SPARQL-Based Search Engine and Agent for Finding Brain Literature and Converting References to NPDS Metadata Records

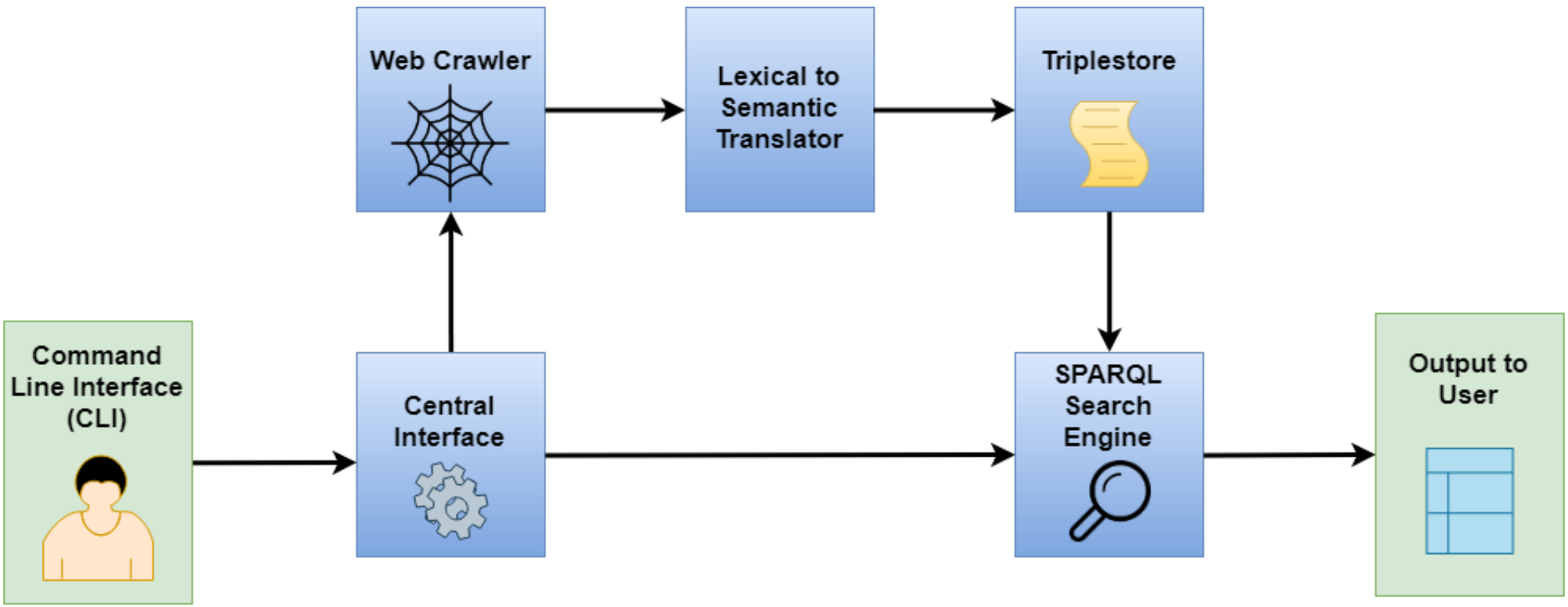
Shiladitya Dutta and Carl Taswell

December 2, 2018

Introduction

- The Nexus-PORTAL-DOORS System (NPDS) is a meta data management system that handles repositories of lexical and semantic representations of resources.
- Some tasks planned for NPDS require stores of brain literature descriptions and a method of retrieving and analyzing the semantic metadata
- To remedy this CoVaSEA (Concept-Validated Search Engine Agent) was developed
- Built via Python in Visual Studio using the NLTK and Stanford CoreNLP natural-language processing toolkits
- Consists of 3 main sections: Web crawler, Lexical to Semantic converter, and SPARQL query engine
- Combine to create a hybrid internal/external search system oriented towards brain literature and research

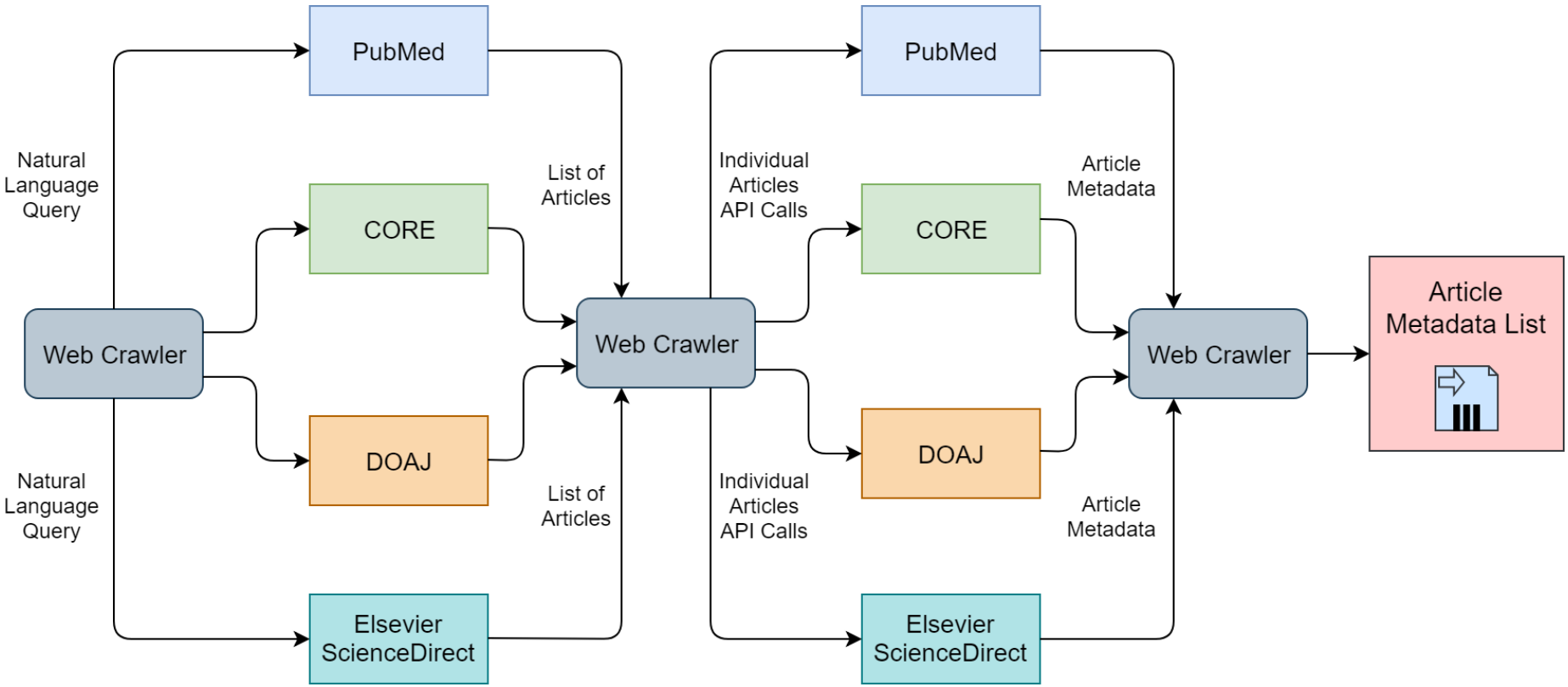
Data Flow Diagram



Web Crawler

- Based on the previous CoVaSEA web crawler developed in Javascript (Bae et al., 2017)
- The user inputs the database they want to search, the general search query that is used to search the literature database, and the number of articles they want to request
- Webcrawler has the option to go to 4 different brain literature databases: DOAJ, ScienceDirect, CORE, and PubMed
- Utilizes REST API to retrieve citation metadata (authors, name, publication date, etc.)
- First uses the inbuilt article search functionality with a user-provided general search query to find articles of interest
- Then retrieves the meta data for each article individually
- The meta data is passed to the lexical to semantic translator

Web Crawler Cont.



Overview of Lexical to Semantic Translation

- Objective is to translate the abstract received from crawler into Resource Description Framework (RDF) triples that describe the abstract
- Translation occurs in 3-steps:
  1. **Pre-processing:** Converts the abstract into sentence constituency trees
  2. **Extraction:** Derives subject-verb-object triples from the constituency trees
  3. **Post-processing:** Translates the subject-verb-object triples into RDF
- Creates a semantic description of the lexical data contained within the abstract for use by NPDS

Pre-processing

- First performs task that depend on or deal with the raw text of the abstract:
  - Edits the abstract so that it is in a standardized format for parsing (i.e. removing aberrant spacing or deleting tags from the beginning or end of abstract text).
  - Performs Co-reference resolution to determine what subjects the pronouns in the abstract refer to (Finkel et al., 2005)
  - Conducts Named Entity Recognition to identify proper nouns in the abstract (Recasens et al., 2013)
- Executes a constituency parse on the individual sentences to create a constituency tree of each sentence (Chen et al., 2014)
- The constituency tree acts as a tree-based syntactic representation of the sentence where the leaves are words and the nodes are grouping of the words (e.g. noun phrase, prepositional phrase, etc.)

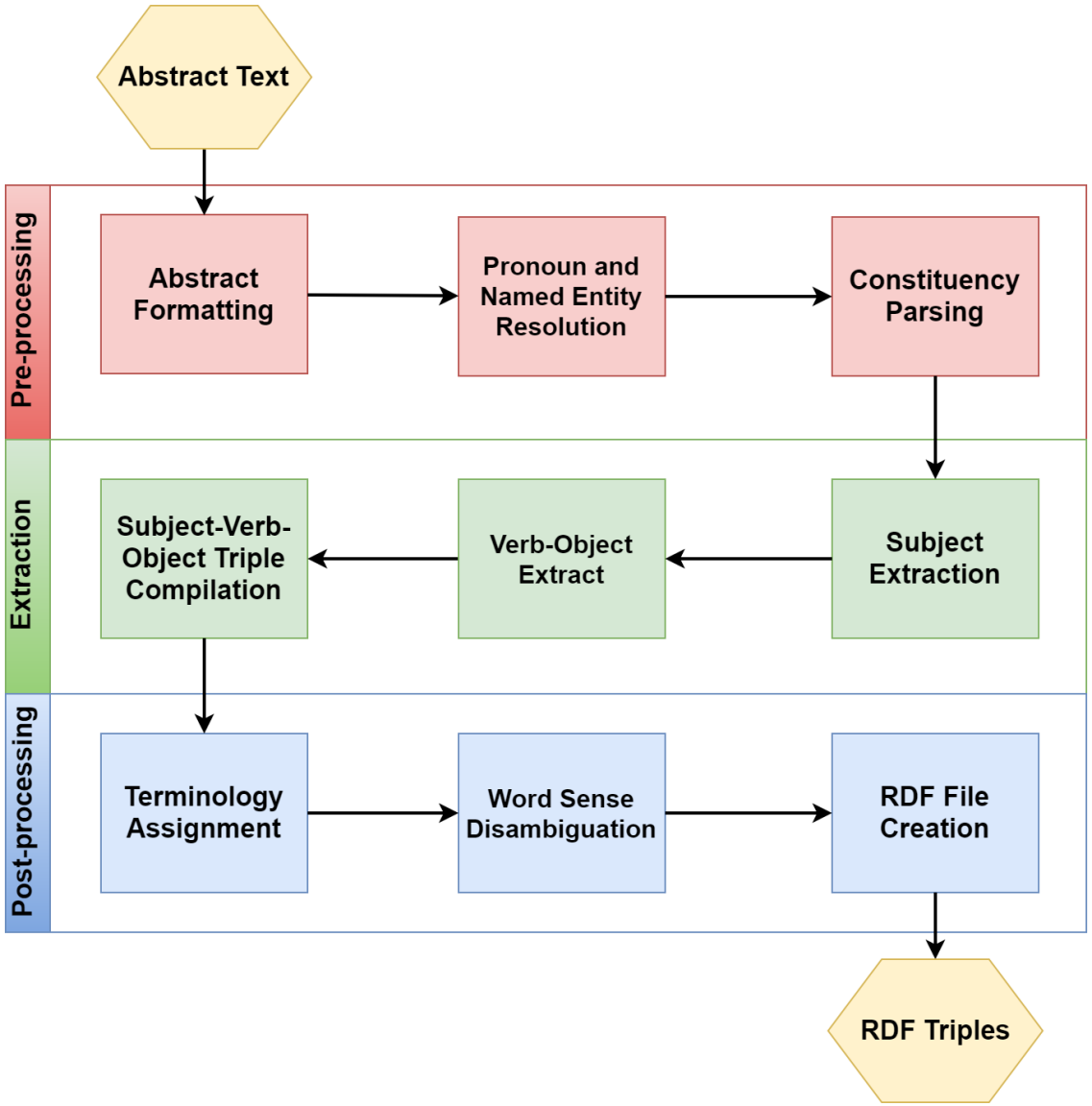
Extraction

- Derives Subject-Verb-Object triples from the constituency tree (Rusu et al., 2007)
- Breadth-first search is used to find the highest noun phrase in the tree
- The noun phrase is split into the individual subjects of the sentence and any adjectives in the noun phrase are linked to the subjects they are referring to
- Verb phrase is found by breadth-first search
- The verb is split from the object by using depth-first search on the verb phrase and the object is linked to its corresponding verb to form a verb-object pair
- The subject and verb-object pairs are combined into subject-verb-object triples

Post-processing

- Subject-verb-object triples are converted to RDF
- Each part of the subject-verb-object triples are assigned Unique Resource Identifiers (URI)
- Terminology is assigned a URI via domain-specific vocabulary databases
- Word Sense Disambiguation is performed by assigning standard nouns and verbs to WordNet synsets using the Lesk algorithm (Banerjee and Pederson, 2002)
- WordNet synsets (Miller, 1995) are groups of synonyms that are semantically equivalent for data retrieval purposes
- Names and numbers are put into RDF literals.
- The converted triples are encoded into RDF files for storage on the triplestore

Natural Language Processing Pipeline



Triple Store

- To store the semantic meta-data of the articles it has converted, CoVaSEA records the citation meta-data triples and the triples that are built by the natural-language parser
- Records can be stored in either a local triplestore and/or DOORS directory
- Graph consists of two sections: the citation metadata section and the semantic representation section.
- Citation metadata section stores basic metadata (author name, publication data, title, etc.)
- Semantic metadata stores triples derived from the abstract by the lexical to semantic translator

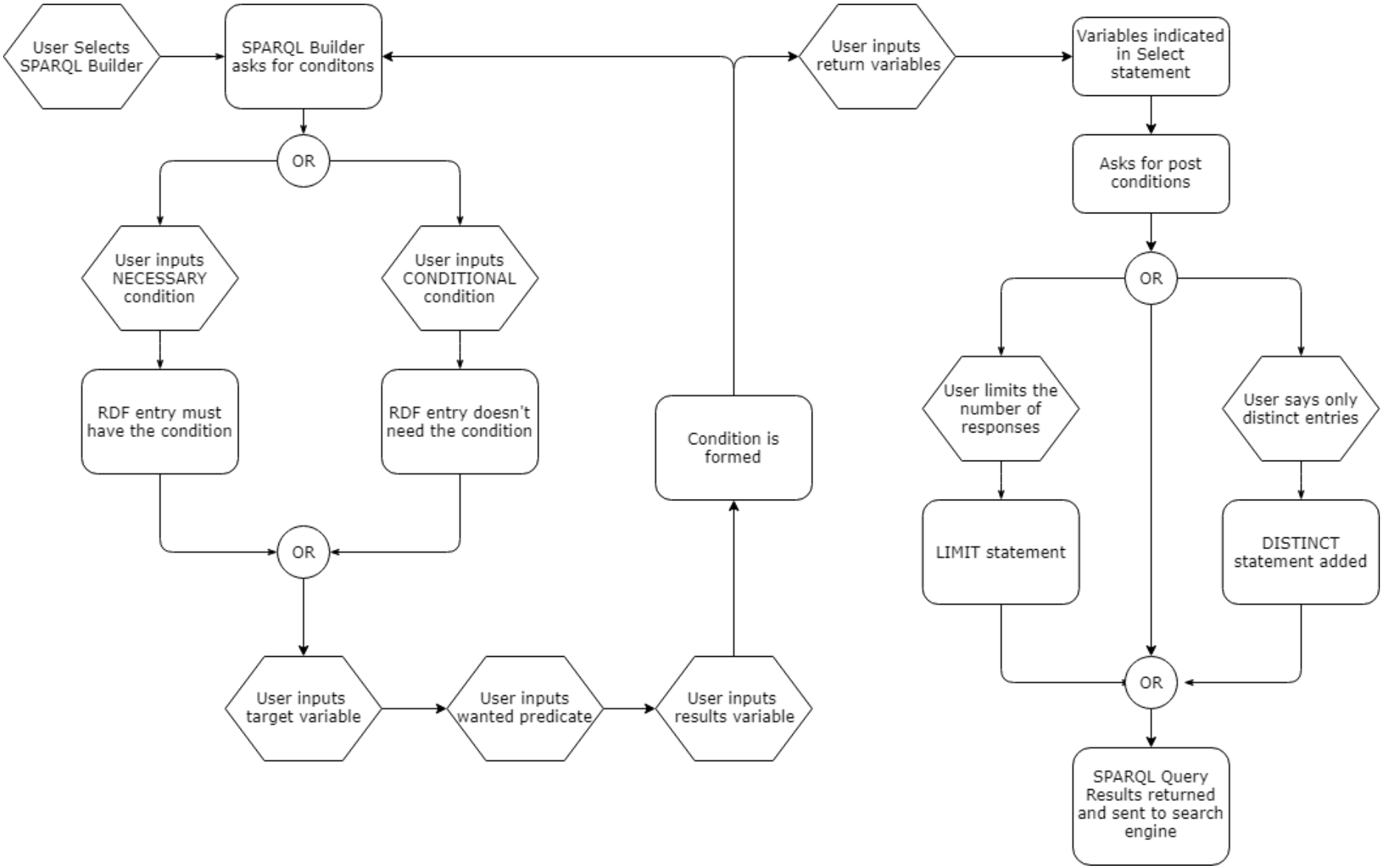
SPARQL Query Engine

- Performs a SPARQL Query search of the triplestore using RDFLib
- SPARQL allows for complex and versatile searches to be made by the user which query for specific data
- Compiles a local graph on the machine in order to perform the search
- The input has the option to be given directly by the user or be compiled via a SPARQL Query Builder
- The SPARQL Query builder is based on the Wikidata SPARQL Query builder (Vrandečić and Krötzsch, 2014)
- Main difference is that the CoVaSEA query builder allows for the user to specify the target variable of a condition whereas the Wikidata builder only allows for conditions to be targeted towards the article

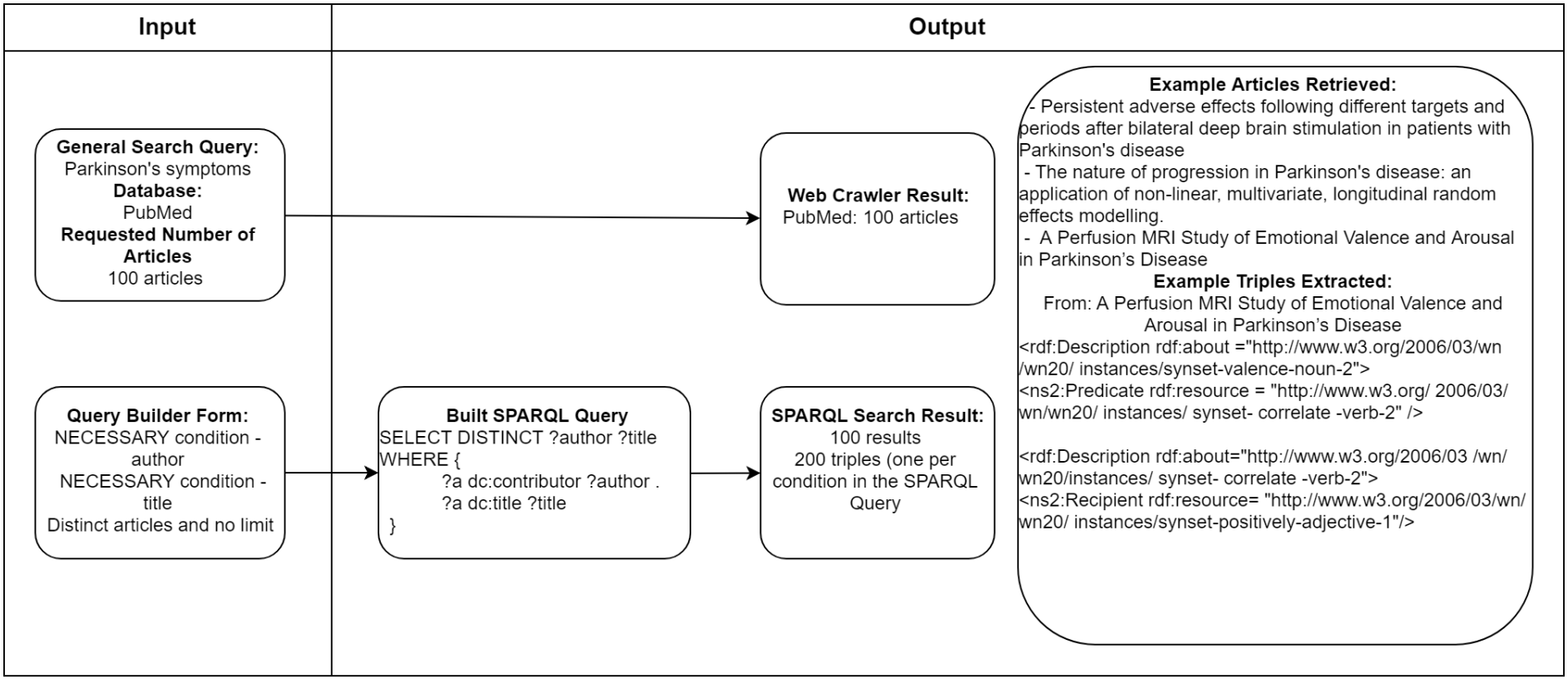
SPARQL Query Builder

- Resource for users who do not want to use SPARQL syntax to build a search query
- Is a SPARQL Builder form that helps users create their own queries
- First the user enters a series of conditions
- Each condition has four parts: type of condition, variable being searched, thing being searched for, and variable to which the result is assigned
- The type of condition can either be a necessary or optional
- Then the user decides which variables they want to return
- Finally, the user decides if they want want only distinct results and if they want to limit the amount of results
- Cannot replicate the full power of SPARQL syntax, but still a potent resource

SPARQL Query Builder Form



Example Query



Results

Note: Successful translation is defined by when the lexical to semantic translator is able to successfully convert an abstract to RDF triples

Database	# of Articles Requested	Query	# of Articles Received	Lexical to Semantic Abstract Translation	SPARQL Search Result	Runtime
DOAJ	10 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	10 articles	Successfully Translated: 10 articles # of Triples: 64	10 results 20 triples	37.12 seconds
	100 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	97 articles	Successfully Translated: 96 articles # of Triples: 573	96 results 192 triples	312.34 seconds
	1000 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	983 articles	Successfully Translated: 967 articles # of Triples: 6854	966 results 1932 triples	2589.32 seconds
PubMed	10 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	10 articles	Successfully Translated: 10 articles # of Triples: 93	10 results 20 triples	31.84 seconds
	100 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	99 articles	Successfully Translated: 99 articles # of Triples: 745	99 results 198 triples	283.31 seconds
	1000 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	1000 articles	Successfully Translated: 984 articles # of Triples: 8321	984 results 1968 triples	2391.12 seconds
Elsevier ScienceDirect	10 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	10 articles	Successfully Translated: 10 articles # of Triples: 74	10 results 20 triples	45.31 seconds
	100 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	97 articles	Successfully Translated: 96 articles # of Triples: 455	96 results 192 triples	332.14 seconds
	1000 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	998 articles	Successfully Translated: 954 articles # of Triples: 7213	954 results 1908 triples	2431.21 seconds
CORE	10 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	10 articles	Successfully Translated: 9 articles # of Triples: 50	9 results 18 triples	44.32 seconds
	100 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	100 articles	Successfully Translated: 94 articles # of Triples: 398	94 results 188 triples	390.32 seconds
	1000 articles	General Search: Parkinson's Symptoms SPARQL Search: Request for author and title	973 articles	Successfully Translated: 912 articles # of Triples: 8434	911 results 1822 triples	2945.32 seconds

Discussion

- Translation success rate is not 100% due to either irregular formatting of results files or inability to properly parse abstracts that have more advanced sentence structure
- Number of abstract-derived semantic triples varies highly depending on the length of the abstract
- The runtime scaling between the 10, 100 and 1000 article parse is not linear due to the constant runtime of initializing the Stanford NLP Parser and the initial web crawler natural-language search.

Conclusion

- Here we present CoVaSEA: a Web crawler/SPARQL query engine
- Combines the capability to search externally on the open-web for articles and internally in its previously built semantic records with SPARQL
- CoVaSEA has 3 main parts:
  1. The web crawler retrieves articles from brain literature databases
  2. The lexical to semantic converter converts retrieved text into RDF triples and stores in a triplestore
  3. SPARQL query engine allows users to search through the triplestore
- These mesh together to create a system that can access and broaden semantic records for brain literature and research for NPDS

References

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.

D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 740–750.

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.

G. A. Miller, "Wordnet: A lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

M. Recasens, M.-C. de Marneffe, and C. Potts, "The life and death of discourse entities: Identifying singleton mentions," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013, pp. 627–633.

D. Rusu, L. Dali, B. Fortuna, M. Grobelnik, and D. Mladenic, "Triplet extraction from sentences," in *Proceedings of the 10th International Multiconference" Information Society-IS*, 2007, pp. 8–12.

S.-H. Bae, A. G. Craig, C. Taswell, *et al.*, "Expanding nexus diristries of dementia literature with the npds concept-validating search engine agent,", 2017.

D. Vrandečić and M. Krötzsch, "Wikidata: A free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

Questions?

Brain Health Alliance Virtual Institute

Ladera Ranch, California

Shiladitya Dutta: sdutta@bhavi.us

Carl Taswell: ctaswell@bhavi.us