Design of a Scalable Natural Language Report Management System

Manu Madhavan
M.Tech Computational Linguistics
Govt. Engg. College, Palakkad
India - 678 633
mmnamboodiry@gmail.com

Robert Jesuraj K.
M.Tech Computational Linguistics
Govt. Engg. College, Palakkad
India - 678 633
robertjesuraj@gmail.com

P. C. Reghu Raj
Professor,
Dept. of Computer Sc. and Engg,
Govt. Engg. College, Palakkad
pcreghu@gmail.com

Abstract

Understanding natural language text by automated systems is becoming popular as the need for conversion of unstructured text to structured text increases. We report the design of a scalable Natural Language Report Management System in which information is collected from unstructured texts is done using statistical natural language processing tools like NLTK (Bird S, 2009), Stanford CoreNLP (Stanford CoreNLP, 2013) etc. The extracted information is then stored in a graph database to form the knowledge base. More information is added to this knowledge base using semantic web technology, DBpedia, and Geo-ontology. Reasoners like *Pellet* are used to improve the reasoning capabilities of the system. The query processing system, with the query being in natural language, will search for an absolute match in the stored knowledge base. A Natural language generation module is integrated with the system, using which the processed query result is articulated to produce answers in natural language.

1 Introduction

Natural language report management (NLRM) is a sub-problem in the area of knowledge management, which involves the systematic process of finding, selecting, organizing, distilling and presenting information coded in natural language. Knowledge Management (KM) is increasingly pervasive in industries, which marks NLRM important. The necessity for intelligent automatic report management arises mainly because of the following two circumstances:

- It is cheaper to teach the machine,
- In many cases, to make a well informed decision or to find relevant information, one needs to read, understand, and take into consideration a quantity of text thousands of times larger than what one person will physically be able to read in his lifetime.

Understanding natural language texts involves extracting information. The process of information extraction requires identification of entities and their relationships. After information extraction, next step is the storage of this information in the knowledge base which is in machine understandable standard (JSON/ XML). Ontology, the backbone of semantic web technology, is the most popular standard for representing the domain knowledge (Liyang Lu, 2013). Use of ontologies for knowledge management permits sharing of common knowledge and conceptualization of the domain.

NLRM has many applications, including business intelligence, resume harvesting, media analysis, sentiment detection, patent search, and email scanning. A particularly important area of current research involves the attempt to extract structured data out of electronically available scientific literature, especially in the domain of biology and medicine (Siddhartha, 2005).

This paper discusses the approaches, challenges, and solutions in designing a natural language report management system. The proposed method makes use of the semantic web vision of the World Wide Web, which relates web pages based on conceptual relationss. The organization of knowledge in the form of ontology will help achieve the de-facto standard architecture in semantic web. Scalability issue in knowledge base

storage is handled by using graph database and ontology based Natural Language Generation (NLG) techniques that allow good interfaces. The paper also explains the integration of different open source NLP tools for the implementation of report management system.

An example: The need for natural language report management can be explained by the following example from Wikipedia article about Sachin Tendulkar:
Tendulkar was born at Nirmal Nursing Home on 24 April 1973. His father Ramesh Tendulkar was a reputed Marathi novelist and his mother Rajni worked in the insurance industry. On 14 November 1987, Tendulkar was selected to represent Mumbai in the Ranji Trophy. A year later, on 11 December 1988, aged just 15 years and 232 days, Tendulkar made his debut for Mumbai against Gujarat at home and scored 100 not out in that match making him the youngest Indian to score a century on first-class debut.

An individual interested in cricket can understand all the details given in this passage. But, designing an intelligent system to do this is a challenge. The semantics of each concept in the domain has to be added to the system. This requires the identification of entities like *person*, *organization*, *date*, and *place* from the raw text. This extracted information will be represented in a structured manner, which can be further accessed through NL queries like "where was Tendulkar born?". The complex, unstructured nature of the input text makes the information extraction challenging.

The remaining part of the paper is organized as follows: Sec.2 discusses some related work in this area. Sec.3 defines the problem and the proposed solution strategy. This section also gives the details of implementation of the proposed system. Finally we enlist the steps involved in making a natural language interface to database in Sec.4.

2 Related Work

An approach to populate an existing ontology with instance information present in the natural language text input has been explained in (Anantharangachar, 2013). This approach starts with a list of relevant domain ontologies created by human experts, and then describes the techniques for identifying the most appropriate ontology to be extended with information from the given text. The authors also demonstrate how to extract information from the unstructured text and add it to the selected ontology as structured information. This identification of the relevant ontology is critical, as it is used in identifying relevant information in the text. The authors extract information in the form of semantic triples from the text, guided by the concepts in the ontology, and convert the extracted information into Resource Description Framework (RDF) to be appended to the existing domain ontology. This approach have achieved 95% accuracy of information extraction (Anantharangachar, 2013).

The system called RitaWN presented a new representation for Wordnet ontology using Neo4j graph storage. This work analysed that the graph databases yield much better results than traditional relational databases in terms of response time even under extreme workloads, thus demonstrating their promised scalability (Kaled Nagi, 2013).

The details of designing domain specific ontology is explained in (Noy N, 2005). The various approaches for ontology guided question answering is discussed by Gyawali (Gyawali B, 2011). This also describe a generalized architecture for the same by identifying a set of factoid questions that can be asked using a domain ontology.

IBM designed IBM Watson (IBM Watson, 2011), which understands the medical records well. Watson uses IBM Content Analytics to perform critical NLP functions. Unstructured Information Management Architecture (UIMA) is an open framework for processing text and building analytic solutions.

The ONTOSUM system uses Natural Language Generation (NLG) techniques to produce textual summaries from Semantic Web ontologies (Kalina, 2005). This work shows how the existing NLG tools can be adapted to Semantic Web ontologies, in a way which minimizes the customization effort while offering more diverse output than template-based ontology verbalizers.

3 System Design

The overall design of the system can be briefly explained as follows. The system will take natural language text reports as input. The input is processed by the IE module. Domain specific entities, relations and other information is extracted. This information is represented in some machine readable structures (like JSON/ XML) and passed

on to populate the ontology instance module. Ontology is the conceptual representation of the domain. The extracted information is compared with the base ontology and matching instances are populated to the ontology. The knowledge base (ontology) gets updated with the new instances obtained from the extracted information. The next stage is developing an interface to the knowledge base. The method discussed here is a natural language interface, which make use of Natural language generation (NLG) techniques. The user asks natural language queries, the search engine retrieves the answer found in the knowledge base and responds back in natural text. NLG module converts the retrieved answer to meaningful sentence and displays to the user. The system architecture is shown in Fig. 1.

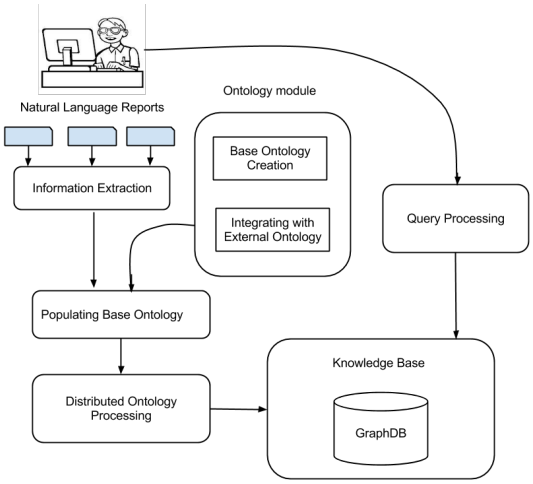

Figure 1: System Architecture

3.1 Creating the base ontology

The first task in any knowledge management system is to design the knowledge base. In this work, the knowledge base is the ontology, which is a graphical representation of concepts and their relationships. The base ontology in Semantic web terminology will serve the functionality of Entity Relation diagram in the object data model . There is no standard methodology for designing an ontology(Liyang Lu, 2013). The important thing in ontology design is identification of domain concepts and entities. We are treating the document as a description of domain events (Noy N, 2005). In most cases, the verbs serve as the events. Then the other entities related to the verbs act as the participants of the event. A Paninian (karaka based) grammar based analysis of the text will help to identify the thematic roles of entities in the event (Bharati et al, 1995). This will create a frame for each event in the domain, with slots for entities. Then the information extraction will populate the slots, with individuals.

For example, in the report given in Section 1, birth, debut, and match are three events. The birth event has, person, date and location as entities and are related to the event by relations hasPerson, hasDateOfBirth and hasLocation respectively. The ontology is represented in web ontology language (OWL)[1], a world wide web consortium (W3C) standard for semantic web representation. In OWL, the events and entities are represented as classes and the relations are called properties. OWL provides many features to add semantics to the ontology. For example, a property can be defined as the inverse of another property. In our running example, we can define a property *hasPerson*, which relates an event and person. Then a property *hasEvent* is defined as inverse of hasPerson, which relates the person to event class. The classes and properties can be arranged in hierarchy, by adding sub-class, super-class relations. When a reasoner is enabled, it can infer the implicit relations existing between the entities. A sample base ontology structure is shown in Fig. 2.

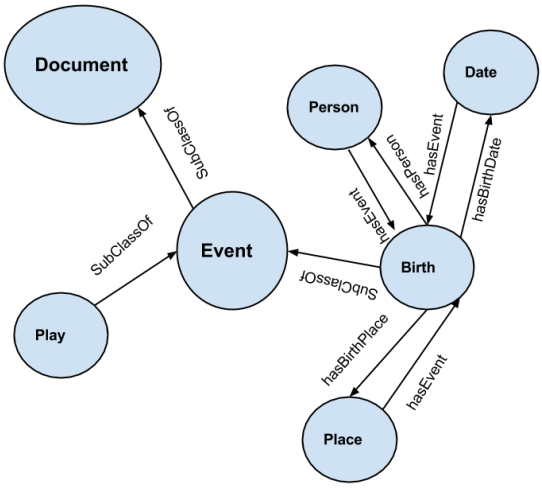

Figure 2: Base ontology structure

Another technique in ontology design involves reusing the existing general ontologies (Sofia H, 2000). For example, if we have an entity like *location* in our ontology, we can inherit the properties of location specified in another ontology.

[1]http://w3c.org/TR/

GeoOntology gives latitude-longitude information ((GeoOntology, 2013)). DBPedia gives relations between person, place and organization which uses Wikipedia database(DBPedia, 2013). FOAF (friend of a friend relation between person) is another ontology available as open source. All these can be used to enhance the knowledge base. The use of such general purpose ontologies is purely the choice of the designer. The base ontology is implemented using Jena API (Jena, 2013).

3.2 Information Extraction (IE)

The information extraction is the process of identifying named entities and relations from the natural language text (Bird S, 2009). The IE module analyzes the input text for the entities and relations specified in base ontology. There are 3 main approaches viz. Statistical based, rule based and hybrid approach. Hybrid approach is used in the current research, each has its own merits and demerits. The process of extracting information from an unstructured text involves pipelines of activities shown in Fig.3 (Bird S, 2009).

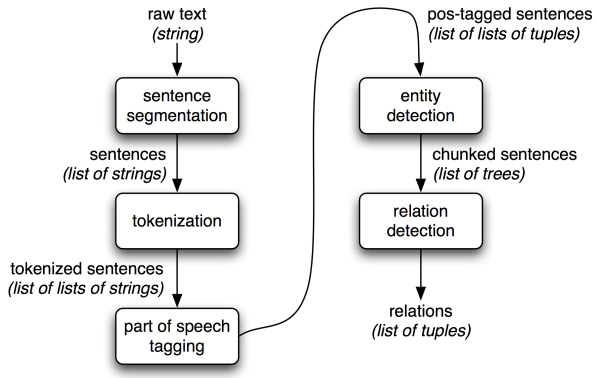

Figure 3: Steps in IE

The process starts with the segmentation of the raw text into sentences using a sentence segmenter. Each sentence is further subdivided into tokens using a tokenizer. The tokens are tagged with POS Tags, which is essential to identify the named entities. The basic technique used for NER is chunking. Chunking is the process of segmenting and labeling multi token sequence into an entity. The low level chunking uses the token and POS information. The high level chunking uses low level chunks along with the other informations like POS. The chunk level parsing also extracts the named entities and the relation existing with the entities. The IE system may also include a dictionary lookup, to identify the domain specific entities such as place, person and organization names. These process could be done using Stanford CoreNLP parser (Stanford CoreNLP, 2013) an efficient statistical NLP tool, to identify the sentence dependency and co-reference resolution.

A chunk can be specified using different means. The most simple method is specification by regular expression. A regular expression defines a pattern and acts as a template for a chunk entity. By executing such expressions, the chunks can be identified and then labeled with special tags. The Common pattern specification language (CPSL) based grammar rules are widely applied in domain specific IE applications (Douglas, 1998). If the domain of application is general and availability of training sample is large, the most flexible method for NER is statistical based.

From the implicit context, entities could be identified using rules. For example, "John went to New Delhi on his birthday". Here, no words explicitly represent the date of event. But, it is implicitly mentioned by birthday. Hence, identifying the context of such words and applying inferences from knowledge base will improve the intelligent behavior of the system. The extracted information is represented in a JSON[2] structure. JSON is a lightweight, structured representation standard, widely used in current web applications.

3.3 Populating the Ontology

When the information extraction is completed, the next step is storing the information into knowledge base. The information is added as an instance of an ontology classes. Jena provides methods to add individuals(entities) to ontology. The labels given to the chunks will help to identify the base class and the value is added as the individual to that class (Jena, 2013). For example, consider the first line in our running example, "Tendulkar was born at Nirmal Nursing Home on 24 April 1973". IE step will identify this as a birth event, with Nirmal Nursing Home as location and 24 April 1973 as date. Then class *birth* is selected from the ontology and the properties *hasPerson*, *hasLocaion* and *hasBirthDate* will be populated with Tendulkar, Nirmal Nursing Home, 24 April 1973 respectively.

The resulting triples after inserting into the base ontology will look like:

<Tendulkar_was_born_at_Nirmal_Nursing_

[2]JavaScript Object Notation

```
Home_on_24_April_1973 hasBirthDate
24_April_1973>
<Tendulkar_was_born_at_Nirmal_Nursing_
Home_on_24_April_1973 hasPerson
Tendulkar>
<Tendulkar_was_born_at_Nirmal_Nursing_
Home_on_24_April_1973 hasLocation
Nirmal_Nursing_Home>
```

The challenge in the ontology based data model is to provide a scalable and efficient persistent storage scheme. The simplest way to store RDF triples comprises of a relation/table consisting of three columns (one each for subjects, predicates and objects). However, this approach suffers from the lack of scalability and degraded query performance, as the single table becomes long and narrow with increase in the number of RDF triples. The recent interest in using the NoSQL/graph database is a positive move regarding this issue. In our work, Neo4j (Neo4j, 2007), a popular open source NoSQL graph database is used for scalable persistent ontology storage. Neo4j has got inbuilt graph algorithms for finding shortest paths, all paths, Dijkstra and A$*$. Friend of a Friend (FOAF) relation can also be implemented using Neo4j. Since social networks can be easily modeled as large graphs of interconnected users, these tools can be very useful application for graph databases (Ian R, 2013). Pellet reasoners could be used to enhance the power of reasoning in the knowledge base (Pellet Tutorial, 2013). The reasoner will help to generate more triples that are valid, generated by the triples that are previously stored in the knowledge base.

4 Interface to the Knowledge Base

The final stage in the system is designing an interface to the knowledge base. Even though the representation in OWL makes the knowledge machine readable, users may find it difficult to understand the imparted knowledge. Asking questions to databases in natural language is a convenient method of data access, specially for casual users who do not understand complex database query language such as SQL. Natural Language (NL) based interface is attempted in this work, which use query in NL and generate the result in NL. This requires the intelligence in both NL understanding and NL Generation. The NLG is used here in two applications. In the former case, it will convert the NL query to a SPARQL query and represents the result in NL sentence. The latter application uses NLG to generate natural language description of the OWL statements. NaturalOWL (Dimitris, 2000) tool is used for this purpose.

In this work, a basic template matching method is used for NLG. In order to answer the NL query, the system performs IE steps on the query sentence and identifies the concepts, and question type. Based on these results, a SPARQL query is executed. The result of SPARQL is then used to fill the answer template. For example, consider the question "Where was Tendulkar born?". The query analysis will identify that the question contains entities like Birth, Tendulkar and since the question term is *where*, it expects a *location* as the answer. So the corresponding SPARQL query will be:

```
SELECT ?loc
    WHERE {
    ?birth hasPerson Tendulakr .
    ?birth hasLocation ?loc
    }
```

The query will return Nirmal Nursing Home as result. Thus the input the NLG system, can be represented as:

```
[Tendulkar:  Subject,
Nirmal Nursing Home:  Location
Birth:  Event]
```

Then this structure is matched with the template to generate the output sentence:

```
subject was event at location .
Tendulkar was born at
Niraml Nursing Home .
```

The detailed explanations of NLG can be obtained from (Batesman, 1991) and is beyond the scope of this paper. The idea of using natural language query instead of SQL has prompted the development of new type of processing method called Natural Language Interface to Database systems (NLIDB) (Ana-Maria et al, 2003). NLIDB is a step towards the development of intelligent database systems (IDBS) to help the users to perform flexible querying on databases.

5 Experimental Setup and Results

For experimental purpose, a system with following configuration was chosen: Intel i5 processor, 4 GB RAM, Ubuntu 12.04 Operating system. The softwares that are required for this include, python 2.7, with following supporting modules: nltk, rd-

flib, SPARQLWrapper. SPARQLWrapper helps to connect with DBPedia.

Let us see the obtained results, suppose the user queries using natural language as,
Who is Sachin Tendulkar?
Answer: *Indian Cricketer*
The above result was obtained finding a triplet match using SPARQL query as:
Select ?thing
where {
<http://dbpedia.org/page/Sachin_Tendulkar, dbpprop:shortDescription, ?thing>
}
After processing through the NLG module, the result is, *Sachin Tendulkar is an Indian Cricketer.*

6 Conclusion

This paper outlines the design of an end to end system for natural language report management. The proposed system uses the semantic web technologies for handling the extracted information from natural language text. An ontology is used to represent the conceptual knowledge. The information extraction module uses a hybrid approach combining the benefits of both statistical and rule based systems. The use of Neo4j graph database, instead of the relational one, for persistent storage makes the system scalable. The NL based interface combined with NLG from ontology represents a new approach to automated report management. The proposed system can be extended by the integration of more linguistic techniques like systemic functional grammar, discourse structures, etc.

References

Akshar Bharati. Vineet Chaitanya. Rajeev Sangal., 1995. *Natural Language Processing: A Paninian-Perspective* Prentice-Hall of India, New Delhi.

Ana-Maria Popescu , Oren Etzioni , Henry Kautz, 2003. Towards a Theory of Natural Language Interfaces to Databases. *IUI '03* , Miami, Florida USA.

Anantharangachar R, Ramani S, Rajagopalan S, 2013. Ontology Guided Information Extraction from Unstructured Text,

Apache Jena Project. *http://jena.apache.org/*, Visited on July 2013.

Batesman J A, 1991. The Theoretical Status of Ontologies in Natural Language Processing *in Proceedings of the workshop on Text Representation and Domain Modeling–Ideas from Linguistics and AI,* Technical University Berlin.

DBPedia, 2013. *http://dbpedia.org/About*, Visited on July 2013.

Dimitris G, George K and Ion A 2000. How to install and use Natural OWL, *Athens University of Economics and Business,* Greece.

Douglas E A, 1998. The Common Pattern Specification Language. *Artificial Intelligence Centre,* Meno Park, CA.

GeoNames Ontology, 2013. *http://www.geonames.org/ontology/documentation.html*, Visited on July 2013.

Gyawali B. 2011. *Answering Factoid Questions via Ontologies : A Natural Language Generation Approach*, M.Sc. Dissertation, Dept of Intelligent Computer Systems, University of Malta.

Ian R, Jim W, 2013. *Graph Databases.* First Edition, O'Reilly Media.

IBM Watson Engagement Advisor *http://www-03.ibm.com/innovation/us/watson*, Visited on July 2013.

Kalina Bontcheva 2005. Generating Tailored Textual Summaries from Ontologies, in ESWC, pp. 531–545.

Khaled Nagi 2013. A New Representation of WordNet using Graph Databases *in The Fifth International Conference on Advances in Databases, Knowledge, and Data Applications, Spain,.*

Liyang Lu, 2007. *Introduction to Semantic Web and Semantic Web Technologies.* Champman and Hall/CRC.

Neo4j Project. *www.neo4j.org*, Visited on July 2013.

Noy N. F and McGuinness L. D, 2005. Ontology Development 101: A Guide to Creating Your First Ontology, *Stanford University*, Stanford.

Pellet Tutorial, 2013. *http://clarkparsia.com/pellet/tutorial/*, Visited on July 2013.

Siddhartha B, Pallab D, Dipti D, 2005. A System Architecture for Reasoning and Summarizing hybrid text inputs using Semantic Web, *International Journal of Lateral Computing,* vol.2, No.1.

Sofia H , Martins J P, Reusing Ontologies, In AAAI 2000 Spring Symposium on Bringing Knowledge to Business Processes.

Stanford CoreNLP, 2013. *http://nlp.stanford.edu/software/corenlp.shtml*, Visited on July 2013.

Steven Bird, Klein E, and Loper E, *Natural Language Processing with Python,* O'Reilly Media Inc.