Chatbot in Python using AI: Loading and Preprocessing dataset

Author: THEJAS P J

Reg no: 961621104050

Project Overview

A chatbot is an intelligent piece of software that is capable of communicating and performing actions similar to a human. Chatbots are used a lot in customer interaction, marketing on social network sites and instantly messaging the client. There are two basic types of chatbot models based on how they are built; Retrieval based and Generative based models.

Dataset

- Dataset Link: [Dataset on Kaggle](https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot)

- The provided dataset consists of simple dialogs for training the chatbot.

Phase 1: Data Loading

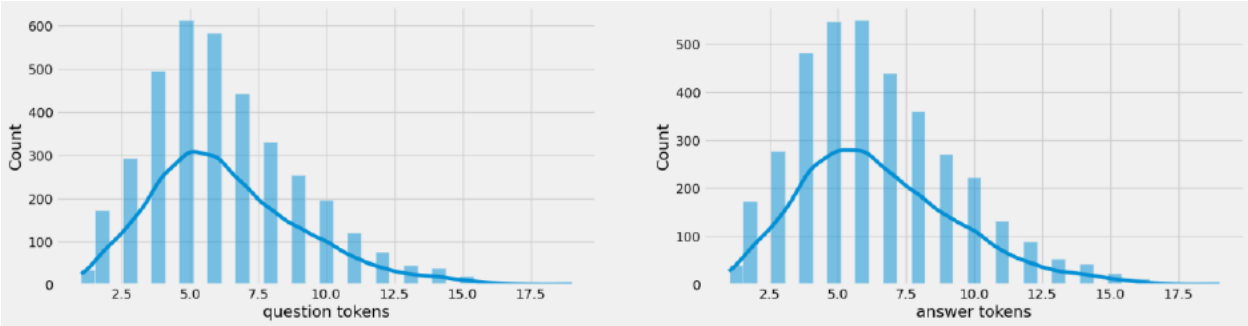
First, make a file name as train_chatbot.py. We import the necessary packages for our chatbot and initialize the variables we will use in our Python project.

The data file is in JSON format so we used the json package to parse the JSON file into Python.

```
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
import json
import pickle

import numpy as np
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout
from keras.optimizers import SGD
import random

words=[]
classes = []
documents = []
ignore_words = ['?', '!']
data_file = open('intents.json').read()
intents = json.loads(data_file)
```



Phase 2: Data Preprocessing:

When working with text data, we need to perform various preprocessing on the data before we make a machine learning or a deep learning model. Based on the requirements we need to apply various operations to preprocess the data.

Tokenizing is the most basic and first thing you can do on text data. Tokenizing is the process of breaking the whole text into small parts like words.

Here we iterate through the patterns and tokenize the sentence using nltk.word_tokenize() function and append each word in the words list. We also create a list of classes for our tags.

```
for intent in intents['intents']:
    for pattern in intent['patterns']:

        #tokenize each word
        w = nltk.word_tokenize(pattern)
```

```
        words.extend(w)
        #add documents in the corpus
        documents.append((w, intent['tag']))

        # add to our classes list
        if intent['tag'] not in classes:
            classes.append(intent['tag'])
```

Now we will lemmatize each word and remove duplicate words from the list. Lemmatizing is the process of converting a word into its lemma form and then creating a pickle file to store the Python objects which we will use while predicting.

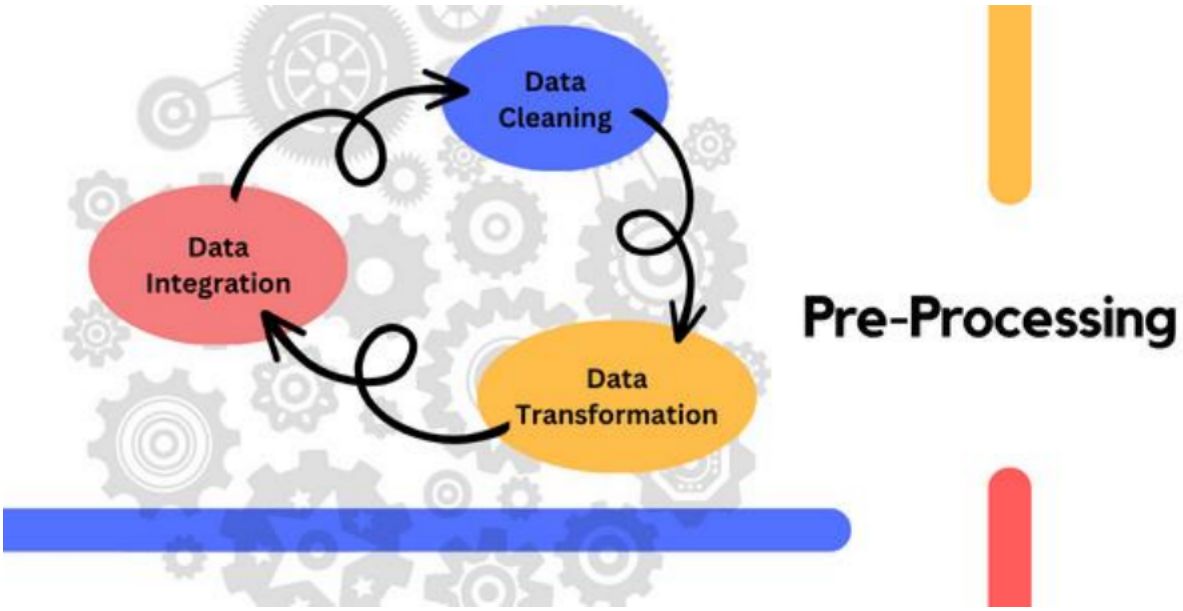
```
# lemmatize, lower each word and remove duplicates
words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in
ignore_words]
words = sorted(list(set(words)))
# sort classes
classes = sorted(list(set(classes)))
# documents = combination between patterns and intents
print (len(documents), "documents")
# classes = intents
print (len(classes), "classes", classes)
# words = all words, vocabulary
print (len(words), "unique lemmatized words", words)

pickle.dump(words,open('words.pkl','wb'))
pickle.dump(classes,open('classes.pkl','wb'))
```

- Data cleaning, including text normalization, tokenization, and removal of special characters.

- Splitting the dataset into training, validation, and testing sets.

use of ensemble methods, such as stacking or bagging, to combine multiple models and improve prediction accuracy.



Conclusion

In conclusion, building a chatbot in Python using an artificial intelligence dataset is an exciting project that can serve a variety of purposes, from customer support to information retrieval and entertainment. By following the steps outlined in the project overview, you can create a chatbot with the ability to engage in meaningful conversations and provide value to users.

However, it's essential to understand that building a successful chatbot is a complex task that involves natural language processing, machine learning, and continuous improvement. The quality of your dataset, the choice of NLP and ML libraries, and the sophistication of your model play a significant role in the chatbot's performance.

As you embark on this journey, be prepared for challenges and be open to iterating on your chatbot to enhance its capabilities. Regular testing, user feedback, and maintenance are critical aspects of creating a chatbot that can truly meet its intended goals.

Ultimately, the development of a chatbot can be a rewarding experience, providing users with a valuable tool for interaction and information. It's a testament to the power of artificial intelligence and natural language processing to create intelligent and engaging applications.