From Literature to Knowledge: Exploiting PubMed to Answer Biomedical Questions in Natural Language

Pinaki Bhaskar, Marina Buzzi, Filippo Geraci(✉), and Marco Pellegrini

CNR, Institute for Informatics and Telematics, Via G. Moruzzi 1, Pisa, Italy
{pinaki.bhaskar,marina.buzzi,filippo.geraci,
marco.pellegrini}@iit.cnr.it

Abstract. Researchers, practitioners and the general public strive to be constantly up to date with the latest developments in the subjects of biomedical research of their interest. Meanwhile the collection of high quality research papers freely available on the Web has increase dramatically in the last few years and this trend is likely to continue. This state of facts brings about opportunities as well as challenges for the construction of effective web-based searching tools. Question/Answering systems based on user interactions in Natural Language have emerged as a promising alternative to traditional keyword based search engines. However this technology still needs to mature in order to fulfill its promises. In this paper we present and test a new graph-based proof-of-concept paradigm for processing the knowledge base and the user queries expressed in natural Language. The user query is mapped as a subgraph matching problem onto the internal graph representation, and thus can handle efficiently also partial matches. Preliminary user-based output quality measurements confirm the viability of our method.

1 Introduction

The steady increase of the amount of biomedical data available in internet accessible repositories in the last two decades has come together with a similar increase in internet accessible biomedical literature (full papers or abstracts). Moreover, this trend has been reinforced by the fact that a few entry points (like the MEDLINE/PubMed repository) are sufficient to obtain a fairly representative view of the worldwide high quality biomedical scientific production. This state of facts brings about opportunities as well challenges.

The opportunity is for researchers, practitioners and the general public to be constantly up to date with the latest developments and findings in their medical (sub)field of interest from authoritative sources. For example, the goal of "evidence-based medicine" (EBM) [21], that is the idea that the diagnosis of individual patients can be based on, or at least supported by, finding relevant facts via "ad-hoc" searches in the biomedical literature, is indeed made more realistic by the availability at a finger-tip of the whole corpus of recent literature.

M.E. Renda et al. (Eds.): ITBAM 2015, LNCS 9267, pp. 3–15, 2015.
DOI: 10.1007/978-3-319-22741-2_1

The challenge is mainly in bridging the gap between rather vaguely defined and informally expressed user needs and the rigidity of standard traditional key-word based search engine interfaces currently in use.

To cope with this challenge a new generation of IR systems has emerged, which are based on formulating questions in unstructured natural language, and expecting as an answer a specific concise description in natural language of a relevant fact extracted from the appropriate knowledge base (QA systems).

A recent development, which witness the pressing need for technological improvements in Biomedical QA systems, has been the organization of dedicated QA tool evaluation challenges (e.g. http://www.bioasq.org/ now in its third year, and the medical track of CLEF http://www.clef-initiative.eu/).

In this paper we describe a proof-of-concept system with the intent of exploring the potentiality of novel graph representation of the knowledge-base and the use of efficient on-line sub-graph pattern matching algorithms in order to extract at query-time a pool of ranked relevant text snippets. This approach can be contrasted with that of a recent work by D. Hristovski et al. [13] where the the semantic relationship among named entities extracted from the off-line literature digestion are just fed to a database (thus missing the key graph abstraction). The user query in [13] is directly translated into a direct single query to the database, expressed with the sql-like Lucene DB query interface, oriented to exact matches. Our approach to query processing based on sub-graph pattern matching is more general and flexible, allowing for an easy expression also of partial matches.

2 Related Work

The first and most obvious way to solve a user's health information need is to ask a query to one of the generalist internet search engines. Wang et al. [26] compared usability and effectiveness of four general purpose search engines (namely: Google, Yahoo!, Bing, and Ask.com) for medical information retrieval, showing that current ranking methods need to be improved to support users in the task of retrieving accurate and useful information. Allam et al. [1] note that for controversial health issues the standard document ranking strategies adopted by generalist SE are at risk of introducing a strong cognitive bias. These two phenomena highlight the importance of developing simple user-friendly tools for direct access to authoritative medical information sources.

Specialized search engines based on medical literature such as MedSearch [17] and MedicoPort [5] have been developed within the general SE framework of a keyword search generating a list of snippets. In particular, Can et al. [5] focus on usability for untrained people, and show how increasing result precision is attained with various techniques including query conversions, synonyms mapping, focussed crawling, etc. Moreover the final ranking is based on a Pagerank-like method. Luo et al. [17] also addresses similar challenges using various techniques to improve its usability and the quality of search results. Lengthy queries are converted into shorter representative queries by extracting

a subset of important terms to make the query processing faster while improving the results' quality. Moreover, their system provides diversified search results by aggregating documents with similar informative content.

A third approach that has similarities, but also key differences with the two cases mentioned above is that of Clinical Decision Support Systems (DSS) that use medical literature SE as one of their information sources [7,23]. The role of DSS is in supporting clinicians in medical tasks (diagnoses, therapies, research, case studies, etc.) and must make extensive use of many different information sources, besides the literature, and in particular of the phenotype of the individual patients for which a decision has to be made.

Extended surveys on the biomedical QA systems have been compiled by S.J. Athenikos and H. Han [2] and O. Kolomiyets and MF. Moens [16]. They recognize this research area as one of the most dynamic within the more general field biomedical text mining [8,22,27].

Specific attempts in the direction of user-friendly accurate medical and/or biological QA systems are embodied in the systems askHERMES[1] [6], EAGLi[2] [10] and HONQA[3] [9], which are well described in the comparative usability study [3].

3 Method

In this section we describe a novel framework aimed at answering a user question posed in natural language. As a knowledge base we used the set of open access articles (where we extracted title and abstract) provided by PubMed in the NCBI ftp site.

In short, our system works in four main steps (see Fig. 1 for a graphical representation). Firstly, we extract from the question a set of relevant concepts and the relationships among them, then we retrieve all the sentences in our knowledge base containing these concepts with the appropriate relationships, later we assign a relevance score to each sentence according to the predicted meaning of the question, finally we merge sentences belonging to the same article into a single snippet, rank them and present the results to the user.

We modeled each sentence of a document as a graph where each node corresponds to a concept and an edge connects two concepts if they are related within the document. As dictionary of concepts we used the MeSH ontology. We determined the relationships among concepts using a customized set of rules. Then, we merged all the generated graphs into a single master graph collapsing all the nodes corresponding to the same concept into a single node. We used the graphDB model to store the graph into a MySQL database.

Once the user poses a new question, we convert it into a graph by extracting the relevant concepts and their relationships. The retrieval of all the possibly relevant sentences (and the corresponding articles) reduces to a sub-graph matching

[1] http://www.askhermes.org/.
[2] http://eagl.unige.ch/EAGLi/.
[3] http://services.hon.ch/cgi-bin/QA10.

problem. In our system we used the method described in [14] since it is based on the graphDB model. Than, we assign a score to each retrieved sentence according to its relevance to the question.

Our score is based on the relative position of terms in the phrase, on the classical TF-IDF, and on the type of the terms (biomedical/not biomedical). In particular, we give high score to terms according to their presence inside: the question, the MeSH dictionary or both.

In order to revert to articles instead of sentences as base result unit, we merge together in a single snippet the sentences belonging to the same article, averaging their score. Finally we rank the snippets by sorting them in decreasing score order and return them to the user.

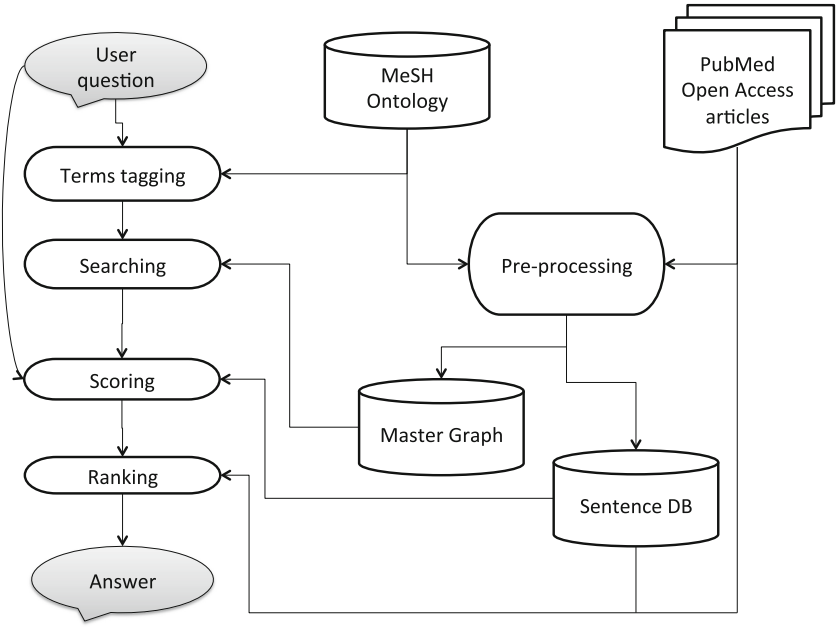

Fig. 1. Architecture of our QA framework

3.1 Knowledge Base Creation

The acceptance in the large majority of biomedical communities of the open access publication policy had the effect of making available a large collection of research articles that can be freely downloaded and used as a knowledge base. In particular, we focused on the *Open Access Subset* dataset provided by PubMed in the NCBI ftp site. This dataset is frequently updated and, at the moment we downloaded it, consisted in more than 790 thousand documents belonging to about 4 thousand journals for a total size of 52.92 GB.

Our knowledge base consists in a large undirected graph where each node corresponds to a well-defined biomedical concept or named entity, and the edges

model the relationship among pairs of concepts. For our retrieval purposes we did not consider the direction of the relationship and we did not assign a category to the nodes. Although this could appear as a simplistic approach, it guarantees a higher matching probability for small questions.

As a dictionary of biomedical terms we exploited the MeSH ontology. MeSH is a vast thesaurus organized as a hierarchy with 16 main categories including: anatomy, organisms, diseases, chemicals, drugs, processes therapeutic techniques and equipment. MeSH associates to each term a list of *entry terms* which consist in synonyms and alternate forms that can be used interchangeably. We collapsed all the entry terms into a single node of our master graph. Including articles in our knowledge base, we restricted to the title and abstract only. As observed in [4] this is not a limiting choice since most of the concepts exposed in a paper can be inferred from its title and abstract, while the rest of the paper contains only technical details. As a result, this restriction did not affect the effectiveness of our system.

As first step to include articles (from now on we refer to them indicating only title and abstract) in the knowledge base we split them into sentences. We treated titles as single phrases. Each sentence is scanned moving a sliding window and searching MeSH with the corresponding text. Since terms can have variable length ranging from a single word to five or six words, we used sliding windows of different sizes so that to capture all the possible medical terms inside the sentence. Notice that scanning with different window length can be done in parallel to avoid blowing the computational cost up.

More often multi-word terms are specific concepts where each component is itself a medical term. For example, the term *adult stem cell* includes the term *cell* which is more general and the term *adult*. In our case, however, using the most general term can determinate an increase of the number of retrieved irrelevant documents. In fact, in the above example, the focus of the term is not the cell in general, but a specific category of cells. Complicating the things, the term *adult* defines a concept belonging to a different MeSH category. However, considering it as an independent concept can be misunderstanding because in the above example it is used as an adjective to better specify the category of stem cells. In order to remove these general or misunderstanding terms, once we parsed a sentence we checked for nested terms (i.e. terms completely included into another term) and removed those included in a longer term.

A different situation arises when two concepts overlap but they are not nested. For example, the snippet sentence *tandem repeats sequence analysis* contains two overlapping terms *tandem repeats sequence* and *sequence analysis*. In this case tandem repeats are the object of the sequence analysis and thus both terms are relevant and related. The overlap is only the result of the fact that the two concepts are close in the text and have a word in common. In addition, the concept *tandem repeat sequence* has the alternative form *tandem repeat* in its entry terms list. Using this alternative form the two concepts of the example stop to overlap.

Inferring relationships among relevant terms in a text is a key point in the natural language processing field and general rules are far apart. In absence

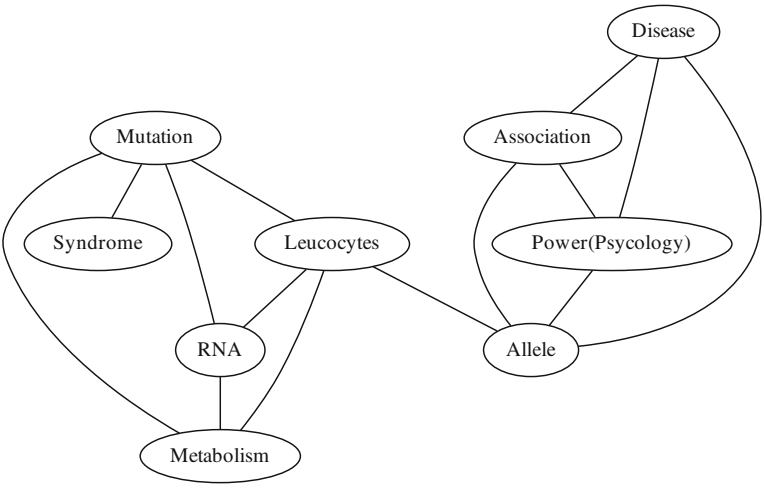

Fig. 2. Generated graph for two abstracts

of manually curated training data that enable the use of statistical methods, only rule-based algorithms are available [20]. Following this latter approach we observed that a minimal set of rules is able to correctly capture most of the relationships among medical concepts contained in an abstract. We observed that the distance between two related concepts is quite small and never exceeds the sentence. If a certain term is semantically connected to two concepts belonging to different sentences it is repeated in both phrases. According to this observation we constrained our system to check relationships only among terms within the same sentence.

Another agreed statement in natural language processing is that relationships are most often expressed with verbs. Motivated from this assumption, some effort has been done to create manually curated resources that are freely available. Among them, VerbNet [15] is one of the most complete for English. We observed that, in the abstracts, the verb is likely to be between the two related terms. The most frequent exception is when, instead of a single term, a sentence contains: two terms connected through a conjunction, or a list of terms. We treated lists as if they were a single concept. For example, in the sentence: *indels* and *SNPs* are common genomic *mutations*, we connect independently both the terms *indels* and *SNPs* with *mutations*, but we do not connect them among each other.

In Fig. 2 we show an example outcome of the processing of two abstracts.

3.2 Data Structure Organization

Data organization is a particularly important part of the structural design of those applications required to manage large datasets. Different data structure designs, can lead to a drastic speeded up of certain operators at the cost of a lower performance of other operations. In our QA system the three biggest data structure store: the sentences, the MeSH ontology and the master graph. Except at pre-processing time, all these structures do not require any on-line update.

Since the retrieval of sentences does not require full text searching, but it is only based on a pattern matching over the master graph, we can store them into a SQL table. We used a unique external key to associate the nodes of the master graph (namely the biomedical terms) with the sentences where they have been identified.

MeSH terms are accessed at query time to identify biomedical terms in the question. This operation can be efficiently done using an inverted index. Standard DBMSs (i.e. MySQL, postgreSQL) allow indexing string fields using the standard *select* command for querying. As a result, we were able to map the MeSH ontology into an SQL table.

Efficiently managing a large graph with thousands of nodes and millions of edges can be a thorny task. Trivial solutions like the adjacency matrix or the adjacency list are impractical for memory or performance reasons. We used the graphDB model [12] that exploits the DBMS strength to store the graph into a SQL database. According to this model, each node and its annotations are stored into a table and the edges are stored into another table. One of the practical advantages of the graphDB model is that it provides a simple and scalable way to store large graphs without requiring explicit handling of low-level details such as the memory management and the possible concurrent access. As shown in [14], another important characteristic of this model is that all the instances of a pattern can be efficiently retrieved using a limited number of *select* operations and a single *join*.

3.3 Question Answering Process

The first step of the question answering process consists in translating the user question into a graph. This process is done using the same procedure described in Sect. 3.1 for abstract preprocessing.

Searching. We use the question graph is as a model to perform a structural pattern-matching search using the algorithm proposed in [14]. In short, the algorithm works as follows: given an arbitrary pivot node, the question graph is divided by mean of a depth-first visit into a limited number of paths originating from the pivot. Each of these paths is retrieved with a single SQL *select* operation. Returned paths are pruned and merged using a SQL *join*. Since in our master graph nodes corresponding to the same concept have been collapsed into a single node, the output of the searching procedure consists in a single resulting graph.

Reverting from the subset of retrieved nodes on the master graph to sentences is done maintaining a mapping between concept nodes and IDs of the sentences where the concepts have been identified. According to the approximate pattern-matching model [11] it is suffice that all the retrieved nodes map a certain sentence ID to add it to the list of results.

Scoring. In order to evaluate the relatedness of a result sentence (from now on referred as answer) with the question, our system tokenizes the answer and identifies three categories of terms:

- **question's medical term**: that are biomedical terms belonging to both the question and the answer;
- **general medical term**: that are biomedical terms belonging to the answer but not present in the question;
- **general question's term**: that consist in the words of the question not tagged as medical terms.

Only answer's terms belonging to one of these three categories will contribute to the scoring and thus to the final ranking. Our score leverage on: the distance from the boundaries of the sentence, the TF-IDF, and the type (medical/non medical) of the terms. Let $A = \{t_1, t_2, \ldots, t_n\}$ be an answer with n terms and t_i be the i-esim term of A. We define the relatedness of the answer A with the question Q as:

$$R(A,Q) = \sum_{i=1}^{n} \left(\lambda(t_i) + i \left(1 - \frac{i-1}{n} \right) + \frac{tf(t_i)}{df(t_i)} \right) \times b(t_i) \tag{1}$$

where $\lambda()$ is:

$$\lambda(t_i) = \begin{cases} \lambda \text{ if } \neg\exists j < i \quad : \quad t_j = t_i \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

the boosting factor $b()$ is:

$$b(t_i) = \begin{cases} 5 \text{ if } t_i \text{ is a question's medical term} \\ 3 \text{ if } t_i \text{ is a general medical term} \\ 1 \text{ if } t_i \text{ is a general question's term} \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

and $\frac{tf(t_i)}{df(t_i)}$ is the standard TF-IDF for the term t_i in the knowledge base.

The rationale of the function $\lambda()$ is that of assigning an extra score to the first occurrence of a new term. As a result, an answer matching all the question's terms will have a higher score than an answer matching multiple times a subset of the question's terms. We empirically set $\lambda = 50$ so that to ensure that the score of a term occurring multiple times has a low probability to become greater than the sum of the scores of two terms occurring only once.

In order to define a priority order among the three categories of terms, the boosting factor $b()$ is introduced in Eq. (1). In absence of a formal criterion, the values of the boosting factor $b()$ have been empirically set.

Except stop words, general question terms are very relevant words that play an important role to clarify the meaning of a question and thus their contribution to the relevance score is essential. However, general words are more common than medical concepts and, thus, the term $\lambda()$ in Eq. (1) can dominate the overall score. In order to avoid this problem, we set $\lambda = 0$ for this category of terms.

Ranking. After scoring we merge together all the sentences belonging to the same article into a single snippet averaging their score. Then we rank the snippets by sorting them in decreasing score order. Finally, the ranked snippets are returned to the user interface for displaying.

4 User Interface for Biomedical Q&A

An intuitive user interaction is a key aspect for the success of a searching interface. A visual and aesthetic experience, emotion, identification, stimulation, meaning and value are main drivers of the user experience [18]. The variety of goals and skills of medical literature searching audience highlights the need to provide an interface available on different devices and browsers.

We designed our system user interface applying participatory design techniques, fast prototyping and early test with a restricted set of users (two people with biomedical background one computer scientist). This working organization allowed us a fast refinement loop of the system interface and functionalities.

We used HTML5, CSS3, and Javascript to develop a device independent and responsive UI. Our user interface has been designed as to enable a simple interaction and assuring a high level of accessibility and usability also for people with different abilities. In particular, we matched the constraints to assure an easy interaction also using assistive technologies (i.e. screen readers and voice synthesizers) for navigating the interface.

The interface design is minimalist and includes only a box for the search and the underlying results area. The result interface is split in 2 logical sections: the search box and the results, that are marked using WAI-ARIA role search (within the form) and the main with associated the aria-labelledby "Results". Furthermore heading level ($< h1 >$) can be associated to each result to enable easy jump from one to another using a screen reader command.

Focus is another important element to make interactions easier. Once the search page is loaded, we set the focus on the search box enabling the user to immediately formulate the question. Once the results are presented to the user, the focus moves to the first result to facilitate exploration.

5 Experiments and Validation

In this section we report details about our tests on the ability of our system to retrieve a relevant answer to user questions. As for most information retrieval tasks, in absence of a manually curated golden standard, the evaluation of a question answering system is a complicated task. The subjective nature of the concept of relevance makes the creation of a golden standard a hard task [24] and the formal assessment of a QA platform impractical. Among the few experimental assessment strategies, user evaluations are often preferred.

As mentioned in Sect. 3.1 we used the PubMed's *Open Access Subset* dataset to build the knowledge base. After preprocessing we obtained a graph with $27,149$ nodes and $4,732,885$ edges. We carried out a user study where we evaluated the system effectiveness for 96 well-formulated questions submitted from human experts. We requested the user to inspect the first ten results of each question and judge whether the answers are appropriate. Judgment is a score in the range $(1, 5)$ where 3 is the minimum score to consider the answer relevant. We left to the user the option of not evaluating some answers. Our evaluation

strategy allowed us to reduce the effort of the experts at the cost of the impossibility to measure the system's recall.

5.1 Evaluation Metrics

Let $C_k(Q) = \{c_1, \ldots, c_k\}$ be outcome of the judgment of the top k answers to a certain question Q, we define

$$N_k(Q) = \frac{\max_{i=1}^{k} c_i}{5}$$

as the normalised correctness rate of the question Q. The score $N_k(Q)$ is bounded in the range $[0, 1]$ and it is maximum only if there exists an answer with highest judgment score among the first k answers. Let n be the overall number of evaluated questions, we define the overall correctness rate N_k as

$$N_k = \sum_{i=1}^{n} N_k(Q_i).$$

We exploited different metrics to evaluate different aspects our system. In particular, we used: an extended version of the $c@1$ introduced in 2009 in the ResPubliQA exercise of the CLEF[4] initiative [19], the standard accuracy measure, and the mean reciprocal rank (MRR) [25].

Given a set of n questions, $c@1$ is a measure of the global correctness of the answers returned by a QA system. We extended the original formulae to the case of systems that return k answers as follows:

$$c@k = \frac{1}{n}\left(N_k + |U|\frac{N_k}{n}\right) \tag{4}$$

where U is the subset of unanswered questions. Notice that, since in the CLIF exercise the judgment of answers was binary (correct/not correct), the term N_k in the original formulae reduces to the number of correctly answered questions.

As a second performance measure, we used the *accuracy*@k which is a traditional measure applied to QA evaluations. We introduced a slight modification to the formulae in order to remove from the denominator unanswered questions.

$$accuracy@k = \frac{N_k}{n - |U|}$$

Ranking per se is one of the most important components of a QA system. The above measures give limited information about the ranking quality that can be derived only comparing the measures for different assignments of k. We included in our evaluation also the mean reciprocal rank (MRR) that provides a quantification of the ranking quality. Let $Rank_i$ be the position of the first correct answer for question Q_i, in our case we classify an answer as correct if the evaluator assigned a score higher or equal to 3. The mean reciprocal rank is defined as:

[4] http://www.clef-initiative.eu/.

$$MRR = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{Rank_i}$$

5.2 Results

Table 1 reports the result of the evaluation with the three measures for different values of k.

Table 1. Evaluation results

Measure	$k = 1$	$k = 5$	$k = 10$	$k > 10$
$c@k$	0.72	0.76	0.83	0.94
$accuracy@k$	0.56	0.67	0.78	0.91
MRR	0.66			

The MRR value 0.66 indicates that, on average, a good scoring answer is found either in the first or the second position in the returned list of results. The $c@k$ value for $k = 1$ and $k = 5$ indicates that about 85 % of the times an high quality answer is found within the top 5 answers. These measures attest the good quality of the ranking function. The $c@k$ value for $k > 10$ indicates that a good scoring answer is found for about 95 % of the queries. This indicates the quality of the subgraph matching strategy.

A manual inspection of the cases of failure (only 11 of the 96 questions remained unanswered) indicates that these are mostly due to mistakes in the first steps of the query processing procedure relative to named entity recognition with the MeSH ontology, due, for example, to the use of acronyms or commercial names of drugs. We reckon that we can improve our performance by integrating specialized knowledge data bases within our framework for specific medical sub-domains.

6 Conclusions

This work presents a proof-of-concept system for a QA search engine in the medical domain based on a Natural Language user interface and on an efficient partial sub-graph pattern matching methodology. Though the first evaluations are encouraging, thus establishing the valitdity of the approach, there is scope for improvements by incorporating additional domain knowledge and by refining the named-entity recognition step. We plan this as future work.

Acknowledgments. We acknowledge the support of the Italian Registry of ccTLD ".it" and the ERCIM 'Alain Bensoussan' Fellowship Programme.

References

1. Allam, A., Schulz, P., Nakamoto, K.: The impact of search engine selection and sorting criteria on vaccination beliefs and attitudes: two experiments manipulating google output. J. Med. Internet Res. **16**(4), e100 (2014)
2. Athenikos, S.J., Han, H.: Biomedical question answering: a survey. Comput. Meth. Prog. biomed. **99**(1), 1–24 (2010)
3. Bauer, M., Berleant, D.: Usability survey of biomedical question answering systems. Hum. Genomics **6**(1), 17 (2012)
4. Bleik, S., Mishra, M., Huan, J., Song, M.: Text categorization of biomedical data sets using graph kernels and a controlled vocabulary. IEEE/ACM Trans. Comput. Biol. Bioinf. **10**(5), 1211–1217 (2013)
5. Can, A.B., Baykal, N.: Medicoport: a medical search engine for all. Comput. Meth. Programs Biomed. **86**(1), 73–86 (2007)
6. Cao, Y., Liu, F., Simpson, P., Antieau, L., Bennett, A., Cimino, J.J., Ely, J., Yu, H.: Askhermes: an online question answering system for complex clinical questions. J. Biomed. Inf. **44**(2), 277–288 (2011)
7. Celi, L.A., Zimolzak, A.J., Stone, D.J.: Dynamic clinical data mining: search engine-based decision support. JMIR Med. Inf. **2**(1), e13 (2014)
8. Cohen, A.M., Hersh, W.R.: A survey of current work in biomedical text mining. Briefings Bioinf. **6**(1), 57–71 (2005)
9. Cruchet, S., Gaudinat, A., Boyer, C.: Supervised approach to recognize question type in a QA system for health. Stud. Health Technol. Inf. **136**, 407–412 (2008)
10. Gobeill, J., Patsche, E., Theodoro, D., Veuthey, A.L., Lovis, C., Ruch, P.: Question answering for biology and medicine. In: 9th International Conference on Information Technology and Applications in Biomedicine, 2009. ITAB 2009, pp. 1–5, November 2009
11. Gori, M., Maggini, M., Sarti, L.: Exact and approximate graph matching using random walks. IEEE Trans. Pattern Anal. Mach. Intell. **27**(7), 1100–1111 (2005)
12. Güting, R.H.: GraphDB: modeling and querying graphs in databases. In: VLDB, vol. 94, pp. 12–15. Citeseer (1994)
13. Hristovski, D., Dinevski, D., Kastrin, A., Rindflesch, T.C.: Biomedical question answering using semantic relations. BMC Bioinf. **16**(1), 16 (2015)
14. Kaplan, I.L., Abdulla, G.M., Brugger, S.T., Kohn, S.R.: Implementing graph pattern queries on a relational database. Technical report, Lammerce Livermore National Laboratory (2008)
15. Kipper, K., Korhonen, A., Ryant, N., Palmer, M.: Extending verbnet with novel verb classes. In: Proceedings of LREC, vol. 2006, p. 1. Citeseer (2006)
16. Kolomiyets, O., Moens, M.F.: A survey on question answering technology from an information retrieval perspective. Inf. Sci. **181**(24), 5412–5434 (2011)
17. Luo, G., Tang, C., Yang, H., Wei, X.: Medsearch: a specialized search engine for medical information retrieval. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 143–152. ACM (2008)
18. Nielsen, J.: Designing Web Usability: The Practice of Simplicity. New Riders Publishing, Thousand Oaks (1999)
19. Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, Á., Forăscu, C., Alegria, I., Giampiccolo, D., Moreau, N., Osenova, P.: Overview of respubliQA 2009: question answering evaluation over european legislation. In: Peters, C., Di Nunzio, G.M., Kurimo, M., Mandl, T., Mostefa, D., Peñas, A., Roda, G. (eds.) CLEF 2009. LNCS, vol. 6241, pp. 174–196. Springer, Heidelberg (2010)

20. Regev, Y., Finkelstein-Landau, M., Feldman, R., Gorodetsky, M., Zheng, X., Levy, S., Charlab, R., Lawrence, C., Lippert, R.A., Zhang, Q., Shatkay, H.: Rule-based extraction of experimental evidence in the biomedical domain: the KDD cup 2002 (task 1). SIGKDD Explor. Newsl. **4**(2), 90–92 (2002)
21. Sackett, D.L., Rosenberg, W.M.C., Gray, J.A.M., Haynes, R.B., Richardson, W.S.: Evidence based medicine: what it is and what it isn't. BMJ **312**(7023), 71–72 (1996)
22. Simpson, M.S., Demner-Fushman, D.: Biomedical text mining: a survey of recent progress. In: Aggarwal, C.C., Zhai, C. (eds.) Mining Text Data, pp. 465–517. Springer, New York (2012)
23. Soldaini, L., Cohan, A., Yates, A., Goharian, N., Frieder, O.: Retrieving medical literature for clinical decision support. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 538–549. Springer, Heidelberg (2015)
24. Voorhees, E.M., Tice, D.M.: Building a question answering test collection. In: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2000, pp. 200–207. ACM, New York (2000)
25. Voorhees, E.M., et al.: The TREC-8 question answering track report. In: TREC. vol. 99, pp. 77–82 (1999)
26. Wang, L., Wang, J., Wang, M., Li, Y., Liang, Y., Xu, D.: Using internet search engines to obtain medical information: a comparative study. J. Med. Internet Res. **14**(3), e74 (2012)
27. Zweigenbaum, P., Demner-Fushman, D., Yu, H., Cohen, K.B.: Frontiers of biomedical text mining: current progress. Briefings in Bioinf. **8**(5), 358–375 (2007)