Open Domain Question Answering System Using Wikipedia

Preetam B. A.[1], Ria Singh[2], Prince Zalavadia[3], Dr. Sharvari Govilkar[4]
[1,2,3]*Student, Department of Information Technology, Pillai College of Engineering, New Panvel*
[4] *H.O.D, Department of Computer Engineering, Pillai College of Engineering, New Panvel*

***Abstract*— The open-domain question answering task has recently been addressed using unstructured data such as websites and online encyclopedias. Here, open-domain questions are answered by making full use of knowledge sources of Wikipedia via its API for many types of questions, it is critical to analyze user questions in terms of the nature of the answers being sought. The analyzed result of a question has three components: Answer Format, Answer Theme and Question Target (question analysis). The next step involves finding the most relevant documents or passages related to the question using either word embedding distances or Deep Learning (document retrieval). Finally, the answers are extracted from the passage (machine comprehension). PageRank technique can be used while computing the "document score" to assess relevance of a document to a query. BERT or ALBERT architecture (document reader trained on SQuAD 2.0 dataset) can be used which will dramatically improve performance for machine comprehension. The Answer Ranker (SoftMax function) extracts the 1-2 lines of answer for the query. The question provided and the answer generated are in audio format using the SpeechRecognition library and gTTS API.**

Index Terms— Question Answering System, Open Domain, Wikipedia, Deep Learning, Natural Language Processing, Document Retrieval, Document Reader, Answer Ranker.

I. INTRODUCTION

Question answering (QA) is a computer science discipline within the fields of information retrieval and natural language processing (NLP), which is concerned with building systems that automatically answer questions posed by humans in a natural language. Open-domain question answering deals with questions about nearly anything and can only rely on general ontologies and world knowledge. In information retrieval, an open domain question answering system aims at returning an answer in response to the user's question. The returned answer is in the form of short texts rather than a list of relevant documents. The system uses a combination of techniques from computational linguistics, information retrieval and knowledge representation for finding answers.

Keyword extraction is the first step for identifying the input question type by finding the words that can indicate the meaning of the question (‘Who’, ‘What’, ‘How’, ‘When’, ‘Where’, ‘Why’). A lexical dictionary such as WordNet can then be used for understanding the context. An information retrieval system is used to find a set of documents containing the correct key words. A tagger and NP/Verb Group chunker can be used for it. Only the relevant paragraphs are selected for ranking. A vector space model can be used as a strategy for classifying the candidate answers. Check if the answer is of the correct type as determined in the question type analysis stage. An inference technique can also be used to validate the candidate answers. A score is then given to each of these candidates and the answer is then translated into a compact and meaningful representation by parsing.

Our proposed model works towards answering open-domain questions by making full use of Wikipedia for many types of questions, analysis of the query input by the user followed by its classification and reformulation based on answer format, answer theme and question target, information retrieval, filtering and ordering by ranking of passages and deep learning approaches and answer identification, extraction and validation.

The major issues with the Open Domain Question Answering System are that the accuracy of the results obtained by them is quite less than the Closed Domain Question Answering System. The context with higher token length performs poorly as compared to those with lower token lengths. Wikipedia API sometimes provides with the articles irrelevant to the question asked possibly due to the ambiguity and the time taken to retrieve the answer in the Open Domain Question answering System is higher than the Closed Domain Question Answering System.

II. LITERATURE SURVEY

A. Literature Survey on Question Answering System

Ali Allam and Mohamed H. Haggag et.al. [1] compared each research against the others with respect to the components that were covered and the approaches that were followed. The QA models used for this purpose were: BASEBALL, LUNAR, PLANES, LIFER, SYNTHEX. This research involved the use of many techniques and algorithms for question classification and for QA systems: QUALM/NSIR/TREC-8/TREC-10, Rule based/Machine learning. Sanjay K Dwivedia and Vaishali Singh et.al. [2] proposed 3 classifications for characterizing QA approaches in terms of linguistic, statistical and pattern matching approach. Linguistic approach was presented using BASEBALL/LUNAR/ELIZA/GUS model, Clark et al./Chung et al. and Mishra et al./START/Quarc/Cquark model. For statistical approach SVM/Bayes/Rocchio classifiers, N-gram mining/Sentence similarity models/Okapi similarity measurement were used.

B. Literature Survey on Open Domain QA System

Zhonglin Ye, Zhen Jia, Yan Yang, Hongfeng Yin et.al. [4] focused on algorithms for the extraction of answers of a query. SPE algorithm is presented to find answers from the knowledge base, WKE algorithm is used to extract answers from search engine query results, SWJTU Chinese Word Segmentation System/CRF algorithm for subject recognition, Baidu Zhixin for knowledge extraction, etc. Vivek Datla, Sadid A. Hasan, Joey Liu, Yassine Benajiba Kathy Lee, Ashequl Qadir, Aaditya Prakash and Oladimeji Farri et.al. [5] dealt with Live-QA task which involves real user questions extracted from the stream of most recent questions submitted to the Yahoo Answers (YA) site. The systems are needed to provide an answer which is less than 1000 characters length in less than 60 seconds. Ibrahim Alturani et.al. [8] proposed a new approach architecture for open domain question-answering systems depending on the ontology and wordnet to improve answer accuracy.

C. Literature Survey on Open Domain QA System using Wikipedia

Pum-Mo Ryu and Hyunki Kim et.al. [3] described the use of Wikipedia as a knowledge source. Multiple answer matching modules were used based on different types of semi-structured knowledge sources of Wikipedia, including article content, info boxes, article structure, category structure, and definitions. Danqi Chen, Adam Fisch, Jason Weston and Antoine Bordes et.al. [7] dealt with the task of machine reading at scale (MRS) which combines the challenges of document retrieval (finding the relevant articles) with that of machine comprehension of text (identifying the answer spans from those articles). Rohan Sampath and Puyang Ma et.al. [9] proposed to extend the DrQA implementation for both document retrieval and machine comprehension which uses finding Wikipedia articles relevant to the question (document retrieval) and Predicting an answer span within the retrieved articles (machine comprehension of text).

D. Literature Survey on Techniques / Models / Architecture for Machine Comprehension

Natural Language Computing Group, Microsoft Research Asia et.al. [6] introduced R-net, an end-to-end neural networks model for reading comprehension style question answering, which aims to answer questions from a given passage. The question and passage were matched to obtain the question-aware passage representation. Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova et.al. [10] used pretraining deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers and demonstrating the importance of bidirectional pre-training for language representations. Zhuosheng Zhang1, Junjie Yang and Hai Zhao et.al. [10] researched about MRC. Machine reading comprehension (MRC) is an AI challenge that requires machines to determine the correct answers to questions based on a given passage.

E. Summary of Literature Survey

Literature survey has been done rigorously in four major types of paper like Question Answering System (Q-A), Open Domain Q-A System, Open Domain Q-A System using Wikipedia and Techniques / Models /

Architecture for Machine Comprehension from 2012 to 2020. The papers suggest that the results aren't as substantial with questions having longer answers. Accuracy of the results given in the Open Domain Question Answering System are less accurate than the Closed Domain Question Answering System. There's a substantial amount of work that can be done with the decoders and verifiers to increase the performance of the Machine Comprehension Models.

III. PROPOSED SYSTEM

Here we would be discussing the proposed system. In order to achieve better domain results, we propose the following system as solution:

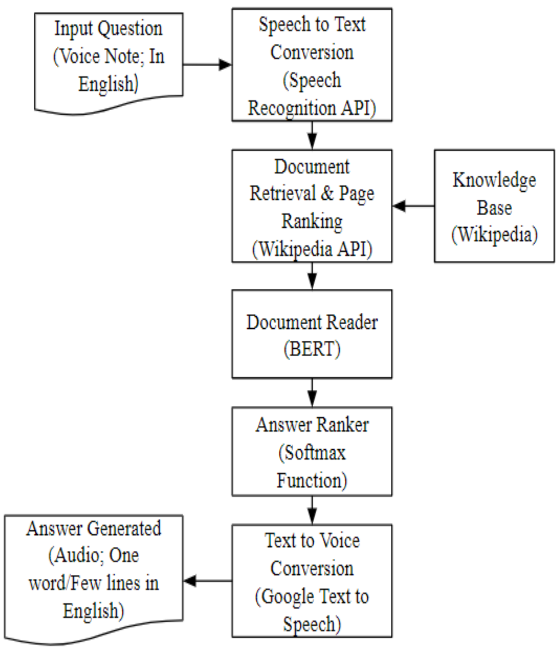

Fig 3.1 Open Domain QA System

A. Input Question

An Input Question in the Audio format will be recorded. A library called ffmpeg is used. ffmpeg is extremely powerful, but its command-line interface gets really complicated rather quickly - especially when working with signal graphs and doing anything more than trivial. The audio signals thus received through microphone are converted into .wav format for further processing. The question can also be entered in textual format in English.

B. Speech to Text Conversion

Speech is the most common means of communication and the majority of the population in the world relies on speech to communicate with one another. Speech recognition systems basically translate spoken languages into text. There are various real-life examples of speech recognition systems. For example, Apple SIRI which recognizes the speech and truncates into text. The audio is further processed to convert it into text. For this SpeechRecognition library is used.

1. *SpeechRecognition:* SpeechRecognition is a library for performing speech recognition function, with support for several engines and APIs, online and offline. For this project, Google Speech Recognition API is used. The module thus converts the Audio Question into Text Format and that text can be used by further modules.

C. Document Retrieval and Page Ranking

1. *Document Retrieval:* Document Retrieval is the computerized process of producing a list of documents that are relevant to an inquirer's request by comparing the user's request to an automatically produced index of the textual content of documents in the system. These documents can then be accessed for use within it. Nearly everyone today uses Document Retrieval systems, although they may not refer to them as such, but rather as Web-based search engines.
2. *Page Ranking:* Page Ranking is the process of ranking the pages in the order of their relevance to the query. It's a process of arranging the pages or documents in the descending order of their relevance with the most relevant page on the top of the list. Top 5 Pages can be used from the list to extract the relevant data to process the corresponding query.
3. *Wikipedia API:* Data scraping has seen a rapid surge owing to the increasing use of data analytics and machine learning tools. The Internet is the single largest source of information, and therefore it is important to know how to fetch data from various sources. And with Wikipedia being one of the largest and most popular sources for information on the Internet, this is a natural place to start. With the Wikipedia API, we can programmatically pull the information right from Wikipedia, and format it to fit our project, app, or website nicely - only showing the information that we want. It is used for extracting data from the

web and to get a variety of information such as a page's title, category, links, images, and retrieve articles based on geo-locations. Thus, it can be used for Document Retrieval and Page Ranking.

D. Document Reader

After the successful selection of relevant documents or passages, the next step is to extract the answer. The prime purpose of the document reader is to apply reading comprehension algorithms to text segments for answer extraction. Modern reading comprehension algorithms come in two broad flavors: feature-based and neural-based. Feature-based answer extraction can include rule-based templates, regex pattern matching, or a suite of NLP models (such as parts-of-speech tagging and named entity recognition) designed to identify features that will allow a supervised learning algorithm to determine whether a span of text contains the answer. Neural-based reading comprehension approaches capitalize on the idea that the question and the answer are semantically similar. Rather than relying on keywords, these methods use extensive datasets that allow the model to learn semantic embeddings for the question and the passage. Similarity functions on these embeddings provide answer extraction.

1. *BERT:* BERT stands for Bidirectional Encoder Representations from Transformers. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create a state-of-the-art model for a wide range of NLP tasks. BERT is based on the Transformer architecture. BERT is pre-trained on a large corpus of unlabeled text including the entire Wikipedia and Book Corpus (800 million words). This pre-training step is half the magic behind BERT's success. This is because as we train a model on a large text corpus, our model starts to pick up the deeper and intimate understandings of how the language works. This knowledge is the swiss army knife that is useful for almost any NLP task. Bidirectional means that BERT learns information from both the left and the right side of a token's context during the training phase. We can fine-tune it by adding just a couple of additional output layers to create state-of-the-art models for a variety of NLP.
2. *SQuAD 2.0:* SQuAD (Stanford Question Answering Dataset) is a dataset on which we train a learning model to answer questions from a given comprehension. SQuAD was created out of crowdsourcing, where crowd workers were posed questions on Wikipedia articles for which the answers were answerable from the comprehension. Learning algorithms can then be trained on this model and then could be tested. This dataset has been extensively used in the research community for evaluation purposes. The 2.0 version of the large-scale dataset Stanford Question Answering Dataset allows researchers to design AI models for reading comprehension tasks under challenging constraints. Thus, a BERT pretrained language model which can be fine-tuned on SQuAD 2.0 datasets, can be used to identify the span of the answer for the given question from the document.

E. Answer Ranker

The BERT model gives the start_scores and end_scores for each word in the information source. The start_scores denote the measure of that word being the starting span of the answer while end_scores denote the measure of that word being the ending span of the information source.

1. *SoftMax Function:* The softmax function is a function that turns a vector of K real values into a vector of K real values that sum to 1. The input values can be positive, negative, zero, or greater than one, but the softmax transforms them into values between 0 and 1, so that they can be interpreted as probabilities. If one of the inputs is small or negative, the softmax turns it into a small probability, and if an input is large, then it turns it into a large probability, but it will always remain between 0 and 1. The softmax function is sometimes called the soft argmax function, or multi-class logistic regression. It can be used in a classifier only when the classes are mutually exclusive. The softmax formulaic as follows:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad (1)$$

Thus, the softmax equation (1) can be used to generate the scores. The Start and End tokens of the information or the context can be taken by considering maximum of each. Thus, the most probable answer span can be obtained from the context for the corresponding.

F. Text to Voice Conversion

A text-to-speech (TTS) system converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. Synthesized speech can be created by concatenating pieces of recorded speech. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

1. *Google Translate text-to-speech API:* There are several APIs available to convert text to speech in python. One of such APIs is the Google Text to Speech API commonly known as the gTTS API. gTTS is a very easy to use tool which converts the text entered, into audio which can be saved as a mp3 file. The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow. gTTS gives a better solution and thus can be used to convert the textual answer into the Audio file.

G. Answer

The answer thus generated through the above modules is finally converted into audio format. The saved audio file can be played through any media player to get the output in audio. Thus, a complete voice-based approach is used, right from taking questions as audio input to answering that question through voice. Along with the audio output the answer in Textual Format is also displayed in the output. This marks the end of the question answering system.

IV. INPUT / OUTPUT DETAILS

Some sample input questions provided to the system as well as its associated answer produced as output by the system as a part of the testing and evaluation process is as follows. The below table contains an example of questions being answered correctly, incorrectly and neutrally in each and every domain respectively.

Table- I: Input / Output details

Domain	Question	Answer
Education	What is force ?	a vector quantity
	Who is the father of computer ?	Henry Babbage
	what is the length of human DNA ?	3 . 1 billion base pairs
	When was atom discovered ?	Couldn't Find the Answer!!!
Technology	Name of founder of Facebook	Mark Zuckerberg
	How does a fuse work ?	Couldn't Find the Answer!!!
	when was domain name registration free ?	\
	what is the full form of Yahoo ?	Couldn't Find the Answer!!!
Entertainment	When was shutter island released ?	February 19 , 2010
	How many awards did Life of Pi film win ?	16
	American Psycho was inspired by which actor ?	Patrick Bateman
	which is the famous dialogue of Liam Neeson ?	Couldn't Find the Answer!!!
Politics	who is the president of USA ?	Joe Biden
	who is Hitler ?	Couldn't Find the Answer!!!
	what did the black man George LFoyd shout before he was killed by the police ?	knelt on Floyd ' s neck
	which Union territory has a high court of its own ?	Couldn't Find the Answer!!!
Sports	where was cricket originated ?	South East England
	who is the inventor of volleyball?	Dale Callaghan
	what is the national sport of India ?	Kabaddi
	who is the winner of Premier League trophy ?	Couldn't Find the Answer!!!
Health & Nutrition	how is leptin made ?	adipose cells and enterocytes
	what is mental health ?	a disorder of the entire body
	what are nutrients ?	energy sources

	where is the highest blood flow in our body ?	Couldn't Find the Answer!!!

V. RESULT ANALYSIS

We have tested our model with different questions and the working of our model has been explained thoroughly.

A. Taking Input

The audio input is recorded through the microphone as the program is executed. The User is then supposed to ask the question orally. After the completion of asking the question, the user is asked to stop the recording by clicking on the button.

B. Speech Recognition

After successful recording of the audio question, the audio is converted into text for further processing. The question thus asked during the demonstration was “Why is the sky blue?” which is correctly recognized using the Speech Recognition module.

C. Document Retrieval and Page Ranking

Wikipedia API is later used to find the relevant articles by performing a search on the API. The results thus obtained are the articles are the ones that are relevant to the question and are sorted in their order of relevance. The Top 5 articles are displayed in the output. The most relevant article is further used as context for the BERT model.

D. Answer

The question and the context are provided as input to the BERT model along with adequate tokenizations. The output thus obtained is the most probable span of the context. The span is displayed in the output. Thus, the answer for the question is “Rayleigh Scattering” and is displayed correctly.

E. Audio Output

The textual answer is converted into speech using gTTS API and the audio file thus generated is played using an audio player. Hence the output answer is obtained.

F. Performance Evaluation

The total domains tested were (6): Education, Technology, Entertainment, Politics, Sports, Health & Nutrition. Total questions asked as part of testing were: Phase I - 201 (Minimum 33 questions from each domain) and Phase II - 300 (50 questions from each domain). The system is measured using metrics such as precision (P), recall (R), accuracy (A), exact match (EM) & F1-Score (FS).

Table- II: Performance evaluation metrics

Domain	P	R	A	EM	FS
Education	0.8000	0.9474	0.7800	36	0.8675
Technology	0.7872	0.9737	0.7800	37	0.8706
Entertainment	0.7826	0.9474	0.7600	36	0.8571
Politics	0.7234	0.9714	0.7200	34	0.8293
Sports	0.7021	0.9429	0.6800	33	0.8049
Health & Nutrition	0.8000	0.9730	0.8000	36	0.8780
Total	0.7653	0.9593	0.7533	212	0.8514

1. *Precision:* It is the measure of exactness, which determines the fraction of relevant items retrieved out of all items. Precision (P) is the proportion of correct answers that were truly correct.

$$P = \frac{TP}{TP + FP} \quad (2)$$

In our project from eq (2),
TP = 212, FP = 65
Precision= 212/ (212+65) =0.7653

2. *Recall:* It is a measure of completeness, which determines the fraction of relevant items retrieved out of all relevant items. It is the proportion of all proper answers.

$$R = \frac{TP}{TP + FN} \quad (3)$$

In our project from eq (3),
TP = 212, FN= 9
Recall = 212/ (212+9) = 0.9593

3. *Accuracy:* It is the ratio of number of correct predictions to the total number of input samples.

$$A = \frac{TN + TP}{TN + TP + FP + FN} \quad (4)$$

In our project from eq (4),
TP = 212, FP = 65, TN = 14, FN = 9
Accuracy = (212+14)/ (212+65+14+9) = 0.7533

4. *Exact Match:* It is a measure of the characters of the model's prediction which exactly match the characters of (one of) the True Answer.

$$EM = \text{Correctly Answered Questions} \quad (5)$$

In our project from eq (5),
Exact Match = 212

5. *F1-Score:* It is defined as the HM (harmonic mean) of recall(R) and precision (P).

$$FS = \frac{2 * Precision * Recall}{Precision + Recall} \quad (6)$$

In our project from eq (6),
Precision = 0.7653, Recall = 0.9593
F1-Score = (2*0.7653*0.9593)/(0.7653+0.9593) = 0.8514

6. *Trigger Word Confusion Matrix:* This is the matrix which portrays the classification of the Answer Type (AT) as compared to the Question Type (QT).

Table- III: Trigger word confusion matrix

	AT							
QT	*How*	*What*	*When*	*Where*	*Which*	*Who*	*Why*	-
How	22	1	1	1	0	0	2	7
What	5	62	1	2	1	2	2	12
When	0	0	27	0	0	0	1	5
Where	0	0	0	16	0	0	0	3
Which	0	0	2	2	22	0	0	12
Who	0	1	0	0	0	68	0	8
Why	0	1	0	0	0	0	10	1

7. *Confusion Matrix:* This is the matrix which summarizes the predictions into 4 classes. These classes are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

Table- IV: Confusion matrix

Response	Answer Present in Corpus	Answer Absent in Corpus
Correct	True Positive	---
Incorrect	False Positive	False Negative
Neutral	False Positive	True Negative

In our project, the confusion matrix is as follows.

Table- V: Confusion matrix of evaluated system

Domain	TP	TN	FP	FN
Education	36	3	9	2
Technology	37	2	10	1
Entertainment	36	2	10	2
Politics	34	2	13	1
Sports	33	1	14	2
Health & Nutrition	36	4	9	1
Total	212	14	65	9

Here 212 answers have been classified correctly and 14 of them are correctly classified as neutral with 65 labelled as incorrect in spite of being in the corpus and 9 as incorrect which were absent in the knowledge base.

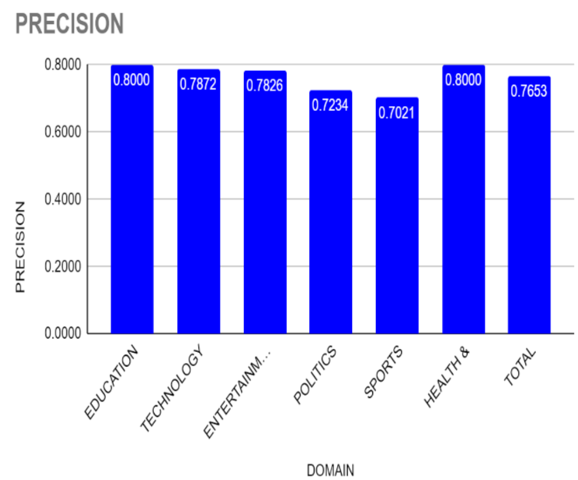

Fig 5.1 Graph plot of Precision vs different domains

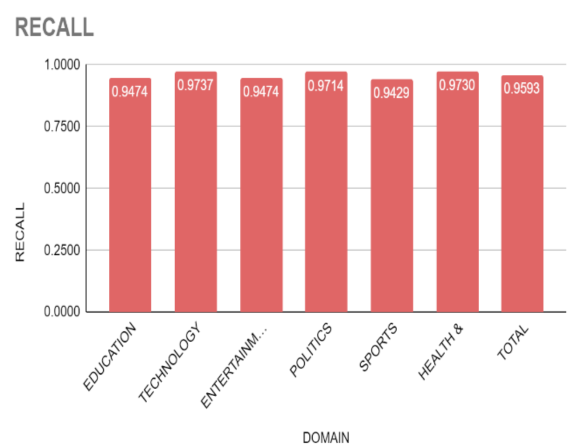

Fig 5.2 Graph plot of Recall vs different domains

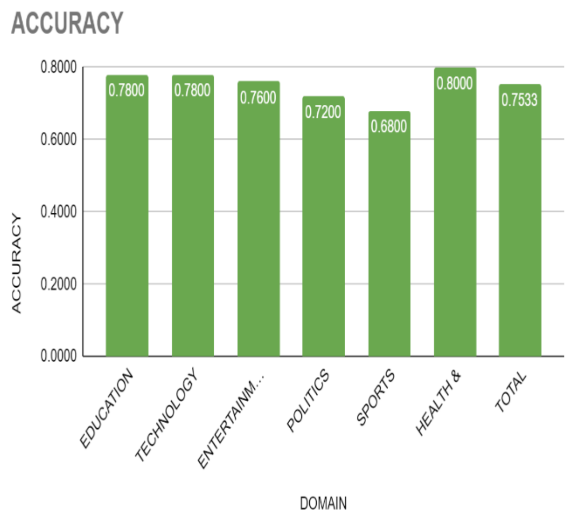

Fig 5.3 Graph plot of Accuracy vs different domains

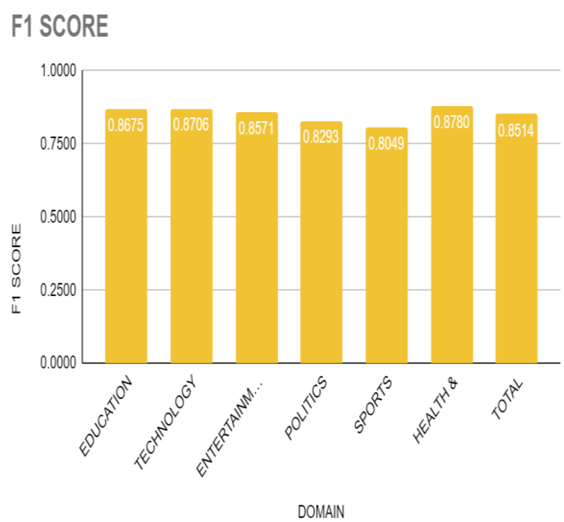

Fig 5.4 Graph plot of F1-Score vs different domains

The above graphs depict the graph plot of various evaluation metrics vs different domains used for testing the system. From the above graphs we can observe that the model has high precision, recall, accuracy and F1-Scores resulting in low errors. Here we also observe that the education and health & nutrition domain has the highest precision as 0.8, the technology domain has the highest recall as 0.9737, the health & nutrition domain has the highest accuracy as 0.8 and again the health & nutrition domain has the highest F1-Score as 0.8780.

VI. CONCLUSION

Thus, the Open Domain QA System answers questions by making full use of knowledge sources of Wikipedia for many types of questions. It uses various algorithms, techniques and models for implementation including the Wikipedia API, BERT model and SoftMax function. Rigorous testing has been done using various performance metrics such as accuracy, precision, recall, etc. It has a wide range of applications in the medical domain, education, research.

The accuracy and efficiency of the Open Domain QA System were improved by clearly and properly framing the questions, using the appropriate trigger words with higher token length, choosing only 5 relevant articles from the corpus and extracting the answer in 1-2 sentences, asking the questions correctly and for the answers which were absent in the corpus, the system informed the user about the same. One of the future plans may be to apply the Speech Based Open Domain QA System. It can truly benefit the users as there won't be any limitation towards any specific domain. Extending work on Visual QA systems where the answer to the question can be answered on the basis of image or set of images provided is also interesting.

VII.ACKNOWLEDGMENT

We are immensely obliged to Dr. Sharvari Govilkar, H.O.D. of Computer Engineering Department and Project Supervisor for providing us with the understanding and idea of this project topic and her indispensable support. We would like to offer our gratitude towards our college, Pillai College of Engineering, New Panvel and our principal Dr. Sandeep Joshi for providing us with all the facilities to complete our project. We would also take this opportunity to thank Dr. Satishkumar Varma, H.O.D of Information Technology and Prof. Gayatri Hegde, BE Project coordinator and other professors for their valuable support. Also, we would like to express our grateful appreciation to all those people and staff who provided us the possibility to complete this report.

REFERENCE

[1] Ali Allam, Mohamed H Haggag, "The Question Answering Systems: A Survey", College of Management & Technology, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt, 2012.
[2] Sanjay K Dwivedi, Vaishali Singh, "Research and reviews in question answering system",

International Conference on Computational Intelligence: Modelling Techniques and Applications (CIMTA), 2013.

[3] Pum-Mo Ryu, Myung-Gil Jang, Hyun-Ki Kim, “Open domain question answering using Wikipedia-based knowledge model”, Electronics and Telecommunications Research Institute, 138 Gajeongno, Yuseong-gu, Daejeon, Republic of Korea, September 2014.

[4] Zhonglin Ye1, Zheng Jia, Yan Yang, Junfu Huang and Hongfeng Yin, “Research on Open Domain Question Answering System”, School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China, fourth CCF Natural Language Processing and Chinese Computing Conference, October 2015.

[5] Vivek Datla, Sadid A. Hasan, Joey Liu, Yassine Benajiba, Kathy Lee, Ashequl Qadir, Aaditya Prakash, Oladimeji Farri, “Open Domain Real-Time Question Answering Based on Semantic and Syntactic Question Similarity”, in the Live-QA track of TREC, 2016.

[6] Natural Language Computing Group, Microsoft Research Asia, “R-Net: Machine Reading Comprehension with Self-Matching Networks”, May 2017

[7] Danqi Chen, Adam Fisch, Jason Weston & Antoine Bordes, “Reading Wikipedia to Answer Open-Domain Questions”, April 2017.

[8] Ibrahim Mahmoud Ibrahim Alturani and Mohd Pouzi Bin Hamzah, “A New Approach For Open-Domain Question Answering System”, International Journal of Computer Science and Network Security, June 2018.

[9] Rohan Sampath, Puyang Ma, “QA with Wiki: improving information retrieval and machine comprehension”, 2019.

[10] Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, May 2019.

[11] Zhuosheng Zhang1, Junjie Yang, Hai Zhao, “Retrospective Reader for Machine Reading Comprehension”, January 2020.