International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Self-Learning Bot

V R Shushma Reddy, K Yuva Sahithya Preethi, T Mounika, Nisba Kousar

School of Computer Science And Technology, Presidency University, Bangalore, Karnataka
SHUSHMA.20211CST0078@presidencyuniversity.in, KARANI.20211CST0022@presidencyuniversity.in
TIRUMALA.20211CST0015@presidencyuniversity.in, NISBA.20211CST0090@presidencyuniversity.in

ABSTRACT:

In today's technology, imagine having a user-friendly bot which is always ready to help and make user decisions or any queries easier. An interactive tool that is designed to understand and classify the questions that feels like chatting with human. It aim's to classify the user queries as tech or non-tech. Bot has the capability to recognize the specific domain of any query that is posted. MultinomialNaviebayes algorithm is applied here to predict the right answer to the user query. The bot itself is intelligent enough to identify the frequently asked unanswered question and learns mistakes from users feedback by using Machine Learning. Once user can ask a query to the bot it can generate responses based on classification (Tech or non-tech). By learning from previous interactions this bot improves answer relevancy over time, providing accurate responses for both technical and non-technical users . Then the responses will be translate into different languages by applying Transliteration. This research paper shows a new paradigim for bot development by using different tools and techniques.

Keywords: Natural Language Processing(NLP) , TF-IDF vectorization , Machine Learning , MutlinominalNaviebayes , Transliteration , self learning bot.

Introduction:

This paper delves into the requirements and methodologies for creating a self-learning bot that can offer tailored responses based on a user's technical knowledge. By improving answer relevancy through deep learning, the bot ensures that both technical and non-technical users receive appropriate responses. Furthermore, this research highlights how these bots can significantly impact organizations like Cognizant by streamlining internal and external communications.

We have used a simple yet interesting way to show the way a chatbot can learn from the user feedback and give the classification, transliteration and a very interactive bot.

Literature Survey:

Introduction

The integration of artificial intelligence into enterprise communication tools has been on the rise, with a focus on creating bots that can understand and respond to queries from a diverse set of users. As enterprises grow, the need for intelligent bots that can adapt to users' needs, whether they are technical experts or laypersons, becomes paramount. Modern Natural Language Processing (NLP), deep learning, and transliteration cognitive techniques offer the foundation to build such adaptive bots.

This paper delves into the requirements and methodologies for creating a self-learning bot that can offer tailored responses based on a user's technical knowledge. By improving answer relevancy through deep learning, the bot ensures that both technical and non-technical users receive appropriate responses. Furthermore, this research highlights how these bots can significantly impact organizations like Cognizant by streamlining internal and external communications.

Key Machine Learning Techniques in Chatbot Development

Natural Language Processing (NLP)

Natural Language Processing (NLP) forms the foundation of modern chatbots, enabling them to understand and generate human-like responses. Key components of NLP include:

- Text Preprocessing: Tokenization, lemmatization, and removal of stop words to prepare text for analysis.

- Intent Recognition: Identifying user intent using models like Support Vector Machines (SVM) and transformer-based models such as BERT (Devlin et al., 2019).
- Named Entity Recognition (NER): Extracting entities such as dates, names, and locations from user input.
- **User-Centric Responses**: One of the key innovations in this paradigm is the bot's ability to differentiate between technical and non-technical users. Through contextual analysis and real-time learning, the bot can assess a user’s technical ability and tailor its responses accordingly. For example, a non-technical user might receive simplified explanations, while a technical expert would receive more detailed and complex information.
- **Transliteration and Multilingual Capabilities**: Another important aspect of this paradigm is transliteration. The bot can process and respond in multiple languages or character systems, enhancing accessibility for users from different linguistic backgrounds.
- **Self-Learning Mechanism**: The bot’s learning mechanism ensures continuous improvement. Using reinforcement learning algorithms, the bot refines its responses based on user feedback, interaction history, and real-time processing of new data. This results in increasingly accurate and relevant responses over time.

CHALLENGES AND CONSIDERATIONS

Training Data Quality: The bot's performance is heavily dependent on the quality of training data. Poor-quality data can lead to inappropriate learning patterns, reducing response accuracy.

Complexity of Technical and Non-Technical Differentiation: Accurately determining a user’s technical level in real-time can be challenging. Incorrect classification could lead to frustration for users who receive overly simplified or too complex answers.

Privacy and Security Concerns: Handling user data responsibly, especially in terms of learning from past interactions, is critical to maintaining privacy and security standards.

Continuous Learning Maintenance: While self-learning capabilities are advantageous, maintaining the learning model requires ongoing monitoring to ensure that it does not drift into inaccurate or biased response patterns.

Additional Considerations:

Ethical AI Usage: Developers must ensure that the bot’s learning mechanisms adhere to ethical guidelines, especially concerning user data privacy and security.

Bias Mitigation: Careful attention must be paid to avoid biases in the bot’s learning process. Diverse training data should be used to prevent the bot from favoring certain user groups over others.

User Feedback Integration: Mechanisms for collecting user feedback should be integrated to ensure that the bot learns from real-world scenarios and continuously improves its performance.

Methodology

This involves detailing how query classification, response generation and multilingual translations are to be performed within a hybrid system. The paper describes a system that has been developed for user enhanced interaction through intelligent query classification, the generation of response contextually relevant to the user and translating the response into the user’s preferred language.

Architecture of the System:

The system has three major components:

- Query Classifier
- Answer Generator
- Translator

Each element works together to provide a user centric efficient experience. Architecture makes it modular. Each component should work independently and be integrated into one pipeline.

Query Classification

It means classifying the user query into predefined categories like ‘tech’ and ‘non-tech’. The classification enhances and downstream tasks by adapting responses to the query type.

The following is the classification process:

- **Model Initialization:**

For solving the challenge, a pipeline was developed with the help of 'Scikit-Learn', composing a 'TFIdfVectorizer' for feature extraction along with a classifier 'SGDClassifier'. Because the requirement us to be probabilistic, the 'log-loss' function will be specified as a loss parameter.

The classifier used six labeled queries, with three queries for each category to train the classifier in an initial set that formed a starting point.

- **Training and Retraining:**

The baseline data is used to train the pipeline during initialization. It uses user feedback for misclassification; when the system is retrained using 'partial-fit', it will adapt its model to account for every new prediction in light of the feedback it received. It keeps happening within a loop hence continuously learning and further improving.

Answer Generation

The second part involves generating response to user queries. This module uses the GPT-based model by OpenAI, 'got-3.5-turbo', for contextually appropriate and human like answers.

- **API Integration:**

The response is generated from the OpenAI, taking a user query a an input, returning via the chat completion endpoint.

- **Context Management:**

Query classification guides the prompt sent into the model, ensuring that responses are within the intent of the user. For instance the technical query may require the prompt to specify providing a full and technical response.

Translation

The translator component translates responses in the user's preferred language to serve a multilingual client.

- **Translation Mechanism:**

GPT translation OpenAI's GPT model is utilized to perform the translation. Given user input, a dynamic construction with a target language in mind constructs the prompt.

- **Multilingual Support:**

It allows users to request translations in any of the languages covered by OpenAI's model. This helps reach more users.

Feedback Mechanism

The process of refining the system really hinges on user feedback. Following are the feedback loops implemented:

- **Classification feedback:**

The result after classification is confirmed or corrected by the user. Corrections will start a retraining process to include this new data.

- **Satisfaction Feedback:**

Users rate their satisfaction with the generated responses. While this feedback is not automated yet into retraining, it shows perspectives that are useful for the next steps.

Implementation Details

- **Development Environment:**

Programming Language: Python

Libraries: scikit-learn, OpenAI, Pickle

- **Deployment:**

The system is designed to be interactive and run in a command-line interface.

Workflow

- The workflow of the system goes as follows:
- The user has inserted a query.
- The Query Classifier categorizes the query into tech or non-tech.
- Collected classification feedback may serve the goal of retraining if this is necessary.

- The query serves as the basis for the answer produced by the Answer Generator.
- The user can ask for translation of the response in a certain language.
- Satisfaction feedback is collected to measure the performance of the system.

Evaluation

The performance of the system was evaluated based on the following:

- **Classification Accuracy:** Iterative improvements based on user feedback are likely to occur.
- **Response Relevance:** User satisfaction ratings.
- **Quality of Translation :** Informally tested by users who have a knowledge of the target languages.

Limitations and Future work

Limitations:

- Limited initial training data may impact early stage classification accuracy.
- The feedback processing of responses being generated has not yet been made automated.

Future Work:

- Increasing the size of training datasets to improve initial classification.
- Automate the incorporation of feedback into the answer generation.
- Development of a graphical user interface to make it more accessible.

Discussion:

The chatbot implements the active interaction capability, user preference and feedback loops for continuous improvement. The major features are the selection of language preference, such as English and Hindi, and query classification into predefined categories like "tech" and "non-tech.".

The chatbot designed successfully identified the initial query ("how to make cake?") but misclassified it as "tech". Upon receiving user feedback, the system updated its model by retraining to improve future predictions. Then the queries," about cake," were it correctly classified as "tech," showcasing the efficacy of the feedback loop in refining classification accuracy

Chatbot also supports multilingual responses, provides corresponding responses in both English and Hindi based on the user's preference.

One of the strong points of the system is that it can incorporate user feedback directly into its learning process. This iterative mechanism aligns with adaptive machine learning principles, ensuring the model evolves with more interaction data. However, this may also mean slower learning curves in the early stages of deployment when user input is limited.

Improvements that might be expected relate to expanding categories for classification and increasing initial classification accuracy by incorporating other languages. A greater user experience is feasible through enhancements with higher ability orders via advanced NLP.

Overall, this chatbot provides a robust framework for interactive and adaptive conversational systems, making it a valuable tool for applications in customer service, education, and other domains requiring real-time query handling.

Benefits:

1. **Enhanced User Experience**: By providing context-aware responses, the bot improves user satisfaction, as it can respond to both technical and non-technical users appropriately.
2. **Scalability**: Once developed, the bot can be deployed across various departments and scaled to handle numerous user interactions simultaneously, reducing the need for human intervention.
3. **Cost Efficiency**: Self-learning bots reduce the need for continuous manual updates and intervention, saving costs in terms of resources and time for maintaining customer support or technical assistance systems.

Conclusion:

The development of a bot with self-learning capabilities, equipped with modern NLP, deep learning, and transliteration cognitive features, presents a breakthrough in enterprise communication.

These intelligent bots are designed to revolutionize the way organizations interact with their users by providing tailored responses that meet the needs of both technical and non-technical audiences. This capability not only enhances user experience but also improves operational efficiency by automating repetitive tasks and providing quick, contextually relevant responses.

A key feature of these bots is their **self-learning ability**. Over time, they can analyze user interactions and feedback, allowing them to evolve and provide more accurate and personalized answers. This dynamic adaptability is especially crucial in large-scale organizations where users come from diverse technical backgrounds, ranging from experts to novices. By differentiating between the needs of these groups, bots can ensure that the responses are neither too simplistic for technical users nor too complex for non-technical ones. This targeted communication can improve satisfaction and reduce frustration, thereby improving customer service and internal communication workflows.

References:

I. Kalmath, Manjula. (2024). Development of Chatbots. 10. 506-511.

II. Atham, Saira Banu. "VOICE BASED AUTOMATED RESPONSE SYSTEM FOR AGRICULTURAL DOMAIN USING ARTIFICIAL INTELLENGENCE." International Scientific Journal of Contemporary Research in Engineering Science and Management 5, no. 2 (2020): 12-16.

III. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

IV. Jurafsky, D., & Martin, J. H. (2023). Speech and Language Processing (3rd ed.). Pearson.

V. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep Learning. Nature, 521(7553), 436-444.

VI. Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is All You Need. Advances in Neural Information Processing Systems, 30, 5998-6008.

VII. Rajpurkar, P., Jia, R., Liang, P. (2018). Know What You Don't Know: Unanswerable Questions for SQuAD. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 784-789.

VIII. Atham, Saira Banu, Rakesh Ahuja, and Mohammed Farooq Abdullah. "Implementation of Soft Skills for Humanoid Robots Using Artificial Intelligence." In Internet of Things, pp. 257-272. CRC Press, 2022.

IX. A. M. Seeger and A. Heinzl, “Human versus machine: Contingency factors of anthropomorphism as a trust-inducing design strategy for conversational agents,” in Lecture Notes in Information Systems and Organisation, vol. 25, Springer, Cham, 2017, pp. 129–139.

X. A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, “A new chatbot for customer service on social media,” in Proc. of the 2017 CHI Conference on Human Factors in Computing Systems - CHI ’17, 2017, pp. 3506–3510.