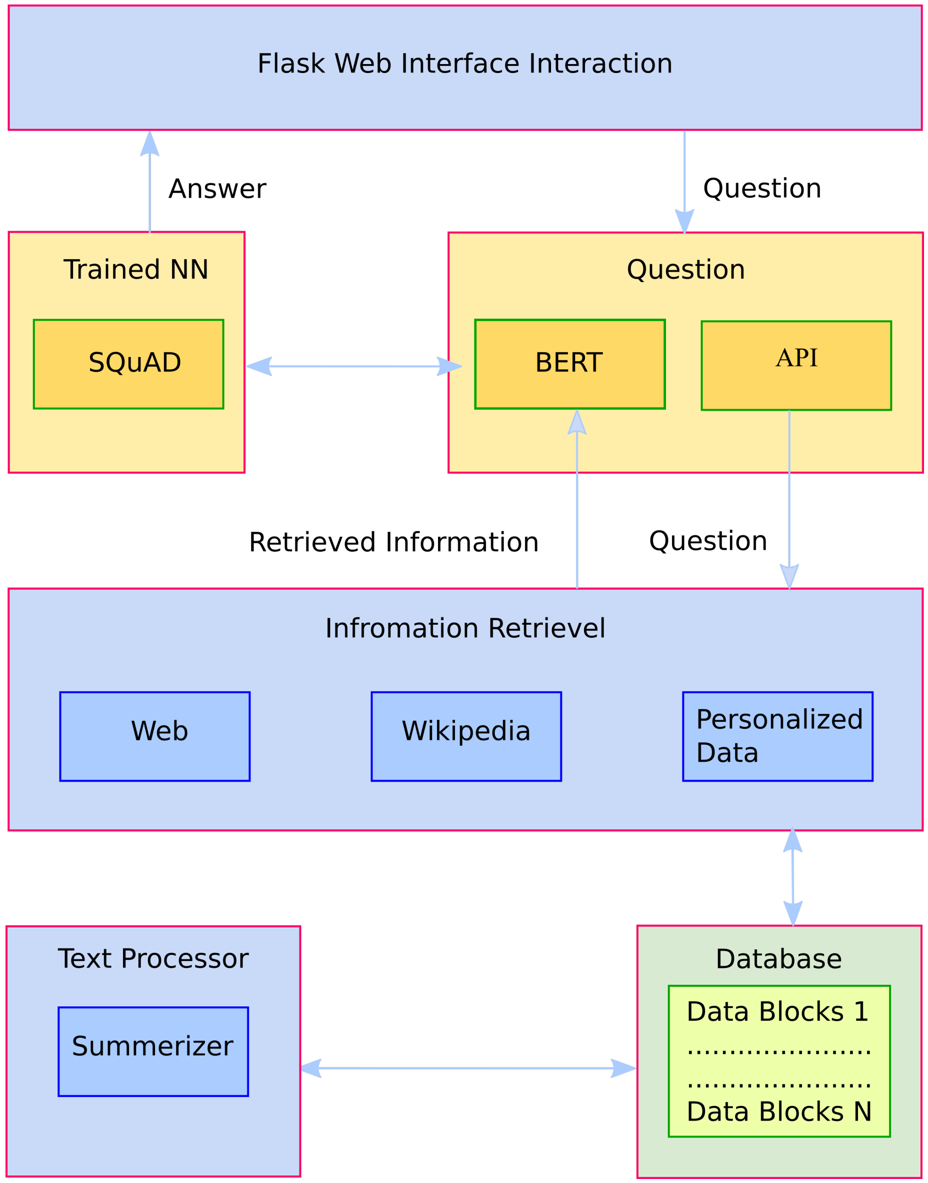
Graphical Abstract

A Chatbot Design using Deep Learning based NLP

Jason Pandian, Dr. K. Krishneswari



Highlights

A Chatbot Design using Deep Learning based NLP

Jason Pandian, Dr. K. Krishneswari

- A design for deep learning based chatbot has been presented.
- This chatbot can handle open-domain and domain-specific questions.

A Chatbot Design using Deep Learning based NLP

Jason Pandian[a], Dr. K. Krishneswari[a]

[a]*Department of Computer Science and Engineering, Tamilnadu College of Engineering, Anna University, Coimbatore, 641659, Tamilnadu, India*

Abstract

Chatbots have recently become a significant application in industry. People of this decade use web-based technology more than ever before. In the business part, the companies and organizations use skilled persons to keep communicate with their customers and solve their queries by feedback. As it was literally involved by humans effort. In organizations, skilled persons spend their time by answering a lot of questions to their customers. To avoid those problems and reduce human involvement and time, a pre-trained chatbot can be used. The chatbot system is cost-efficient and it will need no rest and retirement. In this work, we will address a hybrid model of a chatbot using BERT based deep learning network that can respond to either Domain-specific or Open-domain questions.

Keywords: NLP, IR, Machine Learning, Deep Learning, AI

1. Introduction

It will be difficult for the user to achieve particular information directly from search engines, where they got many results to parse manually. Getting specific information about a product from an organization takes a long time by a normal human to human conversation and even while browsing the web site of the organization or information booklets about an organization. Information Retrieval (IR) and Natural Language Processing (NLP) are important technologies dealing with this problem.

1.1. Motivation

Most of the human advancements of this century were only because of the development of communication technologies. IR and NLP based technologies. Most of the available solutions are proprietary technologies so one can not get open access to understand the internal mechanisms of such NLP technologies. This motivates me to do a custom 1 AI Chatbot using Deep learning-based NLP technologies. Furthermore, in this decade, there is a lot of research activity in these NLP technologies because of the recent growth in AI and deep learning networks. Since NLP is the future of the internet and human communication, it is motivating me do take a small step in that direction.

1.2. A Brief Survey on Related Works

In [1], Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova from Google AI Language introduce a new language representation model called BERT, which stands for Bidirectional Encoder Representations of Transformers. The pretrained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. BERT obtains new state-of-the-art results on natural language processing tasks. This model could not handle large text blocks during training and testing. So a chatbot using this model can not handle large text blocks for training and testing.

Transformer-based models are unable to process long sequences due to their self-attention operation, which scales quadratically with the sequence length. To address this limitation, [2] Iz Beltagy, Matthew E. Peters, and Arman Cohan introduce the Longformer with an attention mechanism that scales linearly with sequence length, making it easy to process documents of thousands of tokens or longer. This model can accommodate large text blocks during training and testing in a very memory efficient manner. So this model can be adopted to implement a good AI chatbot.

In [3], Amit Mishra and Sanjay Kumar Jain surveyed QASs and classify them based on different criteria. They identify the current status of the research in each category of QASs and suggest the future scope of the research.

In [4], Mohammad Nuruzzaman and Omar Khadeer Hussain conducted an in-depth survey of recent literature, examining over 70 publications related to chatbots published in the last 5 years (before 2019). Based on that literature review, that study made a comparison from selected papers according to the method adopted. This paper also presented why current chatbot models fail to take into account when generating responses and how this affects the quality of conversation.

1.3. Disadvantages of the Previous System

Classical Chatbot designs that are based on early NLP techniques are not efficient. They can not provide a specific answer to a question. Generally, they are implemented for a specific task with limited domain knowledge. Further, they can not provide answers to Open-domain questions. Since early QA Chotbot models may use old machine learning algorithms they can not handle complex QA tasks.

1.4. Proposed System & its Advantages

It will provide appropriate and selective answers from the web so the user need not filter or parse bulky search results. The system can handle Open-domain questions. So it can predict answer almost every time. The Open domain search was fully focused on client privacy because it uses Duck-Duckgo and Wikipedia APIs for search. It Can be easily implemented and customized for a specific task in a specific application domain It is cost-effective and well suited to work in every browser. The system can answer user queries with chit-chat answers, which reduces the traditional knowledge gathering/learning time. The system uses DL based NLP technology to minimize the bot misunderstanding so that the answers will be more specific than traditional systems.

1.5. Project Objectives

The objective of the project is to create a question answering system that can answer the queries based on the DuckDuckgGo search engine and Wikipedia and also provide a personalized chatbot for a particular organization to provide a quick, reliable and direct answer for customers questions from an information base.

1.6. General Constraints of the Project

Our implementation of this Chatbot supports the English language only. Generally, a Chatbot can not hold the previous conversation held with the user to understand the context of the discussion. However, in our implementation, we will instruct the chatbot to set the context before the QA session. In our implementation, we use remote cloud resources for computation. So that this implementation of Chatbot consumes time while processing a type of responses. In our implementation, Chatbot can't perform math operations.

1.7. Overview of this Paper

This section provides a minimal introduction to NLP and DNN. The next section will present a brief history of NLP. Section III will explain artificial Neural Networks and Deep Neural Networks in detail. Section IV will present some popular DL based NLP systems. Section V will present the comparison of the performance of those popular DL based NLP systems. And finally, section VI of the paper will conclude with a brief conclusion.

2. Background Study

2.1. Natural Language Processing

NLP is one of the interesting and useful applications of Artificial Intelligence. In simple terms, an NLP system will try to mimic the ways of the human language understanding model to make sense of textual or speech content. Generally, this kind of NLP tasks will require the system to have the capability to translate, analyze and synthesize the language to make sense out of it.

Humans can effortlessly process both textual or speech content and understand them. The main tasks of an artificial NLP system are to replace the human aspect of understanding with a machine aspect of understanding and make the machine understand the natural language input and act accordingly. Mimicking the process of understanding natural languages will be difficult to develop because of the numerous ambiguities and levels of contextual meaning involved in natural language. The inherent ambiguity of a language makes the NLP difficult to understand it from the perspective of a machine. There are five main categories into which language ambiguities fall: syntactic, lexical, semantic, referential, and pragmatic[5]. These five aspects of language make it difficult to design a machine to act as a human to understand the language content.

2.2. A Brief History of NLP

In [6], Karen Sparck Jones did a detailed review of NLP techniques from their origin in the late 1940s to the early 1990s. The review identified and clearly distinguished four phases of NLP history based on the characteristics namely 1) emphasis on machine translation, 2) by the influence of artificial intelligence, 3) by the adoption of a logico-grammatical style, and 4) by an attack on massive language data. She defined the first phase of work in NLP as lasting from the late 1940s to the late 1960s, the second from the late 60s to the late 70s and the third to the late 80s, and the fourth phase from the late 80s to the late 90s. In the previous paragraph, Karen Sparck Jones' survey covered and classified almost 40 to 50 years of NLP history. Our survey extends her work by adding a fifth phase of NLP history which covers the remaining 30 years of NLP history until now.

2.3. Artificial Neural Networks (ANN)

Traditional ANNs or simply NNs are simple mathematical structure that generally denotes neurons with less number of hidden state. Hence, ANN can process entry-level tasks. In a lot of earlier works, the Perceptron network and BPN were used to solve different kinds of Image processing and classification tasks.

2.4. Deep Neural Networks(DNN)

DNN was inspired by the human brain and tries to mimics the functions of the human brain. A DNN is a network with a large number of hidden states. These hidden states (layers of neurons) can be used to process multidimensional inputs and can be tuned for complex classification tasks.

2.5. Recurrent Neural Network(RNN)

The word "recurrent " means repetition, by processing the sequential data recurrently and storing its past sequences, makes RNN works like a looped(feedback) network. Thus RNN can make an accurate prediction with temporal data-based applications. It was the key to the development of powerful models like LSTM and GRU networks. The vanishing gradient problem was the major problem in RNN. From the perspective of NLP, an RNN can handle only small text sequences(RNN has short-term memory). It is hard to train long sequences using normal RNN. Further, training an RNN requires a large amount of cost and time. RNN is the foundation of different NLP models which are available today.

2.6. Convolutional Neural Network (ConvNet/CNN)

A CNN is a feed-forward Deep Learning network that passes inputs only in one direction, forward, from the first convolution layer, through all other convolution layers, and to the fully connected layer, and output is obtained expresses various aspects in the input. A convolution is an operation that changes two functions into a new function. The pre-processing required in

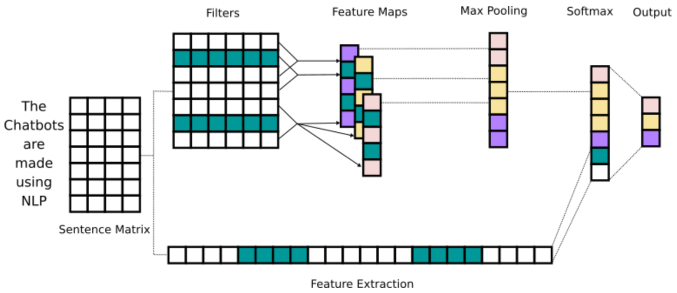


Figure 1: A CNN based NLP task.

CNN is much lower as compared to classical classification algorithms. While in primitive methods filters are hand-engineered, with enough training, on the other hand, CNN has the ability to learn these filters/characteristics on its own.

CNN needs no manual pre-processing. The CNN filters are not defined instead the value of each filter is learned during the training process itself. CNN executes hierarchical feature learning. CNN does not encode the position and orientation of objects. The convolution operations can be performed well in lower-end machines, such as smartphones, embedded systems, and IoT devices. CNN architecture is considered as the key for the development of almost all the state-of-the-art NLP models which are good in performing real-time applications that are available today.

2.7. Deep Learning based NLP Systems

Deep Learning was not a novel field. The first Deep Learning model was published by Alexey Ivakhnenko and Lapa in 1967. A 1971 paper described a deep network with eight layers trained by the group method of data handling, while the algorithm worked, training required by this algorithm took 3 days[17]. It is clear that Deep Learning requires high computation power. By definition, it was a family member of Machine learning that can automatically extract the features from the data while processing a large dataset. It was often interchanged with the word deep neural networks or artificial neural networks. Hence nothing was changed comparing the late 60s and 70s, except the extreme computation power and parallelization techniques of 21st-century computing machines.

In this section, we present some of the popular and most widely used deep learning models in recent years. The models described here are widely used in the design of NLP based Question-Answering systems. The one commonality in all the discussed models is: all of them are ‘transformers’ based deep neural network models.

2.8. Bidirectional Encoder Representations from Transformers (BERT)

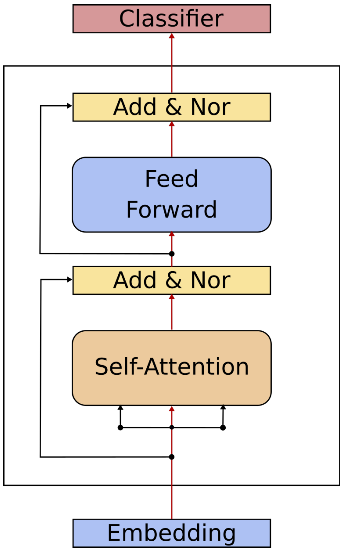


Figure 2: BERT Architecture with Self-attention.

BERT was published in 2018 by Devlin et al. from Google and performed very well. They[8] make use of transformers and stacks multiple transformer encoders on top of each. It used bidirectional learning as opposed to directional models[8]. BERT tries to understand the context of each word from left and right. BERT architecture is based on Multi-Head Attention. BERT uses two pre-training tasks: masked language modeling (MLM) and the next sentence prediction (NSP). These two tasks are performed on the pretraining data of the BookCorpus (800 million words) and English Wikipedia (2500 million words). BERT is distinguished from past DNN because it was already pretrained(they call it as fine-tuned). It can ready used in decent machines. Even BERT can be modified by adding one or few core layers in its end by the process called transfer learning. These salient features of the BERT made NLP enthusiasts to personalize and optimize new models.

The“Fig. 2” shows the architecture of BERT. As shown in this figure, the self-attention block is responsible for possible word prediction with the help of two methods as described in the BERT paper, which is the masked language modeling(MLM) and next sentence prediction(NSP). Other blocks are common functional blocks found almost in every transformer based architectures.

3. ChatBot System Design

Modules of the Project Chat Server Module Domain-specific QA Open Domain QA ChatBot Personality QA Chat Client Module HTML Part CSS Java Script

Hardware Specification System: Desktop/Laptop Intel Core: i7 2.4 GHz. Hard Disk: 500GB SSD. Monitor Display: 14' Mouse: Optical Mouse. Ram: 8Gb. Keyboard: 101 Keyboard.

Cloud Environment Processor: Intel(R) Xeon(R) CPU @ 2.20GHz Internal Memory Capability: 13Gb Disk Capability: 32Gb

Software Specification Front-end : Python, HTML, CSS, Javascript Tools/Libraries: Flask, ngrok, Wikipedia API, DuckDuckGo API, Summarizer Operating System: Linux based system IDE: Google's Colabatory with Jupyter environment

3.1. The Proposed System Architecture

The "Fig. 3" describes the abstract design of the proposed system. It can be divided into five blocks as follows:

- Flask Web Interface Interaction - It is a client-side module. This helps the user(client) to raise the question to the proposed chatbot system.

- Question - The question block contains two subblocks. The "API" subblock caries the question to the next block by processing some logical conditions with a question. Then, the "BERT" subblock was bidirectionally connected to two external blocks. The function of the "BERT" subblock will be explained in consecutive blocks.

- Information Retrieval - This block is responsible for retrieving the information. It retrieves the information with the help of three sub-blocks namely: 1)web retrieval, 2)Wikipedia articles retrieval, and 3)personalized data retrieval. The first two sub-blocks use the internet for information retrieval, so it is Open-domain in nature. However, the specialty of the third subblock is retrieving information from any customized data(which may include database/with or without internet).

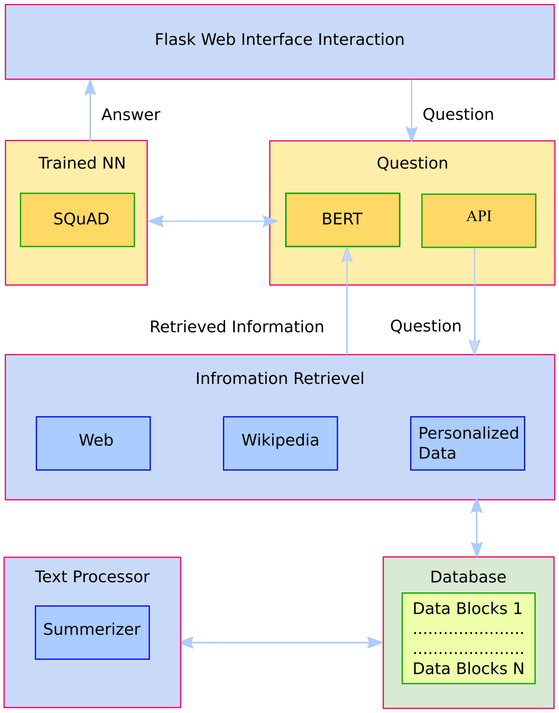


Figure 3: Proposed System Architecture.

Therefore, it can perform like a functional, personalized chatbot for specific domains and it is adaptive in nature.

- Database - The database mentioned here may be an offline or online-based QA database. In the case of Open-domain, Wikipedia and DuckDuckGo data were fetched by using two API's as mentioned in the previous block. In the case of Domain-specific, the offline database mode will be used to retrieve data from any personalized dataset.

- Text Processor - It is a simple summarizer. It is used to summarize large text information from database blocks into meaningful summary text.

- Trained NN - This block contains a fine-tuned language model(DNN), that is present inside subblock "BERT" of the "Question" block. To visualize the process, the "BERT" subblock was placed inside the "Question" block. And, this "BERT" subblock was responsible for retrieving information from the "Information Retrieval" block and fed it into the trained NN (actually BERT/SQuAD). Then, the trained NN produces the predicted answer to the (user)client by sending it to the “Flask Web Interface Interaction” block. From the above blocks, we can understand the life cycle of a QAS session.

3.2. The Training Process

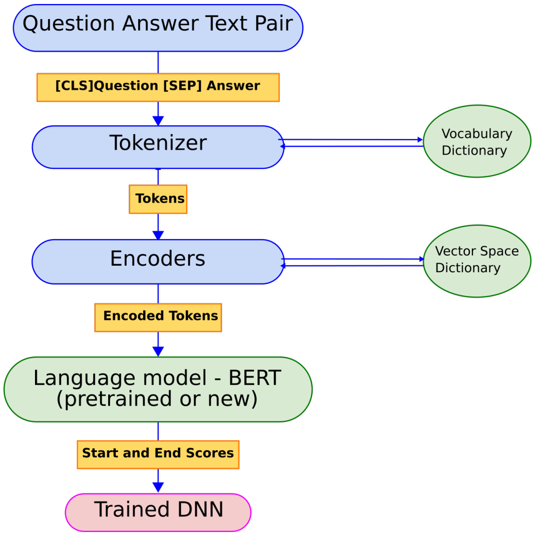


Figure 4: The Training Process.

The “Fig. 4” , represents the step by step process of training of an language model(DNN). In the first step, it was clear to understand that the question-answer text pair was fed into the language model. Then, the intermediate step next to the first step describes the feeding process that the process of using two functions: classifier and separator. Using these functions, the question text was followed by a classifier, and the answer text was separated from the question with the help of a separator, and the end of the answer text was also separated by a separator. Then the question-answer text pair fed into the third step. In this step, it describes the tokenization process, the process of converting a sequence of word data(input) into independent tokens, and this conversion process was carried out by a dictionary called a vocabulary dictionary. Then these tokens are fed to the next consecutive steps. The fifth step of the training process describes the encoders, the process of encoding(converting) tokens(tokenized words with vocabulary representation) into encoded tokens(machine understandable language), this encoding is done with the help of a dictionary called vector space dictionary. The vector space dictionary is nothing but numerical values, which hold the information of magnitude and direction, and it is also helpful to measure the distance between dependent and independent values. Generally, the tokenizer to encoder steps are jointly called as word embeddings. Then, these encoded tokens are taken into the next consecutive steps. This step was called as the Language model(DNN), a DNN specially innovated for Q&A tasks. So the training approach was different from other DNNs. Because most of these language models use attention mechanisms, it is an unsupervised process of extracting 'key - values' using attention over input sequences. The 'key - values' are the relationship between keys and values, and the key was developed from queries, every key has a specific value, that is vector. The vectors help to evaluate the distance from one key to another key using distance measurement algorithms(like euclidean distance). Generally, the language model performs a few complex mathematical functions, while pre-training a Q&A dataset.

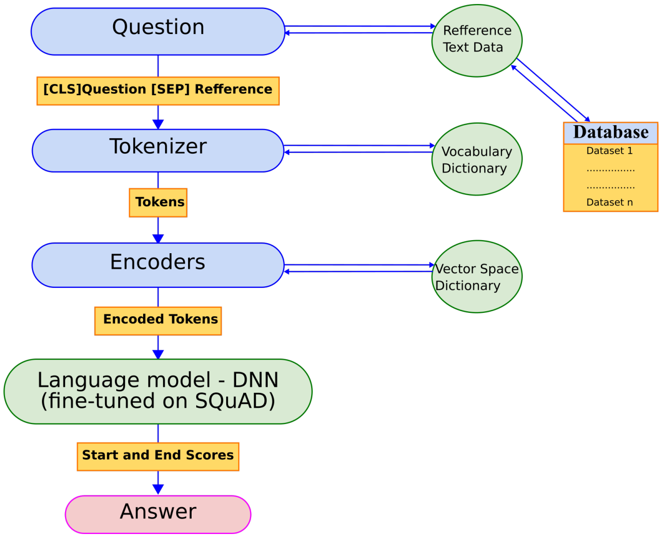


Figure 5: The Testing Process.

The “Fig. 5” , explains the step by step process of testing a language model(DNN). In the first step, it is clear to understand that the question was taken into a database, which holds the reference text data. Then, the question with reference text was fed into the intermediate next step. In this intermediate step, it describes the feeding process, that a process of using two functions: a classifier and separator. Using these functions, the question text was followed by a classifier, and the reference text was separated from the question with the help of a separator, and the end of the reference text was also separated by a separator. Then the question reference text pair feed into the next consecutive steps. The next consecutive steps are the same steps seen in “Fig. 4”. Only major difference is in the fourth step as seen in “Fig. 5”, which is a trained language model and it is used to provide the answer text for the respective question text from the particular reference text as this can be seen as the result in the final step.

3.4. The Working Logic of the Overall ChatBot

The “Fig. 6”, explains the complete working logic of Open domain mode of the chatbot and that is also divided into the client and server modules of the chatbot. As this system, “Fig. 6” was implemented in the cloud environment and its working logic diagram was different from the traditional implementation. So it is divided into four blocks namely,

- Block One: This block describes the important server module dependencies and client module scripts. There are some important server dependencies used for demonstrating this chatbot. The corresponding server dependencies are downloaded from PyPI website using pip commands. And then the client module scripts are initialized and then stored in the specific directory of the cloud environment.

- Block Two: This block describes the initializing process of the downloaded server dependencies and trained language model(DNN). The imported dependencies are initialized first and then fine-tuned language model with its tokenizer was also downloaded and initialized.

- Block Three: This block is the most important block of the chatbot. Because it describes the complete working logic of the Open domain mode. In the case of Domain specific mode, only the QAS mode will be changed, and there will be no changes in other blocks of the chatbot.

- Block Four: This block describes the answer provided from “Block Three” and displays the result to the client module of the system.

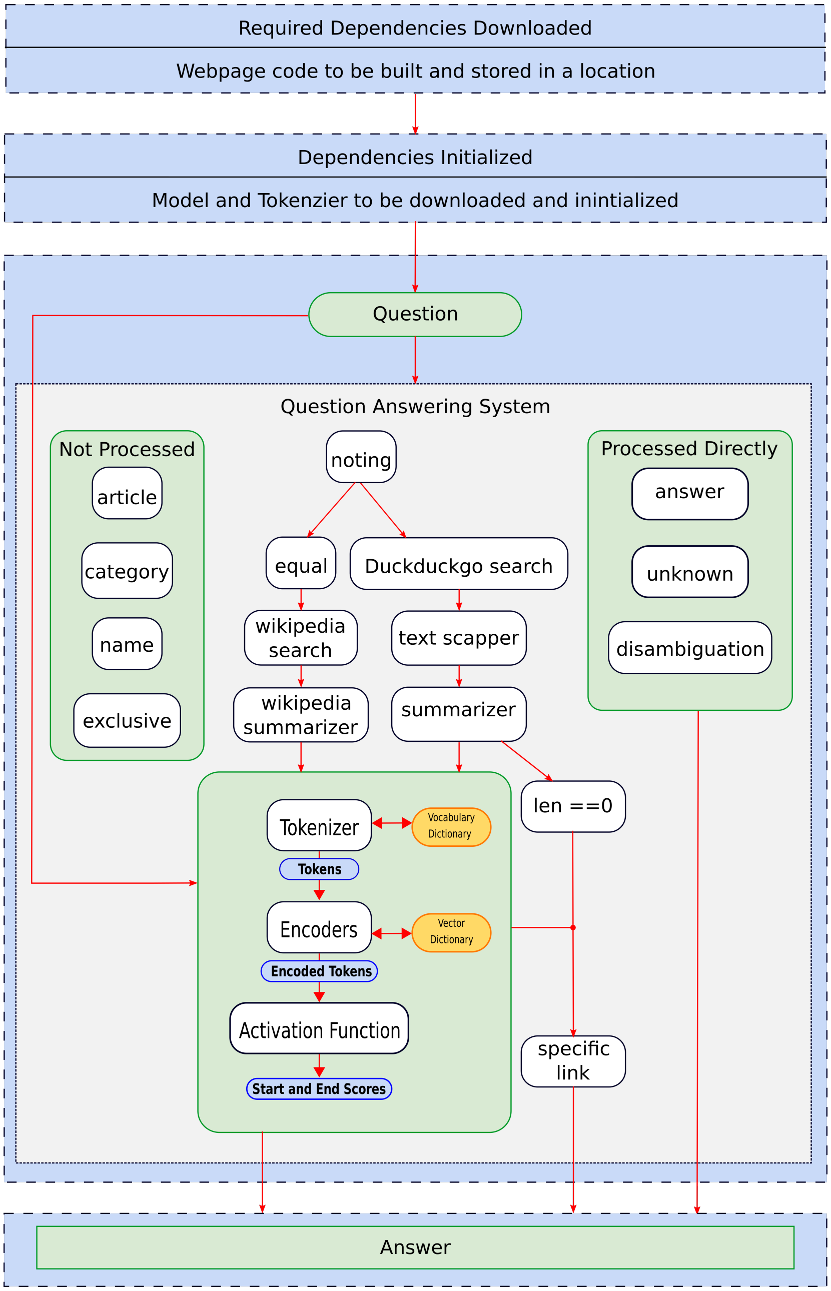


Figure 6: The Working Logic of Overall ChatBot.

3.5. Working Logic of Open domain Server QAS module

The “Fig. 6”, explains the working logic of the Open domain QA server module of the Chatbot. In this mode, the QAS is based on Wikipedia and DuckDuckGo search engines. In the first step, the question is passed into the QAS. And the QAS process by classifying it into three phases:

- Not Processed: This subblock describes the “not processed elements” of the Open domain QAS module, which are the articles, category, name, and exclusive. If the question comes under this block it does not provide the answer.

- Nothing: In this sub-block, “Noting” decides whether the question is taken to DuckDuckGo or Wikipedia. It is a step-by–by-step process.In the case of Wikipedia, first of all, the question will be identified by the Wikipedia search API, if the corresponding information is present, it will be fed into Wikipedia API built-in summarizer, and then the summarized text feed into the trained DNN - language model(as discussed in the above testing process). and provides the answer as output. If Wikipedia can not process the data, DuckDuckGo API will perform a search to the corresponding question and provides multiple links as results. Then the top-ranked link is taken into the text scraper(Beautiful Soup) to retrieve the text data. Then the retrieved text data get summarized with the help of a summarizer. Finally, the checker step checks for two conditions 1. If the length of the summarized text is equal to zero then it provides a specific link answer as output. 2. If the length of the summarized text is equal to the whole number, then the summarized text fed into the trained DNN - language model(as discussed in the above testing process). and provides the answer as output. If the question comes under this block, it takes a few seconds to provide the answer.

- Processed directly: In this sub-block, the elements “answer”, “unknown”, and “disambiguation” are directly processed by the DuckDuckGo module. Then provide the answer as output. If the question comes under this block it takes a few milliseconds to provide the answer.

3.6. Working Logic of Domain-specific Server QAS module

The “Fig. 7”, explains the working logic of the Domain-specific QA server module of the Chatbot:

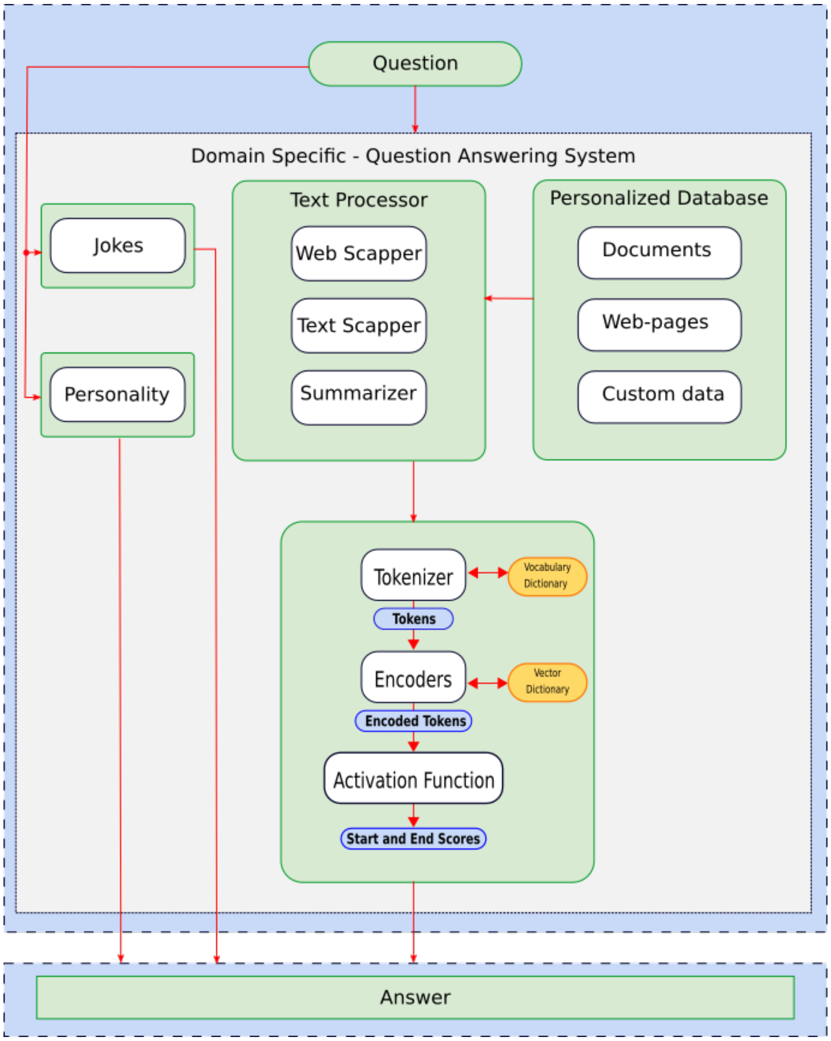


Figure 7: The Working Logic of Domain-specific QAS module

- Personality - It is a custom dataset prepared by chatbot developers. It is about the chatbot’s bio-data and its interests. This dataset makes the chatbot behave like a human. If the client’s question is “tell me about you? or who are you?", then it comes under this block and provides its personal information as the answer.

- Personalized database - It was a dataset prepared by the annotator/developers/information scrappers or prepared automatically from facts, documents, webpages, FAQ, books, etc. It can be customized and personalized according to the client's needs. This makes this dataset as "domain-specific in nature.

- Text Processor - It was a text processing unit that mainly focus on creating human-understandable text from scarped data(web, documents, or custom dataset). Then summarizing those texts whit the help of a summarizer. This summarizer is used to make large text sequences into a meaningful text summary.

- Language model - It was the identical block that was already described in "Fig. 5". It is used to process domain-specific questions with the Domain specific reference data. The question with reference data is processed into the fine-tuned language model and provides its predicted result as an answer. If the client's question is "where is TCE or when TCE was established?", then it comes under this block and provides its specific information as the answer.

- Jokes - It is a computer science based joke module called "PyJokes" implemented in Python on the server module, As like modern real-time chatbot systems, it can also tell a joke to the user and entertain him/her. If the client's question is "tell me a joke", then it comes under this block and provide a joke as the answer.

3.7. Experimental Setup

As discussed in the above sections this project requires a lot of computing resources. The Chatbot server-side module is made to run on the Google cloud environment. Even the client-side module will be served from the server to the client machine and then made run at the client's browser.

4. Implementation Results and Discussion

About the Text Dataset sources The SQuAD, Wikipedia articles, Web Search based on DuckDuckGo, and personalized dataset are the text datasets used in this paper.

4.1. Stanford Question Answering Dataset (SQuAD)

SQuAD is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable[36]. SQuAD2.0 combines the 100,000 questions in SQuAD1.1[35] with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

It is a question and reference text dataset prepared by Stanford University, it was famous for NLP tasks like training the neural network in it and testing its prediction by comparing other neural network predictions. BERT and Longformers are fine-tuned with SQuAD. It performs out of the box for NLP tasks. It was a structured dataset, provided with question and answers under specific labels. Thus is dataset is based on large text carpus and Wikipedia English.

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph[36].

In Proposed system, the deep neural networks already trained with SQuAD are used as a base neural network for performing our QAS.

4.2. Wikipedia

Wikipedia is a digital encyclopedia with millions of information based on multi-domain. In the Proposed system, the Wikipedia API for python is used for information retrieval from Wikipedia articles.

4.3. DuckDuckGo

DuckDuckGo is an internet search engine that maintains defending searcher's privacy and avoiding the filter bubble of personalized search results. It distinguishes itself from other search engines by not profiling its users and by showing all users the same search results for a given search term. In the Proposed System, the DuckDuckGo API for python is used for information retrieval from DuckDuckGo searched web pages.

4.4. Personalized Dataset

This dataset is a custom dataset prepared by experts, it may be structured or unstructured text data. This dataset is prepared by the process called scraping. These personalized data are scrapped from FAQs, documents, Web-pages of an institute or organization with legal permission and confirmation by the specific institute or organization. In the Proposed System, the Scrapping tools for python are used for preparing Personalized Dataset.

5. Observations and Findings

As we surveyed [37], we realized the important improvements in NLP techniques of the different phases of NLP history. Based on that observation, concerning performance we can classify all algorithms of NLP into five classes as algorithms that are giving:

- Below Human level Performance
- Near Human level Performance
- Equal Human level Performance
- Above Human level Performance and
- Super/Hyper Human level of Performance

For better understanding, we present the Table1 from [37] . In this we present the performance of five popular deep neural network models in terms of EM score and F1 score. If we carefully study the results, we can understand the outstanding performance of BERT based NLP models that even outperform human performance,

Except for the results of DistilBERT, all others were from the SQUAD2.0 test dataset. These results were obtained from the SQUAD2.0 leaderboard. The result of DistilBERT is from [14] and this is the scores were originally obtained with SQUAD2.0 DEV dataset. The BERT-based models perform well on SQuAD2.0. So, BERT was taken as the language model for developing this chatbot.

Table 1: Performance of NLP Models with - SQUAD2.0

Method	**Date**	**Performance**	
		EM	**F1 Score**
BERT (Single model) by Google AI Language	Nov 09, 2018	80.005	83.061
XLNet (Single model) by Google Brain & CMU	Nov 15, 2019	87.926	90.689
RoBERTa (Single model) by Facebook AI	Jul 20, 2019	86.820	89.795
DistilBERT[14] (Single model) [a]	-	66.259	69.670
ALBERT (Single model) by Google Research & TTIC	Sep 16, 2019	88.107	90.902
Human Performance by Rajpurkar & Jia et al. Stanford University	2018	86.831	89.452

[a] This result is from[14].

5.1. BERT based NLP Models

In Table 1, we described various transformer-based models and we tried BERT fine-tuned on SQuAD as our first DNN model that to be implemented in our chatbot. But it was a complete disaster. Even though these transformer-based language models were good in the SQuAD leader board, the main problem was with document accessing and retrieving, which are only capable of processing the small documents or processing documents as chunks, and sometimes it may result in out-of-memory problems. So we understood that BERT based models that we had surveyed were not helpful for our problem. So we came with another language model called Longformer.

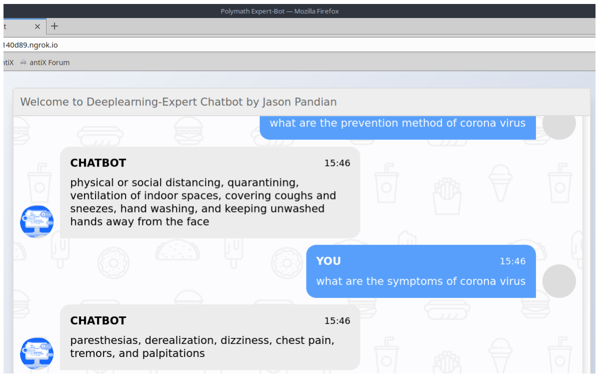
5.2. Longformer based chatbot

Longformers was a transformer-based model[2], that is capable of processing long text documents. As our project is meant for making an AI chatbot, it often deals with long documents. So to avoid chunking or to reduce out-of-memory problems. Our chatbot was implemented using the Longfomer language model. Hopefully, It was already fine-tuned with SQuAD2.0. So It can be used for building our AI chatbot. By using this Longfromer with Open-domain mode APIs and Domain-specific dataset, this project server module is built. And providing client interface, the server module is connected with client module with the help of flask framework with ngrok server.

5.3. Result Decision Maker

The Results are provided in the below sections, that based on chat conversations by the user to the chatbot. The chatbot is made up of two modes, Open-domain and Domain-specific. If the question is based on an Open-domain it would use the APIs: DuckDuckGo and Wikipedia search engines. The AI chatbot has some logical reason to understand the query(question), by understating the nature of the query, it will decide to use Wikipedia or DuckDuckGo search engine. Then, the query was parsed into the following APIs and fed into the language model. Finally, the answer is displayed to the client. In the case of Domain-specific AI chatbot mode can be changed for a particular dataset and process the same procedure and displays the answer to the client.

5.4. Results from PC



Welcome to Deeplearning-Expert Chatbot by Jason Pandian

what are the prevention method of corona virus

CHATBOT 15:46
physical or social distancing, quarantining, ventilation of indoor spaces, covering coughs and sneezes, hand washing, and keeping unwashed hands away from the face

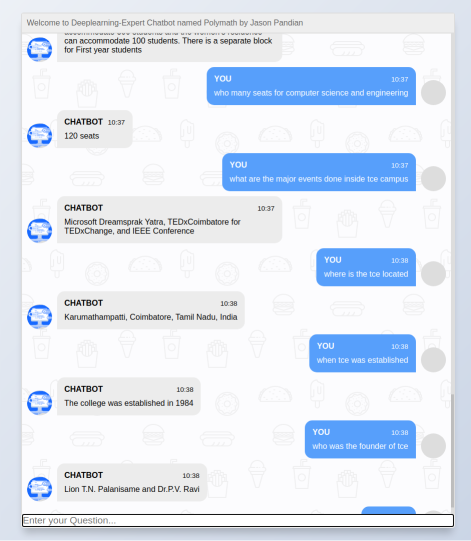
YOU 15:46
what are the symptoms of corona virus

CHATBOT 15:46
paresthesias, derealization, dizziness, chest pain, tremors, and palpitations

Figure 8: Open Domain mode of the Chatbot

In Fig. 8 describes the screenshot of the web application running on the client screen, In that application, the client asks two questions to the chabot, which was set on Open-domain mode - web. Both questions were based on the Coronavirus. It was clear that the chatbot was able to fetch appropriate information from the web by using DuckDuckGo and Wikipedia APIs and provide appropriate results with the help of the fine-tuned language model.

5.5. *Result of Domain-specific mode*



Welcome to Deeplearning-Expert Chatbot named Polymath by Jason Pandian

can accommodate 100 students. There is a separate block for First year students

YOU 10:37
who many seats for computer science and engineering

CHATBOT 10:37
120 seats

YOU 10:37
what are the major events done inside tce campus

CHATBOT 10:37
Microsoft Dreamsprak Yatra, TEDxCoimbatore for TEDxChange, and IEEE Conference

YOU 10:38
where is the tce located

CHATBOT 10:38
Karumathampatti, Coimbatore, Tamil Nadu, India

YOU 10:38
when tce was established

CHATBOT 10:38
The college was established in 1984

YOU 10:38
who was the founder of tce

CHATBOT 10:38
Lion T.N. Palanisame and Dr.P.V. Ravi

Figure 9: Domain-specific mode based on TCE

In Fig. 9 shows the screenshot of the web application running on the client's mobile, In that application, the client asks the questions about 'Tamilnadu college of engineering(TCE)' to the chabot, which was set on was set to Domain-specific mode. In Domain-specific mode, the chatbot replies accurately to all possible questions asked by the client. So we can understand that this system can be implemented in any domain easily and perform its best without any extra training or fine-tuning the new dataset.

6. Conclusion

We have successfully implemented an AI Chatbot using Deep Learning based NLP techniques. The implemented system performed very well. From the results of the previous section, it is clear that the Chatbot was able to respond with appropriate information.

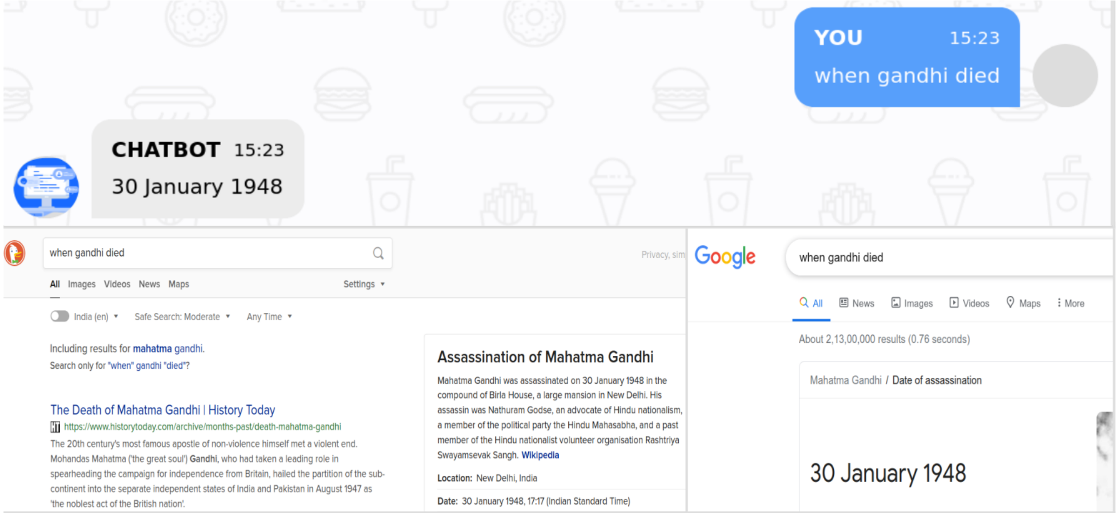
As we observed in our survey, DL based NLP models performed poorly even until 2017 and their results are very lower than human performance. After that, with the availability of heavily powered machines, DL based NLP models started to perform good but their performance was still a little bit low comparing with actual human performance. During 2018, some State of the

Art models evolved and played a huge role in NLP tasks and achieved equal to human performance. After that, Multi powered(layered or ensembled) State of the Art models evolved and started to achieve above human level performance in various NLP tasks. As we mentioned in the previous section, the NLP models based on Deep Learning are quite interesting and were able to achieve above human level of performance.

Today, DNN based NLP systems capable to translate any language text or speech to any other known language. In the future, NLP may become super-intelligent and there may be a possibility of fine-tuned NLP system that may decipher the ancient, unknown, language scriptures and hence replace the traditional language expert. Future NLP model may provide human-like or better human-like Bots/Chatbots that can be your future caretaker, companion, or next-generation virtual teacher. The future NLP model may show a possibility to communicate with aliens(if there will be something) that often fantasize in best-selling science-fiction novels and fiction emulated movies. By surviving past and present of NLP and DNN, this survey ends with an endnote, that one day humans may achieve DNN/NLP/AI systems that can be super-intelligent and can be optimized to solve present unsolved problems of Quantum Physics and other Mathematical secrets of nature.

Appendix - More Results

6.1. Comparison of results with search engines



YOU 15:23

when gandhi died

CHATBOT 15:23

30 January 1948

when gandhi died

Including results for **mahatma** gandhi.

Search only for "when" gandhi "died"?

The Death of Mahatma Gandhi | History Today

https://www.historytoday.com/archive/months-past/death-mahatma-gandhi

The 20th century's most famous apostle of non-violence himself met a violent end. Mohandas Mahatma ('the great soul') **Gandhi**, who had taken a leading role in spearheading the campaign for independence from Britain, hailed the partition of the sub-continent into the separate independent states of India and Pakistan in August 1947 as 'the noblest act of the British nation'.

Assassination of Mahatma Gandhi

Mahatma Gandhi was assassinated on 30 January 1948 in the compound of Birla House, a large mansion in New Delhi. His assassin was Nathuram Godse, an advocate of Hindu nationalism, a member of the political party the Hindu Mahasabha, and a past member of the Hindu nationalist volunteer organisation Rashtriya Swayamsevak Sangh. **Wikipedia**

Location: New Delhi, India

Date: 30 January 1948, 17:17 (Indian Standard Time)

when gandhi died

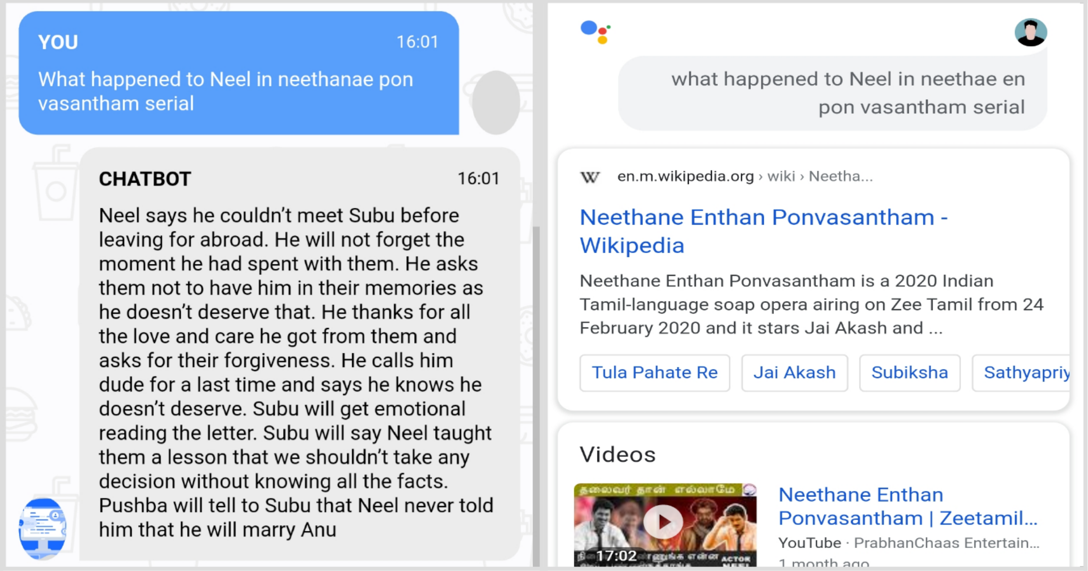
About 2,13,00,000 results (0.76 seconds)

Mahatma Gandhi / Date of assassination

30 January 1948

Figure 10: Comparision of Google, DuckDuckGo and Chatbot

In Fig. 10 shows the screenshots of the web applications running on the client computer, In this, the client asks a question about Mahatma Gandhi to the Google search engine, DuckDuckGo search engine, and our chabot. The DuckDuckGo search engine only provides a summary of Gandhi. However, Google provided a specific answer. Like Googe, our chatbot also capable of providing a specific answer, even though internally, it is using DuckDuckGo and Wikipedia.



YOU 16:01

What happened to Neel in neethanae pon vasantham serial

CHATBOT 16:01

Neel says he couldn't meet Subu before leaving for abroad. He will not forget the moment he had spent with them. He asks them not to have him in their memories as he doesn't deserve that. He thanks for all the love and care he got from them and asks for their forgiveness. He calls him dude for a last time and says he knows he doesn't deserve. Subu will get emotional reading the letter. Subu will say Neel taught them a lesson that we shouldn't take any decision without knowing all the facts. Pushba will tell to Subu that Neel never told him that he will marry Anu

what happened to Neel in neethae en pon vasantham serial

en.m.wikipedia.org › wiki › Neetha...

Neethane Enthan Ponvasantham - Wikipedia

Neethane Enthan Ponvasantham is a 2020 Indian Tamil-language soap opera airing on Zee Tamil from 24 February 2020 and it stars Jai Akash and ...

Neethane Enthan Ponvasantham | Zeetamil...

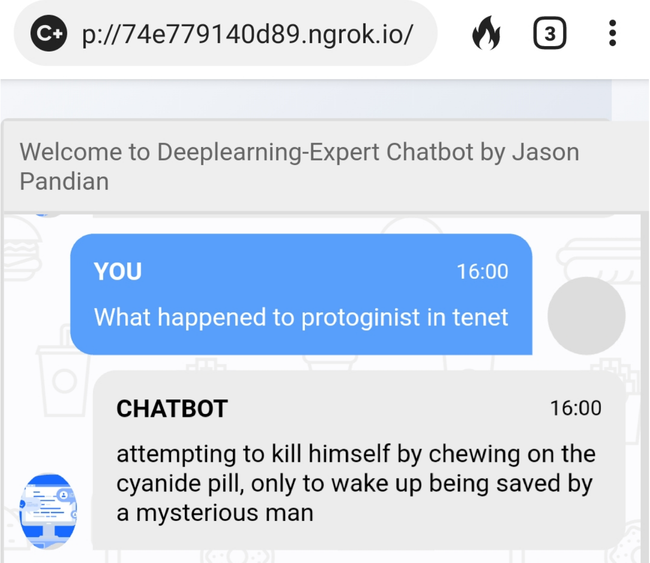
YouTube · PrabhanChaas Entertain...

1 month ago

Figure 11: Outperforms Google Assistant

The Fig. 11 shows the screenshot of the web application running on a client's mobile, In this, the client asks a question about a 'Tamil Television Serial' to the Google Assistant and our chabot. In this case, our Chatbot outperforms Google Assistant by providing an accurate answer.

6.3. Indirect Question answering



Welcome to Deeplearning-Expert Chatbot by Jason Pandian

YOU 16:00

What happened to protoginist in tenet

CHATBOT 16:00

attempting to kill himself by chewing on the cyanide pill, only to wake up being saved by a mysterious man

Figure 12: Open Domain Chatbot Result about Indirect Question answering.

The Fig. 12 shows the screenshot of the chat interface running on a client's mobile, In this, the client asks an indirect question to the chatbot about a movie 'TENET'. The chatbot tries to understand the question and gives very accurate results.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova , ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding ”, arXiv:1810.04805v2 [cs.CL] 24 May 2019.

[2] Iz Beltagy, Matthew E. Peters and Arman Cohan , ”Longformer: The Long-Document Transformer”, Seattle, WA, USA, arXiv:2004.05150v2 [cs.CL] 2 Dec 2020.

[3] Amit Mishra, Sanjay Kumar Jain, ”A survey on question answering systems with classification”, Journal of King Saud University - Computer and Information Sciences (2016) 28, 345-361.

[4] Mohammad Nuruzzaman and Omar Khadeer Hussain , ”A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks”, IEEE 15th International Conference on e-Business Engineering (ICEBE).

[5] Finlay, Janet, and Alan Dix, ”An introduction to Artificial Intelligence”, UCL Press, London, 1996

[6] Karen Sparck Jones, ”Natural Language Processing: A Historical Review”, Computational Linguistics, vol. 9-10; Pisa, Dordrecht, 1994.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, ”Attention Is All You Need”, Google Brain, arXiv:1706.03762v5 [cs.CL] 6 Dec 2017.

[8] Jacob Devlin, Ming-Wei, Chang Kenton Lee, Kristina Toutanova, ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, Google AI Language, arXiv:1810.04805v2 [cs.CL] 24 May 2019.

[9] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le, ”XLNet: Generalized Autoregressive Pretraining for Language Understanding” , Carnegie Mellon University, 2Google AI Brain Team, rXiv:1906.08237v2 [cs.CL] 2 Jan 2020.

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov,

Paul G., ”RoBERTa: A Robustly Optimized BERT Pretraining Approach”, arXiv:1907.11692v1 [cs.CL] 26 Jul 2019.

[11] Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF, ”DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter”, arXiv:1910.01108v4 [cs.CL] 1 Mar 2020.

[12] Zhenzhong Lan1 Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut, ”ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS”, A conference paper at ICLR 2020, arXiv:1909.11942v6 [cs.CL] 9 Feb 2020.

[13] Keith D. Foote, ”A Brief History of Deep Learning”, 2017. https://www.dataversity.net/brief-history-deep-learning/

[14] Melanie Beck, ”Evaluating QA: Metrics, Predictions, and the Null Response”, Cloudera Fast Forward Labs, Minneapolis, MN, Jul, 2020.

[15] Rosenblatt F, ”The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain”, Psychological Review Vol. 65, No. 6,1958.

[16] Linnainmaa, Seppo, ”The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors”, Univ. Helsinki, 1970

[17] Ivakhnenko, A. G., Lapa, V. G., ”Cybernetics and Forecasting Techniques”, American Elsevier Publishing Co. ISBN 978-0-444-00020-0, 1967.

[18] Williams, Ronald J.; Hinton, Geoffrey E., Rumelhart, David E. , ”Learning representations by back-propagating errors”. Nature. 323 (6088): 533-536, ISSN 1476-4687. October 1986.

[19] David H. Hubel and Torsten N. Wiesel, ”Brain and visual perception: the story of a 25-year collaboration”, Oxford University Press US. p. 106. ISBN 978-0-19-517618-6, 2005.

[20] Fukushima, K., ”Neocognitron”, doi:10.4249/scholarpedia.1717, 2007.

[21] Fukushima, Kunihiko, ”Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position”, Biological Cybernetics, 1980.

[22] Collobert, Weston, ”Machine Learning, Proceedings of the Twenty-Fifth International Conference”, Univ.Helsinki, Finland, 2008.

[23] Mikolov Tomas,(etal.), ”Efficient Estimation of Word Representations in Vector Space”. ArXiv:1301.3781, 2013.

[24] Graves, A., Schmidhuber J., ”Framewise phoneme classification with bidirectional LSTM and other neural network architectures”, 2005.

[25] Sutskever, Ilya, (etel.), ”Sequence to sequence learning with neural networks”, ArXiv:1409.3215, 2014.

[26] Dzmitry Bahdanau, Cho Kyunghyun, Yoshua Bengio, ”Neural Machine Translation by Jointly Learning to Align and Translate”, ArXiv:1409.0473, 2014.

[27] Alex Graves, Greg Wayne, Ivo Danihelka, ”Neural Turing Machines”, Google DeepMind, arXiv:1410.5401v2 [cs.NE], 2014.

[28] Jason Weston, Sumit Chopra, Antoine Bordes, ”Memory Networks”, Facebook AI Research, arXiv:1410.3916v11 [cs.AI], 2015.

[29] Sainbayar Sukhbaatar, (etel.), ”End-To-End Memory Networks”, New York University, arXiv:1503.08895v5 [cs.NE], 2015.

[30] Ankit Kumar, (etel.), ”Ask Me Anything: Dynamic Memory Networks for Natural Language Processing”, CA USA, arXiv:1506.07285v5 [cs.CL], 2016.

[31] Graves, Alex, (etel.), ”Hybrid computing using a neural network”, 2016.

[32] Mikael Henaff, (etel.), ”TRACKING THE WORLD STATE WITH RECURRENT ENTITY NETWORKS”, A Conference paper at ICLR 2017, arXiv:1612.03969v3 [cs.CL], 2017.

[33] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang, ”SQuAD: 100,000+ Questions for Machine Comprehension of Text”, Stanford University, 2016.

[34] Do-Hyoung Park, Vihan Lakshman, ”Question Answering on the SQuAD Dataset”, Stanford University, 2018.

[35] SquAD(closed), https://rajpurkar.github.io/SQuAD-explorer/explore/1.1

[36] SquAD2.0, https://rajpurkar.github.io/SQuAD-explorer/

[37] Jason Pandian, ”Survey on Deep Learning Based Natural LanguageProcessing Systems”, An unpublished work, 2021