

DERİN ÖĞRENME TEMELLİ OTOMATİK YARDIM MASASI SİSTEMİMetin YILMAZ ¹, Efnan ŞORA GÜNAL ^{2,*}¹ Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskişehir, ORCID No: <http://orcid.org/0000-0001-9478-4114>² Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskişehir, ORCID No: <http://orcid.org/0000-0001-6236-174X>

Anahtar Kelimeler	Öz
Derin Öğrenme Doğal Dil İşleme Metin Sınıflandırma Yardım Masası	<i>Yardım masası, bir organizasyonun çalışanlarına veya müşterilerine merkezi bilgi ve destek yönetimi hizmeti sağlayan iletişim noktasıdır. Organizasyonun verimliliği açısından, yardım masasına gelen taleplerin doğru kategorilere ayrılarak, doğru kişilere ve zamanında yönlendirilmesi büyük önem arz etmektedir. Bu sebeple, bu çalışma kapsamında, derin öğrenmeye dayalı otomatik bir yardım sistemi önerilmiştir. Önerilen sistem, talepleri, başlıklarında yer alan cümlelere göre otomatik olarak uygun kategorilere ayırmaktadır. Bu işlem için kelime gömme (ing. word embedding) yöntemi kullanılmıştır. Metin ön işleme adımlarından sonra, gömme, düzleştirme (ing. flatten) ve yoğun (ing. dense) oluşan üç katmanda öğrenme gerçekleştirilerek, yardım masası taleplerinin ait olduğu kategori belirlenmektedir. Bu amaçla, kurumsal bir şirkete ait BT yardım masası talepleri kullanılmıştır. Dokuz farklı kategoride toplam 28.104 talepten oluşan veri kümesi, %60 eğitim, %20 doğrulama ve %20 test kümesine ayrılmıştır. Yapılan deneyler sonucunda %98'e ulaşan sınıflandırma doğruluğu, önerilen modelin otomatik bir yardım masası sistemi için iyi bir aday olduğunu ortaya koymuştur.</i>

AUTOMATED HELP DESK SYSTEM BASED ON DEEP LEARNING

Keywords	Abstract
Deep Learning Natural Language Processing Text Classification Help Desk	<i>A help desk is an organization's point of contact that provides a centralized information and support management service to its employees or customers. For the efficiency of the organization, it is of great importance that the queries coming to the help desk are grouped into the correct categories and directed to the right people on time. Therefore, in this study, an automatic help desk system based on deep learning is proposed. The proposed system automatically categorizes queries according to the sentences in their titles. Word embedding method was used for this process. After the text preprocessing steps, learning is performed in three layers (embedding, flatten, and dense) and the category to which the help desk queries belong is determined. For this purpose, IT help desk queries belonging to a corporate company were used. The dataset, consisting of a total of 28.104 requests in nine different categories, is divided into 60% training, 20% validation, and 20% test set. As a result of the experiments, the classification accuracy reaching 98% revealed that the proposed model is a good candidate for an automated help desk system.</i>

Araştırma Makalesi	Research Article
Başvuru Tarihi : 19.12.2021	Submission Date : 19.12.2021
Kabul Tarihi : 22.06.2022	Accepted Date : 22.06.2022

* Sorumlu yazar; e-posta: esora@ogu.edu.tr

Bu eser, Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) hükümlerine göre açık erişimli bir makaledir.This is an open access article under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>).

1. Giriş

Günümüzde, iç ve dış birimlere hizmet veren kurumlar bünyesinde, yardım masası uygulamaları üzerinden gelen talepleri değerlendirmek ve ilgili birimlere aktarmak, sorunların hızlı giderilmesi açısından büyük önem taşımaktadır. Özellikle, bilişim teknolojileri gibi farklı alanlara hizmet veren birimlerde, taleplerin hızlı çözülerek zaman kaybını en aza indirmek gerekmektedir. Gelen talebin doğru kategoriye ayrılarak, doğru kişiye zamanında yönlendirilmesi önem arz etmektedir. Bu işlem genel olarak, talepleri okuyarak ilgili birime yönlendiren birinci seviye destek hizmetinden sorumlu bir personel tarafından veya metin içeriğini analiz edebilen yönlendirme yazılım uygulaması ile gerçekleştirilir.

Yardım masası uygulamasının atama işlemini otomatik olarak yapabilmesi için, talebi açan kullanıcının talep şablonunu doğru belirleyerek, ilgili tüm alanları doldurması gerekmektedir. Ancak, talep açan kullanıcıların bu konuda ihmalkâr davrandığı ya da talebinin tam olarak hangi şablona ait olduğunu bilmediği durumlar çıkabilmektedir. Bu sebeple, giriş yapan kişinin, en yakın gördüğü bir şablondan talebi açarak, ilgili alanları da boş bıraktığı ya da doldurulması zorunlu alanları rastgele karakterler ile doldurduğu görülmektedir. Talepleri kategorize etmek için, bir personel görevlendirme konusunda ise; eğer görevli kişi destek personeli ise ve tüm birinci seviye destekten o kişi sorumlu ise problem yaşanmamaktadır. Bu şekilde, küçük ölçekli işletmelerde sorunun üstesinden gelinebilmektedir. Ancak, çok farklı alanlara hizmet verilen daha büyük işletmelerde, görevli personel kendi alanı dışında bir taleple karşılaştığında, talebin kategorisini belirlemede güçlük çektiğinde problemin çözümünde zaman kaybı yaşanmaktadır. Bu durum, aynı zamanda, işyeri verimliliğinin ve iç müşteri memnuniyetinin de azalmasına sebep olmaktadır. Bu nedenle, yardım masası taleplerinin sınıflandırılmasının makine öğrenmesi destekli metin analizi yöntemleri ile yapılması, insan kaynaklı sorunlara da çözüm getirecektir. Bu çalışma kapsamında, yukarıda ifade edilen soruna bir çözüm sunmak için, hiçbir alan doldurulmasa dahi, sadece başlık metni kullanılarak, kurum bünyesinde önceden açılmış taleplerin türleri ile eğitilmiş model kullanılarak, kategorilere ayırma hedeflenmiştir. Bu işlem için, derin öğrenme teknikleri kullanılarak metin sınıflama gerçekleştirilmiştir.

Metin sınıflandırmanın temel amacı, verilen metni denetimli öğrenme algoritmaları kullanarak önceden tanımlanmış kategoriler halinde sınıflandırmaktır. Son yıllarda, makine öğrenmesinin bir yan kolu olarak sıklıkla kullanılmaya başlayan derin öğrenme algoritmaları, bir veri setinden alınan örnekler kullanılarak, metinler ve kategoriler arasında ilişkiyi belirtmek için eğitilebilir. Eğitilen model yardımıyla, verilen metnin kategorisi tahmin edilebilir. Bir metin sınıflandırma görevi, belgelerin ayrıştırılması,

belirteçlere ayırma, kök oluşturma, durak kelimelerin kaldırılması, metin terim matrisinin temsil edilmesi, öznitelik seçimiyle birlikte, eğitim ve test yoluyla, en iyi sınıflandırıcı modellerinin seçilmesinden oluşur (Borko ve Bernick, 1963).

Son zamanlarda, metin sınıflandırma (kategorizasyon), farklı dijital platformlarda bulunan çok sayıda belge ve metin nedeniyle birçok araştırmacının dikkatini çekmiştir. Kilimci ve Akyokus (2018), çalışmalarında, kelime yerleştirmeleri ve metin sınıflandırması için temel sınıflandırıcılar topluluğu yararına farklı belge gösterimleri kullanmıştır. Sekiz farklı veri kümesinde, üç farklı belge gösterimi yöntemi kullanarak, beş heterojen temel öğrenme ve iki entegrasyon yönteminden oluşan bir topluluğun performansını değerlendirilmiştir. Sınıflandırıcıların tekli kullanımda ortalama sınıflandırma doğruluk sırası şu şekilde bulunmuştur: random forest (RF) > convolutional neural network (CNN) > multinomial naïve Bayes (MNB) > support vector machine (SVM) > multivariate Bernoulli naïve Bayes (MVNB). CNN'in temel öğrenme setine dahil edilmesiyle, Türkçe Hürriyet ve AHaber veri kümeleri için sırasıyla %89,55 ve %92,70 ile daha iyi doğruluk oranları elde edilmektedir. Kowsari ve diğ. (2017) çalışmalarında, belge sınıflaması problemini, mevcut belge sınıflandırma yöntemlerinden farklı bir şekilde ele alınmıştır. Metin sınıflandırmada, Hiyerarşik Derin Öğrenme (ing. Hierarchical Deep Learning for Text classification - HDLTex) adında bir yaklaşım kullanarak hiyerarşik sınıflandırma gerçekleştirilmektedir. HDLTex, belge hiyerarşisinin her seviyesinde derin öğrenme mimarileri yığınları kullanır. Bu yaklaşım, geleneksel belge sınıflandırma yöntemlerine göre gelişmiş doğruluk sergilemektedir.

Metin sınıflandırmada, son yıllarda sıklıkla kullanılan, etkili yöntemlerden biri olan kelime gömme ise, anlamsal olarak benzer olan kelimelerin, birbirine daha yakın vektörler halinde temsil edilmesidir (Jurafsky ve Martin, 2000). (Li, Hu, Cui ve Hu, 2018), patent sınıflandırma problemini çözmeye, evrimsel sinir ağları (ing. Convolutional Neural Networks - CNN) otomatik hiyerarşik öznitelik çıkarımı ile derin sinir ağlarının güçlü modelleme gücünden yararlanarak, kelime vektör gömmeye dayalı DeepPatent algoritması kullanılmıştır. İlk olarak, sözcük belirteçlerini etkili bir şekilde öznitelik vektörlerine dönüştürür. Daha sonra, sözcük düzeyi vektörleri metin düzeyi yoğun matrisini oluşturmak için birleştirilmiştir. Deneysel sonuçlar, özet bölümün patent sınıflandırmasında, anahtar bilgilere sahip olduğu sonucu elde edilmiş olup, özet bölümüne ilave olarak başlık bölümünün birlikte kullanımı ile daha yüksek başarımlar elde edilebileceği doğrulanmıştır. ALRashdi ve O'Keefe (2019), çalışmalarında farklı sinir ağı mimarileri ve farklı genel amaçlı kelime yerleştirmeleri kullanmıştır. Tweet sınıflandırma modellerinin performansını iyileştirme yeteneklerini araştırmak için, farklı sinir ağlarının genel amaçlı ve

alana özgü kelime düğümleriyle birlikte uygulanması incelenmiştir. Çalışmada kullanılan CrisisNLP veri setinde, dört tweet sınıflandırma modelini değerlendirip, karşılaştırılabilir sonuçlar elde edilmiştir. Sonuçlar, GloVe gibi genel amaçlı kelime gömme tekniği, özellikle çift yönlü uzun ömürlü kısa-dönem belleği (ing. Bidirectional Long Short-Term Memory - Bi-LSTM) ile, sonuçların %62,04 F1-skoru ile en yüksek performansı bildirdiği alana özgü kelime gömme yerine kullanılabilirliğini göstermiştir.

GloVe yönteminin yanı sıra, günümüzde Mikolov, Chen, Corrado ve Dean (2013) tarafından önerilen Word2Vec, en popüler kelime gömme tekniğidir. Kocmi ve Bojar (2017), çalışmasında, derin ağlarda kullanılan kelime gömmeler için çeşitli rastgele ve önceden eğitilmiş ağlarla, bunların hem tekrarlayan hem de evrimsel mimarilerle dört doğal dil işleme (DDİ) görevi üzerindeki performans etkilerini incelemişlerdir. Önceden eğitilmiş kelime gömmelerin, özellikle öğrenme hızı göz önüne alındığında, rastgele başlatmadan biraz daha iyi olduğunu gözlemlenmiştir. Büyük bir veri üzerinde eğitilen "Word2Vec" kelime gömme, vektör uzayındaki sözcüksel ilişkiler ve aritmetik işlemler arasında uyuma göstermektedir. Zhang ve diğ. (2018), çalışmalarında, bibliyometrik verilerden konuları etkili bir şekilde ayıklayan bir çözüm oluşturmak için, bir kelime gömme modeliyle bir araya gelen yeni bir çekirdek K-means kümeleme yöntemi önermektedir. Bu yöntemin iki bibliyometrik veri kümesindeki dört kümeleme taban çizgisi (örneğin, k-means, bulanık c-means, ana bileşen analizi ve konu modelleri) ile karşılaştırılmasının deneysel sonuçları, nispeten geniş bir disiplin yelpazesinde veya belirli bir etki alanında etkinliğini göstermektedir. Yapılandırılmamış metin verilerinden, konuları etkili bir şekilde ayıklamada bir çözüm oluşturmak için, kelime gömmelerini içeren bir çekirdek k-means kümeleme metodolojisi önerilmektedir. Word2Vec yöntemini derin öğrenmenin temsili bir tekniği olarak kullanılmıştır. Word2Vec yönteminin hesaplama verimliliğini artırmak için sinir ağlarının kullanımını basitleştirmeye çalışmıştır.

Metin sınıflandırmanın kullanıldığı bir başka çalışmada ise, bir kuruma ait sahadaki anomali içeren arıza ya da sel gibi olağanüstü durumlarda, gelen metin verilerinin içerikleri hızlı bir şekilde analiz edilerek, doğru çözüm üretebilen kararlar alınması sağlanmıştır. Bu ortamda, gelen alarmlar gerçekten oluşmadan önce doğru bir şekilde tahmin edilebilirse, operatörler zamanında düzeltici eylemlerde bulunan anormal davranışları ele alma ve muhtemelen kaçınma şansına sahip olabilir. Cai, Palazoglu, Zhang ve Hu (2019), çalışmalarında, doğal dil işlemede derin öğrenme uzun ömürlü kısa-dönem belleği (ing. Long Short-Term Memory - LSTM) uygulamasından esinlenilerek, bir süreç ayarındaki bir sonraki uyarı tahmin etmede, kelime gömme ve tekrarlayan sinir ağlarına dayalı bir uyarı tahmin

yöntemi sunmaktadır. Bu, hem alarm yönetimine yeni bir yaklaşımı hem de bu soruna doğal dil işleme ve derin öğrenme tekniklerinin yeni bir uygulamasını temsil eder. Metin madenciliği, biyomedikal araştırmalar için de önemli bir araç haline gelmiştir. En temel metin madenciliği görevi, genler, kimyasallar ve hastalıklar gibi biyomedikal adlandırılmış varlıkların Varlık İsmi Tanıma (ing. Named Entity Recognition - NER) tanınmasıdır. Geçerli NER yöntemleri, varlık türlerinin belirli yüzey özelliklerini, tipik yerel bağlamın özelliklerini, arka plan bilgisini ve dil bilgilerini yakalamaya çalışan önceden tanımlanmış özelliklere dayanır. Habibi, Weber, Neves, Wiegandt ve Leser (2017), çalışmasında, derin öğrenme ve istatistiksel kelime gömmelere dayanan tamamen genel bir yöntemin (ing. long short-term memory network-conditional random field, LSTM-CRF) son teknoloji varlığa özgü NER araçlarından ve genellikle büyük bir farkla daha iyi performans göstermektedir. Bu amaçla, LSTM-CRF yönteminin beş farklı varlık sınıfını kapsayan 33 adet veri kümesindeki performansını, sınıfının en iyisi NER araçları CRF uygulamasıyla karşılaştırmışlardır.

Duygu sınıflandırma teknikleri, kullanıcı görüşlerini analiz etmek için, yaygın olarak kullanılmaktadır. Geleneksel denetimli öğrenme yöntemlerinde, alanın tam olarak anlaşılmasını gerektiren özniteliklere ihtiyaç vardır. Sosyal medya paylaşımları genellikle çok kısa olduğundan, etkili sınıflandırma için öznitelik eksikliği vardır. Böylece, kelime gömme modelleri, çeşitli bağlamlarda farklı kelimeleri öğrenmek için kullanılabilir. Kısa metinlerden duygu kutupluluğunu (ing. sentiment polarity) tespit etmede, derin öğrenme yöntemlerini kullanarak kelimelerin daha derin anlam bilimlerini keşfetmek gerekir. Wang, Liu, Luo ve Wang (2018), çalışmasında, sosyal medyada duygu sınıflandırması için kelime gömme ve LSTM etkilerini araştırmaktadır. İlk olarak, yazılardaki kelimeler, kelime gömme modelleri kullanılarak vektörlere dönüştürülür. Daha sonra, cümleler arasındaki uzun mesafeli bağlamsal bağımlılığı öğrenmek için, cümlelerdeki kelime dizisi LSTM modeline verilir. Deneysel sonuçlar, derin öğrenme yöntemlerinin yeterli eğitim verisi verildiğinde, sosyal medya bağlamında kelime kullanımını etkili bir şekilde öğrenebileceğini göstermiştir. İlk olarak, kısa metinlerdeki kelimeleri vektör olarak temsil etmek için Word2Vec tabanlı bir kelime gömme modeli kullanılır. İkincisi, LSTM, kısa metinlerdeki kelime dizisi arasındaki uzun mesafeli bağımlılığı öğrenmek için kullanılır. Son zaman noktasından elde edilen nihai çıktı, tahmin sonucu olarak kullanılır. Birkaç sosyal veri kümesinde duygu sınıflandırması deneylerinde, LSTM yönteminin performansını naive Bayes (NB) ve Extreme Learning Machine (ELM) ile karşılaştırılmıştır. Deneysel sonuçlarına bakıldığında, önerilen yöntem geleneksel olasılık modeline ve daha fazla eğitim verisine sahip sinir ağlarına göre daha iyi performans sağlayabilir.

Semberecki ve Maciejewski (2017), çalışmasında, LSTM birimleri ile derin sinir ağı kullanarak metin belgelerini sınıflandırma yöntemini sunar. Sınıflandırılacak belgeleri temsil eden öznitelik vektörleri oluşturmada farklı yaklaşımlar test edilip; Word2Vec aracını kullanarak sözcükleri vektör gösterimlerine ve bu vektör gösterimlerinin dizilerini belgelerin öznitelikleri olarak kullanılmıştır. Öznitelik vektörü belgeleri temsil etmek ve sınıflandırıcıları eğitmek için kavramsal olarak farklı bir yaklaşım sunulmuştur. Temel fark, sözcüklerin frekanslarına dayanan öznitelik vektörlerini kullanmak yerine, belgelerdeki sözcük dizilerine dayanan öznitelik vektörleri kullanmaktır. Bunun bir sonucu olarak, kullanılan sınıflandırıcı türünü de değiştirmek gerekir: LSTM ağını eğitmek için sözcük dizilerini temsil etmenin iki farklı yolu, kelimelerin basit kodlamasıyla ve vektör uzayındaki kelimeleri temsil eden Word2Vec yöntemi kullanılmaktadır. Teknik olarak, sınıflandırmaya sunulan yaklaşımda, GPU üzerinde eğitilen modellerle derin sinir ağları için Tensorflow kütüphanesine sahip Keras kullanılmıştır. Keras, modelin oluşturduğu katmanları tanımlayan sinir ağları katmanlarını sadece birkaç kod satırında oluşturma yeteneği veren Tensorflow'a ait bir kütüphanedir.

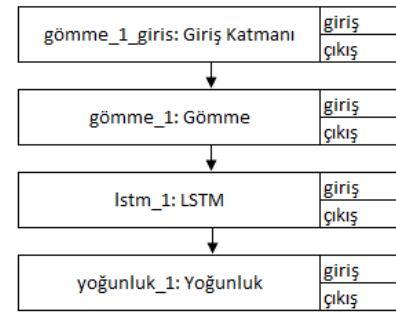
Güncel literatürden bir başka örnekte, Ibrahim, Khan, Mehmood, Asim ve Mahmood (2021), Yinelemeli sinir ağı (ing. Recurrent Neural Network - RNN), LSTM ve Kapı Özyinelemeli Geçitler (ing. Gated Recurrent Unit - GRU) ile 3 farklı şekilde eğitilmiş sinir ağları kullanarak, klinik veri seti üzerinde metin sınıflandırma yapan GHS-NET uygulaması geliştirmiştir. Bir diğer çalışmada, Liang ve Yi (2021a) üç yönlü kararlar (ing. three-way decisions - 3WD) ile politik metin sınıflaması yapmıştır. Bu yöntemde, sınıflandırma için 3WD-AdaCNN-SVM (3WD-AdaCNN modeli 3WD ve CNN modelini taban almaktadır) modelinin kombinasyonu kullanılmıştır. Liang ve diğ. (2021b) ise Uzamsal Görüş Dikkati Evrimsel Sinir Ağı (ing. Spatial View Attention Convolutional Neural Network SVA-CNN) yöntemini kullanmıştır. Araştırmacılar, bu modeli kullanarak belge sınıflandırması, duygu sınıflandırması ve tematik sınıflandırma dahil olmak üzere üç sınıflandırma görevi için beş veri kümesi (AG News Corpus (NEWS-4), Hate Speech (HATE-3), Amazon Review (AMZ-5), Movie Review (IMDB-2), Emotion Text (TWT-13)) üzerinde deney yapmıştır.

Çalışmamızda ise kelime gömme yöntemi tabanlı, yardım masasından alınan talep başlıklarının sınıflandırılmasında, son yıllarda literatürde sıklıkla tercih edilen kelime gömme tekniği temelli model eğitimi kullanılarak, talepler kategorilere ayrılmıştır. Kurumsal bir şirketin 2019-2020 yıllarına ait yardım masası verileri kullanılarak, taleplerin kategorilere ayrılmasında %89,22 F1-skor değerinde başarımlar sağlanmıştır. Veri setinin daha az sayıda kategori şeklinde ele alınmasıyla yapılan deneysel çalışmalarda ise %91,31 ve %93,42 F1-skor değerlerine ulaşılmıştır.

Çalışmanın bir sonraki bölümü olan yöntem başlığı altında, algoritmanın sunumu ve kullanılan tekniklerin açıklaması yer alırken, deneysel çalışma başlığı altında, oluşturulan veri seti hakkında bilgi verilerek, eğitim ve test sonucu elde edilen tahmin sonuçları karşılaştırmalı tablolar ile sunulmuştur. Tartışma bölümünde ise önerilen yöntemin sonuçları değerlendirilmiştir.

2. Yöntem

Bu çalışma kapsamında, ele alınan yardım masası talep içeriklerinin sınıflandırılması, derin öğrenme temelli standart bir LSTM ağı, kelime gömme yöntemi ile metin sınıflandırma problemi olarak ele alınmıştır. Şekil 1'de LSTM model mimarisi görünmektedir. Metin vektör dönüşümü için word2vec yaklaşımı kullanılmıştır. Metin analizi işlemleri için, Python dilinde Doğal Dil İşleme (DDİ, ing-Natural Language Processing, NLP) kütüphaneleri (Loper ve Bird, 2002) kullanılmıştır. Çalışmamızda, araştırma ve yayın etiğine uyulmuştur.



Şekil 1. LSTM model mimarisi

2.1. Veri Seti Hazırlama

Yöntemde kullanılan Türkçe dilindeki veri seti, bir işyerine ait bilişim teknolojileri biriminin kendisine gelen talepleri almak için kullandığı yardım masası uygulaması verileridir. 2019 ve 2020 yılına ait veriler 9 kategoride, 28.104 adet talepten oluşmaktadır. Talepler başlıklar üzerinden değerlendirilmiş, cümleler ve kategorileri kontrol edilerek düzenlenmiştir. Veri setindeki kategori dağılımı Tablo 1'de verilmiştir.

Yardım masasından alınan metin verileri, manuel incelenerek, kategorileri doğrulanmıştır. Gereksiz ve hatalı açılan talepler veri setinden çıkarılmıştır. Taleplerin kategorilerinin etiketlenmesi esnasında, kontrol edilerek, içeriği net olmayan metinler iptal edilmiştir. Böylece, modelin eğitiminde kullanılacak temiz bir veri seti oluşturulmuştur. Hazırlanan veri setinden alınan veriler, %60 eğitim, %20 doğrulama ve %20 test olmak üzere bölümlenmiştir.

Tablo 1

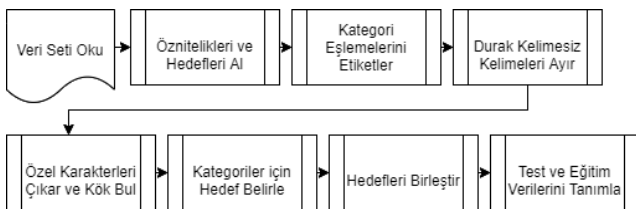
Talep kategori dağılımı

Kategori	Talep Sayısı
Donanım	5.037
E-posta	2.505
Kullanıcı İşlemleri	4.432
Kurumsal Uygulamalar	1.827
SAP	2.317
Sunucu	1.077
Veri İşlemleri	3.067
Yazıcı	1.104
Yazılım	6.738

2.2. Ön İşlem Aşamaları

Sınıflandırma öncesinde, ön işleme aşamalarında kullanılan durak sözcük çıkarma ve kök bulma için, bir Python kütüphanesi olan Natural Language Toolkit (NLTK) kullanılmıştır. NLTK, insan dili verileriyle çalışmak için Python programlama dili ile geliştirilmiş ve geliştirilmekte olan 50'nin üzerinde derlem (ing. corpus) ve sözcük kaynağı (ing. lexical resources) ile oluşturulmuş açık kaynaklı bir kütüphanedir (Loper ve Bird, 2002).

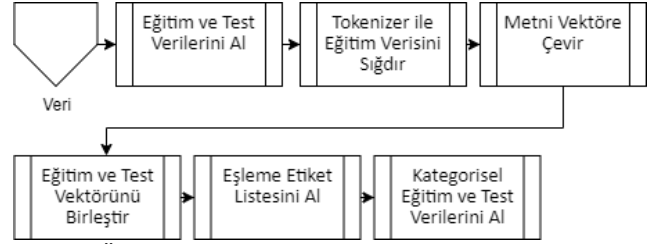
Bu kütüphanede, ön işleme, makine öğrenmesi algoritmaları, Twitter API ile işlemlerinin yapılabildiği modüller mevcuttur. Örnek olarak; bir cümledeki kelimeleri ayırma (ing. tokenization), kelimenin kökünü bulma (ing. stemming) işlemleridir (Loper ve Bird, 2002). Şekil 2'de ön işlem aşamaları görülmektedir.



Şekil 2. Ön işlem aşamaları

2.3. Metinden Öznitelik Çıkarılması

Öznitelik oluşturma adımı ise; ön işlemden geçirilmiş olan, eğitim ve test verileri alınır. Girdi verilerinin tamsayı olarak kodlanmasını gerektirir, böylece her kelime benzersiz bir tamsayı ile temsil edilir. Bu adım, Keras ile birlikte sağlanan Tokenizer API kullanılarak gerçekleştirilir. Test ve eğitim verilerindeki metinleri dizi (ing. Sequence) vektörüne çevrilir. Bu dizi vektörleri diziye çevrilir. Oluşan vektör dizisi kategori etiketleri ile eşleşerek kategorilendirilmiş vektörler eğitim ve test verisi için hazır hale gelir. Şekil 3'te öznitelik çıkarımı aşamaları görülmektedir.



Şekil 3. Öznitelik çıkarımı aşamaları

2.4. Sınıflandırma

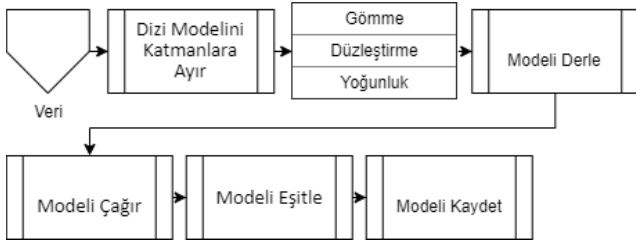
Bu çalışmada derin öğrenme temelli, Keras Embedding Layer kullanılarak eğitilen kategori belirlemek için metin sınıflama kullanılmıştır. Keras Embedding Layer temelde üç katmandan oluşmaktadır:

- Gömme
- Düzleştirme
- Yoğun

Keras, metin verilerindeki sinir ağları için kullanılacak bir gömme katmanı sunar (Keras, 2021). Girdi verilerinin tamsayı olarak kodlanmasını gerektirir, böylece her kelime benzersiz bir tamsayı ile temsil edilir. Gömme katmanı rastgele ağırlıklarla başlatılarak, eğitim, veri setindeki tüm kelimeler için bir yerleştirme öğrenir. Gömmenin modelin kendisiyle birlikte öğrenildiği bir derin öğrenme modelinin parçası olarak kullanılabilir. Bir tür aktarım öğrenme türü olan önceden eğitilmiş bir kelime gömme modelini yüklemek için kullanılabilir. Gömme katmanı, bir ağırlık gizli katmanı olarak tanımlanır. 3 bağımsız değişken belirtmelidir (Brownlee, 2017):

- input_dim: metin verisindeki kelime dağarcığının boyutudur. Örneğin, veri 0-10 arasındaki değerlere kodlanmış tamsayı ise, kelime haznesinin boyutu 11 kelime olacaktır.
- output_dim: kelimelerin gömüleceği vektör uzayının boyutudur. Her kelime için bu katmandan çıktı vektörlerinin boyutunu tanımlar. Örneğin, 32 veya 100 veya daha büyük olabilir.
- input_length: bir Keras modelinin herhangi bir girdi katmanı için tanımlanabileceği gibi, girdi dizilerinin uzunluğudur. Örneğin, tüm girdi metinleri 1.000 kelimeden oluşuyorsa, input length 1.000 olur.

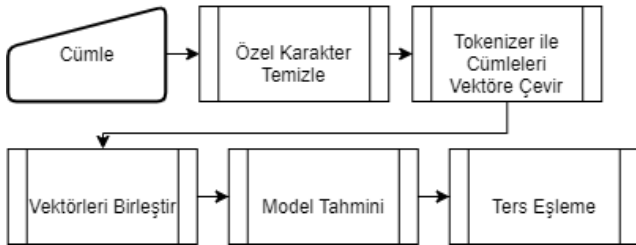
Metin sınıflandırma için Kelime Gömme yöntemi test edilmiştir. Üç katmanlı bir yapı ile kelime gömme işlemi gerçekleştiren Tensorflow Keras Layer kütüphanesi (REF) kullanılmıştır. Eğitim modeli olarak "Sequential Model" kullanılmıştır. Sequential Model (Sıralı model), her katmanın tam olarak bir giriş vektörüne ve bir çıkış vektörüne sahip olduğu düz bir katman yığını için uygundur. Son olarak test verileri ilgili kategorilere tahmin edilmesi denemiştir. Modelin eğitim aşamaları Şekil 4'te gösterilmiştir.



Şekil 4. Model eğitim aşamaları

Sınıflandırma için, öncelikle Sequential model katmanlarına ayrılır. Kelime gömme için kullanılan katmanlar; Embedding katmanı, öğrenilen ağırlıklara sahiptir. Modeli kaydedilirken buna Embedding katmanı ağırlıkları da dahil edilecektir. Embedding katmanı çıktısı, sözcüklerin giriş sırasındaki her sözcük için bir gömme içeren bir 2B vektördür. Flatten katmanı kullanarak 2B çıktı matrisi 1B vektörüne düzeltilir. Böylece Dense katmanı doğrudan Embedding katmanına bağlanır. Gömme işlemi sonrasında model derlenip, eğitime sağlanır. Eğitim sonunda belirtilen adrese model dosyası kaydedilmiş olur.

Eğitilmiş model tahmine hazır hale gelmiştir. Seçilen test verileri tahmin için yönlendirilerek çalışmaya başlanır. Belirlenen vektörler model kullanılarak tahmin edilmesi sağlanır. Tahmin sonucunda bulunan değer ters eşleme ile hangi kategoriye ait olduğu belirlenmiş olur. Tahmin aşamaları Şekil 5'te gösterilmiştir.



Şekil 5. Tahmin aşamaları

2.5. Başarım Metrikleri

Çalışmanın TP (doğru pozitif, ing. True Positive), FP (yanlış pozitif, ing. False Positive), TN (doğru negatif, ing. True Negative), FN (yanlış negatif, ing. False Negative) değerinden yola çıkılarak kesinlik, hassasiyet, F-Skor ve doğruluk hesaplamaları yapılmaktadır.

2.5.1 Kesinlik ve Hassasiyet (ing. Precision & Recall)

Kesinlik, bir ölçme aletinin aynı fiziksel boyuta ait tekrarlanan çeşitli ölçümlerde aynı değeri verebilme ölçüsüdür (Powers, 2020). Yapılan çalışmada doğru kategoride bulunan sonuçların yüzde kaçının gerçekten de doğru kategoride olduğudur. Eşitlik (1)'de Kesinlik formülü görünmektedir. Hassasiyet, yapılan bir ölçümde getirilen doğru bilginin, yapılan çeşitli

ölçümlerle getirilmesi gereken doğru sonuçlara oranını ölçmektedir (Powers, 2020). Yani; yapılan çalışmada doğru kategoride bulunan sonuçların yüzde kaçının pozitif doğru olduğudur. Eşitlik (2)'de Hassasiyet formülü görünmektedir.

$$Kesinlik = \frac{TP}{TP + FP} \quad (1)$$

$$Hassasiyet = \frac{TP}{TP + FN} \quad (2)$$

Bu denklemlerde görünen ifadeler, donanım ve diğer kategoriler olarak ele alınan bir modelin başarımını ölçmede;

TP: Gerçekte "donanım" kategorisinde olup, "donanım" olarak tahmin edilen;

FP: Gerçekte "donanım" kategorisinde olmayıp, "donanım" olarak tahmin edilen;

FN: Gerçekte "donanım" kategorisinde olup, "donanım" olarak tahmin edilmeyen;

TN: Gerçekte "donanım" dışındaki kategori/kategorilerde olup, "donanım" dışındaki kategori/kategorilerde tahmin edilmesidir.

2.5.2 F-Skoru (ing. F-Score)

F-Skoru, hassasiyet ve kesinlik değerlerinin harmonik ortalamasıdır. Eşitlik (3)'te F-Skoru formülü ve çalışmanın F-skoru görünmektedir (Powers, 2020).

$$F - Skoru = 2 \times \frac{Kesinlik \times Hassasiyet}{Kesinlik + Hassasiyet} \quad (3)$$

2.5.3 Doğruluk (ing. Accuracy)

Doğruluk değeri, Eşitlik (4)'te görünen, doğru tahmin edilen kategorilerin toplam veri kümesine oranıdır (Powers, 2020).

$$Doğruluk = \frac{TP + TN}{TP + TN + FN + FP} \quad (4)$$

Kullanılan deneyde birden fazla kategori olması durumunda; TP, TN, FP ve FN sonuçları farklılaşır ve yorumlamak zorlaşır. Bu sebeple karışıklık matrisini (ing. confusion matrix) yorumlamak için her bir kategori değeri ayrı hesaplanarak ortalamalarının alınması gerekmektedir. Örneğin Tablo 2'de görüldüğü gibi Donanım kategorisini tek başına ele alıp, donanım ve diğer kategoriler olarak işlem yapılmaktadır.

Tablo 2

Donanım karışıklık matrisi

		Gerçek Kategoriler	
		Donanım	Diğer
Tahmin Edilen Kategoriler	Donanım	208	50
	Diğer	157	32

3. Deneysel Çalışma

Python 3.8 kullanılarak Spyder IDE'si ile geliştirme ortamında, veri seti üzerinde, ön işleme ve model eğitim adımları gerçekleştirilmiştir.

Uygulama, 9 kategoriye sahip veri seti için, eğitim ve tahmin olmak üzere ayrılmıştır. Eğitim bölümünde model eğitimi gerçekleştirilmiştir. 28.104 adet veri, %60 eğitim, %20 doğrulama ve %20 test kümesi olarak ayrılarak, ön işleme adımlarında, kök bulma ve durak kelimelerin atılması sağlanmıştır. Elde edilen veriler dizi vektörüne dönüştürülür. Burada, word2vec vektör uzayının boyutsallığı 100'e eşit olan eğitim kümesinden hesaplanmıştır. Gömme Katmanı için LSTM ağı değerleri Tablo 3'te verildiği şekilde belirlenmiştir.

Tablo 3

Çok terimli sınıflandırma için vektörleştirici ile kullanılan LSTM ağı

input_dim:	MAX_WORDS(28.104)
output_dim:	OUTPUT_DIM(50)
input_length:	MAX_SEQUENCE_LENGTH(100)

Model üç katmandan oluşmaktadır. Sonuçlar Tensorflow paketi ile Keras framework kullanılarak elde edilmiştir. Tüm deneyler 16GB RAM ile NVIDIA GeForce GTX 1650 Ti GPU üzerinde yapılmıştır. Eğitimde kullanılan maksimum dönem sayısı (ing. epochs) 5 ve parti boyutu (ing. batch_size) 200'dir. İkili sınıflandırma için ikili çapraz entropi (ing. Crosstentropy) ve bir kayıp (ing. loss) fonksiyonu olarak çok terimli seyrek kategorisel çapraz entropi (ing. categorial crossentropy) ve optimize edici, varsayılan parametrelere sahip Adam fonksiyonu kullanılmıştır. Aktivasyon fonksiyonu olarak geleneksel olarak kullanılan softmax tercih edilmiştir.

Veri seti üzerinde gerçekleştirilen deneysel çalışmalar, farklı kategori ayrımları gerçekleştirilerek, üç farklı yapıda ele alınmıştır. Böylece, kategoriye bağlı başarımların analizi de sağlanmıştır. Dördüncü deneyde ise, modelin eğitiminde başarımların epoch sayısına bağlılığı incelenmiştir.

3.1. Deney 1: Genel Çalışma Sonuçları

Dokuz kategoriden oluşan veri seti, Keras Gömme katmanı kullanılarak eğitilmiş olup, derin öğrenme tabanlı bir kategori sınıflandırma deneyi yapılmıştır. Elde edilen tahminlerin kategori bazlı dağılımına göre hesaplanan Karışıklık Matrisi Tablo 4'te, Başarımlar Metrikleri ise Tablo 5'te görünmektedir.

Tablo 4'te algoritmanın doğru ve hatalı sınıflandırma sonuçlarının kategorilere göre dağılımı gösterilmektedir. Bu matristen yola çıkılarak, donanım ve kullanıcı kategorilerinin birbirlerine fazlasıyla karıştığı gözlenmektedir olup, birbirinin yerine karar verildiği gözlemlenmektedir.

Veriler incelendiğinde, taleplerin kullanıcı yetkilendirmelerinin ve donanım isteklerinin benzer ifadelerle anlatıldığı ya da kullanıcı yetkilendirme talebi olan birinin donanım ihtiyacı varmış gibi belirttiği ve donanım ihtiyacı olan birinin ise kullanıcı yetkilendirme gibi ifadeler kullandığı görülmektedir.

Problem dokuz sınıflı olarak ele alındığında %97,60'lık genel doğruluk oranına ulaşılırken, hassasiyet ve kesinlik oranları %89,22 olarak elde edilmektedir. F-skor değeri ise %89,22'dir.

Tablo 5'teki başarımlar metriklerinden görüleceği üzere, elde edilen sınıflandırma modeli ile %89,22'lik genel F1-skoru elde edilmiştir. Sonuçlar değerlendirildiğinde %94,95 başarımlar ile "E-Posta" kategorisindeki tahminler en yüksek değeri elde etmiştir. Bunun sebebi olarak "E-Posta" kategorisinde açılan taleplerin, daha anlaşılır ve tek bir konuya hizmet edecek şekilde ifade edildiği görülmektedir. Yaklaşık %83'lük oranla, diğer kategorilere oranla daha düşük başarımlara sahip "Yazıcı" kategorisindeki talepler incelendiğinde ise, bu alandaki ifadelerin çok farklı bir biçimde yazıldığı, kullanıcıların yazıcı sorunlarını aktarma biçimlerinin çok farklı olduğu gözlemlenmektedir. En düşük başarımlara sahip "Sunucu" kategorisinin ise teknik anlamda kullanıcıların isteklerini tam olarak ifade edemediği gözlemlenmiştir.

3.2. Deney 2: Yüksek Başarımlı Kategoriler

Kategori azaltılarak, Donanım, E-Posta, Kullanıcı, Sap, Yazılım kategorileri kullanılarak model eğitilerek, test sonuçları alınmıştır. Alınan sonuçların başarımlar metrikleri ise Tablo 6'da görünmektedir.

Buna göre 6 kategori üzerinden çalışma yapılmış ve başarımlar oranlarının belli kategorilerde azalsa da toplamda genel F-Skor değerinin %91'e kadar arttığı gözlemlenmektedir.

Tablo 4

Karışıklık matrisi

	Donanım	E-Posta	Kullanıcı	Kurumsal	SAP	Sunucu	Veri	Yazıcı	Yazılım
Donanım	942	14	58	24	18	18	36	34	57
E-Posta	2	470	3	1	3	2	2	0	5
Kullanıcı	21	4	788	14	15	8	28	8	19
Kurumsal	4	4	3	291	4	1	1	2	2
SAP	4	2	1	4	409	1	2	1	0
Sunucu	4	1	3	7	3	168	8	2	7
Veri	9	2	24	8	6	12	529	1	10
Yazıcı	8	1	0	3	3	0	3	170	1
Yazılım	13	4	7	10	3	6	5	2	1.247

Tablo 5

Başarım metrikleri (%)

	Doğruluk	Kesinlik	Hassasiyet	F-skör
Donanım	94,23	78,43	93,55	85,33
E-Posta	99,11	96,31	93,63	94,95
Kullanıcı	96,16	87,07	88,84	87,95
Kurumsal	98,36	93,27	80,39	86,35
SAP	98,75	96,46	88,15	92,12
Sunucu	98,52	82,76	77,78	80,19
Veri	97,21	88,02	86,16	87,08
Yazıcı	98,77	89,95	77,27	83,13
Yazılım	97,31	96,14	92,51	94,29
Toplam	97,60	89,22	89,22	89,22

Tablo 6

Azaltılmış kategori çıktıları başarım metrikleri (%)

Kategori	Doğruluk	Kesinlik	Hassasiyet	F-skör
Donanım	91,46	83,42	91,63	87,34
E-posta	98,76	94,79	95,59	95,19
Kullanıcı	95,94	90,47	90,97	90,72
Sap	98,06	94,96	85,33	89,89
Yazılım	94,57	95,69	92,08	93,85
Toplam	95,95	91,07	91,54	91,31

3.3. Deney 3: Ekip Bazlı Dağılım

İlgili kategorileri BT de sorumlu ekiplere gidecek şekilde güncelleyerek, 9 kategori toplamda 3 ekibe dağılacak şekilde dağıtılarak, veri seti buna göre güncellenmiştir. Model, ilgili 3 ekibe göre eğitilerek sonuçlar alınmıştır. Buna göre; Donanım, Yazılım ve Yazıcı Destek; E- Posta,

Sunucu ve Kullanıcı Sistem; Kurumsal, SAP ve Veri Mimari ekiplerine dağıtılmış ve oluşan tahmin sonuçlarına göre Başarım Metrikleri de Tablo 7'de gösterilmiştir. Ekip bazlı dağılımda tahmin oranının %93'e kadar çıktığı gözlemlenmektedir.

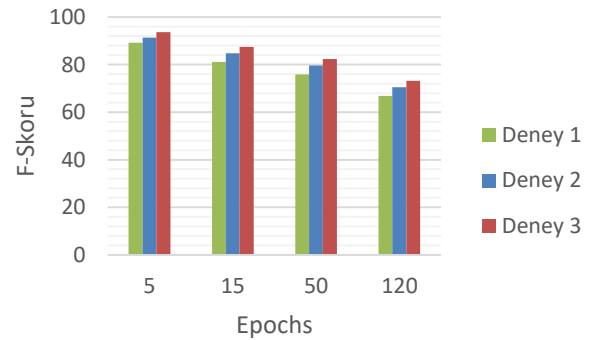
Tablo 7

Ekip bazlı kategori çıktıları başarım metrikleri (%)

Kategori	Doğruluk	Kesinlik	Hassasiyet	F-Skor
Destek	91,31	93,87	96,33	95,08
Mimari	93,91	92,07	94,09	93,07
Sistem	93,02	90,16	91,85	91,00
Toplam	92,88	92,36	94,49	93,42

3.4. Deney 4: Parametrik Güncellemeler

Kategori düzenlemelerinden sonra, model üzerindeki parametre değerleri ile sistem kararlılığının değişimini gözlemlemek için epochs (dönem sayısı) değerini 5 ile 120 arasında değiştirerek, her bir deney tekrarlanarak ilgili sonuçlar alınmıştır. Şekil 6'daki epochs dağılım grafiğine bakıldığında, epochs sayısının en yüksek olduğu değer 5 olarak görülmüş, epochs değeri arttıkça başarımın ortalama olarak düştüğü gözlemlenmiştir.



Şekil 6. Epochs dağılımı

3.5. Deney 5: Karşılaştırmalı Sonuçlar

Veri seti, Bi-LSTM algoritması ile test edilmiş ve mevcut LSTM algoritması ile karşılaştırılmıştır. Tablo 8'de tüm kategoriler sınıflandırılmış ve karşılaştırılmıştır. Tablo 9'da ise ekip bazlı dağılım kategorileri karşılaştırılmıştır. Her iki tabloda görüldüğü üzere,

LSTM algoritması, Bi-LSTM algoritmasına göre mevcut veri setinde daha iyi sonuçlar vermektedir.

Tablo 8

Başarım metrikleri (%) karşılaştırması

Algoritma	F-Skor
LSTM	89,22
Bi-LSTM	50,26

Tablo 9

Ekip bazlı kategori başarımleri (%) karşılaştırması

Algoritma	F-Skor
LSTM	93,08
Bi-LSTM	83,04

4. Tartışma

Yazılı olarak iletilen yardım masası talep içeriklerinin belirlenmesi probleminin metin madenciliği teknikleri ile ele alındığı yaklaşımda, GPU üzerinde eğitilen modellerle derin sinir ağları için Tensorflow kütüphanesine sahip Keras kullanılmıştır. Kelime dizileriyle temsil edilen talep başlıkları LSTM sinir ağını eğitmek için kullanılmıştır. Dizi vektör dönüşümü kelimeleri kodlamanın sınıflandırma sonuçlarının word2vec vektör uzayında temsil ile elde edilmiştir. Bu çalışma performansı, word2vec kodlu kelime dizileriyle eğitilmiş LSTM modelleri ile gerçekleştirilmiştir. Derin sinir ağlarının eğitimi genel olarak yüksek kaynak gerektiren ağlar olarak kabul edilse de bu çalışmada sunulan derin LSTM sinir modellerinin eğitiminin Tensorflow'lu Keras gibi sağlam kütüphane ve orta ölçekli GPU cihazı (16 GB RAM) kullanılarak mümkün olduğu görülmüştür. Deney sonuçları değerlendirildiğinde, tüm kategoriler için yapılan Deney 1'de %89'luk bir F-skor ile kayda değer başarımler sağlanmıştır. Buna ilave olarak Deney 2 çalışması ile kategoriler azaltılarak, F-skoru artmış, daha az kategori sınıflandırmasında sistemin daha yüksek başarımla çalıştığı gözlemlenmiştir. Kurum içerisindeki ekiplere göre dağıtım da ise (Deney 3) bu oran %93'e çıkarak kurum içerisinde sağlıklı bir şekilde kullanılabilirdiği gözlemlenmiştir. Hazır veri setlerinden farklı olarak, iş yeri ortamından alınan veriler kullanılmış olup, gerçek hayatta bir problemin çözümüne katkı sağlanmıştır. Geliştirilen yardım masası talep sınıflandırma modeli ile, kurum bünyesinde sorun çözümünde zaman ve iş gücü kaybının azaltılmasına katkıda bulunacaktır. Makine öğrenmesi ve derin öğrenme temelli uygulamaların iş odaklı olarak kullanılması, iş dünyasındaki verimlilik artışını hedeflemektedir. Bu yetkinlik önerilen yaklaşıma esnek geliştirme imkânı sunmaktadır. Birbirleri ile karışan sınıflardaki eğitim

verileri, modelde uyarlanabilir olarak sürekli güncellenerek veya taleplerin başlıkları ile birlikte, içeriğinde yer alan metinler de analize dahil edilen yeni modeller ile başarımları artırılabilir.

Teşekkür

Gerçek dünya uyarlamasına katkılarından dolayı TEI - TUSAŞ Motor Sanayii A.Ş.'ye teşekkür ederiz.

Çıkar Çatışması

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir.

Yazarların Katkısı

Bu çalışmada; Metin YILMAZ, çalışmanın tasarımı, uygulanması, bilgisayara ortamına aktarılması, bilimsel yayın araştırması, makalenin oluşturulması; Efnan ŞORA GÜNAL, bilimsel yayın araştırması, makalenin oluşturulması, istatistiki analizler, sonuçların elde edilmesi konularında katkı sağlamıştır.

Kaynaklar

- ALRashdi, R., & O'Keefe, S. (2019). Deep learning and word embeddings for tweet classification for crisis response. arXiv preprint arXiv:1903.11024.
- Borko, H., & Bernick, M. (1963). Automatic document classification. *Journal of the ACM (JACM)*, 10(2), 151-162.
- Cai, S., Palazoglu, A., Zhang, L., & Hu, J. (2019). Process alarm prediction using deep learning and word embedding methods. *ISA Transactions*, 85, 274-283.
- Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., & Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14), i37-i48.
- Ibrahim, M. A., Khan, M. U. G., Mehmood, F., Asim, M. N., & Mahmood, W. (2021). GHS-NET a generic hybridized shallow neural network for multi-label biomedical text classification. *Journal of biomedical informatics*, 116, 103699.
- Brownlee, J. (2017). How to Use Word Embedding Layers for Deep Learning with Keras. Eişim Adresi: <http://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras>.
- Liang, D., & Yi, B. (2021a). Two-stage three-way enhanced technique for ensemble learning in inclusive policy text classification. *Information Sciences*, 547, 271-288.

- Liang, Y., Li, H., Guo, B., Yu, Z., Zheng, X., Samtani, S., & Zeng, D. D. (2021b). Fusion of heterogeneous attention mechanisms in multi-view convolutional neural network for text classification. *Information Sciences*, 548, 295-312.
- Jurafsky, Daniel; H. James, Martin (2000). *Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J.: Prentice Hall. ISBN 978-0-13-095069-7.
- Keras, (2021). *Sequential_model*. Erişim adresi: http://keras.io/guides/sequential_model.
- Kilimci, Z. H., & Akyokus, S. (2018). Deep learning-and word embedding-based heterogeneous classifier ensembles for text classification. *Complexity*.
- Kocmi, T., & Bojar, O. (2017). An exploration of word embedding initialization in deep-learning tasks. *arXiv preprint arXiv:1711.09160*.
- Kowsari, K., Brown, D. E., Heidarysafa, M., Meimandi, K. J., Gerber, M. S., & Barnes, L. E. (2017). Hdltext: Hierarchical deep learning for text classification. In *16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 364-371.
- Li, S., Hu, J., Cui, Y., & Hu, J. (2018). DeepPatent: patent classification with convolutional neural networks and word embedding. *Scientometrics*, 117(2), 721-744.
- Loper, E., & Bird, S. (2002). *Nltk: The natural language toolkit*. *arXiv preprint cs/0205028*.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.
- Semberecki, P., & Maciejewski, H. (2017). Deep learning methods for subject text classification of articles. In *IEEE Federated Conference on Computer Science and Information Systems (FedCSIS)*, 357-360.
- Wang, J. H., Liu, T. W., Luo, X., & Wang, L. (2018). An LSTM approach to short text sentiment classification with word embeddings. In *30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018)*, 214-223.
- Zhang, Y., Lu, J., Liu, F., Liu, Q., Porter, A., Chen, H., & Zhang, G. (2018). Does deep learning help topic extraction? A kernel k-means clustering method with word embedding. *Journal of Informetrics*, 12(4), 1099-1117.