19.05.2025

# IMG_256

# Derin Öğrenme Uygulamaları

# Ders Projesi

# (BİLGİSAYAR MÜHENDİSLİĞİ\_YL)

Yasin Ünal

503020250009

# İçindekiler

# Giriş

Bu proje, gerçek dünyadaki bilgi yükü problemini azaltmak için geliştirilmiş yapay zeka destekli bir metin işleme asistanı sunmayı hedeflemektedir. Her gün onlarca haber ve bilgi akışı arasında bazı olaylar gözden kaçabilmektedir. Bu sorunu çözmek ve daha çok bilgiye daha kısa sürelerde erişmek için bu proje geliştirilmiştir.

# Projenin Amacı

Projede iki farklı işlev bulunmaktadır. İlk modül kullanıcılara basit bir arayüz ile istedikleri metinleri özetleme imkanı sunmaktadır. Burada girilen metin arka planda çalışan derin öğrenme modeli ile özetlenir ve kullanıcıya kısaca gösterilir. Özetin kaç saniyede yapıldığı ve kaç karaktere düşürüldüğü de bu sayfada yer alır. İkinci modül ise üç farklı haber kaynağından son haberleri çekerek (web-scrapping), güncel haberleri özetleme imkanı sunmaktadır. Burada bulunan arayüzde doğrudan habere gidebilir, haberi özetletebilir veya sayfanın üstünden tüm haberleri özetletebilirsiniz.

Kısaca bu projenin sağladıkları:

* Kullanıcıların uzun Türkçe metinleri kolayca özetlemesini sağlamak
* Farklı kategorilerdeki haberleri otomatik şekilde çekip özetlemek
* Kullanıcılara modeli ve özelliklerini tanıtan bir bilgi paneli sunmak
* Kullanışli bir arayüz ile hızlı ve etkili bir deneyim sağlamak

1. **Kullanılan Teknolojiler**

Bu projenin geliştirilmesinde aşağıdaki teknolojiler kullanılmıştır.

**Streamlit**: Hızlı web arayüz geliştirme için

**HuggingFace Transformers**: Derin öğrenme modeli için

**mT5-small**: T5 mimarisinin çok dilli versiyonu

**BeautifulSoup & Requests**: Haber içeriklerini web sitelerinden çekmek için

1. **Kullanılan Derin Öğrenme Modeli**

Özetleme için kullanılan derin öğrenme modeli HuggingFace platformunda bulunan bir modeldir. Bu modelin bilgileri aşağıdaki gibidir :

**Model**: ozcangundes/mt5-small-turkish-summarization

**Mimari**: mT5-small (Multilingual T5)

**Parametre Sayısı**: ~300M

**Model Boyutu**: ~1.2 GB

**Eğitim Aracı**: PyTorch Lightning

**Eğitim Verisi**: MLSUM veri setinden 20.000 haber ve 4.000 özet

**Eğitim Detayları**

**Epoch**: 10

**Batch Size**: 8

**Learning Rate**: 1e-4

**Max Input Length**: 784

**Max Summary Length**: 64

Bu model, önceden eğitilmiş bir mT5 modeli olup, Türkçe haber özetleme görevine özel olarak fine-tune edilmiştir. Model encoder-decoder yapısıyla çalışmakta ve sequence-to-sequence işlemleri yürütmektedir. Mimari şemaya rapor sonunda yer verilecektir.

1. **Uygulama**

Uygulamanın gerçeklenmesi için streamlit framework’ü kullanılmıştır. Streamlit kolayca indirilerek python ile arayüzler, etkileşimli araçlar, veri analizi ve makine öğrenimi uygulamaları oluşturmak için kullanılabilir. Uygulama iki ana modüle ayrılmıştır. İlk modül kullanıcıdan alınan metinin özetlenmesiyken ikinci modül haberlerin öztlenmesidir.

## Modül 1 : Metin Özetleyici

Bu modül, kullanıcının serbest metin girmesine imkan tanır ve model üzerinden özetleme yapar. Yan panelde kullanıcının seçebileceği özet uzunluğu ve beam search parametreleri gibi ileri ayarlar bulunur. Özetleme sonrasında orijinal ve özet metnin karakter sayısı kıyaslanarak istatistik sunulur.



Kullanıcıdan input almak için aşağıdaki kod kullanılır

text = st.text\_area("📌 Haber Metnini Girin:", default\_text, height=200)

Bu sayfada bir chat kutusu oluşturur. Buraya girilen kullanıcı metni aşağıdaki kod ile bir buton oluşturularak modele gönderilir.

if st.button("🧠 Özetle"):

    if len(text.strip()) == 0:

        st.warning("Lütfen özetlenecek bir metin girin.")

    else:

        start\_time = time.time()

        with st.spinner("Metin özetleniyor..."):

            try:

                summary = summarizer(

                    text,

                    max\_length=max\_length,

                    min\_length=min\_length,

                    num\_beams=num\_beams,

                    no\_repeat\_ngram\_size=no\_repeat\_ngram\_size,

                    early\_stopping=True

                )[0]['summary\_text']

*# process üsresi*

                process\_time = time.time() - start\_time

                st.subheader("📄 Özet:")

                st.success(summary)

                st.info(f"Özetleme süresi: {process\_time:.2f} saniye")

*# Karakter sayıları gösterimi*

                col1, col2 = st.columns(2)

                with col1:

                    st.metric("Orijinal metin", f"{len(text)} karakter")

                with col2:

                    st.metric("Özet", f"{len(summary)} karakter",

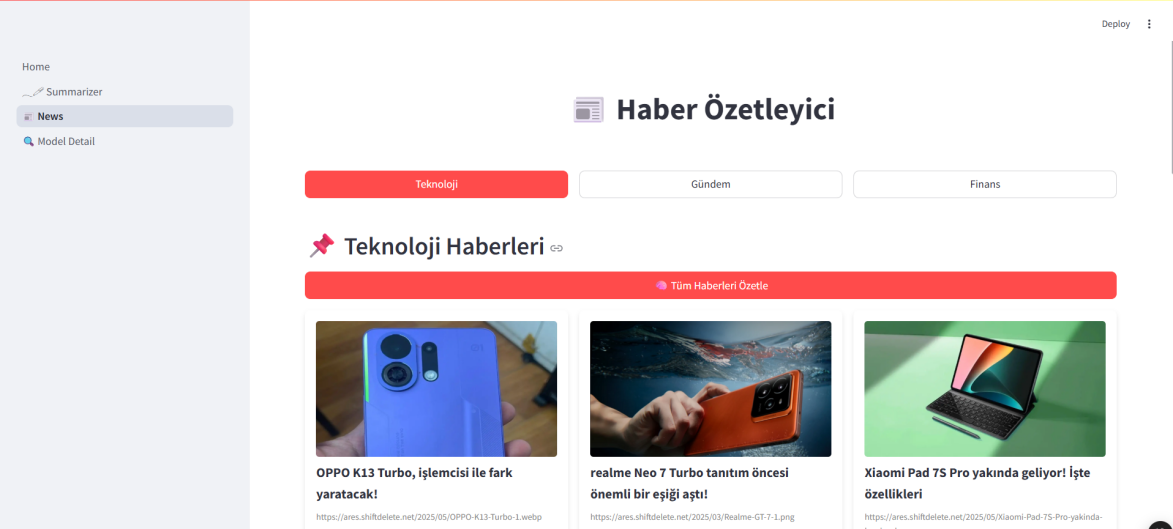
                              f"-%{int((1 - len(summary)/len(text)) \* 100)}")

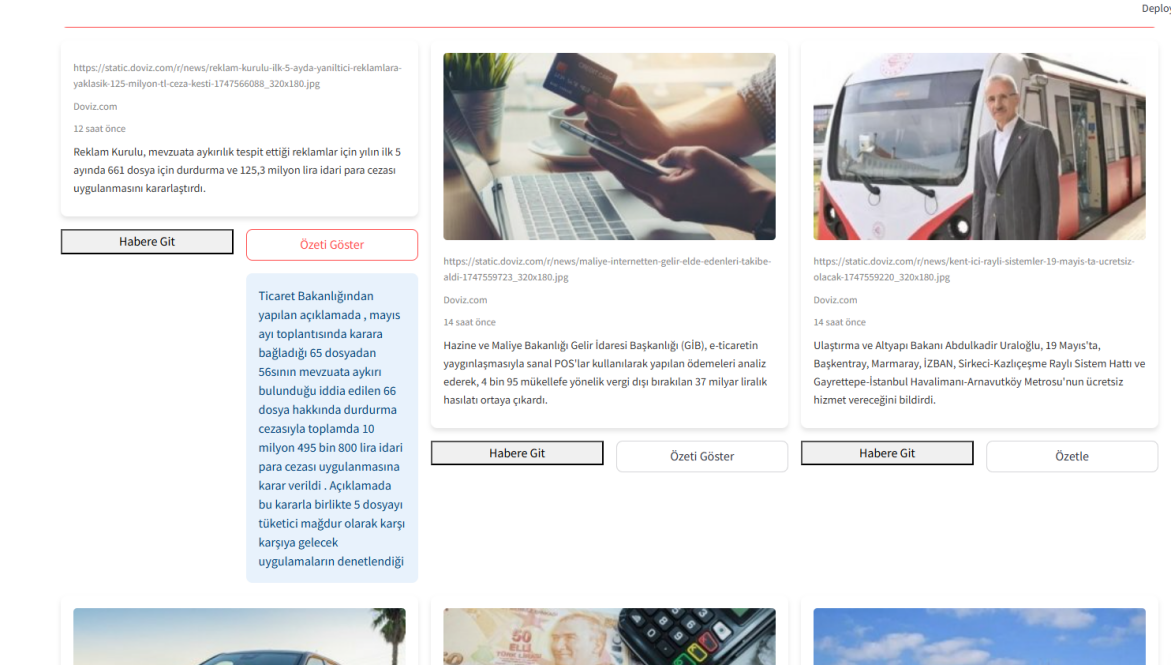
            except Exception as e:

                st.error(f"Özetleme sırasında bir hata oluştu: {str(e)}")

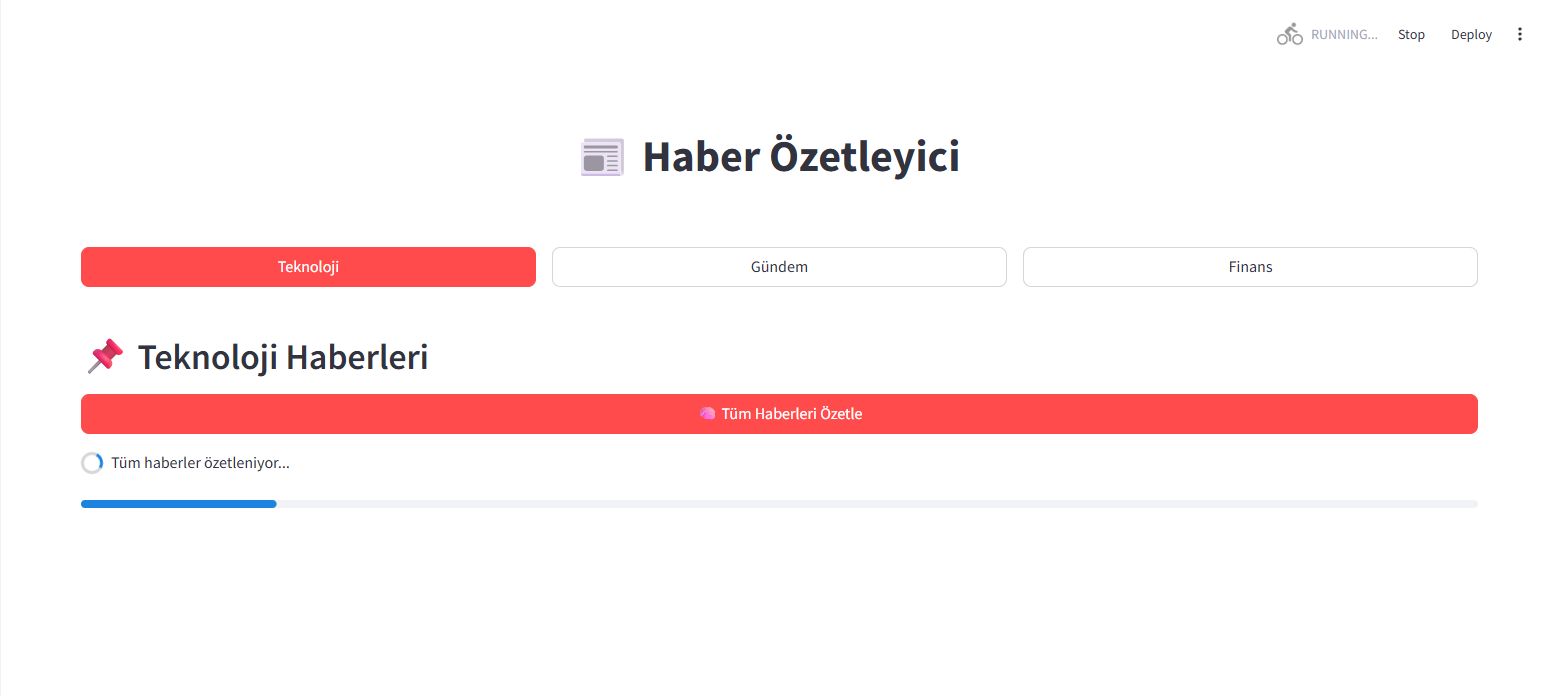
## Modül 2 : Haber Özetleyici

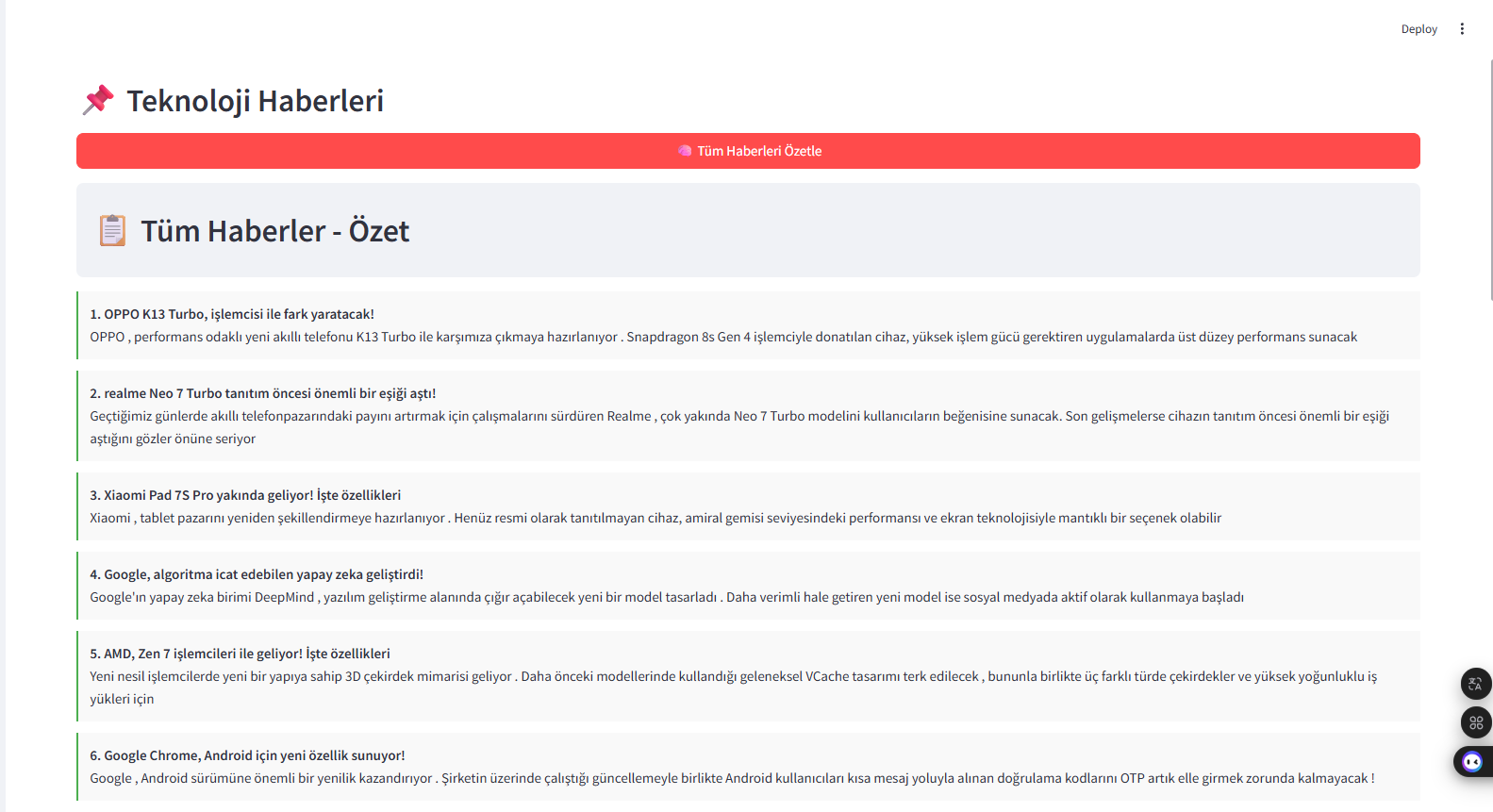
Bu modül, haber sitelerinden (ShiftDelete, Doviz, Hürriyet) otomatik olarak haberleri çeker ve her bir haberin detayını model ile özetler. Kategori seçimi yapılarak sadece ilgili haberler listelenebilir. Kullanıcı her haberi teker teker veya toplu olarak özetleyebilir. Özetler sayfada dinamik olarak gösterilir.





Tüm haberlerin özetlenmesi :





Ana sayfada ilk haber türü seçimi kısmıdır. Bu kısım aşağıdaki kodlar ile oluşturulmaktadır :

*# Kategori tanımları*

CATEGORIES = {

    "Teknoloji": {

        "url": "https://shiftdelete.net",

        "selector": "custom"  *# Özel işleyici için işaret*

    },

    "Gündem": {

        "url": "https://www.hurriyet.com.tr/gundem/",

        "selector": "div.container div.news-card"

    },

    "Finans": {

        "url": "https://www.doviz.com",

        "selector": "custom"  *# Özel işleyici için işaret*

    }

}

. . .

for i, category in enumerate(CATEGORIES.keys()):

        with cols[i]:

            if st.button(

                category,

                key=f"cat\_{category}",

                use\_container\_width=True,

                type="primary" if st.session\_state.selected\_category == category else "secondary"

            ):

                change\_category(category)

Seçilen kategoriye göre haber içerikleri aşağıdaki kodda alınmaktadır :

st.session\_state.news\_data = fetch\_news\_list(st.session\_state.selected\_category)

Burada yer alan fetch\_news\_list metodu aşağıdaki gibidir :

def fetch\_news\_list(category):

    news\_list = []

    try:

        cat\_info = CATEGORIES[category]

*# Teknoloji kategorisi için özel işleme*

        if category == "Teknoloji":

            return [

                {

                    "id": hashlib.md5(news["link"].encode()).hexdigest(),

                    "title": news["title"],

                    "url": news["link"],

                    "image": news["image"],

                    "summary": news["summary"],

                    "source": "ShiftDelete.Net",

                    "date": news["time"],

                    "category": category,

                    "content": news["content"],

                    "ai\_summary": None

                }

                for news in teknoloji\_news(cat\_info["url"], limit=10)

            ]

*# Finans kategorisi için özel işleme*

        if category == "Finans":

            return [

                {

                    "id": hashlib.md5(news["link"].encode()).hexdigest(),

                    "title": news["title"],

                    "url": news["link"],

                    "image": news["image"],

                    "summary": news["summary"],

                    "source": "Doviz.com",

                    "date": news["time"],

                    "category": category,

                    "content": news["content"],

                    "ai\_summary": None

                }

                for news in finans\_news(cat\_info["url"], limit=10)

            ]

        headers = {

            'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'

        }

        st.write(f"Veri çekiliyor: {cat\_info['url']}")

        response = requests.get(cat\_info["url"], headers=headers, timeout=10)

        if response.status\_code != 200:

            st.error(f"Veri çekilirken hata oluştu: {response.status\_code}")

        else:

            st.success(f"STATUS[{response.status\_code}]")

            if response.status\_code == 200:

                soup = BeautifulSoup(response.text, 'html.parser')

                news\_items = soup.select(cat\_info["selector"])

                for item in news\_items[:10]:

                    st.write(f"İşleniyor: {item}")

                    try:

                        title\_element = item.select\_one("div.news-title a, h3.news-title a")

                        if not title\_element:

                            continue

                        title = title\_element.text.strip()

                        url = title\_element['href']

*# Eğer URL tam değilse (bağıl yol), tam URL'ye dönüştür*

                        if not url.startswith('http'):

                            parsed\_base = urlparse(cat\_info["url"])

                            base\_url = f"{parsed\_base.scheme}://{parsed\_base.netloc}"

                            url = base\_url + url

*# Resim bulma*

                        img\_element = item.select\_one("img")

                        img\_url = img\_element['src'] if img\_element and 'src' in img\_element.attrs else ""

*# Eğer resim URL'si tam değilse (bağıl yol), tam URL'ye dönüştür*

                        if img\_url and not img\_url.startswith('http'):

                            parsed\_base = urlparse(cat\_info["url"])

                            base\_url = f"{parsed\_base.scheme}://{parsed\_base.netloc}"

                            img\_url = base\_url + img\_url

*# Özet bulma*

                        summary\_element = item.select\_one("div.news-summary, div.news-description")

                        summary = summary\_element.text.strip() if summary\_element else ""

*# Kaynak ve tarih tanımlama*

                        source = "Hürriyet"

                        date = datetime.now().strftime("%d.%m.%Y")

*# Haberi benzersiz tanımlamak için hash oluşturma*

                        news\_id = hashlib.md5(url.encode()).hexdigest()

                        news\_list.append({

                            "id": news\_id,

                            "title": title,

                            "url": url,

                            "image": img\_url,

                            "summary": summary,

                            "source": source,

                            "date": date,

                            "category": category,

                            "content": None,  *# İçerik sonradan çekilecek*

                            "ai\_summary": None  *# AI özeti sonradan oluşturulacak*

                        })

                    except Exception as e:

                        continue

        return news\_list

    except Exception as e:

        st.error(f"Haberler çekilirken hata oluştu: {str(e)}")

        return []

Bu kodda yer alan