4139 in LNAI, pages 380–387. Springer, August 2006.

[3] O. Bodenreider, T. C. Rindflesch, and A. Burgun. Unsupervised, corpus-based method for extending a biomedical terminology. In *Workshop on Natural Language Processing in the Biomedical Domain (ACL2002)*, pages 53–60, 2002.

[4] W. Chapman, W. Bridewell, P. Hanbury, G. Cooper, and B. Buchanan. A simple algorithm for identifying negated findings and diseases in discharge summaries. *J Biomed Inform. 2001 Oct;34(5):*, 34(5):301–10, 2001.

[5] D. Damljanovic, M. Agatonovic, and H. Cunningham. 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*, ESWC'10, pages 106–120, Berlin, Heidelberg, 2010. Springer-Verlag.

[6] M. Damova, D. Dannélls, and R. Enache. Multilingual retrieval interface for structured data on the web. In *Workshop on Natural Language Interfaces for Web of Data*, 2014.

[7] E. Franconi, C. Gardent, X. Juarez-Castro, and L. Perez-Beltrachini. Quelo natural language interface: Generating queries and answer descriptions. In *Workshop on Natural Language Interfaces for Web of Data*, 2014.

[8] A. Freitas, J. ao C. Pereira da Silva, and E. Curry. On the semantic mapping of schema-agnostic queries: A preliminary study. In *Workshop on Natural Language Interfaces for Web of Data*, 2014.

[9] V. Janjić and N. Pržulj. The core diseasome. *Mol Biosyst*, 8(10):2614–2625, Aug 2012.

[10] E. Kaufmann and A. Bernstein. 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.

[11] J.-D. Kim and K. Cohen. Triple pattern variation operations for flexible graph search. In *Workshop on Natural Language Interfaces for Web of Data*, 2014.

[12] Y. Kim, Y. Hahm, D. Hwang, and K.-S. Choi. A non-morphological entity boundary detection approach for korean text. In *Workshop on Natural Language Interfaces for Web of Data*, 2014.

[13] N. Kuchmann-Beauger and M.-A. Aufaure. Natural language interfaces for datawarehouses. In *8èmes journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA 2012), Bordeaux*, volume B-8 of *RNTI*, pages 83–92, Paris, Juin 2012. Hermann.

[14] M. Kuhn, M. Campillos, I. Letunic, L. J. Jensen, and P. Bork. A side effect resource to capture phenotypic effects of drugs. *Molecular Systems Biology*, 6(1), 2010.

[15] V. le Clément de Saint-Marcq, Y. Deville, C. Solnon, and P.-A. Champin. Un solveur léger efficace pour interroger le web sémantique. In *JFPC 2012*, 2012.

[16] J. Lehmann and L. Bühmann. Autosparql: Let users query your knowledge base. In *Proceedings of the 8th extended semantic web conference on The semantic web: research and applications*, volume Part I, pages 63–79, 2011.

[17] L. Letouzey and A. Gabillon. Implementation d'un modèle de contrôle d'accès pour les documents rdf. In *SarSsi 2010*, 2010.

[18] D. Lukovnikov and A.-C. Ngonga Ngomo. SESSA - keyword-based entity search through coloured spreading activation. In *Workshop on Natural Language Interfaces for Web of Data*, 2014.

[19] A. T. McCray, A. C. Browne, and O. Bodenreider. The lexical properties of the gene ontology (GO). In *Proceedings of the AMIA 2002 Annual Symposium*, pages 504–508, 2002.

[20] G. Montoya, M.-E. Vidal, and M. Acosta. A heuristic-based approach for planning federated sparql queries. In *COLD*, 2012.

[21] C. Pradel, O. Haemmerlé, and N. Hernandez. Des patrons modulaires de requêtes SPARQL dans le système SWIP. In *Journées Francophones d'Ingénierie des Connaissances (IC)*, pages 412–428, juin 2012.

[22] A. Ranta. *Grammatical Framework: Programming with Multilingual Grammars*. CSLI Publications, Stanford, 2011.

[23] RxNorm, a standardized nomenclature for clinical drugs. Technical report, National Library of Medicine, Bethesda, Maryland, 2009. Available at `www.nlm.nih.gov/research/umls/rxnorm/docs/index.html`.

[24] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In D. Jones and H. Somers, editors, *New*

Methods in Language Processing Studies in Computational Linguistics, 1997.

[25] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[26] V. Tablan, D. Damljanovic, and K. Bontcheva. A natural language query interface to structured information. In L. N. in Computer Science Volume 5021, editor, *The Semantic Web: Research and Applications*, pages 361–375, 2008.

[27] C. Unger, L. Bühmann, J. Lehmann, A.-C. N. Ngomo, D. Gerber, and P. Cimiano. Template-based question answering over rdf data. In *WWW*, pages 639–648, 2012.

[28] D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey. Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Research*, 34:D668–D672, 2006. Database issue.