Intelligent Question Answering System
NLP, Knowledge Graph, Machine Learning.

Sai Sriharsha Sudulaguntla
Ss42f@mail.umkc.edu

Prudhvi Sai Suggala
Ps5m6@mail.umkc.edu

Lakshminikitha Kona
Lnkwd5@mail.umkc.edu

Sadanand Kallakuri
Sk789@mail.umkc.edu

Abstract— A domain expert can process data to make meaningful interpretations and answer question from the data. However, this is an expert- and manual-intensive task. This paper presents an end-to-end system that integrates heterogeneous data sources into a knowledge graph in the RDF (Resource Description Framework) format using an ontology which is used to build intelligent question answering system. Then the user can easily question the system on which the we have constructed a knowledge graph and ontology of the same, so that the model answers the users input question after several Natural Language Processing steps on the question and the model determines the question type and assigns the question to the particular matching question type models present in the model. The question is then processed by the specific question type model which fetches the most accurate answer from the knowledge graph and the machine learning process implemented will ensure the low error rate while fetching the answer as fast as possible. We tested the system using the several question on the Obama dataset and achieved 83.3% accuracy on overall question answering system.

Introduction:

Intelligent Question answering system is primarily based on NLP; Natural language processing. The input dataset Obama dataset is subjected to Natural Language Processing, the data is subjected to the process of lemmatization, tokenization and stop word removal. Post this process the Information Retrevial is performed. In Information Retrieval process the Term Frequency TF and Inverse Document Frequency IDF are generated. The term frequency is then compared to eh inverse term frequency in the document. Basic identification of sets of words that are discriminative for
documents in the collection. A document containing such a term is more likely to be relevant than a document that doesn't, but it is not a sure indicator of relevance.
For frequent terms, we want positive weights for words like high, increase, and line but lower weights than for rare terms. The tf-idf weight of a term is the product of its tf weight and its idf weight. Best known weighting scheme in information retrieval Increases with the number of occurrences within a document Increases with the rarity of the term in the collection. Thus the weight of each term in the corpus is determined.

Then the corpus data is subjected to further Information Retrieval approaches such as N-Gram and Word2Vec. N-gram: a sequential list of n words, often used in information retrieval and language modeling to encode the likelihood that the phrase will appear in the future. A two-layer neural net that processes text. Input is a text corpus. Output is a set of vectors: feature vectors for words in that corpus. Measuring cosine similarity. Word2vec “vectorizes” about words for natural language computer-readable performing operations on words

to detect their similarities. Word2vec trains words against other words that neighbor them in the input corpus. Then LDA is a generative probabilistic model of a corpus. Document modeling, text classification, image processing, collaborative filtering is done by LDA. Further, the processed corpus is subjected to OpenIE and wordnet to generate triplets and synonyms respectively for the corpus data.

A. *Approach*

We used different approaches bases on the semantics of sentences and ontology based representation which produced a knowledge graph. Both approaches are mainly used to structure the natural language and try to deduce inference on it. The Ontology base approach help to create the creates the more relations because of the better representation of data as the entities and their hierarchical relations. The Syntactical bases approach helps for a fixed syntax type, there are rather considered as the syntax finders than the inference machines like knowledge representations. The syntax based approach are really complex to identify, but once this could be done by an expert this probably be a robust if the our syntax book. This would be very complex problem to make this work for contextual base question.

B. *Syntax Approach*

We try to match a syntax that we identified from observing many question,We try to look for the same type of occurrence of the word patters, Question words and other patterns from the query from the user and same kind of pipeline of functions are done on the data set to remove the redundant information from the dataset and work with the rest of the data. By this step we had completed the basic preprocessing technique which cab further be used to apply NLP techniques.

We perform basic NLP tasks to the data and then try to match one of our syntax to the data and get the keywords extracted from data and presented to user. We had observed the most probable questions that could be from users and Accordingly made the syntax to each and every question type that we made. We though that When would be the key word that can be answered as the location of a particular place in the dataset and try to search for the subject mentioned the question to get the appropriate locations among all the locations available in the dataset.

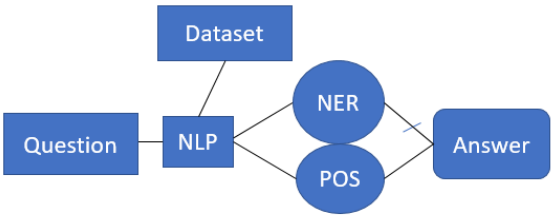

Figure 1: pipeline for Questions below

Question: When did the incident take place ?
Answer: Incident took place at Kansas

Question: Who are the main character in the data set
Answer: Main character Barack Obama

We ran the above pipe line of the process on the question to that results in identifying *when* and *incident*, this process identifies questions words by POS tagging and identifies *Incident* NER tagging. this is searched among the processed dataset to find and populate the answer. Other Question is answered by the process of identifying the PERSON tag from the NER tag and finding the highest frequency would results it.

Our second approach deals with finding the full name of the name which is provided in the question provided by the user, we uses B-gram approach to make many different grams from the corpus and we try to search for the consecutive NN entity in the POS tags this filters other junk grams from the answer. we also find the most occurrence from them.

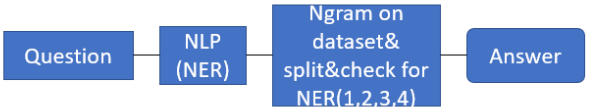

Figure 2: pipeline for full-name Question

Question : What is the full name of Obama?
Answer : *Michael Obama*

We the below architecture to handle this kind of questions, we solve this kind of question by the Tf-Idf and Wordnet to find the most important words in the corpus and the Wordnet help used to find the synonyms and Hypernyms using the wordnet. We used a external api to get data from the wordnet. we found the Tfidf words for the total corpus if we find the most frequent words or other equivalent words in the question.
Wordnet with the certain parameters gives the synonyms for the word identified from the question.

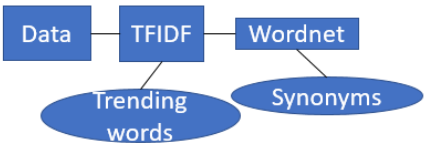

Figure 3: pipeline for the wordnet and Synonym

Question: What are the trending topics in the Dataset?
Answer: Obama, Michael

Question: What is the Synonym of the Word XXX?
Answer: synonyms of xxx

Then to make some more complex pipeline this would find some more complex analogies from the data, We dealt with pipeline to extract open triplets can be sentences of the corpus and retracting the required triplets from them by filtering them with key words in the question and we check the same words in the subject of the triplets and return them to find the answer in the predicate of the same triplet. This technique answers the below question, We also solve the other question which is mentioned below this try to search the numbers from the corpus and check the key words from the question to occur in adjacent to them to make sure that it would the possible answer.

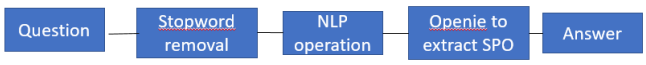

Figure 4: pipeline with openie

Question: Who is president of United States ?
Answer: Barack Obama

Question : When Obama was born ?
Answer : Date in corpus

The finally , we try to answer the abstract questions by a quick method than by normal searching , we try to map each sentence to the stopword removal of the and them to then we pass it to the openie then this creates less numbers of more important words to be searches . so, this would be helpful in the serch of that particular required sentences and then this would be served with openie on that filtered sentences which contain questions key phrases from the question .

C. Knowledge graph based approach

\

This approach aims to create the knowledge graph for the text corpus , user try to find the classes using the main entities from the corpus and this helps to main classes for creating the knowledge graph . Then we extract the individuals for the extracted classes the we find some relations form the text corpus which results in the knowledge . We performed openie to get triplets and found every thing required to knowledge graph .We used the Obama dataset which had results in the below knowledgegraph. .

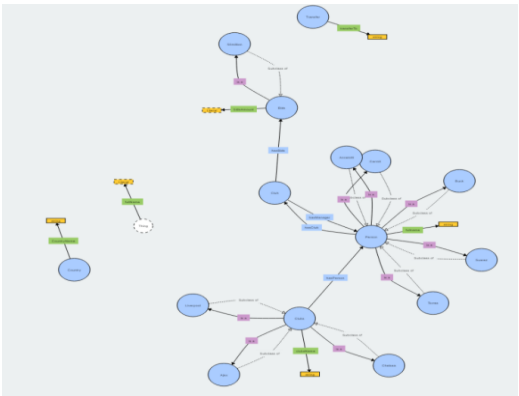

Figure 5:Knowledge graph for Obama dataset

We use this knowledge graph for the further query on it to get our answers from the questions , We do restrict our questions to the write in the syntactical way i.e we need to query up by the relations available in the knowledge graph we described below , so this would be tedious task to one who don't know the relations between the entities in the graph . this searches for the main object in the graph and searches for the appropriate relation to the that object and results to the user .

Question : Electionon *value* Obama
Answer : 20jan2009
Question : memberof *value* Obama
Answer : Democratic

In the above samples of the questions , Electionon , memberof defines the relation between the two entites Obama and value . this also created reasonable answers for the questions but the questions starts to become more logical this try to fail .

D. *Analysis & results*

We used both methods on the same dataset syntax and knowledge based where both methods used to perform similar NLP task on the question processing and then we choose different procedures bases for the syntax . We fetch required identified reaction values from the knowledge graph .

We uses a set of 37 questions to analyze the performance of the both systems , they are simple questions based on the dataset that we had chosen . We try to consider all 9 different types of syntax for the approach and we used the creates knowledge graph for the second approach , we just added a step to directly write a query manually if we have appropriate matching relation exist if not we considered as a missed question . We recorded a 12 0f 37 for approach 17 and 37 out for graph based approach .

Being a question answering system this is can't be enough to interactive to a person the main improvements that we need to do for the approach , For the approach 1 this need to brutal manual identification of best questions from both dataset and the question making . So this method needs to be depreciated.

For approach 2 , We just used the open ie before the construction of relations from the graph ,we think coreference might deduce good relations and helps to infer for more complex questions and updating the question pipeline for approach 2 which don't exist till now would be one of the reason for more unanswered questions.

E. *Future work*

We try to add machine learning review system on top of it , we try to collect the rating of answer randomly and try to save all the significantly highly rated questions and try to categorize the similar questions and learn the synonyms for them . This would be external learning for the improvement of the question part of the system .