Türkçe Anahtar Sözcük Çıkarımında LSTM ve BERT Tabanlı Modellerin Karşılaştırılması

A Comparison of LSTM and BERT Based Models in Turkish Keyword Extraction

Özlem AYDIN
Trakya Üniversitesi
Bilgisayar Mühendisliği Bölümü
Edirne, Türkiye
ozlema@trakya.edu.tr
ORCID: 0000-0002-6401-4183

Hüsein KANTARCI
Trakya Üniversitesi
Bilgisayar Mühendisliği Bölümü
Edirne, Türkiye
huseinkantarci@trakya.edu.tr
ORCID: 0009-0000-7590-6454

Öz

Günümüzde internet ortamında metne dayalı veri çok hızlı bir şekilde artış göstermektedir ve bu büyük veri içinden istenilen bilgiyi barındıran doğru içeriklere ulaşabilmek önemli bir ihtiyaçtır. İçeriklere ait anahtar sözcüklerin bilinmesi bu ihtiyacı karşılamada olumlu bir etki sağlayabilmektedir. Bu çalışmada, doğal dil işleme ve derin öğrenme modelleri ile Türkçe metinleri temsil eden anahtar sözcüklerin belirlenmesi amaçlanmıştır. Veri seti olarak Türkçe Etiketli Metin Derlemi ve Metin Özetleme-Anahtar Kelime Çıkarma Veri Seti birlikte kullanılmıştır. Derin öğrenme modeli olarak çalışmada iki farklı model ortaya konmuştur. İlk olarak Uzun Ömürlü Kısa Dönem Belleği (Long Short-Term Memory, LSTM) katmanlı bir Diziden Diziye (Sequence to Sequence - Seq2Seq) model tasarlanmıştır. Diğer model ise BERT (Bidirectional Encoder Representations from Transformers-Transformatörler ile İki Yönlü Kodlayıcı Temsilleri) ile oluşturulmuş Seq2Seq bir modeldir. LSTM katmanlı Seq2seq modelin başarı değerlendirmesinde ROUGE-1 ölçütünde 0,38 F-1 skoruna ulaşılmıştır. BERT tabanlı Seq2Seq modelde ROUGE-1 ölçütünde 0.399 F-1 değeri elde edilmiştir. Sonuç olarak Transformatör (Transformer) mimarisini temel alan BERT tabanlı Seq2Seq modelin, LSTM tabanlı Seq2seq modele kıyasla daha başarılı olduğu gözlemlenmiştir.

Anahtar sözcükler: Anahtar Sözcük Çıkarımı, Derin Öğrenme, Seq2Seq mimarisi, Transformatör mimarisi

Abstract

Nowadays, text-based data on the internet is increasing very rapidly and it is an important need to reach the right content that contains the desired information from this big data. Knowing the keywords of the content can provide a positive effect in meeting this need. In this study, it is aimed to determine the keywords representing Turkish texts with natural language processing and deep learning models. Turkish Labeled Text Corpus and Text Summarization-Keyword Extraction Data Set were used together as data set. Two different deep learning models were presented in this study. Firstly, Sequence-to-Sequence (Seq2Seq) Model with Long Short-Term Memory (LSTM) layers is designed. The other model is a Seq2Seq model with BERT (Bidirectional Encoder Representations from Transformers). In the evaluation of success of the LSTM layered Seq2seq model, an F-1 score of 0.38 was achieved in the ROUGE-1 criterion. In the BERT-based Seq2Seq model, an F-1 value of 0.399 was obtained in the ROUGE-1 criterion. As a result, it has been observed that the BERT based Seq2Seq model based on the Transformer architecture is more successful than the LSTM based Seq2Seq model.

Keywords: Keyword Extaction, Deep Learning, Seq2seq architecture, Transformer architecture

1. Giriş

Herhangi bir konuda internet üzerinde arama yapıldığında çoğu durumda ilgili bilgiye ulaşmada yetersizlikler görülebilmektedir. Yapılan aramada doğru bilgiye ulaşmak için metinlerin ana özelliklerini, içeriğini, temasını vb. özelliklerini temsil eden bir sözcük alt kümesine, yani anahtar sözcüklere sahip olunması fayda sağlayabilir. Anahtar sözcükler bir belgenin içeriğini özetleyebilen en küçük birimlerdir. Anahtar sözcüklerin kullanıma örnek olarak akademik yayınlar da verilebilir. Bu tür yayınlarda çalışmayı en iyi temsil eden sözcükler/sözcük öbekleri anahtar sözcükler olarak yayınlanan metnin başında belirtilmektedir. Böylelikle çalışma hakkında okuyucuya bir ön bilgi verilmiş olur. Okuyucu, anahtar sözcükler üzerinden metnin tamamını okuyup okumama kararını kolaylıkla verebilir.

Genel olarak, bir belgenin içeriği hakkında özet bir bilgiye ihtiyaç duyan herhangi bir görevde anahtar sözcüklerin varlığının yararlı olduğu düşünülebilir. Anahtar sözcükler metnin özetinin özeti olarak düşünülebilir ve bu bağlamda,

metnin içerdiği en önemli bilgileri temsil eden en küçük sözcük grubu olduğu söylenebilir. Anahtar Sözcük Çıkarımı (ASÇ) görevi, bu "gerçek" anlamı en iyi temsil eden sözcükleri çıkarmayı amaçlamaktadır. Bir belgenin neyle ilgili olduğunu öğrenmek için anahtar sözcükleri analiz etmek, benzer belgeleri bulmak için anahtar sözcükleri karşılaştırmak yeterli olacaktır. Dolayısıyla ASÇ görevi bu şekilde birbiri ile alakalı olan belgeleri seçmenin hızlı bir yolunu da sağlayabilir.

ASÇ ile her türdeki metnin anahtar sözcükleri tespit edilebilir: normal belgeler, iş raporları, sosyal medya yorumları, çevrimiçi forumlar ve incelemeler, haber raporları vb. metinler. Günlük olarak gönderilen ve alınan 290 milyardan fazla e-posta ve her dakika gönderilen yarım milyon tweet atılan bir ortamda, makinelerin büyük veri kümelerini analiz etmek ve önemli bilgileri çıkarmak için kullanılması kesinlikle çok faydalı bir konudur [1].

2. Literatür araştırması

ASÇ denetimli ve denetimsiz olmak üzere iki farklı yaklaşımla gerçekleştirilmektedir. Günümüze kadar yapılan çalışmalarda denetimsiz yaklaşım ile uygulanan yöntemler daha popüler olmuştur. Bunun nedeni alan bağımsız olmaları ve etiketli bir eğitim verisine ihtiyaç duymamalarıdır. Diğer taraftan denetimli yöntemlerin daha güçlü bir modelleme yeteneği vardır ve denetimsiz yaklaşımlara göre doğruluk değerinin daha yüksek olduğu söylenebilir [2]. Başlangıçta, denetimli yaklaşımlar için ASÇ görevi bir ikili sınıflandırma problemi olarak ele alınmıştır. Bu yaklaşımda bir sınıflandırıcının eğitimi, bir aday anahtar sözcüğün bir anahtar sözcük olup olmadığını belirlemek üzerinedir. KEA [3] ve GenEx [4] sistemleri alanda geliştirilen en tipik ve bilinen sistemlerdir. Denetimli sistemlerde sınıflandırma için Naive Bayes, Karar Ağaçları, Karar Destek Makinaları, Çok Katmanlı Algılayıcılar vb. algoritmalar kullanılmıştır. İfade edilen bu geleneksel makine öğrenmesi algoritmaları dışında son yıllarda derin öğrenme algoritmaları ile de ASÇ görevi yerine getirilen çalışmalar yapılmıştır [5,6,7].

Denetimsiz yaklaşımlar beş farklı grupta incelenmektedir: çizge-tabanlı sıralama, konu tabanlı kümeleme, eş zamanlı öğrenme, dil modelleme ve istatiksel yöntemler. Çizge tabanlı yöntemlerde metinler çizge olarak temsil edilirler. Bir aday anahtar sözcüğün önemi dokümandaki diğer aday anahtar sözcükler ile ne kadar ilişkili olduğu ile tanımlanır. Bir aday anahtar sözcük, çok sayıda aday anahtar sözcük ile alakalı ise ve bu adayların önemli olması durumunda kendisi de önemli olur. TextRank [8], SingleRank [9], CiteTextRank [10], ve RAKE [11] yöntemleri en bilinen çizge tabanlı yöntemlerdir. Konu tabanlı kümelemede aday anahtar sözcükler konularına göre gruplanır. Ana konuları belirlemek için kümeleme teknikleri ve Gizli Dirichlet Ayrımı (Latent Dirichlet Allocation) kullanılmaktadır. KeyCluster [12] ve TopicRank [13] konu tabanlı kümeleme yaklaşımıyla geliştirilmiş sistemlerdir. Eş zamanlı öğrenme yaklaşımı, metin özetleme ve ASÇ'nin eş zamanlı yerine getirilmesi durumunda birbirlerine fayda sağlayacakları varsayımı ile ortaya çıkmıştır. Araştırmacı [14] özetleme ve ASÇ görevlerini eş zamanlı gerçekleştiren bir çizge tabanlı yaklaşım önermiştir. Bu yaklaşım, "Bir cümle önemli sözcükler içeriyorsa önemlidir ve önemli sözcükler de önemli cümlelerde bulunur" düşüncesi üzerinden uygulanmaktadır. Dil modelleme yaklaşımı ile yapılan bir çalışmada aday anahtar sözcükleri çıkaran ve anahtar sözcükleri sıralama adımlarını birleştiren bir yaklaşım önerilmiştir [15].

Türkçe dili için günümüze kadar yapılmış birkaç ASÇ çalışması bulunmaktadır. Bu alandaki ilk çalışmada literatürde yeri olan KEA algoritması Türkçe için uygulanmış ve gerçekleştiriminde bu algoritmanın özgün kök bulucusunun ve etkisiz sözcük (stopword) listesinin Türkçe karşılıkları ile yer değiştirilmiştir [16]. KEA-TR adını verdikleri sistemlerinde özgün KEA algoritmasında bulunmayan "relative length" özelliğini algoritmaya dâhil etmişlerdir. 50 doküman üzerinde ortalama %62 eşleşme oranını elde etmişlerdir. Diğer bir çalışmada, isim öbeği (noun phrase) ve isim öbeği başı (NP heads) için bunların uzunluğu ve belgenin başında bulunma olasılıklarına ait istatistik verilerini kullanılmıştır [17]. Bu çalışmada 60 dokümanda %35,50 eşleşme oranına ulaşmışlardır. Çoklu kriterli sıralama (Multi-Criterion Ranking-MCR) yönteminin anahtar sözcük grubu (keyphrase) seçiminde uygulandığı bir çalışma vardır [18]. Bu yöntem iki aşamadan oluşmaktadır. İlk aşamada aday anahtar sözcük grupları metinden çıkarılır ve her sözcük grubu özellikleri üzerinden hesaplanır. İkinci aşamada aday sözcük gruplarından bir Hasse diyagramı oluşturulur. Sonrasında MCR yöntemi ile uygun olan anahtar sözcük grupları seçilir. Çalışma eğitimli derleme ihtiyaç duymamaktadır. TurKeyX [17] çalışmasından daha başarılı sonuçlar elde edilmiş, KEA-TR ile ise yakın başarım göstermiştir. Başka bir çalışmada metin verisi olarak akademik yayınları kullanarak ASÇ gerçekleştirilmiştir [19]. Doğal Dil İşleme (DDİ) yöntemleri ile öncelikle edatları ve bağlaçları metinlerden çıkarmışlardır. Sonrasında metindeki sözcüklerin TF-IDF değerlerini hesaplamışlar ve TextRank algoritması ile anahtar sözcüklerini belirlemişlerdir. 80.181 tane akademik makale üzerinde 0,832 doğruluk değerine ulaşmışlardır. Başka bir çalışmada, Yıldız Teknik Üniversitesi bünyesinde iç paydaşlar tarafından aktif olarak kullanılan 7/24 Yıldızlı Hat Yönetim Sistemi içerisinde biriken mesajlar anahtar sözcük özellik çıkarımı için kullanılmıştır [20]. KiKare, Bilgi Kazancı ve TF-IDF yöntemleri ile gerçekleştirilen analiz sonuçlarında anlamlı ve faydalı anahtar sözcükler bulunmuştur.

Transformatör (transformer) mimarisi kullanan derin öğrenme tabanlı bir yöntemi firmaların internet sitelerindeki anahtar sözcükleri tespit etmek için kullanan bir çalışmada 0,39 F-1 skoruna ulaşılmıştır [21]. Bu çalışmadaki veri seti hazırlanırken 25 farklı firmanın internet sitelerindeki anahtar sözcükler insan eliyle seçilmiştir. Sentence-BERT modeli ile anahtar sözcükleri belirlendikten sonra, bu sözcükleri kümeleme yoluyla firmaların faaliyet alanları ortaya çıkarılmaya çalışılmıştır. Türkçe için yapılan bir diğer çalışmada çizge merkezcilik ölçütlerinin ASÇ başarımına etkisi TextRank algoritması üzerinden incelenmiştir [22]. Deneysel sonuçlar incelendiğinde özvektör merkeziliği ölçütünün TextRank algoritması ile 0,249 F-1 değeri ürettiği görülmektedir. TextRank ve TF-IDF yöntemlerinin Türkçe bilimsel makaleler üzerinde ASÇ başarısının kıyaslandığı bir çalışmada, TextRank yöntemi ile 0,39 F-1, TF-IDF yöntemi ile 0,40 F-1 değerleri elde edilmiştir [23]. Bu başarım değerlerine

beş anahtar sözcük sayısı ile ulaşılmıştır. Başka bir çalışmada araştırmacılar hızlı ve rekabetçi bir ASÇ modeli tasarlamış ve RAKE, YAKE ve TF-IDF modelleri ile oluşturulmuş diğer çevrimiçi ASÇ modelleriyle performansını kıyaslamışlardır [24]. Çalışmalarında amaç, kullanıcıların video toplantılara katılmadan önce katılacağı toplantı hakkında bilgi sahibi olmaları için anahtar sözcükleri görebileceği bir uygulama (için alt yapı) oluşturmaktır. Sonuç olarak geliştirdikleri modelin diğer Türkçe çevrimiçi ASÇ modellerine göre daha iyi performans gösterdiğini ispatlamışlardır.

Bu çalışmada Türkçe dilinde hazırlanmış bir veri setinde ASÇ için derin öğrenme yöntemleri ile iki farklı model ortaya konmuştur. Geliştirilen ilk modelde LSTM katmanlı bir Seq2seq model geliştirilmiştir. Çalışmanın anlatımında bu model Model-1 olarak adlandırılmıştır. Veri seti olarak Türkçe Etiketli Metin Derlemi ve Metin Özetleme-Anahtar Kelime Çıkarma Veri Seti birlikte kullanılmıştır. Model-1'in eğitiminde veri setinde yapılan ön işlemler sonucu elde edilen 59.892 metin kullanılmıştır. Bu modelde ROUGE-1 ölçütünde 0,38 F-1 skoruna ulaşılmıştır. Diğer geliştirilen model literatürde özellikle son yıllarda derin öğrenme çalışmalarında başarılı sonuçlar elde ettiği gözlemlenmiş olan Transformatör mimarisini temel alan BERT(Bidirectional Encoder Representations from Transformers-Transformatörler ile İki Yönlü Kodlayıcı Temsilleri) modeli ile oluşturulmuştur. Çalışmada Model-2 olarak adlandırılan bu modelde aynı veri setinin tamamı, yani 137.894 metin eğitime verilebilmiştir. Bu veri setinin modelde iki farklı şekilde eğitimi yapılmıştır. Kök bulma işleminin uygulanmadığı veri seti ile eğitilen modelde ROUGE-1 ölçütünde 0,394 F-1 skoru elde edilmiştir. Kök bulma işleminin uygulandığı veri seti ile eğitilen modelde ise ROUGE-1 ölçütünde 0,399 F-1 skoruna ulaşılmıştır.

3. Materyal ve Metot

Bu bölümde öncelikle, modellerin geliştirilmesinde ve sonuçların elde edilmesinde faydalanılan araçlardan bahsedilmiştir. Sonrasında sırasıyla kullanılan veri seti hakkında genel bilgiler verilmiş, veri seti üzerinde gerçekleştirilen ön işlemlerden bahsedilmiş ve modellerin ayrıntılı mimarisi açıklanmıştır. Son olarak da modellerden elde edilen bulgular paylaşılmıştır.

3.1 Programlama dili ve kütüphaneler

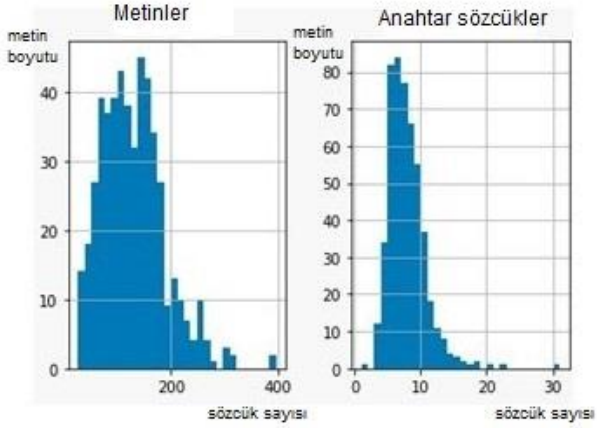
Model-1'e ait uygulamanın geliştirilmesinde yüksek seviyeli bir dil olan Python programlama dili kullanılmıştır. Metnin ön işlemesi, eğitimi, testi ve geliştirilen modelin başarımının değerlendirilmesi gibi tüm aşamalarda birkaç kütüphane kullanılmıştır. Bunlar: NLTK (Natural Language Toolkit) [25], NumPy [26], Pandas [27], Pyhton Regex, TensorFlow [28], ve Rouge-Score'dur[29]. Model-1'in eğitimin gerçekleştirilebilmesi için Google Colab Pro servisi kullanılmıştır.

Model-2'nin geliştirilmesinde Python programlama dili, eğitim ve üretim aşamalarında TensorFlow ve Keras kütüphaneleri, veri ön işleme aşamalarında ise Pandas ve Datasets kütüphaneleri kullanılmıştır. Veri işleme, eğitim ve üretim aşamalarında, yüksek performans gerektiren işlemleri gerçekleştirebilmek için güçlü bir uzaktan sunucu kullanılmıştır. Bu sunucu, 4608 Cuda çekirdeğine sahip 48 GB Quadro RTX 8000 grafik işlemcisi ve 256 GB RAM belleğiyle donatılmıştır. Bu yüksek performanslı donanım, büyük veri kümesini işlemek, BERT modelini eğitmek ve anahtar sözcük tahminleri üretmek için gerekli olan yoğun hesaplama işlemlerini hızlı ve etkili bir şekilde gerçekleştirmeye yardımcı olmuştur. Büyük bellek kapasitesi ise veri kümesini ve model parametrelerini bellekte tutabilmek için önemlidir.

3.2 Veri seti

Çalışmada, Türkçe DDİ çalışmalarındaki derlem ihtiyacını gidermek adına hazırlanmış olan Türkçe Etiketli Metin Derlemi ile Metin Özetleme-Anahtar Kelime Çıkarma Veri Seti birlikte kullanılmıştır. Türkçe Etiketli Metin Derlemi, Türkçe makalelerin başlık, özetçe ve anahtar sözcüklerinden oluşmaktadır [30]. Özellikle metin sınıflandırma için oluşturulmuş bir derlem olmasına rağmen içeriği nedeniyle ASÇ, başlık üretme ve metin özetleme gibi alanlarda da kullanılabilecek niteliktedir. Bu derlemde 35 farklı konuda, konu başına 200 belge bulunmaktadır. Bazı konuların belgelerinde anahtar sözcükler bulunmadığı için 28 konu başlığı altındaki toplam 5.254 makale bu çalışma için ayrılmıştır. Metin Özetleme-Anahtar Kelime Çıkarma Veri Seti ise Fırat Üniversitesi Büyük Veri ve Yapay Zeka Laboratuvarı tarafından hazırlanmıştır ve 132.642 haber kaydından oluşmaktadır [31]. Bu farklı iki derlemin bir araya getirilmesi ile oluşan veri seti, toplamda 137.894 satırdan oluşmakta olup, özgün metinler ile anahtar sözcükler sütunlarından oluşan bir XML dosyası şeklinde oluşturulmuştur. Model-2'nin eğitimlerinde bu veri setinin tamamı kullanılmıştır.

Hazırlanan bu veri setinde metinler ve anahtar sözcüklerdeki sözcük dağılımı Şekil1'de verilmiştir. Buna göre en uzun metinler 0 ile 200 sözcük aralığında iken, en uzun anahtar sözcük sayısının 8 olduğu görülmektedir. Model-1'in eğitim modelindeki öğrenme zamanını düşürebilmek için bu uzunluklar dikkate alınarak, bu uzunlukların üzerinde sözcük sayısına sahip metinler eğitime verilmemiştir. Bu işlem sonucunda Model-1'in eğitiminde kullanılacak olan veri setinin boyutu 59.892 satıra düşürülmüştür.



Şekil-1: Veri setindeki metinler ve anahtar sözcüklerin sözcük sayısı dağılımı

3.3 Ön işleme

Ön işleme DDİ çalışmalarında en önemli adımlardan biridir. Bu aşamada verilerin dağılımına, hangi tekniklere ihtiyaç duyulduğuna ve ne kadar derinlemesine temizlenmesi gerektiğine dikkat edilmelidir. Bu aşamanın hiçbir zaman tek bir kuralı yoktur, kullanılan derleme ve geliştirilecek uygulamaya göre farklılık göstermektedir. Bu çalışmada ön işleme için öncelikle veri setinin özgün metni ve anahtar sözcüklerinin içerdiği noktalama işaretleri kaldırılmıştır. Sonrasında sözcüklerin küçük harfe dönüştürülmesi gerçekleştirilmiştir. Bu işlemlerde Regex kütüphanesinden faydalanılmıştır. Model-1'de eğitime verilen veri setinde Türkçe karakter dönüşümü gerçekleştirilmiştir.

Model-2 için veri setinin ön işlemesinde metinlerin ve anahtar sözcüklerin noktalama işaretleri kaldırılmış, sözcüklerin küçük harfe dönüşümü yapılmış ve metinlerdeki sayılar çıkarılmıştır. Son olarak etkisiz sözcükleri veri setinden çıkarılmıştır. Etkisiz sözcükler metinlere herhangi bir ek değer vermeyen en yaygın kullanılan sözcüklerdir. Bunların kaldırılması hesaplamayı ve verimliliği arttırmaktadır. Bu ön işlem için etkisiz sözcük listesi [32] edinilmiş ve makalelerin içeriğine göre filtrelenmiştir. Model-2 için yapılan deneyler için veri setine ayrıca kök bulma işlemi de uygulanmıştır. Veri setinden rastgele seçilmiş bir metnin ve anahtar sözcüklerinin özgün hali ve ön işlenmiş son halleri Çizelge-1'de verilmiştir.

Çizelge-1: Ön işlenmiş metinler ve anahtar sözcükleri için bir örnek

		Anahtar sözcükler
Özgün Metin	Bu çalışmada Hacıkadın Vadisi'nin (Keçiören, Ankara) florası araştırılmıştır. Ankara'nın Keçiören İlçesi'ne bağlı Hacıkadın Vadisi, İran-Turan fitocoğrafik bölgesinde ve Davis tarafından Türkiye Florası'nda uygulanan grid kareleme sistemine göre A4 karesinde yer almaktadır. Kasım 2006 ve Mayıs 2008 tarihleri arasında araştırma alanında yapılan 14 arazi çalışması, herbaryum ve literatür taramaları sonucunda, alanda 63 familyaya ait 258 cins, 480 takson, 5 alttür ve 4 varyete tespit edilmiştir. Bu taksonların 45'i endemik olup, endemizm oranı %9.3'tür. Taksonların fitocoğrafik bölgelere göre dağılımları ve oranları: 89 takson, %18.5 İran-Turan-, 45 takson, %9.3 Avrupa-Sibirya- ve 43 takson, %8.9 Akdeniz fitocoğrafya bölgesinin elementidir. 303 takson, %63.1 ise geniş yayılışlı veya fitocoğrafik bölgesi henüz belirlenemeyenlerdendir. Tespit edilen taksonların 1'i Pteridophyta, 479'u ise Spermatophyta divizyosuna aittir.	Flora, Hacıkadın Vadisi, Ankara, Türkiye
Model-1 için ön işlenmiş metin	Bu calismada hacikadin vadisinin kecioren ankara florasi arastirilmistir ankaranin kecioren ilcesine bagli hacikadin vadisi iranturan fitocografik bolgesinde ve davis tarafindan turkiye florasinda uygulanan grid kareleme sistemine gore a4 karesinde yer almaktadir kasim 2006 ve mayis 2008 tarihleri arasinda arastirma alaninda yapilan 14 arazi calismasi herbaryum ve literatur taramalari sonucunda alanda 63 familyaya ait 258 cins 480 takson 5 alttur ve 4 varyete tespit edilmistir bu taksonlarin 45i endemik olup endemizm orani 93tur taksonlarin fitocografik bolgelere gore dagilimlari ve oranlari 89 takson 185 iranturan 45 takson 93 avrupasibirya ve 43 takson 89 akdeniz fitocografya bolgesinin elementidir 303 takson 631 ise genis yayilisli veya fitocografik bolgesi henuz belirlenemeyenlerdendir tespit edilen taksonlarin 1i pteridophyta 479u ise spermatophyta divizyosuna aittir	flora hacikadin vadisi ankara turkiye
Model-2 için ön işlenmiş metin (Kök bulma işlemi olmayan)	çalışmada hacıkadın vadisinin keçiören ankara florası araştırılmıştır ankaranın keçiören ilçesine bağlı hacıkadın vadisi iranturan fitocoğrafik bölgesinde davis tarafından türkiye florasında uygulanan grid kareleme sistemine göre a karesinde yer almaktadır kasım mayıs tarihleri arasında araştırma alanında yapılan arazi çalışması herbaryum literatür taramaları sonucunda alanda familyaya cins akson alttür varyete tespit edilmiştir taksonların i endemik olup endemizm oranı tür taksonların fitocoğrafik bölgelere göre dağılımları oranları takson iranturan takson avrupasibirya takson akdeniz fitocoğrafya bölgesinin elementidir takson geniş yayılışlı fitocoğrafik bölgesi henüz belirlenemeyenlerdendir tespit edilen taksonların i pteridophyta u spermatophyta divizyosuna aittir	flora hacıkadın vadisi ankara türkiye
Model-2 için ön işlenmiş metin (Kök bulma işlemi olan)	çalış hacıkadın vadi keçiören ankara flora araştır ankara keçiören ilçe bağ hacıkadın vadi iranturan fitocoğrafik bölge davis tarafından türkiye flora uygula grid karele sistem göre a kare yer al kasım mayıs tarih ara araştır alan yap arazi çalış herbaryum literatür tara sonuç alan familya cins takson alttür varyete tespit et taksonların i endemik ol endemizm oran tür taksonların fitocoğrafik bölge göre dağılım oran takson iranturan takson avrupasibirya takson akdeniz fitocoğrafya bölge element takson geniş yay fitocoğrafik bölge henüz belirle tespit et taksonların i pteridophyta u spermatophyta divizyosuna ait	flora hacıkadın vadi ankara türkiye

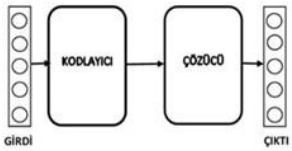
3.4 Geliştirilen modeller

Bu bölümde geliştirilen LSTM katmanlı bir Seq2seq model ve BERT ile oluşturulmuş Seq2Seq modelle ilgili detaylı bir anlatım yapılacaktır.

3.4.1 LSTM tabanlı Seq2seq model

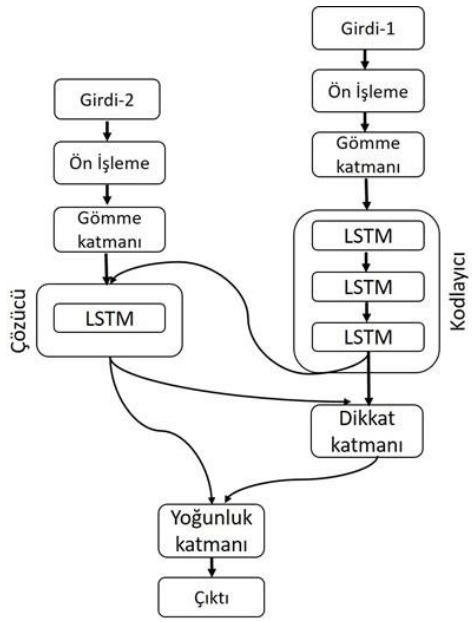
Seq2Seq modeller hem girdinin hem çıktının bir dizi olduğu problemlerde kullanılabilen bir ağ modelidir. Bu sebeple makine çevirisi, metin sınıflandırma, metin özetleme, ASÇ vb. birçok DDİ uygulamasında Seq2Seq modeller kullanılabilmekte ve başarılı çözümler sunmaktadır. Seq2Seq modeller, kodlayıcı ve çözücü olmak üzere iki sinir ağından oluşan bir makine öğrenimi modelidir [33]. Bu iki sinir ağı genellikle aynı yapıya sahip olmaktadırlar. Birinci sinir ağı olan kodlayıcı girdi olarak bir cümle almakta ve bir sayı dizisi çıkarmaktadır. İkinci sinir ağı, yani çözücü ise bu sayı dizisini girdi olarak almakta ve bir cümle üretmektedir. Sonuç olarak, Şekil 2'de görülen kodlayıcı-çözücü mimarisi temel olarak girdi ve çıktı dizilerinin farklı uzunluklarda olduğu diziden diziye problemlerini çözmek için kullanılır. ASÇ probleminde

bu modelin girdisi uzun bir sözcük dizisi iken çıktısı giriş dizisinin kısa bir hali olacaktır.



Şekil-2: Seq2seq modeli genel gösterimi

Geliştirilen model için veri setinin %90'ı eğitim verisi, %10'u da test verisi olarak ayrılmıştır. Eğitim modeline veri setindeki verileri girdi olarak verip bir çıktı alabilmek için öncelikle sözcüklerin bir tamsayı dizesine dönüştürülmesi gerekmektedir. Bunun için Keras Tokenizer kullanılmıştır [34]. Keras'ın Tokenizer sınıfı, tokenize işleminde ya her metin girişini tamsayı dizisine ya da ikili değerler biçimindeki her belirteç için bir katsayıya sahip olan bir vektöre dönüştürür. Tokenize işleminden sonra, Şekil-3'de verilen eğitim modeli oluşturulmuştur. Modelde 100 boyutlu bir gömme katmanı (embedding layer), kodlayıcı için üç adet 300 boyutlu Uzun Ömürlü Kısa Dönem Belleği (Long Short-Term Memory, LSTM) katmanı, çözücü için yine 300 boyutlu bir LSTM katmanı ve son olarak da SoftMax fonksiyonuna sahip bir yoğunluk katmanı (dense layer) ile Keras'ın AdditiveAttention denilen dikkat katmanı (attention layer) kullanılmıştır.



Şekil-3: Geliştirilen modelin mimarisi

Geliştirilen modelin mimarisinde kodlayıcı kısım, bir dizi girdi (özgün metinler) almakta ve bunu sabit uzunlukta bir vektör temsiline dönüştürmektedir. Bu vektör, çıktı dizisini oluşturmak için gerekli olan girdi dizisiyle ilgili tüm bilgileri içermektedir. Modelde kodlayıcı, giriş sırasını sırayla işleyen ve ondan yararlı özellikler çıkaran üç adet LSTM katmanından oluşmaktadır. Mimarinin çözücü kısmı, kodlanmış vektörü almakta ve çıktı dizisini (anahtar sözcük dizisi) üretmektedir. Çözücü ayrıca, kodlanmış vektörü girdi olarak alan ve bir dizi çıktı üreten bir adet LSTM katmanından oluşmaktadır. Çözücü her çıktıyı adım adım üretmekte, her çıktı adımı bir önceki çıktıya ve kodlanmış vektöre göre koşullanmaktadır. Modelde çözücünün doğru çıktı sırasını oluşturmasına yardımcı olmak için bir dikkat (attention) mekanizması kullanılmıştır. Bu mekanizma, kod çözücünün çıkış dizisini oluştururken farklı zamanlarda kodlanmış giriş dizisinin farklı bölümlerine odaklanmasını sağlar. Özellikle, her zaman adımında çözücü çıkışına dayalı olarak kodlayıcı çıktılarının ağırlıklı bir toplamını hesaplayan bir ek dikkat katmanı kullanılmıştır. Bu ağırlıklı toplam daha sonra çözücü çıktısı ile birleştirilmekte ve elde edilen vektör, çıktı dizisini üretmek için bir yoğunluk katmanından geçirilmektedir.

Genel olarak, kodlayıcı-çözücü mimarisi, modelin girdi dizisinin bağlamsal bilgilerini yakalamasına ve bunu bir çıktı dizisi oluşturmak için kullanmasına izin verir. Model, bir dikkat mekanizması kullanarak girdi dizisinin farklı bölümlerine seçici olarak odaklanarak daha doğru çıktı dizileri oluşturmasına olanak tanımaktadır.

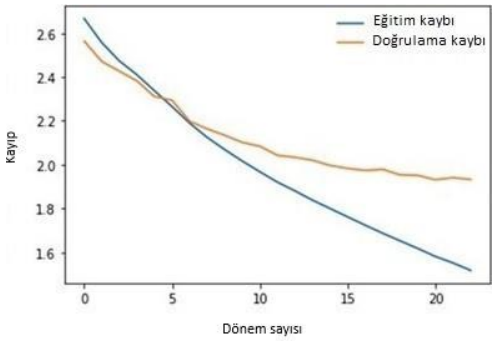
Çalışmada kullanılan veri setinde anahtar sözcüklerin yani çıktıların başına "sostok" (start of string token) ve sonuna "eostok" (end of string token) eklenmiştir. Bunlar çözücüye verilmeden önce hedef diziye eklenen özel belirteçlerdir (token). Çözücü çalışırken cümlenin nerden başlayıp nereye kadar çözmesi gerektiğini bu belirteçler sayesinde bilmektedir.

Modelin eğitilmesinde, eğitimin ne zaman sonlanacağına karar vermek için Keras'ın EarlyStopping sınıfı kullanılmıştır. Öğrenme sırasında doğrulama kaybı (validation loss) ile test kaybı (test loss) arasındaki farkın açılmaya başlaması modelin ezberlemeye (overfitting) başladığı veya gürültüyü öğrendiği anlamına gelmektedir. Bir eğitimin amacının kaybı en aza indirmek olduğunu düşünürsek, izlenecek metrikler 'val_loss' ve 'mod min' olacaktır. 'Val_loss' metriği, modelin eğitim sırasında görmediği yeni verilere ne kadar iyi genelleştirilebildiğinin bir ölçüsüdür. Bu "model kontrol noktası" adı verilen bir tekniği kullanır; burada modelin ağırlıkları her döngüde yalnızca doğrulama kaybı bir önceki döngüden bu yana azalmışsa kaydedilir. 'Mod min' metriği ise kontrol noktasının şimdiye kadar görülen minimum doğrulama kaybına dayalı olduğunu gösterir.

Belirli bir eğitim veri kümesini kullanarak modeli eğitmek için fit() yöntemi kullanılır. Girdi olarak eğitim verileri, eğitilecek dönem (epoch) sayısı, doğrulama verileri gibi farklı argümanları alır. Bir model.fit() eğitim döngüsü, 'min_delta' ve varsa 'patience' değerlerini göz önünde bulundurarak, her dönem sonunda kaybın artık azalıp azalmadığını kontrol edecektir. 'Min_delta' ve 'patience' değerleri, Keras'ta EarlyStopping sınıfına argüman olarak verilebilen değişkenlerdir. 'Min_delta', izlenen miktardaki (örneğin, doğrulama kaybı) minimum değişikliğin bir gelişme olarak kabul edilmesini belirtir. 'Patience' değişkeni ise eğitimi durdurmadan önce iyileştirme olmadan beklenecek döngü sayısını belirtir. Örneğin, min_delta=0,2 ve patience=3 ise, doğrulama kaybı art arda 3 döngü için en az 0,2 oranında iyileşmezse eğitim durmaktadır.

Eğitimin bir dönemi sonunda kaybın düşmediği tespit edildiğinde eğitim sonlandırılmaktadır. Çalışmada model eğitimi 23 dönem sonra EarlyStopping fonksiyonu sayesinde

otomatik durmuş ve modelin eğitimi tamamlanmıştır. EarlyStopping, Keras'ta, izlenen miktar (örneğin doğrulama kaybı) belirli sayıda döngü boyunca gelişmemişse eğitimi erken durdurmak için kullanılabilen bir fonksiyondur. Maksimum döngü sayısına ulaşmadan eğitimi durdurarak aşırı öğrenmeyi önlemeye ve zamandan tasarruf etmeye yardımcı olabilir. Modelin eğitimi sürecinde her dönemde gerçekleşen eğitim ve doğrulama kaybı değerleri Şekil 4'de görülmektedir. Eğitim sonlandıktan sonra, eğitilmiş modelden "sostok" ve "eostok" dikkate alınarak çözücü ile anahtar sözcük tahmin etme kısmına geçilmiştir.



Şekil-4: Model eğitimindeki eğitim ve doğrulama kaybı değerleri

3.4.2 BERT tabanlı Seq2Seq model

Transformatör tabanlı kodlayıcı-çözücü modelleri, [35] çalışmasında önerilmiş olup son zamanlarda büyük bir ilgi görmüş ve DDİ alanında büyük bir çığır açmıştır. Geleneksel dil işleme yöntemlerine kıyasla daha etkili ve verimli bir yaklaşım sunan modellerdir. Girdi sekanslarını paralel olarak işleyen dikkat mekanizmaları ve katman normalizasyonu gibi tekniklerle bağlamsal ilişkileri yakalayarak dil modellerini eğitirler. Özellikle BERT ve GPT gibi önceden eğitilmiş Transformatör modelleri, genel dil bilgisini öğrenerek çeşitli dil işleme görevlerinde etkileyici sonuçlar elde etmektedirler. Transformatörler, metin sınıflandırma, dil çevirisi, metin üretimi, ASÇ gibi birçok alanda kullanılmaktadırlar ve DDİ'de temel bir yapı haline gelmişlerdir.

Geleneksel dil işleme yöntemlerinde, Yinelemeli Sinir Ağı (Recurrent Neural Network, RNN), LSTM ve Geçitli Yineleme Birimi (Gated Recurrent Unit, GRU) gibi tekrarlayan sinir ağları kullanılır. Bu yöntemlerde, bir metin sekansı t-1 zaman adımındaki gizli durumla t zamanındaki girdi vektörünü işleme sokar. Yani, bir sözcüğün anlamını çıkarmak için önceki sözcüklerin bağlamsal bilgisi kullanılır. Bu yaklaşım, ardışık işlemler nedeniyle hesaplama maliyeti ve zaman açısından kısıtlayıcı olabilir. Transformatörler ve geleneksel mimariler arasındaki önemli bir fark, Transformatörlerin son derece paralelleştirilebilir olmalarıdır, bu da onların hesaplamayı en uygun hale getirmesini sağlar ve son derece büyük modellerin eğitilmesine olanak sağlar. Transformatörleri diğerlerinden ayıran diğer önemli özelliği dikkat mekanizmasını kullanmalarıdır [36]. Transformatör mimarisi Şekil5'de verilmiştir.

Transformatör mimarisi sol ve sağ olmak üzere iki bloktan oluşmaktadır. Soldaki blok kodlayıcı, sağdaki blok kod çözücü bloktur. Bu bloklar içerisinde var olan bileşenler:

Girdi Gömmeleri (Input Embeddings): Giriş dizilerini gömme vektörlerine dönüştürmek için kullanılan katmandır. Her sözcüğü bir vektör şeklinde ifade eder.

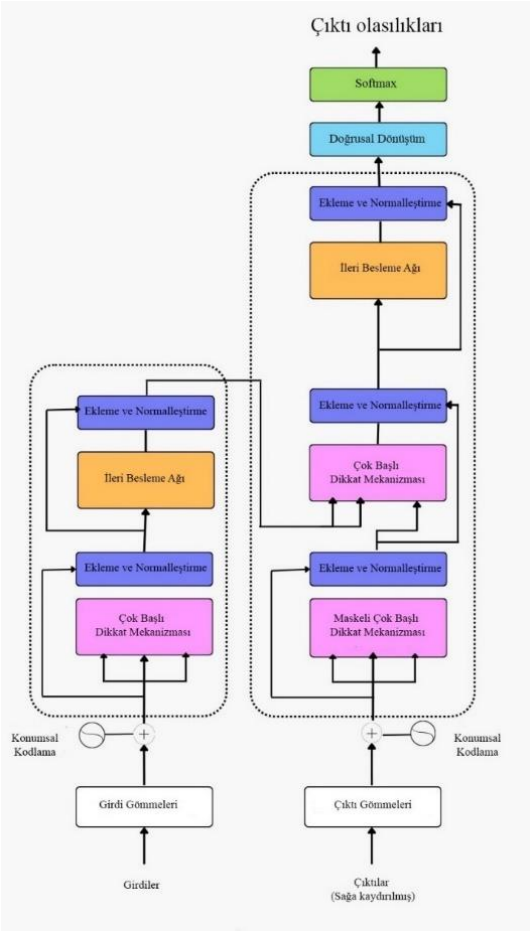
Konumsal Kodlama (Positional Encoding): Sözcük sırasını dikkate almak için konumsal kodlama kullanılır. Konumsal kodlama, her sözcüğün gömme vektörüne eklenen bir konumsal bilgi vektörüdür. Bu sayede modele, sözcük sırasının anlamını kavraması sağlanır.

Çok Başlı Dikkat Mekanizması (Multi-Head Attention Mechanism): Transformatör mimarisinin en önemli bileşenlerinden biri çok başlı dikkat mekanizmasıdır. Bu mekanizma, her bir sözcüğün diğer sözcüklerle etkileşimini hesaplar. Her başlık, farklı bir gömme uzayında dikkat hesaplaması yapar ve bu çok başlı dikkat hesaplamaları daha sonra birleştirilir.

Ekleme ve Normalleştirme (Addition and Normalization): Çok başlı dikkat çıktısı, girdi gömmeleriyle birleştirildikten sonra, bir ekleme işlemine tabi tutulur. Bu ekleme, dikkat çıktısını orijinal gömme vektörlerine ekler. Ardından, normalleştirme işlemi uygulanır, yani çıktıların ölçeğini ve dağılımını düzeltir.

İleri Besleme Ağları (Feed-Forward Networks): Transformatör mimarisinde, her bir dikkat bloğunun ardından bir ileri besleme ağı bulunur. İleri besleme ağları, her sözcüğün gömme vektörünü işleyerek daha karmaşık ve yüksek boyutlu bir uzayda temsil etmeye yardımcı olur.

Kodlayıcı ve kod çözücü: Kodlayıcı, giriş metnini işleyerek anlamsal temsiller oluştururken, kod çözücü oluşturulan temsilleri kullanarak hedef metni oluşturur.



Şekil-5: Transformatör mimarisi bileşenleri

BERT ilk olarak Google AI Language'daki araştırmacılar tarafından yayınlanan bir makalede ortaya çıkmıştır [37]. BERT'in temel teknik yeniliği, dil modellemeye popüler bir dikkat modeli olan Transformatör'ün iki yönlü eğitiminin uygulanmasıdır. Bu yeni yaklaşım, bir metin dizisini yalnızca soldan sağa değil, hem soldan sağa hem de sağdan sola bakarak incelemeyi mümkün kılmaktadır. Bu şekilde, dil modeli daha derin bir dil bağlamını ve akışını anlama yeteneğine sahip olur. Bu yöntem önceki tek yönlü dil modellerine göre daha kapsamlı bir anlam çıkarma sağlamaktadır. Yapılan bir çalışmada araştırmacılar, bu iki yönlü eğitimi daha önce mümkün olmayan modellerde gerçekleştiren Maskeli Dil Modeli (MLM) adında yeni bir teknik sunmaktadır [38].

Model-2'de veri setinin %80'ı eğitim verisi, %10'u test verisi ve %10'u doğrulama verisi olarak ayrılmıştır. Geliştirilen modelde veri seti yüklendikten sonra, makale özetlerini ve anahtar sözcükleri belirteçlere dönüştürmek için bir tokenizer kullanılmıştır. Burada kullanılan tokenizer, Türkçe metinler için önceden eğitilmiş bir BERT modeli olan "dbmdz/bert-base-turkish-128k-uncased" modeline aittir. Bu model, metinlerin dilbilgisi yapısını ve sözcük anlamlarını belirlemek için kullanılır. Modelin Türkçe metinler üzerindeki performansı oldukça yüksektir ve birçok DDİ görevinde etkili sonuçlar vermektedir. Bert-base-turkish-128k-uncased modelin mevcut sürümü, farklı derlemler bir araya getirilerek eğitilmiştir. Bu derlemler; Türkçe OSCAR derleminin filtrelenmiş ve cümlelere bölünmüş bir sürümü, güncel bir Wikipedia dökümü, çeşitli OPUS derlemleri ve Kemal Oflazer tarafından sağlanan özel bir derlemdir. Elde edilen eğitim derlemi 35GB boyutunda ve 4.404.976.662 belirtece sahiptir ve model için 128.000 sözcük haznesi kullanılmıştır. Bu veri işleme ve tokenize adımları, BERT modelinin girdi formatını karşılamak ve modelin doğru şekilde çalışmasını sağlamak için önemlidir. Hazırlanan belirteç dizileri, daha sonra modelin eğitimi veya tahminleme sürecinde kullanılmıştır.

Tokenize işleminden sonra Kodlayıcı-Kod Çözücü mimarisini kullanarak otomatik ASÇ yapmak için Transformers kütüphanesinin EncoderDecoderModel sınıfı kullanılmıştır. "dbmdz/bert-base-turkish-128k-uncased" modelini hem kodlayıcı hem de çözücü olarak kullanarak önceden eğitilmiş bir modelden yeni bir kodlayıcı-çözücü modeli oluşturulmuştur.

Daha sonra bu modeli, metinler ve onlara karşılık gelen anahtar sözcükler ile eğitebilmek için Transformers kütüphanesinden Seq2SeqTrainer sınıfı kullanılmıştır. Seq2SeqTrainer, Transformer kütüphanesinin Trainer sınıfının kodlayıcı-kod çözücü modeller için genişletilmiş halidir. Bu model metin özetleme ve ASÇ gibi çoğu dizi girdisinden dizi çıktısı üretme görevinde, modelin performansını doğrulamak için gerekli olan üretme (generate) işlevinin değerlendirme sırasında kullanılmasına izin verir.

Model-2'nin eğitimi 3,5 saat sürerken, test verisi üzerinde anahtar sözcük tahmin etmesi yaklaşık olarak 15 dakika sürmüştür.

4. Deneysel bulgular

Model-1'in tahminleme (prediction) işlemi sonucunda bazı metinlerde aynı anahtar sözcükler çıkarılmış olsa da bazı metinlerde farklı ve daha uygun anahtar sözcüklere rastlanmıştır. Çizelge-2'de örnek bir metnin özgün anahtar sözcükleri ve Model-1'in tahmin ettiği sözcükler görülmektedir. Tahminleme tamamlandıktan sonra, modeli yeniden eğitmeye gerek kalmadan kullanabilmek adına geliştirilen model ve kodlayıcı-çözücü modeli '.h5' dosya uzantısı şeklinde kaydedilmiştir. Sonraki testlerde bu modeller üzerinden denemeler yapılmıştır. Eğitim ve test sonrasında elde edilen sonuçların değerlendirilmesi ROUGE (Recall Oriented Understudy for Gisting Evaluation) ölçütü ile yapılmıştır [39]. Bu değerlendirme ölçütü özellikle metin özetleme çalışmalarında yaygın olarak kullanılmakla birlikte ASÇ görevi için de uygun olmaktadır.

Çizelge-2: Özgün metin, özgün anahtar sözcükler ve modeller ile tahmin edilen anahtar sözcükler

Özgün metin	Genelkurmay Başkanlığı'ndan insansız hava araçlarının katıldığı operasyonlar ile ilgili bir açıklama yapıldı. Açıklamada yapılan operasyonda sivil vatandaşlara zarar verecek bir uygulama olmadığının altı çizilerek şu ifadelere yer verildi; "Türk Silahlı Kuvvetleri bugüne kadar hiçbir sivil, masum vatandaşımıza yönelik zarar verecek uygulama içinde olmamıştır. 31 Ağustos 2017 tarihinde Hakkari Oğulköy kırsalında SİHA ile icra edilen operasyonda da aynı esas ve usuller kullanılmıştır. İcra edilen faaliyetler kapsamında, 1 Ocak-10 Eylül 2017 tarihleri arasında insansız hava aracı/insanlı keşif uçağı (İHA/İKU) destekli toplam 562 kara ve hava operasyonu gerçekleştirilmiştir. Bu operasyonlar neticesinde yurt içinde 986, sınır ötesinde 867 terörist olmak üzere toplam bin 853 terörist etkisiz hale getirilmiştir. İcra edilen operasyonlarda toplam 887 sığınak, barınak imha edilmiş, bin 105 mayın/EYP, bin 943 değişik tipte silah ve 638 bin 462 muhtelif cins ve çapta mühimmat ele geçirilmiştir.
Özgün anahtar sözcükler	Genelkurmay Başkanlığı, SİHA, açıklama
Tahmin edilen anahtar sözcükler	
Model-1	tsk genelkurmay
Model-2 (Kök bulma işlemi olmayan)	genelkurmay başkanlığı terörist
Model-2 (Kök bulma işlemi olan)	genelkurmay başkanlık terörist

ROUGE ölçütü 0-1 aralığında değerler almaktadır. Tahmin edilen anahtar sözcükler ile metnin anahtar sözcükleri arasındaki eşleşen sözcük sayısının artışı ROUGE değerini 1'e yaklaştırmaktadır. ROUGE-N, ROUGE-L, ROUGE-S, ROUGE-W ve ROUGE-SU olmak üzere beş farklı ROUGE ölçütü bulunmaktadır. Geliştirilen modelin değerlendirilmesi ROUGE-N ölçütü üzerinden N=1 alınarak, yani ROUGE-1 ölçütü ile yapılmıştır. ROUGE-N ölçütü, modelin önerdiği anahtar sözcükler ile metnin anahtar sözcükleri arasındaki n-gram benzerliğe bakar. Bu ölçütün hesaplanmasına ait formül (1)'de verilmiştir.

$$ROUGE - N = \frac{\sum c \in ASK \sum gram_n \in c Hesap_{Eşleşme}(gram_n)}{\sum c \in ASK \sum gram_n \in c Hesap(gram_n)} \quad (1)$$

Bu formülde ASK bütün metinlerin anahtar sözcük listesinin kümesini göstermektedir. C değişkeni, ASK kümesindeki herhangi bir anahtar sözcük listesini temsil eder. Hesap($gram_n$) fonksiyonu ilgili anahtar sözcük listesindeki n-gram sayısını verirken, $Hesap_{Eşleşme}(gram_n)$ fonksiyonu metindeki anahtar sözcükler ile tahmini yapılan anahtar sözcüklerin eşleşme sayısını hesaplar.

ROUGE-1 ölçütü için hesaplamalar bir Python Kütüphanesi olan Rouge-Score üzerinden yapılmıştır. Bu skor hesaplama işlemi veri setindeki metinler üzerinde gerçekleştirildikten sonra, geliştirilen Model-1'de Çizelge-3'de verilen başarım değerleri elde edilmiştir.

Model-2'nin tahminleme aşamasında, eğitilmiş olan yeni modelin en son checkpoint'i kullanılarak %10 test verisi üzerinde anahtar sözcük üretilmesi istenmiştir. Çizelge-2'de bu model ile yapılan iki farklı deney üzerinden bir metnin özgün anahtar sözcükleri ve tahmin ettiği sözcükler görülmektedir. Model-2'nin tahmin ettiği anahtar sözcükler ile makalelerin özgün anahtar sözcüklerinin karşılaştırması ROUGE-1, ROUGE-2, Meteor, Bleu, SacreBleu ve BertScore skorlama yöntemleri kullanılarak yapılmıştır. Bu skorlama yöntemleri çeşitli DDİ görevlerinde kullanılan değerlendirme yöntemleridir. Bu skorlar, ASÇ görevinde, üretilen anahtar sözcüklerin metinlerin özgün anahtar sözcüklerine ne kadar benzediğini ölçmek için aşağıda verildiği şekilde kullanılır:

ROUGE-1: Üretilen ve özgün anahtar sözcükler arasındaki örtüşme oranını değerlendirir.

ROUGE-2: İki veya daha fazla sözcüğün ardışık olarak eşleşme oranını değerlendirir.

Meteor: Üretilen ve özgün anahtar sözcükler arasındaki kesin eşleşmeyi ve benzerliği ölçer.

Bleu: Üretilen ve özgün anahtar sözcükler arasındaki n-gram benzerliğini değerlendirir.

SacreBleu: Bleu metriğinin geliştirilmiş bir sürümüdür ve dil işleme görevlerinde yaygın olarak kullanılır.

BertScore: Üretilen ve özgün anahtar sözcükler arasındaki benzerliği BERT (Bidirectional Encoder Representations from Transformers) modeli ile hesaplar.

Model-2'de elde edilen başarım değerleri Çizelge-3'de verilmiştir.

Çizelge-3: Modellerin performans metrik sonuçları

		Model-1	Model-2 (Kök bulma işlemi olmayan)	Model-2 (Kök bulma işlemi olan)
ROUGE-1	F-1 Score	0,38	0.394	**0.399**
	Precision	**0,49**	0.443	0.455
	Recall	0,35	**0.422**	0.421
ROUGE-2	F-1 Score	-	0.200	**0.242**
	Precision	-	**0.239**	0.202
	Recall	-	**0.207**	0.199
	Meteor	-	**0.275**	**0.275**
	Bleu	-	0.065	**0.067**
	SacreBleu	-	**18.541**	18.392
	BertScore		**0.674**	0.670

4. Sonuç

Bu çalışmada, Türkçe metinlerin anahtar sözcüklerini belirlemek için doğal dil işleme ve derin öğrenme yöntemlerine dayalı bir yaklaşım benimsenmiştir. Bu amaçla, öncelikle LSTM katmanlı bir Seq2seq model geliştirilerek metinlerin anahtar sözcükleri otomatik olarak belirlenmeye çalışılmıştır. Model, eğitim sürecinde özgün anahtar sözcüklerle tahmin edilen anahtar sözcükler arasındaki benzerliği ölçen ROUGE-1 ölçütü üzerinden değerlendirilmiştir. Geliştirilen modelin ROUGE-1 ölçütü üzerinden 0,38 F-1 değerine ulaştığı görülmüştür. Geliştirilen diğer model Transformatör mimarisini temel alan BERT ile oluşturulmuş bir derin öğrenme modelidir. Bu modelin deneysel amaçlı iki farklı şekilde eğitimi yapılmıştır. İlk yapılan deneyde kök bulma işleminin uygulanmadığı veri seti ile eğitilen modelde ROUGE-1 ölçütünde 0,394 F-1 skoru elde edilmiştir. Kök bulma işleminin uygulandığı veri seti ile eğitilen modelde ise aynı model ROUGE-1 ölçütünde 0,399 F-1 skoruna ulaşılmıştır. Model-2'de elde edilen bu sonuçlara göre kök bulma işleminin model başarımını olumlu düzeyde etkilediği görülmektedir. Çizelge-3'de verilen diğer ölçütlerdeki değerler ile de, kök bulma işleminin etkisi kısmen görülebilmektedir. Her iki modelin testi ile elde edilen sonuçlara göre ise Transformatör mimarisi, LSTM mimarisinden daha başarılı sonuçlar vermiştir. Aynı zamanda Transformatör mimarisi ile elde edilen anahtar sözcüklerin diğer mimariye göre metin içeriğini kısmen daha uygun temsil ettiği gözlemlenmiştir.

ASÇ alanında yapılan ve derin öğrenme yöntemleri kullanması yönüyle benzer olarak nitelenebilecek başka bir çalışmada ROUGE-1 ölçütü üzerinden 0,39 F-1 değerine ulaşıldığı ifade edilmiştir [21]. Tasarladığımız BERT tabanlı Seq2Seq model ile, [21]'deki modelin başarımı kıyaslandığında elde edilen 0.399 F-1 değeri ile yakın bir başarıma ulaşıldığı görülmektedir.

Gelecek çalışmalarda, modelin performansını yükselterek ASÇ görevini daha başarılı kılacağı düşünülen iyileştirmeler şunlardır:

- Eğitim veri seti boyutunu arttırılabilir ve model daha büyük veri setini işleyebilecek şekilde optimize edilebilir.
- Sözcük gömme yöntemi olarak Keras kütüphanesinin gömme katmanı yerine Word2Vec, GloVe gibi farklı gömme yöntemleri kullanılabilir.
- Test dizisini çözmek (decode) için açgözlü yaklaşım (argmax) yerine ışın arama (beam search) stratejisi kullanılabilir. Işın arama daha geniş bir arama alanı kullanmakta ve dizi oluştururken önceki adımların tahminlerini de dikkate alarak olasılıkları hesaplamaktadır. Ancak burada ışın arama yönteminin daha fazla hesaplama kaynağı gerektireceği de göz önünde bulundurulmalıdır.
- Modelin eğitim sürecinde farklı optimizasyon algoritmaları (Adam, RMSprop, Adagrad vb.) kullanılarak ve öğrenme hızları denenerek modelin eğitim süreci optimize edilebilir.

Bu öneriler doğrultusunda gerçekleştirilecek çalışmalar, geliştirilen modelin Türkçe metinlerin anahtar sözcüklerini belirleme performansını daha da iyileştirebilir. Bu süreçte elde edilen sonuçların karşılaştırılması, daha etkili ve başarılı bir model geliştirme sürecine katkı sağlayacaktır.

Kaynakça

[1] Hashemzahde, B. vd., *Improving keyword extraction in multilingual texts,* Int J Electric Comput Eng, 2020, 10:5909-5916.

[2] Papagiannopoulou, E., Tsoumakas, G., *A review of keyphrase extraction*, CoRR, 2019.

[3] Witten, I. H., Paynter, G. W., Frank E., Gutwin, C., NevillManning, C. G., *Kea: Practical Automatic Keyphrase Extraction*, In Proceedings of the 4th ACM Conf. of the Digital Libraries, 1999, Berkeley, CA, USA.

[4] Turney, P., *Learning algorithms for keyphrase extraction*, Information Retrieval, 2000, 2:303–336.

[5] Zhang, Q., Wang, Y., Gong, Y., *Keyphrase extraction using deep recurrent neural networks on Twitter*, In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2016, Austin, Texas, pp. 836–845.

[6] Zhang, Y., Yang, F., Xiao, W., *Deep keyphrase generation with a convolutional sequence to sequence model*, In Proceedings of the 4th International Conference on Systems and Informatics, Hangzhou, 2017, China, pp. 1477–1485.

[7] Chen, W., Gao, Y., Zhang, J., King, I., Lyu, M. R., *Title-guided encoding for keyphrase generation*, In Proceedings of AAAI Conference on Artificial Intelligence, 2019, pp. 6268–6275.

[8] Mihalcea, R., Tarau, P., TextRank: Bringing order into text, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP '04), 2004, Barcelona, Spain, pp. 404-411.

[9] Wan, X., Xiao, J., *Single document keyphrase extraction using neighborhood knowledge*, In Proceedings of the 23rd AAAI Conference on Artificial Intelligence, 2008, pp. 855-860.

[10] Gollapalli, S. D., Caragea, C., *Extracting keyphrases from research papers using citation networks*, In Proceedings of the 28th AAAI Conference on Artificial Intelligence, 2014, pp. 1629-1635.

[11] Rose, S., Engel, D., Cramer, N., Cowley, W., *Automatic keyword extraction from individual documents*, Text Mining:Applications and Theory, 2010, pp. 1-20.

[12] Liu, Z., Li, P., Zheng, Y., Sun, M., *Clustering to find exemplar terms for keyphrase extraction*, In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, 2009, Singapore, pp. 257–266.

[13] Bougouin, A., Boudin, F., Daille, B., *TopicRank: Graph-based topic ranking for keyphrase extraction*, In Proceedings of the 6th International Joint Conference on Natural Language Processing, 2013, pp. 543-551.

[14] Zha, H., *Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering*, In Proceedings of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002, pp. 113-120.

[15] Tomokiyo, T., Hurst, M., *A language model approach to keyphrase extraction*, In Proceedings of the ACL 2003Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18. Association for Computational Linguistics MWE'03, 2003, pp. 33-40.

[16] Pala, N., Cicekli, I., *Turkish keyphrase extraction using kea*, In Proceedings of the 22nd International Symposium on Computer and Information Sciences (ISCIS 2007), 2007, Ankara, pp. 1-5.

[17] Kalaycilar, F., Cicekli, I., *TurKeyx: Turkish keyphrase extractor*, In Proceedings of the 23rd International Symposium on Computer and Information Sciences (ISCIS 2008), 2008, Istanbul, pp. 1-4.

[18] Ozdemir, B., Cicekli, I., *Turkish Keyphrase Extraction Using Multi-Criterion Ranking*, In: 24th International Symposium on Computer and Information Sciences, 2009, pp. 269-273.

[19] Müngen, A. A., Kaya, M., *Extracting abstract and keywords from context for academic articles*, Social Network Analysis and Mining 8, 2018, pp. 1-11.

[20] Yıldız, O., *Metin Madenciliğinde Anahtar Kelime Seçimi Bir Üniversite Örneği*, Yönetim Bilişim Sistemleri Dergisi, 2(3), 2017, pp. 29-50.

[21] Ayan, E. T., Arslan, R., Zengin, M. S., Duru, H. A., Salman, S., Bardak, B., *Turkish Keyphrase Extraction from Web Pages with BERT*, In 29th Signal Processing and Communications Applications Conference, 2021.

[22] Göz, F., Mutlu, A., Küçük, K., Temur, M., Gün, A., *Effect of Centrality Measures for Keyword Extraction from Turkish Documents*, In 29th Signal Processing and Communications Applications Conference, 2021.

[23] Yıldız, A. M., *Keyword Extraction with Textrank and Tfidf From Turkish Articles*, The International Informatics Congress 2022 (IIC2022), 2022.

[24] Erzurumlu, H. Y., Akgul, Y. S., *Adaptive Keyword Extraction Service for Turkish*, IntelliSys (2) 2022, 2022, pp. 495-506.

[25] Natural Language Toolkit, https://www.nltk.org

[26] NumPy Kütüphanesi, https://numpy.org

[27] Pandas Kütüphanesi, https://pandas.pydata.org

[28] Tensorflow Kütüphanesi, https://www.tensorflow.org

[29] Rouge Scorer Kütüphanesi, https://pypi.org/project/rouge-scorer

[30] Öztürk, S., Sankur, B., Güngör, T., Yılmaz, M. B., Köroğlu, B., Ağın, O., İşbilen, M., Ulaş, Ç., Ahat, M., *Türkçe Etiketli Metin*

Derlemi, 22. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU-14), 2014.

[31] Metin Özetleme-Anahtar Kelime Çıkarma Veri Seti, http://buyukveri.firat.edu.tr/veri-setleri

[32] Akın, Ahmet A., Zemberek-NLP, Github. https://github.com/ahmetaa/zemberek-nlp

[33] Sutskever, I., Vinyals, O., Le, Q. V., *Sequence to sequence learning with neural networks*, Proceedings of NeurIPS, 2014, pp. 3104-3112.

[34] Keras Kütüphanesi, https://keras.io

[35] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., Attention is All You Need, In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS), Long Beach-California-A.B.D, 6000-6010, 4-9 Aralık, 2017.

[36] Unzueta, D., Transformers: An Overview of the Most Novel AI Architecture,2022. https://towardsdatascience.com/transformers-an-overview-of-the-most-novel-ai-architecture

[37] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 1, 2019, pp. 4171-4186.

[38] Horev, R., BERT Explained: State of the art language model for NLP, 2018.

https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270

[39] Lin C.-Y., *ROUGE: A Package for Automatic Evaluation of Summaries*, Text Summarization Branches Out, Association for Computational Linguistics, 2004, pp. 74-81.