SPARQL Bot: a Spoken Dialog System for Knowledge Base Generation and Querying

Will Kearns
University of Washington
kearnsw@uw.edu

1 Introduction

The Resource Description Framework (RDF) was developed by the World Wide Web Consortium (W3C) as a standard to semantically link knowledge on the web. It was initially published in 1999 and has become the most widely used language for knowledge storage and interchange.

Freebase was launched in 2007 as a community constructed knowledge base. (Google, ¡year¿) This database was shutdown in 2015 in order to consolidate with the Wikidata project another community constructed knowledge base of over 26 million data items.

SPARQL Bot is a simple attempt to create, modify, and query ontologies. It uses a combination of named entity recognition, regular expression matching, and keywords.

2 Background

RDF triple stores easily allow for the addition/merging of new triples without duplication. This is beneficial when telling the system something that it already knows as part of a multipart addition.

There have been prior attempts to acquire knowledge from users using natural languag. (Pappu and Rudnicky, 2014)

3 System Commands

The system has a few functionality that will be described in this section. The system makes use of a keyword system for dialog management. These keywords are stored in text files and read in once by the system at initialization. The benefit of this approach is that these keywords can be modified at any time either as the system develops and we analyze user utterances or by the user themselves.

3.1 Adding

The system allows a user to add triples to an RDF triple store using the default keywords ”add” and ”remember”. For example:

```
User: Add John Glenn is an astronaut.
System: I heard you say that you
wanted to add <user query> to your
database, is that correct?
User: Yes, that’s correct.
System: Ok, I have added that to your
<database name> database.
```

For convenience, the system allows the user to specify multiple predicate object pairs for one subject using the ”and” conjunction. For example:

```
User: Remember John Glenn is from
Florida and is a Republican.
```

3.2 Removing

Sometimes a user realizes a mistake and needs to remove a triple from the data store. This can be done using the command ”forget”, ”remove”, or ”delete”. For example:

```
User: Forget John Glenn is a
Republican.
System: Are you sure you want to
remove John Glenn is a Republican?
This cannot be undone.
User: Yes, I’m sure.
System: Ok, I’ve removed that from
your <database name> database.
```

3.3 Querying

The system allows for writing SPARQL queries using a natural language syntax. Using question keywords. For example:

```
User: Who is an astronaut and is a
senator?
System: John Glenn is an astronaut
and is a senator.
```

To get all information about an entity the user can use the "describe" keyword. For example:
`User: Describe John Glenn.`
`System: John Glenn is an astronaut and is a senator and is from Florida.`

4 Results

Several tests were ran on the ReVerb ClueWeb extraction.(Fader et al., 2011). These were too slow running on my laptop for use in a dialog system, so a smaller sample of Wikipedia extractions was used (data available at http://reverb.cs.washington.edu/). However, in these databases the entity to predicate ratio is very low meaning that it is hard to find a use case for query that will return more than one entity. For example:
`User: Where is just east of England`
`System: The Netherlands is just east of England`

Examples like this are very prevalent and not just due to incompleteness of the sample, but also due to the lack of normalization of the predicates. That is, "is just east of" is different than "is east of". Finally, since this information comes from the web much of it is inaccurate. This defeats the purpose of this knowledge based approach and a statistical text summarization method would be more appropriate.

For these reasons, I decided to abandon those two resources and focus primarily on the creation and editing of knowledge bases by voice.

5 Limitations

This work requires a restrictive syntax and is sensitive to changes in predicate argument structure. This can often make queries awkward to speak when using a RDF triple store developed not using natural language. For example using Wikidata knowledge graph:
`User: Who educated at Yale University and position held President`

This is a severe limitation because one of the powerful aspects of SPARQL is its ability to run federated queries across multiple knowledge bases and most of these are not developed using natural language. This can lead to additional scalability issues when having to adjust syntactic rules for each knowledge base separately.

Fortunately, for most manually curated knowledge bases the number of predicates is relatively small compared to the number of entities and therefore it may be possible to match the predicate extracted from a natural language query to the list of predicates from the knowledge base. This was prohibitive in my system due to speed, but seems that it would be feasible if the database was separated over several nodes and queries could therefore be run concurrently.

Additionally, the system only handles multiple word entities recognized by the Stanford NER caseless model. Otherwise, the system only handles single word entities. Sometimes, this model can be inconsistent with

The speech recognition software used is also not sensitive to biomedical terms and therefore is unsuitable for querying biomedical ontologies.

6 Future Directions

This program would benefit from the use of a commercially available speech recognition system with support for biomedical terms. This would allow the system to leverage existing ontologies available through bioportal, e.g. the UniProt database.

Semantic Role Labeling is an essential subtask to development of this type of system as it would allow for more fine-grained argument structures and automatic classing of actors. I looked into EasySRL for this, but have yet to hear back from the authors regarding a missing jar file.(Lewis et al., 2015)

7 Dependencies

This program uses several third-party libraries and software:

- Stanford NER (caseless model)
- Natural Language Toolkit (python)
- Stardog 5 (SPARQL endpoint)
- gTTS, Google Text-To-Speech (python)
- SpeechRecognizer (python)

References

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '11)*. Edinburgh, Scotland, UK.

Google. ¡year¿. Freebase data dumps. https://developers.google.com/freebase/data.

Mike Lewis, Luheng He, and Luke Zettlemoyer. 2015. Joint a* ccg parsing and semantic role labelling. In *Empirical Methods in Natural Language Processing*.

Aasish Pappu and Alexander I. Rudnicky. 2014. Learning situated knowledge bases through dialog. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH* (September):120–124.