Brain Informatics 2018 B277

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

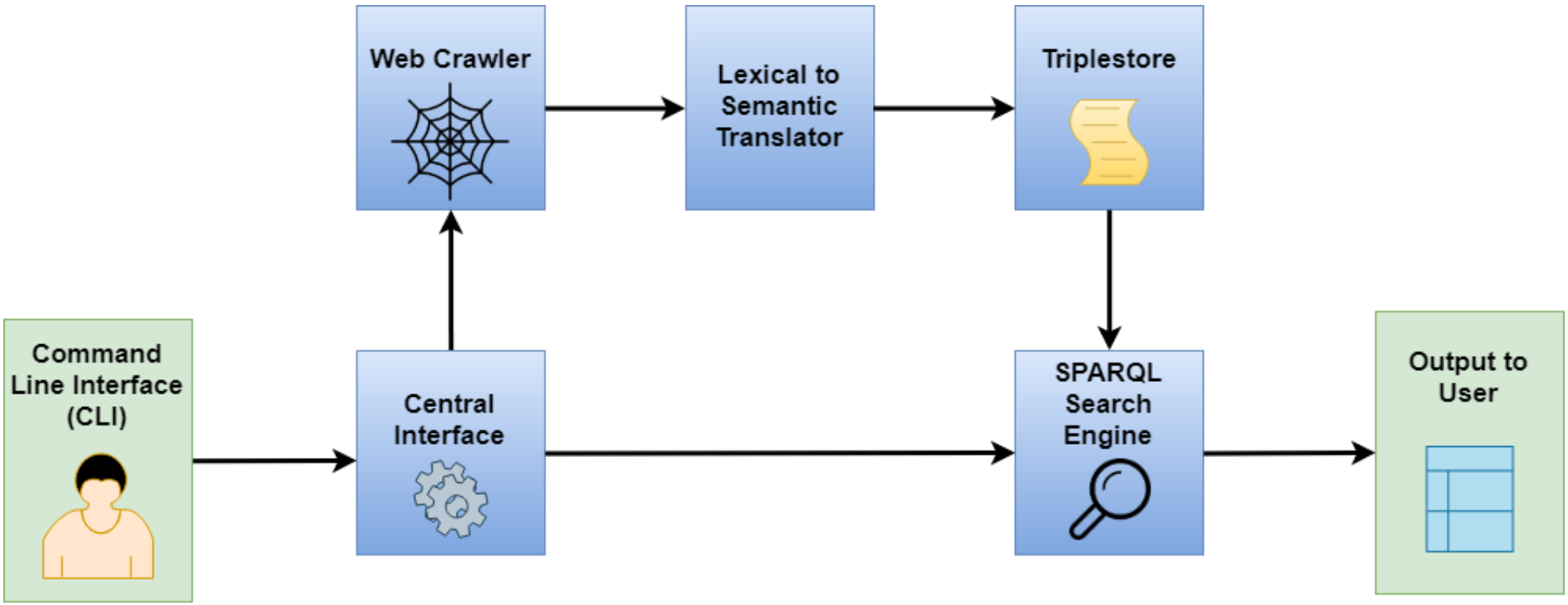
Shiladitya Dutta and Carl Taswell

December 9, 2018

Introduction

- The Nexus-PORTAL-DOORS System (NPDS) provides a metadata management system with distributed repositories of lexical and semantic representations of both online and offline resources
- Planned applications for NPDS, such as automated meta-analyses of the brain literature, require stores of research articles, descriptive metadata records about the articles, and tools and methods for retrieving and analyzing the semantic metadata
- CoVaSEA (Concept-Validated Search Engine Agent) has been developed to support the infrastructure for addressing this problem
- Built with Python in Microsoft Visual Studio using the NLTK and Stanford CoreNLP natural-language processing toolkits
- Consists of 3 main components: a web crawler, a lexical to semantic converter, and a SPARQL query engine
- Integrated to create a hybrid internal/external search system focused on the brain research literature

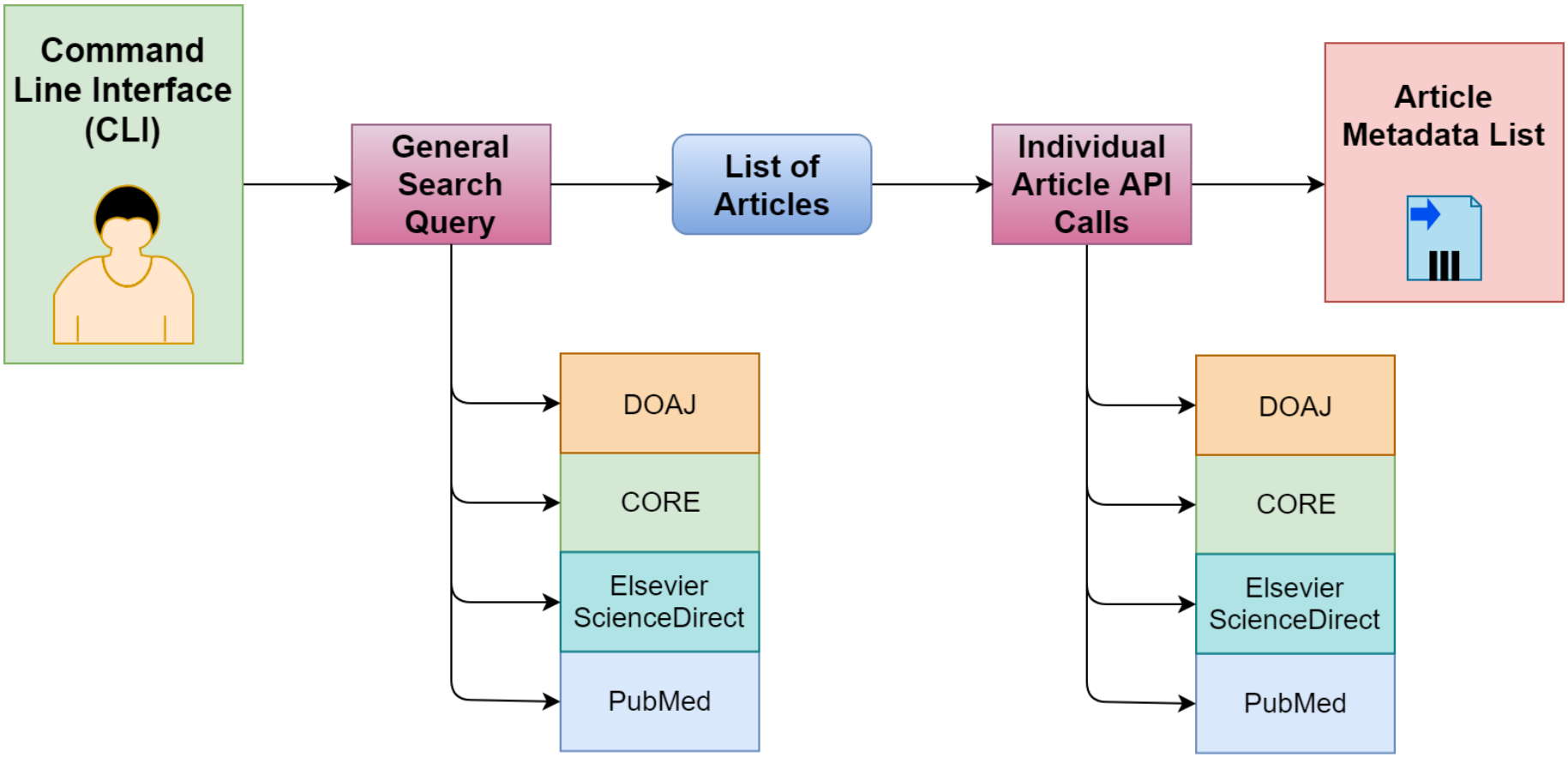
Data Flow Diagram



Web Crawler

- Based on our prior work with the previous version of the CoVaSEA web crawler developed in JavaScript (Bae et al., 2017)
- The user selects the literature database they want to search, the general search query used to search the database, and the number of articles desired
- The CoVaSEA web crawler currently supports 4 different literature databases: CORE, DOAJ, PubMed and ScienceDirect
- Utilizes the REST API of the literature database to retrieve citation metadata (authors, title, DOI, publication date, and abstract)
- First accesses the search functionality of the literature database with a general search query submitted by the user to find articles of interest
- Next retrieves the metadata for each article individually
- Then each abstract is processed by 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 summarize and describe the concepts and claims of 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)
  - Performs named entity recognition to identify proper nouns in the abstract (Recasens et al., 2013)
- Executes a constituency parse on individual sentences to create a constituency tree for 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.)

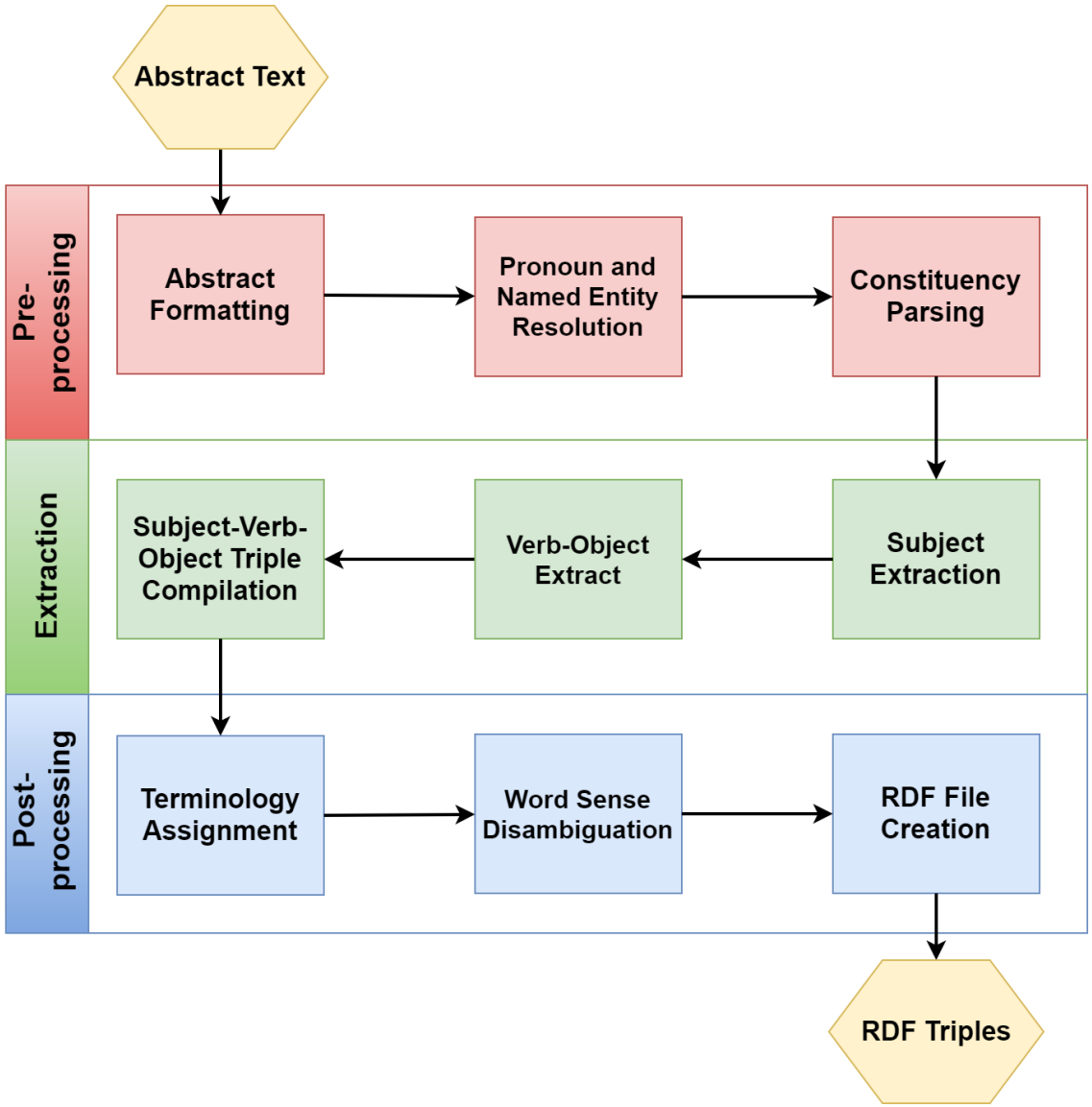
Extraction

- 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
- Breadth-first search is used to find the highest verb phrase in the tree
- 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 with 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 in the triple store (embedded in Description field of NPDS repository record)
- Examples of triple store databases (independent of NPDS): Apache Jena and Ontotext GraphDB

Natural Language Processing Pipeline



Triple Store

- To store the semantic metadata of the articles it has converted, CoVaSEA records both the citation metadata triples and the abstract representation triples
- Records can be stored in either a local triple store and/or a DOORS directory
- Each article's RDF file consists of two sections: the citation metadata and the abstract representation
- Citation metadata section stores RDF triples for the basic metadata (authors, title, journal, publication date, etc.)
- Abstract representation section stores RDF triples derived from the abstract as translated by the natural-language parser and converter (the lexical to semantic translator)

SPARQL Query Engine

- Performs a SPARQL query search of the triple store using RDFLib
- 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 the SPARQL query builder
- Design of the CoVaSEA SPARQL query builder based on the Wikidata SPARQL Query builder (Vrandečić and Krötzsch, 2014)

SPARQL Query Builder

- Resource for users who do not want to use SPARQL syntax to build a search query
- A query builder form that helps users create their own SPARQL queries
- First the user enters a series of conditions
- The type of condition can either be required 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

Results

Notes: For all tests, the general search query is "Parkinson's symptoms"

Database	# of Articles Requested	# of Articles Received	Lexical to Semantic Abstract Translation	SPARQL Search Result	Runtime
DOAJ	10 articles	10 articles	Successfully Translated: 10 # of Triples: 64	10 results 20 triples	37 seconds
DOAJ	100 articles	100 articles	Successfully Translated: 96 # of Triples: 573	100 results 200 triples	312 seconds
DOAJ	1000 articles	983 articles	Successfully Translated: 967 # of Triples: 6854	983 results 1966 triples	2589 seconds
PubMed	10 articles	10 articles	Successfully Translated: 10 # of Triples: 93	10 results 20 triples	31 seconds
PubMed	100 articles	100 articles	Successfully Translated: 99 # of Triples: 745	100 results 200 triples	283 seconds
PubMed	1000 articles	1000 articles	Successfully Translated: 984 # of Triples: 8321	1000 results 2000 triples	2391 seconds
Elsevier ScienceDirect	10 articles	10 articles	Successfully Translated: 10 # of Triples: 74	10 results 20 triples	45 seconds
Elsevier ScienceDirect	100 articles	100 articles	Successfully Translated: 96 # of Triples: 455	100 results 200 triples	332 seconds
Elsevier ScienceDirect	1000 articles	998 articles	Successfully Translated: 954 # of Triples: 7213	998 results 1996 triples	2431 seconds
CORE	10 articles	10 articles	Successfully Translated: 9 # of Triples: 50	10 results 20 triples	44 seconds
CORE	100 articles	99 articles	Successfully Translated: 94 # of Triples: 398	99 results 198 triples	390 seconds
CORE	1000 articles	973 articles	Successfully Translated: 912 # of Triples: 8434	973 results 1946 triples	2945 seconds

Discussion

- Translation success rate not 100% due to inability to parse properly those abstracts with more complicated sentence structures
- Can be ameliorated with further development of the lexical to semantic translator by increasing the number of different sentence formats it can parse correctly
- It should be noted that the number of abstract-derived semantic triples depends on the length of the abstract (thus a shorter abstract will have fewer triples and a longer abstract will have more triples)
- 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 general search.

Conclusion

- CoVaSEA: a concept-validating search engine and agent comprised of a web crawler and SPARQL query engine that converts articles from literature databases to NPDS metadata records
- Combines the capability to search externally on the open public web for research articles from literature databases that expose REST APIs and internally in NPDS repositories with SPARQL queries
- 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 triple stores
- These components have been integrated in a tool that can create semantic records of the brain literature represented in NPDS repositories

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