# Retrieval Learning Bot (PLUTO) Documentation

## Overview

The provided code implements a simple retrieval-based conversational agent named PLUTO (Retrieval Learning Bot). PLUTO uses natural language processing (NLP) techniques to understand and respond to user input. The main components of PLUTO include text preprocessing, greeting detection, and response generation using TF-IDF (Term Frequency-Inverse Document Frequency) and cosine similarity.

## Code Sections

#### **1. Reading Data**

f=open('/content/My Drive/Colab/data.txt','r',errors='ignore')

raw\_doc=f.read()

Reads the content from the file data.txt and stores it in raw\_doc.

#### **2. Text Preprocessing**

raw\_doc=raw\_doc.lower()

sentence\_tokens = nltk.sent\_tokenize(raw\_doc)

word\_tokens = nltk.word\_tokenize(raw\_doc)

Converts the text to lowercase and tokenizes it into sentences and words using NLTK.

#### **3. Lemmatization**

lemmer = nltk.stem.WordNetLemmatizer()

def LemTokens(tokens):

return [lemmer.lemmatize(token) for token in tokens]

remove\_punc\_dict = dict((ord(punct),None)for punct in string.punctuation)

def LemNormalize(text):

return LemTokens(nltk.word\_tokenize(text.lower().translate(remove\_punc\_dict)))

Defines functions for lemmatization of tokens and normalization of text by removing punctuation.

#### **4. Greeting Detection**

greet\_inputs = ('hello','hi','how are you?','hey',)

greet\_responses = ('hi','hey!','good to see you again','hey there!')

def greet(sentence):

for word in sentence.split():

if word.lower() in greet\_inputs:

return random.choice(greet\_responses)

Checks if the user's input contains a greeting, and responds accordingly.

#### **5. TF-IDF and Cosine Similarity**

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

TfidfVec = TfidfVectorizer(tokenizer=LemNormalize, stop\_words='english')

tfidf = TfidfVec.fit\_transform(sentence\_tokens)

def response(user\_response):

global robo1\_response

robo1\_response = ""

# ... (cosine similarity calculation and response generation)

return robo1\_responseUtilizes TF-IDF and cosine similarity to generate responses based on user input.

#### **6. Main Conversation Loop**

flag = True

print('Hello! I am the Retrieval Learning Bot PLUTO. Start typing your questions after greeting to talk to me. For ending convo type bye!')

while(flag == True):

user\_response = input()

user\_response = user\_response.lower()

# ... (processing user input, detecting greetings, and generating responses)Implements the main conversation loop where PLUTO interacts with the user, detects greetings, and generates responses until the user decides to end the conversation by typing 'bye'.