International Journal of Research in Engineering, Science and Management
Volume-2, Issue-12, December-2019
www.ijresm.com | ISSN (Online): 2581-5792

Construction of a Knowledge Graph for Query System

Abhilash. T. Bedadur[1], Deshmukh S. Vaishali[2]
[1]*Student, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune, India*
[2]*Assistant Professor, Dept. of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune, India*

***Abstract*: The Knowledge Graph (KG) plays an important role for any text based retrieval system. The main challenge is to construct KG due to its nature. The KG is constructed by transforming the unstructured text into the structured text. This facilitate the integration and retrieval of the information. There are many approaches have been proposed for the construction of the KG by exploiting Machine learning algorithms, semantic web, etc. with the existing systems. The application of these knowledge graph feeds most of the real world problems. In this paper we aim to design a human friendly question answering interface with higher level of accuracy and increased customer response satisfaction. We aim to replace the rudimentary systems that rely on string parsing and response storage table, with an enhanced and more accurate system that could answer a multitude of questions quickly and correctly. The system aims at providing succinct information for answers that need to be extracted using wisdom about the relations that are implied coherently.**

***Keywords*: Knowledge Graph, Semantic web.**

1. Introduction

An information in any form is consumed day by day by the people in many forms such as web applications, social networking, banks, libraries, search engines etc. The growth of these kind of information is exponential. Since the data of these information is humongous and fed to the end user continuously, most of the data is unstructured. It is estimated that 95% of the data that propels on the web is unstructured that means most of the valuable information is not been explored. Hence this unstructured data needs to be transformed into the structured data in order to be used by users and processed by application. There various and effective approaches such as the Machine Learning, Information Extraction, Natural Language Processing for transforming the data.

With the help of the Knowledge Graph we could develop a question answer system which we will be calling it as the 'Chatbot'. [6] The Frequently Asked Questions are just that frequently asked questions, for not-so-frequently asked questions you're left alone, or rather left to figure out how to get the human help. A chatbot only just covers up the question which are been programmed or which are pre-fetched from the database or Frequently Asked Questions which makes it difficult for user if the answers they get is not satisfactory. Often times, in public domain, users come with questions such as "where is my order" or "I need to change my upcoming appointment". Chatbots built on FAQs are not able answer this kind of questions.to design a human friendly question answering interface with higher level of accuracy and increased customer response satisfaction. In this paper, we aim to replace the rudimentary systems that rely on string parsing and response storage table, with an enhanced and more accurate system that could answer a multitude of questions quickly and correctly. The system aims at providing succinct information for answers that need to be extracted using wisdom about the relations that are implied coherently.

2. Literature survey

The Paper [1] T2KG: An End-to-End System for Creating Knowledge Graph from Unstructured text: This paper proposes an end-to-end system, it is a hybrid combination of rule based approach and a similarity based approach. This approach is further used for mapping a predicate of a triple extracted from unstructured text to its identical predicate in a knowledge graph. This paper only involves construction process. Further querying is transformation are not discussed [1].

The paper [2] OpenIE-based approach Knowledge Graph construction from text: The paper proposes an approach to generate the knowledge graph by using binary relation produced by an Open Information Extraction (OpenIE). The strategies used here for favouring the extraction and linking of named entities with the knowledge graph individuals and using the grammatical units to generate more logical facts. Here the potential RDF triples is also created for knowledge graph by providing decision for selecting the extracted information. An efficient way to counter problems pertaining to entity selection, entity and property linking and representation is explained. Although it covers relations useful for specific representations of sentences, users may face contrasting data needs that are better addressed by other kinds of approaches for a comprehensive linguistic analysis [2].

The paper [3] Querying Web-Scale Knowledge Graphs Through Effective Pruning of Search Space to accelerate query processing: This paper proposes an algorithm which is efficient for finding the best k answer of a query given for system without computing any graph indices. A novel technique for bounding the matching scores during the computation. The

quality of an answers is depended upon the score which have been computed online. The bounds depending on the scores low quality answers can be pruned efficiently. The only disadvantage is decision of lower bound should be optimal for obtaining results from the graph [3].

The paper [4] Exploiting Linked Data and Knowledge Graphs in Large Organizations: This paper Proposes a detailed describing the use of knowledge graphs in Question Answering systems and effective. Gives brief discussion for effectively deploying linked-data graphs with the organization. Best suitable for Enterprise documents, efficiency may be lesser for normal QA systems [4].

3. Proposed methodology

This section presents the proposed method of the construction of the knowledge graph from the plain text received from the user is the form of a query. The proposed method for constructing knowledge graph is based upon the combination of Natural Language Processing (NLP) and Information Extraction (IE). [2] We create a chatbot for handling the user query with the help of the knowledge graph which is the backbone of the whole system. The queries here are in the form of plaintext which we call it as the question(?). This query further is mapped with the existing knowledge graph by rule based of either with the similarity based predicate mapping. The response is generated depended upon the relevancy of the expected answer and it is fed to the user.

We propose an architecture for generation of the knowledge graph, there are mainly 5 major components: 1) Pre-processing 2) Entity mapping 3) Triple Extraction 4) RDF triples 5) Predicate Mapping. Once the query is received in the form of the plain text, we presume the text as the unstructured, the next step is to pre-process the text in order to parse elements of the information and other descriptions which will be useful for extraction and mapping further. The entity mapping is used for linking an entity from the unstructured to the corresponding entity in the knowledge graph. The Coreference Resolution is used for detecting the different expression, pronouns, abbreviation. The aim of RDF is to extract triples from the unstructured texts. Predicate Mapping is used to map triples to an identical triple in the knowledge graph.

A. Architecture

As shown in figure 1 The contribution work is, to implement the knowledge Graph adjoining with the various entities.

The plaintext is assumed to be the unstructured text received from the user. For pre-processing we Natural Language Processing (NLP). The NLP has four major components for extracting the elements:

- Sentence Segmentation: Segmentation in which the input text is split into the sentences. Here the text is efficiently organized into sequence of small, self-contained grammatical clause for processing further. It is helpful for interpretation of the text.
- Parts-Of-Speech (POS): Every word in the plain text has grammatical values, it may be either noun, pronoun, verb, preposition etc. The POS categorize such word.
- Syntax tree parsing: Parsing is done in order to organize the group of words according to their grammatical sense.

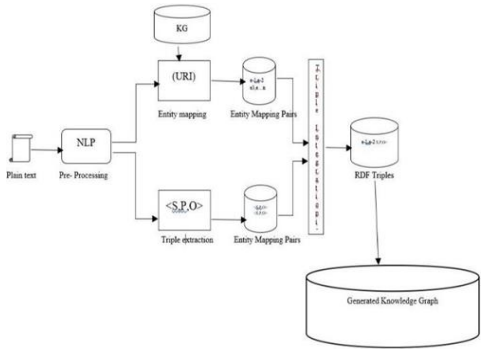

Fig. 1. Proposed system architecture

The entity mapping is used for linking an entity from the unstructured to the corresponding entity in the knowledge graph. The mapping is done from unstructured text and the output is Uniform Resource Identifier (URI). The extracted entity is mapped into an identical entity in the knowledge graph, the URI should match the in that knowledge graph if not then a new URI is given to that entity. Further the triple extractions are used to extract triples, by applying linguistic theory, the meaning of arbitrary sentence is interpreted by observing the relations and association among the sentences. The pairs of Entity mapping and Triple extraction are fed to the triple integration in which they are integrated and generate the Resource Description Framework (RDF), After the Generation of the Knowledge Graph, the response selector selects the response by doing domain specified calculations for processing the user request, and select the response according to the relevancy of the match value.

B. Algorithms

Data: PlainText sentence, Mapping entities

Result: *NP _entities //*

Chunk-tags ←

ObtainConstituency(sentence)

NP _entities ← *{∅}*;
NP s ← FilterNPs(Chunk-tags); /* Keep NP chunks only *{np*0, *np*1, ..., *npj*-1*}* */ **forall the** *np* ∈ *NP s* **do** *assocEntities* ← *{∅}*; **forall the** *ne* ∈ *EEL* **do** /* Iterate over entities */ **if** *ne.SF* ⊆ *np* **then** /* Matching surface form (SF) against NP */ *assocEntities* ← *assocEntities* ∪ *ne*; **end end**

NP _entities.append(*hnp, assocEntitiesi*);
end
Data: NP_entities:Ent, SRLpredicates, RE:R
Result: NP_entities selected for representation
pred ←
Matching(R.predicate,SRLpredicates); /* Get SRL predicate matching the verb phrase in the Semantic Relation */ *case* ←
DetermineCase(*pred*); /* Every predicate has arguments A0, A1, AN */ **if** *case* = 1 **then** /* Complete identification of arguments */
/* Search for an NP-entity matching any of the arguments */ *agent* ←
EntityMatching(*pred.arg*0,Ent); *consumer* ←
EntityMatching(*pred.arg*1,Ent); **end**
if *case* = 2 **then** /* Partial identification */
if *pred.arg*0 **then** /* If argument identified is 0 */
agent ←
EntityMatching(*pred.arg*0,Ent) /* Identifies if entities are taken from subject or object of the semantic relation */ *position* ←
GetRelPosition(*agent*); *consumer* ←
GetNearestEntity(Ent.*position*); **Else** start assigning consumer; /* The same process starting with patient */ **end end if** *case* = 3 **then** /* No identification */ *agent* ←
GetNearestEntity(Ent.*subject*);
consumer←
GetNearestEntity(Ent.*object*); **end**

4. Result and discussion

The experiments we have conducted is designed to evaluate the performance of the knowledge Graph and the system. In the experiment a series of the Wikipedia articles are selected randomly and applied it for the preprocessing. In preprocessing the hyperlinks, annotations, HTML markups are removed, any duplicate sentences which are occurred are also been eliminated. The results of the processing have been observed as below.

Appr.	Precision	Recall	F-measure
Rule Base	0.54123	0.344	0.4341
Our System	0.54111	0.312	0.41223

The system will have higher efficiency than the rudimentary one and will be able to replace it with further advancements. The information which is consumed by the human users on the Internet/Web are unstructured in nature, which therefore makes it difficult to processed by the applications.

5. Conclusion

The proposed QA system will provide relevant application specific information for a vivid range of questions, answers of which can be obtained as certain relation from the dataset pertaining to the application stored as a graph. The system will have higher efficiency than the rudimentary one and will be able to replace it with further advancements. The information which is consumed by the human users on the Internet/Web are unstructured in nature, which therefore makes it difficult to processed by the applications. To overcome this issue, the Semantic Web 1460 provides a way to structure this unstructured information through various data models, extractions, vocabularies, and other such tools. Although it may seem to be a solution for improving information consumption, representing information on a formal structure is a very complex and time-consuming process because on 1460 structured data do not have features and description to support formal representation

References

[1] J. Martinez-Rodriguez, I. Lopez-Arevalo and A. Rios-Alvarado, "OpenIE based approach for Knowledge Graph construction from text", Expert Systems with Applications, vol. 113, pp. 339-355, 2018
[2] J. Jin, J. Luo, S. Khemmarat and L. Gao, "Querying Web-Scale Knowledge Graphs Through Effective Pruning of Search Space", IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 8, pp. 23422356, 2017.
[3] Je Z. Pan, Guido Vetere, Jose Maneul, Gomez-Perez, Honghan Wu" Exploiting Linked Data and Knowledge Linked Data Graph in Large Organisations", 2017.
[4] N. Kertkeidkachorn and R. Ichise, "T2KG: An End-to-End System for Creating Knowledge Graph from Unstructured Text", The AAAI-17 Workshop, 2018.
[5] Y P. Surmenok, "Chatbot Architecture – Pavel Surmenok – Medium", 2016.
https://medium.com/@surmenok/chatbotarchitecture-496f5bf820ed.
[6] K. Panetta, "5 Trends Emerge in the Gartner Hype Cycle for Emerging Technologies, 2018", Gartner.com, 2018.
[7] Augenstein, I., Maynard, D., and Ciravegna, F. (2016). Distantly supervised web relation extraction for knowledge base population. 1510 Semantic Web Journal, 7, 335–349.
[8] Berners-LeeT. (2010) The future of rdf,
https://www.w3.org/DesignIssues/RDF-Future.html
[9] Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; and Ives, Z. 2007. DBpedia: A nucleus for a web of open data. Springer
[10] Augenstein, I.; Pado, S.; and Rudolph, S. 2012. Lodifier: Generating linked data from unstructured text in The Semantic Web: Research and Applications. Springer. 210–224.