Context Based Data Query- Retriever

Abstract—**A data query retriever is a software capable of providing quick and accurate responses to natural language queries, retrieving relevant information from large datasets and integrating data from multiple sources to improve the efficiency and effectiveness of information retrieval. Even though there are so many developments in Natural Language Processing (NLP) and Artificial Intelligence(AI), creating a good data retriever remains a challenge in this field. This software can be used in almost all the working background as data is generated by every company or business or organisations. In general, they need to understand the user's context and deliver accurate and appropriate replies. At present, there are two basic models used in developing this software, Generative based models and Retrieval based models. In this paper the software has multiple layers to learn and preprocess the data and process the data. This can be used without any technical knowledge making the work easier.**

Keywords—Natural Language Processing, bag of words, machine learning, similarity

I. Introduction

In today's information age, our generation rely heavily on search engines and other information retrieval systems to access the information that is needed by the user. However, traditional keyword- based search methods often fall short in providing relevant and accurate results, particularly in cases where the user's query is ambiguous or lacks sufficient context. This is where context-based query retrieval comes in.Context-based query retrieval is an advanced approach to information retrieval that helps using context of a user's query to provide more accurate and relevant results. The development of context-based query retrieval requires sophisticated algorithms and models that can analyse language, understand context, and retrieve relevant information quickly and accurately. This field is rapidly evolving, driven by advances in artificial intelligence and natural language processing, and has many practical applications in areas such as e-commerce, healthcare, and education. Overall, context-based query retrieval represents an exciting frontier in the field of information retrieval, with the potential to revolutionise the way search for and consume information.

II. Literature Review

A deep neural network's hidden layers should be given more weight in order to provide the best results. The paper [1] evaluates the effectiveness of different optimizer algorithms and weight initializer types using a conventional neural network model with one hidden layer and the most popular activation function. The Stochastic Gradient Descent (SGD) optimizer leads the competition in this neural network model, and among the weight initializers, Glorot Normal, He Normal, and Identity initialization strategies provide the highest levels of accuracy. It advises building a chatbot to assist with tasks connected to medicine, such as reminding patients of appointments and assisting with blood pressure readings and adverse drug reactions..The chatbot will employ deep learning models and natural language processing to generate responses that resemble those of a human.

The development of a chatbot for banks that responds to consumer questions in a timely and accurate manner using machine learning (ML) and natural language processing (NLP) techniques. The chatbot is intended to reduce the workload on humans by removing needless enquiries and enhancing customer service [2]. The chat bot's input class is determined by comparing seven classification methods, and the FAQs from bank websites provided the training dataset.

Eight different sentence types from actual WhatsApp interactions between a small business owner and consumers were used to create a corpus for the experiment [3]. The chatbot's algorithm uses Support Vector Machines, a supervised learning method having uses in classification, regression, novelty recognition, and outlier detection. The SVM method creates a hyperplane that may divide points into either class with a considerable gap between them by plotting training data samples as points in space. The SVM algorithm also does non-linear classification using a kernel method.

Additionally, it was found that the best options for the chatbot system included the ”linear” kernel parameter, ”1” for the ”C” and ”gamma” hyperparameters, and the C-Support Vector Classifier, which had the highest accuracy (86.25%). The study also found that by reducing response times from 18.55 minutes to 0.8 seconds, the chatbot system could provide customer care 1391 times faster than the owner of UMKM. The accuracy and precision of the chatbot system were determined to be 87.81% and 90.65%, respectively [3].

SVM-based chatbot development for customer service could speed up information transfer and make it easier for buyers and sellers to communicate. Additionally highlighted are the benefits of clean data in processing classification prediction data as well as the use of TF-IDF to translate language into numerically calculable forms. The study will be useful for researchers and small business owners interested in developing chatbots to reduce response times and improve customer service.Users can easily query, explore, and study multidimensional datasets using a chatbot system dubbed ”TALK to Your Data” [4]. The system's straightforward and user-friendly interface allows users to engage with their data regardless of their level of technical expertise.

It is an effective tool for organisations, researchers, and data analysts who must make sense of enormous and complex datasets. The chatbot system is simple to link with current data management systems, such as databases or data warehouses, and can be quickly tailored to meet the specific needs of each user [4].

There are various possibilities for chatbots in the healthcare industry, including the ability to arrange patient data, provide emergency support, and suggest solutions for minor medical issues. It also examines the drawbacks of modern chatbots, such as their inability to comprehend ambiguous language, spot grammatical errors, and exhibit self-learning abilities. To get over these limitations, chatbots with deep learning capabilities must be developed [5].

A web application for college management is seen to have a chatbot system connected with it that analyses user questions and responds using AI algorithms. In order to create relevant responses from the database, the study trained the chatbot using suitable machine learning techniques and WordNet [6]. The article does draw attention to chatbots' shortcomings in recording all conversational data, which may reduce their usefulness in some situations.

Additionally, it offers a Chatbot system that instantly communicates information to Indonesian clients using Python programming and the WhatsApp app.The Chatbot system, which is built to function with 15 contacts at once, depends on the server connection's speed for the transmission and reception of messages. Stickers, emoticons, and animated gifs cannot be read by the system [7]. According to the paper, the system might be improved by include a random message mechanism.

Reference [8] gives a full explanation of how the Rasa Framework was used to build an SPMB chatbot. The website for "Politeknik Siber dan Sandi Negara" is dedicated to responding to questions about the admission of new students. The chatbot is created using the Dual Intent and Entity Transformer (DIET) model classifier, which is based on deep learning techniques and attention-based architecture. To create the dataset used to train the algorithm, existing chat data was mixed with information from university social media. When the chatbot is assessed using accuracy metrics and F1 scores, the precision value is 0.99, the recall value is 1.0, and the F1 score is 0.99.

Reference [9] describes the creation of a chatbot using the Rasa Framework for college inquiries. When students have questions about admissions, the chatbot promptly and accurately responds. The study uses various metrics to assess the model's accuracy, including F1 score, precision, and accuracy, which are all calculated to be 0.628, 0.725, and 0.669, respectively.It also emphasises how using Rasa has advantages over other chatbot frameworks, such as requiring fewer datasets and requiring less time to train models. Rasa NLU and Rasa Core, two open-source Python chatbot libraries, are mentioned in the literature review sections of numerous publications.

The establishment of human-chatbot relationships (HCRs) is examined in [10] in the context of the Social Penetration 8.Theory, which describes how relationships progress through self-disclosure. Since users usually feel more comfortable sharing personal information to chatbots, the review emphasises the significance of self-disclosure in HCRs. In the study, which examines users' interactions with a chatbot named Replika, researchers discovered that HCRs commonly grow as users' trust and quantity of self-disclosure increase and are motivated by curiosity. Regarding how HCRs affected their larger social surroundings, participants' opinions varied; some claimed to feel stigmatised by them.The paper makes recommendations for other lines of investigation, such as lengthy and experimental investigations that focus on the importance of chatbot features.

By providing clients with customised and efficient services, chatbots have the potential to disrupt a number of industries, according to the analysed articles. Chatbots must be developed and designed well, though, in order for users to find them useful. Particularly emotional intelligence and personalization can increase user pleasure and engagement. Furthermore, the employment of chatbots in multidimensional data analysis could be a beneficial tool for academics and professionals in a variety of fields.The research contributes to our understanding of a phenomenon that is likely to remain significant in the future.

III. Framework and system design

A. Architecture Diagram

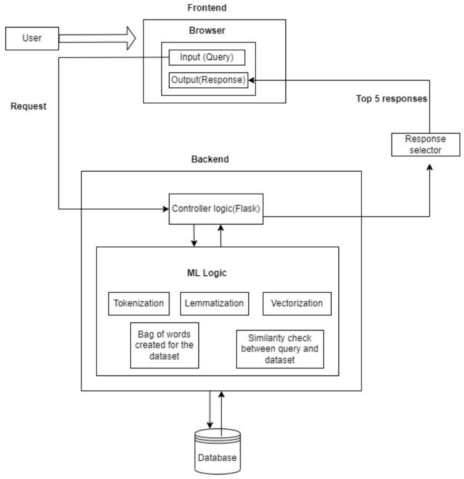


Fig. 1. Architecture diagram

Here in Fig. 1 the user will interact with the system through a web application, where the user can enter their query in a provided text box. When the user submits their query, it will be processed by the bot controller logic, which is implemented using the Flask framework. This logic will handle user requests and provide answers to queries as responses.

The query will then be sent to the business logic and machine learning logic. The business logic will perform preprocessing of the user input query using the Natural Language Processing (NLTK) library, including tokenization, removal of unnecessary spaces and stop-words, and extraction of lemmas for each token. The text-format query will be converted to vectorized format using vectorization techniques.

The machine learning logic will apply a classification algorithm to the transformed query to determine the class it belongs to. This classification algorithm will be based on a previously saved model executed on train data. All questions from the input data having the same class as the retrieved class will be fetched, and cosine similarity will be applied to these questions. Based on the similarity values obtained, the system will return the most similar answer to the user as a response.

B. Use Case Diagram

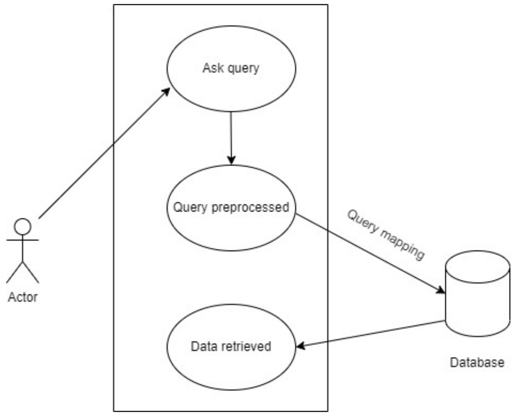


Fig. 2. Use Case diagram

Here, in Fig. 2 the Actor is the user who is asking the query to the system using a natural language. After that the query is preprocessed with various steps, which is mentioned ahead in the paper. The data derived after the preprocessing is mapped using cosine similarity to the database. Then, the information that matches the most with the query is returned to the user.

C. Class Diagram

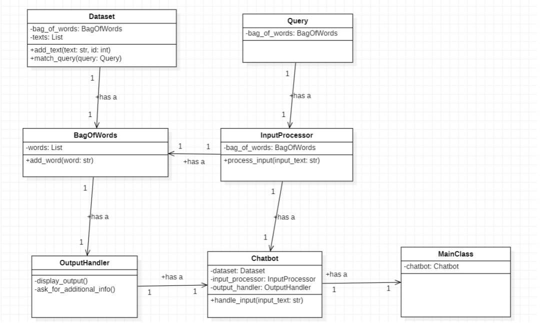

Dataset
-bag_of_words: BagOfWords
-texts: List
+add_text(text: str, id: int)
+match_query(query: Query)
Query
-bag_of_words: BagOfWords
BagOfWords
-words: List
+add_word(word: str)
InputProcessor
-bag_of_words: BagOfWords
+process_input(input_text: str)
OutputHandler
-display_output()
-ask_for_additional_info()
Chatbot
-dataset: Dataset
-input_processor: InputProcessor
-output_handler: OutputHandler
+handle_input(input_text: str)
MainClass
-chatbot: Chatbot
+has a
1

Fig. 3. Class Diagram

Here, in Fig. 3, the class diagram shows a system that analyses user inquiries using natural language processing (NLP) and produces helpful responses. To do this, the system consists of a variety of classes that cooperate.The Chatbot class, which is the primary class that engages with users, is the foundation of the system. An InputProcessor object is used by the Chatbot class to handle user requests. The user's query is tokenized and stemmed by the InputProcessor object, which creates a BagOfWords object to describe it.

The dataset that the Chatbot uses to match the user's inquiry is represented by the Dataset class. The dataset is represented as a BagOfWords object that is part of the Dataset class.The output that is shown to the user is represented by the OutputHandler class. It accepts the Chatbot object's output and prepares it for display.In general, the system operates by employing NLP techniques to process the user's query, matching it against the dataset, and producing the relevant output. The Chatbot class serves as the hub, coordinating all other classes and acting as the user's primary interface.

D. Sequence Diagram

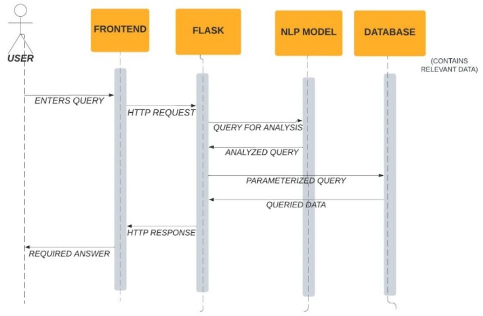


Fig. 4. Sequence Diagram

Here, in Fig. 4, Sequence diagram is shown which illustrates the flow of messages and actions between the different objects involved in the context-based data query retriever system. The system consists of five main objects: the User, Frontend, Flask, NLP ML Model, and CSV File. The User begins by entering a query into the input field of the Frontend. The Frontend then sends an HTTP request with the query data to Flask. Flask receives the request and sends the query to the NLP ML Model for analysis. The NLP ML Model processes the query and returns the analysed query back to Flask. Flask then queries the CSV File for the relevant data based on the analysed query, and the CSV File returns the queried data to Flask. Flask then sends an HTTP response back to the Frontend with the queried data. The Frontend receives the response and displays the queried data to the User. Throughout this process, certain conditions must be met. For instance, the CSV file must have the relevant information Flask needs, and the NLP ML Model must successfully process the query and return the necessary information. If any of these conditions isn't met, the system might not be able to get the proper data or might have problems.

IV. Implementation

1. **Pre-processing** : Using the NLTK tool for Natural Language Processing, the dataset is preprocessed. Because user input will be given in natural language to aid the machine in understanding this language, NLP is used. To further

decrease ambiguity, we process the data and query, which includes the following steps: Tokenization:A list of words is generated using the user's input query and then added to the database. Stop words should be eliminated to increase system performance. Most common terms that don't need to be considered during processing, such ”want,” ”are,” and ”can,” will be removed. Lemmatization: Using the WordNet lemmatizer, each token is assigned a lemma, or the word's root form. For instance, the terms ”organisation” and ”organising” both share the lemma ”organ”.

2. **Vectorization**: The database and the text data in this case are vectorized using the Bag of Words (BOG) concept. It is a method for preparing the text for the input of our machine learning algorithm. All of the articles are used to construct a vocabulary for the BOG model, which then models each one by keeping note of how frequently each term appears.

3. **Developing learning model**: In this stage, algorithms for vectorization, classification, and natural language processing are combined to produce a model that is saved for later use. Every time a new query is received by the system, it searches the stored model to compare it to it and identify the query's class. By reducing the need to train the models each time a new query is made, this shortens the processing time.

4. **Query mapping and getting answers (using Cosine Similarity)**: Once the classifier gives us the class, all the questions are extracted that have this class from our data set. Later cross checking for cosine similarity of the user's query with these extracted questions is done. Then the answer of the most similar question is chosen as response to the user's query and is returned to him.The program has a set of threshold on values of cosine similarity measure for handling queries that are out of domain.

V. Experiments

Dataset: The dataset used for the Context based data retrieval is a CSV file containing spacecraft data. The CSV files have information about a flight like:

1) Flight Number
2) Launch Date
3) Launch Time
4) Launch Site
5) Vehicle Type
6) Payload Type
7) Payload Mass
8) Payload Orbit
9) Customer
10) Customer Type
11) Customer Country
12) Mission Outcome
13) Failure Reason and more

Using natural language queries entered through a Flask-based HTML/CSS frontend. Our aim is to assess the user-friendliness of the system, and to gather feedback on its usability and effectiveness. Two types of experiments were conducted to achieve these objectives: accuracy and efficiency testing, and user testing. Accuracy and Efficiency Testing: Possible queries that can be asked by the user:

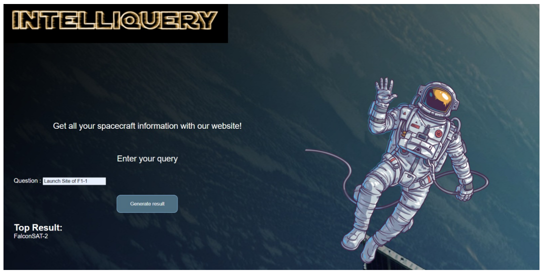
1) What is the Launch Date of flight F1-1?
2) What is the Payload Type of F9-10?
3) Launch Site of F1-3?
4) What is the PayLoad Orbit of F1-5?
5) What is the PayLoad Name of F9-17?
6) Launch Time of F9-5?
7) Landing Type of F9-10?
8) What is the Vehicle Type of flight F9-10?
9) Launch Date of F1-5?
10) Launch Type of F1-3?

The natural language processing approach eliminates the stop words from these queries as they are entered in and only returns the relevant terms. Using cosine similarity, the Bag of Words constructed for our dataset and the Bag of Words produced from these keywords are compared. The best answers are then provided by the model. As an illustration, the NLP model takes the keywords ”launch site” and ”F1-3” from the list of questions above and looks for them in the preprocessed dataset before delivering the result ”Marshall Islands.”

To test the accuracy and efficiency of the system, a set of 10 queries of varying complexity and relevance to the dataset was taken into account. Each query was run through the system and the time taken to retrieve and display the relevant data was recorded. The retrieved data was compared to the expected results based on the queries, to measure the accuracy of the system.

User Testing: To test the user-friendliness of the system and gather feedback on its usability, different queries were given in the frontend to analyse. According to the responses the common themes or issues are identified, and used this feedback to improve the system and address any usability concerns.

Below Fig. 5. shows a query “Launch Site of F1-1” which is then preprocessed by the software and then matched with the bag of words and generates the result i.e. “FalconSAT-2”. In Fig. 6 the query is similar to what we put in Fig. 5 but in a different form i.e. “ What is the F1-1 launch site?” to which our software gives the same result as above. Therefore, it can be concluded that the software is accepting natural language queries and returning correct answers. In Fig. 7 the query



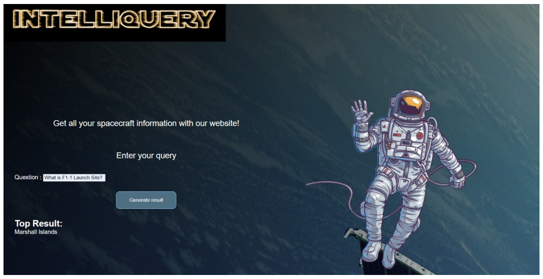
Get all your spacecraft information with our website!

Enter your query

Question : Launch Site of F1-1

Top Result:
FalconSAT-2

Fig. 5. Query 1


INTELLIQUERY

Get all your spacecraft information with our website!

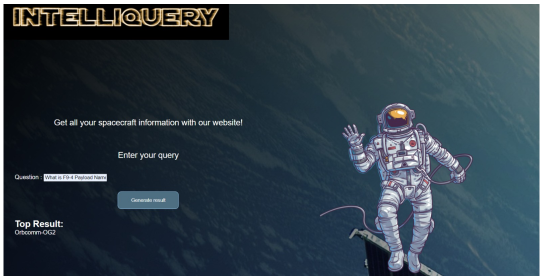
Enter your query

Question : What is F1-1 Launch Site?

Top Result:
Marshall Islands

Fig. 6. Query 2

shown is “ What is F9-4 PayloadName?” to which our software preprocess the query and matches it with the bag of words and the result is displayed as “Orbcomm-OG2”


INTELLIQUERY

Get all your spacecraft information with our website!

Enter your query

Question : What is F9-4 Payload Name

Top Result:
Orbcomm-OG2

Fig. 7. Query 3

VI. Results

We evaluated the accuracy of our model by comparing the predicted answers to a series of questions with the actual answers. The model's performance in generating the correct answers is shown by the accuracy score. The model generated anticipated answers for the identical set of questions. These predictions were made by putting the code into action. The accuracy score was calculated by comparing each predicted response to the corresponding ground truth response. If the anticipated and actual responses lined up, the forecast was deemed to be correct. The total number of questions was divided by the quantity of right guesses to obtain the accuracy score. The accuracy score for the TabularQA predictions was calculated to be 80%, indicating the proportion of correctly answered questions out of the total questions evaluated.

$$AccuracyScore = \frac{Numberofcorrectpredictions}{totalnumberofquestions} \quad (1)$$

The accuracy score provides a quantitative measure of the performance of the model in terms of generating accurate answers to the given questions.

VII. Conclusion

The proposed approach would be a first step in developing an intelligent query handling system that would allow users to query the dataset without previous technical expertise. Traditional query languages can be difficult for non-technical people to traverse; our method aims to provide a more user-friendly and intuitive experience. Before retrieving the required data, the system assesses and understands the user's query using natural language processing algorithms. The final result is a data searching experience that is more efficient and user-friendly.

TABLE I
Your table caption

Question	Ground Truth Answer	Predicted Answer
Where did SpaceX CRS-9 launch from?	Cape Canaveral AFS LC-40	Cape Canaveral AFS LC-40
What is SES-8 Payload Mass(kg)?	3170.0	3170.0
What is the launch site of flight number F1-1?	Marshall Islands	22:30
Who is FalconSAT-2 customer?	DARPA	DARPA
What is Flight Number F1-2 Failure Reason?	Engine Shutdown	Collision During Launch
What is CASSIOPE Landing Type?	Ocean	Ocean
What is FalconSAT-2 Mission Outcome?	Failure	Failure
What is SES-8 Customer Country?	Luxembourg	Luxembourg
What is SpaceX CRS-10 Launch Site?	Cape Canaveral AFS LC-40	Cape Canaveral AFS LC-40
What is Dragon Spacecraft Qualification Unit Launch Date?	04-Jun-10	04-Jun-10

VIII. Future Scope

Further study may improve the accuracy of natural language query processing by increasing the amount of training data available to the system and by enhancing the algorithms utilised for query analysis. is the programmeThe combination of cosine similarity and semantic similarity is another option. By including voice-based inquiries into the course, there is also possibility for improvement. Finally, by include objective measurements in the review process, the reliability of the results may be increased.

IX. References

References

[1] G Krishna Vamsi, Akhtar Rasool, and Gaurav Hajela. Chatbot: A deep neural network based human to machine conversation model. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2020.

[2] Chaitrali S Kulkarni, Amruta U Bhavsar, Savita R Pingale, and Satish S Kumbhar. Bank chat bot–an intelligent assistant system using nlp and machine learning. *International Research Journal of Engineering and Technology*, 4(5):2374–2377, 2017.

[3] Dylan Malvin, Abdul Haris Rangkuti, et al. Whatsapp chatbot customer service using natural language processing and support vector machine. *Int J Emerg Technol Adv Eng*, 12(3):130–136, 2022.

[4] Maria Helena Franciscatto, Marcos Didonet Del Fabro, Celio Trois, Luis Carlos De Bona, Jordi Cabot, and Leon Augusto Gonçalves. Talk to your data: a chatbot system for multidimensional datasets. In *2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC)*, pages 486–495. IEEE, 2022.

[5] Soufyane Ayanouz, Boudhir Anouar Abdelhakim, and Mohammed Benhmed. A smart chatbot architecture based nlp and machine learning for health care assistance. In *Proceedings of the 3rd international conference on networking, information systems & security*, pages 1–6, 2020.

[6] Hrushikesh Koundinya, Ajay Krishna Palakurthi, Vaishnavi Putnala, and Ashok Kumar. Smart college chatbot using ml and python. In *2020 International Conference on System, Computation, Automation and Networking (ICSCAN)*, pages 1–5. IEEE, 2020.

[7] Achmad Ramaditiya, Suci Rahmatia, Aris Munawar, and Octarina Nur Samijayani. Implementation chatbot whatsapp using python programming for broadcast and reply message automatically. In *2021 International Symposium on Electronics and Smart Devices (ISESD)*, pages 1–4. IEEE, 2021.

[8] Lia Fauzia, Raden Budiarto Hadiprakoso, et al. Implementation of chatbot on university website using rasa framework. In *2021 4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pages 373–378. IEEE, 2021.

[9] Siddhant Meshram, Namit Naik, VR Megha, Tanmay More, and Shubhangi Kharche. College enquiry chatbot using rasa framework. In *2021 Asian Conference on Innovation in Technology (ASIANCON)*, pages 1–8. IEEE, 2021.

[10] Marita Skjuve, Asbjørn Følstad, Knut Inge Fostervold, and Petter Bae Brandtzaeg. My chatbot companion-a study of human-chatbot relationships. *International Journal of Human-Computer Studies*, 149:102601, 2021.