Wikilogy: Representing a Wikipedia entity as an Ontology

Kanchan Arora
M.Tech, Indraprastha Institute of Information Technology, Delhi

Abstract – In field of knowledge management and engineering, ontologies play a vital role. A human or machine intending to get information about a particular entity and its related entities through Wikipedia is required to visit and read through some Wikipedia pages to extract the required information. In this paper, a framework is provided using which ontology can be created for each Wikipedia page by utilizing knowledge from YAGO and DBpedia. This ontology contains all the information provided in YAGO and DBpedia about an entity and also infers some new facts based on the rules provided. For getting these facts otherwise, a human or a program may be required to parse many related Wikipedia pages. Using this representation of Wikipedia entity, unexpected relationships between entities can also be found.

Keywords: Wikipedia, Ontology, YAGO, DBpedia.

I. INTRODUCTION

In computer science, ontology is an explicit and formal specification of conceptualization of a particular domain [13]. It includes machine-understandable descriptions of concepts in the domain and relations among them. To analyse the domain knowledge so that new information can be gathered is one of the most important goals of creating ontologies. Ontologies are useful in application areas such as semantic web, software engineering, information extraction, natural language processing and many other related areas. Their use is still not extensive as manual construction of ontologies is both time-consuming and labor-demanding. In this paper a semi-automatic framework is provided to create ontology for a Wikipedia entity by using the knowledge in YAGO [3] and DBpedia [2]. The organization of this paper is as follows:

-In section II, essentials of ontologies in general and terminologies with respect to the semi-automatic ontology construction setting of this paper are briefly described.

-In section III, the proposed method for ontology construction is described,

-In section IV, analysis for the proposed methodology with an example is provided.

-In section V, paper is concluded by discussing the advantages and disadvantages of the given approach.

II. TERMINOLOGIES

This section describes the integrals of ontology in general and some entities with respect to ontology construction environment of this paper. An ontology consists of collection of terms and relationships between these terms. These terms are nothing but *concepts/classes* that describe the domain of ontology. For example in an ontology for poetry setting, writers, poets, Indian poets and English language poets are some concepts. Most of the typical ontologies are hierarchical in nature. The *relationships/predicates* describe the hierarchy of classes or concepts. In a hierarchy a class X is a subclass of another class Y if all the objects of X are objects of Y. Fig 2.1 describes the hierarchy of an ontology in poetry domain. In fig 2.1 "Abhay Kumar" is the *instance of* class "English language Poets" and "English language writers". Along with the subclass relationship, ontology can also contain other *properties/predicates* like writer *writes* articles. Properties, in turn can also have subsumption hierarchy [13].

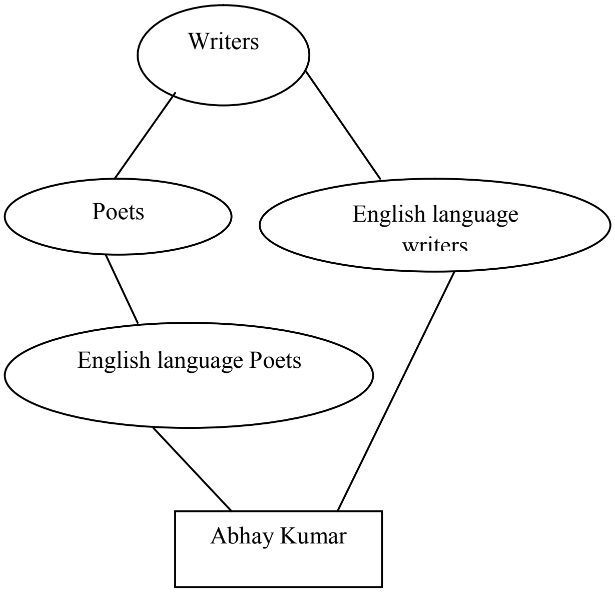

Fig. 2.1 Example of a hierarchy

In order to model such hierarchies in the real world concepts, there is a need of language in which ontologies can be expressed. *RDF* (Resource Description Framework) [1] is the most important ontology language which models data for concepts and relations between them, thereby providing semantics to the data. *RDF Schema* is a description language for describing properties and classes of RDF resources, with semantics for generalization hierarchies of such properties and classes [13]. *OWL* (Web Ontology Language)[15] is richer than RDF in terms of describing properties, classes, relations between classes and other constraints.

Knowledge in RDF is expressed as a list of *Statements/Triples*. Each triple represents a statement of a relationship between the subject and the object that it links [9]. Triple has three parts a subject, an object, and a predicate that denotes a relationship. RDF representation can be a N3 representation or turtle representation (.ttl)[14]. In N3 representation facts are represented as :{subject,predicate,object} while syntax for turtle representation is: <subject><predicate><object>. YAGO makes facts available in both the formats.

In the past decade, many large and cross-domain ontologies have been created. YAGO [3] and DBpedia [2] are the two most widely used Wikipedia based ontologies. YAGO [3] is a huge semantic knowledge-base, derived from Wikipedia, WordNet[8] and GeoNames. It describes more than 10 million entities (like persons, organizations, cities, etc.) with more than 120 million facts about these entities. YAGO2[4] is an extension of the YAGO knowledge base, in which entities, facts, and events are anchored in both time and space. It contains 447 million facts about 9.8 million entities. YAGO3[5] is also an extension of the YAGO knowledge base that combines the knowledge from the Wikipedias in multiple languages. It contains 1 million new entities and 7 million new facts.

DBpedia extracts structured information from the Wikipedia's info-boxes which use a template mechanism, images depicting the article's topic, categorization of the article, links to external web-pages, intra-wiki links to other articles, inter-language links to articles about the same topic in different languages [2]. The DBpedia project uses RDF to represent the extracted information and consists of 3 billion RDF triples, 0.580 million extracted from the English edition of Wikipedia and 2.46 billion from other language editions. The ontology currently covers 685 classes which form a subsumption hierarchy and are described by 2,795 different properties. It uses the SPARQL query language to query this data. SPARQL stands for "SPARQL Protocol and RDF Query Language". SPARQL [6] is based on matching graph patterns against RDF graphs. Using either of the subject, object, or predicate the rest of two can be queried about by using SPARQL.

III. PROPOSED METHOD

In this section a semi-automatic framework for creating ontology for a Wikipedia entity is proposed. Following steps are followed to create the desired ontology:

1) Collection of Facts: A Wikipedia document can be represented as a conjunction of facts. Facts are available in the RDF (Resource Description Framework) format provided by YAGO and DBpedia knowledge base.

a) Collection of Facts from YAGO:

a1) YAGO has provided SPARQL query endpoint [17] to query its database. This query interface is used to get the facts about the article. SPARQL query formed as the title of the article as subject and predicate and object are queried.

a2) Average number of facts for each article obtained from stepa1 is low. To make the ontology enriched with facts, the facts related to the article are also added. To add facts following steps are followed:

a2.1) Text is extracted from a Wikipedia document using Jsoup API [12]. This text is then passed to Stanford Entity Tagger and returned entities are stored in Set. Stanford Entity Tagger [11] labels sequences of words in a text which are the names of things, such as person and company names etc.

a2.2) From each fact obtained in step a1, subjects are looked in the YagoFacts.ttl(turtle file storing all the facts of YAGO database). Facts containing the subjects are returned. Objects are also looked in YAGOFacts.ttl and facts containing the objects are returned.

a2.3) For the facts obtained after step a2.2, objects and subjects are checked in set formed in step a2.1. If both the object and subject is found in set then that particular fact is kept as shown in fig. 3.1

Step a2.2) and a2.3) are iterated till no new fact is added.

b) Collection of Facts from DBpedia: DBpedia has provided SPARQL query end-point (http://dbpedia.org/sparql) to query its database. DBpedia provides substantial number of facts. Facts obtained from YAGO and DBpedia are kept in separate turtle files so that ontology knowledge can be added in the next step.

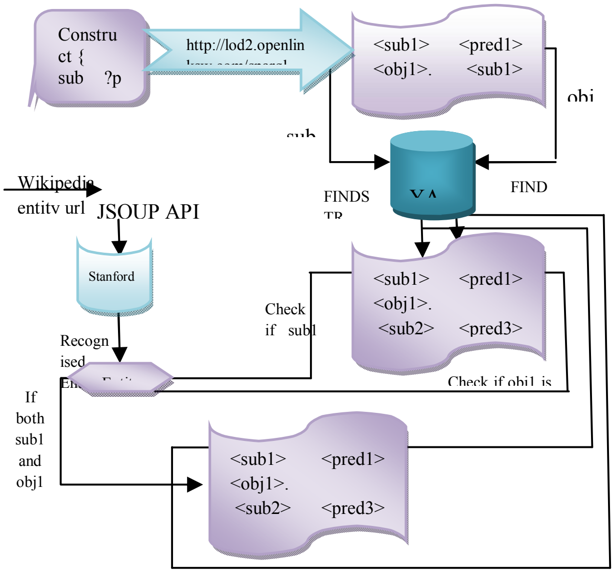

Fig. 3.1 Collection of facts from YAGO

2) Addition of Ontology knowledge

a) Ontology knowledge contains the knowledge of the classes to which instances in the facts belong to and the class hierarchy associated with those classes.

b) The knowledge about the domain and range of predicates in the facts are also added. Class to which the instances belong to are added using the following SPARQL query:

```
CONSTRUCT {

 :instance <rdf:type> ?o.

} WHERE

{ :instance rdf:type ?o . }
```

Fig. 3.2 SPARQL query to find classes of entities

Hierarchy of the classes are captured using the query:

```
CONSTRUCT {

 ?c1 rdfs:subClassOf ?c2.

} WHERE

{ :instance rdf:type ?c1 . ?c1 rdfs:subClassOf ?c2. }
```

Fig. 3.3 SPARQL query to capture class hierarchy

Domain and range knowledge of properties is added using the query:

```
CONSTRUCT {

:prop <rdfs:domain> ?o.

} WHERE { :prop rdfs:domain ?o . }
```

Fig. 3.4 SPARQL to find domain and range of properties

Ontology knowledge is added in both the files. After this step, two ontology files are formed, one file containing the YAGO facts about the article and background YAGO ontology knowledge and the other containing the DBpedia facts and DBpedia background knowledge of classes and properties. Next step is to merge these two ontologies into a consistent ontology.

3) Alignment of two ontologies: An alignment (or matching)[10] between two ontologies can be expressed in RDF as a set of rdfs:subClassOf, rdfs:subPropertyOf, and owl:sameAs statements between resources of the two ontologies. The restriction of the alignment to each of these types of statements is respectively the class alignment, the relation alignment and the entity alignment.

YAGO provides knowledge of equivalent classes in DBpedia and YAGO (yagodbpediaClasses.ttl). It also provides information about entities which are common in DBpedia and YAGO (yagodbpediainstances.ttl). PARIS [7] project provides information about subsuming relations of YAGO and DBpedia(ydpoperties.ttl). Information in these three files is used to add alignment information in the final ontology which contains the facts of both the files generated in previous step along with the alignment information.

In the file generated after this step contains:

a) Facts from YAGO for a Wikipedia document.

b) Facts from DBpedia for a document.

c) Class hierarchy and properties in YAGO ontology. (rdf:type,rdfs:subClassOf,rdfs:domain,rdf:range).

d) Class hierarchy and properties in DBpedia ontology. (rdf:type,rdfs:subClassOf,rdfs:domain,rdf:range)

e) Alignment information(owl:equivalentClass, owl:sameAs, rdfs:subPropertyOf)

All this information forms a complete and consistent ontology. Step by step analysis of the above procedure is provided in the next section.

IV. ANALYSIS AND FACTS DEDUCTION

To analyse the procedure, the entities from Wikipedia category "English_language_poets_from_India" were used. After extracting facts (for entity Abhay_Kumar) from YAGO and DBpedia file shown in fig 4.1 was created:

```
<Abhay_Kumar><hasWebsite><http://gov2.in/speakers/ab
hay-k> .

<Abhay_Kumar><hasGender><male> .

<Abhay_Kumar><wasBornIn><Nalanda> .

<Abhay_Kumar><isCitizenOf><India> .

<Abhay_Kumar><livesIn><New_Delhi> .

<Abhay_Kumar><graduatedFrom><Jawaharlal_Nehru_U
niversity> .

<Abhay_Kumar><wasBornOnDate>"1980-##-
##"^^xsd:date.

<Abhay_Kumar> <source> "AbhayKumar \u00ABThe
Soul Song\u00BB"@en .

<Abhay_Kumar> <quote> "I was always here,\nAsthe
blowing wind\nOrthe falling leaves,\nAsthe shining
sun\nOrthe flowing streams,\nAsthe chirping birds\nOrthe
blooming buds,\nAsthe blue sky\nOrthe empty
space,\nIwas never born\nIdid not die\u2026"@en .

<Abhay_Kumar> <wikiPageUsesTemplate>
<Template:Infobox_person> .

<Abhay_Kumar> <website>
<http://gov2.in/speakers/abhay-k> .

<Abhay_Kumar> <knownFor> "Introduction and
exceptional use of Social Media in the Public Diplomacy
Division of the Ministry of External Affairs India, writings
and art works on Planetary Consciousness"@en .

<Abhay_Kumar> <residence> <New_Delhi> .
```

Fig. 4.1 Example turtle file containing facts

After attaching ontology knowledge Abhay Kumar's document contains information of class hierarchies and facts collected from YAGO and DBpedia as shown in fig 4.2:

```
<Abhay_Kumar><hasGender><male> .

<Abhay_Kumar><wasBornIn><Nalanda> .

<Abhay_Kumar><isCitizenOf><India> .

<Abhay_Kumar><livesIn><New_Delhi>

<Abhay_Kumar> <type>
<wikicategory_Indian_civil_servants>.

<wikicategory_Indian_civil_servants> <subClassOf>
<wordnet_civil_servant_109925459>.

<wordnet_civil_servant_109925459> <subClassOf>
<wordnet_official_110372373>.

<wordnet_official_110372373> <subClassOf>
<wordnet_skilled_worker_110605985>.

<wordnet_skilled_worker_110605985> <subClassOf>
<wordnet_worker_109632518>.

<wordnet_worker_109632518> <subClassOf>
<wordnet_person_100007846>.

<wordnet_person_100007846> <subClassOf>
```

```
<wordnet_causal_agent_100007347>.

<wordnet_causal_agent_100007347> <subClassOf>
<wordnet_physical_entity_100001930>.

<wordnet_physical_entity_100001930> <subClassOf>
<Thing>.

<New_Delhi> rdf:type <wikicategory_Capitals_in_Asia>.

<Nalanda>rdf:type
<wikicategory_Places_connected_with_Jainism>.

<isCitizenOf> rdfs:range <wordnet_country_108544813>.
```

Fig. 4.2 Example file containing facts and ontology knowledge

After the ontology knowledge has been added in file, it is loaded into Protégé [16] and following ontology graph was generated:

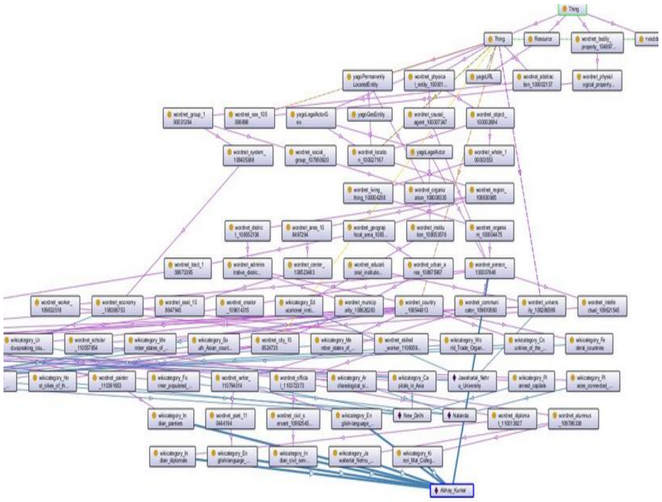

Fig.4.3 Ontology graph

Interesting patterns occurring in facts are found. New facts can be deduced by defining some DL/SWRL rules. For instance, if rdf triple like :<Abhay_Kumar> <type> <wikicategory_Kirori_Mal_College_Alumni> exists then his implies he must have done some form of education from Kirori_Mal_College i.e. new fact <Abhay_Kumar> <studiedAt> < Kirori_Mal_College> can be deduced by defining rule like:

type(X,Y)^subclassOf(Y,wordnet_alumnus)^Z(Kirori_Mal_College) -> studiedAt(X,Z)

V. CONCLUSION

In this research work it is found that representing each Wikipedia entity as a complete ontology may lead to deduction of new facts which are not present in already existing ontologies with the help of inference rules based on some common pattern. It may also help the humans and machines to find out the knowledge about an entity which is scattered in many Wikipedia pages through a single ontology which is created for that particular entity. The idea also has a disadvantage that as the common ontology hierarchy is stored for each entity which is leading to redundancy and requires more storage.

REFERENCES

[1] R. Cyganik, D. Wood and M. Lanthaler (2014), RDF 1.1 Concepts and Abstract syntax.

[2] S. Auer, C. Bizer, G. Kobilarov ,J. Lehmann, R . Cyganiak, and Z. Ives(2007), DBpedia: A nucleus for a Web of Open data. In Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference, ISWC 2007/ASWC 2007,pages 722-735, Springer.

[3] F. M. Suchanek, G. Kasneci, and G. Weikum,(2008), YAGO -A Large Ontology from Wikipedia and WordNet. J. Of Web Semantics, 6(3), September .

[4] J. Hoffart, Fabian M. Suchanek, K. Berberich, G. Weikum,(2013) YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. Artif. Intell., 194.

[5] F. Mahdisoltani,J. Biega,Fabian M. Suchanek,(2015), YAGO3: A Knowledge Base from Multilingual Wikipedias,7th Biennial Conference on Innovative Data Systems Research(CIDR 2015), Asilomar, California, USA.

[6] B. Quilitz and U. Leser,(2008), Querying Distributed RDF Data Sources with SPARQL, In proceedings of the 5th European Semantic Web Conference on The Semantic Web:Research and Applications,ESWC'08,pages 524-538.

[7] http://webdam.inria.fr/paris/

[8] C. Fellbaum,(1998) WordNet- An Electronic Lexical Database, MIT Press.

[9] https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[10] Doan A, Madhavan J, Domingos P, Halevy A (2003) Ontology matching: a machine learning approach. In: Staab S, Studer R (eds) Handbook on ontologies in information systems. Springer, Berlin Heidelberg New York

[11] J. R. Finkel and C. D. Manning. 2009. Joint parsing and named entity recognition. In Proceedings of NAACL , pages 326–334.

[12] http://jsoup.org/

[13] https://www.dcc.fc.up.pt/~zp/aulas/1011/pde/geral/bibliografia/MIT.Press.A.Semantic.Web.Primer.eBook-TLFeBOOK.pdf

[14] Beckett, D., Berners-Lee, T. (2008). Turtle - Terse RDF Triple Language - W3C Team Submission. Retrieved July 23, 2009, http://www.w3.org/TeamSubmission/turtle/

[15] Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D.L.; Patel-Schneider, P.F.; and Stein, L.A. OWL Web ontology language 1.0 reference. W3C proposed recommendation (www.w3.org/TR/owl-ref).

[16] N. F. Noy and D. L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technical Report SMI-2001-0880, Stanford Medical Informatics, 2001.

[17] http://lod2.openlinksw.com/sparql