

**Yıldız Teknik Üniversitesi**  
**Elektrik Elektronik Fakültesi**  
**Bilgisayar Mühendisliği Bölümü**



**Yapay Zeka Dersi**  
**Dönem Projesi Raporu**

20011024 – Sait Yalçın

20011901 – Muhammed Kayra Bulut

sait.yalcin@std.yildiz.edu.tr

kayra.bulut@std.yildiz.edu.tr

**Dersin Yürütücüsü:** Prof.Dr. Mehmet Fatih AMASYALI

# 1. CHATBOT NEDİR?

Chatbotlar, insan dilini analiz ederek ve uygun şekilde yanıtlayarak yazılı iletişimde bulunan bir tür yazılım uygulamasıdır. Genellikle müşteri desteği sağlamak, rezervasyonları yönetmek ve bilgi yaymak gibi çeşitli görevler için programlanırlar. İnsan dilini anlama ve yanıtlama yeteneği sayesinde, chatbotlar genellikle bir web sitesinde veya bir mesajlaşma platformunda kullanılarak insanların belirli sorgularını yanıtlar veya gereken işlemleri yürütürler.

## 1.1 Chatbotların Yapay Zeka İle İlgisi Nedir?

Chatbotlar ve yapay zeka (AI) sıkı sıkıya bağlıdır çünkü çoğu modern chatbot, doğal dil işleme (NLP), makine öğrenmesi ve derin öğrenme gibi AI teknolojilerini kullanır. Bu teknolojiler, chatbotların insan dilini anlamasına, kullanıcının girdisine göre yanıtlar oluşturmaya ve daha doğru ve özelleştirilmiş yanıtlar vermek için geçmiş etkileşimlerden öğrenmesine yardımcı olur. AI, bir chatbot'un kullanıcının niyetini anlamasına ve daha karmaşık sorguları yanıtlamasına olanak tanır. Öyleyse, yapay zeka, chatbotların daha insana benzer bir iletişim kurabilmesini ve kullanıcıların ihtiyaçlarına daha etkili bir şekilde hizmet edebilmesini sağlar.

## 1.2 Chatbotlar nasıl çalışır?

Chatbotlar, çeşitli yöntemlerle programlanırlar ve çalışırlar. Bir chatbot, bir insanla yazılı olarak sohbet edebilmek için özel bir dil öğrenir ve anlar. Bu dil, genellikle insan diline benzer bir dil olur, ancak bir chatbot tarafından anlaşılması için daha yapay bir dil olarak tasarlandığı da olabilir. Chatbotlar, insan dilini anladıklarında, bu dil üzerinden yapılan sorulara yanıt verirler. Chatbotların yanıtları, genellikle önceden programlanmış cevaplar veya veritabanından çekilen bilgilerden oluşur.

## 2. CHATBOT ÇEŞİTLERİ

Chatbotlar genellikle işlevlerine ve kullanılan teknolojilere göre sınıflandırılır. İşte en yaygın chatbot çeşitleri:

**2.1 Kurallara Dayalı Chatbotlar:** Bu chatbotlar önceden belirlenmiş kurallara göre çalışır. Sadece belirlenen senaryoları ve komutları anlarlar ve sınırlı yanıtlar verirler. Karmaşık soruları yanıtlamakta zorlanırlar.

**2.2 Yapay Zeka Chatbotları:** Bu chatbotlar, doğal dil işleme (NLP), makine öğrenmesi ve derin öğrenme gibi yapay zeka teknolojilerini kullanır. Kullanıcıların ihtiyaçlarını daha iyi anlamak ve onlara daha özel yanıtlar verebilmek için kullanıcıların önceki konuşmalarıyla kendini besleyebilir.

**2.3 Hizmet Chatbotları:** Bu chatbotlar, belirli bir hizmeti gerçekleştirmek için tasarlanmıştır. Örneğin, taksi çağırma, berber rezervasyonu yapma ya da yemek sipariş etme gibi.

**2.4 Bilgi Chatbotları:** Bu chatbotlar, belirli bir konuda bilgi vermek üzere tasarlanmıştır. Mesela, kullanıcıların kanser hakkındaki sorularını yanıtlamak gibi.

**2.5 Sosyal Medya Chatbotları:** Bu chatbotlar, sosyal medya platformlarındaki kullanıcı etkileşimlerini otomatikleştirmek için tasarlanmıştır. Örneğin, bir Instagram hesabındaki mesajları yanıtlama ya da Tinder hesabını yönetme gibi.

**2.6 Sesli Chatbotlar:** Bu chatbotlar, sesli girdi ve çıktıyı destekler. Örneğin, Apple'ın Siri'si ya da Google Asistan gibi.

### 3. CHATBOT PROJEMİZ

Projemizde transformer model kullanarak reddit üzerinden belli konu başlıklarının sayfalarından web scraping metodu ile çektiğimiz verilerle modelin eğitimin gerçekleştireceğiz. Projemizi gerçeklerken tensorflow, selenium, Numpy, beautifulsoup4, flatbuffers, keras gibi Python kütüphanelerini ve Python 3.10 versiyonunu kullandık.

Doysamız ve açıklamalarına kısaca değinirsek;

- **layers.py** – Transformer model için gereken sayısallaştırma metodlarını içerir.
- **utils.py** – Transformer model için gerekli olan tokenizer metodlarını ve diğer yardımcı metodları içerir.
- **export.py**– Modelin dosyadan checkpointleri çekip, ilklendirilmesini sağlayan init fonksiyonu ve girdi olarak verilen kelimelere çıktı üretilmesini sağlayan fonksiyonları içerir.
- **config.yml** – Çekilen verilerilerin sonucunda oluşan **train.from** ve **train.to** dosyalarını kullanarak, **training\_data** dosyasını, tek sayıdaki satırlar soru, çift sayıdaki satırlar cevap olacak şekilde oluşturur.
- **main.py** – Çalıştırılma durumuna göre, **train-eval-test** modunda modelin çalıştırılmasını sağlar.

Projemizi kabaca özetlersek

1. Web scraping metodlarını hazırladık ve verileri toplayabileceğimiz konuları belirledik.
2. Web scraping metodlarımızı çalıştırarak verileri topladık.
3. Modelimizi kaydederek yeni sorgulara hazırladık.
4. Arayüz üzerinden yeni sorguları alarak eğitilen modelimizdeki oluşan tahmini cevapları ekrana gönderdik.

### 3.1 Veri Setini Oluşturma ve Yükleme

Öncelikle, içeriğini bizim oluşturduğumuz, temel soru ve cevapları oluşturmak için oluşan `get_reddit_data.py` isimli bir dosya oluşturduk. Sonrasında bu kod yardımıyla redditteki verileri soru-cevap olarak çektik. Bunları `train.from` ve `train.to` olarak kaydettikten sonra, `dataloader.py` dosyası yardımıyla da verileri tek ayıdaki satırlar soru, çift sayıdaki satırlar cevap olacak şekilde oluşturduk.

Burada bazı sorular kendini tekrar ediyor, bunun sebebine bakacak olursak, bir sorunun cevabına karşılık başka bir cevap olabilmesi. Verilerimizin bir kısmı da aşağıdaki fotoğrafta görüldüğü gibidir.

```
78 training_data.txt
79 Because you haven't mastered the art of Naruto running
80 Preach
81 Preach
82 Preach!
83 Because you haven't mastered the art of Naruto running
84 That or all the women in his town have.
85 Because you haven't mastered the art of Naruto running
86 Teach me master. I'll do....anything
87 Why can't I get a girlfriend?
88 Because you don't use the right question word to formulate your interrogations.
89 Because you don't use the right question word to formulate your interrogations.
90 Dammit
91 Dammit
92 The aliens are only interested in alien boyfriends.
93 I am the Naruto Runner - AmA
94 oh dang, I just clicked on your profile and found this.
95 oh dang, I just clicked on your profile and found this.
96 Hi
97 Hi
98 THE MESSIAH HAS ACKNOWLEDGED ME OMG
99 THE MESSIAH HAS ACKNOWLEDGED ME OMG
100 deep breaths dude
101 deep breaths dude
102 Don't grovel, that would be weird. Just get on your knees and get the praying done with
103 THE MESSIAH HAS ACKNOWLEDGED ME OMG
104 Sh bby is ok
105 The gunman would've needed to master the art of Naruto running before they even came close
106 Well they did have a board meeting about it so some of them might be closer to that than you think
107 I am the Naruto Runner - AmA
108 Can I be a friend of Mr.Fishy's?
109 Can I be a friend of Mr.Fishy's?
110 Yes ok maybe
111 Yes ok maybe
112 How about me?
113 How about me?
```

### 3.2 Veri Setini Temizleme ve Ön İşlemden Geçirme

Metin verileriyle çalışırken makine öğrenimi veya derin öğrenme modeli yapmadan önce veriler üzerinde çeşitli ön işlemler gerçekleştirmemiz gerekir. Gereksinimlere bağlı olarak, verileri önceden işlemek için çeşitli işlemler uygulamamız gerekir.

Tokenization, metin verileri üzerinde yapabileceğiniz en temel ve ilk şeydir. Tokenization, tüm metni kelimeler gibi küçük parçalara ayırma işlemidir.

Burada kalıpları yineliyoruz, cümleyi tokenize ediyoruz ve kelime listesindeki her kelimeyi ekliyoruz. Ayrıca etiketlerimiz için bir sınıf listesi oluşturuyoruz.

Sonrasında verileri çeşitli ön işlemlerden geçiriyoruz. Bu da modelin başarısının artması için önemli bir etkiye sahip.

### 3.3 Modeli Oluřturma

Biz model olarak Transformer modeli kullandık. Transformer modeli, dikkat mekanizmalarını kullanarak karmařık dil iřleme grevlerini gerekleřtirebilen bir tr derin ğrenme modelidir. Transformer modelini eğitmek iin **main.py** dosyası iinde, train fonksiyonunu kullandık. Kod, verilen girdi ve hedef deęerleri zerinde belirli bir sayıda epoch boyunca alıřacak řekilde tasarlanmıřtır. Her bir epochta, kod ncelikle giriř ve hedef verilerinden maske oluřturur ve bu maskeleri Transformer modeline gnderir.

Modelden tahminler alındıktan sonra, kod hedef ve tahmin deęerleri arasındaki loss'u hesaplar. Bu loss, modelin ne kadar iyi performans gsterdięinin bir ltdr; kayıp deęeri dřtke, modelin performansı artar. Daha sonra kod, bu kayıp deęerini kullanarak modelin eęitilebilir parametrelerine gre gradyanları hesaplar ve bu gradyanları bir optimizr aracılıęıyla uygular. Bu, modelin parametrelerini gnceller ve modelin gelecekte daha dřk bir kayıp deęeri vermesini saęlar.

Kod ayrıca kayıp ve doęruluk metriklerini hesaplar ve bu deęerleri ekrana yazdırır. Bu, modelin eęitim sreci boyunca nasıl performans gsterdięinin bir lsdr. Son olarak, belirli bir dnem sayısında, kod eęitim srecinin bir checkpoint'ini kaydeder. Bu, eęitim srecini durdurma ve daha sonra kaldıęı yerden devam etme yeteneęi saęlar.

### 3.4 Cevap Tahmini

Eęitilmiş modeli ykleyerek ve ardından bottan gelen yanıtı tahmin edecek bir grafik kullanıcı arayz kullanacaęız. Bunun iin yine **main.py** dosyası iindeki evaluate fonksiyonunu kullandık.

Bu fonksiyon kodu basit anlamda, eęitilen modeli kullanarak aldıęı girdiye gre bir ıktı retilir. Kod ncelikle bařlangı ve bitiř belirtelerini girdi cmlesine ekler ve girdiyi modelin bekledięi řekle getirir. Ardından, ıktının bařlaması gereken yer iin bir bařlangı belirteci ekler.

Daha sonra kod, byk bir for dngs iinde alıřır. Her dngde, kod ncelikle girdi ve ıktı iin maske oluřturur ve bu maskeleri modelimize gnderir. Modelden tahminler ve dikkat aęırlıkları alındıktan sonra, kod en son tahmini alır ve bu tahminin en olası sonucunu belirler. Eęer bu sonu, ıktının bitiřiyle eřleřirse, kod ıktı dizisini ve dikkat aęırlıklarını dndrr. Eęer sonu bulunmazsa, kod tahmin edilen sonucu ıktı dizisine ekler ve dng devam eder.

Modeli oluřturduęumuzda eęitim 100 **Epoch** srd. **Loss** 0.1425 ve **Accuracy** 0.3635'a eřit oldu.

## 4. PROGRAMI ÇALIŞTIRMA

Chatbotu çalıştırmak için öncelikle train sonucu oluşan checkpointlerin checkpoints klasörü ismiyle ana dizinde bulunması gerekmektedir. *main.py* dosyası çalıştırılarak train işlemi gerçekleştirilebilir. Eğer hazır checkpoint kullanılmak istenirse, yaptığımız trainler sonucu oluşan checkpointlerin bulunduğu drive linki aşağıdadır;

[https://drive.google.com/drive/folders/13XL-9g-tn-Qb4aoH1I\\_fdc2kHWVTe9hS?usp=sharing](https://drive.google.com/drive/folders/13XL-9g-tn-Qb4aoH1I_fdc2kHWVTe9hS?usp=sharing)

Daha sonra *app.py* çalıştırılarak model için konfigürasyonların yüklenmesi gibi ilkendirme işlemleri yapılır ve server ayağa kaldırılır.

Bu işlemten sonra da *yzproje-ui* klasörü içinde konsol üzerinde *npm start* komutu çalıştırılarak arayüz de çalıştırılmış olur ve bu sayede uygulama kullanılmaya hazır hale gelir.

## 5. BAŞARIM DEĞERLENDİRİLMESİ

Dil modelimizi değerlendirmek için ilk olarak F1 score kullanmaya çalıştık. Ancak kullandığımız model transformer model olduğu için sürekli olarak farklı çıktılar sağladığı ve F1 score kullanabilmek için çıktılarının ortak tokenları olması gerektiği için sayısal başarı elde etmeye çalıştığımızda '*unseen token*' hata metniyle bir hata aldık. Daha sonra yaptığımız araştırmalar sonucunda sayısal başarıyı değerlendirebilmek için chatbot için BLEU score ile değerlendirmenin daha uygun olabileceğini ve bizim de BLEU score kullanabileceğimizi gördük ve sayısal başarıyı ölçebilmek için bunu kullandık. Ortalama BLEU score değerimiz aşağıda görüldüğü şekildedir;

Average BLEU Score: 0.0019700372

```
PS C:\Users\saity\Desktop\yzf1> & C:/Users/saity/AppData/Local/Microsoft/WindowsApps/python3.8.exe c:/Users/saity/Desktop/yzf1/main.py
C:\Users\saity\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Python3\site-packages\nltk\translate\bleu_score.py:552: UserWarning:
The hypothesis contains 0 counts of 2-gram overlaps.
Therefore the BLEU score evaluates to 0, independently of
how many N-gram overlaps of lower order it contains.
Consider using lower n-gram order or use SmoothingFunction()
  warnings.warn(_msg)
Average BLEU Score: 0.0019700372
```

## 6.SONUÇ

Doğal dil işleme ve veri bilimi alanındaki projemizde, sohbet botlarının nasıl çalıştığını inceleyerek bu konudaki kavramları ve dinamikleri öğrenmeye çalıştık. Aynı zamanda, derin öğrenme tekniklerini Python programlama dili kullanarak bir sohbet botu üzerinde uyguladık. Proje boyunca, veri setimizi belirli iş gereksinimlerine uygun şekilde özelleştirebilmeyi başardık. Veri içeriğimizi, Yapay Zeka konseptine dayanarak hazırladık. Çok miktarda veriyi manuel olarak girmek oldukça zaman alıcı bir süreçti, Bundan dolayı verileri sosyal medyadan alarak işledik. Chatbot'ların yaygın bir şekilde kullanıldığı ve birçok işletmenin iş süreçlerinde chatbot uygulamalarını dahil etmeye başladığı bir dönemden geçiyoruz. Bundan dolayı proje konusu aslında popüler bir konuydu. Artık elimizde Yapay Zeka konusunda cevap alabileceğimiz bir sohbet botu var!!!



## 7. Kaynaklar

[https://tr.wikipedia.org/wiki/Yapay\\_zek%C3%A2](https://tr.wikipedia.org/wiki/Yapay_zek%C3%A2)

<https://www.cbot.ai/tr-blog/chatbot-konusuna-genel-bir-bakis/>

<https://www.smartmessage.com/tr/chatbot-nedir-nasil-calisir-avantajlari-nelerdir/>

<https://medium.com/@akanesen/chatbot-nedir-ve-%C3%A7e%C5%9Fitleri-nelerdir-1b1ca0321e65#:~:text=G%C3%BCn%C3%BCm%C3%BCzde%20kullan%C4%B1lan%20chatbotlar%20kendi%20i%C3%A7lerinde,ve%20DD%C4%B0%20teknolojisi%20kullanan%20chatbottur.>

<https://sites.google.com/view/mfatihamasyali/yapay-zeka>