

T.C.
ERCİYES ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

Rüya Yorumlayabilen Yapay Zeka
(24221601)

Hazırlayan
Duygu Gözde KAYABAŞI
1030510338

Danışman
Doç.Dr. Özkan Ufuk NALBANTOĞLU

Bilgisayar Mühendisliği
Design Project
Final Raporu

Haziran
2024
KAYSERİ

Rüya Yorumlayabilen Yapay Zeka

Özet

Kullanıcı ile iletişimini text üzerinde olan , aldığı text verisini alıp değerlendirip yorumunu yapan elde ettiği sonucu kullanıcıya yine text yoluyla iletten bunu yaparken Türkçenin dil bilgisi yapısını göz önünde bulundurup mecaz anlamları iyi algılayıp vereceği cevaplara yön kazandıracak şekilde rüya yorumlaması yapacak bir yapay zeka tasarımlı , kodlaması

Anahtar kelimeler: kullanıcı, text, ifade, algılamak, anlamlandırmak, iletmek, rüya, rüya yorumlama, yorumlama , yapay zeka, kod, tasarım...

1. Rüya: Sistemin ana amacıdır.
2. Yorum: Yorumlar genellikle bir durumu, bir metni, bir olayı veya bir gözlemi daha derinlemesine anlamaya veya açıklamaya yönelikdir. Sistemin belirli çerçevelerle verisi işlemesidir.
3. İfade: Kullanıcının rüyasını ifade etmesi için kullanılır.
4. Algılamak: Sistem, kullanıcının rüya içeriğini anlamak için çeşitli yaklaşımalar kullanır.
5. Sembole: Kullanıcı, rüyasında gördüğü semboller hakkında bilgi verir.
6. Anlamlandırmak: Kullanıcının rüyasını yorumlamak için gereklidir. Sistem, kullanıcının rüya içeriğini anlamlandırmak için bu kelimeyi arar.
7. İletişim: Sistem tarafından yapılan yorumun kullanıcıya iletilmesi için kullanılır.
8. Kodlama: Sistemin geliştirilmesi için gereklidir. Bu kelime, yapay zekanın geliştirilmesi ve kodlanması sürecini ifade eder.
9. Dil bilgisi:Sistem, dil bilgisini kullanarak kullanıcının rüyasını daha iyi anlamak için çaba gösterir.
10. Tasarım: Yapay zeka sisteminin tasarım sürecini ifade eder. Bu kelime, sistemin nasıl yapılandırılacağını ve işleyeceğini belirtir.

Rüya Yorumlayabilen Yapay Zeka

AraYüz tasarımı

 Home Page

 Daydream



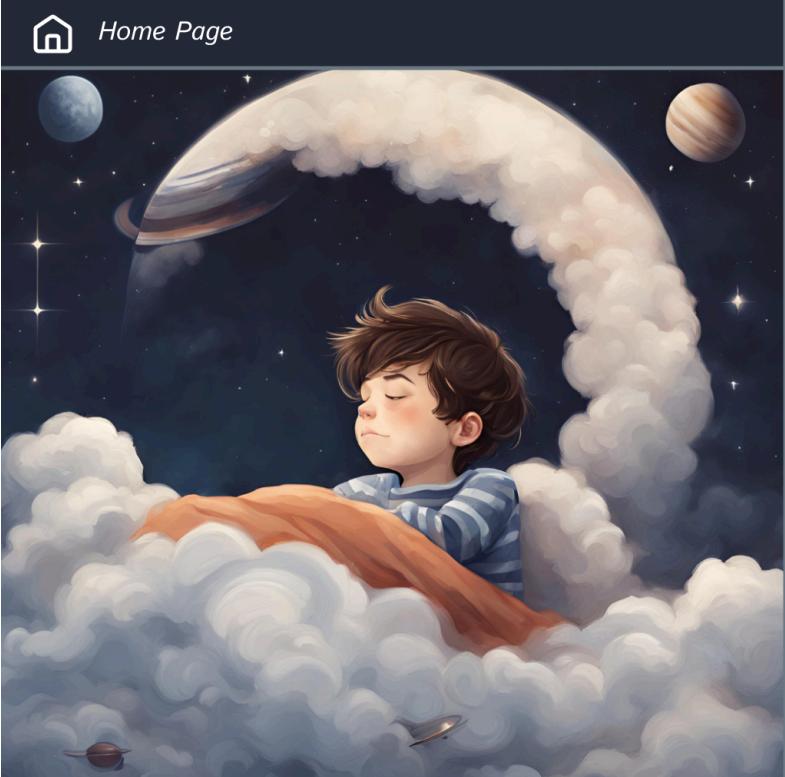
Rüyani anlatmak ister misin ?

"Dreamers" adlı bu uygulama rüyanızı dinleyip yorum yapan yapay zeka destekli bir uygulamadır. Umarım aradığınız cevabı bulabilirsiniz.

[Giriş Yap](#)

 Home Page

 Midnight



Rüyani anlatmak ister misin ?

"Dreamers" adlı bu uygulama rüyanızı dinleyip yorum yapan yapay zeka destekli bir uygulamadır. Umarım aradığınız cevabı bulabilirsiniz.

[Giriş Yap](#)



Home Page



Daydream

Rüyayı anlatmak ister misin ?

"Dreamers" adlı bu uygulama rüyanızı dinleyip yorum yapan yapay zeka destekli bir uygulamadır. Umarım aradığınız cevabı bulabilirsiniz.

Giriş Yap



Home Page

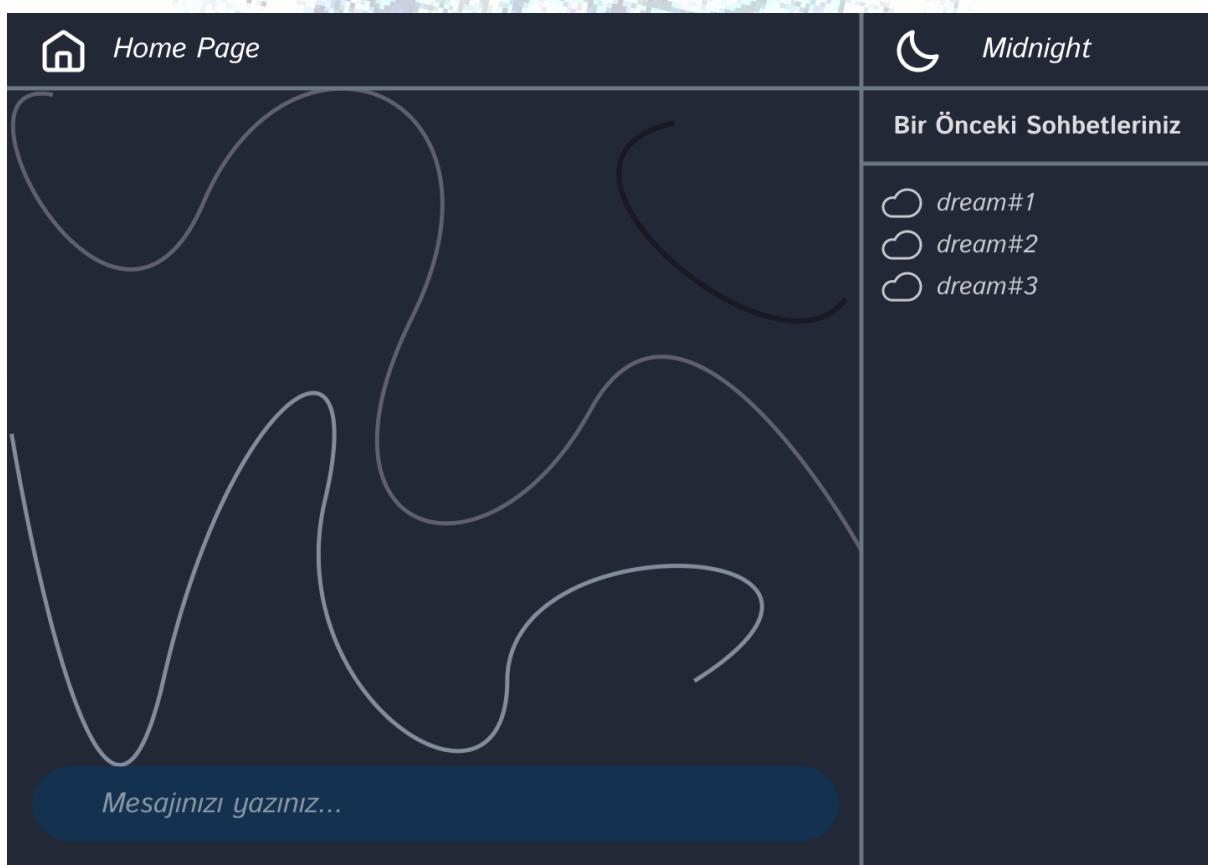


Midnight

Rüyayı anlatmak ister misin ?

"Dreamers" adlı bu uygulama rüyanızı dinleyip yorum yapan yapay zeka destekli bir uygulamadır. Umarım aradığınız cevabı bulabilirsiniz.

Giriş Yap



Not: [Figma](#) buradaki linkten tasarımın hepsine ulaşabilirsiniz.

GİRİŞ

Çalışmanın tanıtılması ve detaylı olarak anlatılması

Projenin hedefi (ACHIEVEMENTS)

-Bu çalışmadan gerçekleştirmesini beklediğimiz fonksiyonların bir listesi :

- **Doğru Data Seti** seçilerek eğitimli bir model oluşturulması,
- **Text Sınırı** karakter sınırlaması getirilerek modelin daha iyi performans vermesi amaçlanması,
- **Olayları Yorumlanabilmesi için Text'in parçalanmasıyla** somut öğelerin rüyalardaki anlamlarına bakılması
- **Simge ve Sembollerin** anlamlarının birbirleri ile bağdaştırıp olayların daha iyi yorumlanması
- **Olumlu - Olumsuz** bir ögenin olup olmadığını araştırılması,
- **Rüya çeşitlerini** tanııp bilip kullanıcıya bu konuda geri bildirim verilmesi,
- **En net hatırladığı kısmın** öğrenilip yorum yaparken göz önünde bulundurması,
- **Net olmayan kısımlar** da aynı işlemden geçmesi,
- **Önceki konuşmaların hatırlanması beklenmemelidir**
- **Cevap rastgeleleştirme** konusunda belirli bir key atanılarak bir önceki cevaplarla karşılaştırılıp farklılık oluşturulması,**loss function** bu konuda yardımcı olacaktır.
- **Cevap Sınırı çıktıya bağlı olarak eklenip eklenmeyeceği tartışma** konusudur.

Yukarıdaki yapılması planlanan hedeflerin proje sonrası son durumları aşağıda belirtilmiştir.

1. **Doğru Data Seti** sadece İngilizce olarak bulunmuştur. Arka tarafta kısa bir işlemle kullanıcının girdisi İngilizceye çevrilmektedir. Çünkü sentiment hesaplarken kullandığımız lib ‘TextBlob’ İngilizce kelimelerle çalışmaktadır.
2. Text Sınırı konulmamıştır.
3. **Olayları Yorumlanabilmesi için Text'in parçalanması** işlemi gerçekleştirılmıştır. Somut öğelerin rüyalardaki anlamlarına bakılması kısmında sorun vardır.
4. **Simge ve Sembollerin** anlamlarının birbirleri ile bağdaştırıp olayların daha iyi yorumlanması gerçekleştirilemedi.
5. **Olumlu - Olumsuz** bir ögenin olup olmadığını araştırılması yapıldı ve rüyanın, rüya gören kişiye bıraktığı hissiyattan bahsedildi.
6. **Rüya çeşitlerini** tanııp bilip kullanıcıya bu konuda geri bildirim verilmesi (gerçekleştirilemedi)
7. **En net hatırladığı kısmın** öğrenilip yorum yaparken göz önünde bulundurması gerçekleştirilemedi,
8. **Net olmayan kısımlar** da aynı işlemden geçmesi gerçekleştirilemedi ,
9. **Önceki konuşmaların hatırlanması beklenmemelidir**

10. Cevap rastgeleleştirme konusunda belirli bir key atanılarak bir önceki cevaplarla karşılaştırılıp farklılık oluşturulması,**loss function bu konuda yardımcı olacaktır.**
Bunun yerine rastgelelik zamana bağlı olarak farklı cevaplar verecek şekilde değişikliğe gidildi.

11. Cevap Sınırı çıktıya bağlı olaraktan eklenip eklenmeyeceği tartışma konusudur.
Bunun yerine ChatBot'un kapatılabilmesi için küçük bir kod yazıldı.

Piyasa çalışması

Buna benzer çalışmalar aşağıda yer almaktadır.

- [Dream Interpreter AI](#)
- [AI Dream Interpreter](#)
- [Dream Interpreter AIM: Uncover Hidden Insights in Your Dreams with AI | Deepgram](#)

Müşteri kitlesi

OpenAI, Chat GPT'yi Kasım 2022'de kullanıma sundu ve Ocak 2023 itibarıyla 100 milyondan fazla kullanıcıya ulaştığı bildirildi. Bu geniş kullanıcı tabanı, ChatGPT'nin dil modeli alanında önemli bir yenilik olduğunu göstermektedir.

Yapay zekanın başarısı: Microsoft Bing arama motoru, günlük 100 milyon aktif kullanıcıyı aştı (Mar 9, 2023)

TARTIŞMA, SONUÇ ve ÖNERİLER

Tartışma, Sonuç ve Öneriler

Kodlama süreci boyunca çıktılara bağlı olaraktan şu anda planan süreçten belki bağlı kalınarak belki de dışına çıkılarak yeni yöntemler , yeni algoritmalar, yeni yaklaşımalar denenebilir ,daha yüksek performanslı yöntemler geliştirilebilir.

Gelecekteki Çalışmalar

- Beklenilen sürede yapılamayan eksik kalınan yönlerin tamamlanması,
- Kullanıcıya sunulduktan sonra gözden kaçan sorunların, hataların iyileştirilmesi,
- Kullanıcıya sunulduktan sonra kullanıcıya bağlı olarak eklenmesi istenilen yeni daha önce düşünülmemiş fonksiyonlar
- Bakımları

Arayüz Ana Ekran için :

```
from pathlib import Path

# from tkinter import *

# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

OUTPUT_PATH = Path(__file__).parent
ASSETS_PATH = OUTPUT_PATH /
Path(r"D:\lecturers\odev\build\assets\frame0")

def relative_to_assets(path: str) -> Path:
    return ASSETS_PATH / Path(path)

window = Tk()

window.geometry("1440x1024")
window.configure(bg = "#FFFFFF")

canvas = Canvas(
    window,
```

```
        bg = "#FFFFFF",
        height = 1024,
        width = 1440,
        bd = 0,
        highlightthickness = 0,
        relief = "ridge"

)

canvas.place(x = 0, y = 0)

canvas.create_rectangle(
    0.0,
    0.0,
    1441.0,
    1024.0,
    fill="#A1B3C2",
    outline="")

button_image_1 = PhotoImage(
    file=relative_to_assets("button_1.png"))

button_1 = Button(
    image=button_image_1,
    borderwidth=0,
    highlightthickness=0,
    command=lambda: print("button_1 clicked"),
    relief="flat"
```

```
)  
  
button_1.place(  
  
    x=0.0,  
  
    y=0.0,  
  
    width=419.0,  
  
    height=90.0  
  
)  
  
  
  
canvas.create_text(  
  
    1044.0,  
  
    307.0,  
  
    anchor="nw",  
  
    text="Rüyani anlatmak ister misin?",  
  
    fill="#FFFFFF",  
  
    font=("IstokWeb Bold", 34 * -1)  
  
)  
  
  
  
canvas.create_text(  
  
    1044.0,  
  
    415.0,  
  
    anchor="nw",  
  
    text="“Dreamers” adlı bu uygulama rüyanızı dinleyip yorum yapan  
yapay zeka destekli bir uygulamadır. Umarım aradığınız cevabı  
bulabilirsiniz.",  
  
    fill="#FFFFFF",  
  
    font=("IstokWeb BoldItalic", 34 * -1)
```

```
)  
  
    canvas.create_rectangle(  
        1016.0,  
        -3.0,  
        1172.0,  
        282.0,  
        fill="#5E697D",  
        outline="")  
  
  
    canvas.create_rectangle(  
        992.0,  
        19.0,  
        1346.2433776855469,  
        1068.991455078125,  
        fill="#747685",  
        outline="")  
  
  
    canvas.create_rectangle(  
        1218.0,  
        93.0,  
        1505.7860412597656,  
        1021.2744140625,  
        fill="#252839",  
        outline="")
```

```
button_image_2 = PhotoImage(  
    file=relative_to_assets("button_2.png"))  
  
button_2 = Button(  
    image=button_image_2,  
    borderwidth=0,  
    highlightthickness=0,  
    command=lambda: print("button_2 clicked"),  
    relief="flat"  
)  
  
button_2.place(  
    x=1047.0,  
    y=887.0,  
    width=379.0,  
    height=90.0  
)  
  
  
button_image_3 = PhotoImage(  
    file=relative_to_assets("button_3.png"))  
  
button_3 = Button(  
    image=button_image_3,  
    borderwidth=0,  
    highlightthickness=0,  
    command=lambda: print("button_3 clicked"),  
    relief="flat"
```

```
)  
  
button_3.place(  
    x=1024.0,  
    y=0.0,  
    width=419.0,  
    height=90.0  
)  
  
  
  
image_image_1 = PhotoImage(  
    file=relative_to_assets("image_1.png"))  
  
image_1 = canvas.create_image(  
    563.0,  
    561.0,  
    image=image_image_1  
)  
  
  
  
canvas.create_rectangle(  
    -5.0,  
    85.0,  
    1444.0,  
    90.0,  
    fill="#697783",  
    outline="")  
  
  
  
canvas.create_rectangle(  
    -5.0,
```

```
    1016.0,  
    -5.0,  
    1021.0,  
    1024.0,  
    fill="#697783",  
    outline="")  
  
window.resizable(False, False)  
  
window.mainloop()
```

1. Imports:

- `from pathlib import Path`: Path nesnesini kullanarak dosya yollarını işlemek için.
- `from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage`: Tkinter kütüphanesinden bazı bileşenleri içe aktarır. Bazılarına yönelik doğrudan adlandırma (`from tkinter import *`) yapılmamış, bu şekilde yaparak kodun daha okunabilir olmasını sağlar.

2. Constant Definitions:

- `OUTPUT_PATH`: Dosyanın bulunduğu dizini içeren bir `Path` nesnesi tanımlar.
- `ASSETS_PATH`: GUI'deki görsel varlıkların (resimler) bulunduğu dizini tanımlar.
- `relative_to_assets()`: Görsel varlıkların yollarını bulmak için kullanılan bir yardımcı işlev. Görsel varlık yollarını `ASSETS_PATH`'e göre oluşturur.

3. Tkinter Window Oluşturma:

- `Tk()` ile bir Tkinter penceresi oluşturulur.
- `window.geometry("1440x1024")`: Pencerenin boyutunu ayarlar.
- `window.configure(bg="#FFFFFF")`: Pencere arka plan rengini beyaza ayarlar.

4. Canvas Oluşturma:

- `Canvas` bileşenini oluşturur. Canvas, diğer bileşenleri (resimler, metinler, düğmeler vb.) içine yerleştirmek için kullanılır.
- Canvas'ın özelliklerini (boyut, arka plan rengi vb.) ayarlar.
- `place()` metoduyla pencere içindeki konumunu belirler.

5. Resim, Metin ve Düğme Ekleme:

- `PhotoImage` ile görsel varlıkları yükler.

- **Button** nesnelerini oluşturur ve özelliklerini ayarlar. Her bir düğme, tıklama olayını bir fonksiyona bağlar.
- **Canvas** üzerine metin ekler.
- **create_rectangle()** ile Canvas üzerine dikdörtgenler ekler.

6. Pencereyi Açma:

- **window.mainloop()**: Tkinter penceresini açar ve etkinleştirir.

Bu kod, bir pencere oluşturarak içinde çeşitli görsel varlıklar, metinler ve düğmeler bulunduran basit bir GUI uygulamasıdır.

Arayüz Mesajlaşma kısmı için:

```
from pathlib import Path

# from tkinter import *
#
# Explicit imports to satisfy Flake8

from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage

OUTPUT_PATH = Path(__file__).parent
ASSETS_PATH = OUTPUT_PATH /
Path(r"D:\lecturers\odev\build\assets\frame5")

def relative_to_assets(path: str) -> Path:
    return ASSETS_PATH / Path(path)

window = Tk()
```

```
window.geometry("1440x1024")

window.configure(bg = "#252839")

canvas = Canvas(
    window,
    bg = "#252839",
    height = 1024,
    width = 1440,
    bd = 0,
    highlightthickness = 0,
    relief = "ridge"
)

canvas.place(x = 0, y = 0)

canvas.create_rectangle(
    0.0,
    0.0,
    1441.0,
    1024.0,
    fill="#252839",
    outline="")
    canvas.create_rectangle(
```

```
1.1471894042820168,  
131.35895805352345,  
1004.935247949189,  
795.0710898150719,  
fill="#FFFFFF",  
outline="")
```

```
canvas.create_rectangle(724.5333828814519,  
                      124.99734182105055,  
                      1002.65283203125,  
                      368.26995078724667,  
                      fill="#1C1A25",  
                      outline="")
```

```
canvas.create_rectangle(0.2757432019229782,  
                      83.85853491446392,  
                      1023.217529296875,  
                      642.765803940667,  
                      fill="#747685",  
                      outline="")
```

```
canvas.create_rectangle(  
    -0.9888310011074282,
```



```
    image=button_image_2,  
    borderwidth=0,  
    highlightthickness=0,  
    command=lambda: print("button_2 clicked"),  
    relief="flat"  
)  
  
button_2.place(  
    x=1021.0,  
    y=0.0,  
    width=419.0,  
    height=90.0  
)  
  
  
  
canvas.create_rectangle(  
    -5.0,  
    85.0,  
    1444.0,  
    90.0,  
    fill="#697783",  
    outline="")  
  
  
  
canvas.create_rectangle(  
    1016.0,  
    -5.0,  
    1021.0,
```

```
1024.0,  
fill="#697783",  
outline="")  
  
canvas.create_rectangle(  
30.0,  
900.0,  
994.0,  
990.0,  
fill="#000000",  
outline="")  
  
canvas.create_rectangle(  
1016.0,  
175.0,  
1444.0,  
180.0,  
fill="#697783",  
outline="")  
  
canvas.create_text(  
1059.0,  
114.0,  
anchor="nw",  
text="Bir Önceki Sohbetleriniz ",
```

```
        fill="#FFFFFF",
        font=("IstokWeb Bold", 30 * -1)

    )

canvas.create_text(
    1111.0,
    202.0,
    anchor="nw",
    text="dream#1\ndream#2\ndream#3",
    fill="#FFFFFF",
    font=("IstokWeb Italic", 30 * -1)
)

image_image_1 = PhotoImage(
    file=relative_to_assets("image_1.png"))

image_1 = canvas.create_image(
    1069.0,
    227.0,
    image=image_image_1
)

image_image_2 = PhotoImage(
    file=relative_to_assets("image_2.png"))

image_2 = canvas.create_image(
    1069.0,
```

```
    277.0,  
  
    image=image_image_2  
  
)  
  
  
image_image_3 = PhotoImage(  
  
    file=relative_to_assets("image_3.png"))  
  
image_3 = canvas.create_image(  
  
    1069.0,  
  
    327.0,  
  
    image=image_image_3  
  
)  
  
  
canvas.create_rectangle(  
  
    30.0,  
  
    131.0,  
  
    994.0,  
  
    886.0,  
  
    fill="#FFFFFF",  
  
    outline="")  
  
window.resizable(False, False)  
  
window.mainloop()
```

1. Imports:

- **from pathlib import Path**: Yine dosya yollarını işlemek için **Path** sınıfını içe aktarıyoruz.

- `from tkinter import Tk, Canvas, Entry, Text, Button, PhotoImage`: Tkinter bileşenlerini içe aktarıyoruz.
 2. **Constant Definitions:**
- `OUTPUT_PATH` ve `ASSETS_PATH`: İlk örnekte olduğu gibi dosya ve görsel varlık yollarını tanımlıyoruz.
 3. **Tkinter Window Oluşturma:**
- Pencere boyutu ve arka plan rengi ayarlanıyor.
 4. **Canvas Oluşturma:**
- İlk örnekte olduğu gibi bir Canvas bileşeni oluşturuluyor. Ancak bu sefer arka plan rengi ve bazı dikdörtgenlerin boyutları ve konumları farklı.
 5. **Dikdörtgenler ve Metinler:**
- Canvas üzerinde çeşitli dikdörtgenler ve metinler oluşturuluyor. Bu, arayüzün farklı bölümlerini ve metin içeriğini temsil eder.
- 6. **Düğmeler ve Resimler:**
- İlk örnekte olduğu gibi düğmeler ve resimler ekleniyor. Ancak bu sefer resimlerin ve düğmelerin konumları ve özellikleri farklı.
- 7. **Pencereyi Açıma:**
- `window.mainloop()` metodu ile pencereyi görüntülüyoruz.

Sentiments Analysis için:

```
from textblob import TextBlob

import numpy

import pandas


nightmares = pandas.read_csv("nightmares.csv")

dream = list(nightmares['answer_text'])

e = '\n'.join(map(str, dream))

#print(e)

dreams = e.splitlines()

for row in dreams:

    blob = TextBlob(str(row.strip()))

    sentiment = blob.sentiment.polarity

    print(sentiment)
```

1. Libraries Imports:

`from textblob import TextBlob`: TextBlob kütüphanesinden `TextBlob` sınıfını içe aktarır. TextBlob, metin işleme ve duyarlılık analizi için kullanılır.

`import numpy`: Numpy, bilimsel hesaplama için kullanılan bir Python kütüphanesidir.

`import pandas`: Pandas, veri manipülasyonu ve analizi için kullanılan bir Python kütüphanesidir.

2. Veri Yüklemesi:

`nightmares = pandas.read_csv("nightmares.csv")`: Bir CSV dosyasını okur ve `nightmares` adında bir pandas DataFrame'e yükler. Bu DataFrame'in 'answer_text' sütunu, metin verilerini içerir.

3. Metin Duyarlılık Analizi:

`dream = list(nightmares['answer_text'])`: 'answer_text' sütunundaki metin verilerini bir liste olarak alır.

`e = '\n'.join(map(str, dream))`: Liste elemanlarını birleştirerek tek bir metin dizesi oluşturur. Her bir metin ögesi arasında bir satır atlama karakteri (`\n`) ekler.

`dreams = e.splitlines()`: Birleştirilmiş metin dizesini satırlara böler ve `dreams` adında bir liste oluşturur.

Her bir satır için:

- a. `blob = TextBlob(str(row.strip()))`: Her satırı alır, boşlukları siler ve `TextBlob` nesnesine dönüştürür. `TextBlob` nesnesi, metnin duyarlılık analizini yapmamızı sağlar.
- b. `sentiment = blob.sentiment.polarity`: `TextBlob` nesnesinin `sentiment` özelliğini kullanarak metnin duyarlılık (olumlu/negatif) puanını alır.
- c. `print(sentiment)`: Her metnin duyarlılık puanını ekranaya yazdırır.

Kullanıcının anlattığı rüyayı İngilizceye çeviren kod :

```
import tkinter as tk
from translate import Translator
```

```

def on_enter(event):
    # Check if the pressed key is Enter
    if event.keysym == "Return":
        translate_to_english()

def translate_to_english():
    # Get the input text
    input_text = text_input.get("1.0", "end-1c")

    # Translate the text to English
    translator = Translator(to_lang="en")
    english_translation = translator.translate(input_text)

    # Display the translated text in the English text box
    english_text.delete("1.0", tk.END)
    english_text.insert(tk.END, english_translation)

# Create the Tkinter window
root = tk.Tk()
root.title("Text Translator")

# Input text box
text_input = tk.Text(root, height=10, width=50)
text_input.pack()

# English text box for translated text
english_text = tk.Text(root, height=10, width=50)
english_text.pack()

# Bind the Enter key press event to the on_enter function
text_input.bind("<KeyPress>", on_enter)

# Run the Tkinter event loop
root.mainloop()

```

Kütüphanelerin İçe Aktarılması:

- `import tkinter as tk`: Tkinter kütüphanesini `tk` kısaltmasıyla içe aktarır.
- `from translate import Translator`: Çeviri işlemi için `translate` kütüphanesinden `Translator` sınıfını içe aktarır.

Çeviri Fonksiyonu (`translate_to_english`):

- `translate_to_english` fonksiyonu, kullanıcının girdiği metni İngilizce'ye çevirir.
- Metin girişini `text_input` adlı metin kutusundan alır.
- `Translator` sınıfını kullanarak metni İngilizce'ye çevirir.
- İngilizce çeviriyi, `english_text` adlı metin kutusuna yazdırır.

Tkinter Penceresi Oluşturma:

- `root = tk.Tk()`: Tkinter penceresi oluşturur ve `root` değişkenine atar.
- `root.title("Text Translator")`: Pencerenin başlığını ayarlar.

Metin Kutuları Oluşturma:

- `text_input` ve `english_text` adında iki metin kutusu oluşturur. İlk metin kutusu kullanıcidan metin girişi alırken, ikincisi çevrilen metni gösterir.

Olayları İşleme:

- `text_input.bind("<KeyPress>", on_enter)`: Kullanıcının bir tuşa basmasını dinler ve `<KeyPress>` olayını `on_enter` fonksiyonuna bağlar. Bu sayede, kullanıcı Enter tuşuna bastığında metni çevirmek için `translate_to_english` fonksiyonunu çağırır.

Elde edilen İngilizce rüya textinin duyu analizi

```
from textblob import TextBlob  
  
import numpy  
  
  
blob = TextBlob(english_translate)  
  
sentiment = blob.sentiment.polarity  
  
print(sentiment)
```

Chatbot için kullanılacak Json:

```
import json

json_string = {"intents": [
    {
        "tag": "selamlaşma",
        "patterns": ["Merhaba", "Selam", "Günaydın", "Tünaydın", "İyi akşamlar", "İyi geceler", "N'aber?", "Ne var ne yok?", "Selamünaleyküm", "Hey!", "İyi günler"],
        "responses": ["Hey!", "Senin için ne yapabilirim?", "Nasilsınız?", "Merhaba, nasilsınız?", "Selam, uzun zamandır görüşemedik!", "Merhaba", "Selam", "N'aber?", "Ne var ne yok?", "İyi günler"],
    },
    {
        "tag": "ruyani anlatma",
        "patterns": [ "", "", "", "", "", ""],
        "responses": ["Dün gece ilginç bir rüya gördün mü?", "Rüyaların çok merak ediyorum, dün gece ne rüya gördün?", "Bu sabah nasıl bir rüyayla uyandın?", "Dün gece rüya gördüsen, anlatmak ister misin?", "Rüyalarını paylaşmayı sever misin? Dün gece ne gördün?", "Rüyaların genellikle nasıl olur? Dün geceki rüyanı nasıldır?", "Rüyani anlatırsan çok sevinirim, neler gördün?", "Bazen rüyalar çok ilginç olabiliyor, senin dün geceki rüyanı nasıldı?", "Rüyalar bazen önemli ipuçları verir, dün gece ne rüya gördün?", "Dün gece ilginç bir şey yaşadın mı rüyanda?"],
    },
    {
        "tag": "olaylar",
        "patterns": [ "", "", "", "", "", ""],
        "responses": ["Rüyanda neler oldu, biraz daha detaylı anlatabilir misin?", "Rüyanın hangi kısmı seni en çok etkiledi?", "Rüyandaki yer nasıldı, etrafında neler vardı?", "Rüyanda kimler vardı, onları nasıl tanımlarsın?", "Rüyanın başından sonuna kadar neler yaşadı?", "Rüyanda hissettiklerini anlatabilir misin?", "Rüyandaki mekanları biraz daha"]
    }
]}  
print(json.dumps(json_string, indent=4))
```

tarif edebilir misin?", "Rüyanda geçen olaylar nasıl başladı ve nasıl gelişti?", "Rüyanda karşılaşışının kişilerle ne tür etkileşimlerde bulundun?", "Rüyanda geçen olayların sırasını hatırlıyor musun?", "Rüyanda en çok dikkatini çeken ayrıntılar nelerdi?", "Rüyandaki olaylar seni nasıl hissettirdi?", "Rüyandaki en ilginç veya tuhaf detay neydi?", "Rüyanın sonunda neler oldu, nasıl bitti?"],
},

{"tag" : "olumlu",

"patterns" : [],

"responses": ["Rüyan seni çok mutlu etmiş gibi görünüyor.", "Rüyanın seni ne kadar iyi hissettiğini görebiliyorum.", "Bu rüya seni gerçekten rahatlatmış gibi görünüyor.", "Rüyanın ardından kendini daha huzurlu hissettiğini söyleyebilirim.", "Rüyanın sende olumlu bir etki bıraktığını anlıyorum.", "Bu rüya sana yeni bir bakış açısı kazandırmış gibi duruyor.", "Rüyanın ardından daha enerjik ve motive görünüyorsun.", "Rüyanın seni ne kadar motive ettiğini görmek güzel.", "Bu rüya sana gerçekten ilham vermiş gibi.", "Rüyanın seni nasıl olumlu etkilediğini duymak çok güzel.", "Rüyanın ardından kendini daha iyi hissettiğini söylemek doğru olur mu?", "Bu rüya sana umut ve mutluluk getirmiş gibi görünüyor.", "Rüyanın sana ne kadar iyi geldiğini görmek harika.", "Rüyanın ardından pozitif bir değişim fark ettim sende."],
},

{"tag" : "olumsuz",

"patterns" : [],

"responses": ["Rüyan seni biraz rahatsız etmiş gibi görünüyor.", "Rüyanın seni ne kadar huzursuz ettiğini görebiliyorum.", "Rüyanın sende olumsuz bir etki bıraktığını fark ettim.", "Bu rüya seni gerçekten tedirgin etmiş gibi görünüyor.", "Rüyanın ardından kendini daha gergin hissettiğini söyleyebilirim.", "Rüyanın seni ne kadar rahatsız ettiğini anlıyorum.", "Bu rüya seni biraz endişelendirmiş gibi duruyor.", "Rüyanın ardından daha stresli ve kaygılı görünüyorsun.", "Rüyanın seni ne kadar korkuttuğunu görmek üzücü.", "Bu rüya sana gerçekten sıkıntı vermiş gibi.", "Rüyanın seni olumsuz etkilediğini duymak üzücü.", "Rüyanın ardından kendini kötü hissettiğini söylemek doğru olur mu?", "Bu rüya sana huzursuzluk ve endişe getirmiş gibi görünüyor.", "Rüyanın sana ne kadar kötü geldiğini görmek üzücü.", "Rüyanın ardından olumsuz bir değişim fark ettim sende."],

```
},  
  
{"tag" : "lucid",  
  
"patterns" : [],  
  
"responses": ["Rüyanın farkında olduğun anlar oldu mu?", "Rüyanda, rüya gördüğünü biliyor muydun?", "Rüyanda olayları kontrol edebildiğin anlar oldu mu?", "Rüya sırasında bilinçli olarak kararlar alabildin mi?", "Rüyanın içinde olup bitenleri yönlendirebildin mi?", "Rüyanda, rüya olduğunu fark ettiğinde nasıl hissettin?", "Rüyanda, gerçek dünyadan farkı algılayabildin mi?", "Rüyanda, bilinçli olarak yaptığı bir şey var mıydı?", "Rüyanda çevrendeki detayları daha net veya daha canlı bir şekilde algıladın mı?", "Rüyanda kendini uyandırmak istediği anlar oldu mu?", "Rüyanda, normalde yapamayacağın şeyleri yapabildin mi?", "Rüyanda herhangi bir testi, örneğin eline bakmak veya bir metni tekrar okumak gibi, gerçekleştirdin mi?", "Rüyanda bilinçli olarak uçma veya nesneleri hareket ettirme gibi şeyler deneden mi?", "Rüyanda, kendi iradenle bir yerden bir yere gitmeyi başarabildin mi?", "Rüyanın akışını değiştirmek istediğiinde başarılı oldun mu?"],  
  
},  
  
{"tag" : "nightmare",  
  
"patterns" : [],  
  
"responses": ["Rüyanda korkutucu veya rahatsız edici bir şey yaşadın mı?", "Rüyanın seni çok ürkütüğünü hissettin mi?", "Bu gece rüyan seni uykundan uyandırdı mı?", "Rüyandan sonra ter içinde mi uyandın?", "Rüyanın etkisiyle kalbin hızlı mı atıyordu?", "Rüyadan sonra çarpıntı veya nefes darlığı yaşadın mı?", "Rüyanda kendini kaçarken veya saklanırken buldun mu?"]  
  
],  
  
},  
  
{"tag" : "gelecek",  
  
"patterns" : [],  
  
"responses": ["Rüyanda ne tür geleceğe dair olaylar veya görüntüler gördün?", "Rüyanda gördüğün gelecekle ilgili olaylar seni nasıl etkiledi?", "Gelecekle ilgili rüyaların sıkça mı olur?" ]},
```

```
}

]

intents = json.dumps(json_string)

intents
```

Intelligence Chatbot Ai için:

```
import random

import json

import pickle

import numpy as np


import nltk

from nltk.stem import WordNetLemmatizer


from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense, Activation, Dropout

from tensorflow.keras.optimizers import SGD


lemmatizer = WordNetLemmatizer()


intents = json.loads(open('intents.json').read())


words = []

classes = []
```

```
documents = []

ignore_letters = ['?', '!', '.', ',']

for intent in intents['intents']:

    for pattern in intent['patterns']:

        word_list = nltk.word_tokenize(pattern)

        words.extend(word_list)

        documents.append((word_list, intent['tag']))

        if intent['tag'] not in classes:

            classes.append(intent['tag'])

words = [lemmatizer.lemmatize(word) for word in words if word not in
ignore_letters]

words = sorted(set(words))

classes = sorted(set(classes))

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

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



training = []

output_empty = [0] * len(classes)

for document in documents:

    bag = []
```

```
word_patterns = document[0]

word_patterns = [lemmatizer.lemmatize(word.lower()) for word in
word_patterns]

for word in words:

    bag.append(1) if word in word_patterns else bag.append(0)

output_row = list(output_empty)

output_row[classes.index(document[1])] = 1

training.append([bag, output_row])

random.shuffle(training)

training = np.array(training)

train_x = list(training[:, 0])

train_y = list(training[:, 1])

model = Sequential()

model.add(Dense(128, input_shape=(len(train_x[0]),),
activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(64, activation='relu'))

model.add(Dropout(0.5))

model.add(Dense(len(train_y[0]), activation='softmax'))

sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
```

```

model.compile(loss='categorical_crossentropy', optimizer=sgd,
metrics=[ 'accuracy'])

model.fit(np.array(train_x), np.array(train_y), epochs=200,
batch_size=5, verbose=1)

model.save('chatbot_model.model')

print("Done")

```

Kütüphanelerin İçe Aktarılması:

- `import random`: Rastgele öğeleri seçmek için.
- `import json`: JSON dosyalarını okumak ve yazmak için.
- `import pickle`: Verileri seri hale getirmek ve kaydetmek için.
- `import numpy as np`: Dizileri işlemek için.
- `import nltk`: Doğal dil işleme işlevleri için.
- `from nltk.stem import WordNetLemmatizer`: Kelimeleri köklerine ayırmak için.
- `from tensorflow.keras.models import Sequential`: Keras ile sinir ağı modelleri oluşturmak için.
- `from tensorflow.keras.layers import Dense, Activation, Dropout`: Keras ile katmanlar oluşturmak için.
- `from tensorflow.keras.optimizers import SGD`: Keras ile optimize edicileri kullanmak için.

Ön İşleme ve Veri Hazırlama:

- `WordNetLemmatizer` sınıfı kullanılarak kelimeler köklerine ayrılır.
- JSON formatındaki etiket ve kalıplar dosyası (`intents.json`) okunur.
- Kelimeler, sınıflar ve belgeler listeleri oluşturulur.
- Her bir belge için bir "torba" oluşturulur. Torba, belirli bir belgenin içinde geçen kelimelerin varlık durumlarını içerir.
- Eğitim verileri hazırlanır ve karıştırılır.

Model Oluşturma ve Eğitimi:

- `Sequential` modeli oluşturulur.
- Model katmanları tanımlanır: giriş katmanı, gizli katmanlar ve çıkış katmanı.
- Model derlenir: Kayıp fonksiyonu, optimize edici ve metrikler belirlenir.
- Eğitim verileriyle model eğitilir.

Modelin Kaydedilmesi:

- Eğitilen model `chatbot_model.model` dosyasına kaydedilir.

Bitti Mesajı:

- Eğitim tamamlandığında "Done" mesajı görüntülenir.

Rüyanın Sentiment değerine bağlı olarak cevap vermesi için yazılan kod :

```
def chatbot_response(msg):  
  
    blob = TextBlob(msg)  
  
    sentiment = blob.sentiment.polarity  
  
    if sentiment > 0:  
  
        # Positive sentiment  
  
        tag = 'olumlu'  
  
    elif sentiment < 0:  
  
        # Negative sentiment  
  
        tag = 'olumsuz'  
  
    else:  
  
        # Neutral sentiment  
  
        tag = 'olaylar'
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı : Duygu Gözde KAYABAŞI

Uyruğu : Türkiye (T.C.)

Doğum Tarihi ve Yeri : 18.09.2002 - MERSİN

Telefon : 0 535 055 33 06

E-posta : duygugozde33@gmail.com

1030510338@erciyes.edu.tr

Adres : Limonluk Mah. , 2430. Sok., Selenevler Apartmanı, No: 8,

Kat : 10 / Daire : 20,

YENİŞEHİR/MERSİN

TÜRKİYE

EĞİTİM

Derece Kurum

Mez.Yılı

Lise 75. YIL FEN LİSESİ , MERSİN 2020

Ortaokul ÇANKAYA ORTAOKULU, MERSİN 2016

KAYNAKÇA

- [Dream Interpreter AI](#)
- [AI Dream Interpreter](#)
- [Dream Interpreter AIM: Uncover Hidden Insights in Your Dreams with AI | Deepgram](#)
- [Use Figma to create a drag and drop User Interface for Python using TkInter](#)
- [Intelligent AI Chatbot in Python](#)
- [Twitter Sentiment Analysis in Python](#)
- [Simple Sentiment Text Analysis in Python](#)
- [Dreamers](#)
- [ChatBot DataSet](#)
- [ChatGPT](#)
- [How to translate languages in Python with Google Translate and DeepL \(plus more\) - Lokalise Blog](#)
- [How to detect if a specific key pressed using Python? - GeeksforGeeks](#)
- [How to convert CSV columns to text in Python? - GeeksforGeeks](#)
-

Duygu Gözde KAYABAŞI
1030510338