“Information Retrieval and Conversational AI”

By
Sharath Kumar, Ganganna (sg17402)
sg17402@essex.ac.uk

A thesis submitted for the degree of Master of Science in Artificial intelligence

Supervisor: Woods, John C
School of Computer Science and Electronic Engineering
University of Essex

August 2018

Abstract:

The project describes the implementation of an Artificial Intelligent Conversational Chatbot and relevant information extraction from the internet in real time. Trying to mimic true artificial intelligence behaviour by data mining and web scrapping query results from the internet. All this is achieved by not having an own database of information or knowledge this makes this project challenging which is overcome by obtaining the information in real time and processing the information by giving only meaning full insight to the user for the query raised by the user. We compare present state-of-the-art Technologies that are trying to mimic AI conversational behaviour and their limitations along with the implemented system and also the application and scope of the implemented system.

Table of Contents

Table of Figures:

1. Introduction

During human evolution, through innovative breakthrough, the electronics and computational achievement have made enormous progress in the advancement of technology. Since the 21st century there has been a remarkable advancement from radio system to cell phones and in the previous 4 decades the Engineering Field has been taken over by intense consumer PC and compact cell phones. There has been a verifiable advancement and development of PCs which are computationally powerful. The innovation of convenient PCs developed exponentially in a very short span of time. Today we have unmatched computational power in our grasp as in an everyday useable cell phone. When we had lot of powerful laptops, every company invested a lot of money in research to conquer the frontier of portable devices as the consumer market wanted all the facilities and applications to be reachable at their fingertip, after complete revolutionising of the smartphone industry the next big fortune is to have an invisible system or personal assistance which can do all kinds of possible help and activities for us. There is no ambivalence we are making a future of hands-free interface along with the developing innovative advancement of consumer client gadgets which conveys us to the question of producing a genuine AI reasoning framework which can perform hands-free jobs and give better and reliable outcomes to the consumer gadgets or hand-free systems.

As the final goal is to make a stable portable personal assistance system, there are many personal assistance systems such as - heart monitoring systems, home automation systems available in the market but we don't have a single system which supports all these features. As advancements in technology are seen, a necessity to meet varying demands of the consumers and for a stable presence in this competitive world, there is a need to design energy efficient and reliable system which has all these facilities integrated into it. Though there are many personal assistant systems available in the market, we don't have a complete hands-free environment in any system. In case of some complicated search, many systems fail to answer user queries. It has many areas to deal with, starting with hardware and software perspective. Only the software perspective will be implemented and researched in this project. The portable aspect of the system will be discussed in the future work and goal. As for software implementation, A fully functional information retrieval system through a Chatbot will be discussed through the paper.

The pre-eminent objective of this experiment is to fabricate a framework which can be a extremely wide way to deal with mimic artificial intelligence and give results to any kind of question asked by the client through voice, for example, we have best in class advancements from Tech-Giants like Apple, Google and Amazon attempting to give assistant through voice. For instance, Organizations like Apple Siri, OK Google, and Alexa. This cutting-edge advancement is constrained to particular functionalities and a predefined stream of working flow or pre-defined answers to a query made by the client. But there has been an immense move in the improvement of such advances as we unwind better strategies to make a conversational framework. At a point when a special or intriguing inquiry is asked to the current models, it steers to the web with respect to the pertinent outcomes yet not giving any sufficient input on the inquiry made

by the client. So here we attempt to tackle that section of what is missing by endeavouring to offer in debrief criticism to the client by Scrutinizing related data from the web.

An intuitive interface will be made which will acknowledge the query from the client by voice to text through a speech synthesizer and process the question and imitate a genuine AI conversational framework. For the execution of the conversational AI framework, A Chatbot will be constructed utilizing cutting-edge innovation such as a profound neural system with LSTM word (seq2seq) grouping to succession strategy prepared on a different diverse conversational dataset which will be extremely necessary while trying to achieve a general point of view of the AI conversational Chatbot. The informational data set for dialogue could be from a collection of famous motion picture or discussions between two individuals or a large arrangement of captions and discussions between A2B. For the construction of the information of retrieval framework for the inquiry raised by the client, the information must be extracted from the web by mining and scraping of information. This will be performed by utilizing extensive API of different sorts and a scope of internet sites in parallel. For example, scratching and mining data from Wolfram Alpha, Wikipedia, Quora, Google, and Yahoo. A website will be utilized as an interface to execute the planned framework as it is anything but difficult to feature the system and exhibit the working of the framework utilizing speech to text and text to speech in the browser than on a portable hardware build framework.

2. Literature review

2.1. Background

2.1.1. Encoder Decoder model

The encoder-decoder design for serial neural systems is the standard neural machine interpretation policy that opponents even the traditional methods and beats the moderate machine translation techniques. This kind of architecture engineering is a very new approach, having just been begun in 2014[12], in spite of the fact that it has been embraced as the centre of innovation inside Google's translate service [33] [11]. The Encoder-Decoder engineering with repetitive neural systems has turned into a powerful and standard approach for both neural machine translation: NMT and sequence-to-sequence - seq2seq [33] in general for prediction in the industry.

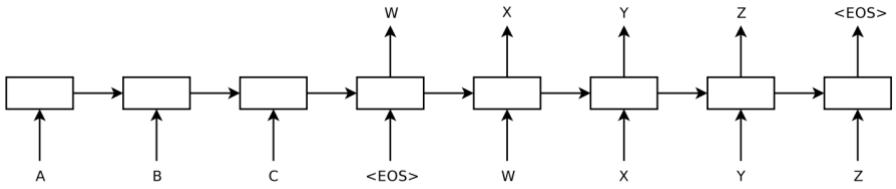


Figure 1 Encoder-Decoder Model for Text Translation

The key advantages of the approach are the potential to prepare an individual end-to-end Model on point from the various source and target sentences

collected and the potential to deal with the variable length of information or data and still yield an output of content. An Encoder-Decoder architecture was initially produced when an entire sequence of words from the sentence was accepted and encoded into the model as input. A decoder system then arranges the utilized interior replica to yield words until the point where the sentence ends of sequence token. LSTM systems were utilized for both the encoder and decoder [12].

" The concept is to utilize one LSTM to peruse the input sequence, one time step at a time, to get large fixed dimensional vector replica, and then later to utilize another LSTM to extract the output sequence arrangement from that vector"

2.1.2. Seq2Seq Chatbot

Chabot's these days that uses profound learning is generally using some variety of a sequence-to-sequence (Seq2Seq) is advancing broadly in the market. A Seq2Seq model is made out of two essential parts, an encoder and a decoder. Recurrent neural network from an abnormal case, the encoder's action is to illustrate the information of the data content into a settled copy. The decoder's action is to take that reproduction and make a variable length message that best responds to it [25].

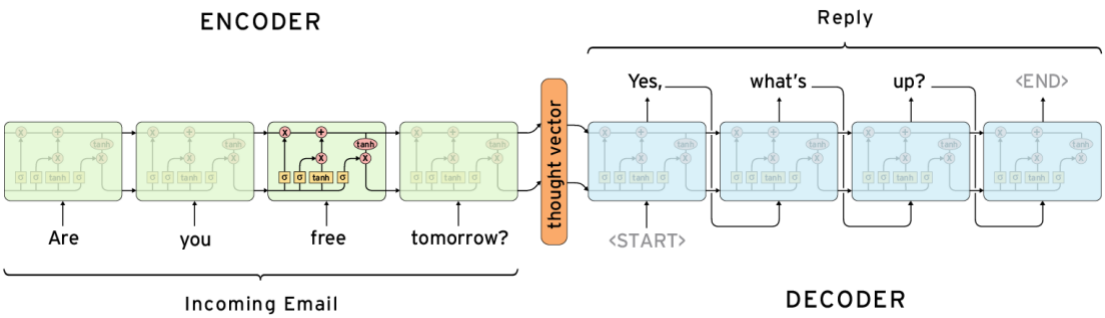


Figure 2 words to words seq2seq framework

RNN or fundamental RNNs are one kind of a neural frameworks that are prepared for overseeing progressive data as recordings with the frame to frame sequence and more often than making a critical impact on the model while training. The model with Sequence to Sequence (seq2seq) includes two Recurrent neural networks, an encoder and a decoder as of now explained.
The encoder examines the data input, word by word and transmits a succession of the setting of contribution to the model which the entire informational index would be dealt with as numbers and not as words or characters. We accomplish this by utilizing the word to vector directory which can be obtained by self-collection or depending standard libraries for converting a change of the discussion.

2.1.3. LSTM

One of the interests of recurrent Neural network RNNs is the possibility that they may have the capacity to interface past data to the present job. For example, utilizing past video edges may advise the comprehension of the present casting of frame. On the off chance that recurrent Neural networks could do this, they'd be helpful to a great extent. But it depends if they can help with this problem or not. Sometimes, we just need to see at late data to play out the present task. For instance, consider a language model attempting to foresee the following word in view of the past ones in the event that we are attempting to anticipate the last word in "the mists are in the sky," we need not bother with any further setting – it's quite evident the following word will be the sky. In such cases, where the gap between the relevant data and the place that it's required is little, RNNs can figure out how to utilize the past data.

LSTMs are commonly just called as Long Short-Term Memory frameworks. This is a remarkable combination of RNN, prepared for adjusting aggregate conditions. In 1997, They were exhibited by Hochreiter and Schmidhuber [12]. LSTMs are explicitly proposed to evade the long-haul dependence issue. At that point, later LSTM was intended to determine the issue unequivocally.

The key to LSTMs is the cell present state, the flat line going through the highest point of the diagram. The cell state is somewhat similar to a transport line It runs straight down the whole chain with just some minor direct communications. It's simple for data to simply stream along it unchanged. The LSTM has the capacity to expel or add data to the phone state precisely controlled by structures called gates. Gates are an approach to alternatively let data through. They are made out of a sigmoid neural net layer and a pointwise multiplication method.

.

2.1.4. Data Scraping

Data scraping, otherwise called web scraping is the way towards bringing in data from a site into a spreadsheet or locally saved in the memory on your PC. It's a stand out amongst the most productive approaches to get information from the web, and at times to channel that information to another site. Mainstream uses of data scraping incorporate, for example, Research for web content/business knowledge, evaluating for travel booking destinations/value, discovering potential customers/directing statistical surveying by crawling open information sources. For example, yelp and Twitter. Sending item data from a web-based business webpage to another online merchant such as Google Shopping. Furthermore, that rundown simply touching the most superficial layer. Information scraping has a countless number of advantages and applications – it's helpful in pretty much any situation where information should be moved from one place to another.

"Web scraping," likewise called crawling or spidering, is the robotized way of assembling the information from another person's site. Scraping is a basic piece of how the Internet capacities. For instance, Google utilizes web scraping to manufacture its pursuit database worth several billions of dollars. Numerous other online administrations, extensive and little, utilize scratching to manufacture their databases too. Often sites will permit outsider scraping. For instance, most sites give Google the express or inferred consent to list their site pages. Web Scraping (additionally named Screen Scraping, Web Data Extraction, Web Harvesting and so forth.) is a method utilized to remove a lot of information from sites whereby the information is extricated and saved to memory for future use or real-time usage.

The information shown by most sites must be seen utilizing the internet browser. They don't offer the versatility to save a duplicate of this information for individual use. The main choice at that point is to physically replicate the information - an exceptionally tedious activity which can take infinite hours or days to finish. It is the method of automating this procedure so that rather than physically duplicating the information from sites, the Web Scraping programming will play out a similar errand inside a small amount of time. This technique, generally revolves around the difference in unstructured data of HTML on the web into composed data which we can make utilization of. Web scratching can be performed in various ways, utilizing mainstream libraries like beautiful soup and request in python or utilizing standard API given by tech giants and information collection aces like Wikipedia and Wolfram Alpha.

2.1.5. Beautiful Soup

Beautiful Soup is a Python library intended for fast turnaround ventures like screen-scraping. main highlights which make it ground-breaking are Beautiful Soup gives a couple of straightforward strategies and Pythonic expressions for exploring, seeking, and adjusting a parse tree: a toolbox for analysing an archive and separating what you require. It doesn't take much code to compose an application. Beautiful Soup consequently changes over approaching records to Unicode and active archives to UTF-8. there is no need to consider encodings except if the case doesn't indicate an encoding and Beautiful Soup can't identify one. During that period, you simply need to indicate the first encoding.

Beautiful Soup sits over well-known Python parsers like LXML and html5lib, enabling you to experiment with various parsing systems or exchange speed for adaptability. Beautiful Soup parses anything you give it and does the tree traversal method for the extracted HTML data from the website. You can let it know "Discover every one of the connections", or "Discover every one of the connections of class external Link", or "Discover every one of the connections whose URLs coordinate to given example also, or "Locate the table heading that is got strong content". Valuable information that was once secured up ineffectively outlined sites is currently in scope. Tasks that would have taken hours now take just minutes with Beautiful Soup.

2.1.6. Question and Answer System

Question answering (QA) is a software engineering method inside the fields of data retrieval and natural language processing (NLP), which is anxious about building frameworks that consequently answer questions postured by people in a characteristic dialect. A QA execution, as a rule, a PC program, may build its answers by questioning an organized database of learning or data, for the most part, an information base QA frameworks can pull answers from an unstructured gathering of common dialect reports. A few cases Some examples of natural language document collections for QA frameworks, for example, a neighbourhood gathering of reference writings, interior association records, and pages, assembled newswire reports, an arrangement of Wikipedia pages or a subset of World Wide Web pages [5]

Web indexes display a ranked list of significant reports in reaction to client figured catchphrases in light of different perspectives, for example, notoriety measures, catchphrase coordinating, frequencies of getting to archives, and so on. Be that as it may, they don't genuinely achieve the assignment of data recovery as clients have to look at each record one by one for getting the coveted data. it makes data recovery a tedious procedure. In a perfect world, a web index ought to return a couple of significant and brief sentences as answers along with their relating web join. Many have been created since the 1960's Androutsopoulos et al.1995, Kolomiyets, 2011[5]. Current QASs endeavour to reply to questions asked by clients in common dialects subsequent to recovering, what's more, handling data from various information sources even like the semantic web [16]. The arrangement of answers is moreover going to be changed from straightforward content to mixed media. QASs created since 1960s address distinctive areas, information sources, sorts of questions, configurations of answers, and so forth. The quantity of such QASs is too vast. To evaluate the achievement of these QASs and their capacity to fulfil present and future needs, an efficient review of all these QASs winds up important.

2.2. Research

In the article review on Chatbot Design Techniques in Speech Conversation [9] all the primitive ways to deal with building a conversation framework are looked at. Pros and cons of the central strategies used to fabricate the Chatbot such has parsing, design co-ordinating and also AIML (Artificial Intelligence Mark-up Language) or even simple methods, lead-based framework such as Markov Chain and chat contents are predefined. Clarifying the unquestionable required, field information and necessities should be considered before building a Chatbot. Likewise, it clarifies the mechanical principles of building a remarkable Chatbot looking at the standard set up by competitions like Loebner Prize and the Turing Test [9].

Deep Reinforcement Learning is one of the most spoken or buzzword of this decade in deep learning and for Open-Domain Dialogue Creation. It is the most cutting-edge problem statement to make a model to deal with ongoing generative conversation and continuous learning procedures and only conceivable by reinforcement Learning. The paper gives knowledge into the Reinforcement learning design for neural response creation by re-enacting

dialogue between two specialists [11], consolidating the characteristics of neural Seq2seq (sequence to sequence) frameworks and reinforcement learning for a conversational model. As a hypothetical idea of reinforcement, learning is to remunerate for the move made by the model continues to offer criticism to the framework. The paper clarifies how the strategies and prizes can be incorporated into a conversational model to deal with human-like learning conduct. what is more clarifying the significance of having a major informational collection of two conversational operators to prepare the model where they utilized an excess of 10 million conversations from Open Subtitles dataset which is one of the fascinating datasets to prepare a Chatbot.

Apple Siri On-Device Deep Learning [36] [35] the paper clarifies how Apple dealt with a Guided Unit Selection Text-to-Speech System and subtle elements to Apple's crossbreed unit determination conversation synchronization framework. In the paper, it is a blend of a review of the continuous TTS motor and the voice building process. Furthermore, numerous methods which influence Siri to do on-gadget administrations and streamlining which gives the voice to Siri and Apple Maps. The framework is all around prepared to work with 6 distinct languages all working at 40% superior to the standard of the separate language. The framework utilizes deep and recurrent mixture blend network systems to actualize target and connection costs.

The Neural Conversational model did not propose any original thought. It accomplished something intriguing that the creator connected the Seq2Seq model, described in Sequence to Sequence Learning by Neural Networks, to not rendering translation but rather the conversation building task. Hence, this paper named A Neural Conversational Model [13] Investigating the Seq2Seq method, which depends on an intermittent neural system, it pursues the information sequence one token at any given moment and predicts the result sequence, additionally one token at once. Amid preparing, the genuine result sequence is given to the model, so learning should be possible by backpropagation. Also, the model is prepared to boost the cross-entropy of the right sequence given its specific reasoning. Since given that the genuine result sequence isn't seen between induction, the model basically feeds the anticipated result token as a contribution to foresee the following result. This is a "greedy" deduction approach. A less voracious approach is utilized BEAM search [13] and feed a few results at the last step to the following stage. The anticipated sequence can be chosen in view of the likelihood of the sequence.

The creator showed that Seq2Seq model is straightforward and general, and the best approach to utilize Seq2Seq to fabricate conversational models demonstrating is straight and easy. The info sequence can be the connection to what has been spoken until this point (the unique situation) and the result sequence is the answer. In any case, the Seq2Seq model won't have the capacity to effectively solve the issue of modelling conversation because of a few clear disentanglements: The target work being streamlined does not catch the genuine goal accomplished through human messages, which is normally long haul and in view of trading the data in exchange to the subsequent stage is the prediction. The absence of a model to guarantee consistency and general world learning is another conspicuous limitation of a simply unsupervised model. In the paper,

the creator utilizes this Seq2Seq model to deal with two datasets and demonstrate the outcomes one is a closed-domain IT helpdesk investigating dataset and another is an open-source motion picture transcript dataset.

3. Technologies and approach

3.1. Backend Technologies

3.1.1. Seq2seq Chatbot

3.1.1.1. Data Set

To make the Chatbot wider stream we need to infuse many streamline conversations dataset to make the Chatbot powerful enough to handle a general conversation. The Chatbot can be as good as the dataset. To start a working model, we start with the infamous Cornell movie dataset which is used and referenced in many papers to research and explore seq2seq models and LSTM Chatbot. The Cornell movie dataset contains 220,579 conversational dialogues between 10,292 pairs of film actors and also around 9,035 characters from around 617 movies which are from many genre and storylines. In the data set total, we can find 304,713 utterances in the Cornell movie dataset which is really well-formatted compared to others, ready to use for training with little pre-processing.

3.1.1.2. Pre-Processing

The average size of the words in each sentence in the dataset can range from 20 to 30 words and a maximum of 45 words of very lengthy sentences from the movie dialogue conversation between two people. For the purpose of having a stable powerful data set, the sentences are shopped logically and grammatically to still make sense and hold the meaning of the sentence. The logical solution like breaking a sentence when a '.' the sentence ends with a dot only the first half of the sentence is used after that if the sentence is still more than required length, will be chopped using conjunction words such as 'and', 'for', 'but', 'yet', 'so', 'though', 'although', 'before', 'now then', 'once', 'since', 'when', 'whenever' etc.

3.1.1.3. Word to Word model

Implementation of the seq2seq Keras LSTM model, the model consists of 2 recurrent neural networks RNNs one of which is the encoder which maps a variable-length reference sequence of input to a fixed-length vector and the decoder which maps the vector representation back to a variable-length of the target sequence which is the output. 2 RNNs are trained simultaneously to maximize the dependent probability of the target sequence given a reference sequence.

For the seq2seq word model before training, we work on the dataset to convert the variable length sequences into fixed length sequences as each subset of the

sample may contain different length according to its contextual meaning. By padding we use a few special symbols to fill in the sequence which can make all the samples to the same fixed length for training the seq2seq model. The important terminology to be known is:

- 'END': End of sentence
- 'PAD': Fill
- 'START': Start decoding
- 'UNK': Unknown; a word not in the vocabulary

Consider the following query-response pair.

- Question: How are you?
- Answer: I am fine.

As we need our sentences to be of fixed length, let say 10 for example, this pair will be changed to:

- Question: [PAD, PAD, PAD, PAD, PAD, PAD, "?", "you", "are", "How"]
- Answer: [START, "I", "am", "fine", ".", END, PAD, PAD, PAD, PAD]

3.1.1.4. Keras LSTM encoder decoder model

Using Keras we can build a Sequential model by giving a list of layer instances to the constructor prior to training of the model. You need to set the learning process, the rate and the momentum of the learning curve and also what sort of classification or regression is being carried out by the model. We also need to specify the loss function methods and optimization method in this case, we will be selecting categorical cross-entropy for loss and 'rmsprop' for optimization of the model.

The encoder and decoder part of the LSTM model is added to the areas of the model as input and output by preparing the encoder and the decoder model and then adding it to the main Keras model configuring where and how it is connected to the model.

```
encoder_inputs = Input(shape=(None,), name='encoder_inputs')
encoder_embedding = Embedding(input_dim=num_encoder_tokens, output_dim=HIDDEN_UNITS,
                              input_length=encoder_max_seq_length, name='encoder_embedding')
encoder_lstm = LSTM(units=HIDDEN_UNITS, return_state=True, name='encoder_lstm')
encoder_outputs, encoder_state_h, encoder_state_c = encoder_lstm(encoder_embedding(encoder_inputs))
encoder_states = [encoder_state_h, encoder_state_c]

decoder_inputs = Input(shape=(None, num_decoder_tokens), name='decoder_inputs')
decoder_lstm = LSTM(units=HIDDEN_UNITS, return_state=True, return_sequences=True, name='decoder_lstm')
decoder_outputs, decoder_state_h, decoder_state_c = decoder_lstm(decoder_inputs,
                                                                 initial_state=encoder_states)
decoder_dense = Dense(units=num_decoder_tokens, activation='softmax', name='decoder_dense')
decoder_outputs = decoder_dense(decoder_outputs)
```

Figure 3 Keras Encoding and Decoding implementation

When all the layers are confirmed and added in sequence to each other the defined model is built as a single body by compiling the model. The model is now complete and can perform many inbuilt features and option from Keras which also facilitates us to see the summary of the model in brief.

Layer (type)	Output Shape	Param #	Connected to
encoder_inputs (InputLayer)	(None, None)	0	
encoder_embedding (Embedding)	(None, 10, 256)	48896	encoder_inputs[0][0]
decoder_inputs (InputLayer)	(None, None, 187)	0	
encoder_lstm (LSTM)	[(None, 256), (None,	525312	encoder_embedding[0][0]
decoder_lstm (LSTM)	[(None, None, 256),	454656	decoder_inputs[0][0] encoder_lstm[0][1] encoder_lstm[0][2]
decoder_dense (Dense)	(None, None, 187)	48059	decoder_lstm[0][0]

Total params: 1,076,923
Trainable params: 1,076,923
Non-trainable params: 0

Figure 4 Keras Sequential Model summary

The pre-processed dataset is now index of each word to an ID and a directory is saved in memory which will be used to get the word back from the id in the decoding part of the model. Also, a directory with ID to words is also produced and saved in memory using numpy in python which can be accessed again.

```
Question: ['gosh', 'if', 'only', 'we', 'could', 'find', 'kat',
'a', 'boyfriend']
Answer: ['START', 'let', 'me', 'see', 'what', 'i', 'can', 'do'
, 'END']

Question ['right', 'see', 'you', 'are', 'ready']
Answer: ['START', 'i', 'do', 'not', 'want', 'to', 'know', 'how
```

The words 'UNK', 'PAD' which will be used to fill in the sentence to make it a fixed length also has to index in the numpy dictionary as 'UNK' with index 0 and 'PAD' with index 1. This will help the decoded identify the 'UNK' and 'PAD' in the decoding part of the model. After padding the input sentence to the preferred length, the words and the result will be encoded with the saved word to ID directory and will be fed to the model to train the input data and same for the target data as validation.

3.1.1.5. 100 sample experiments

For the 100-sample experiment, we use word dimension of 187 total number of word tokens and a maximum of 12 decoded sequence length and 191 encoder word tokens with 10 max sequence length:

- 'num_decoder_tokens': 187
- 'decoder_max_seq_length': 12
- 'num_encoder_tokens': 191
- 'encoder_max_seq_length': 10

With 100 samples used to train the Keras model, each epoch took 5 seconds as it used just 77 samples for training and 23 for testing and validation to test the accuracy and loss of the model. The model starts learning gradually and the

drop in the loss could be seen very soon in an early stage from 200% to 40% in just 40 epochs and to 3% at 100th epoch for 100th sample. The model took 100 epochs to reach a good accuracy to learn all the samples and target output. This gave a hint suggesting that the number of samples is directly proportional to the number of epoch the model has to train for so the model learns from the dataset. The model can clearly answer any question asked from the bucket of 100 samples and it feels like it's a complete system whereas it was only trained on 100 samples. Hence, the goal is to have a proper dataset with a good number of samples in it and the time and resource such as a high-end GPU to train the model. From this, an inference was made to understand the training of the model and how much time the model will need to be trained on 100,000 samples or even 2000,000 samples.

But the experiments do not hold good with the above theory. When tried with 1000 samples the model expects more than 1000 epochs to even reach 30% loss and 70% accurate. The number of the epoch is exponentially higher to the number of sample in the training example. Keras and TensorFlow suggest 20,000 to 30,000 epoch minima for a powerful Keras model to learn from 100,000 samples and be capable to handle a conversation.

Epoch NO.	Time to train		Loss
1	6s	loss:	2.3198
2	5s	loss:	2.1057
3	5s	loss:	1.9998
4	5s	loss:	1.9011
5	5s	loss:	1.8391
6	5s	loss:	1.7755
7	5s	loss:	1.7342
8	5s	loss:	1.6594
9	5s	loss:	1.6327
10	5s	loss:	1.616
..	..	..	..
93	5s	loss:	0.0397
94	5s	loss:	0.0396
95	5s	loss:	0.0389
96	5s	loss:	0.0384
97	5s	loss:	0.0381
98	5s	loss:	0.0381
99	5s	loss:	0.038
100	5s	loss:	0.0373

Table: training logs of 100 samples

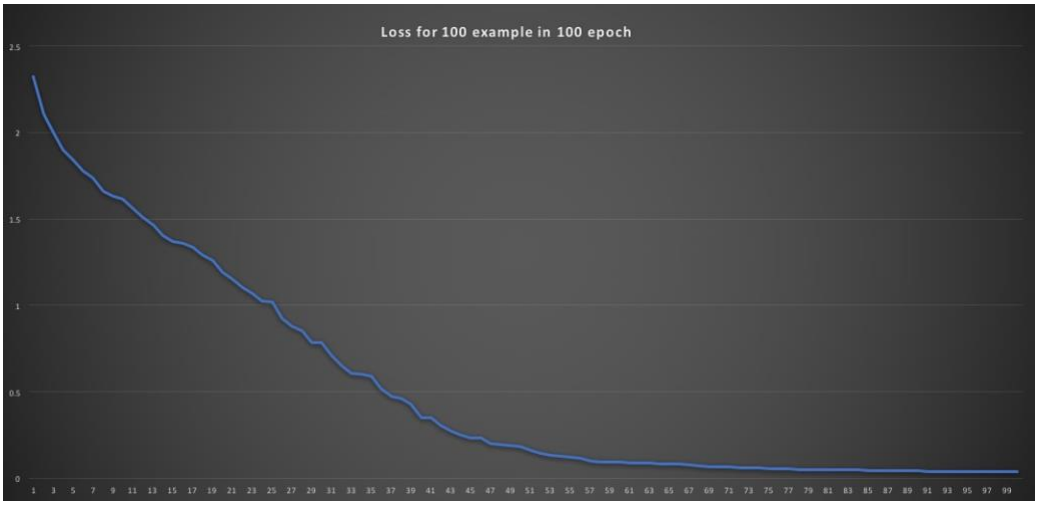


Figure 5 training loss histogram of 100 sample in the model

For the main goal of the project, we use 10,000 examples and train the model to have a working Chatbot. The training of the model is set up in a cloud platform to provide all the resources and uninterrupted platform for the Keras model to train on the samples. Selecting the best cloud platform was a tedious task as all platforms had many options and parameters to choose from and was expensive. The model was set up on "paper space", fast virtual desktops to boost the task with the next generation of computing framework. Using High-end GPU such as Nvidia k80 with 16gb of graphical memory is the highest available in the market for consumer use.

After many iterations of trial and error to make the model stable the trails were performed on the local personal computer for hours together and training data and behaviour of the model was noted down for future use and the model was tested by communicating to it with random test samples which paved way to get better understanding of the weakness of the model before training with a larger dataset. The model was set up with CPU 32gb and powerful GPU. The model was trained for 5 days straight with 10,000 examples and could only reach 35% loss mark and after 2500 epochs in 5 days an accuracy of 65% was achieved. As the model is not completely ready it is prone to have the index 0 "UNK" words suggested many a times in place of less probabilistic cases which can be got rid of when the model achieves less than 5% loss.

3.1.2. Django and python

Designers are continually hunting down for the best. They look for the best language to code in, the best instruments to utilize, and they are continually searching for what is at the cutting edge of program development. In any case, distinguishing which language and tools are the best can be troublesome. The

decision relies upon the engineer, the task, and the instruments accessible. The tech gaits, for example like Facebook Netflix, think of employing Django as the web system of your project is the most ideal approach to transform your thought into business reality. Django web system is composed on the snappy and ground-breaking Python language.

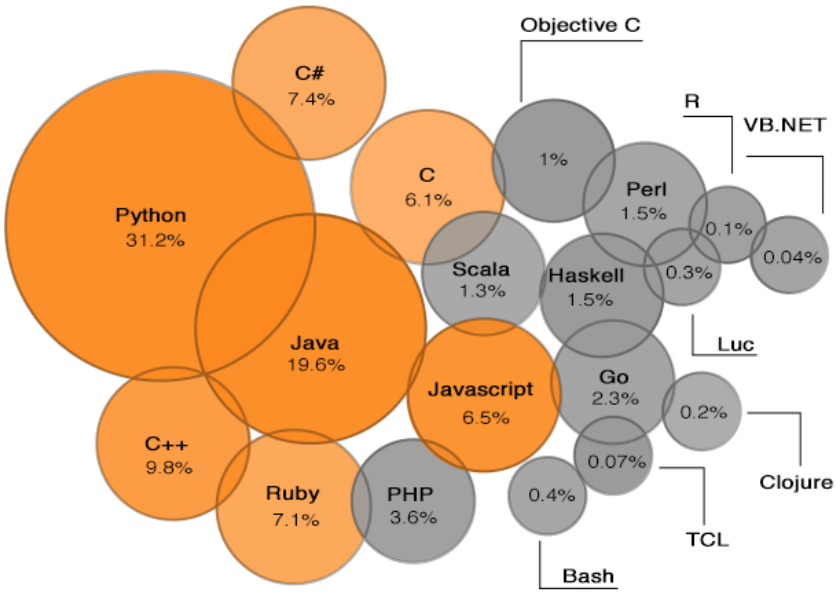


Figure 6 seniority of python against other language

Python chips away at any stage and is likewise open source. The language depends on stand-alone theory, "The Zen of Python", [30] These standards of code composing are one of a kind to Python, giving the language leverage over others that have no rationality controlling them. The standards are what helps Python designers produce top-quality code.

For Django, there is much to be said in regards to the benefits of Python. It has been the most well-known advancement language for quite a while and keeps on being a most desired framework among talented programmers. Even if the project is created by one group all the way, using Django turns the development process remarkably fast. Django is the best answer for making an MVP system that can be additionally built on based on the fact that it comes completely featured standalone package, reserve out of the container.

3.1.3. Incorporating Wikipedia

Wikipedia is an Internet reference book, sustained and facilitated by the non-benefit Wikimedia Foundation. It is a free-of-cost reference book [31] with its articles being free-content the individuals who utilize Wikipedia can generally alter any article available. Wikipedia is positioned among the ten most prominent sites and constitutes the Internet's biggest and most prevalent general reference work. Jimmy Wales and Larry Sanger propelled Wikipedia on January 15, 2001. Sanger authored its name, a portmanteau of wiki and

encyclopaedia. At first, just in the English dialect, Wikipedia rapidly turned up multilingual as it created comparable forms in different dialects, which vary in content and in altering practices. The English Wikipedia is presently one of 291 Wikipedia releases and holds the biggest measure of articles with more than 5,154,440 - having outperformed 5,000,000 articles in November 2015. Over every single current encyclopaedia, Wikipedia comprises an amazing aggregate of more than 38 million articles in more than 250 distinct dialects. As of February 2014, it had 18 billion site hits and about 500 million distinct guests every month. A companion audit of 42 science articles found in both Encyclopaedia Britannica and Wikipedia was distributed in Nature in 2005 and found that Wikipedia's level of exactness approaches Encyclopaedia Britannica's. Notwithstanding, Wikipedia has been abstractly criticized, claims demonstrate that Wikipedia shows foundational inclination, displays a blend of "certainties, misleading statements, and a few deceptions", and questionable themes could be controlled and spun. [31]

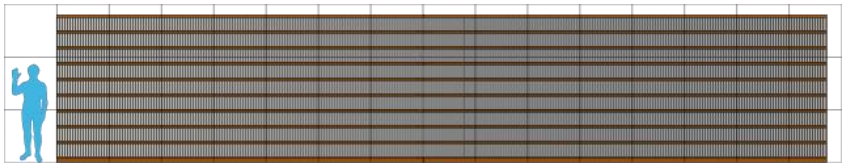


Figure 7 all the information of Wikipedia printed and visualized

The whole Wikipedia dataset itself can be calculated to 100GB as per the recent fact from Wikipedia publications [30]. getting the data for self-using can be a better option than to go on scraping the website but research purpose scraping the website was a better approach, As Wikipedia is open source there are many standard python API which facilitates scraping of information from Wikipedia in real time.

3.1.4. Integrating Wolfram alpha

Wolfram Alpha - is a computational learning engine or answer engine created by Wolfram Research, which was established by Stephen Wolfram. It is an online administration that answers truthful inquiries straightforwardly by processing the appropriate response from remotely sourced "curated data", as opposed to giving a rundown of archives or pages that may contain the appropriate response as an internet searcher may. Wolfram Alpha, which was released on May 18, 2009, depends on Wolfram's prior lead item Mathematica, a computational stage or toolbox that incorporates PC variable based math, representative and numerical calculation, perception, and insights abilities Additional information is assembled from both scholastic and business sites, for example, the CIA's The World Fact book, the United States Geological Survey, a Cornell University Library production called All About Birds, Clients submit inquiries and calculation demands by means of a content field. Wolfram Alpha at that point processes answers and pertinent representations from an information base of curated data, organized information that originate from different destinations and books. The site utilizes an arrangement of robotized

and manual strategies, including insights, perception, source cross-checking, and master audit." The curated information makes Alpha unique in relation to semantic web indexes, which list a substantial number of answers and after that attempt to coordinate the inquiry to one perspective. Wolfram Alpha can just give vigorous inquiry results in view of computational realities, not inquiries on the sociologies, social investigations or even numerous inquiries regarding history where reactions require more depth and multifaceted nature. It can react to especially expressed common dialect certainty based inquiries, for example, "Where was Mary Robinson conceived?" or more unpredictable inquiries, for example, "How old was Queen Elizabeth II in 1974?"

It shows its interpretation of such an inquiry utilizing standardized expressions, for example:

- "period of Queen Elizabeth II (royalty) in 1974",

the appropriate response of which is

- "Age at beginning of 1974: 47 years", and a history correlated with the above data.

Wolfram Alpha does not answer inquiries which require a story reaction, for example, "What is the distinction between the Julian and the Gregorian calendars?" however will answer true or computational inquiries, for example:

- "June 1 in Julian date-book".

Scientific symbolism can be parsed by the engine, which regularly reacts with more than the numerical outcomes. For instance:

- "lim(x->0) (sin x)/x"

yields the right limiting estimation of 1, and also a plot, up to 235 terms of the Taylor arrangement.

3.1.5. Integrating Quora

Quora – is a question-and-answer website where questions are asked, answered, edited and organized by its community of users. The firm was founded in June 2009, and the website was made available to the public on June 21, 2010. Quora aggregates questions and answers to topics.

Users can collaborate by editing questions and suggesting edits to other users' answers as we all know that Quora are the most informative source for any kind of query user can have, say we need to know about joining navy or army. or very common questions like what should i do after completing my masters or anything it might be as it is an open place for users to come in and collaborate questions and answers its a very general and open platform. People can up vote or down vote solutions, and propose edits to existing solutions provided by other users. The Quora neighbourhood includes some well-known people, such as Marc Andreessen or Dustin Moskovitz etc.

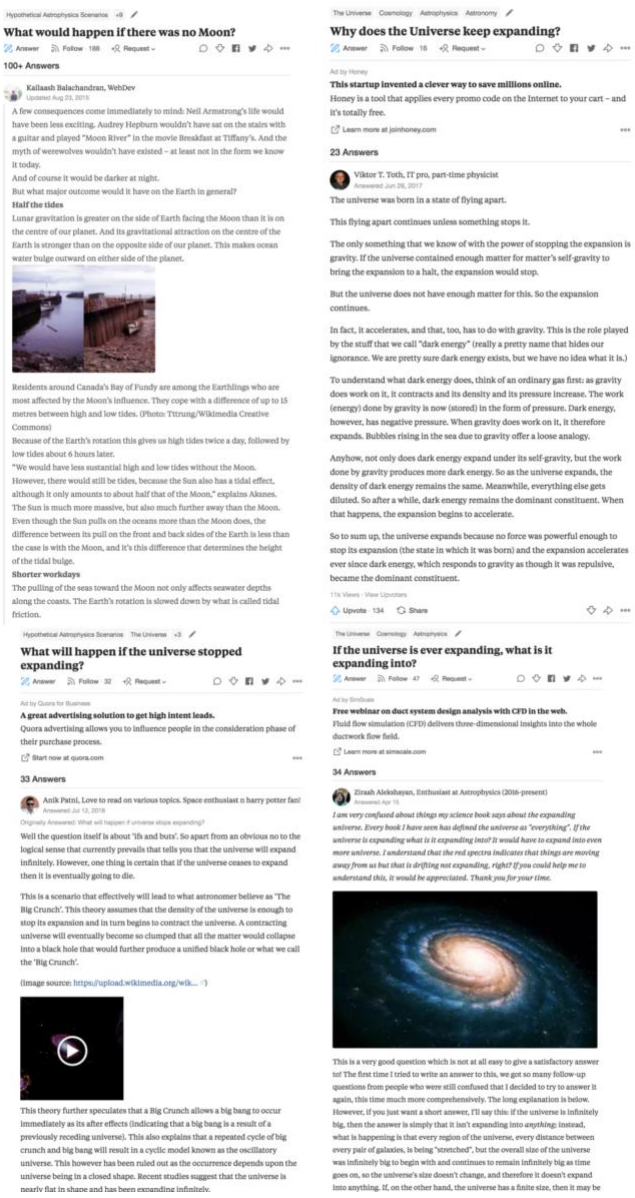


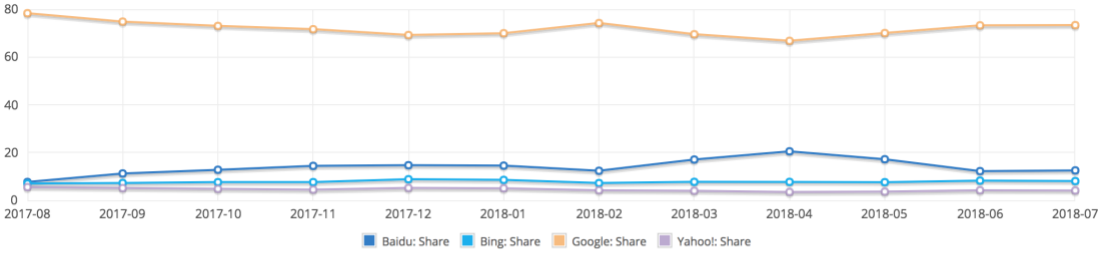
Figure 8 web structure of Quora and examples

Quora has users from all parts of the world and it's not closed to any kind of community. Quora uses the Pylons and Comet technologies for its backend and Ubuntu Linux as its operating system with MySQL as its database. Quora uses Nginx as a mirror proxy server and HAP Proxy for load balancing. Quora has formed its own algorithm to sort results or rank them after the user gives insight to a question asked, which works likewise to Google PageRank. Quora uses Amazon Elastic Compute Cloud technology to host the servers that manage its website. Quora requires users to register with their real names rather than an Internet screen name and the site is essentially unusable if a user is not logged in and using cookies. Users may also log in with their Google or Facebook accounts using the OpenID protocol. here we get the data from Quora by web scraping the actual answer to the question raised by the user via a registered user from Quora through beautiful soup python.

3.1.6. Integrating Google, Bing and Yahoo

A large number of people don't need a dozen web crawlers, particularly individuals who are not technical web users. The vast majority need a solitary web search tool that conveys three key highlights such as Important outcomes the results you are really keen on and what you need. naive to read interface. Accommodating alternatives finalities to widen or mend an inquiry. Search engine Results Pages - SERP are the pages shown via web indexes in case of an inquiry by a user. The primary segment of the SERP is the posting of results that are returned by the web index in case of a watchword question, despite the fact that the pages may likewise contain different outcomes, for example, promotions or restricted results filtered from the provider. Because of the colossal number of things that are accessible or recognized with the inquiry, there is a strategy to limit the results shown t the user on many conditions which may be some degree, typically geographic, similar to state, metro zone, city, or neighbourhoods if not least old and new articles.

the search engine giant Google says that it stores data in interests to those 30 trillion pages in the Google Index, which is presently at 100 million gigabytes. That is around a thousand terabytes, and you'd require more than three million 32GB USB thumb drives to store every one of that information.



Show 10 entries Search:

Search Engine	Share
Google	71.98%
Baidu	14.04%
Bing	7.76%
Yahoo!	4.44%
Yandex	0.92%
Ask	0.33%
DuckDuckGo	0.23%
Naver	0.12%
AOL	0.05%
Dogpile	0.04%

Figure 9 Famous Search engine and their real time historical metrics

As per the most recent market report (January 2018), 74.52% of searches were fuelled by Google and just 7.98% by Bing which is owned by Microsoft. The project will comprise of 4 main engines as Google as a primary candidate to depend for results and rest as backup failsafe as follows:

- Google Search
- Duck Duck Go

- Bing
- Yahoo

the main goal of the project is to understand that they all kind of information is available everywhere using all the available information and organizing a yield which makes more sense to the user. the results from search engine have so much information on the SERP itself as shown in figure the required result website URL can be extracted and used later for text extraction and the description from the SERP page can be used collectively and summarize a meaning full sentence for example.

- **Query: why should you sleep on your right side?**

  Results only the description from the SERP:

  - *On The Right Side And On The Left Side? The side in which you mull over can likewise assume a part in your wellbeing. Thinking about the ride side can intensify acid reflux. In any case, mulling over the left side can put a strain on inward organs like the liver, lungs, and stomach, yet in addition while decreasing heartburn.*

  - *A free, fetal position (where you're your ally and your middle is slouched and your knees are bowed)— particularly on your left side—is awesome in case you're pregnant.*

  - *How Might I Train Myself To Sleep On My Left Side? why you should think about your left side. Maya Borenstein for LittleThings. There are a few simple and powerful ...*

  - *Regardless of whether you're not pregnant (or a lady), mulling over the left side may take some weight off the heart, as gravity can encourage both lymph waste*

  - *You should stick to lying on your left side while anticipating. It enhances dissemination to your developing child and keeps your uterus from squeezing against your liver.*

  - *How you lie in bed decides if you get a decent night's rest. A perfect dozing ... So pregnant ladies should mull over their left sides.*

A summarized version of just the description:

"The side in which you sleep on can also play a role in your health. ... However, sleeping on the left side can put a strain on internal organs like the liver, lungs, and stomach, but also while reducing acid reflux. A loose, fetial position (where you're on your side and your torso is hunched and your knees are bent)—especially on your left side—is great if you're pregnant. ... Even if you're not pregnant (or a woman), sleeping

on the left side may help to take some pressure off the heart, as gravity can facilitate both lymph drainage."

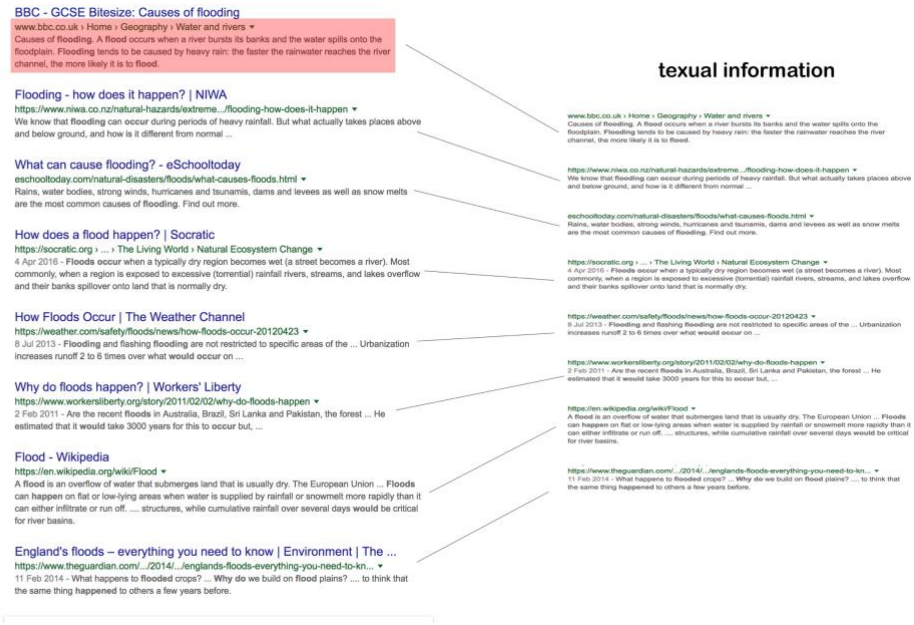


Figure 10 access to textual information on SERP

The URL from the SERP will be used as a list of options to consider to web scrap the textual information to be gathered and summarized to the users.

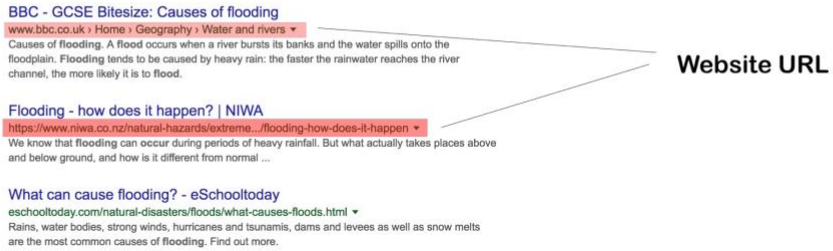


Figure 11 obtaining URL form SERP

3.1.7. Integration of Wiki How

wikiHow is an online wiki-style network comprising of a broad database of how-to guides. Established in 2005 by Internet business visionary Jack Herrick, the site intends to make the world's most useful how-to guidelines to empower everybody on the planet to figure out how to do anything. [21] wikiHow was organized with the objective of making a broad how-to guidebook with exact, up and evolving directions in various dialects. Most how-to articles take after a comparable structure with steps, tips, alerts, a posting of things you'll require, and are supplemented with pictures to enable a user to figure out how to finish a task or a How-to confusion. It is a hybrid association, a revenue-driven organization keep running for a social mission. [21] wikiHow is an open source and open substance venture. The adjusted MediaWiki programming is freely

released and the substance is unpacked under a Creative Commons permit. In February 2005, wikiHow had more than 35.5 million distinct guests. In August 2017, wikiHow contains in excess of 190,000 free how-to articles and more than 1.6 million enlisted clients. On April 11, 2010, a wikiHow article titled "How to Lose Weight Fast" achieved 5 million online visits, a first for the website. "Step by step instructions to Take a Screenshot in Microsoft Windows" is the site's most well-known article.

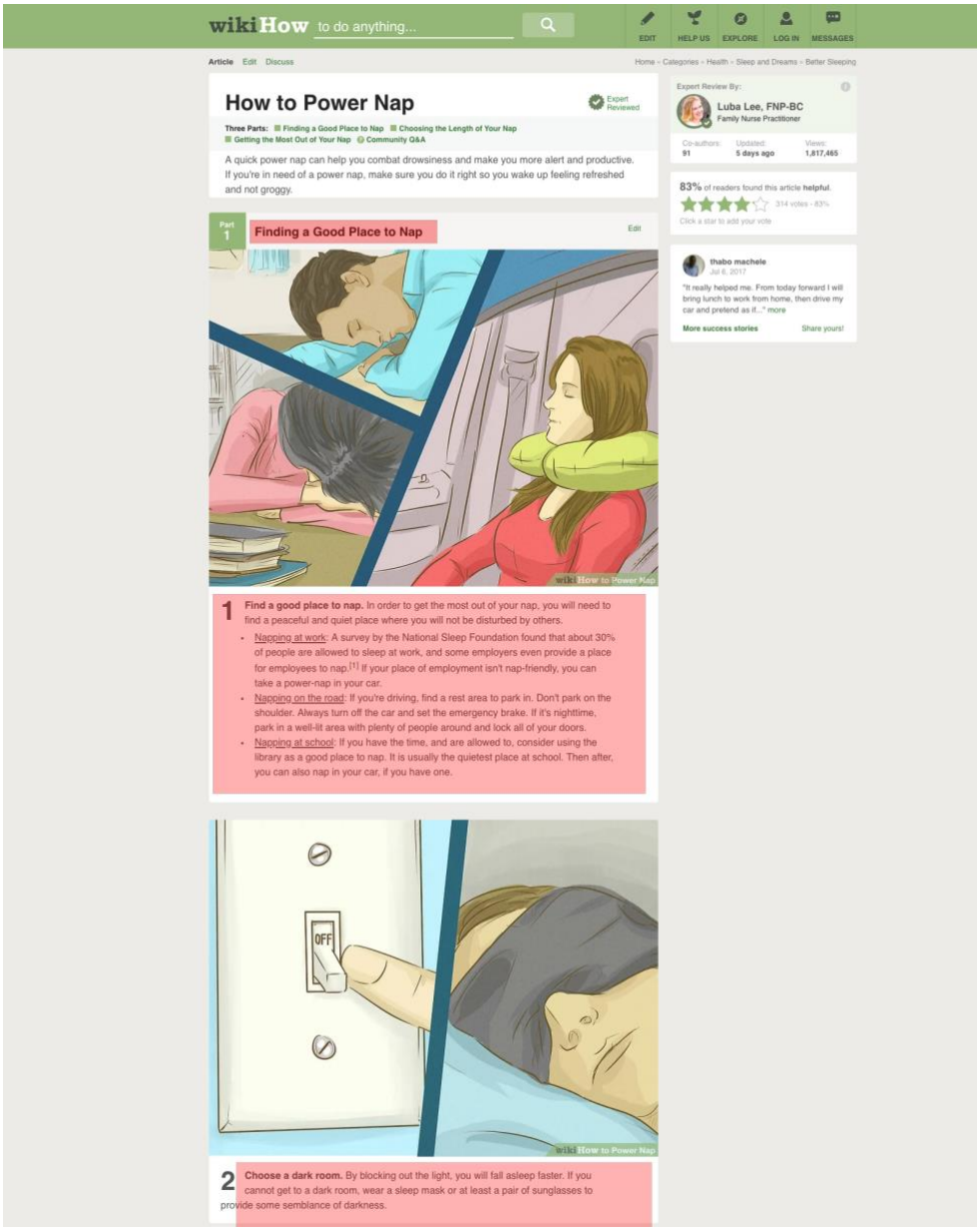


Figure 12 Wiki How web flow and HTML structure

3.1.8. Text extraction and scrapping

Text extraction from HTML pages is one of the crucial components of the web scrapping part of the system. when the query is raised by the user the query is processed and the results from the SERP URL are obtained. The list of URL is then passed to a working flow which looks at each URL and logical selection is made whether to get the results from this URL or not. if the original URL is selected and decided to be processed and obtained the information from the

URL. it is then passed to beautiful soup to accesses the URL using Python URL library with the request and get the raw HTML from the website. the result got back from beautiful soup is very clean and easily structured tree data which can be traversed and the needed information is obtained from the beautiful soup tree data. for extraction of the main core article of the web page all the unwanted tags such as title, header images, Google SEO tags, AdWords and many other unwanted tags from the HTML raw data is dropped and the leftover is now pre-processed with a set of strategies to obtain the core chunk of paragraph which contains the main article. As shown in the figure [13] the main preference is given to the main article text chunk and rest information from the page is discarded. the final result is considered for further pre-processing to fine tune the textual information or even summarize the main article so the final result can be compressed to a smaller paragraph containing all the information what the article has to convey.

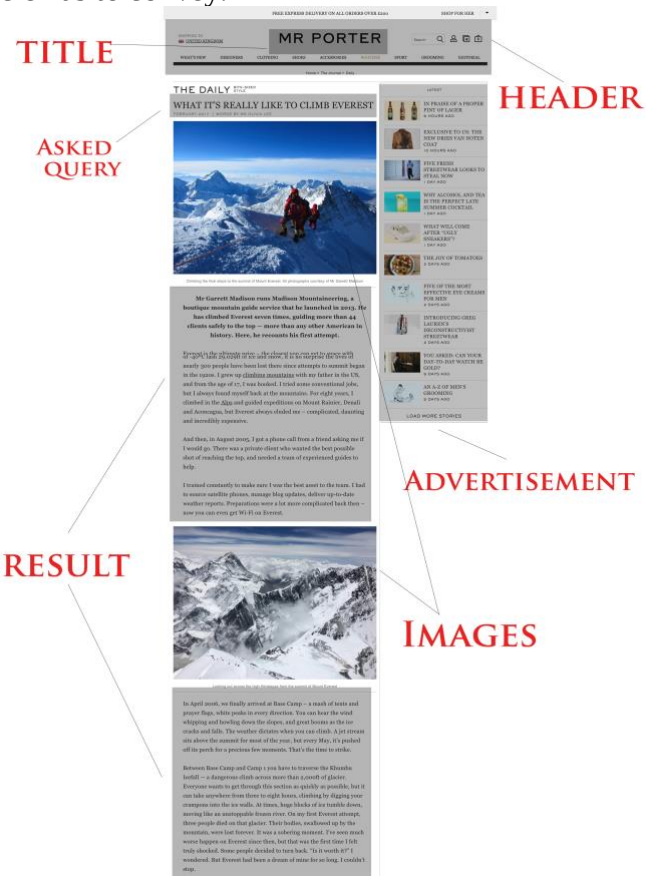


Figure 13 core information extraction from a website

3.1.9. Text summarization

summarization is the way toward shortening a content document or huge chunk of text from the paragraph with programming, to make a rundown of the significant purposes of the original report and still make the same sense as of the original report. Advancements that can make a reasonable summary consider factors, for example, length, composing style, and sentence structure. The content summary is the issue of making a short, precise, and familiar outline of a more drawn out content of record which can be read quickly and obtained

same insight as for the one obtained by reading the original document. Programmed content summarization techniques are incredibly demanded to address the regularly developing measure of content information accessible online to both better help find important data and to devour pertinent data quicker. For the project, NLTK Text summarizer is implemented to perform summarization of any textual information more than 100 words are obtained in the final pre-processing of the system working flow. An example of the system summarizing the results from the text extracted from the HTML web page can be seen as follows:

- Not Summarized but original HTML text data 213 words:

"Populace development is known as one of the main thrusts behind ecological issues on the grounds that the developing populace requests to an ever-increasing extent (non-inexhaustible) assets for its own application. So why precisely does the human populace extend too quickly? To comprehend this, we should first clarify a little about the contrast amongst direct and exponential development, at the end of the day, add a little fundamental math to the condition. Development is typically thought of as a direct procedure: an expansion by a consistent sum over some stretch of time. The new sum isn't impacted by the sum officially present. For exponential development, this is extraordinary, on the grounds that the expansion of a factor is relative to what is as of now there. At the point when cells partition, there will be a consistent multiplying of the cells effectively present. As far as populace development, the quantities of individuals officially exhibit dependably impacts the number of kids conceived in any nation. It is anyway not a basic matter of a consistent multiplying of the sum. Different components, for example, fruitfulness and death rates, impact populace development, and the sex and time of individuals effectively present, and balanced choices impact regardless of whether individuals will really have at least one youngsters."

- NLTK Summarized version of the above text chuck 67 words from 213 words**:**

"Populace development is known as one of the main thrusts behind ecological issues on the grounds that the developing populace requests to an ever-increasing extent (non-inexhaustible) assets for its own application. ... Different components, for example, fruitfulness and death rates, impact populace development, and the sex and time of individuals effectively present, and balanced choices impact regardless of whether individuals will really have at least one youngsters."

3.1.10. Core of Backend

The query is passed to a query processor which tries to identify if the query is a question or a just a conversational sentence or depth analysis like NLTK named entity recognition and percentage of question type in the query is found out and a decision is made whether to pass the query to Chatbot or the web scrapping flow as shown in the figure [14].

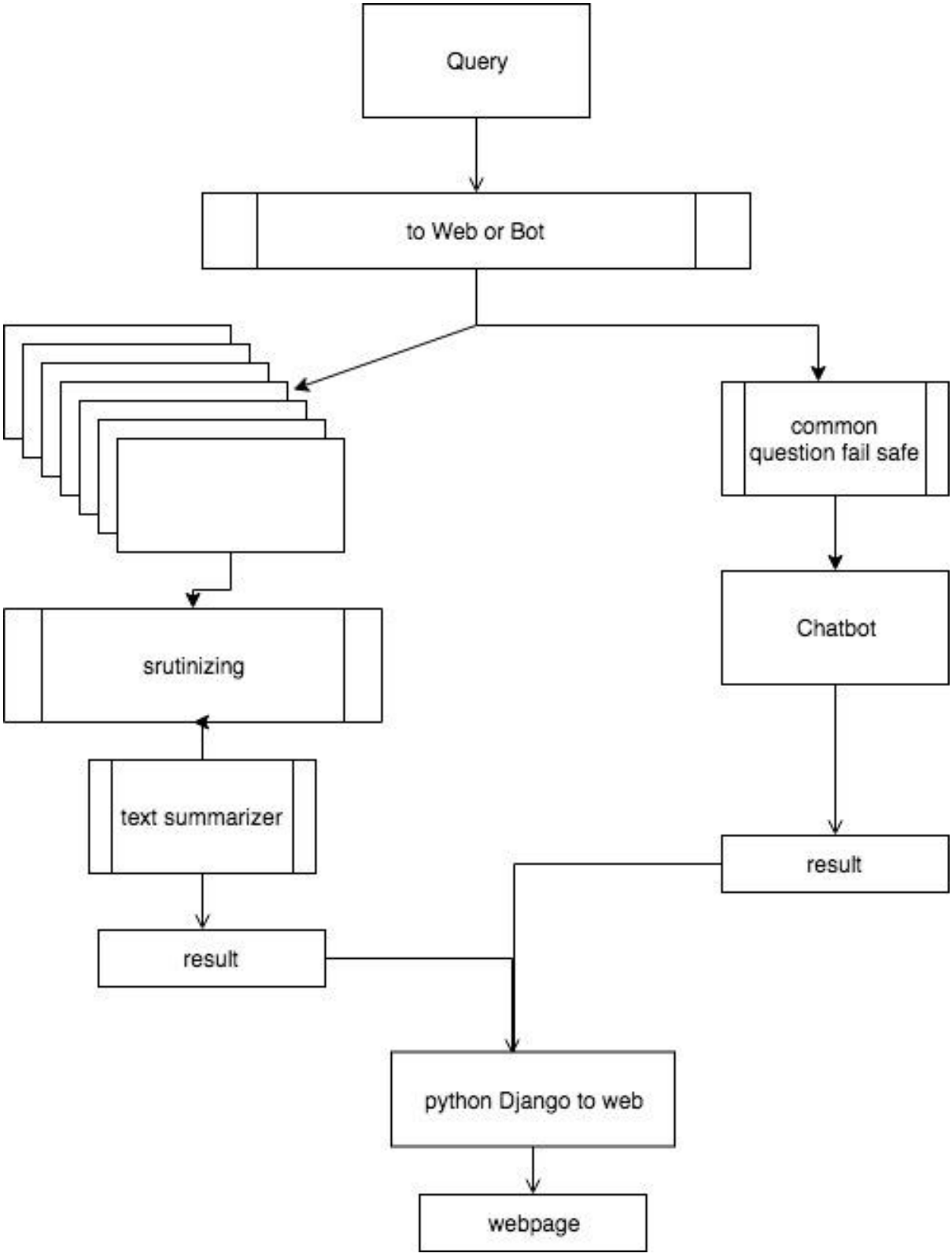


Figure 14 working flow of core system

3.2. Frontend Technologies

3.2.1. User view Home page for the System

A user interface is created to simulate a assistance like service. The system is able to accept a query from the user in form of voice. which will be accepted using Google Web speech API and converted to text. the obtained text is passed to backend core working flow of the system via the Django framework which acts as a two-way connection bridge. later the summarized result from the core system is passed back to the web page as a response to the query asked by the user. the response is completely handled in the website and displayed and visualized in a neat way so the user can process the information.

Figure 15 web view of system

3.2.2. JavaScript and angular JS

JavaScript and angular JS framework along with HTML5 and CSS are used to build the web home page Vanilla JavaScript is very powerful and facilities easy use of inbuilt features to make very beautiful and powerful website along with angular JS from Google both running on V8 engine build by google. The angular HTTP request feature is used to talk to Django to send the query back to the backend core system and receive information back from the system via the request.

3.2.3. Speech recognition and generation

This is an innovative technology framework provided to the user in the browser to empower application to do more than ever before. this initiative is from Google and works well in google chrome browsers along with the v8 engine of Google. The SpeechSynthesis feature is a read-only property of the browser Window object which only returns a SpeechSynthesis object, which is the entry point into using Web Speech API. This can be used in the front end of the system to accept the voice

utterance from the user device microphone and process the information through the web speech API which is having a very high recognition percentage. this feature then spits out a text of what the user had to say and this is accepted as input. later the text back from the system is given to the SpeechSynthesis which can generate the voice from the text. the generated voice is now echoed back to the user which is the result of the query asked by the user.

4. Evaluation and Testing

As the model is a collective system of many components in place which is facilitating the system to work as a single system for the evaluation of the working of the model each component is stripped apart for testing.

All the components making up the core system are tested independently with trial an error method. Each component of the system expects a input and gives an output. all model has the same goal to fetch information one or other way and yield the result back to the core system. the components are tested with various random queries and the stability of each component is noted down. The system was made stable and also hosted on website power by heroku with preferred URL. So, people can be invited to test the system as it's a very general system which serves to get answers to any query asked by users. The behaver of the system was monitored for each questioned asked by the user

4.1. Chatbot

The Chatbot is only 55% accurate to use the proper words to form the sentence when the query is asked for the sake of testing the sentence, not in the dataset was asked to the bot which surprisingly made a comeback with 50-70% sensible answers most of the time and 100% meaningful sentences for few questions. few questions and answers as of the testing are listed below:

- *Can You Be Friends with Me*
- *I Will Have to Ask I Am*

- *Shall We Dance*
- *They Are Looking*

- *We Are Friends*
- *Okay We Are*

- *You Are A Very Bad Person*
- *Oh, I Have Got Any Bad*

- *You Are Very Smart*
- *Well I Am Afraid Not At The House You Have*

- *The Police Are Coming*

- *You Do Not Know I Are*

- *So, What You Going to Do It*
- *We Do Not Know. I Do Not Know*

4.2. Wikipedia, Wiki how and Quora

As Wikipedia and wiki How and Quora are pure web scrapping and the HTML of the page and the structure do not change substantially. the web scrapping algorithm has to written once for this static page which do not change the structure of the HTML more frequently and the API are very stable they can be left independent to perform the task of fetching the data for the asked query.

4.3. wolfram alpha

wolfram alpha is a very powerful tool as SIRI itself completely depends on Wolfram Alpha results to serve to users. query from many fields can be asked an interesting combination of question can also be framed and asked to this framework few examples such as:

"which is Third Tuesday of April" or even "what happened on Easter 1910".

The complex question like "photon energy of a 200nm" collaborative questions like "famous people named John". after testing for many iterations, the result from wolfram alpha were useful only for short factual question and not a general question. which is very useful to handle complex computation short answer question all by itself where the general complex question can be handled by the normal flow.

4.4. Query with no results

Query with no result has to be handled with fail safe back up responses from the conversational system as not giving back anything will make the user feel the system is not working or it's broken. set of hard filters are set in place to tackle such situation with responses or match the user query to set off the most common question asked by the user to Chatbot and facilitate the answer back to the user. For example:

- 'Q': 'Where are you?',
- 'A': ['in the cloud, next to the flying castle',
  'near the flying castle, in the cloud']

- 'Q': 'How can you help me?',
  - 'A': ['ask me anything',
    'ask me factual questions']

- 'Q': 'Which languages do you speak?',
  - 'A': ['I manage English for now', 'English']

4.5. Comparison with present System

Comparison with present System personal assistance device SIRI from apple with the implemented working system.

When query such as factual question asked by Siri which runs on wolfram alpha and few predefined answers perform very badly and can only perform task such as assisting to set up alarm call a friend or even send a text message to a friend.

When asked “who is the inventor of bitcoin” which is a factual question but the system fails to even fetch that from wolfram alpha or any other source but gives set of google search results of the asked question as shown in figure [16]

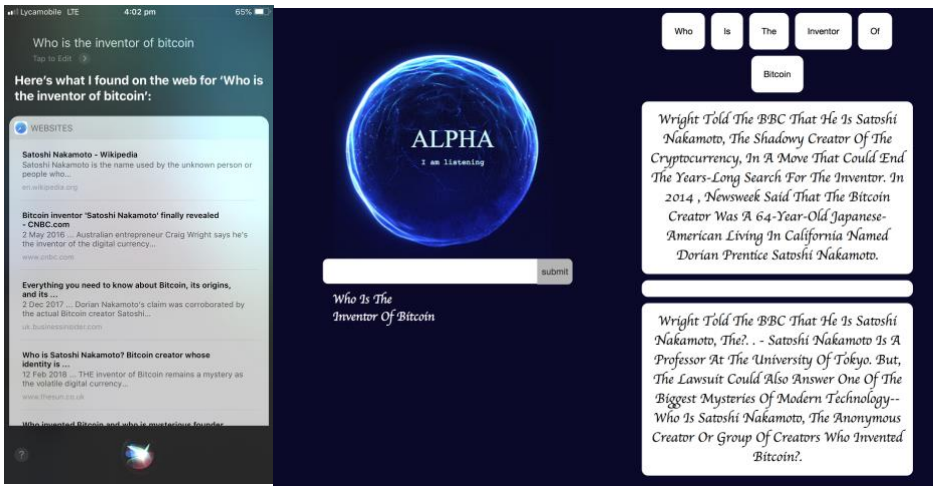


Figure 16 comparing built system with present systems

When asked "How many bottles of water should i drink to be healthy" this is a very general question and has many reasons to be an important question. the user is asking a factual question and also an opinion based question which will be very useful to the user with the proper answer is given back to the user. the present system such as Google and Siri try to direct us to the website 90% of the time and Google is recently started to fetch information from the website by itself and try to provide to the user but its limited to set of question or types of question.

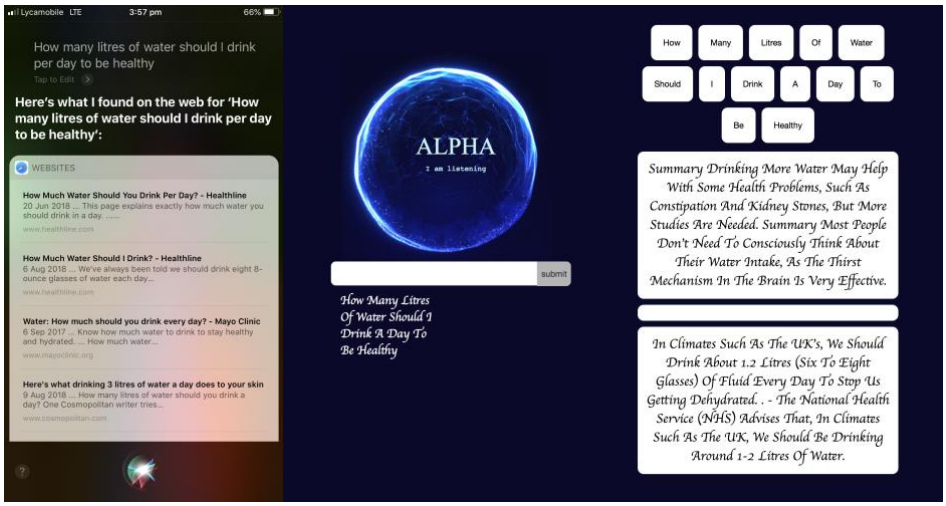


Figure 17 comparing built system with present systems 2

An example such as "how to become rich" is a very curious and infamous question to ask any chatbot or conversational engine in this case Siri completely mistakes the question and provides information from Wolfram Alpha. which is completely interleaved. whereas the system fetched relevant information and summarized it to the user.

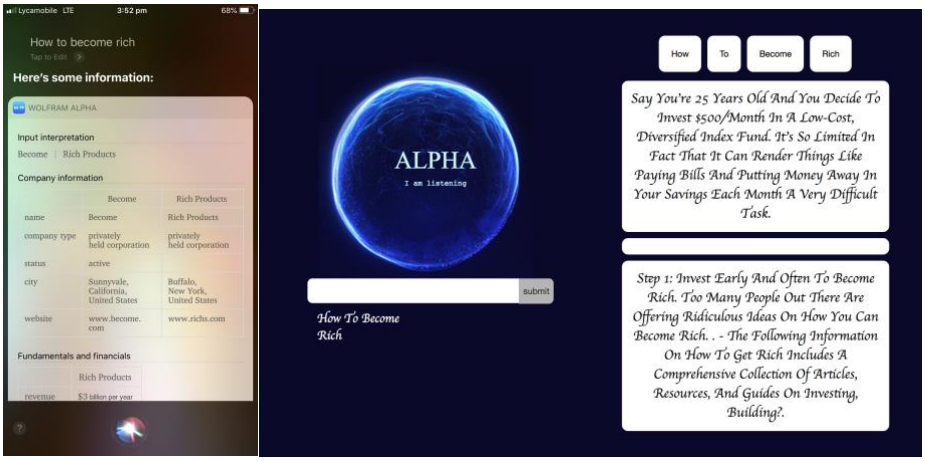


Figure 18 comparing built system with present systems 3

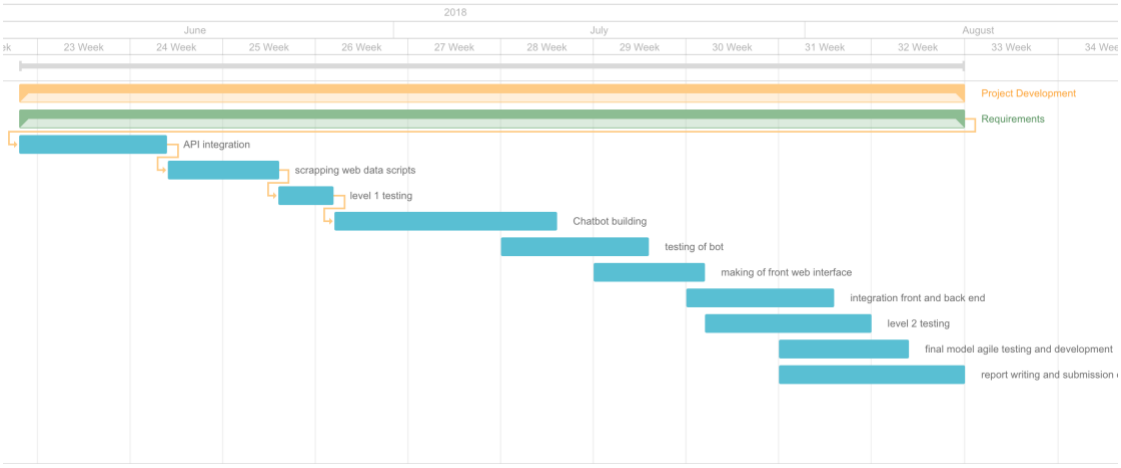
5. Discussion

The project trying to solve the general domain query is a very challenging problem. As the main goal was to make it for general purpose use and not domain specific had many pros and cons. basically it was not very difficult to get the innovative idea to find sources to fetch information from and integrating them. later testing each component of its positive side and how much the component can add value to the system to make the general approach more stable. if the goal was to make a system fixed in a domain of constraints for example of health sector or technical sector or even a customer care agency the implementation would be completely different and more complex. to find source only from the same domain which is open source and has no legal obligation to get the data for public use. the training of Chatbot would be more complicated as proper data set containing the domain dialect had been gathers for the bot to learn the conversation. the implementation of the Chatbot was the hardest part of all components next to data scraping from HTML.

The tropical understanding of how a seq2seq deep neural network works and how LSTM can be used to boost the decreasing gradient of the model by using short memory flips. the small 100 sample example experiment was very useful get insight on planning how to train the model with a large dataset. Also realizing to break the sentences into smaller sentences was a helpful insight from the trial and error experiments. next step was to train with 1000 or 10,000 samples which took more than 8 to 10 days of daily training and testing the model by end of the day. trying to find the needed resource to train the model was the most difficult part. As could computing and GPU were very expensive and pay per hour policy usage. each trail was run for 300 to 400 epochs with 10,000 samples which took more than 10 hours and the results were tested by asking a random set of question to the model each time. the next model had changes and feedback from the previous model and paper parameters where tuned for the new model and a final model was selected. the final stable model was trained for 2500 epochs for

6 days. which could only give 55% accuracy to the Chatbot which is not completely ready to frame all words in the sentence.

The agile methodology was used as planned for the project to be built. As agile methodology was quicker resolve for the problem statement of this kind. the feedback from each trail testing was used as a feedback and changes were implemented and tested again.



As there are so many Personal assistance systems in the market with state of art technologies such as "OK Google", SIRI, Alexa and many more. which focus on assisting the user with a much cross-domain application such has booked an uber, or even booking a personal dining table by calling the restaurant a new feature from Google. but as a knowledge perspective or a system which can mimic true intelligence or act like a holy grail and answer any question asked by the user is one of its kind and no other system is currently discussed to perform in such a manner. By upgrading the implemented system to handle personal assistance and more complex conversation situation it can overcome the present state of the art system and perform better by entertaining the user by always answer the question. Flipping between the Chatbot mode and information retrieval model is the plus point of the system.

6. Conclusion and Future work

The system was designed and implemented to achieve one goal to fascinate the user by providing answer any sort of questions asked by the user and also trying to connect the world wide web rich knowledge to be available at your voice request. This can also facilities another goal of applications or domain of people. such as restricting the system to be a domain-specific this can be a powerful assistance device and also be an entertaining platform for the user to talk t the system. or the people who are visually disabled can take full power fo this system to talk to and bet any information need instantly. the main use case can be a technical person who has the thirst to know something and always searching web for more and more information like a teacher hunt this can be a perfect platform to get all the required information. the main challenges faced while implementing the system are the Chatbot and the web scrapper. which were overcome with intense trial and error method testing of each technique. the final project showed sign of performance better than any assistance device available in the market.

The project was implemented and displayed on a website due to the limitation of time and complexness of making a unique portable hardware capable enough to run the project on its CPU, devices such as Raspberry Pi which is the size of a credit card can also be used. which is easy to use easy to set up and weight around 30 grams but completing the system to be self-sustained and always connected to the internet will make it more complicated and fails to be truly portable. so better suitable hardware solution has to be discussed as the project focus on more complex hardware configuration to achieve a high-end light portable system.

Technologies like projector screen wristband which are many readily available in the market for an example the prototype as Hi-tech wristwatch technology. This will use a miniature 'Pico projector' and eight miniature contiguity sensors to replicate a picture of your smartphone screen on to your arm. Low energy Bluetooth will interact with your mobile device or any smart device talking up and down to transfer information. A Wi-Fi component will connect you to the network available to the system in both ends.

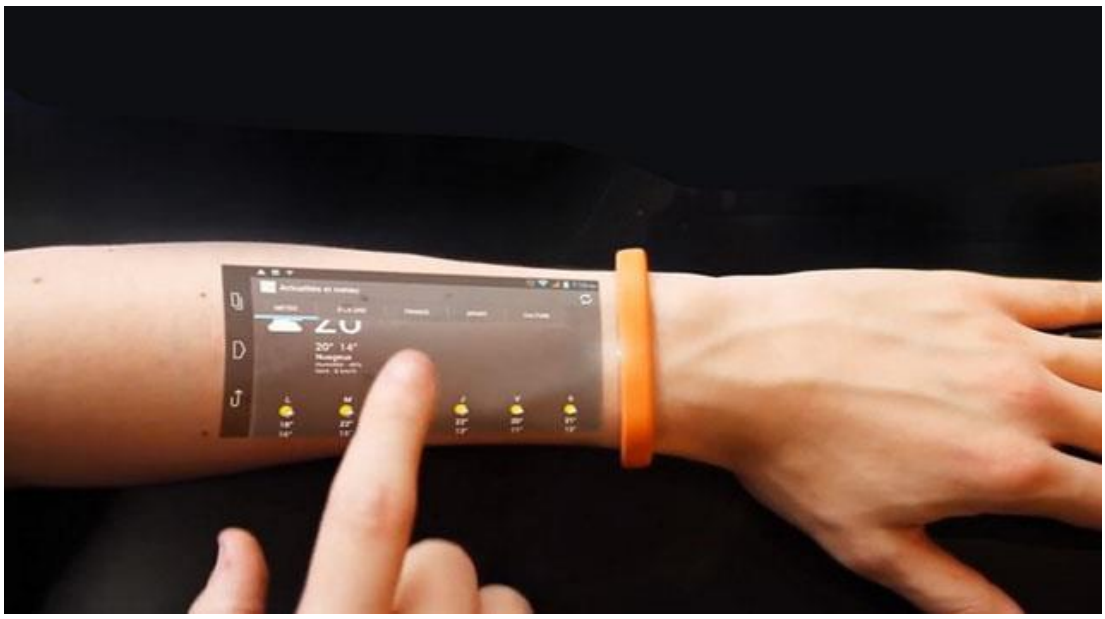


Figure 19 visualization of wrist screen projector

The above-described device can be considered to be used as a display hardware for the project like a head-up display to show all the important information related to the query asked by the user. Including news, images, and videos related to the query raised by the user and for the computational part of the system we have to consider a system with a cutting-edge design which can talk to server up and down. which can also have an efficient way of connecting to the internet. the main goal of the hardware is to talk to the server which holds the core computational system in the cloud.

Figure 20 abstract for cloud computing

Cloud computing is a topic that many find confusing. In fact, most of those who claim not to understand the subject is actually part of the majority that uses it daily. In basic terms, cloud computing is the phrase used to describe different scenarios in which computing resource is delivered as a service over a network connection. Cloud computing is, therefore, a type of computing that relies on sharing a pool of physical or virtual resources, rather than deploying local or personal hardware and software. So such a system can hold the core computation of the system and only talk to the hardware as a listener. this hardware can be a cutting edge in earphone types.

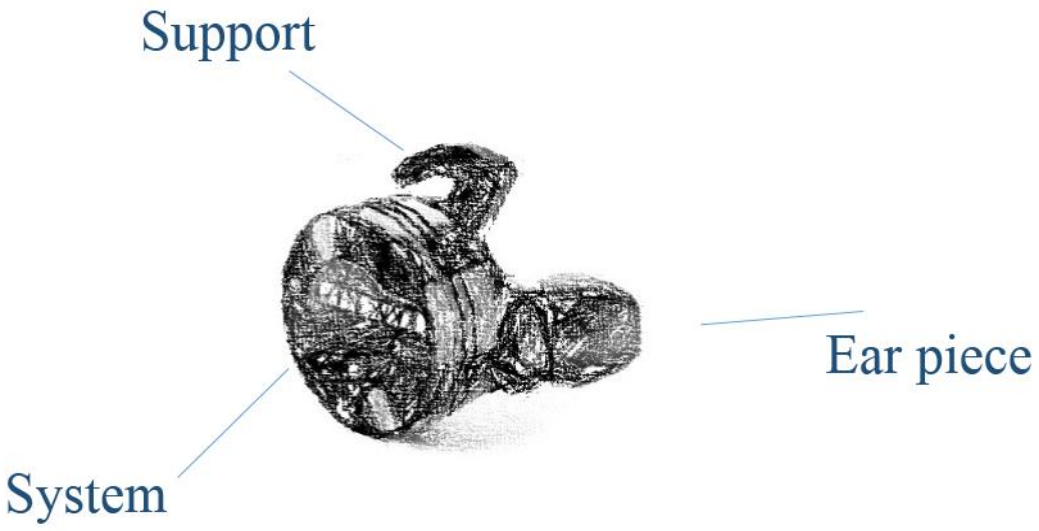


Figure 21 wire Frame for discussed high end device

Single platform brings in the hardware and computational part to accept the voice input as speech to text and pass the input to cloud servers to process the data to compute query asked by the user and then answer and receive the data as text to speech back to the user with visual display in the smartwatch or wristband with projector and get the voice through earpiece for the user. by this method, the only portable system carried with the user will be the small lightweight earpiece providing complete hands-free communication to the system connecting the user to endless knowledge and information.

Figure 22 collective working of all components via cloud visualization

7. Reference

[1] S. Hochreiter and J. Urgen Schmidhuber, "LONG SHORT-TERM MEMORY," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] I. Mani, *Automatic Summarization*, vol. 3. Amsterdam: John Benjamins Publishing Company, 2001.

[3] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the Special Issue on Summarization," *Comput. Linguist.*, vol. 28, no. 4, pp. 399–408, Dec. 2002.

[4] S. Chakrabarti, *Mining the Web : discovering knowledge from hypertext data*. Morgan Kaufmann, 2003.

[5] B. Galitsky, *Natural language question answering system : semantic headers*. Australia: Advanced Knowledge International, 2003.

[6] B. Galitsky and R. Pampapathi, "Can many agents answer questions better than one?," *First Monday*, vol. 10, no. 1, Jan. 2005.

[7] D. Glez-Peña, A. Lourenço, H. López-Fernández, M. Reboiro-Jato, and F. Fdez-Riverola, "Web scraping technologies in an API world," *Brief. Bioinform.*, vol. 15, no. 5, pp. 788–797, Sep. 2014.

[8] Y. Kim, "Convolutional Neural Networks for Sentence Classification," Aug. 2014.

[9] S. A. Abdul-Kader and J. Woods, "Survey on Chatbot Design Techniques in Speech Conversation Systems," *IJACSA) Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 7, 2015.

[10] S. M. AL-Ghuribi and S. Alshomrani, "Bi-languages Mining Algorithm for Extraction Useful Web Contents (BiLEx)," *Arab. J. Sci. Eng.*, vol. 40, no. 2, pp. 501–518, Feb. 2015.

[11] B. Batrinca and P. C. Treleaven, "Social media analytics: a survey of techniques, tools and platforms," *AI Soc.*, vol. 30, no. 1, pp. 89–116, Feb. 2015.

[12] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, "Semantically Conditioned LSTM-based Natural Language Generation for Spoken Dialogue Systems," Aug. 2015.

[13] O. Vinyals and Q. Le, "A Neural Conversational Model," Jun. 2015.

[14] D. Hingu, D. Shah, and S. S. Udmale, "Automatic text summarization of Wikipedia articles," in *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, 2015, pp. 1–4.

[15] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, "Deep Reinforcement Learning for Dialogue Generation," Jun. 2016.

[16] A. Mishra and S. K. Jain, "A survey on question answering systems with classification," 2016.

[17] K. Zhao and C. Wang, "Sales Forecast in E-commerce using Convolutional Neural Network," 2017.

[18] M. Sorostinean, K. Sana, M. Mohamed, and A. Targhi, "Sentiment Analysis on Movie Reviews," 2017.

[19] T. Capes *et al.*, "Siri On-Device Deep Learning-Guided Unit Selection Text-to-Speech System," 2017.

[20] K. K. Bowden, S. Oraby, A. Misra, J. Wu, and S. Lukin, "Data-Driven Dialogue Systems for Social Agents," Sep. 2017.

[21] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading Wikipedia to Answer Open-Domain Questions," *arXiv*, 2017.

[22] G. López, L. Quesada, and L. A. Guerrero, "Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces," Springer, Cham, 2018, pp. 241–250.

[23] L. Ozaeta and M. Graña, “A View of the State of the Art of Dialogue Systems,” Springer, Cham, 2018, pp. 706–715.

[24] F. J. de Cos Juez *et al.*, Eds., *Hybrid Artificial Intelligent Systems*, vol. 10870. Cham: Springer International Publishing, 2018.

[25] T. Liu, B. Wei, B. Chang, and Z. Sui, “Large-Scale Simple Question Generation by Template-Based Seq2seq Learning,” Springer, Cham, 2018, pp. 75–87.

[26] R. Zhang, Z. Wang, and D. Mai, “Building Emotional Conversation Systems Using Multi-task Seq2Seq Learning,” Springer, Cham, 2018, pp. 612–621.

[27] P. Haeusser, T. Frerix, A. Mordvintsev, and D. Cremers, “Associative Domain Adaptation.”

[28] Z. Yan *et al.*, “DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents,” pp. 516–525.

[29] I. Sutskever, O. Vinyals, and Q. V Le, “Sequence to Sequence Learning with Neural Networks.”

[30] R. Mitchell, *Web scraping with Python : collecting data from the modern web*. .

[31] “Ten impressive, weird and amazing facts about Wikipedia | WIRED UK.” [Online]. Available: http://www.wired.co.uk/article/ten-fun-facts-wikipedia. [Accessed: 22-Mar-2018].

[32] J. E. Kelly Iii, “Computing, cognition and the future of knowing How humans and machines are forging a new age of understanding.”

[33] P. Guo, Y. Xiang, Y. Zhang, and W. Zhan, “Snowbot: An empirical study of building chatbot using seq2seq model with different machine learning framework.”

[34] I. Androutsopoulos, G. Ritchie, and P. Thanisch, “Masque/sql{ An EEcient and Portable Natural Language Query Interface for Relational Databases.”

[35] G. López, L. Quesada, and L. A. Guerrero, “Alexa vs. Siri vs. Cortana vs. Google Assistant: A Comparison of Speech-Based Natural User Interfaces,” Springer, Cham, 2018, pp. 241–250

[36] T. Capes *et al.*, “Siri On-Device Deep Learning-Guided Unit Selection Text-to-Speech System,” 2017.