

E-COMMERCE CHATBOT USING LSTM

Kajendiran M

kajendiranm1@gmail.com

Reg No: 422521104015

University College of Engineering Villupuram

OUTLINE

- **Problem Statement**
- **Proposed System/Solution**
- **System Requirements**
- **Algorithm and Deployment**
- **Who are the End Users?**
- **Result**
- **Conclusion**
- **References**

PROBLEM STATEMENT

The problem in eCommerce customer support is high response times and operational inefficiencies. We implemented an automated chatbot using deep learning techniques to provide real-time, consistent, and cost-effective customer support. This solution aims to enhance user experience and streamline the support process.

PROPOSED SYSTEM/SOLUTION

Chatbot Architecture Design:

Design a chatbot architecture using LSTM networks for natural language processing, complemented by text cleaning, tokenization, and intent classification to deliver precise and relevant responses.

Deep Learning Model:

Utilize LSTM networks to process and understand the sequential nature of natural language data effectively. This involves capturing long-term dependencies in sequences, facilitating the model's understanding and classification of user queries.

Text Preprocessing:

Implement cleaning and organizing of raw text data to ensure clean input for the model. This involves converting text to lowercase, lemmatizing words, and removing stopwords to prepare the text data for model input.

Hyperparameter Tuning:

Utilize Keras Tuner to automate the optimization of model parameters to improve performance and accuracy. This involves systematically exploring the hyperparameter space to find the optimal configuration for the model.

Tokenization and Padding:

Implement tokenization and padding techniques to convert text data into numerical sequences and ensure a consistent input length for the model. This involves tokenizing textual data to convert words into numerical form and applying padding to ensure a uniform sequence length.

SYSTEM APPROACH

System Requirements:

1. Hardware:

- CPU: A multicore processor to handle computational tasks efficiently, supporting the intensive processing demands of deep learning algorithms.
- RAM: Minimum of 8GB RAM to ensure smooth performance during model training and inference, accommodating the memory requirements of large datasets and complex computations.
- Internet Connection: Stable and high-speed internet connection to facilitate data retrieval, model updates, and seamless interaction with external services and APIs.

SYSTEM APPROACH

2. Software:

- Python: Programming language used for developing the chatbot application.
- TensorFlow/Keras: Deep learning libraries used for building and training the LSTM model.
- NLTK: Natural Language Toolkit used for text preprocessing and tokenization.
- scikit-learn: Library for various machine learning tasks like data preprocessing and model evaluation.
- Jupyter Notebook: Interactive computing environment used for prototyping, data analysis, and model development.
- Optional: Django for web-based interface (if applicable).

ALGORITHM AND DEPLOYMENT

1. Data Preparation:

- Retrieved the dataset from Kaggle to serve as the basis for training and testing the chatbot model.

2. Text Preprocessing:

- Cleaned and organized raw text data using NLTK for text processing.
- Removed punctuation and stopwords to ensure clean input.
- Lemmatized words to reduce them to their base form.
- Tokenized text to convert sentences into a list of words for analysis.

ALGORITHM AND DEPLOYMENT

3. Deep Learning Model: Networks

- Implemented LSTM networks to process and understand natural language data.
- Configured the model with two LSTM layers, each having 110 units.
- Utilized the Adam optimizer with a learning rate set to 0.01.
- Employed a dense layer with 208 units followed by a softmax output layer.

4. Hyperparameter Tuning: Keras Tuner

- Utilized Keras Tuner to automate the optimization of model parameters.
- Tuned the number of units in LSTM layers ranging from 50 to 150.
- Explored the number of dense layers in the model from 1 to 20.
- Adjusted learning rates between 0.01, 0.001, and 0.0001 for optimal performance.

ALGORITHM AND DEPLOYMENT

5. Tokenization and Padding:

- Tokenized the text data using Keras Tokenizer to convert words into numerical sequences.
- Applied padding to ensure a uniform sequence length of 200 for model input.
- Set the vocabulary size to 2000 to capture the most frequent words in the dataset.
- Used an out-of-vocabulary token (OOV) to handle words not in the vocabulary during tokenization.

DEPLOYMENT

1. Jupyter Notebook:

The chatbot code is hosted on GitHub by implementing the main code directly on a Jupyter Notebook. This approach allows for easy viewing and execution of the code.

2. Django Web Application:

The chatbot is presented as a user-friendly web-based interface using the Django framework and hosted on GitHub. This method offers a seamless user experience and makes the chatbot accessible via a web browser.

WHO ARE THE END USERS?

- **Online Shoppers:** Individuals looking for product information, recommendations, or assistance with their shopping experience.
- **Customer Support Teams:** Staff members responsible for handling customer queries and providing timely and accurate responses.
- **Website Visitors:** Potential customers exploring the eCommerce website who may have questions or require assistance.
- **Technical Support:** IT professionals or developers involved in maintaining and updating the chatbot's functionality and performance.

RESULT

Executed on Jupyter

jupyter chatbot Last Checkpoint: 4 days ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

```
[nltk_data] Package wordnet is already up-to-date!
You: hi
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
1/1 1s 1s/step
Bot: Hello! Welcome to our e-commerce store. How can I assist you today? (Score: 0.9859431 )
You: track my product
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
1/1 0s 447ms/step
Bot: You can track your product by logging into your account and navigating to the 'Order History' section. There, you will find the tracking information for your shipment. (Score: 0.9072466 )
You: how to leave product review
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
1/1 0s 463ms/step
Bot: To leave a product review, navigate to the product page on our website and click on the 'Write a Review' button. You can share your feedback and rating based on your experience with the product. (Score: 0.9918269 )
You: what is our return policy
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evaluate the model.
1/1 0s 477ms/step
Bot: Our return policy allows you to return products within 30 days of purchase for a full refund, provided they are in their original condition and packaging. Please refer to our Returns page for detailed instructions. (Score: 0.9582239 )
You: bye
Bot: Goodbye!
```

[]:

RESULT

Implemented on Django

Ecommerce Chatbot

The screenshot shows a dark-themed chat interface for an "Ecommerce Chatbot". The conversation starts with a user message "hi" followed by a bot response "Hello! Welcome to our e-commerce store. How can I assist you today?". The user then asks "track my product", and the bot provides instructions on how to track it through the account's Order History section. A "Thanks!" message from the user is also visible. At the bottom, there is an input field with placeholder text "Enter your queries!" and a send button with a right-pointing arrow.

You
hi

Bot
Hello! Welcome to our e-commerce store. How can I assist you today?

Thanks!

You
track my product

Bot
You can track your product by logging into your account and navigating to the 'Order History' section. There, you will find the tracking information for your shipment.

Thanks!

Enter your queries! →

CONCLUSION

Our eCommerce chatbot leverages advanced AI to deliver personalized customer support, enhancing the shopping experience and providing a competitive edge in the digital marketplace. As AI technologies continue to advance, our chatbot is poised to evolve and offer even more innovative solutions to meet the changing demands of the eCommerce industry.

REFERENCES

1. **NumPy**: <https://numpy.org/>
2. **TensorFlow**: <https://www.tensorflow.org/>
3. **Keras**: <https://keras.io/>
4. **scikit-learn (sklearn)**: <https://scikit-learn.org/stable/>
5. **NLTK (Natural Language Toolkit)**: <https://www.nltk.org/>
6. **Keras Tuner**: https://keras.io/keras_tuner/
7. **Pickle (Python's standard library)**: <https://docs.python.org/3/library/pickle.html>
8. **Django**: <https://www.djangoproject.com/>