ISRDO

Deep Learning Based Natural Language Processing E-Commerce Chatbot

Rudri Jani

Department of Computer Science and Engineering with
Specialization in Data Science
Ahmedabad University
Ahmedabad, Gujarat, India

Abstract

Nowadays, chatbots are becoming popular because it feels like talking with humans in natural language in live chat. This research aims to develop an eCommerce chatbot to help customers buy different pet products from online pet stores. The system uses deep learning and natural language processing concepts like text classification, intent classification, multiclass classification, named entity recognition, etc. The system will guide the customer through purchasing the products by asking questions and replying. The system will also extract the entities from the user queries through NER and display the products accordingly to the user's request. It will interestingly engage the user in buying pet products by solving the user's query and providing a quick response to the user. This will also increase sales, raising the revenue of the company.

 Keywords**:** *chatbot, deep learning, natural language processing, named entity recognition, eCommerce, text classification, multiclass classification, rule-based chatbot, ElasticSearch, MongoDB*

1. Introduction

After the COVID-19 pandemic, eCommerce's rise has positively impacted pet stores' online business. Customers can shop at their convenience, and the product is delivered to their doorstep. Also, the variety of pet products and items available at a reasonable rate attracts people to visit the online store.

An e-commerce company provides live customer support, but the traffic on their eCommerce website is unpredictable, and costs are high. The proposed architecture of the chatbot has been developed using AI. This chatbot is created using the concepts of DL-based NLP to understand the user's intent and guide the user to place the order.

The research is carried out for an eCommerce website that sells various products for pets and farm animals. To help customers find a suitable product for their pets, the AI and ML-based NLP chatbot is developed. The data used to train the models is collected through user queries and website datastore.

2. Methodology

First, data was manually prepared for NER and PR to train the model. The format of the data is JSON for PR and sentences for NER. The product data was collected from Aniforte's online pet store to find the products. Custom models using DNN and Spacy were trained to recognize patterns and named entities. MongoDB is used as a database. ElasticSearch is used to search the products.

Deep learning approach for pattern recognition model
Chatbots can be created in two different ways: [1] [2]

1. Rule Based

    Rule-based chatbots are created using if-then-else conditions. They are very deterministic and operate on a cause-and-effect methodology

2. Self-Learned

    The model is trained with a large amount of relevant data for self-learning chatbots. When it is necessary to reply to the user by comprehending their intention, the inputs are tweaked.

    Here the rule-based machine learning(RBML) approach is used to create the chatbot. When there is a need to identify whether the input should detect any animal model, call the NER model, query to ElasticSeacrh, or add the product to the cart, the rule-based approach is used; apart from these conditions, the pattern recognition model is called.

2.1 Named Entity Recognition

NER is used to identify items, animal types, product prices, quantity, product subcategories, and so on.

2.2 Architecture

The below architecture diagram shows the visual representation of all elements used to build the project by abstracting the overall outline of the system and relationships and constraints. Around 1GB of data for the pattern recognition model is stored in the JSON file. Around 1GB of data for training the entity recognition model is stored in MongoDB.
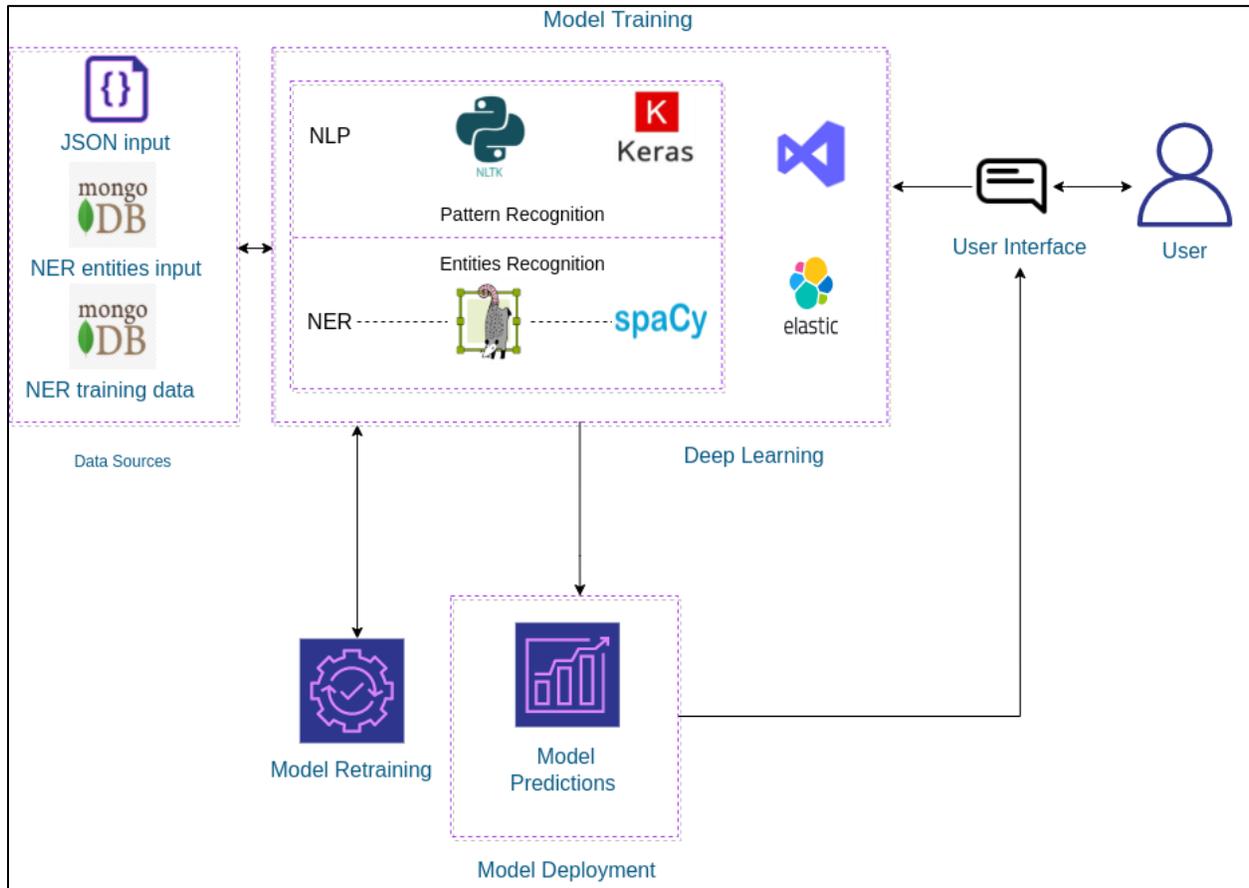
Figure 1: Architecture diagram of Chatbot

The model training part is divided into two parts: one for the PR model and the other for NER. the preprocessing of the PR model is done using the NLTK library, and then the model is trained using TensorFlow and Keras. For the NER model, first, the labeling task is performed using the Label-Studio tool, then the blank English model is trained using Spacy. Elasticsearch is used to search the products. The user interface is designed with the Tkinter Python library, which will take input from the user, and output will be shown there.

2.3 Proposed Approach

As soon as the user starts the chat with the bot, first of all, the time interval will be checked; if it is greater than *5 minutes*, a new chat will start.

The connection with MongoDB will be made, and the trained models will be loaded from the path specified.

The user input will be given to the pattern recognition model, and the model will identify the intent of the model.

The chat history will be added to the *chats* table in the database. It will store input, date/time, main tag, the context of the tag identified, and bot response as a record.
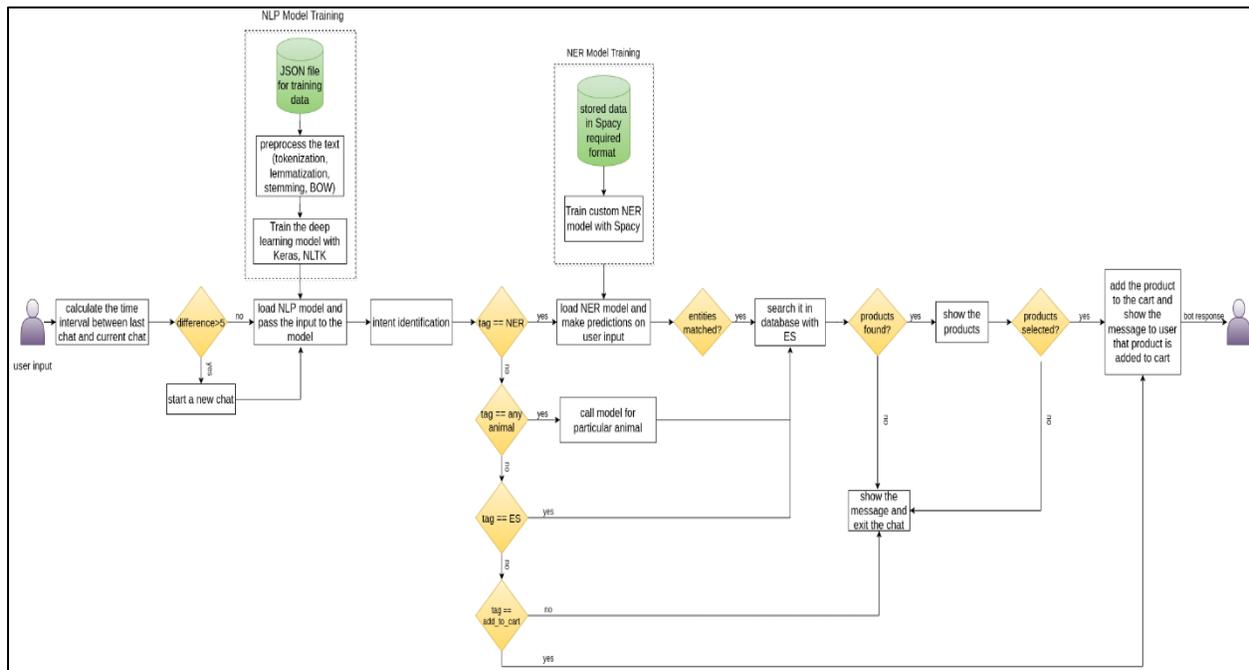
Figure 2: Proposed Flowchart of the System

If the user wants to buy the product, then it will add the product to the cart. In this case, the add_to_cart tag will be identified, and an item will be added to the database. If the user does not want to buy products from the list, they can type exit and quit the chat.

Data Preparation and Preprocessing

a) The data preparation is done by creating the data manually for PR and NER. The preprocessing of data is done first using the NLTK library. The different NLP preprocessing concepts like stemming, lemmatization, and BOW is used. The JSON file contains *tags, patterns, responses, and context* for different intents. References are taken from [3] and [4].

b) Annotating the sentences for training the NER model using Spacy using LabelStudio is carried out. After labeling the data using LabelStudio, this data needs to be converted into the spaCy required format, by which the custom model can be trained using it. The data is then stored in the MongoDB database

c) The pet products data collection task was carried out using a secure web scraping method of the client's website and stored in a data frame. The data collected from Aniforte - an online pet store, is translated into English as it was in the German language. Then the unnecessary columns were removed from it, which were in the German language.

Training the Model

a) Pattern Recognition:

The pattern recognition model was trained using data preprocessed in the first step. The BOW is given as an input to the PR model input layer. This model is trained from scratch. [5]

Here, the DNN is made up of *sequential*, *dense*, and *dropout* layers. As it is a multiclass classification problem of identifying intent, the first layer contains 128 neurons, the second layer contains 64 neurons, and the third output layer contains the number of neurons equal to the number of intents to predict output intent with softmax. The dropout ratio used here is *0.5*. ReLU activation function is used in the internal dense layers. A softmax activation function is used at the last dense layer as it needs to identify one intent from given multiple intents. The evaluation metric used is *accuracy*. *categorical_crossentropy* is used as a loss function. The learning rate of the model is kept at *0.01*. The model is trained by keeping the batch size at *5* and *200* epochs. *Stochastic Gradient Descent* with Nesterov accelerated gradient is used as an optimizer.

b) Named Entity Recognition:

The custom blank NER model is trained from scratch with data prepared from the above step. It uses the entities with a list of (*"start", "end", "label"*) tuples format defining all the entities from the text to train the model [6]. The reference for training the custom model using Spacy is taken from [7]. The blank model is created with the following parameters:

The model is trained with *100* iterations.
The size per batch is *4*.
The optimizer used to train the model is *Stochastic Gradient Descent,* with a dropout ratio of *0.35*.

Indexing and Searching in ElasticSearch

Once the model is trained, the products need to be found. The products can be found by either querying for particular animals or it can be queried for entities recognized using a custom NER model.

Before searching on the database, the data needs to be indexed using elastic search indexing techniques. To index the data, the following steps are followed.

a) Make a custom tokenizer
b) Analyze the created tokenizer
c) Map the analyzer with product attributes
d) Call the bulk indexing API to index multiple data from the JSON file.
e) Enter the search query to find the products.

3. Experimental Result

The following screenshots depict one possible example flow between the user and the chatbot. It explains the procedure of placing an order, beginning with the welcome. It also illustrates how consumers may use the NER idea to immediately make an order.
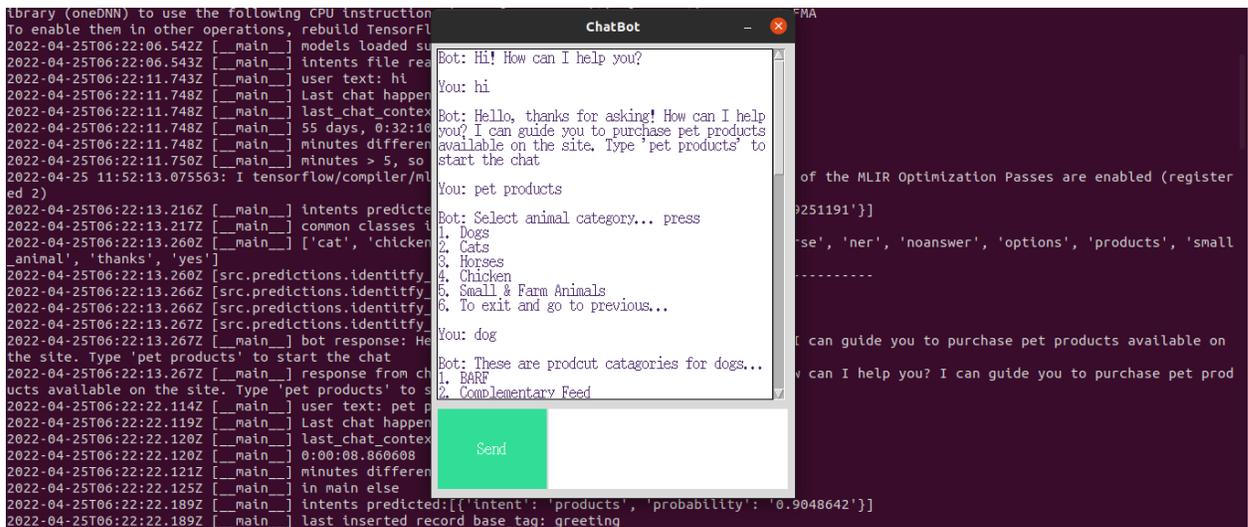
Figure 3: Start the chat and select the animal category

As shown in figure 3, when the user starts the conversation with the bot, it will tell the user to type a message for pet products to show the listing of the animals. Users can select an animal name from the animal list.
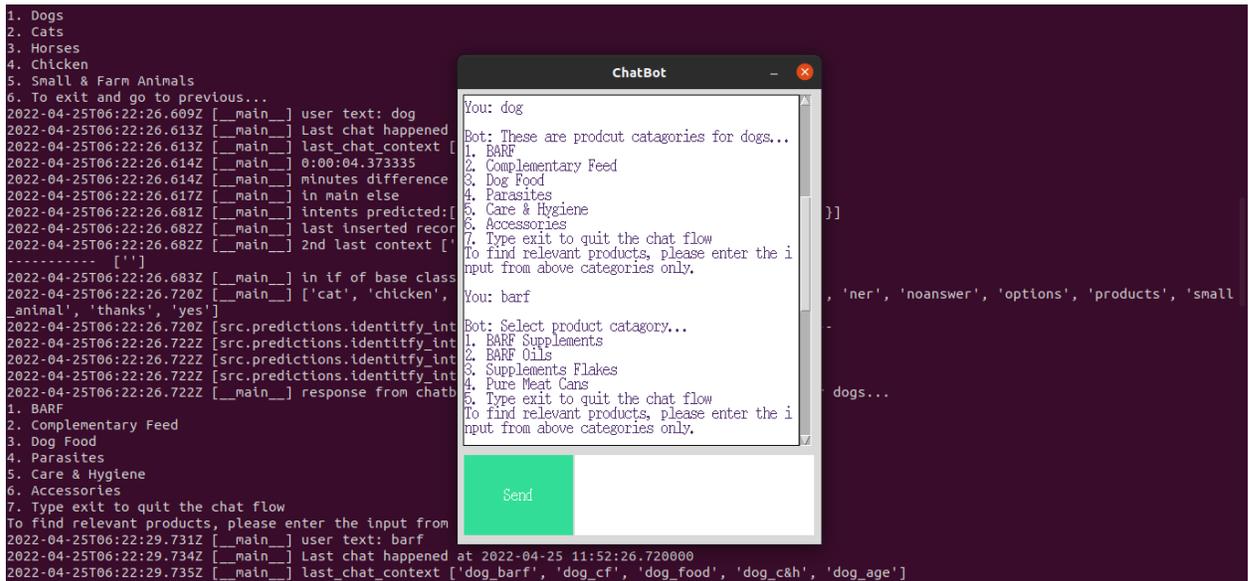

Figure 4: Select product category

Figure 4 shows that if the user selects a dog as an animal, it will show the product categories available for the dogs on the website. At this point, the model will identify the dog's intent, and it will call the pattern recognition model trained for the dog. When a user selects a product from the list, one list will be shown to the user to choose the product subcategory.
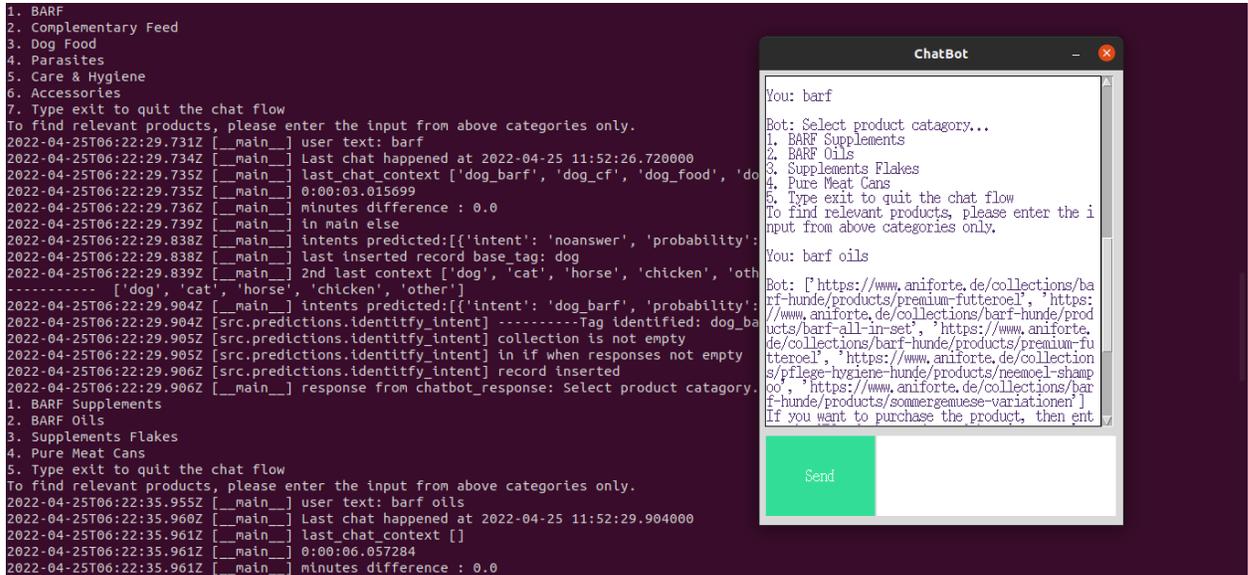
Figure 5: Select subcategory and display the product listing from using query formed for the ElasticSearch

Figure 5 shows that if the user selects the subcategory from the list, the list of products will be shown by searching the products from MongoDB, with the help of a query formed using the elastic search query format. At this moment, the elasticsearch module comes into the picture.
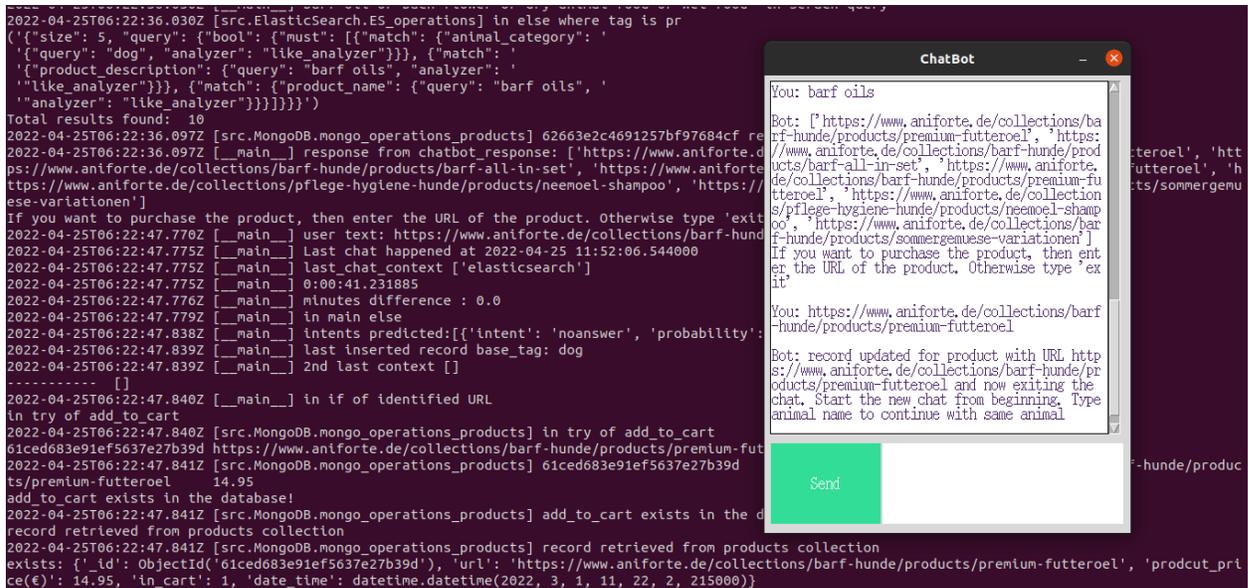

Figure 6: Select the product and add the product to the cart

Figure 6 indicates that if the user wishes to buy the product, they will choose one of the top five goods and add it to the basket. If the product already exists in the add-to-cart table, the status will be updated; otherwise, a new record will be created for the product.
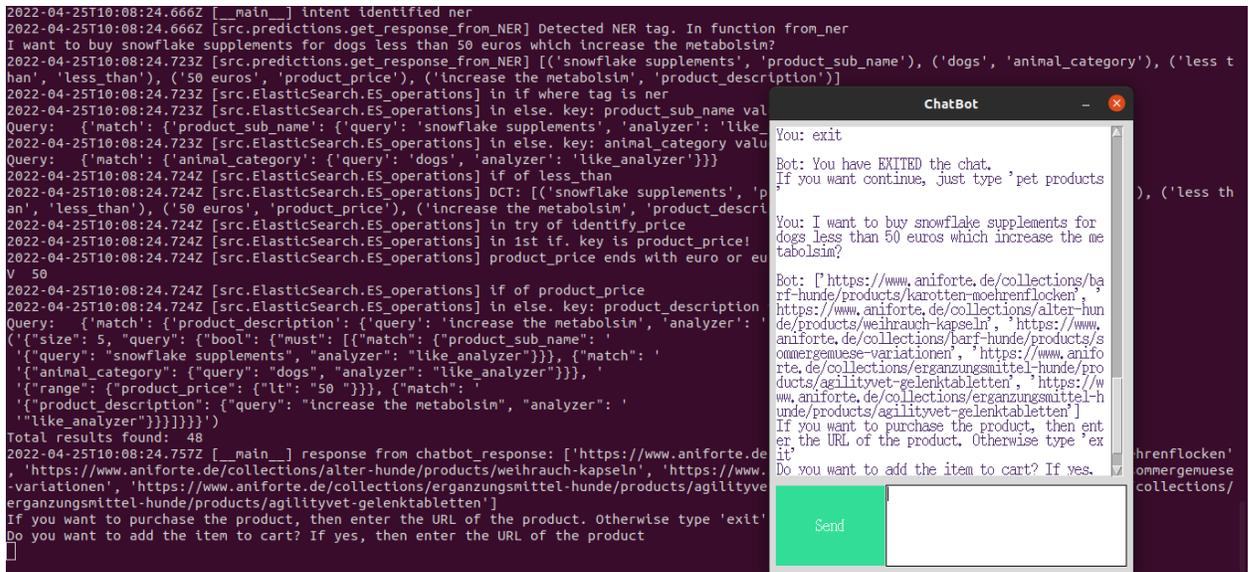
Figure 7: Find the products directly using NER concept

As shown in figure 7, one more module to search the products is using NER. If the user wants to explore the products with specific conditions, they can write the sentence directly in the text area. The chatbot will identify the tag as NER, and the entity recognition model will be called. The model will identify the entities from the sentence, and then these entities will be sent to elastic search. The elasticsearch query will be formed and show the product listing.

4. Conclusion & Future Work

4.1 Conclusion

Creating the hybrid model for the chatbot using rule-based and AI-based approaches rather than only making it rule-based or AI-based has its own advantages. Using a self-learning approach, a chatbot can reply to the user by identifying the intent of the trained model. Using a rule-based approach, the task is performed accordingly if certain conditions are matched. Therefore combining two models and using a hybrid system to develop the chatbot has an advantage. Using the NER model helps to filter out the entities and search the products from the database. Introducing the elastic search helps to improve the search efficiency instead of just using regular expressions to find the products from the database.

4.2 Limitations

Train models with more data to improve performance. As this is the chatbot on an ecommerce website, it cannot replace the google search engine to search and buy the products from the website.

4.3 Future Work

In the future, traffic monitoring and fraud detection tasks will be carried out to identify the traffic on the bot. It can help determine if any user is spamming the bot by keeping it busy. This feature requires the historical and statistical data of the customers chatting on the bot. It will identify the users' buying patterns, and then the model will check if any suspicious activity is detected on the chat.
A question-and-answering module can help customers regarding product details and handle automatic customer support.

References

[1] roy, A. (2020). *Designing a chatbot using Python: A modified approach.* Towards Data Science.

[2] Grunitz, M. (2021, September). *Rule-based {AI} vs machine learning: what's the difference*. Retrieved from wearebrain.com: https://wearebrain.com/blog/ai-data-science/rule-based-ai-vs-machine-learning-whats-the-difference/

[3] Agrawal, R. (2021, June). *Must Known Techniques for text preprocessing in NLP*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/

[4] Harshith. (2021, May). *Text preprocessing in Natural Language Processing using python*. Retrieved from Towards Data Science: https://towardsdatascience.com/text-preprocessing-in-natural-language-processing-using-python-6113ff5decd8

[5] dishashree26. (2019, June). *25 must know concepts for beginners in Deep Learning & Neural Network*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2017/05/25-must-know-terms-concepts-for-beginners-in-deep-learning/

[6] author. (2022, April 26). *Data formats · spaCy {API} documentation.* Retrieved from spacy.io/Data formats: https://spacy.io/api/data-formats

[7] Shrivarsheni. (2022). Training custom NER models in spacy to auto-detect named entities [complete guide]. *Machine Learning Plus*.

[8] mongodb. (n.d.). What is mongodb? *MongoDB*. Retrieved from https://www.mongodb.com/what-is-mongodb.

[9] Foundation, W. (2021, December). *Artificial Linguistic internet computer entity*. Retrieved from en.wikipedia.org: https://en.wikipedia.org/wiki/Artificial_Linguistic_Internet_Computer_Entity

[10] Difference between {IOB} and {IOB2} format? (2022). In *Data Science Stack Exchange*. https://datascience.stackexchange.com/questions/37824/difference-between-iob-and-iob2-format.

[11] What is Elasticsearch? (n.d.). *Elastic*.

[12] Google. (n.d.). Dialogflow documentation | google cloud. *Google*.

[13] Tokenizer reference: Elasticsearch guide [8.1]. (n.d.). *Elastic*.