Analyzing and Expanding NPDS Biomedical Record Databases via a Semantic Search Engine

Shiladitya Dutta and Carl Taswell MD, PhD

***Abstract*—CoVaSEA (Concept-Validating Search Engine Agent) is an automated semantic search engine that is interoperable with the Nexus-PORTAL-DOORS System(NPDS). Interoperable with NPDS, CoVaSEA serves to apply SPARQL-based search to external lexical databases by converting relevant biomedical resources to a linked data format which can be searched by the query engine. CoVaSEA consists of 3 components: the federated query engine searching, the web-crawler expanding, and the quadstore storing. Acting in concert, these can provide a composite open-web SPARQL search utility. For researchers, this opens the possibility to search external biomedical databases via a semantic web approach and is a solution for situations in which SPARQL queries need to be applied to non-linked data resources.**

***Index Terms*—Nexus-PORTAL-DOORS System, CoVaSEA, SPARQL, web-crawler, semantic web**

Introduction

The Nexus-PORTAL-DOORS System (NPDS) is a software system that provides the capability to publish both semantic and lexical resources regarding specific target areas. NPDS has inbuilt REST API services via the Scribe Registrar along with a standardized messaging protocol, enabling exchange among client applications and servers. The structure of the Nexus PORTAL-DOORS System is sub-divided according to the Hierarchically Distributed Mobile Metadata architectural style[1] into 3 primary parts: Nexus, PORTAL, and DOORS. The PORTAL registry controls entities that have unique labels and their associated lexical meta-data. The DOORS directory contains the semantic meta-data which is primarily comes in the form of RDF descriptions. The Nexus diristry (an aggregation of the terms DIRectory and regISTRY) is a single server that serves as a combination of the PORTAL registry and DOORS directory.[2] Semantic descriptions are a prerequisite for many of the goals laid out for the Nexus PORTAL-DOORS System[3], however it is labor-intensive to manually annotate resources from the open-web. To remedy this, we have created CoVaSEA(Concept-Validating Search Engine Agent): an automated web crawler/query engine that can search and expand NPDS metadata records. With this system, users can apply semantic search to external biomedical resources, which may be valuable in a variety of tasks which benefit from a combination of a web-crawler's capability to search the open web with the versatility of a semantic query engine. .

Overview

As an expansion to the past section[4], CoVaSEA integrates several features which benefit NPDS including: **(A)** An implementation of SPARQL query based semantic search to allow retrieval and manipulation of linked data descriptions **(B)** Targeted web-crawling for relevant articles from external biomedical literature databases to expand NPDS records **(C)** Automated translation of free-form text abstracts into RDF triples to derive the semantic representations of lexical data. First the user inputs a Natural Language Query(NLQ) and a SPARQL query, which also has the option to be constructed for the user, into the CLI. These are passed to the interface component which calls the webcrawler. The webcrawler combs through the various databases and retrieves the raw text and the metadata which is passed to the natural language parser. On each article, the natural language parser transforms the text and metadata into semantic triples, which are stored in a local quadstore or DOORS directory. Then the interface components calls the SPARQL query engine. The SPARQL query engine searches through the local quadstore and DOORS directory and returns the requested results to the user. In terms of the structure of the program itself, the web crawler/search engine program consists of a number of interdependent components. Each of these components fulfills a general purpose. The methods are:

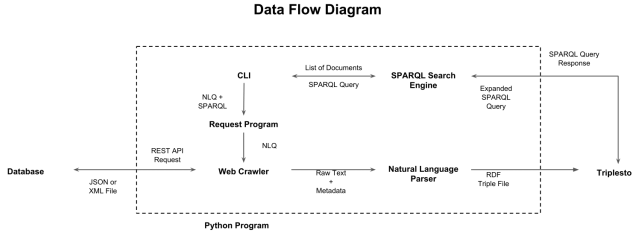

Fig. 1. Data flow diagram of the project and all of its included components

Document created June 20, 2018, revised September 30, 2018; typeset November 1, 2018.

- **Web Crawler** : Crawls through a multitude of sites in order to retrieve data. Currently the system can search through the article records on DOAJ, PubMed, Elsevier ScienceDirect, and CORE[5]. The webcrawler is similar in structure to Seung-Hong Bae. The webcrawler receives a requested query and then it searches through the available databases via REST API. The general method for searching through the databases is first a general lexical search query via the provided natural language query. The database then returns a number of articles. From there, the crawler searches through the list of returned articles and retrieves the basic-metadata from each one. It then passes the basic meta-data to the natural language parser. The basic meta-data constitutes of the abstract, title, author(s), database of origin, DOI(if available), and date of publication. The abstract will have its triples extracted and the rest will be included in the named graph of the article via the DublinCore ontology[6].
- **Natural Language Parser**: Receives the raw text from the web crawler and then parses it into logical form triples. These are put into RDF files and stored.

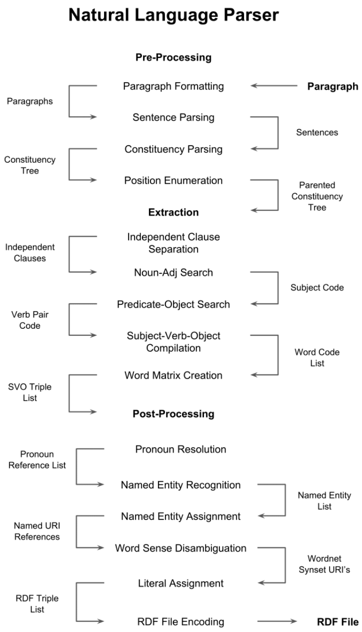

Fig. 2. Data flow diagram of the Natural Language to RDF parser

  - Pre-processing: Designed in order to preprocess the raw text to prepare it for parsing and also to perform tasks that depend on the raw texts such as Named Entity Recognition[7] and Co-Referencing[8]. Pre-processing preparation includes removing troublesome characters, respacing the text to ensure homogeneity, and parsing the paragraphs into sentences.
  - Extraction: Uses a constituency parse to extract relevant logical form triples. First independent clauses are separated from each other and a constituency parse occurs on each independent clause. Breadth-first and depth-first search are used to find subject and verb-object phrases. The breadth-first search is used to find the highest noun phrase in the tree. This noun phrase is then split up via conjunctions and any proximal adjectives are linked to the nouns. Then a two stage a breadth-first and depth-first search is used to find the verb. Breadth-first search is used to find the highest verb phrase and depth-first search is used to separate verb from the prepositional or adverb sections. The verbs and noun phrases are then broken into their individual parts and tagged with positions.
  - Post-processing: This step translates the logical form triples into RDF triples. The step primarily consists of graph compression. It does this by assigning URI's to each part of the logical form triples. First field-specific language and some named entities are assigned URI's via databases. Then standard words are assigned via WordNet[9] synsets using context and part of speech via Lesk[10]. This step aids in graph-compression and also allows for the full semantic meaning to be captured rather than just lexical meaning. Finally named entities and number are assigned to literals. These are all then packaged into a named graph.

- **SPARQL Analysis Engine**: Receives SPARQL query from user and then searches through the triple store to extract relevant data for the user. For a more conducive user experience, the query engine supports a SPARQL builder, thus circumventing the need to know SPARQL syntax. The system works by having the user create a series of either filtering, mandatory, or optional conditions in either an independent or nested connection. Thus the user can build a search query via simple fill-in-the-blank statements. Though it cannot replicate the full power of SPARQL syntax, it is a potent resource for users cannot use SPARQL. With the constructed or inputted query, CoVaSEA searches through the linked database. Due to the dynamic and heterogeneous nature of the data that CoVaSEA handles, the SPARQL query engine divides up the queries into sub-queries such that only the articles that are needed are compiled. This approach avoids the necessity of having to compile a computationally expensive local graph. Since the system is based off of named graphs, it is relatively easy to trace the article of origin of a particular triple if necessary. This iteration based approach helps to optimize runtime and memory-load on computers by eliminating the redundancy of loading unneeded graphs.

- **QuadStore**: To prevent redundant re-rendering of articles, CoVaSEA records the triples which are built by the web-crawler in either a local quadstore or a DOORS directory. Each of the articles in the records has its own named graph, allowing for the identification of the origin of triples. The graph is divided into two sections: the meta-data section and the semantic representation section. The meta-data section stores key information about the article such as author, title, publication date, and database of origin. The semantic representation section stores a set of triples which are a RDF portrayal of the article's abstract. The RDF files are sent and retrieved from the DOORS directory utilizing the Scribe API.

Results and Discussion

The primary point of possible expansion for the system is the Natural-Language Parser. Currently, the parser faces difficulty in 3 main areas: relevance of triples, accuracy of triples, and runtime. The first problem is the relevance of triples and how well they summarize the passages. There are several methods from which we can take inspiration from[11][12]. Most of these focus on document summarization by using a combination of the semantic graph structure of the document and pattern matching. The second issue is that the system cannot accurately parse advanced sentence structures. The best way to remedy this is to use a semantic frame solution such as FrameNet[13] or Boxer[14]. The final problem is that of optimization. The runtime for the parser in its current state is time-consuming in that it takes 30 seconds to parse a paragraph. This impedes us from performing whole articles parses with reasonable runtimes. The best way to solve this is to either use more advanced hardware or optimize the algorithms. Another section of improvement is the SPARQL query engine. SPARQL query expansion should be implemented to increase the breadth of SPARQL searches and to implement a reasoning engine such as RACER[15].

Conclusion

Here we presented a system in which resources from the open-web can be translated into machine-understandable semantic information and be searched via SPARQL. CoVaSEA has the capability to both search ”externally” with the web crawler for semantic data to expand the NPDS knowledge base and ”internally” with SPARQL to provide a method to navigate the data inside the DOORS directory. With the distinct advantage that the system is automated, thus can furnish large amounts semantic descriptions on a regular basis, CoVaSEA lays the groundwork for a variety of future NPDS applications for which linked data stores are a necessity along with providing a method to semantically search external biomedical literature databases.

Acknowledgment

I give my thanks to the other members of the Brain Health Alliance 2018 program for their assistance on technical and methodological points.

References

[1] C. Taswell, “A distributed infrastructure for metadata about metadata: The hdmm architectural style and portal-doors system,” *Future Internet*, vol. 2, no. 2, pp. 156–189, 2010.

[2] ——, “Doors to the semantic web and grid with a portal for biomedical computing,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 2, pp. 191–204, 2008.

[3] C. Taswell, G. TeleGenetics, and C. Ladera Ranch, “Portal-doors infrastructure system for translational biomedical informatics on the semantic web and grid,” *Proceedings of the American Medical Informatics Association Summit on Translational Bioinformatics, San Francisco, CA*, p. 43, 2008.

[4] S.-H. Bae, A. G. Craig, C. Taswell, *et al.*, “Expanding nexus diristries of dementia literature with the npds concept-validating search engine agent,”

[5] P. Knoth and Z. Zdrahal, “Core: Three access levels to underpin open access,” *D-Lib Magazine*, vol. 18, no. 11/12, 2012.

[6] S. Weibel, “The dublin core: A simple content description model for electronic resources,” *Bulletin of the American Society for Information Science and Technology*, vol. 24, no. 1, pp. 9–11, 1997.

[7] J. R. Finkel, T. Grenager, and C. Manning, “Incorporating non-local information into information extraction systems by gibbs sampling,” in *Proceedings of the 43rd annual meeting on association for computational linguistics*, Association for Computational Linguistics, 2005, pp. 363–370.

[8] K. Clark and C. D. Manning, “Entity-centric coreference resolution with model stacking,” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, vol. 1, 2015, pp. 1405–1415.

[9] G. A. Miller, “Wordnet:a lexical database for english,” *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[10] S. Banerjee and T. Pedersen, “An adapted lesk algorithm for word sense disambiguation using wordnet,” in *International conference on intelligent text processing and computational linguistics*, Springer, 2002, pp. 136–145.

[11] J. Leskovec, M. Grobelnik, and N. Milic-Frayling, “Learning sub-structures of document semantic graphs for document summarization,” 2004.

[12] Y. Shang, Y. Li, H. Lin, *et al.*, “Enhancing biomedical text summarization using semantic relation extraction,” *PLoS one*, vol. 6, no. 8, e23862, 2011.

[13] C. F. Baker, C. J. Fillmore, and J. B. Lowe, “The berkeley framenet project,” in *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, Association for Computational Linguistics, 1998, pp. 86–90.

[14] J. Bos, “Wide-coverage semantic analysis with boxer,” in *Proceedings of the 2008 Conference on Semantics in Text Processing*, Association for Computational Linguistics, 2008, pp. 277–286.

[15] V. Haarslev and R. Möller, “Racer: A core inference engine for the semantic web.,” in *EON*, vol. 87, 2003.