AVADHAN: System for Open-Domain Telugu Question Answering

Priyanka Ravva
priyanka.ravva@research.iiit.ac.in
KCIS, LTRC, IIIT-Hyderabad
Hyderabad, Telangana

Ashok Urlana
KCIS, LTRC, IIIT-Hyderabad
Hyderabad, India
ashokurlana@gmail.com

Manish Shrivastava
KCIS, LTRC, IIIT-Hyderabad
Hyderabad, India
m.shrivastava@iiit.ac.in

ABSTRACT

This paper presents the Question Answering (QA) system for a low resource language like 'Telugu' named 'AVADHAN'. This work started with preparing a pre-tagged data set for Telugu Question Classification (QC). We also explained the ambiguities and complexities involved in the data set. AVADHAN exhibits the comparisons between Support Vector Machine (SVM), Logistic Regression (LR) and Multi-Layer Perceptron (MLP) classifiers for achieving the plausible answers. After performing various experiments the overall accuracies obtained, for both 'exact match' and 'partial match' based approaches, were for SVM (31.6%, 68.5%), LR (31%, 66.6%) and for MLP (30%, 67%) respectively.

CCS CONCEPTS

• **Information systems → Web search engines**; *Web crawling*; Content ranking.

KEYWORDS

Telugu QA, Neural networks, Crawling

ACM Reference Format:
Priyanka Ravva, Ashok Urlana, and Manish Shrivastava. 2020. AVADHAN: System for Open-Domain Telugu Question Answering. In *7th ACM IKDD CoDS and 25th COMAD (CoDS COMAD 2020), January 5–7, 2020, Hyderabad, India.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3371158.3371193

1 INTRODUCTION

Languages play an important role in communication whether it is among humans or machines. Around 6500 languages[1] 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. As per our best knowledge, 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. This fact motivated us to choose the Telugu QA task with a pre-processed data set of 1037 Queries. Any Natural Language Processing (NLP) task for low resource languages can be explored by adapting implementations from resource-rich languages. In QA, some advanced information retrieval technologies [6], [4] are being used to extract text from all documents and some models only concentrate on extracting the essential information. Therefore, any QA system should be built with these two techniques (Information retrieval and extraction). Always the core aim of QA lies on the extraction of suitable answers only, not all the related documents to the query.

Many automatic QA systems [18], [16] have been developed over the past few decades. Initially, the rule-based approaches were used as a prominent method to do the question answering [5] 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 [1], [8] 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 [7] 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 [3], [15], are applied to knowledge-based question answering to predict the correct answer for user query.

Keeping these challenges in mind for low resource languages, we attempt to build a QA model 'AVADHAN' for a low resource language Telugu. One of the prime modules of QA system is Question Classifier (QC) which is not available for Telugu. To the best of our knowledge there is no pre-tagged data (query categories) available for Telugu, which makes Telugu QC is a challenging task. 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 AVADHAN obtained the classification accuracies of 73%, 72%, 71% respectively. AVADHAN deals with PERSON, LOCATION, ORGANIZATION, DATE, TIME, NUMBER, PERCENTAGE kind of answer type categories. The next section demonstrates the different kinds of state-of-the-art models related to QA.

2 RELATED WORK

Most of the information retrieval technologies are stimulated from TREC (Text REtrieval Conferences) series since 1999. One of the works mentioned by Zheng et al. [20] is a web-based QA

[1] https://en.wikipedia.org/wiki/Languages_of_India

CoDS COMAD 2020, January 5–7, 2020, Hyderabad, India

ACM ISBN 978-1-4503-7738-6/20/01…$15.00
https://doi.org/10.1145/3371158.3371193

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 TREC-8's data set of 200 queries is 70.5%.

A state-of-the-art model LAMP [19] related to the web-based question answering system, which only takes the snippets returned by a particular search engine like Google. LAMP reduces the time-consumption to retrieve the answer compared to other state-of-the-art models. This research group collected snippets from the top 100 Google results for each query and applied a score function to retrieve the valid answers, also obtained the Confidence Weighted Score (CWS) of 0.60 and the average Mean Reciprocal Rank (MRR) score of 0.47. Using the data set of 60 queries, [12] performed QA task for Hindi language and particularly concentrated on 'where', 'when', 'how many' and 'what time' type of queries. These authors initially translated the Hindi language into Query Logic Language [17] using a tool named as BabelFish (AltaVista's translation tool), which gave an accuracy of 68%. Specifically for Telugu, Bandyopadhyay et al. [11] 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% and precision of 96.34%.

Some other researchers [10] worked on factoid based QA systems by separating the data for only 5 types of categories (Location, Person, Name, Date, Others). While testing with TREC (1999-2003) factoid data, obtained the accuracy of 62.11%. To pick the exact answers from the extracted information, these authors [9] used paragraph indexing. One more question answering system 'LASSO' was proposed by using the paragraph indexing [9] to search for an answer by doing paragraph filtering and ordering. The next section describes the investigation of the Telugu data set.

3 CORPUS ANALYSIS

Before implementing the model, we need a labeled data set. Preparing a data set for Telugu, a low resource language, more human intervention, mixed with various intuitions and analogies are required to reduce ambiguity.

3.1 Ambiguity in Telugu Dataset

The data set preparation was started in August 2017, all the query and answers used to develop the model contains data based on these websites[2] [3] [4]. After that, pre-processing and labeling of corpus was done with the help of three annotators and obtained a Fleiss'[5] kappa score of 0.85. The labeling of the data set was performed completely based on the answer type (Person, Location, Number, Organization, Time, Date, Percentage) related to the query. The main reason behind mentioning the exact time for the corpus preparation lies in answering dynamic queries or queries dependent on time. While preparing this kind of a QA data set, many times, there is a possibility of change in target answers with respect to time (like presidents of the countries, sports records, population records, etc). To overcome this drawback we are using the static data set of 1037 Telugu queries. We release the data set and code at *https://github.com/priyanka-ravva/Telugu-Question-Answering*.

Based on the answer type overall data set is divided into seven major categories, those are PERSON, ORGANIZATION, DATE, LOCATION, NUMBER, TIME and PERCENTAGE. Some times in Telugu a single word can fall in more than one category like 'యమునా' (Yamuna). Which can be either a person name or a river name as per the context. So, to avoid this confusion, we come across with many special category types of queries. Some of them mentioned in Table 1 and created a data set of a special category, which consists of 100 Telugu queries which do not belong to above mentioned seven categories. As mentioned in Table 1, the first query signifies one such scenario of more than one possible answers and the second one deals with a miscellaneous query, which is rhetorical[6] (which needs some discussion to decide the exact answer), the third query in the table is a time-bounded one, in which the exact answer changes with respect to time. The following section explains the full details of AVADHAN.

4 MODEL

The model implementation starts with the given input user query, which is in Telugu. The user input needs to be translated into English to extract the relevant information in a search engine. To obtain that, Python packages urllib2[7] and Google translate API[8] were used. As shown in Fig. 1, AVADHAN has mainly 3 modules, those are information retrieval, question classification and answer extraction.

4.1 Information Retrieval

After translating the user query, to get the relevant information, web scraping technique is used to extract the unstructured data on the web into structured form using the "Bing" search engine. To achieve this, Python libraries BeautifulSoup[9], urllib2 were used. Precisely, urllib2 is used for fetching Uniform Resource Identifier (URLs) and BeautifulSoup is to extract the data from the web pages. To avoid the noise in the data, considered the top 10 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 headings, paragraphs, tables, and images, etc. If any incomplete sentences or meaningless heading were found, they were ignored. 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 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 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. The relevance between

[2]https://mympsc.com/appsc
[3]https://upscgk.com/APPSC-gk
[4]http://services.indg.in/online_quiz/index_te.php
[5]https://en.wikipedia.org/wiki/Fleiss%27_kappa
[6]https://literarydevices.net/rhetorical-question/
[7]https://docs.python.org/3/howto/urllib2.html
[8]https://py-googletrans.readthedocs.io/en/latest/
[9]https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Type of query	Example
More than one answer possible query	2014 లో నోబెల్ శాంతి బహుమతి ఎవరికి వచ్చింది? (who got nobel peace prize in 2014?) Answers: కైలాష్ సత్యార్థి, మలాలా యూసఫ్‌జాయ్ (Kailash Satyarthi, Malala Yousafzai)
Miscellaneous type Query	మీరు సరిగ్గా ఏమీ చేయలేరా? (Can't you do anything right?) Answer: Rhetorical question
Time bounded Query	హాంగ్ కాంగ్ యొక్క ప్రస్తుత జనాభా ఎంత? (what is the current population of hong kong?) Answer: 73.9 లక్షలు (73.9 lakhs)

Table 1: Special types of Queries based on answers

two sentences is obtained based on the measured angle. Similarity and angle are inversely proportional to each other. Smaller the angle, higher the similarity and vice versa.

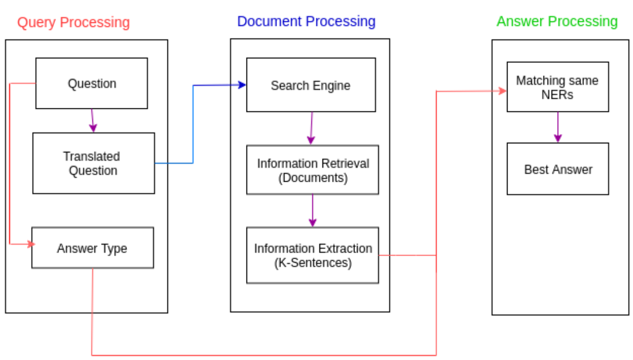

Figure 1: AVADHAN Architecture

4.2 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, Time, Date, Number, Percentage). This predicted classes will play a major role in finding the answers accurately. So we need a Question Classifier (QC) which is not available for Telugu. Modeling a 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 term-frequency (*tf*) and inverse document frequency (*idf*).

In MLP classifier we have considered one input layer, four hidden layers 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_hinge'. Pre-tagged data divided into train and test data like 725, 312 respectively and classifiers (MLP, LR, SVM) accuracies are 71%, 72%, 73%, respectively.

4.3 Answer Extraction

After attaining answer type and most useful K-ranked sentences by using QC, we identified the which type of answer required for the user query. NERs [14], [2], [13] are popularly used in information extraction to recognize the named entities and categorize into various predefined classes. NERs applied on each of the top K-ranked sentences to extract the same category of answer type with the help of Spacy[10]. If more than one possible answers for any particular query was obtained then by using the frequency of occurrence, the best answer will be selected. Which 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

Experiments performed with 1037 pre-processed Telugu queries of which more than 50% of were related to LOCATION, PERSON and NUMBER categories. Data set split into train and test queries as 725 and 312 respectively. To get an intuition on "how many sentences are essentially required to answer a query ? ", initial experiments were performed with varying the number of sentences (K) for fetching the correct answer.

As shown in Table 2 for K = 40, better accuracies were observed as compared to other K values. Considering very less number of sentences may increase the chances of ties among candidate answers because we are following frequency based answer extraction. Gradually increasing K upto 40 we observed better predictions. Having very large K values may give irrelevant sentences, eventually decreasing the accuracies. Hence all further experiments were conducted with K=40 sentences.

[10]https://spacy.io/usage/linguistic-features#named-entities

Classifiers		K=10	K=20	K=30	**K=40**
SVM	EM	21.9	25.9	27.8	**31.6**
	PM	46.7	58.4	61.5	**68.5**
LR	EM	21.1	25.7	29.4	**31**
	PM	46.3	57.5	63.6	**66.6**
MLP	EM	21.5	24.5	29.4	**30**
	PM	47.3	57.4	62.8	**67**

Table 2: Overall performance of AVADHAN (in terms of %) by varying the number of sentences, EM: Exact match, PM: Partial match

Category	Number of samples	Exact match			Partial match		
		C1	C2	C3	C1	C2	C3
Location	353	34.6	35.7	36.4	70	72	72.3
Person	317	18.9	20.8	20.5	68.8	66.4	68.5
Number	170	44.1	44.7	42.6	58.2	59.4	56.8
Date	125	28	32.8	37.2	80	76	87.6
Organization	29	27.6	17.2	20	37.9	31	26.7
Percentage	25	28	26.1	24	40	39.1	48
Time	18	22.2	5.6	18.8	55.6	61.1	75
Overall	**1037**	**30**	**31**	**31.6**	**67**	**66.6**	**68.5**

Table 3: Accuracy (in terms of %) of AVADHAN for individual categories with K=40

5.1 Observations on Exact Match Cases:-

An exact match of the query, looks for an exact match with the specific answer. Most of the times obtaining the correct answer in Telugu context might be a difficult task for some category of queries, which might hamper its accuracy. As mentioned in the Table 3 experiments are conducted with respect to different classifiers to compare the accuracies, in the Table 3, C1, C2, C3 denotes MLP, LR and SVM classifiers respectively. By the normal observation of exact match case, it is known that in case of all the three classifiers, particularly for TIME, PERSON and ORGANIZATION categories the accuracy percentage is very low because of the uncertainty involved in the answer context. As mentioned in the below example for PERSON category a very small difference in context, makes it unsuitable for exact match answer.

"ఆలిస్ ఇన్ వండర్ల్యాండ్" - పుస్తక రచయిత ఎవరు? ("Alice in Wonderland" - Who is the author of the book?)

Predicted answer: లూయిస్ కారోల్ (Louis Carroll)

Correct answer: లెవిస్ కారోల్ (Lewis Carroll)

Because of the irregularity, prefixes and suffixes added to the correct answers or because of the affiliations added to the NAME category, accuracy decreases. As specified in the below example, there is a possibility of occurrence of more than one possible answer for the same query, then it is difficult to tag a particular outcome as an exact result.

భారతదేశం యొక్క క్షిపణి మనిషి ఎవరు? (who is the missile man of India?)

Possible Correct answers:- 1. అవూల్ పకీర్ జైనులాబ్దీన్ అబ్దుల్ కలాం (Avul Pakir Jainulabdeen Abdul Kalam), 2. ఎ. పి. జె. అబ్దుల్ కలాం (A. P. J. Abdul Kalam), 3. డాక్టర్. అబ్దుల్ కలాం (Doctor. Abdul Kalam)

Another prevalent issue that is seen in Telugu is, the choice of word changes with nativity, social distance and the social status (relation) between the speakers, which also decreases the accuracy. Below three statements have the same English conversion, but in Telugu, those three can be used in different contexts.

(1) ఎంత సొమ్ము కావాలి? (How Much Money Do You Need?)
(2) ఎంత డబ్బు కావాలి? (How Much Money Do You Need?)
(3) ఎంత ధనము కావాలి? (How Much Money Do You Need?)

The above mentioned issues are the major reason for obtaining less accuracy in Telugu question answering task. In the next section, experiments were performed using partial match approaches.

5.2 Observations on Partial Match Approach:-

In partial match, by fixing some threshold value, prediction of the final answer is decided. In our experiments, the threshold was fixed as 0.7. Precisely, if the predicted answer has partial-match-score more than or equal to the threshold value, then it is considered as plausible answer. Because of this reason partial match methods give better results compared to exact match cases. To find the partial match, we used the SequenceMatcher[11] library in Python. In this approach also ORGANIZATION and PERCENTAGE answer categories produce low accuracy because of the ambiguity in data set related to that category.

Finally, the overall accuracy of SVM is slightly better than both MLP and LR classifiers, as MLP and LR works better with a very large data set. Since, we are using less than 1500 queries data (because of data scarcity) MLP and LR are not able to produce better results. SVM performs better for lots of features that are naturally on the same scale and fewer data points. It also uses a subset of training points as support vectors, these are used to get the clear margin of separation between two classes. SVMs are efficient and generates a best classification as they obtain the optimum separating surface, which has a good performance on previously unseen data points. Overall the MLP, LR, SVM classifiers obtains the accuries of (30% , 67%), (31%, 66.6%) and (31.6%, 68.5%) for both exact match and partial match scenarios.

AVADHAN model returns a ranked list of top possible answers (N), which are most relevant to the query, and each query has its own ground-truth answer. By observing all the above approaches AVADHAN performs better on LOCATION, NUMBER, DATE categories.

6 CONCLUSIONS AND PROSPECTIVE WORKS

As an initial move towards Telugu QA, this paper broadly explained the perplexities involved in the Telugu data set and also demonstrated various kind of query categories based on the resulting answer. By measuring the number of sentences required for information extraction, various experiments were performed for exact and partial matches with different classifiers. We concluded that with SVM classifier, AVADHAN produces better accuracy compared to MLP and LR. Alternatively, to reduce the time consumption for retrieving the best answer, we can directly take the Google snippets data. There is scope to apply AVADHAN on TREC data set. In future, AVADHAN could also be expanded to multilingual open domain question answering with both static and dynamic data sets.

[11] https://docs.python.org/2.4/lib/sequencematcher-examples.html

REFERENCES

[1] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research* 4 (1996), 129–145.

[2] Asif Ekbal, Rejwanul Haque, Amitava Das, Venkateswarlu Poka, and Sivaji Bandyopadhyay. 2008. Language independent named entity recognition in indian languages. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

[3] Sherzod Hakimov, Soufian Jebbara, and Philipp Cimiano. 2018. Deep Learning Approaches for Question Answering on Knowledge Bases: an evaluation of architectural design choices. *arXiv preprint arXiv:1812.02536* (2018).

[4] Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus, and Paul Morarescu. 2000. Falcon: Boosting knowledge for answer engines. In *TREC*, Vol. 9. 479–488.

[5] KSD Ishwari, AKRR Aneeze, S Sudheesan, HJDA Karunaratne, A Nugaliyadde, and Y Mallawarrachchi. 2019. Advances in Natural Language Question Answering: A Review. *arXiv preprint arXiv:1904.05276* (2019).

[6] Cody Kwok, Oren Etzioni, and Daniel S Weld. 2001. Scaling question answering to the web. *ACM Transactions on Information Systems (TOIS)* 19, 3 (2001), 242–262.

[7] Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1–7.

[8] Abraham Ittycheriah Martin, Martin Franz, and Salim Roukos. 2001. Ibm's statistical question answering system-trec-10. In *In Proceedings of the Tenth Text REtrieval Conference (TREC*. Citeseer.

[9] Dan Moldovan, Sanda Harabagiu, Marius Pasca, Rada Mihalcea, Roxana Girju, Richard Goodrum, and Vasile Rus. 2000. The structure and performance of an open-domain question answering system. In *Proceedings of the 38th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 563–570.

[10] Prakash Ranjan and Rakesh Chandra Balabantaray. 2016. Question answering system for factoid based question. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. IEEE, 221–224.

[11] Rami Reddy Nandi Reddy and Sivaji Bandyopadhyay. 2006. Dialogue based question answering system in telugu. In *Proceedings of the Workshop on Multilingual Question Answering*. 53–60.

[12] Shriya Sahu, Nandkishor Vasnik, and Devshri Roy. 2012. Prashnottar: a Hindi question answering system. *International Journal of Computer Science & Information Technology* 4, 2 (2012), 149.

[13] B Sasidhar, PM Yohan, A Vinaya Babu, and A Govardhan. 2011. A survey on named entity recognition in Indian languages with particular reference to Telugu. *International Journal of Computer Science Issues (IJCSI)* 8, 2 (2011), 438.

[14] Praneeth M Shishtla, Karthik Gali, Prasad Pingali, and Vasudeva Varma. 2008. Experiments in Telugu NER: A conditional random field approach. In *Proceedings of the IJCNLP-08 Workshop on Named Entity Recognition for South and South East Asian Languages*.

[15] Daniil Sorokin and Iryna Gurevych. 2018. Modeling semantics with gated graph neural networks for knowledge base question answering. *arXiv preprint arXiv:1808.04126* (2018).

[16] Manuel E Sucunuta and Guido E Riofrio. 2010. Architecture of a question-answering system for a specific repository of documents. In *2010 2nd International Conference on Software Technology and Engineering*, Vol. 2. IEEE, V2–12.

[17] Maria Vargas-Vera, Enrico Motta, and John Domingue. 2003. AQUA: An Ontology-Driven Question Answering System. *New Directions in Question Answering* 8 (2003).

[18] Caixia Yuan and Cong Wang. 2005. Parsing model for answer extraction in Chinese question answering system. In *2005 International Conference on Natural Language Processing and Knowledge Engineering*. IEEE, 238–243.

[19] Dell Zhang and Wee Sun Lee. 2003. A web-based question answering system. (2003).

[20] Zhiping Zheng. 2002. AnswerBus question answering system. In *Proceedings of the second international conference on Human Language Technology Research*. Morgan Kaufmann Publishers Inc., 399–404.