# Chatbot for Information Retrieval from Website

Eldho Ittan George
*dept. Computer Science and Engineering*
*Muthoot Institute of Technology and Science*
Ernakulam, India
eldhoittangeroge@gmail.com

Dr. Cerene Mariam Abraham
*dept. Computer Science and Engineering*
*Muthoot Institute of Technology and Science*
Ernakulam, India
cerenemariam@mgits.ac.in

***Abstract*—A chatbot is a system that gives appropriate answers to questions expressed in natural languages such as English. A chatbot can help you in finding information efficiently. Generally speaking, we use search engines to search for relevant documents when we look for some information on the web. However, because they show you a document or a webpage, you must read the documents and decide whether they contain the information you need. A chatbot can provide a better solution in this scenario. Chatbots for websites can be built using different methods. But most of the methods require web admins to create either a set of predefined question answering pairs or rules. The responses of the chatbot will be limited to these predefined data. Here we propose a chatbot system for websites that requires no input from the web admins. The system will automatically collect the required data from the website a build a chatbot using that data. To create the model for the chatbot we use a retriever-reader architecture. The input to the model is a set of documents where each document represents a single website page. The retriever is tasked with returning a set of relevant documents and the reader will find the appropriate answer from them. DPR[3] and RoBERTa[4] are used for building the retriever and reader respectively. The build model can be deployed on a web server and the website can communicate with the model using an API.**

***Index Terms*—Deep Learning, NLP**

## I. Introduction

A website can be considered a wealth of information, providing everything that a customer could possibly want to know. However, even though the information is readily available at the customer's disposal, today's busy customers don't want to go digging around for information or answer on a website. Rather than wasting valuable and limited time searching, the customer should be able to interact with the website and get the information more efficiently.

Rather than wasting time searching for required information from the website the chatbot makes things easy for the customers. A chatbot is a system that gives appropriate answers to questions expressed in natural languages such as English. The different types of chatbots that can be built for websites are Menu-Based chatbots, FAQ-Based chatbots, and Keyword-Based chatbots. The most basic sort of chatbot now in use is one that is based on a menu or a button. Most of the time, these chatbots are glorified decision tree hierarchies that are displayed to the user as buttons. These chatbots, demand the user to make many choices to get the ultimate response. A FAQ-Based chatbot will compare the user's query with the predefined FAQs and return the answer corresponding to the most similar query. Keyword recognition-based chatbots, unlike menu-based chatbots, can listen to and reply to what users input. To understand how to respond appropriately to the user, these chatbots use customizable keywords and NLP.

Most of the website chatbot types that we looked at are built with pre-build data, that is the website admins have to prepare data specifically for the chatbot. In most cases, the data will either be a set of predefined question answering pairs or rules. The admins will have to go through the website manually and generate possible questions and corresponding answers. If any new data is added to the website then questions related to that data also have to be generated. Another problem with these approaches is that if an entirely different question is asked which is not in the predefined data that may lead to either no answer or a wrong answer.

To solve these issues we propose a chatbot system. In our proposed solution there is no manual work for the website admins. The system will only require the URL of the website, it will create a chatbot with the website data automatically. If new data is added to the website then rerunning the system will update the chatbot with new data. In an event that the model is not able to find an exact answer for the query, it will return the exact link for the webpage where the user could find the answer.

The proposed chatbot model is based on retriever-reader architecture. Given several documents and a query, the retriever aims to select a set of documents which may contain the answer. The selected documents are given to the reader for extracting the answer. In our proposed solution we have used Dense Passage Retriever(DPR) as the retriever. A RoBERTa model pertained with the SQuAD2 dataset is used as the reader. The documents we collect will be stored as embedding on a document store for efficient search and retrieve. In our proposed solution we have used FAISS(Facebook AI Similarity Search) as the document store.

The output of the project will be a system that can crawl any website and create a chatbot based on that data. The output of the system will be an API that the websites can interact with. The website can send the customers query through the API and the system will return the answer and the specific website page for finding more information.

As we have discussed above we have utilized the retriever-reader architecture for creating the chatbot. Instead of this two-stage approach, we could have also tried a single-stage

approach. That is initially convert all the sentences from the documents to embeddings and store the values in a document store like FAISS. For finding the appropriate answer the question encoding and stored sentence encodings could be compared for similarity and the sentence with more similarity could be returned as the sentence. By choosing an indexing method like IVF(Inverted File Index) we could get a good tradeoff between speed and quality of similarity searching. As an extension of the project this approach could be tried and compared with our proposed method.

## II. Literature Survey

Customer support is one of the most widely used functions of a chatbot. In [6] they propose a conversational AI system that helps a real-estate company to predict its client's contact motive. The main objective of the model is to differentiate a question and forward it to the corresponding department of the company. A common behaviour they identified in their customers is that they start conversations by saying a greeting. This poses a significant challenge for the system as the system tries to predict contact reasons and select a destination department based on this message. To alleviate this issue, they proposed a binary classifier which classifies between a context and non-context message. To classify the context message into appropriate categories they have used a BERT. The main drawback of this system is that the chatbot isn't equipped with providing answers it's only objective is to collect as much information about the query as possible. The answer to the user's query is given by the people.

E-commerce websites are one such sector that has greatly benefited from chatbots. Chatbots are a practical and cost-effective way of answering repetitive questions. [7] Introduces a chatbot system called SuperAgent for e-commerce websites. SuperAgent takes advantage of data from in-page product descriptions as well as user-generated content from e-commerce websites to answer customer queries. SuperAgent has a Fact QA system which is designed to answer questions regarding the facts of the product. The product information is stored in the format of knowledge triples¡, attribute-name, attribute-value¿, where  represents the product name. The other component of SuperAgent is the FAQ search for customer QA pairs. It uses a DSSM model for finding the sentence similarity between the user's query and stored QA pairs. The main drawback to this model is that it is designed specifically for e-commerce sites. The chatbot's knowledge will be limited to a single page. The sentence similarity method used is outdated and better transformer-based methods are available.

[8] Introduces a chatbot to provide an efficient and accurate answer for any query based on university-related FAQs. A prebuild FAQ dataset is created and using techniques like AIML(Artificial Intelligence Markup Language) and LSA(Latent Semantic Indexing) correct answers are generated. AIML is an XML dialect for creating natural language software agents. LSA is a natural language processing approach for assessing relationships between documents and the terms they include by generating a collection of ideas that are connected to the documents and terms. When a user asks a question LSA is utilized to discover likenesses between words as vector representation. The main drawback is that a prebuild FAQ dataset has to be built and questions outside the scope of the dataset will result in a default answer.

Like the previous one [9] is also a chatbot for the education domain. For data, they have collected about 1500 questions and their corresponding answers from educational organizations. From the collected data they have extracted features like number of words in a question, question type, nouns, number of nouns, verbs, number of verbs, Term-Frequency(TF), Inverse Document Frequency(IDF) and TF-IDF. The extracted nouns and verbs assist in preparing a BoW model. Using the collected information they have created a random forest and use it to find suitable answers. Using of non-contextual features is one of the main drawbacks of this method. Random forest is not an ideal method for handling NLP use cases.

Question Answering is the main task of the chatbots. In [10] they provide a comprehensive survey on open-domain question answering methods. In this work, they provide more information about the retriever-reader architecture. Given a corpus of documents and a query the retriever is tasked with extracting relevant documents based on the query. The reader is responsible for extracting the answer from the relevant documents returned by the retriever. This work also mentions the different retriever and reader techniques that could be used based on the use case.

Transfer learning is a machine learning research subject that focuses on storing and transferring information learned while addressing one problem to a different but related problem. [11] Looks into how customer service chatbots can be improved with attention-based transfer learning. What they discovered from the study is that if a chatbot is trained for one domain then transfer learning could be used to build a fairly efficient chatbot for a similar domain. For example chatbot for Xbox queries could be used to build a chatbot for Playstation queries.

## III. Chatbot Model Architecture

The chatbot is the main component of the entire system. It is the one that is responsible for processing the webpage documents and extracting an answer from the document. The model is based on Retriever-Reader architecture. Further details about the architecture will be given in the following sections.

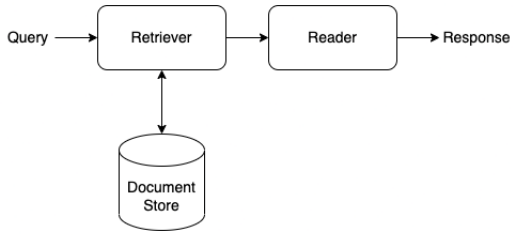


Fig. 1. Chatbot architecture

### A. Retriever

The input to the context model is a set of documents and a query asked by the user. The main task of the retriever is to extract the relevant documents w.r.t to the query asked from the set of documents. We have selected a Dense Passage Retriever(DPR)[3] for this task. DPR uses two BERT models one for encoding passages and the other one for encoding the question. During the training of DPR, a question-context pair is fed into the model, and the model weights will be optimized to maximize the dot product between the two BERT model outputs. Before runtime, we will encode the document using the passage encoder EP(p) and store them in a document store in our case which is FAISS. During runtime, we will use only the question encoder to produce the question encodings EQ(q). Next, the EQ(q) vector is compared against the already indexed EP(p) vectors in our document store where we filter for the vector which returns the highest similarity score.

### B. Reader

The input to the reader is a set of documents selected by the retriever and the query asked by the user. The reader aims at inferring the final answer from the received documents. RoBERTa[4] is used for this task. RoBERTa is an improved version of BERT with higher performance. The performance improvement of RoBERTa is due to longer training time and more data(160GB), removing the next sentence prediction task, training on a longer sequence and dynamically changing the masking pattern applied to the training data. For making the model work for question answering it is pretrained on the SQUAD2 dataset. The output layer of the model is modified to output two values, the start and end index of the answer in a document.

## IV. System Architecture

We understood from the sections above that the chatbot is built using text data from the web pages. This section will look in detail at how a chatbot is built using our system.

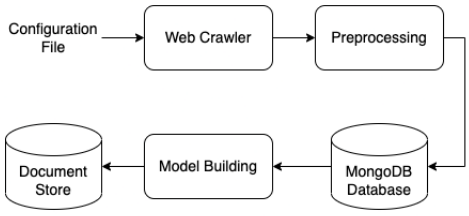


Fig. 2. Chatbot construction workflow

### A. Model Building

*1) Data Collection and Preprocessing:* The data used for creating the model is collected from the website itself. Currently, we are focusing only on text data i.e. information contained in images, audio, or videos is lost. To crawl through all the website's web pages, we use a python framework called scrapy. Scrapy is a free and open-source web-crawling framework written in python. Originally designed for web scraping, it can be used to extract data using APIs or as a general-purpose web crawler. The crawler will be constrained so that it won' be running for infinity and it should not crawl third party websites.

The data collected from websites are usually not very structured and clean. So pre-processing is required before giving the data to the model. The main pre-processing step we perform is converting to lowercase and removing the header and footer data. The header and footer data will be constant on every web page so they have to be removed. After the data is preprocessed it is stored in the MongoDB database. MongoDB is a NoSQL document-oriented database which uses JSON-like documents. MongoDB document is equivalent to a row in SQL but with an optional schema. In our case, each document contains contents for a web page, the corresponding URL and an id to identify the document.

*2) Chatbot Creation:* As mentioned before the chatbot contains two parts retriever and a reader. We are using a pertained DPR(Dense Passage Retriever) as the retriever. The retriever contains two BERT[5] based encoders one for encoding the passage and one for the question. During the training time, the passage encoder will encode all the text data from the database and store them in a document store. We store the embedding in the document store for efficient similarity search. As the reader, we are using a prebuilt RoBERTa.

### B. Model Inference

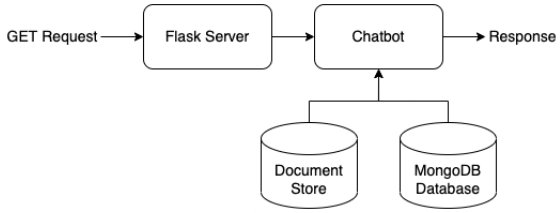


Fig. 3. Chatbot Inference

The different steps involved when a question is asked for the chatbot are explained in the following section. The websites can communicate with the model through an API. The website will send a GET request with the query asked by the user and the model return an answer and a web page URL to get more information. When a request is received it is initially converted to lowercase before forwarding it to the model. Retriever will initially convert the query into an encoding using the query encoder. Then the retriever will search the encodings stored in the document store to find the most relevant document which may contain the answer to the query. The relevance is calculated by taking the dot product between the query encoding and the stored document encodings. The relevant documents along with the query are sent to the reader where the required answer is generated. Along with the answer, the reader will also return the paragraph and the document id of

the source of the answer. Using the id of the document we can retrieve the corresponding web page from MongoDB.

## V. Limitation and Future Scope

Even with the latest techniques used, there are still some limitations to the system. Here we will discuss the main limitation of the system and how these limitations can be overcome in future works. The models we use for the chatbot system are pretrained models. We are not performing any fine-tuning to the models with our data. This causes the chatbot model to miss some domain-specific context meanings. Currently, we are considering text data as the only source of knowledge for the model. But websites contain other sources of information like images, video, and audio files. Limiting the source to only the textual data can affect the performance of the chatbot. The chatbot is designed to handle only the English language which limits it from being deployed on other language websites.

Now we look into how to overcome these limitations. Fine-tuning the models require the data in a question-answer-context format, where context is the paragraph that contains the answer. There are algorithms that could generate question-answer pair given a context or paragraph. After generating this data the chatbot model could be fine-tuned for our domain. To extract text data from images or pdf documents algorithms like tesseract could be used. Audio files could be converted to text using google's speech-to-text APIs. To handle multilingual data we could just replace the current RoBERTa with a one that was pretrained on multilingual data or convert our multilingual training data and query input to English before passing it to the model.

## Result and Conclusion

We were able to create a chatbot that could answer questions based on the website data. The main benefit we see in the system is that there is no data collection or preparation required. Through our testing, the chatbot was able to find answers to questions if they exist in proper sentences. The introduction of our system to a website will help users to find information more efficiently. The users would be able to ask queries directly to the chatbot and get results efficiently. The future scope discussed above could further improve the system and make it more accessible to non-English users.

## References


[1] Schölkopf, B., Williamson, R. C., Smola, A., Shawe-Taylor, J., Platt, J. (1999). Support vector method for novelty detection. Advances in neural information processing systems, 12.

[2] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942.

[3] Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., ... Yih, W. T. (2020). Dense passage retrieval for open-domain question answering. arXiv preprint arXiv:2004.04906.

[4] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

[5] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[6] Barbosa, A., Godoy, A. (2021, November). Augmenting Customer Support with an NLP- based Receptionist. In Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana (pp. 133-142). SBC.

[7] Cui, L., Huang, S., Wei, F., Tan, C., Duan, C., Zhou, M. (2017, July). Superagent: A customer service chatbot for e-commerce websites. In Proceedings of ACL 2017, System Demonstrations (pp. 97-102)

[8] Ranoliya, B. R., Raghuwanshi, N., Singh, S. (2017, September). Chatbot for university-related FAQs. In 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (pp. 1525-1530). IEEE.

[9] Mondal, A., Dey, M., Das, D., Nagpal, S., Garda, K. (2018, November). Chatbot: An automated conversation system for the educational domain. In 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP) (pp. 1-5). IEEE.

[10] Zhu, F., Lei, W., Wang, C., Zheng, J., Poria, S., Chua, T. S. (2021). Retrieving and reading: A comprehensive survey on open-domain question answering. arXiv preprint arXiv:2101.00774.

[11] Bird, J. J. (2021). Improving Customer Service Chatbots with Attention-based Transfer Learning. arXiv preprint arXiv:2111.14621.