CREATE CHATBOT IN PYTHON

Introduction:

At the most basic level, a chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person.

AI chatbots provide customized responses to users and guide them to the right sources with the help of natural language processing and text recognition.

This also helps in gaining the loyalty of the customers

Problem definition:

The problem is to build an AI powered diabetes production system that uses machine learning algorithms to analyses medical data and predict the likelihood of an individual developing diabetes.

The system aims to provide early risk assessment and personalized preventive measures, allowing individuals to take proactive actions

to manage their health.

Responses :

A bot response is a message your bot sends to the user. In Chat Bot, you can choose from 6 types of bot responses:

- **Text**
- **Random Text**
- **Image**
- **Gallery (Carousel)**
- **Button**
- **Quick Reply**

The bot displays responses in the same order you composed them.

You can apply Filters to your bot responses to trigger them only when a condition is met.

You can decide how fast your chat bot should respond to a user's question using the Delay feature.

Integration:

How Do You Integrate A Chat bot?

1 .Define the Application

2. Choose the Chat bot

3. Finalize and personalize the actions

4. Set up Sentiment Analysis.

5. Create a fallback response.

6. Including the Commonly Asked Questions section.

7. Put the Chat bot to the test.

8.Now its time to put it to use.

Innovation:

Exploring advanced techniques like using pre trained language models to enhance the quality of responses.

Pre-Trained Language Models and Their Applications:

Pre-trained language models have achieved striking success in natural language processing (NLP), leading to a paradigm shift from supervised learning to pre-training followed by fine-tuning.

The NLP community has witnessed a surge of research interest in improving pre-trained models.

This article presents a comprehensive review of representative work and recent progress in the NLP field and introduces the taxonomy of pre-trained models.

We first give a brief introduction of pre-trained models, followed by characteristic methods and frameworks.

We then introduce and analyze the impact and challenges of pre-trained models and their downstream applications.

Finally, we briefly conclude and address future research directions in this field.

Keywords :

- Pre-trained models

- Natural Language processing

1.A brief history of pre-trained models:

The concept of pre-training is related to transfer learning .

The idea of transfer learning is to reuse the knowledge learned from one or more tasks and apply it to new tasks.

Traditional transfer learning employs annotated data for supervised training, which has been the common practice for at least a decade.

Within deep learning, pre-training with self-supervised learning on massive annotated data has become the dominant transfer learning approach.

The difference is that pre-training methods useannotated data for self-supervised training and can be applied to various downstream tasks via fine-tuning or fewshot learning

2. Methods of PTMs:

2.1. Different frameworks and extensions of PTMs :

When working with PTMs, it is essential to design efficient training methods that can fully use unannotated data and assist downstream fine-tuning.

In this section, we briefly introduce some widely used pre-training frameworks to date.

summarizes the existing prevalent pre-training frameworks, which can be classified into three categories: transformer decoders only; transformer encoders only; and transformer decoder–encoders.

A brief description of each category is given below, and more detail is provided in the subsections that follow.

2.1.1. Transformer decoders only :

The objective for language modeling is to predict the next token autoregressively, given its history.

The nature of auto-regression entails the future invisibility of input tokens at each position; that is, each token can only attend to the preceding words.

GPTwas the first model to use the transformer decoder architecture as its backbone.

Given a sequence of words as distribution of the next word with the masked multi-head self-attention of the transformer.

In the fine-tuning phase, the pre-trained parameters are set as the initialization of the model for downstream tasks.

GPT is pre-trained on the Books Corpus dataset, which is nearly the same size as the 1B Word Benchmark.

It has hundreds of millions of parameters and improves SOTA results on nine out of 12 NLP datasets, showing the potential of large-scale PTMs.

GPT-2 follows the unidirectional framework with a transformer decoder that was trained with a larger corpus, namely, Web Text, and 1.5 billion model zero-shot setting.

GPT-3 further increases the parameters of the transformer to 175 billion and introduces in-context learning.

Both GPT-2 and GPT-3 can be applied to downstream tasks without fine-tuning.

They achieve a strong performance by scaling up the model size and dataset size.

Unidirectional language modeling lacks attention on its full contexts.

CHATBOT PROGRAM :

```
Python program to create a chatbot in python
#importing necessary libraries
import numpy as np
import pandas as pd
import os
for dirname, _, filenames in os.walk('/kaggle/input'):
for filename in filenames:
print(os.path.join(dirname, filename))
df = pd.read_csv('/kaggle/input/stacksample/Tags.csv')
for i in range(0, len(df)):
if df['Tag'][i] == 'python':
tag.append(df['Id'][i])
print(len(tag))
df = pd.read_csv('/kaggle/input/stacksample/Questions.csv', encoding = 'ISO-8859-1')
questions = []
answers = []
print(len(df['Id'].unique()))
for i in range(0, 100000):
if df['Id'][i] in tag:
questions.append(df['OwnerUserId'][i])
questions.append(df['Body'][i])
print(len(questions))
```

Start building the chatbot model by loading and preprocessing the dataset:

Dataset link :
https://www.kaggle.com/datasets/grafstor/simple -dialogs-for-chatbot

Exploratory Data Analysis:

This includes checking for missing values, exploring the data statistics, and visualizing it.

Data Processing:

Data preprocessing is the process of transforming raw data into an understandable format.

It is also an important step in data mining as we cannot work with raw data.

The quality of the data should be checked before applying machine learning or data mining algorithms.

• Steps in data preprocessing:

➢Data Cleaning

➢Data Integration

➢Data Transformation

➢Data Reduction

➢Data Discretization

➢Data Normalization

Data Cleaning:

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset. When combining multiple data sources, there are many opportunities for data to be duplicated or mislabeled.

Python program for data cleaning using the pandas library:

```
Import pandas as od

 # load the dataset

 df=pd.read_csv('name_dataset.csv')

 # handling missing values

 df.dropna()

 df.fillna(value)

 # Removing the duplicates

 df.drop_duplicate()
```

```
# Correcting the inconsistent data

df[‘column_name’].replace(old value,new value,inplace=true
```

Data Integration:

Data integration refers to the process of bringing together data from multiple sources acrossan organization to provide a complete, accurate, and up-to-date dataset for BI, data analysis and other applications and business processes.

Data Transformation:

Data transformation is the process of converting data from one format to another, typically from the format of a source system into the required format of a destination system.

Data transformation is a component of most data integration and data management tasks, such as data wrangling and data warehousing.

Data reduction:

Data reduction is a capacity optimization technique in which data is reduced to its simplest possible form to free up capacity on a storage device.

There are many ways to reduce data, but the idea is very simple—squeeze as much data into physical storage as possible to maximize capacity.

Data discretization:

Data discretization is defined as a process of converting continuous data attribute values into a finite set of intervals and associating with each interval some specific data value

Data normalization:

Normalization is the process of organizing data in a database.

It includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.

START BUILDING THE PROJECT BY PERFORMING DIFFERENT ACTIVITIES LIKE FEATURE ENGINEERING, MODEL TRAINING, EVALUATION ETC AS PER THE INSTRUCTIONS IN THE PROJECT:

Feature engineering:

Feature engineering is the process that takes raw data and transforms it into features that can be used to create a predictive model using machine learning or statistical modeling, such as deep learning. it is the process of selecting, extracting, and transforming the most relevant features from the available data to build more accurate and efficient machine learning model.

Feature creation:

Feature creation, sometimes just called feature engineering, is the process of training a machine learning model by using existing data to construct new feature.

Feature extraction:

It involves extracting and creating new variables automatically from raw data. The goal of feature extraction is to reduce data volume to a more manageable modeling set automatically. Some feature extraction techniques include cluster analysis, edge detection algorithms, principal components analysis, and text analytics. Feature extraction is used when predictive modeling algorithms cannot directly model observations because they are too voluminous in their raw state. For example, audio, image, tabular, and textual data may have millions of attributes algorithms may not be effective for this unstructured data, unsupervised learning can be very useful. Feature extraction creates features from the existing ones and then discards the original features to reduce the total number of features in a dataset. The new, reduced set of features can summarize the original set of features and the information they contain. The newer, smaller dataset can much more easily be modeled.

Model Training:

Artificial intelligence (AI) training is the process of teaching an AI system to perceive, interpret and learn from data. That way, the AI will later be capable of inferencing— making decisions based on information it's provided. This type of training requires 3 important components: a well-designed AI model; large amounts of high-quality and accurately annotated data; and a powerful computing platform. Properly trained, an AI's potential is nearly limitless. For example, AI models can help anticipate our wants and needs, autonomously navigate big cities, and produce scientific breakthroughs. It's already happening. You experience the power of well-trained AI when you use Netflix's recommendation engine to help decide which TV show or movie you want to watch next. Or you can ride with AI in downtown Phoenix, Ariz. It's home to the robotaxi service operated by Waymo, the autonomous-vehicle developer owned by Google's parent company, Alphabet.

Future of AI Training:

New AI training theories are coming online quickly. As the market heats up and AI continues to find its way out of the laboratory and onto our computing devices, Big Tech is working feverishly to make the most of the latest gold rush. One new AI training technique coming to prominence is known as Reinforcement Learning (RL). Rather than teaching an AI model using a static dataset, RL trains the AI as though it were a puppy, rewarding the system for a job well done. Instead of offering doggie treats, however, RL gives the AI a line of code known as a "reward function." This is a dynamic and powerful training method that some AI experts believe will lead to scientific breakthroughs. Advances in AI training, high-performance computing and data science will continue to make our sci-fi dreams a reality. For example, one AI can now teach other AI models. One day, this could make AI training just another autonomous process. Will the next era of AI bring about the altruism of Star Trek or the evil of The Matrix? One thing's likely: We won't have to wait long to find out.

Evaluation:

Evaluation is the method of understanding the reliability of an API Evaluation and is based on the outputs which are received by feeding the data into the model and comparing the output with the actual answers. In the 1960s, AI research gained momentum with the development of the first AI programming language, LISP, by John McCarthy. Early AI systems focused on symbolic reasoning and rule-based systems, which led to the developmentof expert systems in the 1970s and 1980s. Evaluation provides a systematic method to study a program, practice, intervention, or initiative to understand how well it achieves its goals. Evaluations help determine what works well and what could be improved in a program or initiative. Evaluation is a process that critically examines a program. It involves collecting and analyzing information about a program's activities, characteristics, and outcomes. Its purpose is to make judgments about a program, to improve its effectiveness, and/or to inform programming decisions (Patton, 1987).

CONCLUSION:

A chatbot is one of the simple ways to transport data from a computer without having to think for proper keywords to look up in a search or browse several web pages to collect information; users can easily type their query in natural language and retrieve information.

THANKYOU