International Journal of Science and Research (IJSR)
ISSN: 2319-7064
ResearchGate Impact Factor (2018): 0.28 | SJIF (2019): 7.583

Natural Language Interface System for Querying Wikidata

Muhammad Reza Jafari[1], Dr. Vinod Kumar[2]

[1, 2]Delhi Technological University, Department of Computer Science & Engineering, Bawana Road, Delhi 110042, India

Abstract: ***Nowadays, information has a huge impact in our daily life; hence its extraction from the web is a need for everyone. However, due to lack of knowledge of query languages, an end-user cannot extract information from knowledge bases as they should. Asking question in Natural Language from knowledge bases enable end-user to extract data from such knowledge sources in a more efficient and fast way. Wikidata is one of the biggest knowledge bases which extract data in a triple format from Wikipedia. Therefore this research project aims at designing and implementing Natural Language Interface system for querying Wikidata. To achieve this objective, a Natural Language Interface was designed using Wikidata in python for processing the input question of user and give some answer to user. Firstly, preprocessing of input query was done, which involved tokenization, Part of Speech (POS) Tagging, Stemming and Lemmatization, Name Entity Recognition (NER), Chunking, Synonym check for words taken using WordNet. Then, Question Classification and SPARQL were performed to minimize the search area, show the type of output, and display the answer to the user respectively. The results of the experiment showed that processing is much efficient based on response time.***

Keywords: Natural Language Processing, Wikidata, Natural language Interface, Question Answering System

1. Introduction

Information in Web, Databases is increasing day by day. Nowadays, information retrieval from web is a challenging issue because of different platform, structure, and schema. Data is shown in form of excel, access, relational, JSON, PowerPoint, html, etc. One way to retrieve data from web is semantic web; which is used to make data understandable by computer. The big challenge with semantic web is that, end-user without knowledge of query languages cannot extract data from web, that it enables only expert users to extract information from the web. To address the gap between casual users and retrieval data, Natural Language Processing (NLP) is used. NLP hides the formality as well as executable query languages from users. Many mechanisms have been proposed by different authors and researchers [2-3] and among them include the Ontology-based Natural Language Interface (ONLI) [1], it uses an ontology model to represent question structure and context. It classifies question and by help of that, suggests answers, which have higher correct probabilities. Wikidata [4, 5 and 11] is one of the biggest knowledge bases which bring information in triplet format Wikipedia. Wikidata data available in Resource Description Framework (RDF), RDF is a data model which represents interlinked data in triple format. In this work, an ontology is viewed as ‘‘a formal and explicit specification of a shared conceptualization’’ [6].

The Wikidata repository consists of Items with an identifier. Each item has a statement that contains claim and an optional reference, used to support the claim. A claim consists of property and value and optional quantifier, the type of values which is either items or variable of different type (i.e. integer, dates, string, URLs, geographical coordinates) [14] and each item has a label.

This work is aims at enabling end-users to query in natural language from this Wikidata knowledge base. The system will get input query of user in natural language do some preprocessing steps on it, and then find the synonyms for each word to find the best knowledge base resource for input query of user. It will identify the WH-questions inside the input query of user in order to minimize the search area in knowledge base and find the type of output. The next step is to generate the SPARQL [7] query against the Wikidata knowledge bases and by help of weight function and show the related result to the user.

2. Existing Literature

Natural Language Interfaces to Databases (NLIDB)
It is an NLI for databases that allow casual users to use natural language for querying data of databases. In this paper, it used dependency parsing and use dependency grammar. It used of a domain-independent Natural Languages Interface Data Base (NLIDB) in order to achieve information from input user and then process it. NLIDB do preprocessing on input by parsing and make a sentence tree. [2].

Natural Language and Keyword Based Interface to Database (NLKBIDB)
In [3], it proposed Natural Language and Keyword based interface for databases (NLKBDB). NLKBDB first do the tokenization on input of user that will be in natural language. It uses lexical analyzer and syntax analyzer. And at end it generated a SQL query the performance of NLKBIDB shows 53% increase in accuracy compare to NLIDB.

Natural language interfaces to knowledge bases (NLIKB)
In [15] perform a work on portable NLIKB like Quelo NLI. It use from concept and relationships and help user to build its query language by allowing adding, changing or deleting the part of English.

In [1] Natural Language Interface to Knowledge Bases (NLIB) provide controlled natural language interface. It interpreted controlled query into SPRQL query by help of

description logic concepts and query description trees. It had F-measure value of 0.85 in its experiments.

Semantic Web Search Using Natural Language (SWSNL)
In [4], Semantic Web Search Using Natural Language (SWSNL) the user input analyzed by some preprocessing NLP such as Name Entity Recognition (NER) and semantic analysis and then representation of knowledge base independent semantic of input user is created. And then it change to query against knowledge base. The performance of the system is about 0.73 F-measure.

3. Proposed Method

In this paper, The Natural Language Processing Interface (NLPI) for end user using Wikidata Knowledge base and WordNet was proposed to address the gap between end-user and Wikidata.

The system get query of user in natural language and analyze it, and put it in normalized form, find the synonyms for entities extracted from input query of user by help of WordNet, do the classification on the search and add a weight to each of entities and generate the SPARQL, by help of weight function assign score to each result and show the best suitable result to user as an output. This process has two main steps: 1. Question Processing and Answer Processing as shown in Figure 1. Each step was described in detail as bellow.

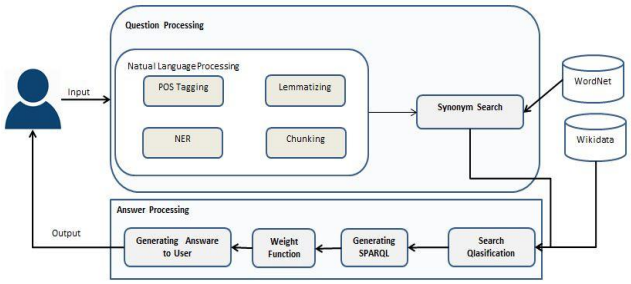

Figure 1: Natural Language Interface System

3.1 Question Processing

Question Processing involves normalizing the user question for the system. It has two main steps; Natural Language Processing (NLP) and Synonym search.

1) NLP has the following steps:

a) **Part of Speech (POS)-Tagging:** Tokenization and Part of Speech (POS) Tagging [9] is process of tokenize the statement of user and assign the part of speech tag to each word which is necessary for next steps.
b) **Stemming:** The process of removing affix from the words to achieve the dictionary form of words called stemming. This process will help us to deal with root form of words.
c) **Lemmatizing:** Sometimes the word that defines a knowledge entity is not the same that the contained in the question or in other word the word has same meaning but different form i.e. Good = Better.
d) **Name Entity Recognition (NER):** The Name Entity Recognition identifies the name in the question and extracted it from preprocessing.
e) **Chunking:** Chunking [13] is process of grouping the word by identification of Part of Speech and short term like noun phrase.

2) Synonym Search
Sometimes the word in input is different from the word in ontology from Wikidata but has the same meaning so we searched the synonym by help of WordNet [10] and could use them in searching the entity ontology of Wikidata.

3.2 Answer Processing

This process is used to find answer for user by the data that gather from question processing.

1) Search Classification:
To reduce the search area of query and improve the performance of system we use a mechanism to identify the type of question, i.e. when input question to the system contains "Where" it means propose of search is some area, building or location, while when user question contains "Who" it means the propose of search is someone or some organization. Table 2 shows the detail of answer type. By this method we can reduce the search space in Wikidata.

Table 1: Classification of WH-Question

Classification of Question	*Subset of Question*	*Type of Answer*
What	What What-Who What-When What-Where	Money: Numeric: xsd:double|number Noun: Name, Work Person, Organization (dbpedia-owl: Person/ Organization) Date: xsd:date Location: dbpedia-owl:Place
Who		Person, Organization: dbpedia-owl:Person|organization
How	How How-Many How-Much How-Far How-Tall How-Large	Manner Numeric: xsd:double, xsd:integer Money, Price: xsd:double Distance (Numeric): xsd:integer Numeric: dbpedia-owl:height Length (Numeric): xsd:integer, dbpprop:length
Where		Location, Place: dbpedia-owl:Place
When		Date: xsd:date
Which	Which-Who Which-Where	Person: dbpedia-owl:Person Location, Place: dbpedia-owl:Place

	Which-When Which-What	Date: xsd:date Proper noun: dbpprop:name, dbpedia-owl:Work,dbpedia-owl:Species, bpedia-owl: Disease|organization:dbpedia-owl:Agent
Why		Reason
Whom		Person, Organization: dbpedia-owl: Person|organization:dbpedia-owl: Agent
Name	Name-Who	Person: dbpedia-owl: Person| organization: dbpedia-owl:Agent

a) **Search Entities:** After analyzing the question, the NLPI search for entities of knowledge base that is related with input text and its synonyms. This process helps to achieve information that is necessary for answer the question of user. Information can be URI of entities, Label of entities, or type of entity. After that a SPQRAL query will generate for every words and its synonyms.
b) **Generating SPARQL:** Once you are done by synonym search you can generate a SPARQL. So entities extracted from previews steps can use to generate SPARQL. In this paper use from SPARQLWrapper from NLTK [8] packages in Python. URIs of resources shows as results of SPARQL. There are different types for searching the Wikidata individual, ObjectProperty, DatatypeProperty, Concept. The knowledge entities contain in question belong to one of this type, according to this four type of search, SPARQL could be different.
c) **Individual:** That is knowledge entities that extracted from input question and they are individuals on ontology (easiest).
d) **Datatype Property:** Knowledge entities that extracted from input question and they are datatype Properties on ontology.
e) **Object Property:** Knowledge entities that extracted from input question and they are Object Properties on ontology.
f) **Concept:** Knowledge entities that extracted from input question and they are classes on ontology

2) Weight Function

Weight function helps to give a score to each knowledge entities in order to identify the similarity level of entities with input elements. The knowledge entity that has highest similarity is the closest answer to user question. This paper uses Knowledge based similarity approach to find weight of words, WuPalmer similarity used, it is path length similarity knowledge based. Wup calculate the similarity based on depths of two synsets in WordNet taxonomies, along with the depth of the LCS(Least Common Subsumer).

$$Wup = 2 * (depth(lcs(s1,s2)/depth(s1)+depth(s2)) \quad (1)$$

The score is in range of [0-1], it uses for disambiguation of words, the words with heights score has most similarity.

3) Generating Answer:

After answers were obtained, the score which is provided to each result by weighted function consider, the result with highest similarity is more accurate result to user which is shown to user.

4. Evaluation

The evaluation had two steps:
1) How accurate the system is which is showed in Table 2.
2) The time which is needed to build a query in SPARQL.

There are five students in M-Tech who are familiar with RDF and Wikidata and give to each 10 questions to make SPARQL, the result was shown in Figure 2.

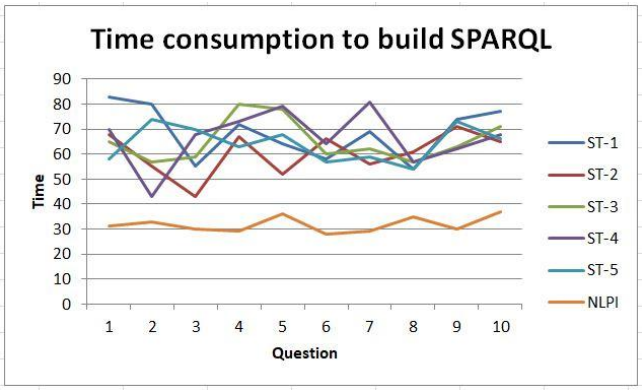

Figure 2: Time consumption to build SPARQL query

Precision = (correct knowledge entities achieve in input) / (total knowledge entities obtained in input)
Recall = (correct knowledge entities achieved in input) / (total knowledge entities achieved in SPARQL query)
F-measure = (2* precision *recall) / (precision +recall)

The 5 student's precision, recall and F-measure are calculated the average is 0.81 is achieved and NLPI system can obtain F-measure of 0.83 which is better than average. If we compare the two approaches they are more or less the same but the time consumption in NLPI system is lower than manually generating SPARQL.

Table 2: Evaluation result

Student	Precision	Recall	F-measure
1	0.81	0.85	0.83
2	0.75	0.81	0.78
3	0.80	0.83	0.81
4	0.81	0.84	0.82
5	0.79	0.83	0.81
Average	0.79	0.83	0.81

5. Conclusion and Future Scope

Retrieving information from web is becoming a challenge for everyone, especially for users without any knowledge about query languages. This paper addresses the gap between end-user and knowledge in this case Wikidata. There are many systems which are developed in natural language processing for different knowledgebase, but this work developed a system for querying Wikidata for the first time in natural

language using ontology. The user asks a question in natural language from Wikidata and after some preprocessing on input text identifies the potential entity and its synonyms and do the query against the Wikidata knowledge base. By help of weighting function it will find the most probable answer to user. It shows that it had a good performance (F-measure 0.83) and it has a less time. For future work it can develop for other languages with fewer resources.

References

[1] Paredes-Valverde, Mario Andrés, Miguel Ángel Rodríguez-García, Antonio Ruiz-Martínez, Rafael Valencia-García, and Giner Alor-Hernández. 2015. "ONLI: An Ontology-Based System for Querying DBpedia Using Natural Language Paradigm." Expert Systems with Applications 42 (12). Elsevier Ltd: 5163–76. doi:10.1016/j.eswa.2015.02.034.

[2] Reinaldha, Filbert, and Tricya E. Widagdo. 2014. "Natural Language Interfaces to Database (NLIDB): Question Handling and Unit Conversion." In Proceedings of 2014 International Conference on Data and Software Engineering, ICODSE 2014. Institute of Electrical and Electronics Engineers Inc. doi:10.1109/ICODSE.2014.7062663.

[3] Shah, Axita, Jyoti Pareek, Hemal Patel, and Namrata Panchal. 2013. "NLKBIDB - Natural Language and Keyword Based Interface to Database." In Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013, 1569–76. doi:10.1109/ICACCI.2013.6637414.

[4] Habernal, I., & Konopík, M. (2013a). SWSNL: Semantic web search using natural language. Expert Systems with Applications, 40(9), 3649–3664. http://dx.doi.org/ 10.1016/j.eswa.2012.12.070.

[5] Wiki.dbpedia.org. (2019). About | DBpedia. [online] Available at: https://wiki.dbpedia.org/about [Accessed 1 Nov. 2019].

[6] Lassila, O., & Swick, R. R. (1998). Resource description framework (RDF) model and syntax specification.

[7] Prud'hommeaux, E., & Seaborne, A. (2006). SPARQL query language for RDF. W3C Working Draft. World Wide Web Consortium, February. Latest version: http://www. w3. org/TR/rdf-sparql-query.

[8] Bird, Steven. 2006. "NLTK." In , 69–72. Association for Computational Linguistics (ACL). doi:10.3115/1225403.1225421.

[9] PVS, A., & Karthik, G. (2007). Part-of-speech tagging and chunking using conditional random fields and transformation based learning. Shallow Parsing for South Asian Languages, 21, 21-24.

[10] Kilgarriff, Adam, and Christiane Fellbaum. 2000. "WordNet: An Electronic Lexical Database." Language 76 (3). JSTOR: 706. doi:10.2307/417141.

[11] "Introduction" Wikidata, https://www.wikidata.org/wiki/Wikidata:Introduction

[12] Warin, M., Oxhammar, H., & Volk, M. (2005). Enriching an ontology with wordnet based on similarity measures.

[13] Parsing, C. (2009). Speech and language processing./ch13 p455/

[14] Bissig, F. (2015). Drawing questions from wikidata. PDF). Zurich, Switzerland: Distributed Computing Group.

[15] Franconi, E., Gardent, C., Juarez-Castro, X., & Perez-Beltrachini, L. (2014, October). Quelo natural language interface: Generating queries and answer descriptions.

Author Profile

Muhammad Reza Jafari completed a Bachelor Degree of Computer Science from Khaja Abdullah Ansari Private Higher Education Institute in 2017. He is currently pursuing M. Tech in CSE at DTU, expecting to graduate in 2020.

Dr. Vinod Kumar is an Associated Professor of Computer Science & Engineering at Delhi Technological University and Research supervisor.