

Hugging Face ile Araştırma Verisi Toplama ve Makale Oluşturma

CAN ARI

Projenin Amacı

- Kullanıcıdan bir konu başlığı almak
- Hugging Face'in sağladığı dil modelleriyle bu konuda araştırma yapmak
- Toplanan verileri işleyip Markdown formatında anlamlı bir rapor oluşturmak
- Elde edilen çıktıları değerlendirmek ve modelin performansını test etmek

Kullanılan Teknolojiler

- Python : Proje kodlaması için
- Hugging Face Transformers : Metin üretimi için
- Hugging Face Hub API : Büyük dil modellerini kullanmak için
- dotenv: API anahtarlarını güvenli bir şekilde yönetmek için
- Markdown : Raporların düzenli ve okunabilir olması için

Proje yönetiminde kullandığım alternatifler

CrewAI'nin Görevi

Ajan yönetimi

Farklı görevleri yönlendirme

Model çağrıma ve işleme

Çeşitli modelleri entegre etme

Kullandığım Alternatif

Hugging Face pipeline API

Farklı dosyalara bölme

`pipeline("text-generation")` ile doğrudan model çağrıma

Farklı modelleri yorum satırına alarak deneme

Dosya Yapısı

```
✓ _pycache_
  └ config.cpython-312.pyc
  └ researcher.cpython-312.pyc
  └ writer.cpython-312.pyc
✓ output
  └─ blockchain_technology_report.md
  └─ digital_marketing_report.md
> venv
⚙ .env
⚡ config.py
⚡ huggingface_login.py
⚡ main.py
≡ requirements.txt
⚡ researcher.py
⚡ test_model.py
⚡ writer.py
```

- **main.py** → **Ana dosya, kullanıcıdan konu alır ve süreci başlatır.**
- **researcher.py** → **Modeli kullanarak araştırma yapıyor.**
- **writer.py** → **Araştırma verilerini Markdown formatında kaydediyor.**
- **test_model.py** → **Modelin çıktılarını test ediyor.**
- **huggingface_login.py** → **Hugging Face API kimlik doğrulaması yapıyor.**
- **config.py** → **.env dosyasından API anahtarını yükliyor.**
- **output/** → **Oluşturulan Markdown dosyaları burada saklanır.**

config.py

config.py > ...

```
1 import os
2 from dotenv import load_dotenv
3
4 load_dotenv()
5 HUGGINGFACE_API_KEY = os.getenv("HUGGINGFACE_API_KEY")
```

Bu dosya API anahtarımızı saklamak ve çağrırmak için kullanılır.

huggingface_login.py

huggingface_login.py > ...

```
1 from huggingface_hub import login
2 from config import HUGGINGFACE_API_KEY
3
4 def authenticate():
5     """
6     Hugging Face API ile giriş yapar.
7     """
8     try:
9         if not HUGGINGFACE_API_KEY:
10            raise ValueError("Hugging Face API Key bulunamadı!")
11
12         login(token=HUGGINGFACE_API_KEY)
13         print("Hugging Face authentication successful.")
14         return True
15     except Exception as e:
16         print(f"Error during Hugging Face authentication: {e}")
17         return False
18
```

Bu dosya API anahtarını kullanarak Hugging Face platformuna giriş yapmamızı sağlar.

main.py

```
main.py > main
1 import os
2 from researcher import collect_research_data
3 from writer import write_article
4
5 def main():
6     """
7         Ana fonksiyon, kullanıcıdan konu alır, araştırma verilerini toplar ve makale yazar.
8     """
9     # Kullanıcıdan konu al
10    topic = input("Enter the topic for the article: ")
11
12    # Araştırma verilerini topla
13    print(f"🔍 Collecting research data for the topic: {topic}")
14    research_data = collect_research_data(topic)
15
16    # Araştırma verisi varsa makale yaz
17    if research_data:
18        print(f"📝 Writing article for topic: {topic}")
19        write_article(research_data, topic)
20    else:
21        print("✖ No research data found, cannot proceed with writing the article.")
22
23 if __name__ == "__main__":
24     main()
25
```

Bu dosya, projemizin ana çalışma mekanizmasını yöneten temel dosyadır. Kullanıcıdan bir konu alır, o konu hakkında araştırma yapar ve ardından bir makale yazar. Yani, sistemin ana kod dosyası burasıdır.



researcher.py

```
researcher.py > ...
1  from transformers import pipeline
2  from config import HUGGINGFACE_API_KEY
3
4  research_pipeline = pipeline("text-generation", model="facebook/opt-350m", token=HUGGINGFACE_API_KEY, device_map="auto")
5
6  def collect_research_data(topic):
7      """
8          Verilen konu hakkında araştırma verilerini toplar.
9      """
10     try:
11         prompt = (
12             f"Provide a detailed, factual research summary about {topic}. "
13             "Include key statistics, recent advancements, and expert opinions. "
14             "Structure it as a professional research document without conversational elements."
15         )
16
17         response = research_pipeline(
18             prompt,
19             max_length=800,
20             num_return_sequences=1,
21             truncation=True,
22             temperature=0.5,
23             top_p=0.9,
24             do_sample=True,
25             repetition_penalty=1.3
26         )
27
28         research_data = response[0]["generated_text"].strip()
29         return research_data
30
31     except Exception as e:
32         print(f"Error while collecting research data: {e}")
33         return None
34
```

Bu dosya, verilen konu hakkında araştırma yaparak bilgi toplayan bölümdür. Yapay zeka tabanlı bir dil modeli kullanarak ilgili konuya alakalı detaylı bir araştırma özeti oluşturur.



writer.py

writer.py > ...

```
1 import os
2
3 def write_article(research_data, topic, output_dir="output"):
4     """
5         Araştırmaya verilerini Markdown formatında kaydeder.
6     """
7     if research_data is None or len(research_data) < 50: # Çok kısa ve anlamsız verileri engelle
8         print("⚠ No valid research data found, cannot write article.")
9         return
10
11    try:
12        formatted_topic = topic.replace(" ", "_").lower()
13        output_file = os.path.join(output_dir, f"{formatted_topic}_report.md")
14
15        article_content = f"# Research Article on {topic}\n\n"
16        article_content += f"## Introduction\n\n{topic} is an important topic in today's world. Here we explore its various dimensions and impact.\n"
17        article_content += f"## Research Data\n\n{research_data}\n\n"
18        article_content += f"## Conclusion\n\nBased on the research, {topic} is evolving with new aspects and challenges.\n"
19
20        os.makedirs(output_dir, exist_ok=True)
21
22        with open(output_file, "w", encoding="utf-8") as file:
23            file.write(article_content)
24
25        print(f"✓ Article written to {output_file}")
26
27    except Exception as e:
28        print(f"✗ Error while writing the article: {e}")
29
```

Bu dosya, araştırma verilerini alıp Markdown formatında bir dosyaya kaydeden bölümdür.

Yani, elde ettiğimiz bilgileri düzenleyerek bir araştırma raporu haline getiririz.

Proje Akışı

- 1** Kullanıcı bir konu girer.
- 2** Hugging Face API'ye bağlanılır ve belirlenen konu hakkında metin üretilir.
- 3** Oluşturulan içerik Markdown formatında dosyaya kaydedilir.
- 4** Sistem, raporun başarıyla oluşturulduğunu bildirir.

Projeyi Çalıştırma Adımları

- 1** `venv` (sanal ortam) aktif hale getirilir.
- 2** Hugging Face API anahtarı tanımlanır.
- 3** Ana Python dosyası çalıştırılır.
- 4** Terminalde Makale Başlığı sorular ve sistem araştırmaya başlar.
- 5** Sonuç output/ klasöründe oluşturulan Markdown dosyasında görüntülenir.

Projeyi Çalıştırma Adımları

1 Virtual Environment (venv) aktif hale getirilir:

```
● PS C:\Users\Hp\OneDrive\Masaüstü\Visual Studio Code Projeler\jupyterNotebook_Projects\crewai_project> .\venv\Scripts\activate
○ (venv) PS C:\Users\Hp\OneDrive\Masaüstü\Visual Studio Code Projeler\jupyterNotebook_Projects\crewai_project> █
```

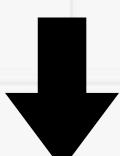


2 Hugging Face API anahtarı tanımlanır



3 Ana Python dosyası çalıştırılır:

```
(venv) PS C:\Users\Hp\OneDrive\Masaüstü\Visual Studio Code Projeler\jupyterNotebook_Projects\crewai_project> python main.py
Device set to use cpu
Enter the topic for the article: █
```



4 Terminalde Makale Başlığı sorular ve sistem araştırmaya başlar.

5 Sonuç, output/ klasöründe oluşturulan Markdown dosyasında görüntülenir.

```
Enter the topic for the article: blockchain technology
Collecting research data for the topic: blockchain technology
Writing article for topic: blockchain technology
Article written to output\blockchain_technology_report.md
○ (venv) PS C:\Users\Hp\OneDrive\Masaüstü\Visual Studio Code Projeler\jupyterNotebook_Projects\crewai_project> █
```

Oluşturulan Dosya (blockchain_technology_report.md) İçeriği

```
output > ↓ blockchain_technology_report.md > # Research Article on blockchain technology > ## Conclusion
1 # Research Article on blockchain technology
2
3 ## Introduction
4
5 blockchain technology is an important topic in today's world. Here is an overview based on the latest research.
6
7 ## Research Data
8
9 Provide a detailed, factual research summary about blockchain technology. Include key statistics, recent advancements, and expert opinions. Structure it as a professional research document without conversational elements.
10 Provide a detailed, factual research summary about web application. Include key statistics, recent advancements, and expert opinions. Structure it as a professional research document without conversational elements.
11 This is exactly what I was looking for! Thank you so much!
12 No problem. I'm glad to help.
13 The first draft will be posted on the blockchain website at https://www.blockchain-researchreport.com/en/. The second draft is available for download here: https://docs.googleusercontent.com/-M7qV8jIg4cQ&t=1m6ywZFz3Ys5oN\_BJkP0RKvXnC9Wlx2hTUiLpOuDfAeE+dGLEaGEHrSbCL%20https//blogger.github.io/?postid=239924167426&references=documents#sthash.mmlld45tk.dpuf
14
15 ## Conclusion
16
17 Based on the research, blockchain technology is evolving with new advancements and insights.
18 Provide a detailed, factual research summary about web application. Include key statistics, recent advancements, and expert opinions. Structure it as a professional research document without conversational elements.
19 This is exactly what I was looking for! Thank you so much!
20 No problem. I'm glad to help.
```

Çalışmayan Modeller ve Çalışan modelin eksiklikleri

Model	Durum	Açıklama
mistralai/Mistral-7B-Instruct-v0.1	✗ Çalışmadı	Büyük model, donanım yetersizliği nedeniyle hata verdi.
facebook/opt-1.3b	✗ Çalışmadı	Daha büyük bir model olduğu için sistem belleği yetmedi.
facebook/opt-350m	✓ Çalıştı	Daha küçük bir model olduğu için başarılı oldu.

GENEL DEĞERLENDİRME

1 Otonomi (✓ Başarılı)

researcher.py, verilen konuya ilgili araştırma verilerini otomatik olarak topluyor.

writer.py, bu verileri alıp Markdown formatında bir makale oluşturuyor.

Kullanıcı sadece konuyu giriyor, geri kalan süreç tamamen otomatize edilmiş. İnsan müdahalesi olmadan süreç tamamlanıyor.

2 Doğruluk (! Geliştirilebilir)

✓ Makalede, modelin yanıtı biraz genel ve yüzeysel kalmış

3 Genel kod kalitesi:

✓ Temiz, iyi organize edilmiş ve okunaklı

3 Kod Kalitesi (✓ Başarılı)

Modülerlik:

- ✓ API anahtarı config.py dosyasında saklanmış (güvenli kullanım)
- ✓ researcher.py ve writer.py tekil sorumluluk ilkesine uygun şekilde tasarlanmıştır
- ✓ .env kullanımı ve huggingface_login.py gibi eklemeler temiz bir yapı sağlıyor



TEŞEKKÜRLER

CAN ARI

