CMPUT 690 Winter 16 Eric Cheung
Assignment 3 – Summary Report 0362519

Solution Summary

The objective of this assignment is to utilize the provided RDF triples to answer a set of questions. The RDF file contains a set of linked entities for us to process and to create a model for storing relationships between entities. However, these triples alone do not contain enough information to answer all the proposed queries. In order to obtain more information, we need to process the short corpuses that are attached to some of the entities via the *hasDocument* property. It is possible to adjust and reuse many of the methods developed from previous assignments, such as the finite-state-machines (FSM) for matching patterns in these documents. For this assignment, the solution used Stanford NLP [2,3] for tokenization and Part-Of-Speech (POS) processing, but Named-Entity-Recognition (NER) was not used in this solution because it may provide very little additional information that can help in answering the questions. The solution also takes advantage of the fact that the corpuses were extracted from Freebase and Wikipedia. This allow us to make the assumption that the corpu is likely describing a particular subject, such as a player, a studium, or a football club as described in [1]. By combining this with the pattern matching FSM, the solution is able to extract additional information about the subjects, and their relationships with other entities (See the appendix section for illustrations of the FSMs). Next, the solution needs to disambiguate the identified, but not yet linked, entities as the *objects* of the *Subject-Property-Object(SPO)* triples. To do that, it utilizes the string distance measuring technique available in the *Second String* [4,5]. After experimenting with several of these measuring methods, the Jaro-Winkler distance measure was selected and using a score of 0.8 as a threshold as a match. The solution attempts to match the unlinked entity with the various literals associated with the *wikipedia.en* property in the RDF graph [4]. If a match is found, the URI key of that object is then stored as part of the triple extracted from the corpus. If a match could not be found in the RDF graph, then the entity is stored as a literal for future processing. All new triples are inserted in the same model as the triples extracted from the given RDF file. The resulting *fused* model is then used to answer the questions by using the Apache Jena toolset. Figure 1 depicts the framework used for processing information from the RDF triples, and then fusing them back into a single model or graph for answering queries.

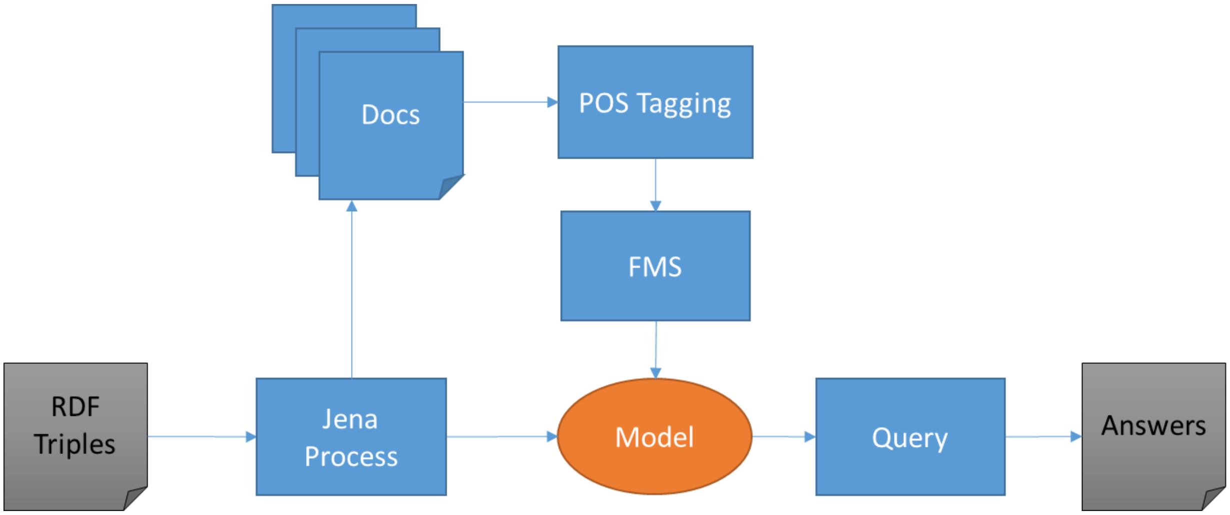


Figure 1 Framework used to extract and merge information for answering queries.

Issues

Although many relationships were successfully identified by the solutions using FSM for POS/Hearst pattern matching and String Distance for matching literals, there are still many aspects of solution could be improved.

Brittle FSM pattern matching – The solution uses FSM for pattern matching, this approach can be considered brittle because the matching conditions can be very rigid in terms finding a pattern. It also takes a lot of manual effort to maintain the design on a continuous basis.

Entity Disambiguation – The entity URI keys provided in the RDF made the task of entity linking significantly easier, especially when paired with the anchor term assumption on Wikipedia corpuses. However, the solution relies on measuring string distance for finding similar named entities in the RDF for entity linking. This method works well if the corpus share similar context, but would fall apart quickly when applied on corpus from different subject areas. Also, it is a risk of incorrect entity linking when two similarly named entities are in the model. For example, the stadium “Stadio Artemio Franchi” could easily be matched with the person “Artemio Franchi” if string distance is the only measurement used to link entities.

NER not used – The solution does not currently take advantage of the NER feature provided by the Stanford NLP package. It was not implemented mainly due to the anticipated low value for the solution with the time given. In retrospect, the NER could be used to reduce the risk of incorrect triple extraction or entity linking from occurring by checking for the correct entity class before applying a property. For example, only an entity classified as *PERSON* could be have a *nationality*.

Undeveloped Co-reference Resolution – The FSM relies heavily on the assumption that the anchor subject is used and the relationships are referring to it as the subject. If the corpus does not meet the assumption, the FSM would incorrectly link relationships with the wrong subject entity.

Areas of Improvement

There are many areas in the solution can be improved if given the opportunity.

Linking to external Knowledge Base – If the solution is able to merge with other knowledge base, then it has the potential to increase the scope of the responses. It is currently limited by the information contained in the given RDF graph and the extracted triples. Also, it is possible to verify the response with other knowledge base to increase accuracy of the responses.

Using NER – A NER classifier that had been trained with similar data set may help in identifying named entities for performing sanity checks on extracted triples.

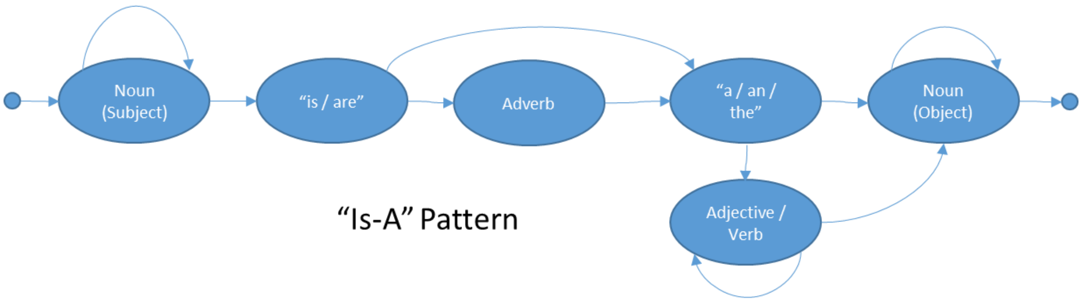
Using Treebank Parsing – In the area of improving agility of the solution, *Parsing* could be used to break the sentences into different noun phrases using treebank format, then different mentions could linked to relationships that within the phrase. This could improve co-reference resolution where the FSM method has difficult with. Using a lexical database, such as the WordNet with Hearst Pattern matching, the solution may be able to provide a more dynamic method of identifying relationships than the current FSM method.

Reference

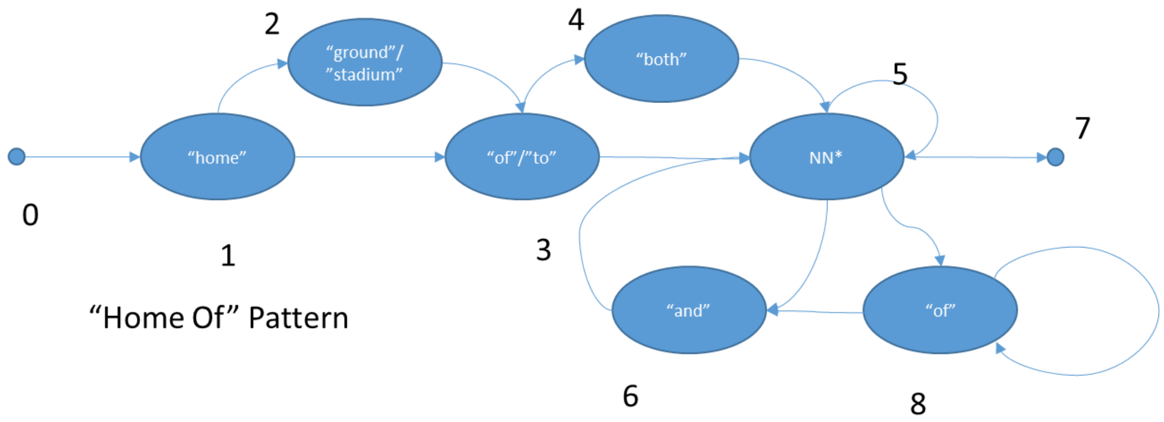
1. Extracting Semantic Concept Relations from Wikipedia. Arnold, Patrick and Rahm, Erhard. In Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS14), pp. 26:1-26:11. ACM, New York, NY, USA, 2014.
2. Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70.
3. Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.
4. Cohen, William W., Pradeep D. Ravikumar, and Stephen E. Fienberg. "A Comparison of String Distance Metrics for Name-Matching Tasks." IIWeb. Vol. 2003. 2003.
5. http://secondstring.sourceforge.net/

Appendix

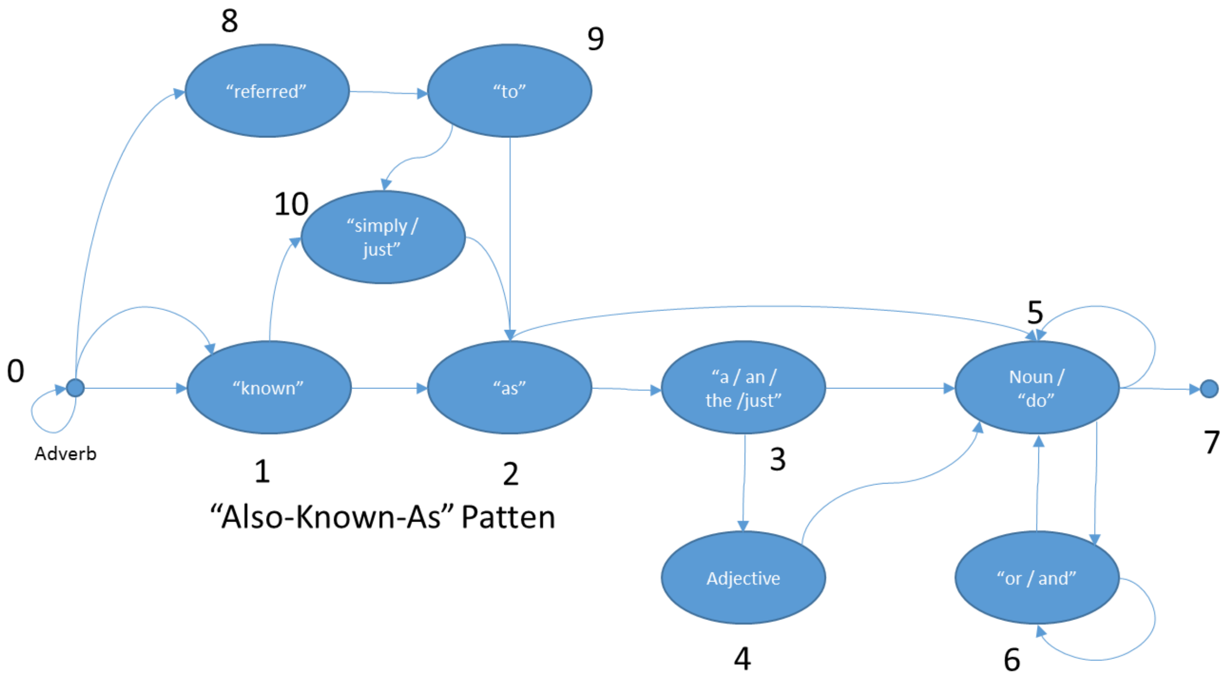
Below are examples of the Finite-State-Machines implemented in the solution.



“Is-A” Pattern

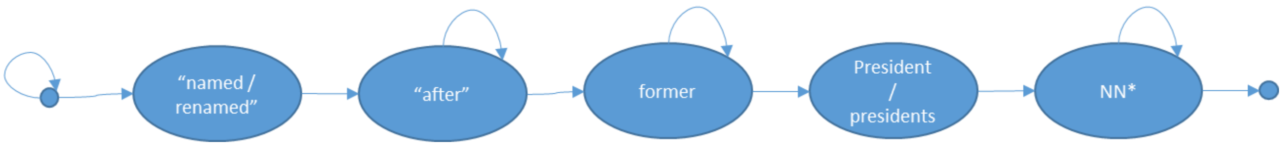


“Home Of” Pattern



“Also-Known-As” Patten

“named after president”


“named / renamed”
“after”
former
President / presidents
NN*