International Institute of Information Technology Hyderabad

Internship Project

Open Domain Question Answering for Telugu Language

LTRC Lab

June 26, 2021

Contributors

- Makka Venu-B161087
- Sk Shareef-B161109
- Aishwarya Vuppala-B161256

Contents

1 Introduction

Languages play an important role in communication whether it is among humans or machines. Around 6500 languages are spoken in the world, among those 1652 are from India.In the modern era of communication, everyone wants to get all the information in their native language to lead the ease of accomplishing their day to day life.

Telugu is one of the widely spoken languages in southern parts of India. Except for some dialogue based Question Answering(QA) systems, no other state-of-the-art model is implemented to do QA in web-based applications for the Telugu language.

1.1 Aims and Objectives

Our aim is to create Telugu QA task with a pre-processed data set of 1455 Queries. Any Natural Language Processing task for low resource languages can be explored by adapting implementations from resource-rich languages. In QA, some advanced information retrieval technologies are being used to extract text from all documents and some models only concentrate on extracting the essential information.

Any QA system should be built with Information retrieval and extraction techniques. Always the core aim of QA lies on the extraction of suitable answers only, not all the related documents to the query.

1.1.1 QA Systems

Many automatic QA systems have been developed over the past few decades.Initially, the rule-based approaches were used as a prominent method to do the question answering task. These approaches use semantics as decision trees to identify the appropriate answer to the given query.

Manual creation of the rules was the main drawback of these systems. The statistical approaches like overcame the drawbacks of the rule-based approaches by predicting the suitable answers based on the huge chunks of available data. Later, it turned to the machine learning models for predicting the answer which comes under knowledge based question answering.

As the information grows rapidly, the features get very expensive to process. As a result the processing speed decreases with the growth of data. To overcome this, deep learning concepts are applied to knowledge based question answering to predict the correct answer for user query.

1.1.2 Telugu QA Model

One of the main tasks of QA system is **Question Classifier(QC)** which is not available for Telugu. To design a Telugu QA model we first created the data set for QC module and then trained Support Vector Machine (SVM), Logistic Regression (LR) and Multi-layer Perceptron (MLP) classifiers.

Using the SVM, LR and MLP, our model obtained the classification accuracies of 76.6,73.9,73.9 percentages respectively. Model deals with PERSON, LOCATION, ORGANIZATION, DATE, TIME, NUMBER, PERCENTAGE,CURRENCY kind of answer type categories.

1.2 Overview of the Report

In this report we have explained about the past work had done in the field of QA Systems. The characteristics of data set is explained in the Corpus Description.The corpus is mainly divided in PERSON, ORGANIZATION, DATE, LOCATION, NUMBER,CURRENCY, TIME and PERCENTAGE.

we have explained Query Processing, Document Processing, Answer Processing in the Model Implementation. The accuracies are tabulated in the results section.

2 Related work

There are many researches going on information retrieval systems for different languages across the Globe. One of the works mentioned by Zheng is a web-based QA system called as AnswerBus. It was implemented for five different languages (English, German, French, Spanish, Italian) by retrieving the information from five search engines (Google, Yahoo, WiseNut,AltaVista and Yahoo News), and also mentioned the overall percentage of predicting exact answers for data set of 200 queries is 70.5 percent.

Some research models laid web based question answering system, had taken information from only snippets and retrieve information from particular search engines like google. This approach reduced the time consumption for answer retrieval and displayed good accuracy.

DeepPavlov Wikipedia-pretrained ODQA system which is based on two component approach namely ranker(which finds relevant articles in a database) and reader(extracts answers from retrieved articles by ranker). It has two pretrained wiki-pedia based models for English(5,180,368 articles) and Russia(1,463,888 articles).

Coming to Telugu open domain question answering, Bandyopadhyay implemented a dialogue based QA architecture on railway information data set, which consists of 82 queries, in those 79 queries produce plausible answers. This approach exhibits the dialogue success rate of 83.96 percent and precision of 96.34 percent.

Avadhan model for Telugu ODQA used statistic data set of of 1037 Telugu queries which are divide into PERSON, ORGANIZATION, DATE, LOCATION, NUMBER, TIME and PERCENTAGE categories based on answer type. Using different classifiers like SVM,LR,MLP they extracted partial(68.5 percentage) and exact(31.6) match accuracy using Bing.

3 Corpus Description

Before implementing the model, we need a labeled data set. We trained our model with a data set which consist 1455 Telugu queries which have categorized labels based on the answer type namely PERSON, ORGANIZATION, DATE, LOCATION, NUMBER, CURRENCY, TIME and PERCENTAGE. It has labeled queries in data set. Data set link is provided

- 346 'PERS'
- 134 'ORGA'
- 135 'DATE'
- 311 'LOCA'
- 363 'NUMB'
- 76 'CURR'
- 35 'TIME'
- 55 'PERC'

$https://drive.google.com/file/d/1yibi3MGPQWKvj6c-rYnfley1w6-2e5hi/view?usp=sharing$

And then we tested our model on the test set which has 300 labeled Telugu queries with answers. Test set link is provided
$https://drive.google.com/file/d/1PGy$fHm54BgXqqtANNZQYoTUmTvmFu5w/view?usp=sharing

4 Model Implementation

Implementation of this model starts with the given input user query, which is in Telugu. To get the answer type for given telugu user query. we need a Question Classifier (QC). The user input needs to be translated into English to extract the relevant information in a search engine.To obtain that,Google translate API is used.So we can implement

this project in 3 modules, those are
1.Question Classification
2.Information Retrieval
3.Answer Extraction.
Below figure represents Achitecture of this project.

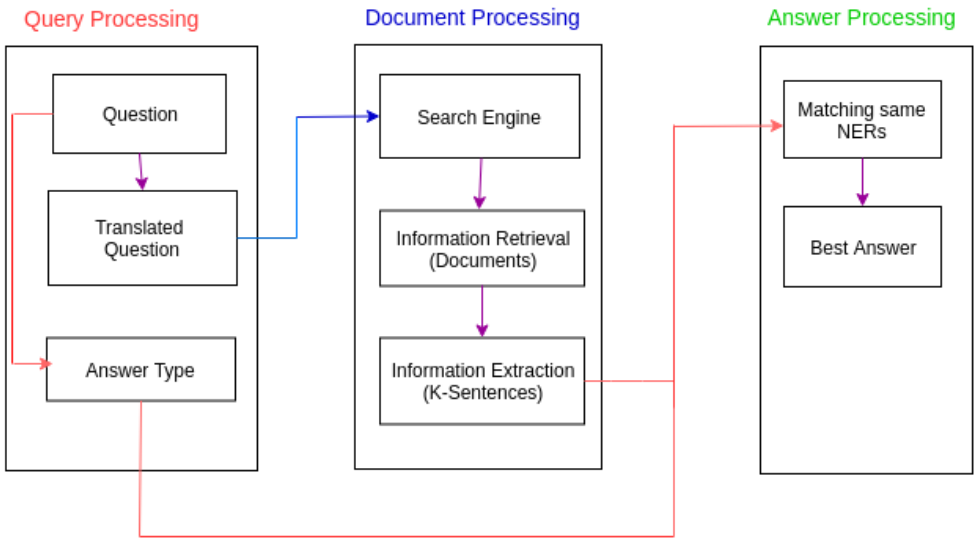

Figure 1: *Architecture of ODQA for Telugu language system*

4.1 Question Classification

To get the answer type for a user query, initially we have to classify the query into predefined Named Entity Recognizer (NERs) classes (Person, Location, Organization,Date,Number,Currency). This predicted classes will play a major role in finding the answers accurately. Modeling a Question Classifier(QC) is a difficult task for Telugu, as there is no pre-tagged data. Even with sufficient pre-tagged data for QC, finding which classifier works better for Telugu is even more challenging task. To do this, experiments were performed with different baseline neural network classifiers like LR, MLP and SVM.
All classifiers need input to be in the form of vector representation, for that Term Frequency Inverse Document Frequency (TF-IDF) vectorizer was used and each query is considered as a document and each word as a term. TF-IDF used to transform text into a meaningful representation of numbers. It is the product of termfrequency (tf) and inverse document frequency (idf).

Pre-tagged data divided into train and test data like 1091, 364 respectively and

classifiers (MLP, LR, SVM) accuracies are 73.9, 73.9, 76.6 percentages respectively. In MLP classifier we have considered one input layer, four hidden layers(80,50,30,8) and one output layer, with 'stochastic gradient descent' optimizer and tanh activation. In case of LR classifier used multi-class as 'multi-nominal' with 'stochastic average gradient' optimizer. Another classifier SVM , taken multi-class as 'one-vs-rest', with loss as 'squared$_h inge'$.

4.2 Information Retrieval

After translating the user query into english, to get the relevant information, web scraping technique is used to extract the unstructured data on the web into structured form using the"google", "Bing" search engines.To achieve this, Python libraries BeautifulSoup9 is used. And BeautifulSoup is to extract the data from the web pages. To avoid the noise in the data, considered the top 5 URLs with the most relevant context for the query. We also extracted the useful meta information from each URL, which was attached with HTML tags like title, headings, paragraphs etc. After observing all the cases, suitable sentences are taken from multiple documents and stored in one document. To find out the important sentences with respect to query, a ranking methodology(cosine similarity and Sequence matching methods) used. With the rank-based approach, there is a huge scope of obtaining better search speed and avoiding the noisy information at the prediction phase. By using cosine similarity matrix method, we ranked the top K-sentences(better accuracy obtained for K=15) useful for the query to give the accurate answer. In cosine similarity, two vectors are projected in a higher-dimensional space, the similarity is obtained by measuring the dot product between the two vectors.

4.3 Answer Extraction

After identifying which type of answer required for the user query and most useful K-ranked sentences, we do NER tagging for all the K-ranked sentences. NERs applied on each of the top K-ranked sentences to extract the same category of answer type with the help of Spacy10. If more than one possible answers for any particular query was obtained then the first answer will be considered as final answer as because first answer containing sentence having more cosine similarity than others. This final answer is the predicted answer for the user query. At the final step, the answer will be converted to Telugu with the help of Google Translate. The upcoming section demonstrates the experimental results.

5 Experiments and Results

We have implemented this Telugu ODQA system in two ways:
1.Snippet Extraction
2.Paragraph Extraction

5.1 GOOGLE SNIPPET EXTRACTION

we have used 3 classifiers SVM,LR,MLP. For each classifier we have calculatedExact match, partial match for cosine similarity greater than 0.7 and partial matchfor cosine similarity greater than 0.5. All the results are tablulated below:

1.SVM Classifier				
NER Tags	Exact Match	PM(>=0.7)	PM(>=0.5)	QC ACCURACY
PERSON	34	52	52	96
LOCATION	18	22	22	90
ORGANIZATION	12	24	32	84
DATE	16	32	48	84
CARDINAL	20	24	28	92
CURRENCY	26	42	44	90
For 300 queries	21	32.666	37.666	89.333

2.LOGISTIC REGRESSION				
NER Tags	Exact Match	PM(>=0.7)	PM(>=0.5)	QC ACCURACY
PERSON	36	52	52	100
LOCATION	14	18	18	86
ORGANIZATION	8	14	22	68
DATE	14	22	40	70
CARDINAL	22	26	34	96
CURRENCY	26	40	44	70
For 300 queries	20	28.66	35	81.666

3.MLP CLASSIFIER				
NER Tags	Exact Match	PM(>=0.7)	PM(>=0.5)	QC ACCURACY
PERSON	36	52	52	100
LOCATION	14	18	18	85
ORGANIZATION	8	24	40	82
DATE	14	20	40	72
CARDINAL	22	26	34	94
CURRENCY	26	44	48	78
For 300 queries	20	30.666	38.666	85.166

5.2 PARAGRAPH EXTRACTION

1.SVM CLASSIFIER						
	BING			Google		
NER Tags	EM	PM(>=0.7)	PM(>=0.5)	EM	PM(>=0.7)	PM(>=0.5)
PERSON	14	30	38	2	24	40
LOCATION	12	16	16	6	8	14
ORGANIZATION	2	10	18	6	16	18
DATE	10	16	36	6	20	32
CARDINAL	12	16	18	10	14	20
CURRENCY	28	52	60	24	46	52
For 300 queries	13	23.33	31	9	21.333	29.3333

2.LOGISTIC REGRESSION						
	BING			Google		
NER Tags	EM	PM(>=0.7)	PM(>=0.5)	EM	PM(>=0.7)	PM(>=0.5)
PERSON	14	26	38	2	22	36
LOCATION	12	14	14	6	8	14
ORGANIZATION	4	10	18	4	14	18
DATE	6	16	32	6	18	30
CARDINAL	4	10	10	10	16	20
CURRENCY	22	52	54	24	46	46
For 300 queries	9	21.33	27.66	8.666	20.66	22.3333

3.MLP CLASSIFIER						
	BING			Google		
NER Tags	EM	PM(>=0.7)	PM(>=0.5)	EM	PM(>=0.7)	PM(>=0.5)
PERSON	10	28	36	2	24	40
LOCATION	12	16	16	6	8	14
ORGANIZATION	4	10	18	6	16	18
DATE	6	10	30	6	20	32
CARDINAL	4	10	10	10	14	20
CURRENCY	22	52	58	24	46	52
For 300 queries	9.66	21	28	9	21.333	26

5.3 Accuracies of Different Classifier

NER Tags	SVM(linear SVC)	LOGISTIC REGRESSION	MLP
PERSON	98	100	100
LOCATION	90	86	85
ORGANIZATION	84	66	82
DATE	80	70	72
CARDINAL	92	96	94
CURRENCY	90	70	78
For 300 queries	89	81.33	85.166

5.4 Observations on Exact match cases

Exact matches are the cases for which cosine similarity between extracted answer and actual answer is 1.0 for some questions, if one letter of the extracted answer is different from actual answer cosine similarity score is 0. That's why we have used Sequence matcher.
EXAMPLE:
Extracted answer:Edgar snow
Actual answer: Edger snow
for above case cosine similarity is 0 but sequence matching is 0.9

Another case where exact match is troubling: Extracted answer:William Shakespeare Actual answer: Shakespeare These cases making exact match accuracy less.

5.5 Observations on Partial match cases

Partial matches are the cases for which cosine similarity between extracted answer and actual answer is greater than or equal to 0.7// In addition to this we have taken another partial match for which cosine similarity is greater than or equal to 0.5. Example below shows why we have used this:
Question: In which year india got independence?
Exracted anser is: August 15, 1947
Actual answer is: 1947
For these answer the cosine similarity is 0.58 which is less than 0.7. But this answer is not completely wrong, that's why we have include partial match for cosine similarity greater than 0.5 also.

6 Future work

In this project we have implemented QA system by two ways Paragraph Extraction and Snippet Extraction. We have got accurate and faster results with Google Snippet Extraction. But for some queries snippets are not available. so for those queries for which snippets not available if we perform paragraph extraction we can increase

the overall accuracy. In future we combine both snippet extraction and paragraph extraction to increase the accuracy of the system.

7 Conclusions

In this project, we have developed telugu ODQA system for queries having 6 answer types(Person,Location,Organization,Date,Number,Currency). Various experiments were performed for exact and partial matches with different classifiers. We concluded that with SVM classifier giving more accurate and taking less time for training than other classifiers like MLP, logistic regression. Regarding Answer extraction Google snippet extraction giving faster and accurate results than paragraph extraction.

References

Appendices

A REFERENCES

[1] Priyanka Ravva, Ashok Urlana, Manish Shrivasthava.2020. AVADHAN:System for Open-Domain Telugu Question Answering. International Institute Of Information Technology,Hyderabad.LTRC lab.

[2] Zhiping Zheng. 2002. AnswerBus question answering system. In Proceedings of the second international conference on Human Language Technology Research.

[3]] Dell Zhang and Wee Sun Lee. 2003. A web-based question answering system.(2003). Morgan Kaufmann Publishers Inc., 399–404.

[4] Spacy model tutorial from website https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da