

# Doğal Dil İşleme Tabanlı Bir Sohbet Botunun Geliştirilmesi

Hasan DENİZHAN  
20110131815  
thedenizhan@hotmail.com

Batuhan EĞİN  
20110131053  
batuhanegin@icloud.com

**Özetçe:** Bu proje, kullanıcıların günlük yaşamlarını kolaylaştırmak amacıyla çeşitli işlevlere sahip bir chatbot geliştirmeyi hedeflemektedir. Chatbot, kullanıcıların sorularına yanıt vermek, önerilerde bulunmak ve eğlenceli etkileşimler sağlamak için doğal dil işleme tekniklerini kullanır. Sohbet botunun temel amacı, kullanıcıların doğal dildeki girdilerini anlamak, uygun yanıtları üretmek ve çeşitli alanlarda uygulamanın amacına göre şekillendirilebilir. Sohbet botunun geliştirilme süreci, kullanılan teknikler, model eğitimi ve sonuçlar ayrıntılı olarak ele alınacaktır.

**Anahtar Kelimeler:** Doğal Dil İşleme (NLP), Sohbet Botu, Derin Öğrenme, NeuralNet, LSTM, GRU, Transformer, KivyMD

## 1. Giriş

Günümüzde dijital dönüşüm, iş dünyasından günlük yaşamımıza kadar her alanda hızla ilerlemektedir. Bu dönüşümün en önemli unsurlarından biri de doğal dil işleme (NLP) teknolojileri ve bu teknolojilerin en yaygın uygulamalarından biri olan chatbotlardır. Chatbotlar, kullanıcılarla doğal dilde etkileşim kurabilen yazılım uygulamalarıdır ve müşteri hizmetlerinden eğitim sektörüne kadar geniş bir yelpazede kullanılmaktadır. Bu makalede, doğal dil işlemenin chatbotlar üzerindeki etkisi ve günlük yaşamda kullanımı kolaylaştıran gelişmiş bir chatbot projesi ele alınacaktır.

## 2. Sohbet Chatbotu: NEU

Chatbot, kullanıcılarla doğal ve akıcı sohbetler yapabilen bir dijital asistandır. Günlük sohbetlerde kullanılacak dil kalıplarını ve yaygın konuları anlayarak, kullanıcılarla anlamlı ve bağlamli diyaloglar kurar. Bu sayede kullanıcılar, botla samimi ve eğlenceli sohbetler gerçekleştirebilir. İşte bu chatbotun öne çıkan özellikleri:

- **Günlük Sohbetler:** Chatbot, kullanıcıların gün içindeki basit sorularına yanıt vererek, günlük sohbetleri daha interaktif hale getirir.
- **Bugün Ne Yemeliyim?:** Kullanıcıların beslenme tercihlerini ve diyet kısıtlamalarını göz önünde bulundurarak yemek önerilerinde bulunur.
- **Evet/Hayır Soruları:** Kullanıcıların evet veya hayır ile cevaplanabilecek basit sorularına hızlı ve net yanıtlar sunar.
- **Şaka Yapabilme:** Kullanıcılara moral vermek ve eğlendirmek amacıyla çeşitli şakalar yapabilir.
- **Masal Anlatabilme:** Özellikle çocuklar için uyku öncesi veya dinlenme anlarında masallar anlatabilir.
- **Ne Giyebileceğini Söyleme:** Hava durumu, kullanıcı tercihleri ve günlük planlar göz önünde bulundurularak giyim önerilerinde bulunur.
- **Planlarını Söyleme:** Kullanıcının günlük planlarını hatırlatarak, gününü daha iyi organize etmesine yardımcı olur.
- **Yazı-Tura Oyunu:** Kullanıcılarla eğlenceli vakit geçirmek için yazı-tura gibi basit oyunlar oynayabilir.

Bu özellikler sayesinde chatbot, kullanıcıların günlük yaşamlarını kolaylaştırmak ve daha keyifli hale getirmek için birçok işlev sunmaktadır.

### 3. Yöntem

#### 3.1 Veri Seti ve İşleme

Bu çalışmanın temelini oluşturan veri seti, çeşitli etkileşim senaryolarını kapsayan örneklerden oluşmaktadır. Bu veri seti, kullanıcıların yaygın olarak kullandığı ifadeler (pattern) ve bu ifadelere verilen yanıtları (response) içeren "intents.json" dosyası kullanılarak oluşturulmuştur. Veri setinin hazırlanması ve işlenmesi aşağıdaki adımlarla gerçekleştirilmiştir:

##### a. Veri Toplama ve Yapılandırma:

Veri seti, çeşitli kullanıcı senaryolarını temsil eden örnek ifadeler ve bunlara karşılık gelen yanıtları içermektedir. Örnekler, belirli niyetler (intents) etrafında gruplandırılmıştır. Her niyet, bir dizi kullanıcı ifadesi (pattern) ve bu ifadelere karşılık gelen yanıtları (response) içerir. Örneğin, selamlaşma niyeti (greeting) için "Hi", "Hello" gibi ifadeler ve bunlara karşılık gelen "Hey :-)", "Hello" gibi yanıtlar bulunmaktadır.

```
{
  "intents": [
    {
      "tag": "greeting",
      "patterns": ["Hi", "Hey", "Is anyone there?", "Hello", "Good day", "What's Up?"],
      "responses": ["Hey :-)", "Hello", "Hi there, What's Up?", "Hi there, I hope everything is good"]
    },
  ],
}
```

##### b. Tokenizasyon:

Kullanıcı ifadeleri, cümlelerin anlamlı parçalara (token) ayrılması işlemine tabi tutulmuştur. Bu işlem, cümlelerdeki kelimeleri, noktalama işaretlerini ve sayıları ayrı tokenlar olarak ayırmak için yapılır. Tokenizasyon işlemi, nltk kütüphanesi kullanılarak gerçekleştirilmiştir.

```
def tokenize(sentence):
    """
    Cümleyi kelimeler/dizeler haline ayır
    Bir dize bir kelime, noktalama işareti veya sayı olabilir.
    """
    return nltk.word_tokenize(sentence)
```

##### c. Kök Bulma (Stemming):

Tokenize edilmiş kelimeler, köklerine indirgenmiştir. Bu işlem, kelimelerin farklı biçimlerinin (örneğin, "running", "ran") aynı kök (örneğin, "run") altında birleştirilmesini sağlar. Kök bulma işlemi, nltk kütüphanesinin PorterStemmer sınıfı kullanılarak gerçekleştirilmiştir.

```
stemmer = PorterStemmer()
Explain | Test | Document | Fix | Ask
5 usages new *
def stem(word):
    """
    Kök bulma = kelimenin kök halini bulmak
    """
    return stemmer.stem(word.lower())
```

##### d. Sözcük Çantası (Bag of Words):

Kullanıcı ifadeleri, sözcük çantası (bag of words) modeli ile temsil edilmiştir. Bu model, cümlelerdeki kelimelerin varlığını veya yokluğunu belirten bir vektör oluşturur. Her bir kelime için 1 veya 0 değeri atanır. Sözcük çantası modeli, veri setindeki tüm benzersiz

```
def bag_of_words(tokenized_sentence, words):
    """
    Kelimelerin bulunduğu bir dizi döndür: cümlede var olan her
    bilinen kelime için 1, aksi takdirde 0
    """
    # Stem each word
    sentence_words = [stem(word) for word in tokenized_sentence]
    # Initialize bag with 0 for each word
    bag = np.zeros(len(words), dtype=np.float32)
    for idx, w in enumerate(words):
        if w in sentence_words:
            bag[idx] = 1
    return bag
```

kelimeleri içerir ve her bir kullanıcı ifadesi, bu kelimelerin varlığına göre bir vektörle temsil edilir.

<u>Training Data</u>							
<u>bag of words</u>							
all words							
["Hi", "How", "are", "you", "bye", "see", "later"]							
"Hi"	→	[ 1, 0, 0, 0, 0, 0, 0 ]	0 (greeting)				
"How are you?"	→	[ 0, 1, 1, 1, 0, 0, 0 ]					
"Bye"	→	[ 0, 0, 0, 0, 1, 0, 0 ]	1 (goodbye)				
"See you later"	→	[ 0, 0, 0, 1, 0, 1, 1 ]					
				<b>X</b>			
				<b>y</b>			

### 3.2 Veri Hazırlama:

İşlenmiş veri seti, eğitim ve test işlemleri için hazır hale getirilmiştir. Her kullanıcı ifadesi, sözcük çantası vektörüne dönüştürülmüş ve etiketlenmiştir. Etiketler, kullanıcı ifadelerinin hangi niyet (intent) grubuna ait olduğunu belirten sınıf etiketleridir.

### 3.3 Model Mimarisi ve Kullanımı

Bu çalışmada, dört farklı derin öğrenme modeli incelenmiştir: Çok Katmanlı Algılayıcı (MLP), Uzun Kısa Süreli Bellek (LSTM), Gated Recurrent Unit (GRU) ve Transformer tabanlı model. Her bir model, girdi cümleleri (bag of words) kullanarak kullanıcı girdilerini anlamak ve uygun yanıtları üretmek amacıyla tasarlanmıştır. Modellerin mimarisi ve kullanım adımları aşağıda ayrıntılı olarak açıklanmıştır.

- **NeuralNet (MLP) Modeli:** NeuralNet modeli, üç katmanlı bir yapay sinir ağıdır. Girdi katmanı, iki gizli katman ve bir çıktı katmanından oluşur. ReLU aktivasyon fonksiyonu kullanılarak doğrusal olmayanlık eklenir.
- **LSTM Modeli:** LSTM modeli, zaman serisi verilerini işleyebilme yeteneği ile bilinen bir RNN türüdür. Bu model, girdileri işleyerek anlamlı bağlam bilgisi oluşturur ve çıktıyı üretir.
- **GRU Modeli:** GRU modeli, LSTM modeline benzer ancak daha az karmaşık bir yapıya sahiptir. Bellek hücrelerini daha verimli kullanarak daha hızlı öğrenme sağlar.
- **Transformer Modeli:** Transformer modeli, dikkat mekanizmaları kullanarak uzun bağlam bilgilerini öğrenmede etkilidir. Bu model, girdileri kodlayarak anlamlı temsiller oluşturur ve çıktıyı üretir.

### 3.4 Model Eğitimi

Modeller, 5 katlı çapraz doğrulama (K-Fold Cross-Validation) yöntemi kullanılarak eğitilmiştir. Eğitim sürecinde, veri seti eğitim ve test alt kümelerine ayrılmış, her model farklı katlarda eğitilmiş ve doğrulanmıştır. Eğitim sürecinde Adam optimizasyon algoritması ve çapraz entropi kaybı fonksiyonu kullanılmıştır.

### Hiperparametreler:

- Epoch sayısı: 2000
- Mini-batch boyutu: 32
- Öğrenme hızı: 0.0001
- Gizli katman boyutu: 128

### 3.5 Model Performansının Değerlendirilmesi

Eğitim ve doğrulama sürecinde, modellerin performansı doğruluk oranları ile değerlendirilmiştir. 5 katlı çapraz doğrulama sonucunda, her modelin ortalama doğruluk oranları hesaplanmıştır. Modellerin performansı karşılaştırılarak, en iyi sonuç veren model belirlenmiştir.

### 3.6 Sonuçlar ve Tartışma

Eğitim ve doğrulama sürecinde elde edilen sonuçlar, modellerin performansını değerlendirmek için kullanılmıştır. 5 katlı çapraz doğrulama sonucunda, her modelin ortalama doğruluk oranları aşağıdaki gibi elde edilmiştir:

- SimpleNN: %92.00
- LSTM: %92.00
- GRU: %88.00
- Transformer: %92.00

Bu karşılaştırma sonuçlarına göre, SimpleNN modeli seçilmiştir. SimpleNN modeli, LSTM ve Transformer modelleriyle aynı doğruluk oranına sahip olup, daha basit ve hızlı eğitim avantajı nedeniyle tercih edilmiştir.

## 4. Uygulama

Geliştirilen sohbet botu, bir KivyMD arayüzü kullanılarak kullanıcılarla etkileşime geçebilen bir uygulama olarak sunulmuştur. Bu uygulama, KivyMD kullanılarak bir masaüstü uygulaması şeklinde tasarlanmıştır. Bu uygulama, kullanıcılara günlük sohbetler, bilgi sağlama ve eğlence gibi çeşitli hizmetler sunmaktadır. Aşağıda uygulamanın bazı önemli özellikleri açıklanmıştır:

- **Kullanıcı Arayüzü:** Basit ve kullanıcı dostu bir arayüze sahiptir. Kullanıcılar, metin alanına mesaj yazıp "Send" butonuna tıklayarak sohbet başlatabilir.
- **Sohbet Geçmişi:** Kullanıcı ve bot arasındaki sohbet geçmişi kaydedilir ve ekranda gösterilir.
- **Yanıt Üretme:** Kullanıcının girdiği metin, önceden eğitilmiş model tarafından işlenir ve uygun yanıt üretilir.

## 5. Sonuç

Bu çalışma, doğal dil işleme teknikleri ve derin öğrenme modelleri kullanarak kullanıcılarla anlamlı sohbetler gerçekleştirebilen bir sohbet botu geliştirilmesini ele almıştır. Farklı model mimarileri incelenmiş, performansları karşılaştırılmış ve en iyi sonuç veren model olarak SimpleNN modeli belirlenmiştir. Belirlenen model ile botumuz eğitilmiş, eğitilen bot için bir masaüstü uygulaması tasarlanmıştır. Bu uygulama ile beraber kullanıcılara kolay bir erişim sağlanmıştır.