QUESTION ANSWERING AND SENTIMENT ANALYSIS FOR BOTS USING LANGUAGE MODELS

Sanjana Rajesh Revankar[1], Chahat Tandon[2], Arpita T M[3], Pratiksha Bongale[4], Dr. Nirmala C R[5], Dr. Hemant Palivela[6]
[1,2,3,4,5,6]Computer Science and Engineering
Bapuji Institute of Engineering and Technology, Davangere, India
[1]sanjanarrdvg@gmail.com, [2]chahat.7874@gmail.com, [3]arpita.telkar@gmail.com, [4]pratikshajb@gmail.com, nirmala.cr@gmail.com, [6]hemant.datascience@gmail.com

Abstract—There are no simple strategies for speedy planning during tests that offer individuals accurate responses inside a solitary snap. People fall out of good preparation mostly due to lack of resources such as books (not easily affordable by the underprivileged), even googling doesn't help much as you don't always get exact answers but a list of different links that MIGHT contain the answer. Both the above methods require time and a lot of patience. These learning strategies do not help aspirants much while preparing for such high-level competency exams. This could cause end-moment tension as well as uncertainty right before examination periods! Existing chatbots are for the most part rule-based which limits them to think past what they have been prepared with and the generative chatbots constructed utilizing neural organizations are not proficiently accomplished utilizing Language Models as their essential undertaking is to comprehend and decipher human dialects which will make our chatbot powerful. Since Chatbot will be based on Machine Reading Comprehension, it can easily evaluate whether the answer is fit or appropriate to the given question or not which aims to answer the question by extracting answers from the given context and it can be useful to measure the goodness of Q n A pair. This Chatbot can be utilized within e-learning platforms as a virtual teacher, assistant, stock advisor, and in many more scenarios.

Keywords—Bots, MRC, Vader Sentiment, Question Answering CdQA.

I. INTRODUCTION

Deep learning is important for a more extensive group of AI strategies dependent on fake neural organizations with portrayal learning. Learning can be directed, semi-regulated, or even solo. Data Science (DS), Machine Learning (ML), Deep Learning (DL), and Artificial Intelligence (AI) is a gigantic field for certain unpretentious and not so unobtrusive contrasts between the subjects. Question answering (QA) is nothing but a software engineering discipline that entails data recovery and natural language processing (NLP). Its main focus is building frameworks that consequently answer questions presented by people in a characteristic language. Machine Reading Comprehension is one of the critical issues in Natural Language Understanding, where the errand is to peruse and grasp a given book section, and afterward answer the questions dependent on it. Feeling investigation considers the abstract data in an articulation, that is, the assessments, examinations, feelings, or perspectives towards a point, individual, or element. Sentiment analysis of any piece of text helps in determining the actual nature of the sentences. The sentiments can be categorized into 3 main parts: Positive, Negative, and Neutral. A language model is a capacity that puts a likelihood measure over strings drawn from some jargon.

Web search tools like Google offer a rundown of connections where answers may be available for any cutthroat test questions available in various languages as well as formats. Even though they have paragraphs and paragraphs of information it can sometimes be difficult to get an immediate and exact response for an inquiry. The current Chatbots utilized for the MRC component are generally rule-based, i.e., their fundamental working guideline is fabricated utilizing Machine Learning calculations. The proposed chatbot produces answers for which there is no straightforward reply present in the dataset on their own utilizing the prior information inside them and with the comprehension of the setting of the subject from past discussions with clients. Such Chatbots are assembled utilizing neural organizations and perform better when contrasted with customary Machine Learning models.

There is an expanded exploration of interest in this kind of framework to deliver more adaptable frameworks that acquire and approve the appropriate responses from numerous and potentially outside sources in situations where a solitary information base isn't sufficient to respond to the inquiry. Specialists presented another kind of data recovery framework where the data hotspot for discovering the appropriate response is either completely made out of the picture data set or is in a content configuration. They recognized this as an exploration pattern propelled by the need to question frameworks that go past the customary limits of the standard-based QA frameworks. The lack of precision and the speed of the proposed framework is the further extent of the examination according to their decision.

Language models can be trained on text, books, even patterns and taught to understand and interpret the context of any textbook required by any organization in the whole world. This can then be used as an AI bot by organizations to help educate their students in a fun manner as well as provide them with precise and apt answers whenever required at their fingertips. This paper aims to make:

- Use of the Language models which are known to understand and interpret human languages which will give better results when compared to traditional ML approaches.
- Use Sentiment Analysis features to make the conversation look more natural and interactive, giving it a human touch.

This Chatbot can be used inside e-learning stages as a virtual educator, partner, yoga teacher, stock guide, and in a lot more situations.

Some of the main objectives of this study are as follows:

1. To develop a framework to train and build a QA Bot using Language Models.
2. To include Sentiment analysis to make the QA Bot livelier and human-like.
3. To differentiate between answerable questions and unanswerable questions.
4. To achieve the best possible interpretation of the context of the sentences and to identify hidden sentiments within them.

II. LITERATURE SURVEY

While Search Engines predominately contribute to answering questions of users from across the globe, they still lag in terms of providing an accurate answer with just one click. Search Engines (SE) like Google, Microsoft, Mozilla Firefox provide a list of third-party websites that are potential sources of answers to the user queries. For users to land the exact answer they might need to surf the Internet for a couple of minutes or even longer sometimes. There has been an increasing demand for Intelligent Question Answering Systems that can provide users with their answers in a single click and need no wastage of time browsing through numerous random websites. This paper focuses on building one for anyone wanting to prepare for any last-minute exams or even as a smart AI assistant for organizations in an extremely hassle-free manner. There have been numerous researches in the Question Answering domain, among which we have surveyed the following works. [1] proposed an integrated framework for Chinese intelligent question-answering in restricted domains. The proposed model fused convolutional neural network and bidirectional long short-term memory network which performs efficient semantic analysis on the question pairs to extract more effective features of the text. Meanwhile, the co-attention mechanism and attention mechanism were combined to obtain the semantic interaction and feature representation of the question pair for providing complete information for subsequent calculations. They introduced the method of question pair matching to implement the Chinese intelligent question-answering in a restricted domain. Experiments were tested and evaluated on the open-source CCKS2018 dataset and a private self-built inverted pendulum control question answering (IPC-QA) dataset for automation control virtual learning environment. Experimental results confirm that the proposed models are efficient and achieve high precision of 0.86042 and 0.8031 on CCKS2018 and IPC-QA respectively.

[2] presents a comprehensive survey on different aspects of machine reading comprehension systems, including their approaches, structures, input/outputs, and research novelties. The paper highlights the recent trends in this field based on 124 reviewed papers from 2016 to 2018. The authors' investigations demonstrate that the focus of research has changed in recent years from answer extraction to answer generation, from single to multi-document reading comprehension, and from learning from scratch to using pre-trained embeddings. They also discuss the popular datasets and the evaluation metrics in this field. The paper ends with investigating the most cited papers and their contributions. In [3], the authors propose new pre-trained contextualized representations of words and entities based on the bidirectional transformer (Vaswani et al., 2017). The proposed model treats words and entities in a given text as independent tokens and outputs contextualized representations of them. The model is trained using a new pretraining task based on the masked language model of BERT (Devlin et al., 2019). The task involves predicting randomly masked words and entities in a large entity-annotated corpus retrieved from Wikipedia. They also propose an entity-aware self-attention mechanism that is an extension of the self-attention mechanism of the transformer and considers the types of tokens (words or entities) when computing attention scores. The proposed model achieves impressive empirical performance on a wide range of entity-related tasks. In particular, it obtains state-of-the-art results on five well-known datasets: Open Entity (entity typing), TACRED (relation classification), CoNLL-2003 (named entity recognition), ReCoRD (cloze-style question answering), and SQuAD 1.1 (extractive question answering). The purpose [4] is to improve the accuracy of the sentiment classification by employing the concept of word embedding. This study uses Word2Vec to produce high-dimensional word vectors that learn contextual information of words. The resulting word vectors are used to train machine learning algorithms in the form of classifiers for sentiment classification. Their experiments on real-world datasets show that the use of word embedding improves the accuracy of sentiment classification.

In an e-commerce environment [5], user-oriented question-answering (QA) text pairs could carry rich sentiment information. This study proposes a novel task/method to address QA sentiment analysis. They propose a three-stage hierarchical matching network to explore deep sentiment information in a QA text pair. First, the segmentation of both the question-and-answer text into sentences and construct several [Q-sentence, A-sentence] units in each QA text pair. Then, by leveraging a QA bidirectional matching layer, the proposed approach can learn the matching vectors of each [Q-sentence, A-sentence] unit. Finally, the characterization of the importance of the generated matching vectors via a self-matching attention layer. In [6], the researchers introduce the Reinforced Mnemonic Reader for machine-reading comprehension tasks, which enhances previous attentive readers in two aspects. First, a re-attention mechanism is proposed to refine current attentions by directly accessing past attentions that are temporally memorized in a multi-round alignment architecture, to avoid the problems of attention redundancy and attention deficiency. Second, a new optimization approach, called dynamic-critical reinforcement learning, is introduced to extend the standard supervised method. It is always encouraged to predict a more acceptable answer to address the convergence suppression problem occurring in traditional reinforcement learning algorithms. Extensive experiments on the Stanford Question Answering Dataset (SQuAD) show that our model achieves state-of-the-art results. Meanwhile, their model outperforms previous systems by over 6% in terms of both Exact Match and F1 metrics on two adversarial SQuAD datasets. The authors in [7], propose a simple yet robust stochastic answer network (SAN) that simulates multi-step reasoning in machine reading comprehension. Compared to previous work such as ReasoNet which used reinforcement learning to determine the number of steps, the unique feature is the use of a kind of stochastic prediction dropout on the answer module (final layer) of the neural network during the training. They show that this simple trick improves robustness and achieves results competitive to the state-of-the-art on the Stanford Question Answering Dataset (SQuAD), the Adversarial SQuAD, and the Microsoft

Machine Reading Comprehension Dataset (MS MARCO).

[8] introduces a new neural structure called FusionNet, which extends existing attention approaches from three perspectives. First, it puts forward a novel concept of "history of the word" to characterize attention information from the lowest word-level embedding up to the highest semantic-level representation. Second, it identifies an attention scoring function that better utilizes the "history of word" concept. Third, it proposes a fully-aware multi-level attention mechanism to capture the complete information in one text (such as a question) and exploit it in its counterpart (such as context or passage) layer by layer. They applied FusionNet to the Stanford Question Answering Dataset (SQuAD) and it achieved the first position for both single and ensemble models on the official SQuADleaderboard at the time of writing (Oct. 4th, 2017). Meanwhile, we verify the generalization of FusionNet with two adversarial SQuAD datasets and it sets up the new state-of-the-art on both datasets: on AddSent, FusionNet increases the best F1 metric from 46.6% to 51.4%; on AddOneSent, FusionNet boosts the best F1 metric from 56.0% to 60.7%.

[9] involves answering sequences of simple but interrelated questions. The paper introduces a dataset of 6,066 question sequences that inquire about semi-structured tables from Wikipedia, with 17,553 question-answer pairs in total. To solve this sequential question answering task, they propose a novel dynamic neural semantic parsing framework trained using a weakly supervised reward-guided search. The model effectively leverages the sequential context to outperform state-of-the-art QA systems that are designed to answer highly complex questions.

[10] proposes the task of free-form and open-ended Visual Question Answering (VQA). Given an image and a natural language question about the image, the task is to provide an accurate natural language answer. Mirroring real-world scenarios, such as helping the visually impaired, both the questions and answers are open-ended. Visual questions selectively target different areas of an image, including background details and underlying context. As a result, a system that succeeds at VQA typically needs a more detailed understanding of the image and complex reasoning than a system producing generic image captions. Moreover, VQA is amenable to automatic evaluation, since many open-ended answers contain only a few words or a closed set of answers that can be provided in a multiple-choice format. The authors provide a dataset containing ∼0.25M images, ∼0.76M questions, and ∼10M answers (www.visualqa.org), and discuss the information it provides. Numerous baselines and methods for VQA are provided and compared with human performance. Their VQA demo is available on CloudCV (http://cloudcv.org/vqa).

Memory Networks are a combination of an inference component and a long-term memory component, they learn how to use these jointly. The long-term memory can be read and written to, with the goal of using it for prediction. The authors investigated these models in the context of question answering (QA) where the long-term memory effectively acts as a (dynamic) knowledge base, and the output is a textual response [11]. In [12], Training large-scale question answering systems is complicated because training sources usually cover a small portion of the range of possible questions. This paper studies the impact of multitasking and transfer learning for simple question answering; a setting for which the reasoning required to answer is quite easy, as long as one can retrieve the correct evidence given a question, which can be difficult in large-scale conditions. They use a new dataset of 100k questions that we use in conjunction with existing benchmarks. The experiments were carried out using Memory Networks as this perspective allows us to eventually scale up to more complex reasoning. [13] presents a system that learns to answer questions on a broad range of topics from a knowledge base using few hand-crafted features. The model learns low-dimensional embeddings of words and knowledge base constituents; these representations are used to score natural language questions against candidate answers. Training our system using pairs of questions and structured representations of their answers, and pairs of question paraphrases. [14] includes a function that maps open-domain questions to queries over a database of web extractions. Given a large, community-authored, question-paraphrase corpus, they demonstrate that it is possible to learn a semantic lexicon and linear ranking function without manually annotating questions. This approach automatically finds words/paraphrases similar to each other from questions and returns the answer.

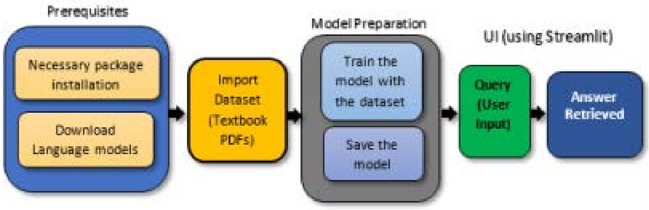
In [15], the authors put forward a system that showcases a novel approach for optimizing the analysis of sentiment in textual data using divergent combinations of the miscellaneous hyperparameters obtainable with various Language Models. They experimented on the performance of the Language Models, ALBERT, RoBERTa, and BERT by varying the method of optimization for a multiclass classification task by pre-processing the customer reviews which gave the finest model achieving futuristic results. They conducted multiple trials of various Language Models using customer reviews of the Google Play app by varying the optimizer used. It was found out that AdamW and Adabound gave the highest accuracies among the three experimented upon and carried out the experiment for 15 complete epochs. Dataset considered was the Google Play app reviews dataset that consists of nearly 16,000 rows with 11 attributes. All the experiments were carried out on a system with 8GB RAM, 1TB of Hard disk, an NVIDIA Tesla K40 GPU, and a Windows 8/10 Operating System. A variation was observed in the final accuracy of the pre-trained model by pre-processing the data. In [16], the researchers have demonstrated a Recognizing Textual Entailment wherein the task is to recognize whether a given hypothesis is true (Entailment), false (Contradiction), or unrelated (neutral) concerning the sentence called a premise. The task is performed by training MNLI corpus along with the manually collected dataset from Amazon Product Reviews each having hypothesis and premise pairs with corresponding labels. With this use case, they propose to bring sustainable development in the classification methods used by major E-commerce companies. The paper focused its experiments on solving NLI tasks for the business and retail industries. To address the lack of retail-based datasets, they handpicked some user/customer reviews for five different products, i.e., a yellow shirt, Crime and Punishment book, a Samsung M11 mobile phone, and Dell and HP laptops from the e-Commerce website, Amazon, and conducted the experiments on them.

III. DATASET

This Chatbot can be utilized within e-learning platforms as a virtual teacher, assistant, stock advisors, gyms, etc. It can be trained on the documents required as per the organization. For this study, the use case considered was for a school/college. The Bot was trained on 6 subject textbooks from the syllabus mentioned for Computer Science and Engineering students of the Visvesvaraya Technological University, India, namely; Machine Learning, User Design Interface, Cloud Computing, Internet of Things, and Operating Systems.

IV. METHEDOLOGY

A. *Workflow*



This paper describes a Question Answering model that is built by initially installing all the necessary packages including cdQA, VADER sentiment, pandas, joblib, etc. These packages are required to build the structure of the model and to include additional capabilities to the model. cdQA and VADER sentiment are the two important packages for the creation of the QnA Bot. cdQA is used for the determination of the right answer and VADER sentiment analyzer to analyze the sentiments hidden within the input question by the user. This model uses the pre-trained Language Models to ease out the process to an extent. We experimented using two different Language Models, BERT and DistilBERT. Since DistilBERT gives us a comparable result with lesser parameters than BERT, we chose DistilBERT for our model. Each of the language models was analyzed and compared before finding the most suitable one. BERT and DistilBERT were considered for this purpose. BERT, introduced by Google, is a bi-directional transformer that is pre-trained on the 16GB of unlabeled data to understand the context of the English language representation which nearly comes up to 3.3 billion words. BERT outperformed most of the NLP state-of-the-art and also includes NSP and MLM. DistilBERT uses a technique called distillation of its parent model BERT, which replaces a large neural network with a smaller one with a base size of 60 which is much lesser compared to RoBERTa and BERT whose base size is around 110. This model is trained on the same data (16GB unlabeled data) used for BERT.

After the analysis and comparison done between the above-mentioned language models, it was found that BERT has slightly better accuracy compared to DistilBERT as it has more parameters but also increases the inference speed minutely. But DistilBERT provides a good answer in a much reasonable time. Therefore, to maintain the inference speed as well as to acquire the better performance of the model, DistilBERT is found to be a reasonable choice required for the Bot creation.

After deciding which pre-trained Language Model to be used for creating the QA Bot, the distilbert-base-uncased pre-trained model is trained on a highly curated dataset consisting of various Portable Document Files (PDFs) of multiple subject knowledge. The entire dataset fed to the model is written in the English language. The content of these files can be anything on which the user wants to throw a question to the Bot. For the pre-processing part, all the PDFs were manually cleaned for being fed into the model. The PDFs finally collected contained only textual data to eliminate the issue of parsing tables and graphics. Pre-trained models are not trained to handle them so we just avoid using them and stick to text data for our downstream application fine-tuning of the pre-trained model used. As the CDQA package is designed to extract the answer only from paragraph context, removing all the images and structured data content like tables, forms, etc. is an essential step to obtain accurate results and for the better performance of the model. After the appropriate dataset creation, the model is trained on them and is saved using the joblib package. Joblib helps save Machine Learning models for later usage just by calling it whenever required.

As described in fig.1, once the model is trained and completely prepared, it is brought into a user-friendly environment for users to easily interact with it. Hence, creating a User Design Interface is an important step in Bot creation. Streamlit is one of the best ways supported by python using which a simple UI for the Machine Learning models can be created. The UI design contains the textbox for the user to input the question to the Bot. If the question is found positive by the VADER sentiment analyzer, then the most probable answer for the question is retrieved by the Bot with the help of cdQA. (The below section gives an idea of the cdQA design and working.) If the question is found to be negative or inappropriate to extract the answer by the Bot, it throws up an appropriate message to the user instead of the answer. This feature is added to the Bot to make it more specific to the needs of the domain in which it can potentially be used.

B. CdQa

Natural Language Processing is in its race in introducing the world to many competing models. With such progress, several improved systems and applications to NLP tasks are expected to come out. One of such systems is the CDQA. Discussing the QA systems, it is important to know about the two different kinds of systems: open-domain QA systems and closed-domain QA systems. Open-domain systems handle queries and requests about anything. On the other hand, closed-domain systems handle the questions which come under the specific field (for reference, medicine, and automotive maintenance), and can exploit site-specific knowledge and intellects by using a model that is tuned to a unique-domain database. cdQA package is built to enable anyone who wants to build a closed-domain QA system easily. The construction of cdQA is majorly based on two elements, one is the Reader and the other is the Retriever. Whenever a query or request question is passed to the model, the functionality of the Retriever is to select a list of documents in the pertained dataset that are expected to hold the answer. This is majorly based on the methodology on the retriever of drQA (open domain system developed by Facebook Researchers), which creates TF-IDF functions based on uni and bi-grams. This calculates the cosine closeness in the query-formed sentence and the document of the pertained dataset. Once the most probable document is selected, the system cuts short each document into paragraphs and sends them with the question to the Reader which is one of the components of the cdQA. The Reader is primarily a pre-trained Deep Learning model. It is the Pytorch version of the conventional Natural Language Processing model called BERT, which was rolled out by Google in the year 2018. The Reader then outputs the most appropriate answer out of all it can find in the list of paragraphs. There is a final layer in the system after the Reader that mainly weighs up the response of the model by making use of an internal score feature that retrieves an apparent answer based on the scores calculated by the internal score function.

The paper showcases the model which is dominantly built on 3 main elements of cdQA: PDF converter, model, and PDF pipeline. The PDF converter is used in converting all the PDF extended files to text for easy search of an answer. The proposed model in this paper mainly has 5 PDFs as a dataset to be converted to a document. The cdQA model is used to bring the answer out of a text document with the help of a QA pipeline. This is a step-by-step process in finding out the most probable answer by analyzing the entire set of documents, one by one, understanding the context of each paragraph by breaking out words into tokens using DistilBERT's pre-trained features. The language model is then trained on these documents and the answer is retrieved using the cdQA pipeline.

C. VADER Sentiment

Sentiment Analysis is a process of text analysis and context judgment that detects contrariness that is a *positive* and a *negative* school of thought from the text, irrespective of whether it is a whole document, paragraph, sentence, or clause. This paper utilizes a pre-trained sentiment analyzer called, VADER Sentiment abbreviated as Valence Aware Dictionary for Sentiment Reasoning, which is one of the sentiments analyzing models that are sensitive to both contrariety (positive/negative) and the ground intensity (strength) of emotion. The advantage of this model is it can be applied directly to unlabeled text data. Hence, the question asked by the user is directly analyzed before moving it to cdQA for answer identification. VADER sentimental inspection of the analysis depends on the dictionary that plots lexical features to emotion strengths known as sentiment scores. The sentiment score of a text is obtained by adding on all the intensity values of each word in the context. For example, when the words like *'like', 'amazing', 'happy', and 'fortunate'* all produce a positive sentiment in themselves. These scores are provided for each of the question inputs asked by the user. Another important criterion of VADER is it is intelligent enough to acknowledge the primary context in each of the words, such as *"did not enjoy"* and "*not happy*" as a negative statement. It also recognizes the emphasis of exclamation and punctuation, such as "*ENJOY*!!".

VADER sentiment has a SentimentIntensityAnalyzer() function that takes strings as the input and returns the scores of the dictionary in each of four categories: 1.negative 2.neutral 3.positive 4. Compound. Negative scores show the level of negativity in the asked question, the same goes for positive and neutral clauses. The compound score tells the average scores of all the other three scores, i.e., positive, neutral, and negative. This primarily decides whether the question is a negative question or a positive/neutral one to be answered. If the compound score is anything above zero or even 0, then it is termed to be an answerable query, else the model refrains from answering it. The compound scores can either be positive, 0, or negative. The results of the VADER sentiment analysis not only seem remarkable but also very encouraging. The system will proceed to search for the right answer only if it finds that the compound score is either 0 or anything greater than 0. In all other cases, it refrains from even searching for the answer and just displays a hard-coded text message to the user saying that it cannot answer that question.

For a false positive question asked by the user, for instance, *"How do I write an Exam?"*, the code is fine-tuned in a way that the model shows the same hard-coded text message to the user as in case of negative questions. The compound score is calculated by the VADER which is analyzed to find out whether the question is a false positive, false negative, true positive, or true positive. The prediction value is considered an important feature when deciding the sentiment of the question. If its value is less than 6, the query's status calls out to be a false positive, restricting the model from answering the question Hence, the model is restricted from answering these questions for efficient handling of a variety of input queries and better accuracy of the model.

V. RESULTS

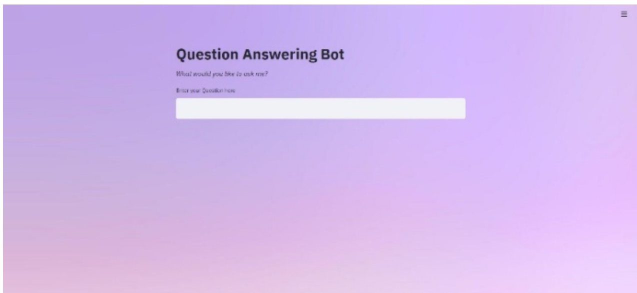


Fig. 2: User Interface of the QA Bot

This is a UI application called Question Answering Bot where the user needs to type the question, they need an answer for, and press enter. The sentiment of the question is checked and the answer is printed.

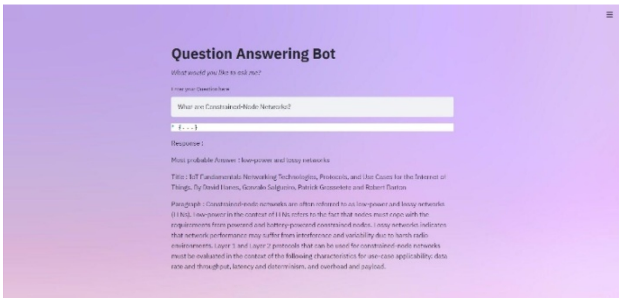


Fig 3. Output for a Positive Question by User

This figure illustrates the output that is the answer given to the user for the question he asked to the Question Answering bot. The format of the answer will be in form of the Most probable answer (most suitable answer), the Title of the textbook from which the bot has taken the answer from and then the paragraph which contains the answer.

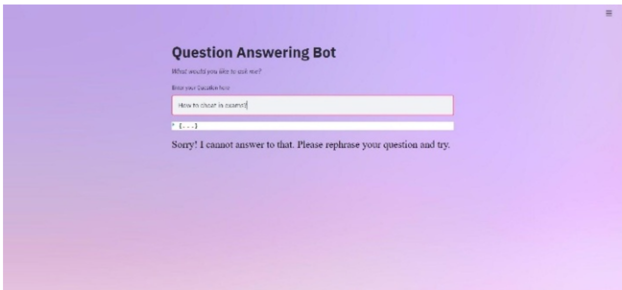


Fig 4. Output for a Negative Question by User

This figure illustrates a negative question by the user and the bot responds with "Sorry! I cannot answer to that. Please rephrase your question and try" because the sentiment analyzer identifies the question as a negative statement.

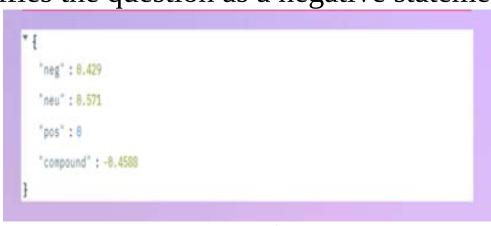


Fig 5. Sentiment values of a Negative Question

The figure defines the sentiment values of negative questions. Compound values are the overall sentiment of the question and its negative in the above figure because it was a negative question.

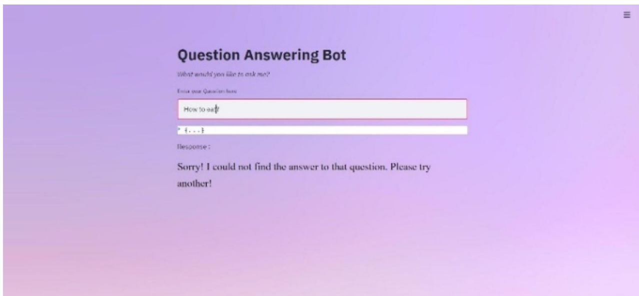


Fig 6. Output for Random Questions by user

This figure illustrates an example of a random question by a user and the bot responds with "Sorry! I could not find an answer to that question. Please try another!" because the bot is not trained to answer random questions which are out of its scope.

VI. CONCLUSION

This paper presents a very easy implementation of a Question Answering Bot that provides the users with answers from within the data that is given as an input to it. Most people tend to waste a lot of time and energy searching for answers to their queries on various Search Engines as Search Engines don't usually provide the exact answer to the questions instead provide a long list of third-party websites that it presumes might contain the required information. This paper intends to solve this problem and to help organizations and individuals save time by providing them with their answers in a single click. The experiments included two pre-trained language models for Question Answering, Bert-base-uncased and DistilBERT-base-uncased with a learning rate of 5e-05 and Adam epsilon optimizer to avoid divide by zero error with a value of 1e-08. The input to the system can be multiple Portable Document Files (PDFs) and the setup also includes a Vader Sentiment Analyzer for sentiment detection of the question. This setup is easily customizable for all organizations and domains. The only change would be to feed in domain-specific data and the Bot is ready to be used. The Bot is brought to the user through Streamlit.

REFERENCES

[1] Intelligent Question Answering in Restricted Domains Using Deep Learning and Question Pair Matching" by Lin-Qin Cai, Min Wei, Si-Tong Zhou, and Xun Yan - 2020.

[2] "A Survey on Machine Reading Comprehension Systems" by Razieh Baradaran, Razieh Ghiasi, and Hossein Amirkhani - 2020.

[3] "LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention" by Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto – 2020.

[4] "Sentiment Analysis with Word Embedding" by Oscar B. Deho, William A. Agangiba, Felix L. Aryeh, and Jeffery A. Ansah - 2019.

[5] "Sentiment Classification towards Question-Answering with Hierarchical Matching Network" by Chenlin Shen, Changlong Sun, Jingjing Wang, Yangyang Kang, Shoushan Li, Xiaozhong Liu, Luo Si, Min Zhang, Guodong Zhou - 2018.

[6] "Reinforced Mnemonic Reader for Machine Reading Comprehension" by Minghao Hu, Yuxing Peng, Zhen Huang, Xipeng Qiu, Furu Wei, Ming Zhou – 2018.

[7] "Stochastic Answer Networks for Machine Reading Comprehension" by Xiaodong Liu, Yelong Shen, Kevin Duh and Jianfeng Gao – 2018.

[8] "FUSIONNET: Fusing Via Fully-Aware Attention with Application To Machine Comprehension" by Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, Weizhu Chen – 2018.

[9] "Search-based Neural Structured Learning for Sequential Question Answering" by Mohit Iyyer, Wen-tau Yih, Ming-Wei Chang - 2017.

[10] "VQA: Visual Question Answering" by Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, Devi Parikh - 2016.

[11] "Memory Networks" by Jason Weston, Sumit Chopra, Antoine Bordes - 2015.

[12] "Large-scale Simple Question Answering with Memory Networks" by Antoine Bordes, Nicolas Usunier, Sumit Chopra, Jason Weston - 2015.

[13] "Question Answering with Subgraph Embeddings" by Antoine Bordes, Sumit Chopra, Jason Weston - 2014.

[14] "Paraphrase-Driven Learning for Open Question Answering" by Anthony Fader, Luke Zettlemoyer, Oren Etzioni - 2013.

[15] "Comparative Analysis of various Language Models on Sentiment Analysis for Retail" by Sanjana R R, Chahat Tandon, Pratiksha Bongale, Arpita T M, Hemant Palivela, Dr. Nirmala C R – 2021.

[16] "Use of Natural language Inference in optimizing reviews and providing insights to end consumers" by Chahat Tandon, Pratiksha Bongale, Sanjana R R, Arpita T M, Hemant Palivela, Dr. Nirmala C R – 2021.