

## Sıradan Sıraya LSTM Modeli ile Telegram Bot Uygulaması

Ali Hakan IŞIK<sup>\*,a</sup>, Ayşenur YAĞCI<sup>b</sup>

<sup>a,\*</sup> Burdur Mehmet Akif Ersoy Üniversitesi Bilgisayar Mühendisliği Bölümü, ORCID:0000-0003-3561-9375, 15300, BURDUR, TÜRKİYE

<sup>b</sup> Burdur Mehmet Akif Ersoy Üniversitesi Bilgisayar Mühendisliği Bölümü, ORCID:000-0001-8664-8067, 15300, BURDUR, TÜRKİYE

### MAKALE BİLGİSİ

Alınma: 23.02.2020  
Kabul: 25.04.2020

#### **Anahtar Kelimeler:**

seq2seq,  
chatbot,  
telegram,  
LSTM

#### **\* Sorumlu Yazar:**

e-posta:  
ahakan@mehmetakif.e  
du.tr

### ÖZET

Sohbet botu (Chatbot), işitsel veya metinsel yöntemlerle kullanıcı ile sohbet eden bir yazılımdır. Gelişmiş sohbet botları, ilgili konuya uygun cevaplar verebilmektedir. Sohbet botlarında yapay zeka yöntemlerinin kullanılması etkinliğini artırmaktadır. Bu kapsamda, çalışmada LSTM (Uzun Kısa Süreli Bellek) ve seq2seq modeli ile telegram bot uygulaması geliştirilmiştir. Çalışmada LSTM kullanılması bir sonraki konuşma eylemini tahmin etmek için konuşma geçmişine geri dönülebilmesini sağlamıştır. Çalışmada kaggle.com'dan alınan chatterbot veri kümesi kullanılarak zaman tasarrufu sağlanmıştır. Kullanıcı etkileşimi için pythonanywhere üzerinden telegram ile entegrasyon yapılmıştır. Çalışmanın eğitimi sırasındaki kayıp oranı ve diğer performans parametreleri TensorBoard ile görselleştirilmiştir. Çalışmada 50 adımlık eğitim, 13 saniyede tamamlanmıştır. Çalışmadaki kayıp oranı her adımda azalarak 50 adım sonunda 0.2772'ye düşmüş ve yüzde 79 doğruluk oranı elde edilmiştir. Çalışma, modüler ve geliştirmeye açık bir şekilde tasarlanmıştır. Modüler yapıdaki çalışmanın eğitim süreci devam edilmesi sağlanarak farklı dilsel ifadelerin öğretilmesi sağlanabilecektir. Çalışmada, açık kaynak kodlu ve ücretsiz yazılımlar kullanılmıştır. Sunulan çalışma, güncel teknolojilerin kullanıldığı literatürde öne çıkan çalışmaların özelliklerinin biraraya getirilmesini sağlamıştır.

<https://dx.doi.org/10.30855/gmbd.2020.01.03>

## Telegram Bot Application with Sequence to Sequence LSTM Model

### ARTICLE INFO

Received: 23.02.2020  
Accepted: 25.04.2020

#### **Keywords:**

seq2seq,  
chatbot,  
telegram,  
LSTM

#### **\* Corresponding**

#### **Authors**

ahakan@mehmetakif.e  
du.tr

### ABSTRACT

Chatbot is a software that chat with the user by audio or textual methods. Advanced chat bots are able to provide appropriate answers related issues. Using artificial intelligence methods in chat bots increases efficiency. In this context, telegram bot application was developed with LSTM (Long Short Term Memory) and seq2seq model. The use of LSTM in the study enabled the return to speech history to predict the next speech action. Time saving was achieved by using chatterbot dataset from kaggle.com in the study. Telegram integration is provided via pythonanywhere for user interaction. The loss rate and other performance parameters during the training of the study were visualized with TensorBoard. In the study, the 50-step training was completed in 13 seconds. The loss rate in the study decreased at each step and decreased to 0.2772 at the end of 50 steps and 79 percent accuracy rate was obtained. The study is designed modularly and open to development. By continuing the education process of the modular study, it will be possible to teach different linguistic expressions. Open source and free software were used in the study. The presented study has brought together the features of the outstanding studies in the literature using state of art technologies.

<https://dx.doi.org/10.30855/gmbd.2020.01.03>

### 1. INTRODUCTION (GİRİŞ)

Sohbet botu (Chatbot), işitsel veya metinsel yöntemlerle kullanıcı ile sohbet eden bir uygulama yazılımdır. Sohbet botlarının ilk

kavramsallaştırılması, 1950 yılında Alan Turing'in "makinelere düşünebilir mi?" sorusu ile başlamıştır. Sonrasında ise zamanla sohbet botu teknolojisi, doğal dil işleme ve makine öğrenmedeki gelişmelerle

birlikte günümüzdeki şeklini almıştır. Aynı şekilde, sohbet botu kullanımı, özellikle Facebook, Kik, Slack, Skype, WeChat, Line ve Telegram tarafından devam etmiş ve yaygın hale gelmiştir [1].

Sohbet botları genellikle diyalog sistemleri, müşteri hizmetleri, iş görüşmeleri veya bilgi edinme dâhil olmak üzere farklı amaçlar için kullanılmaktadır. Sıklıkla kullanılan iki tür sohbet botu bulunmaktadır. İlk sohbet botları, konuyu anlamaya çalışan, aynı konu hakkında her soru için daima aynı cevabı veren sohbet botlarıdır. Temel olarak, bu türden sohbet botları cevap için sınıflandırma algoritmalarını kullanmaktadır. Bu bot uygulamalarında, N tane konu ve N tane cevabın bir listesi bulunmaktadır. Uygulamada, sınıflandırılan konunun olasılığı düşükse kullanıcıdan daha özel ifade kullanmaları ve soruyu tekrar etmeleri istenmektedir. İkinci tür sohbet botları, daha gelişmiş, güncel teknolojileri kullanan ve aynı zamanda daha karmaşıktır. Bu botlardaki cevaplar, sıklıkla RNN (Tekrarlayan Sinir Ağı) kullanılarak oluşturulmaktadır. Bu sohbet botları daha kişiselleştirilmiş ve uygun cevaplar verebilmektedir. Özel bir RNN türü olan Uzun Kısa Süreli Bellek (LSTM) kullanıcının sorularını daha iyi anlayabilmekte ve mümkün olan en iyi cevabı vermektedir. Geleneksel seq2seq sohbet bot modelleri, çıktı cümlelerinin duyarlılığını düşürmeden sadece girdi dizilerinde şartlandırılmış en yüksek olasılıklara sahip cümleleri bulmaya çalışmaktadır [2, 3].

İlgili bilgiler ile ihtiyaç duyulan cevap arasındaki farkın az olduğu durumlarda, RNN'ler geçmiş bilgileri kullanmayı öğrenebilmektedir. Tahmin edilmesi gereken kelime sayısı arttıkça, RNN'ler bilgileri birleştirmeyi öğrenememektedir. LSTM'ler ise bu soruna çözüm olarak bilginin uzun süre hatırlanması için tasarlanmıştır [2]. Çalışmada bilginin uzun süre hafızada kalması ve en yüksek olasılığa sahip olan en uygun cümlenin yerleştirilmesi için LSTM ve seq2seq TensorFlow Kütüphanesi kullanılarak bir sohbet botu tasarımı yapılmıştır.

## 2. İLGİLİ ÇALIŞMALAR (RELATED WORKS)

Yeni ortaya çıkan ve güncel bir teknoloji olduğundan, literatürde sohbet botu alanında kısıtlı çalışma bulunmaktadır. Çalışmada, bu kapsamda öne çıkan akademik çalışmalar sunulmuştur. Anbang Xu ve ark. tarafından yapılan “Sosyal Medyada Müşteri Hizmetleri İçin Yeni Bir sohbet botu” adlı çalışmada, derin öğrenme ile sosyal medyadan kullanıcıların

isteklerine otomatik olarak yanıtlar oluşturmak için yeni bir konuşma sistemi oluşturulmuştur. Değerlendirmede, taleplerin yüzde kırktan fazlasının duygusal olduğu, sistemin kullanıcıların duygusal durumlarla başa çıkmalarına yardımcı olmak için empati göstermede insansı ajanlar (human agents) kadar iyi olduğu sonucu elde edilmiştir [4].

Diğer bir çalışma ise Ming-Hsiang Su ve ark. tarafından yapılan yaşlı bakımı için LSTM mimarisi ile sohbet botu uygulamasıdır. Yaşlılar için böyle bir sistemin diğer sistemlerden daha gerekli olduğu düşünülerek tasarlanmıştır. Çalışmada kullanılan veri kümesi, yaşlılarla yapılan günlük konuşmalardan derlenmiştir. Aynı zamanda yaşlılarla sohbet ederken kelimeler ve cümleler arasındaki anlamsal bilgiyi bir seferde birden fazla cümle ile çıkarmak için LSTM modeli ve çok katmanlı bir gömme modeli uygulanmıştır. Uygun bir soru kalıbı seçmek için öklid mesafesi kullanılmıştır. Önerilen model cevap seçiminde yüzde 79,96 doğrulukla çalışmıştır [5].

“Telegram sohbet botu kullanarak yapay zeka ile IoT tabanlı akıllı çalışma alanı geliştirilmesi” adlı çalışmada, IoT teknolojisiyle sohbet botu teknolojisi birleştirilmiştir. Uygulama, çalışma alanlarındaki elektronik cihazları denetlemektir. Uygulama, internet üzerinden lamba, fan, çekmece kilidi, elektrik prizi ve sıcaklık kontrolü gibi işlevler gerçekleştirmektedir. Bu çalışmada, Telegram Messenger kullanılarak yapay zekâ sohbet botu yardımı ile yapılan kontrolü sağlamak için Esp 8266 ile uzaktan kumanda cihazı kullanılmıştır. Bu çalışma, çalışanların çalışma alanlarındaki elektronik cihazları, akıllı telefonlar veya bilgisayarlar üzerinden kontrol etmesini kolaylaştırmak için yapılmıştır [6]. Literatürdeki bu ve benzeri çalışmalardan da anlaşıldığı gibi sohbet botu, uygulamalarda kolay arayüzüyle bağlantı kolaylığı sağlamasından dolayı tercih edilen bir yazılımdır.

“RNN-LSTM kullanarak otomatik Tayland-SSS sohbet botu” isimli çalışmada belirtildiği üzere e-posta veya canlı sohbet gibi çevrimiçi müşteri hizmetleri bulunan e-ticaret modelinde, müşteriler çoğunlukla hızlı ve kolay kullanılabilir olduğu için canlı sohbeti kullanmaktadır. Bu ihtiyaçla birlikte, yöneticilerin cevap yazmak için ilave zaman harcaması gerekmekte ve müşterilerin cevapları beklemesi oldukça uzun sürmektedir. Bu soruna çözüm olarak geliştirilen çalışmada, metin sınıflandırma için Uzun Kısa Süreli Bellek (LSTM) modeliyle Tekrarlayan Sinir Ağı (RNN) kullanılarak müşterilere otomatik olarak yanıt veren bir sıkça

sorulan sorular (SSS) sohbet botu geliştirilmiştir. Elde edilen sonuçlarda sohbet botunun soruların % 86,36'sını algılayabildiği ve % 93,2 doğrulukla cevaplanabildiği gösterilmiştir [7]. Diğer benzer çalışmalarda da LSTM ile yapılan sohbet botların daha kısa zamanda daha iyi tahminler çıkardığı ve soruları daha yüksek olasılıkla algıladığı anlaşılmıştır.

Sunulan çalışmada, Telegram sohbet botu kolay arayüzüyle bağlantı kolaylığı sağlaması ve geliştirme kapsamında ücretsiz olduğu için tercih edilmiştir. Ayrıca çalışma, güncel teknolojilerin kullanıldığı literatürde öne çıkan çalışmaların özelliklerini biraraya getirilmesini sağlamıştır.

### 3. MATERYAL VE YÖNTEMLER (MATERIALS AND METHODS)

#### 3.1. Sıradan Sıraya Modeli (Sequence to Sequence Model)

Dizi modellerine göre dizilim, RNN mimarisine dayanmakta olup, iki adet RNN'den oluşmaktadır. Bunlar, kodlayıcı ve kod çözücüdür. Kodlayıcının görevi girdiyi, kod çözücünün görevi ise çıktıyı işlemektir. Kodlayıcı giriş olarak bir dizi (cümle) almakta ve her zaman adımında bir sembol (kelime) işlemektedir. Kodlayıcın amacı, gereksiz bilgileri kaybederken sıradaki sadece önemli bilgileri kodlayan bir sembol dizisini sabit boyutlu bir özellik vektörüne dönüştürmektir. Seq2seq modeli, bir kodlayıcı düğümüne karşılık çıktı üreten bir kod çözücü düğümüdür. Bu model, makine çevirisinde basit bir uygulamaya sahip olup, bir seferde yalnızca bir giriş kelimesine bakarak kolayca kod çözücü tarafından oluşturulabilmektedir [2]. Modelde her gizli durum bir sonraki gizli durumu etkilemekte ve son gizli durum dizinin özeti olarak görülmektedir. Bu duruma, dizinin amacını temsil ettiği için bağlam veya düşünce vektörü denilmektedir. Kod çözücü, bağlamdan bir seferde bir sembol (sözcük) olan başka bir sekans üretmektedir. Burada, her bir zaman adımında, kod çözücü içerikten ve daha önce oluşturulan sembollerden etkilenmektedir. Bu modeli kullanırken oluşan sorunlardan biri de modelin değişken uzunluktaki dizileri kullanamamasıdır. Bütün dizi-dizi uygulamaları değişken uzunluklu dizileri içermektedir [8]. Bu sorun, eğitimden önce sabit uzunluk dizileri içine değişken uzunlukta dizileri dönüştürmek için veri kümesi üzerinde padding işlemi yapılarak çözülmüştür.

#### 3.1 Kelime Gömme (Word Embedding)

Kelime gömme, düşük boyutlu bir vektör uzayda kelimelerin yoğun temsilini öğrenmek için bir tekniktir. Her kelime, sabit uzunluktaki bir vektörle

temsil edilen bu alanda bir nokta olarak görülebilmektedir. Kelimeler arasındaki anlamsal ilişkiler bu teknikle ortaya çıkarılmaktadır. Sözcük gömme tipik olarak ağırlık katmanında yapılmaktadır. Sözcük gömme, bir sözcükten sözcük büyüklüğüne göre yoğun bir vektöre sözcük (sözcük dizisinde sözcük dizini) eşleyen katıştırma katmanıdır. Seq2seq modelinde, gömme katmanının ağırlıkları, modelin diğer parametreleri ile birlikte eğitilmektedir [8, 9].

#### 3.2 Dikkat Mekanizması (Attention Mechanism)

Seq2seq çerçevesinin sınırlamalarından biri, giriş cümlesindeki tüm bilgilerin, sabit uzunluklu bir vektör bağlamında kodlanması gerektiğidir. Dizinin uzunluğu büyüdükçe, önemli miktarda bilgi kaybedilmektedir. Bu, temel seq2seq modelinin büyük dizilerin kodunu çözmede iyi çalışmamasının nedenidir. Dikkat mekanizması, hizala ve çevirmeyi ortak olarak öğrenerek yapan sinirsel makine çevirisinin, kod çözücünün kod çözme sırasında giriş sırasına seçici olarak bakmasını sağlamaktadır. Bu, girilen her faydalı bilgiyi kodlamak için kodlayıcıya baskı uygulamaktadır. Kod çözücünün her bir zaman aşımı süresi boyunca, sabit bir bağlam (kodlayıcının son gizli durumu) kullanmak yerine, farklı bir bağlam vektörü  $c_i$ , sözcüğü(yi) oluşturmak için kullanılmaktadır.

$$c_i = \sum_{j=1}^n a_{ij} h_j \quad (1)$$

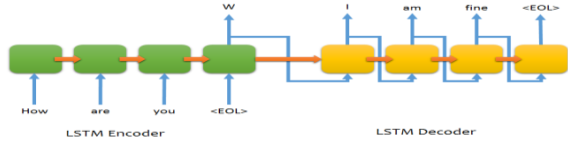
Bir numaralı denklemdeki formülde, vektör  $c_i$ , temel olarak kodlayıcının gizli durumlarının ağırlıklı toplamıdır.  $n$ , giriş dizisinin uzunluğu olan  $h_j$ ,  $j$ ' nin zaman adımındaki gizli durumudur. Kodlayıcıdaki her gizli durum, cümlenin bu bölümündeki yerel içerikle ilgili bilgileri kodlamaktadır. Veriler sıfır kelimesinden  $n$  kelimesine aktıkça, bu yerel içerik bilgisi azalmaktadır. Bu durum, kod çözücünün yerel bağlamları bilmesi için kod çözücünün kodlayıcıdan zirve yapmasını gerekli hale getirmektedir [6].

Giriş dizisinin farklı bölümleri, çıkış dizisinin farklı bölümlerini oluşturmak için gerekli bilgileri içermektedir. Başka bir deyişle, çıkış sırasındaki her kelime giriş sırasının farklı kısımlarına hizalanmaktadır.

Seq2seq hizalama modeli  $i$  pozisyonundaki çıkışın  $j$  pozisyonundaki girdilerle ne kadar iyi eşleştiğinin ölçülmesini sağlamaktadır. Her sözcüğü çıkış dizisinde üretmek için giriş bağlamlarının (gizli durumlar) ağırlıklı bir toplamı alınmaktadır [1,2].

### 3.3 LSTM Modeli (LSTM Model)

Uzun Kısa Süreli Bellek (LSTM), basit RNN' nin giriş ve çıkış kapılarına ek olarak, özel unutkan kapıları (forget gates) olan özel bir RNN türüdür. Şekil 1 de örnek bir LSTM mimarisi sunulmuştur.



Şekil 1. LSTM mimarisi (LSTM architecture) [11]

LSTM'ler, giriş durumunu bir RNN'den daha uzun süre hatırlamak üzere tasarlanmıştır. Bu nedenle uzun dizilerin doğru bir şekilde işlenmesini sağlamaktadır. LSTM'ler, NLP mimarisinin temel bir parçasıdır [8].

Çalışmadaki sohbet botu uygulamasında LSTM ve sıradan sıraya (sequence to sequence) modeli kullanılmıştır. Veri içindeki kelimeler glove yöntemi ile gömülmüştür. GloVe kelime gösterimi için global vektörlerin kısaltmasıdır. Stanford tarafından küresel bir kelime-kelime eş-oluşum matrisini bir korpustan toplayarak kelime gömme oluşturmak için geliştirilen denetimsiz bir öğrenme algoritmasıdır. Uygulamada kullanıcıdan gelen cevaplar kelime gömme yardımıyla gereksiz kelimelerden arındırılarak bellekte tutulmakta ve ona en uygun cevap verilmesi sağlanmaktadır [12].

### 3.4 Veri Kümesi (Data Set)

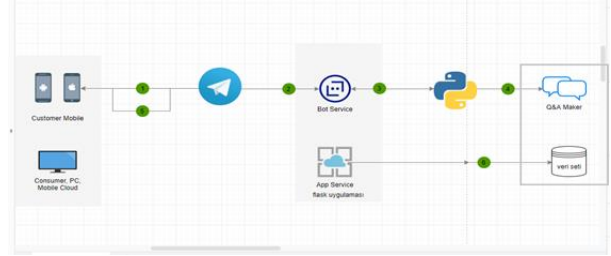
Çalışma, Kaggle.com'da bulunan chatterbot/english veri kümesi ile gerçekleştirilmiştir. Sohbet botunu eğitip test etmek için kullanılan veri kümesinde, yiyecek, tarih, para, sağlık, psikoloji, spor, bilim, literatür, duygu, bilgisayar ve yapay zeka gibi birçok konuyu baz alan diyaloglar bulunmaktadır [13].

### 3.5 Veri Ön İşleme (Data Preprocessing)

Çalışmadaki ön işleme aşamasında, seq2seq modelini eğitmeden önce daha az veri ile verimli eğitim için ön işleme adımı uygulanmıştır. Ön işleme ile verileri normalleştirmek, kelime boyutunu küçültmek, ham metni eyleme dönüştürülebilir değerlere dönüştürmek, gereksiz verileri kaldırmak amaçlanmıştır. Çalışmada ayrıca TensorFlow, keras, numpy, pickle ve ayrıca modellerinin tanımlanmasına yardımcı olan diğer modüller de kullanılmıştır[14].

### 3.6. Sohbet Botu Mimarisi (Chatbot Architecture)

Çalışmadaki Telegram entegrasyonu ile, kullanıcılardan aldığı sorulara LSTM modeli ile öğrenerek farklı cevaplar veren bir sohbet botu uygulaması tasarlanmıştır. Şekil 2 de tasarlanan sistemin mimarisi sunulmuştur.



Şekil 2. Sohbet botu mimarisi (Chatbot architecture)

Sohbet botu mimarisinin temel bileşenleri aşağıda verilmiştir. Bunlar:

- Environment (Ortam)
- Eklentiler, bileşenler
- Soru-cevap sistemi
- Web app

Ortam kısmında, temel doğal dil işleme (NLP) motoru ve bağlam yorumlama işlevi gerçekleştirilmektedir. NLP motoru, kullanıcıların herhangi bir zamanda söylediklerini yorumlayan ve dili sistemin daha fazla işleyebileceği yapılandırılmış girdilere dönüştüren temel bileşendir. Bu kısımda python programlama dili ve LSTM modeli kullanılmıştır. LSTM modeliyle kullanıcıdan alınan sorulardan kodlayıcı-kod çözücü modeline göre cevap çıktılar üretilmiştir.

Soru cevap sistemi, Q-A maker ve veri kümesi bileşenlerinden oluşmaktadır. Soru-cevap sisteminde kullanıcılardan gelen sorulara veri kümesinden işlenmiş uygun yanıtlar ile cevaplar verilmiştir. Veri ön işleme ile gereksiz veriler ayıklanmıştır. Eğitim işlemi sonrasında ise kodlayıcı-kod çözücü modeline göre soru cevap çiftleri oluşturulmuştur. İşlenen veri uygulamaya aktarılmıştır. Daha sonra soru cevap yapıcından geçmekte ve kullanıcıya cevap iletilmektedir.

Eklentiler ve bileşenler kısmında telegram api kullanılmıştır. Arayüz kısmı telegram uygulamasıyla sağlanmıştır. PC ya da mobilden indirilen uygulama aracılığı ile telegram bota giden sorular webhook sayesinde bot servisinde işlenmekte ve python ile yazılan kod ile bağlantısı sağlanmaktadır.

Pythonanywhere, bulutta bir web sitesi geliştirilmesine ve barındırılmasına izin verdiği için

Flask web sunucusu bulut üzerinde pythonanywhere üzerine yerleştirilmiştir. Aynı zamanda python anywhere sohbet botu bağlantısı ve arka uç oluşturmada kolaylık sağlamıştır. Bu aşamada ilk olarak Pythonanywhere üzerinde Python ve Flask kullanarak bir WebApp (webHook) çalıştırılmıştır. Yeni bir WebApp ekledikten sonra web uygulaması için oluşturulan bağlantı girilmiştir. Bot sunulan işlemlerden sonra telegram uygulamasında çalıştırılabilir hale gelmiştir.

### 3.6.1 Kodlayıcı-Kod çözücü modeli(Gecikmeli Türev Eşitliği Tabanlı Kaotik Sistem Modeli)

Çalışmadaki model; gömme, LSTM ve yoğun katmanlara sahip olacak biçimde yapılmıştır. Çalışmadaki veri, 200 farklı boyutta gömülü kelimelerden oluşmaktadır. Temel yapılandırma aşağıdaki gibidir.

- 2 Giriş Katmanı: Biri kodlayıcı giriş verisi (encoder\_input\_data) ve diğeri kod çözücü giriş verisi ( decoder\_input\_data) dir.
- Katman gömme: Boyutlu yoğun vektörlerini düzeltip simge vektörlerine dönüştürmek için kullanılmıştır.
- LSTM katmanı: Uzun Kısa Süreli hücrelere erişim sağlamaktadır.

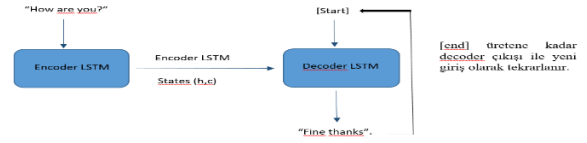
Modelin Çalışma yapısı aşağıda sunulmuştur:

- 1) kodlayıcı\_giriş\_veri, gömme (embedding) katmanından gelmektedir( encoder\_embedding).
- 2) Gömme katmanının çıkışı, 2 durum vektörü üreten (h ve c encoder\_states olan)LSTM hücresine gitmektedir.
- 3) Bu durumlar, kod çözücünün LSTM hücresinde ayarlanmıştır.
- 4) Kod çözücü\_giriş\_veri, gömme katmanından gelmektedir.

Çalışmadaki model, RMSprop optimizasyon ve kategorik crossentropy kayıp fonksiyonu ile adım(epoch) adım eğitilmiştir.

### 3.6.2. Çıkarım modellerinin tanımlanması (Identification of inference models)

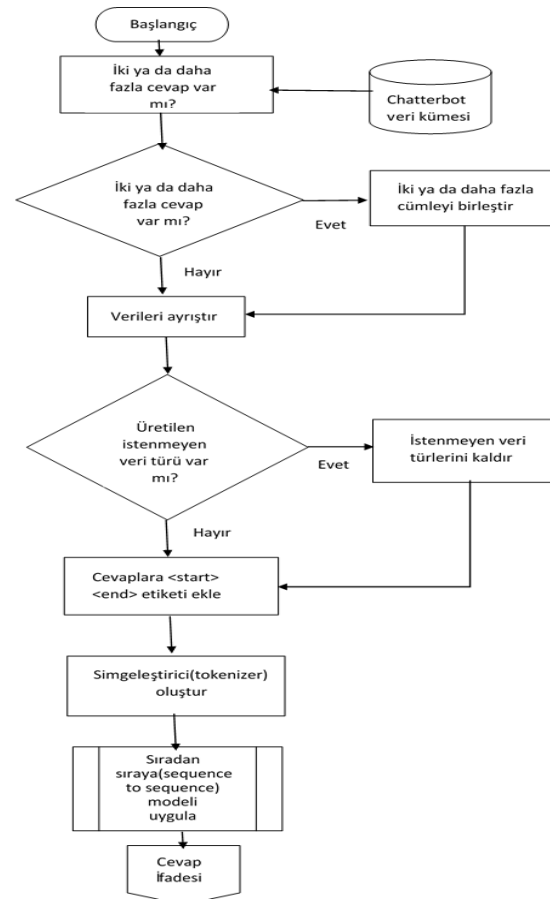
Bu aşamada cevapları tahmin etmede yardımcı olan çıkarım modelleri oluşturulmaktadır. Kodlayıcı çıkarım modeli, soruyu girdi olarak almakta ve LSTM durumlarını çıkarmaktadır ( h ve c).



Şekil 3. Kodlayıcı kod çözücü modelinin yapısı (Structure of the encoder decoder model)

Kod çözücü çıkarım modeli: 2 giriş almakta, birincisi LSTM durumudur (kodlayıcı modelinin çıkışı), ikincisi ise cevap girişi sıralarıdır (<start> etiketine sahip olmayanlar).

Çıkarım modeli olan kodlayıcı-kod çözücü modelinin yapısı şekil 3 de gösterilmiştir. Sistem Akış diyagramı ise şekil 4 de sunulmuştur.



Şekil 4. Sistem Akış Diyagramı (System FlowChart)

### 3.7 Flask Bağlantısı (Flask Connection)

Flask bağlantısı için yönlendirici port yönlendirme ile yapılandırılmadığı sürece uygulamalar ağından dışarıdan erişilememektedir. Bir yönlendiriciyi yönlendirmek bağlantı noktası ve kullanılan yönlendirici modeline bağlı olduğundan bu karmaşık bir işlem olabilmektedir. Bu nedenle, zaman kaybettiren konfigürasyonlardan kaçınmak için, bir

Flask web sunucusu, bulut üzerinde pythonanywhere üzerine yerleştirilmiştir.

Pythonanywhere, bulutta bir web sitesi geliştirilmesine ve barındırılmasına izin vermektedir. Çalışmada sohbet botu bağlantısı ve arka uç oluşturmada kolaylık sağlayan bir platform olan pythonanywhere kullanılmıştır. Bu aşamada ilk olarak Pythonanywhere üzerinde Python ve Flask kullanarak bir WebApp (webHook) çalıştırılmıştır. [15] Yeni bir WebApp ekledikten sonra web uygulaması için bir örnek girilmiştir. Daha sonra Python 3.5' te bir çerçeve olarak Flask seçilmiştir.

WebApp dosya dizininde varsayılan flask\_app.py dosyasına bağlantı için gerekli kod yazılmıştır. Python anywhere Bash konsoluna kurulum paketleri yerleştirilmiştir.

Daha sonra Flask uygulamasına ping atıp bağlantı sağlanmaktadır. Bu aşamada, pythonanywhere üzerine bir Python Flask uygulaması yapılmıştır. Bu bir sunucu oluşturmadan bir WebApp barındırılmasına izin vermektedir. WebApp, gelen mesajlar, chat-bot arayüzü ve giden cevaplar arasında bir aracı işlev sağlaması için kullanılmıştır[15].

### 3.8. Telegram Entegrasyonu (Telegram Integration)

Bot API, Telegram ile bot inşa etmek için geliştiriciler için oluşturulan HTTP tabanlı bir ara yüzdür. Her bot oluşturulduğunda benzersiz bir kimlik doğrulama belirteci verilmektedir. Yanıt, her zaman bir Boole alanına sahip olan ve sonucun insan tarafından okunabilir bir açıklamasını içeren isteğe bağlı bir dizi alanı olan bir JSON nesnesi içermektedir. Sunucularda çalışan yazılıma kullanıcılar tarafından mesajlar, komutlar ve istekler iletilmektedir. Bütün şifreleme ve iletişim, aracı sunucu olan Telegram API ile yönetilmektedir. Telegram API'sinin basit bir HTTPS arayüzü ile bu sunucuyla iletişim kurması sağlanmaktadır. Bu arayüz Bot API olarak adlandırılmaktadır [16]. Özetle, Telegram bir sohbet botu arayüzü oluşturmak için kolay bir yol sunmaktadır. Telegram App, arka ucuna bağlanmak ve sohbet botu uygulamasını çalıştırmak için kullanılan bir erişim belirteci sağlamaktadır.

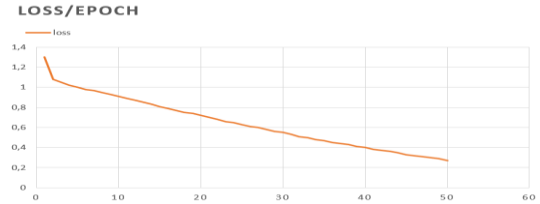


Şekil 5. Telegram bot ile yapılan sohbetin çıktısı (Output of the chat with Telegram bot)

Yapılan çalışmalardan sonra bot test edilebilir hale gelmiştir. Şekil 5 de bulunan ekran görüntüsünde geliştirilen sohbet botunun diyalogları gösterilmiştir. Böylece uygulama ile telegram entegrasyonu sağlanmıştır.

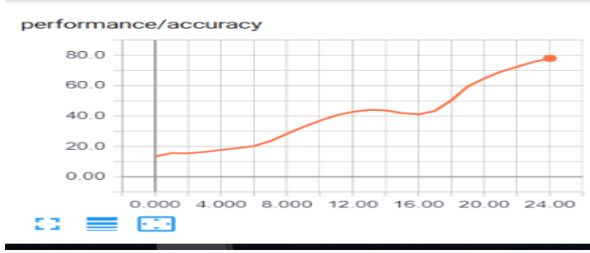
### 5. ANALİZ VE BULGULAR (ANALYSIS AND FINDINGS)

Sohbet botu uygulamalarında eğitim verilerindeki değişiklik, kullanıcıya verilen yanıtı doğrudan etkileyebilmektedir. Tüm öğrenme algoritmaları, kayıp düz bir satıra ulaşıncaya kadar hesaplamaları birçok kez tekrarlamaktadır. Kayıp işlevini en aza indirmek için bir öğrenme oranı tanımlanması gerekmektedir. Bu modelin öğrenilmesi istenilen hızıdır. Amaç, kayıp fonksiyonunu en aza indirmektir. Böylece hata oranı da en aza indirgenmiş olmaktadır. Uygulamanın eğitimi sırasındaki kayıp oranının değişim çıktısı takip edilerek gerçekleştirilmiştir. TensorBoard, bu ölçümleri görselleştirmek ve olası sorunları vurgulamak için bir araç olarak kullanılmıştır. TensorBoard metrikleri çok sık güncellenmektedir. Bu durumda, modelin doğru çalışıp çalışmadığını görmek için sonuna kadar beklenilmesi gerekmemektedir. Bu nedenlerden dolayı çalışmada TensorBoard kullanılmıştır.



Şekil 6. LSTM loss fonksiyon grafiği (LSTM loss function graph)

Şekil 6 de bulunan grafikte, eğri düzleşinceye kadar yinelemenin kaybının azaldığını, yani modelin işlevi gerçekleştirdiğini ve hata oranının azaldığını gösterilmiştir.



Şekil 7. Doğruluk grafiği (Accuracy chart)

Sohbet botu uygulamasında 50 adımlık eğitim, 13s de tamamlanmıştır. Kayıp oranı her adımda azalarak 50.adımda 0.2772' ye gerilemiştir. Doğruluk oranı ise %79 çıkmıştır. Test sonucu kontrolü için yapılan tekrarlar ve sonuç Şekil 7 de sunulan grafikte gösterilmiştir.

## 6. SONUÇ (CONCLUSION)

Sohbet botu çalışmasında, LSTM ve sıradan sıraya dizi modeli ile telegram bot uygulaması geliştirilmiştir. Çalışmanın, iş görüşmeleri öncesi adaylar için ön değerlendirme aşamasında kullanılması planlanmıştır. Çalışmada, veri kümesi olarak 1984 soru-cevap çiftinden oluşan “chatterbot” veri kümesi kullanılmıştır. Çalışma ile, chatterbot veri kümesinden alınan verilerle kullanıcının sorduğu sorulara uygun cevaplar veren kendini tekrar etmeyen bir sohbet botu tasarımı yapılmıştır. Çalışma, RMSprop optimize edici ve categorical\_crossentropy kayıp fonksiyonu ile 50 adımlık dönem için eğitilmiştir. Kayıp oranı son adımdan sonra 0.2772 değerlerine düşürülmüştür. Elde edilen doğruluk oranı ise yüzde 79 dur. Elde edilen verilere göre eğitim sayısı arttıkça doğruluk oranı da artmaktadır. Çalışmanın Facebook Messenger gibi uygulamalar yerine Telegram ile yapılması uygulamanın daha güncel ve kullanılabilir olmasını sağlamaktadır.

Çalışmanın müşteri analizlerinde müşteri profilini belirleme aşamasında, kullanıcı destek, müşteri hizmetleri ya da daha kapsamlı alanlara uyarlanabileceği düşünülmektedir. TPU destekli bilgisayarlarda corpus veri kümesi gibi daha kapsamlı veri kümesi kullanıldığında kapsamı geniş cevaplar üretilebileceği ve yüksek doğrulukta sonuçlar elde edilebileceği düşünülmektedir.

## ÇIKAR ÇATIŞMASI BEYANI (CONFLICT OF INTEREST STATEMENT)

Yazarlar tarafından herhangi bir çıkar çatışması bildirilmemiştir.

## KAYNAKLAR (REFERENCES)

- [1] J. Huang, M. Zhou, D. Yang, “Extracting Chatbot Knowledge from Online Discussion Forums”, *Proceedings of the 20th International Joint Conference on Artificial Intelligence(IJCAI)*, January 6–12 2007, Hyderabad, India, R. Sangal, H. Mehta, R. K. Bagga, Morgan Kaufmann Publishers, Hyderabad, India: 2007, pp. 423-428.
- [2] I. Sutskever, Q. Le, O. Vinyals, (2014). Sequence to Sequence Learning with Neural Networks, In *Advances in neural information processing systems*, December 8–12 2014, Montreal, Canada, Z. Ghahramani, M. Welling, C. Cortes, Morgan Kaufmann Publishers, NY, USA: 2014, pp. 3104-3112.
- [3] F. Bonilla, F. Ugalde, “Automatic Translation of Spanish Natural Language Commands to Control Robot Comands based on LSTM neural network”, *Third IEEE International Conference on Robotic Computing (IRC)*, February 25-27 2019, Naples, Italy, F. G. Rosas NY: Curran Associates, 2019, pp. 125-131.
- [4] A. Xu, Z. Liu, Y. Guo, V. Sinha, and R. Akkiraju, “A new chatbot for customer service on social media”, *In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, May 6-11 2017, Denver, USA, H. Mark, S. Fussell, NY: Association for Computing Machinery, 2017, pp. 3506-3510.
- [5] M. H. Su, C. H. Wu, K. Y. Huang, Q. B. Hong and H. M. Wang, “A chatbot using LSTM-based multi-layer embedding for elderly care”, In *2017 International Conference on Orange Technologies (ICOT)*, December 8-10 2017, Crescent, Singapore, M. Dong, L. Wang, Y. Lu, H. Li, Singapore: Research Publishing Services, 2017, pp. 70-74.
- [6] M. Muslih, D. Supardi, E. Multipli, Y. M. Nyaman and A. Rismawan, “Developing Smart Workspace Based IOT with Artificial Intelligence Using Telegram Chatbot”, *In 2018 International Conference on Computing, Engineering, and Design (ICCED)*, September 6-10 2018, Bangkok, Thailand, T. Mantoro, NY: Curran Associates, 2018, pp. 230-234.
- [7] P. Muangkammuen, N. Intiruk and K. R. Saikaew, “Automated thai-faq chatbot using RNN-LSTM”, *In 2018 22nd International Computer Science and Engineering Conference (ICSEC)*, 30 October -1 November 2018, Chiang Mai, Thailand, E. Boonchieng, NY: Curran Associates, 2018, pp. 1-4.
- [8] W.C. Lee, Y.S. Wang, T.S. Hsu and K.Y. Chen, “Scalable Sentiment for Sequence-to-Sequence Chatbot Response with Performance Analysis”, *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 15–20 2018, Calgary, AB, M. Hayes, NY: Curran Associates, 2018, pp. 6164-6167.

- [9] F. Rahman, “*Sequence to Sequence Learning with Keras*”, Available: <https://github.com/farizrahman4u/seq2seq>, [Accessed: Sept. 12, 2019].
- [10] R. Kompella, “*Neural Machine Translation — Using seq2seq with Keras*”, Available: <https://towardsdatascience.com/neural-machine-translation-using-seq2seq-with-keras-c23540453c74>, [Accessed: Sept. 12, 2019].
- [11] Textminingonline, “*Getting Started with Word2Vec and GloVe*”, Available: <https://textminingonline.com/getting-started-with-word2vec-and-glove>, [Accessed: Sept. 12, 2019].
- [12] J. S. Chawla, “*What is GloVe?*”, Available: <https://medium.com/@japneet121/word-vectorization-using-glove-76919685ee0b>, [Accessed: Sept. 12, 2019].
- [13] R. Kaus, “*Dataset for chatbots*”, Available: <https://www.kaggle.com/kausr25/chatterbotenglish>, [Accessed: Sept. 12, 2019].
- [14] C. Jongerius, “*Quantifying Chatbot Performance by using Data Analytics*”, Utrecht University, Faculty of Science Theses, Master thesis, 2018.
- [15] PythonAnywhere, “*PythonAnywhere in one minute*”, Available: <https://www.youtube.com/watch?v=NH2PhXYvrWs>, [Accessed: Sept. 12, 2019].
- [16] Telegram, “*Telegram Bot Api*”, Available: <https://core.telegram.org/bots/api#available-methods>, [Accessed: Sept. 12, 2019].
- [17] Tensorflow, “*TensorBoard*”, Available: <https://www.tensorflow.org/tensorboard>, [Accessed: Sept. 12, 2019].

### Ali Hakan IŞIK

1980 Burdur doğumlu olan Ali Hakan IŞIK, 2002 yılında Gazi Üniversitesi, Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümünden mezun oldu. Özel sektör ve kamuda Mühendis olarak çalıştı. 2006 yılında Gazi Üniversitesinde çalışmaya başladı. Gazi Üniversitesi Fen Bilimleri Enstitüsünde, 2005 yılında Yüksek Lisans ve 2012 yılında ise Doktora çalışmalarında bulundu. Burdur Mehmet Akif Ersoy Üniversitesi Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Bilgisayar Bilimleri Anabilim Dalında, 2013 yılında çalışmaya başladı. 2017 yılında Bilgisayar Bilimleri ve Mühendisliği Bilim Dalından Doçent ünvanı aldı. Yapay zeka, karar destek sistemleri, biyomedikal, yazılım, optimizasyon konularında makaleleri bulunan Doç. Dr. Ali Hakan IŞIK, evli olup 2 çocuk sahibidir.

### Ayşenur YAĞCI

1997 İzmir doğumlu olan Ayşenur YAĞCI, 2015 yılında Büyükçiftliği Anadolu Lisesinden mezun oldu. 2019 yılında Mehmet Akif Ersoy Üniversitesi, Mühendislik Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümünden mezun oldu. İlgi alanları arasında yapay zeka, raspberry pi ve arduino programlama yer almaktadır.