Solent University

Department of Science and Engineering

HIV and AIDS Chatbot

Author : Amirhosein Mohammadisabet - 102141537

Course Title : Introduction to Artificial Intelligence

Module Leader : Dr Shakeel Ahmad

Date : 15/01/2024

Table of contents

Table of figures

Overview

The Chatbot for HIV and AIDS Information Retrieval is a Python-based conversational agent designed to provide accurate and accessible information on various aspects of HIV and AIDS. Leveraging a diverse set of machine learning algorithms, including Support Vector Machine (SVM), Naïve Bayes, Decision Tree, Random Forest, k-Nearest Neighbors (KNN), Gradient Boosting, Logistic Regression, and Artificial Neural Network (ANN), this chatbot aims to offer users a reliable source of information on symptoms, causes, diagnosis, treatments, and preventions related to HIV/AIDS.

The primary purpose of this chatbot is to serve as an educational tool, offering users a user-friendly and conversational interface to access information about HIV and AIDS. By harnessing the power of machine learning, the chatbot aims to provide reliable and up-to-date information tailored to the specific inquiries of users.

Despite the wealth of information available, accessing accurate and reliable information about HIV and AIDS can be challenging. This chatbot addresses this issue by offering a conversational platform powered by machine learning algorithms, ensuring users receive precise and relevant information tailored to their queries.

Data

Dataset Overview

The chatbot relies on a carefully curated dataset stored in a JSON (JavaScript Object Notation) file. This dataset is a collection of questions related to HIV and AIDS, categorized into distinct topics such as symptoms, causes, diagnosis, treatments, living with HIV and preventions. there is also a part that contains of unrelated questions. The structure of the dataset is designed to facilitate effective training of machine learning models for information retrieval.

Dataset Format

The JSON file follows a structured format, as shown in figure 1 with each entry representing a question-answer pair.

```
{
"what are some healthy tips?": "Follow doctors orders about your prescriptions",
"how hiv treatment affect my lifestyle?": "hiv treatment can help you live a long and healthy life",
"how long have you been a consult to hiv patients?": "A good number of years treating hiv patients",
"how will you keep track of my immune system's health?": "Based on the number of illnesses contacted each year",
"how totrack of my immune system's health?": "Regular blood tests to check how good your immune system works",
"how will you keep track immune system's health?": "Aging plays a major role with a healthy immune system"
}
```

Figure 1, a sample of JSON file that used in the chatbot

Preprocessing

Before using the dataset for training, several preprocessing steps were undertaken to ensure data quality and consistency.

Text Cleaning

Text cleaning encompasses two essential functions: firstly, the transformation of data from Excel files into JSON format, which includes addressing and rectifying any instances of missing values (NaN); subsequently, the consolidation of all JSON files into a unified and coherent dataset.

Tokenization

Tokenization is the process of breaking down a text into smaller units, usually words or sub words, known as tokens. In the context of natural language processing and machine learning, tokens are the building blocks that algorithms use to understand and process text data.

The TfidfVectorizer, it not only tokenizes your input text but also converts these tokens into numerical vectors using the Term Frequency-Inverse Document Frequency (TF-IDF) method. TF-IDF is a numerical statistic that reflects the importance of a word in a document relative to a collection of documents (corpus).

Dataset Size

The dataset comprises a substantial number of question-answer pairs, providing an ample source for training and evaluating the machine learning models. The distribution of questions across different topics is balanced to ensure each aspect of HIV and AIDS is adequately represented. it contains of about 457 questions and answers.

Algorithms

The Chatbot for HIV and AIDS Information Retrieval employs a diverse set of machine learning algorithms to effectively handle user queries across different topics. Each algorithm is tailored to address specific aspects of information retrieval related to HIV and AIDS.

The algorithms are Support Vector Machines, Naive Bayes, Random Forest, Decision Trees, K-Nearest Neighbors, Gradient Boosting, Logistic Regression, and Artificial Neural Networks (ANN).

Support Vector Machine (SVM)

Support Vector Machine is utilized for its ability to classify questions into different categories based on the provided dataset.

```
elif chosen_model == 'SVM':
    classifier = SVC(
        C=1.0,
        kernel='rbf',
        degree=3,
        gamma='scale'
    )
```

Figure 2, SVM algorithm

The SVM algorithm is trained on the preprocessed dataset, with features derived from the tokenized and lemmatized questions. The algorithm optimizes hyperparameters, such as the choice of the kernel function and regularization parameters.

Naïve Bayes

Naïve Bayes is employed for its simplicity and efficiency in handling text classification tasks.

```
elif chosen_model == 'NB':
    classifier = MultinomialNB(alpha=1.0)
```

Figure 3, Naive Bayes algorithm

The Naïve Bayes algorithm is applied to calculate the probability of each category given a question. The model assumes independence between words, making it suitable for the natural language processing context. Laplace smoothing is applied to handle unseen words.

Decision Tree

Decision Trees are employed to create a hierarchical structure for classifying questions based on a series of decisions.

```
classifier = DecisionTreeClassifier(
    criterion='gini',
    max_depth=None,
    min_samples_split=2,
    min_samples_leaf=1,
    max_features='sqrt'
)
```

Figure 4, Decision Tree algorithm

The Decision Tree algorithm is trained to recursively split the dataset based on the most informative features. Pruning techniques are applied to avoid overfitting, and the resulting tree structure is used for predicting the category of user queries.

Random Forest

Random Forest, an ensemble method, is used to improve the overall classification performance by aggregating predictions from multiple decision trees.

```
classifier = RandomForestClassifier(
    n_estimators=100,
    criterion='gini',
)
```

Figure 5, Random Forest algorithm

A collection of decision trees is trained on different subsets of the dataset. The final prediction is determined by combining the outputs of individual trees, providing a robust and accurate classification.

k-Nearest Neighbors (KNN)

KNN is employed for its simplicity and effectiveness in identifying similar questions.

```
classifier = KNeighborsClassifier(
    n_neighbors=5,
    weights='uniform',
    algorithm='auto',
    p=2
)
```

Figure 6, KNN algorithm

The KNN algorithm calculates the distance between the user's question and those in the dataset. The category of the majority of the k-nearest neighbors determines the classification of the user's query.

Gradient Boosting

Gradient Boosting is used to enhance predictive accuracy by sequentially training weak learners.

```
classifier = GradientBoostingClassifier(
    n_estimators=100,
    learning_rate=0.1,
    max_depth=3,
    min_samples_split=2,
    min_samples_leaf=1,
    subsample=1.0
)
```

Figure 7, Gradient Boosting algorithm

The Gradient Boosting algorithm combines the predictions of multiple weak learners, with each subsequent model correcting errors made by the previous ones. Hyperparameter tuning is performed to optimize the boosting process.

Logistic Regression

Logistic Regression is employed for its simplicity and interpretability in binary and multiclass classification tasks.

```
classifier = LogisticRegression(
    C=1.0,
    penalty='l2',
    solver='lbfgs',
    max_iter=100
)
```

Figure 8, Logistic Regression algorithm

The logistic regression model is trained to predict the probability of each category. A softmax function is applied for multiclass classification, and the category with the highest probability is selected as the final prediction.

Artificial Neural Network (ANN)

Artificial Neural Network is employed for its ability to capture complex relationships in data.

```
classifier = MLPClassifier(
    hidden_layer_sizes=(100,),
    activation='relu',
    solver='adam',
    alpha=0.0001,
    learning_rate='constant',
    max_iter=200,
    batch_size='auto'
)
```

Figure 9, MLP algorithm

The ANN consists of multiple layers, including input, hidden, and output layers. The model is trained using backpropagation and gradient descent, with hyperparameters like the number of layers, neurons, and activation functions optimized through experimentation.

User Interface

The user interface (UI) is a critical aspect of the project, ensuring a seamless and visually appealing experience. Three distinct routes cater to different functionalities, maintaining a cohesive design across all interactions.

In the development of this chatbot, the entire user interface (UI) is constructed using the Flask library in Python, encapsulated within the app.py file. This file seamlessly integrates HTML, CSS, and JS components to enhance overall performance and deliver an optimized user experience.

Flask Application (app.py)

The Flask application serves as the backbone of the chatbot, providing a scalable and dynamic platform. The following routes cater to distinct functionalities:

Standard Responses

The standard response shows as route and handles user queries through a conventional approach, retrieving responses from a pre-defined dataset.

```python
@app.route('/', methods=['GET', 'POST'])
def index():
    user_question = None
    answer = None

    if request.method == 'POST':
        user_question = request.form['question']
        user_question = user_question.lower()
        answer = models.get_response_standard(user_question, dataset)
        qa_history.insert(0, (user_question, answer))

    return render_template('index.html', qa_history=qa_history)
```

Figure 10, The route for standard response

Decision Tree (DT)

Utilizes a Decision Tree model to analyze user queries and generate responses based on learned patterns. This UI is accessible as (/dt).

```python
@app.route('/dt', methods=['GET', 'POST'])
def dt():
    user_question = None
    answer = None

    if request.method == 'POST':
        user_question = request.form['question']
        user_question = user_question.lower()
        answer = models.get_response_ML(user_question, model_dt)
        qa_history.insert(0, (user_question, answer))

    return render_template('dt.html', qa_history=qa_history)
```

Figure 11, The route for Chatbot using Decision Tree

Random Forest (RF)

Leverages a Random Forest model for a more nuanced understanding, capturing diverse patterns in user queries on (/rf).

```
@app.route('/rf', methods=['GET', 'POST'])
def rf():
    user_question = None
    answer = None

    if request.method == 'POST':
        user_question = request.form['question']
        user_question = user_question.lower()
        answer = models.get_response_ML(user_question, model_rf)
        qa_history.insert(0, (user_question, answer))

    return render_template('rf.html', qa_history=qa_history)
```

Figure 12, The route for Chatbot using Random Forest

HTML

The project comprises three HTML files: 'index.html,' designed for displaying standard responses; 'dt.html,' dedicated to presenting responses derived from a decision tree model; and 'rf.html,' tailored for showcasing responses generated by a Random Forest model.

CSS

The CSS code in style.css styles the project's UI, setting fonts, colors, and layouts. Body has margins and a light background, the header has a distinct dark background with white text. Styles for forms, buttons, and input fields provide a cohesive design. The question history section has a clean list layout with a subtle hover effect for interactivity.

Java Script

The JavaScript code within the main.js file of the project utilizes the part to initiate smooth scrolling functionality for questions and responses. This implementation enhances user experience by providing a visually pleasing transition between sections on the webpage.

Functionality:

The Chatbot for HIV and AIDS Information Retrieval is designed to provide a seamless and informative user experience by effectively handling a wide range of questions related to HIV and AIDS. The chatbot's functionality is categorized based on different topics, allowing users to inquire about symptoms, causes, diagnosis, treatments, and preventions. It also allows users to use three different approaches of the HIV chatbot with using different routes.

Topic-Based Inquiry

Symptoms

Users can ask questions about the symptoms associated with HIV, such as early signs, common manifestations, and specific indicators. The chatbot utilizes machine learning algorithms to categorize and retrieve relevant information from the dataset.

Your Question:

what are the common symptoms of hiv?

Answer:

Early symptoms may include fever, fatigue, and swollen glands. in later stages, aids symptoms may include weight loss, recurrent infections, and night sweats.

Figure 13, example query about symptoms

Causes

Inquiries related to the causes and transmission of HIV are addressed comprehensively. The chatbot leverages its algorithms to identify and provide accurate information about the ways HIV is transmitted.

Your Question:

how is hiv transmitted?

Answer:

hiv is primarily transmitted through unprotected sexual intercourse with an infected person, sharing needles or syringes, and from an infected mother to her child during childbirth or breastfeeding. it can also be transmitted through blood transfusions or organ transplants from infected donors, although this is rare nowadays due to screening.

Figure 14, Example query about causes

Diagnosis

Users can seek information on the diagnostic methods used for detecting HIV. The chatbot's algorithms assist in classifying questions and offering informative responses regarding testing and diagnosis.

Your Question:

what are the common methods for hiv diagnosis?

Answer:

Early symptoms may include fever, fatigue, and swollen glands. in later stages, aids symptoms may include weight loss, recurrent infections, and night sweats.

Figure 15, example query of diagnosis

Treatments

Questions related to HIV treatments and antiretroviral drugs are handled effectively by the chatbot. The algorithms categorize inquiries to deliver information on various treatment options.

Your Question:

what are the latest advancements in hiv treatment?

Answer:

Since hiv cannot be cured for now, The available combination therapies are meant to be taken for a lifetime inorder to suppress the viral load to the minimal level. These therapies can come inform of a monthly injection, once or multiple times daily.

Figure 16, example query of Treatments

Preventions

Users can inquire about preventive measures for HIV, and the chatbot's algorithms classify these questions to provide information on strategies for HIV prevention.

Your Question:

what are the best practices for hiv prevention?

Answer:

Abstinence: The simple and easy way of preventing hiv infection is by abstinence from sexual activities especially engaging in vaginal or anal sex. Choose sex that is less risky than anal or vaginal sex. There is little to no risk of getting hiv through oral sex. You can't get hiv from sexual activities that don't involve contact with body fluids (semen, vaginal fluid, or blood).

Figure 17, example query of Prevention

Living with HIV

Living with HIV requires a comprehensive approach to health and well-being. It's important to note that while there is currently no cure for HIV, there are effective treatments that can help manage the virus and allow individuals to lead healthy lives.

Your Question:

how to live with hiv

Answer:

if you manage your condition properly by taking your medicine correctly and avoiding illness, you should be able to live a near-normal life.

Figure 18, example query of Living with HIV

Evaluation:

The performance of the Chatbot for HIV and AIDS Information Retrieval was rigorously evaluated using a variety of metrics to ensure the accuracy and effectiveness of the information retrieval system. The evaluation process involved assessing each machine learning algorithm individually and comparing their overall performance.

Accuracy

The primary metric used to measure the overall correctness of the chatbot's responses. It represents the ratio of correctly predicted categories to the total number of predictions.

Precision

Precision measures the accuracy of positive predictions, indicating the proportion of correctly identified relevant information among all instances predicted as relevant.

Recall

Recall assesses the ability of the chatbot to identify all relevant instances, providing a measure of completeness.

F1 Score

The F1 Score is the harmonic mean of precision and recall, offering a balanced evaluation metric that considers both false positives and false negatives.

Chatbot Evaluation

The evaluation of the chatbot's performance is a crucial aspect of ensuring its efficacy in providing accurate information. The eval() function systematically assesses the chatbot against a predefined dataset, utilizing metrics such as accuracy, precision, recall, and F1-score.

Table 1, Evaluation of all Machine learning algorithms in percent

Algorithm	*Accuracy*	*Precision*	*Precision*	*F1-Score*
SVM	85.75	96.56	85	83
Naïve Bayes	13.16	95	13	9.7
Random Forest	98.25	98	98	97.95
Decision Tree	98.25	98.97	98	97.93
KNN	24.12	71.85	24	15
Gradient Boosting	98	98.88	98	97.67
Logistic Regression	15.57	93	15	10
ANN	97.59	98.5	97.58	97

Comparative Analysis

The Chatbot for HIV and AIDS Information Retrieval has demonstrated robust performance across various machine learning algorithms, showcasing its efficacy in providing accurate and informative responses to user inquiries. The results encompass individual algorithm evaluations, comparative analyses, and insights gained during the evaluation process.

Algorithmic Performance

The provided results showcase the performance of various machine learning algorithms on a given task. Random Forest and Decision Tree algorithms achieved the highest accuracy of 98.25%, demonstrating robust performance. Both algorithms also exhibited high precision and F1-score, emphasizing their ability to correctly classify instances and strike a balance between precision and recall. This suggests that these tree-based models are well-suited for the task, possibly capturing complex relationships within the data.

On the other hand, Naïve Bayes and Logistic Regression demonstrated lower accuracy and precision. Naïve Bayes, in particular, struggled with an accuracy of 13.16% and low precision, indicating that it may not be the most suitable choice for the specific task at hand. Logistic Regression also exhibited limited accuracy (15.57%) and precision, suggesting that its linear nature might not be capturing the underlying patterns effectively. These findings highlight the importance of selecting appropriate algorithms based on the characteristics of the dataset and the task requirements.

Gradient Boosting and Artificial Neural Network (ANN) both performed well, with accuracy rates of 98% and 97.59%, respectively. These results underscore the capability of ensemble methods and deep learning approaches to handle complex patterns in the data. Gradient Boosting displayed high precision, while ANN demonstrated competitive precision and F1-score, showcasing their potential for accurate and reliable predictions. Overall, the performance analysis emphasizes the need for a thoughtful algorithm selection process, taking into account the dataset characteristics and the specific goals of the task.

User Interaction

Feedback from user interactions indicated positive experiences, with users expressing satisfaction with the chatbot's ability to provide relevant and accurate information on a wide range of topics related to HIV and AIDS.

Future Enhancements

While the chatbot has proven successful in its current implementation, there are opportunities for further refinement and enhancement. Future work may include expanding the dataset, exploring additional algorithms, and incorporating user feedback to continually improve the chatbot's performance and usability.

The Chatbot for HIV and AIDS Information Retrieval has laid a solid foundation for providing accurate and accessible information. To further improve its capabilities and user experience, several avenues for future work are identified.

Expanding the dataset will contribute to the chatbot's knowledge base, enabling it to handle a more extensive range of user queries. Additional questions covering emerging topics, recent research, and evolving medical information will enhance the chatbot's ability to provide up-to-date and comprehensive responses.

Continued exploration of advanced machine learning algorithms and techniques can potentially improve the chatbot's performance. Investigating state-of-the-art models, transfer learning approaches, or neural network architectures specific to natural language processing may yield further advancements in accuracy and efficiency.

Incorporating user feedback mechanisms can enhance the chatbot's responsiveness to user needs. Implementing a feedback loop where users can rate responses or provide additional context will enable continuous learning and refinement of the chatbot's knowledge base and algorithms.

Expanding the chatbot's capabilities to understand and respond to queries in multiple languages will broaden its accessibility. Implementing natural language processing models that support diverse linguistic inputs can cater to a more global audience and improve the chatbot's inclusivity.

Conclusion

In the realm of chatbot development, algorithmic performance is paramount for ensuring an effective and engaging user experience. Random Forest and Decision Tree algorithms have proven to be robust choices, boasting high accuracy, precision, and F1-scores. Their ability to navigate complex language patterns makes them well-suited for natural language processing tasks inherent in chatbot interactions. Additionally, Gradient Boosting and Artificial Neural Network (ANN) showcase strong performance, offering advanced capabilities in capturing nuanced language nuances. However, it's crucial to carefully weigh the computational requirements and model complexity against the desired performance.

On the flip side, Naïve Bayes and Logistic Regression exhibited suboptimal results, indicating potential limitations in handling the intricacies of natural language. The linear nature of these models may fall short in capturing the dynamic context of conversations. In conclusion, a judicious choice of algorithms, considering both their strengths and limitations, is vital for developing an efficient and user-friendly chatbot. Continuous monitoring and adaptation to evolving user needs will further refine the chatbot's performance over time.

References

Code and Frameworks:

Scikit-learn. (n.d.). Retrieved from https://scikit-learn.org/stable/

Keras. (n.d.). Retrieved from https://keras.io/

Flask. (n.d.). Retrieved from https://flask.palletsprojects.com/

Libraries:

Python Software Foundation. (n.d.). Python Language Reference, version 3.x. Retrieved from https://www.python.org/

NumPy. (n.d.). Retrieved from https://numpy.org/

Pandas. (n.d.). Retrieved from https://pandas.pydata.org/

Appendix A

How to run HIV and AIDS Chatbot

There is two way to run the chatbot

1- Clone the repository using git clone

https://github.com/amirhoseinmohammadisabet/chatbot-HIV.git

2- Use pycharm, vscode or a Terminal to run app.py with python
3- Go to following links to see the http://127.0.0.1:5000/
4- Use main route for standard questions and answers, (/dt) for decision tree and (/rf) for using chatbot with random forest algorithm.

Or simply using python with runing main.py.