Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi

Afyon Kocatepe University Journal of Science and Engineering

AKÜ FEMÜBİD **23** (2023) 015101 (72-88) AKU J. Sci. Eng. **23** (2023) 015101 (72-88)

DOI: 10.35414/akufemubid.1115786

Araştırma Makalesi / Research Article

Adlandırılmış Varlık Tanıma Modelleri ile Türkçe Sosyal Medya Metinlerinde Küfürlü Sözlerin Sansürlenmesi

Resmiye NASİBOGLU[1*], Mustafa GENCER[2]

[1] *Dokuz Eylül Üniversitesi, Fen Fakültesi, Bilgisayar Bilimleri Bölümü, İzmir, Türkiye*

[2] *Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, İzmir, Türkiye*

Sorumlu Yazar e-posta: resmiye.nasiboglu@deu.edu.tr ORCID ID: https://orcid.org/0000-0003-1739-1469*

mustafagencrr@gmail.com ORCID ID: https://orcid.org/0000-0002-8610-8041

Geliş Tarihi: 20.05.2022 Kabul Tarihi: 25.01.2023

Anahtar kelimeler
Küfür Tespiti;
Adlandırılmış Varlık Tanıma;
Doğal dil İşleme;
Bi-LSTM;
BERT

Öz

Adlandırılmış varlık tanıma problemi, veri çıkarımı, doğal dil işleme ve metin madenciliği gibi alanların alt dalı olarak ele alınmaktadır. Adlandırılmış varlık tanıma, yapılandırılmamış metinlerdeki varlık isimlerinin uygunluklarına göre önceden belirlenen kişi ismi, organizasyon ismi veya yer ismi gibi sınıflara atama yapmak için kullanılan bir araçtır. Gelişen teknoloji ile birlikte sosyal ağlar çok insan tarafından kullanılmaktadır. Sosyal medya kullanan kişiler her türlü resim, metin veya video içeriklerini paylaşabilmektedir. Paylaşılan bu içerikler ise bazen uygunsuz yani aile yapısını etkiler nitelikte olabilmektedir. Bu çalışmada, Twitter'daki Türkçe tweetler kullanılarak küfür, hakaret ve uygunsuz kelimeler adlandırılmış varlık tanıma problemi olarak ele alınmış ve bu kelimeler farklı yöntemler ile tespit edilmeye çalışılmıştır. Çalışmada, önce metinlerde geçen kelime ve kelime öbekleri etiketlenmiş daha sonra ise etiketlenen kelimeler vektörleştirilmiştir. Vektörler, Bi-LSTM ve öneğitimli BERT modelleri kullanılarak eğitim yapılmıştır. Bi-LSTM modeli hem eğitimde hem de test aşamasında %99'a yakın doğruluk oranı sergilemiştir. BERT modeli ise eğitim aşamasında %99 civarında doğruluk oranı gösterirken, test başarısının %95 civarında olduğu gözlemlenmiştir. Çalışma hızı açısından, Bi-LSTM modelinin BERT modelinden yaklaşık olarak 3 kat daha hızlı olduğu görülmüştür.

Censorship of Profanity Words in Turkish Social Media Texts with Named Entity Recognition Models

Keywords
Profanity Detection;
Named Entity Recognition (NER);
Natural Language Processing (NLP);
Bi-LSTM;
BERT

Abstract

Named Entity Recognition problem is considered as a sub-branch of fields such as data extraction, natural language processing and text mining. Named entity recognition is a tool used to assign classes such as predetermined person name, organization name or place name according to the suitability of entity names in unstructured texts. With the developing technology, social networks are used by many people. People using social media can share any image, text or video content. These shared contents may be inappropriate, that is, affect the family structure. In this study, using Turkish tweets on Twitter, swearing, insults and inappropriate words were studied as a named entity definition problem and these words were tried to be determined by different methods. In the study, first the words and phrases in the texts were labeled, and then the labeled words were vectorized. Training was done using vectors, Bi-LSTM and pretrained BERT models. The Bi-LSTM model showed close to 99% accuracy both in training and testing. On the other hand, the BERT model showed a training accuracy of around 99% during the training phase, while the test success was observed around 95%. In terms of operating speed, it has been observed that the Bi-LSTM model is approximately 3 times faster than the BERT model.

1. Giriş

Web ve mobil uygulamaların kullanımı arttıkça, kullanıcı katkılarından kaynaklanan uygunsuz içeriğin varlığı daha sorunlu hale gelir. Sosyal haber siteleri, forumlar ve herhangi bir çevrimiçi topluluk, bir topluluğun sosyal normlarına ve beklentilerine uygun olmayanları sansürleyerek, kullanıcı tarafından oluşturulan içeriği yönetmelidir. Bu tür

içeriğin sansürlenmemesi, yalnızca potansiyel kullanıcıları veya ziyaretçileri caydırmakla kalmaz, aynı zamanda bu tür içeriğin kabul edilebilir olduğuna da kanaat getirebilir.

Her şeyin dijital olarak yönetildiği günümüzde, insanların kullandığı birçok çevrimiçi platform ve forum var. Instagram gibi herhangi bir sosyal medya platformundan bir örnek alırsak, onların gizlilik politikası, kullanıcıların herhangi bir müstehcen/kaba dili herkese açık bir platformda paylaşamayacaklarını veya yazamayacaklarını göstermektedir. Birçok kurum ve kuruluş, kamusal alanlardaki yasa dışı faaliyetlerin tespit edilmesi için çeşitli görüntü işleme, sosyal medya metinlerindeki küfürleri tespit etmek için çeşitli metin işleme teknolojileri kullanmaktadır. Bununla birlikte, mevcut küfür algılama sistemleri, çeşitli faktörler nedeniyle hala kusurlu olmaya devam etmektedir. (Su *et al.* 2017, Sood *et al.* 2012, Laboreiro and Oliveira 2014).

Küfür tespitinin genellikle kolay bir iş olduğu düşünülür. Bununla birlikte, geçmiş çalışmalar, mevcut liste tabanlı sistemlerin kötü performans gösterdiğini göstermiştir. Değişen küfürlü argoya uyum sağlamada, gizlenmiş veya yalnızca kısmen sansürlenmiş (örneğin, "@ss, f$#%") veya kasıtlı veya kasıtsız olarak yanlış yazılmış (örneğin, "aptaalll" gibi) küfürlü terimleri tanımlamada başarısız olurlar. Bu nedenlerden dolayı, sistemi atlatmaları kolaydır ve hatırlanmaları çok zayıftır (Sood *et al*. 2012, Sood *et al.* 2012, Lee *et al.* 2018). İkinci olarak, liste tabanlı yaklaşım, her türlü çözüme uydurulan tek boyutlu bir çözümdür. Saygısız veya uygunsuz tanımının, kullanımının ve algılarının tüm bağlamlarda geçerli olduğuna dair varsayımlarda bulunurlar (Laboreiro and Oliveira 2014).

Kişiler tarafından saygısız metinlerin kullanılması, dijital alanın özgürlüğünü ve bütünlüğünü tehdit etmektedir. Bu tür saygısız metinleri kontrol etmek için geleneksel olarak manuel denetleme ve raporlama mekanizmaları kullanılmıştır. İnsan yorumuna bağımlılık ve sonuçların gecikmesi bu sistemin önündeki en büyük engeller olmuştur. Süreci otomatikleştirmek için önceki derin öğrenme tabanlı yaklaşımlar, geleneksel evrişim ve yineleme tabanlı sıralı modellerin kullanımını içermektedir. Bununla birlikte, bu modeller hesaplama açısından pahalı olma eğilimindedir ve daha yüksek bellek gereksinimine sahiptir. Ayrıca, çok etiketli görevlerde nispeten zayıf performans gösterirler. Günümüz dünyasında, metni ikili bir şekilde sınıflandırmak artık yeterli değildir ve bu nedenle, çok etiketli metinler üzerinde iyi genelleme yapabilen esnek bir çözüm gereklidir (Ratadiya and Mishra 2019).

Küfür içeren kelimeler her zaman tespit edilen bariz söz öbekleri içermemektedir. Kelimenin anlamı veya içeriği bağlamsal olarak küfür olabilmektedir. Bu durumlarda liste bazlı ve kural tabanlı sistemler kelimeyi tespit etmekte zorlanabilmektedir. O yüzden bu tarz sistemler liste veya kural dışı bir küfür ile karşılaştığında başarısız olabilmektedir. Yi *et al.* (2021) makalesinde kelime gömme ve LSTM modelini kullanarak bir küfür tespit yöntemi önerilmiştir. Önerilen yöntem, eğitim yapmak için metni "onset", "nucleus" ve "coda" olarak 3 ayrı parçaya bölmüştür.

Yukarıdaki durumlara ek olarak, küfür içeren cümleler her zaman nefret söylemi olmadığından, nefret söyleminin otomatik tespiti için anahtar kelime olarak küfür kullanmak tam olarak mümkün olmayabilir. Örneğin, " What the hell is wrong with this air conditioner?", cümlesinde aslında söylenmek istenen "Bu klimanın nesi var?" Şeklinde çevirilebilir ancak küfürlü "cehennem" kelimesini içermesine rağmen kasıtlı olarak dilin kötüye kullanılmasından çok duygusal bir ifadedir. Benzeri durumlar pek çok dilde görülebilmektedir (Teh and Cheng 2020). Tersine bu durum, nefret, küfür içermeyen muğlak şakalar yoluyla da ortaya çıkabilir. Bu gibi durumları tespit etmekte cümle içerisindeki küfür tespitinin zorluklarından bir tanesidir.

Türkçe küfür tespit çalışmalarından biri Çelik ve Yıldırım (2020) tarafından gerçekleştirmiştir. Çalışmada, önce metin ön işlemlerden geçirilmiştir. Daha sonra eğer kelime sayısı 2 veya daha az ise metin sezgisel modele, 3 kelime ya da daha fazla ise

yapay zekâ modeline yönlendirilmiştir. 3 faklı yapay zekâ modelinin döndürdüğü olasılıklara bakarak hakem modelin karar vermesi sağlanmıştır.

Çizelge 1. Adlandırılmış Varlık Tanıma için Varlık Örnekleri ve Tanımları

TÜR	TANIMI
Kişi	İnsan, hayali karakter
Gruplar	Ulus, din, politik grup
Organizasyon	Şirket, ajans, enstitü
Yer	Ülke, şehir, eyalet adları
Konum	Dağ, su kaynağı,
Ürün	Otomobil, araç, yiyecek
Olay	Adlandırılmış kasırga,
Sanatsal aktivite	Kitap, şarkı vb. Adları
Kanuni belge	Kanunla adlandırılmış
Dil	Adlandırılmış herhangi
Tarih	Mutlak veya göreli
Zaman	Günden daha kısa zaman
Yüzde	% işareti içeren yüzdeler
Para	Birim dahil parasal
Nitelik	Ağırlık veya mesafe
Sıralama ifadeleri	Birinci, ikinci vb.

Bir metni değerlendirirken veya anlarken, insanlar, değerler, konumlar vb. gibi adlandırılmış varlıkları doğal olarak tanırız. Örneğin, "Jack Dorsey, Amerika Birleşik Devletleri'nden bir şirket olan Twitter'ın kurucularından biridir." cümlesinde üç tür varlık tanımlayabiliriz:

Kişi Adı: Jack Dorsey,
Şirket Adı: Twitter,
Konum Adı: Amerika Birleşik Devletleri.

Ancak bilgisayarlar için, onları kategorize edebilmeleri için önce varlıkları tanımalarına yardımcı olmamız gerekir. Bu, makine öğrenimi ve Doğal Dil İşleme (NLP) aracılığıyla yapılır. Doğal dil işleme, dilin yapısını ve kurallarını inceler ve metin ve konuşmadan anlam çıkarabilen akıllı sistemler yaratırken, makine öğrenimi, makinelerin öğrenmesine ve zaman içinde gelişmesine yardımcı olur.

Bir varlığın ne olduğunu öğrenmek için, bir adlandırılmış varlık tanıma modelinin bir kelimeyi veya bir varlığı oluşturan kelime dizisini (örneğin, "İzmir şehri") tespit edebilmesi ve hangi varlık kategorisine ait olduğunu bilmesi gerekir. Adlandırılmış varlık tanıma (Named Entity Recognition - NER), bir metindeki kişi adları, yerler, markalar, parasal değerler ve daha fazlası gibi temel öğeleri kolayca tanımlamamıza yardımcı olur. Çalışmalarda kullanılan bazı adlandırılmış varlık türleri Çizelge 1'de verilmiştir.

Bir metindeki ana varlıkları ayıklamak, yapılandırılmamış verileri sıralamaya ve büyük veri kümeleriyle uğraşmanız gerektiğinde çok önemli olan bilgileri algılamaya yardımcı olur (Sienčnik 2015). Adlandırılmış varlık tanımanın bazı ilginç kullanım örnekleri şunlardır:

- *Müşteri çağrılarını sınıflandırmak* (Subramaniam *et al.* 2009*).*

Müşteri çağrı hatlarında, müşteri isteklerini daha hızlı değerlendirmek ve yanıtlamak için adlandırılmış varlık tanıma teknikleri kullanılabilmektedir. Müşterilerin sorunlarını ve sorgularını kategorilere ayırmak gibi tekrarlayan müşteri hizmetleri görevleri otomatikleştirilebilir ve sorun çözüm oranlarını iyileştirmeye ve müşteri memnuniyetini artırmaya yardımcı olacak değerli zamandan tasarruf edilebilir. Ürün adları veya seri numaraları gibi ilgili veri parçalarını çekmek için varlık ayıklama da kullanabilir ve bu sorunu ele almak için sorunlu durumları en uygun temsilciye veya ekibe yönlendirmeyi kolaylaştırılabilir (Luo *et al.* 2011).

- *Müşteri geri bildirimlerini değerlendirmek* (Meng *et al.* 2012).

Çevrimiçi değerlendirmeler ve incelemeler, müşteri geri bildirimi için kaynak oluşturabilir. Müşterilerin ürünler hakkında neleri beğendiği, beğenmediği ve işletmenin iyileştirilmesi gereken yönleri hakkında bilgiler sağlayabilir. Adlandırılmış varlık tanıma sistemleri, tüm bu müşteri geri bildirimlerini düzenlemek ve tekrar eden sorunları saptamak için kullanılabilir. Örneğin, olumsuz müşteri geri bildirimlerinde en sık bahsedilen konumları ve illeri saptamak için adlandırılmış varlık tanıma kullanılabilir, bu sayede müşteri belirli bir ofis veya şubeye yönlendirebilir (Han *et al.* 2017).

- *İçerik önerisi oluşturmak*.

Netflix ve YouTube gibi birçok modern uygulama, optimum müşteri deneyimleri oluşturmak için öneri sistemlerine güvenir. Bu sistemlerin çoğu, kullanıcı arama geçmişine dayalı önerilerde bulunabilen

adlandırılmış varlık tanımaya dayanır. Örneğin, Netflix'te çok sayıda komedi izliyorsanız, Komedi varlığı olarak sınıflandırılmış daha fazla öneri alırsınız, ya da Youtube'da izlenilen video türüne göre benzer video türleri önerilmektedir. Yine burada video türleri adlandırılmış varlık tanıma olarak ele alınmaktadır (Guo *et al.* 2009, Bowden *et al.* 2018).

- *Özgeçmişleri işlemek*.

İşverenler, günlerinin pek çok saatini özgeçmişleri gözden geçirerek doğru adayı aramakla geçirirler. Her özgeçmiş aynı türde bilgi içerir, ancak bunlar genellikle farklı şekilde düzenlenir ve biçimlendirilir. Varlık adı tanıma yöntemleri kullanılarak, kişisel bilgiler, ad, adres, telefon numarası, doğum tarihi, e-posta, şirket adları, beceriler, sertifikalar, eğitim ve deneyimleriyle ilgili verilere ulaşılabilir (Deepak *et al.* 2020, Pawar *et al.* 2012). Bu sayede adaylarla ilgili en alakalı bilgiler anında çıkarılır ve adaylar iş için hızlı bir elemeye tabi tutulabilir.

Bizim bu çalışmamızda, cümle içerisinde geçen küfür ve hakaret içeren kelimeler ve kelime grupları adlandırılmış varlık olarak ele alınmıştır. Etiketleme için IOB2 etiketleme yöntemi kullanılmıştır. Daha sonrasında etiketlenen kelime grupları Bi-LSTM (Bidirectional Long Short-Term Memory) ve BERT (Bidirectional Encoder Representations from Transformers) modelleri ile eğitilerek küfür ve hakaret olarak tanımlanan kelimeler tahmin edilmiştir.

Makalenin devamında, 2. bölümde, NER ile ilgili literatürdeki ön çalışmalar analiz edilmiştir. Bir sonraki 3. Bölümde, NER çalışmaları için ihtiyaç duyulan metotlar ve teknikler verilmektedir. 4. bölümde hesaplama deneylerinde kullanılan veri seti anlatılmaktadır. Son olarak hesaplama sonuçları ve tartışmalar 5. bölümde yer almaktadır.

2. Literatürdeki Ön Çalışmalar

Adlandırılmış varlık tanıma (NER) uygulamaları arasında, siber zorbalık tespiti ve uygunsuz içeriklerin tespiti gibi alanlarda olan uygulamalar son zamanlarda artan öneme sahiptir. Kişiler tarafından saygısız metinlerin kullanılması, dijital alanın özgürlüğünü ve bütünlüğünü tehdit etmektedir. İnsan yorumuna bağımlılık ve sonuçların gecikmesi bu sistemin otomasyonunun önündeki en büyük zorluklardır. Süreci otomatikleştirmek için derin öğrenme tabanlı yaklaşımlar gittikçe yaygınlaşmaktadır.

Adlandırılmış varlık tanıma problemi ile ilgili ilk çalışmalardan biri 1991 yılında Rau (Rau 1991) tarafından yapılmıştır. Yazar, adlandırılmış varlık tanımayı metin içerisindeki şirket isimlerini bulmak için kullanmıştır. Daha sonra adlandırma yapmak için farklı sınıf isimleri kullanılsa da son ve güncel çalışmalarda CoNLL 2003 (Conference on Computational Natural Language Learning) (Sang and Meulder 2003) ve MUC-6 (Message Understanding Conference) (Grishman 1995) konferanslarında kullanılan veya ortaya atılan varlık adı tanımları kabul görmüştür ve kullanılmıştır. CoNLL'de adlandırılmış varlık tanıma problemi, genel olarak metinde geçen ve ENAMEX olarak adlandırılan kişi adı (Person), yer adı (Location) ve organizasyon adı (Organization) için sınıflandırma işlemi olarak kabul edilmektedir (Sang and Meulder 2003). MUC-6' da ise ENAMEX sınıfı dışında NUMEX (parasal değerler, sayısal değerler ve yüzde ifadeleri) ve TIMEX (saat, tarih) değerleri adlandırılmış varlık tanıma problemine yeni sınıflar olarak dahil edilmiştir (Krupka 1995). Bu üç adlandırılmış varlık tanıma sınıfı, yani ENAMEX, NUMEX ve TIMEX sınıfları dışında çalışmanın kapsamına bağlı olarak veri çıkarımı için alana özgü varlık tanımlamaları da yapılabilmektedir.

Mr. <ENAMEX TYPE="PERSON">Dooner</ENAMEX> met with <ENAMEX TYPE="PERSON">Martin Puris</ENAMEX>, president and chief executive officer of <ENAMEX TYPE="ORGANIZATION">Ammirati & Puris</ENAMEX>, about <ENAMEX TYPE="ORGANIZATION">McCann</ENAMEX>'s acquiring the agency with billings of <NUMEX TYPE="MONEY">$400 million</NUMEX>, but nothing has materialized.

Şekil 1. Örnek Adlandırılmış Varlık Tanıma gösterimi (Grishman 1995).

Şekil 1'de MUC-6 Konferansından alınan yazıdan bazı adlandırılmış varlık tanıma örnekleri verilmiştir. Burada "Mr." önekinden sonra gelen "Dooner" ismi ENAMEX olarak adlandırılmış ve kelime tipi kişi (Person) olarak belirlenmiştir. "Ammirati & Puris" ise yine ENAMEX olarak atanmış, ama kelime tipi olarak organizasyon (Organization) seçilmiştir. Son

olarak "$400 million" NUMEX sınıfına atanmış ve kelime tipi olarak para (Money) olarak belirlenmiştir. Adlandırılmış varlık tanımada bu gibi yapılandırılmamış metinlerdeki varlıkları bulup daha önceden belirlenmiş sınıflara atama işlemi gerçekleştirilmektedir.

NER modellerinde, sözcük vektörlerini oluşturmak için gözetimsiz öğrenme kullanılabilir. Sözcük vektörlerini oluştururken (Mikolov *et al.* 2013) makalesinde skip-gram modeli ile sürekli vektör uzayı temsili kullanılmıştır. Güneş ve Tantuğ (2018) tarafından yapılan çalışmada Milliyet gazetesinin web sitesinden alınan ve daha önceden etiketlenmiş olan Türkçe veriler kullanılmıştır. Kullanılan veri setini yapay sinir ağına eklemek için sözcükler vektörlerden oluşan bir dizi haline getirilmiştir. Sözcük vektörlerini öğretebilmek amacıyla 184 milyon sözcükten oluşan haber yazıları derlemesi kullanılmıştır. Genel olarak bakıldığında, yapay zekâ eğitimi için kelimelerin 3 ana temsil özelliğinden faydalanılmıştır. Bunlar sözcük vektörleri, yazım özellikleri ve biçim bilimsel özelliklerdir. Kelimeleri vektörleştirmek amacıyla açık kaynak kod olan Word2Vec kodları kullanılmıştır. Kelimelerin yazımsal özelliklerini ifade etmek için "hepsi büyük", "hepsi küçük", "ilk harfi büyük", "ayraç var-yok", "nokta var-yok", "sayısal değer değil" gibi kelime özelliklerine bakılmıştır. Biçim bilimsel özelliklerde ise türemiş sözcüklerde son türemedeki sözcük türü etiketi (POS) ve biçim bilimsel etiketler ayrı birer özellik olarak ele alınmıştır. Güneş ve Tantuğ (2018) çalışmasında adlandırılmış sınıf tanıma etiketi olarak ENAMEX ve kelime türü olarak organizasyon (ORG), kişi (PER), yer (LOC) adları kullanılmıştır. Bunların dışında kalan kelimeler, diğer (O) olarak sınıflandırılmıştır. Çalışmada, adlandırılmış varlık tanıma işlemi için LSTM modelinin bir türü olan Bi-LSTM kullanılmıştır (Hochreiter and Schmidhuber 1997).

Güneş ve Tantuğ (2018) makalesinde kullanılan LSTM yapısı Derin Çiftyönlü (deep bidirectional) olarak hazırlanmış yani hem ileri (forward) hem de geri (backward) beslemeli, 50 katmanlı olarak uygulanmıştır. Daha sonra ileri ve geri besleme ile oluşturulan LSTM yapıları birleştirilmiş ve sonuna 100 ve 4 adet iki katmandan oluşan klasik sinir ağı yapısı eklenmiştir. Öğrenme sırasında aşırı öğrenmeden kaçınmak için farklı katmanlarda DBLSTM ve unutma katsayısı kullanılmıştır. En iyi sonuç, 5 katmanlı ve 0.4 unutma katsayılı DBLSTM yapısında elde edilmiştir. Ayrıca, DBLSTM'deki katman sayısı arttıkça başarının arttığı gözlemlenmiştir. Sonuçları değerlendirirken F1-skor değeri kullanılmıştır. Özellikle tek katmanlı yapıdan iki katmanlı yapıya geçildiğinde, başarıda %12,31 puanlık bir artış gözlemlenmiştir. Kütüphane olarak da Tensorflow kütüphanesinin içindeki Keras modülü kullanılmıştır. Çalışmanın sonuçlarını, ENAMEX veri etiketleri ile ölçüm uygulanabilmesi amacıyla, IOB2 (Sang and Veenstra 1999) yapısına dönüştürülmüştür. Ölçme sırasında CoNLL-2003 ölçme stratejisi uygulanmıştır. Eğitim sırasında 3 farklı girdi yapısı oluşturulmuştur. İlk başta sadece sözcük vektörleri tek başına kullanılmış, daha sonra buna yazım özellikleri eklenmiş ve en son olarak da bu ikisine biçim bilimsel özellikler eklenmiştir. En iyi sonuç, 3 girdinin birlikte kullanıldığı zaman, F1 skoru %93,69 olarak elde edilmiştir. Bununla birlikte özellikle yazım özelliklerinin modele eklenmesinden sonra önemli bir artış olduğu gözlenmiştir. Ulaşılan başarı sonucu, Türkçe için oluşturulan adlandırılmış varlık tanıma modellerinde ulaşılmış en iyi sonuç olarak nitelendirilmiştir.

Nasiboglu and Gencer (2021) çalışmasında İngilizce haber yazılarından alınan yaklaşık 47 bin cümlede kişi, yer, kuruluş, tarih ve olay adları tespit edilmiştir. Veri kümesindeki adlandırılmış varlıkları bulmak için önceden eğitilmiş iki farklı model, Stanford ve Spacy kütüphanelerinde NER için oluşturulmuş Derin Öğrenme modelleri kullanılmıştır. Kişi isimlerini tanımada Spacy kütüphanesinin daha iyi olduğu, organizasyon ve yer isimlerini tanımada Stanford kütüphanesinin daha iyi olduğu gözlemlenmiştir. Ayrıca Spacy kütüphanesinin eğitim süresi açısından Stanford kütüphanesinden daha verimli olduğu tespit edilmiştir.

Shen *et al.* (2017) makalesinde derin öğrenme (Deep Learning) yöntemi ile aktif öğrenme yöntemi birleştirilerek, bir adlandırılmış varlık tanıma uygulaması yapılmıştır. Genellikle derin öğrenme

yapabilmek için çok sayıda veri gerekmektedir. Söz konusu makalede bu soruna çözüm olarak oluşturulan bir yapı gösterilmiştir. Yapı, ana hatlarıyla CNN-CNN-LSTM olarak tasarlanmıştır. Makalede geçen CNN (Convolutional Neural Network) modeli, temel olarak incelenmiş ve açıklanmıştır. İlk olarak LeCun tarafından önerilen CNN uygulaması resim tanıma işlemlerinde ve görüntü işlemede çok tercih edilen bir yöntemdir (LeCun *et al.* 1990, 1998).

Shen *et al*. (2017) makalesinde, ilk CNN yapısı karakter kodlayıcı (character encoder), ikinci CNN yapısı kelime kodlayıcı (word encoder) olarak ve son olarak LSTM ise etiket çözücü (tag decoder) olarak tasarlanmıştır. Karakter kodlayıcı, karakterlerine göre kelimelerin özelliklerini çıkarmak için kullanılır. Kelime kodlayıcı bir kelimenin etrafındaki kelime dizilerine bakarak özellik vektörü oluşturur. Etiket kodlayıcı ise kelimeler dizisinin oluşma veya olma olasılıklarını oluşturur.

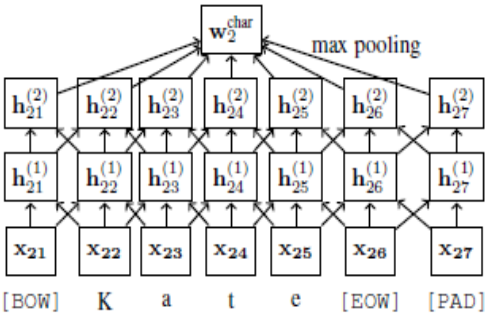
Çizelge 2. Formatlanmış cümle örneği

BOS	Murat	yıllardır	İzmir'de	yaşar.	EOS	PAD
O	B-PER	O	B-LOC	O	O	O

Çizelge 2'de cümle içerisindeki kelimelerin nasıl temsil edildiği gösterilmektedir. Cümlenin başına BOS (Beginning of the sentence) tokeni, cümlenin sonuna ise EOS (Ending of the sentence) tokeni getirilmiştir. Bunlara ek olarak BOW (Beginning of the word) tokeni ve EOW (Ending of the word) tokenleri kullanılmıştır. Birden çok cümlenin hesaplanması için, benzer uzunluktaki cümleler gruplara ayrılmış ve uzunlukları bir demet içinde üniform hale getirmek için cümle sonuna PAD tokenleri eklenmiştir. Biçimlendirilmiş cümle $f(X_{ij})$ olarak belirtilmektedir; burada, $\{X_{ij}\}$, i'inci sözcükteki j'inci karakter bazında bir "one-hot encoding" kodlamasıdır.

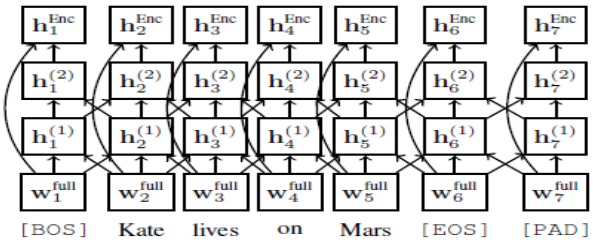
Çalışmada, her i kelimesinin karakter özelliklerini çıkarmak için CNN uygulanmıştır. Şekil 2'de, karakter seviyesinde kodlama (character-level encoding) için iki katmanlı örnek CNN mimarisi gösterilmektedir. Ayrıca her kelimenin karakter seviyesinde kodlama vektörü w_i^{char} ile ifade edilmektedir. Bu işlemlerden sonra karakter seviyesinde özellikler, o kelimeye karşılık gelen gömülü gizli bir kelime ile birleştirilmiştir. Bu vektör w_i^{emb} olarak adlandırılmış. Ayrıca söz seviyesinde girdiler de w_i^{full} ile ifade edilmiştir.

$$w_i^{full} := \left(w_i^{char}, w_i^{emb}\right) \quad (1)$$



Şekil 2. Karakter seviyesinde kodlamaya örnek CNN yapısı (Shen et al. 2017)

Gizli kelime yapılarıyla word2vec eğitimi başlatılmış ve ardından eğitim süresince bu yapılar güncellenmiştir (Mikolov ve ark., 2013). Eğitim verilerinde gizli kelimelere genelleme yapmak için, her kelimeyi, kelime bırakma yöntemine (word-drop method) benzeyen bir yaklaşım olan, eğitim sırasında %50 olasılıkla özel bir [UNK] (bilinmeyen) tokenle değiştirilmiştir. Kelime düzeyde temsiller, her kelimenin pozisyonu kullanılarak CNN ile çıkarılmıştır. Şekil 3'te kelime seviyesinde kodlamaya örnek CNN yapısı gösterilmiştir.



Şekil 3. Kelime seviyesinde kodlamaya örnek CNN yapısı (Shen *et al.* 2017)

Yapının son kısmını da LSTM ile yapılan etiket çözücü (tag decoder) oluşturmaktadır. Etiket çözücü, kelime seviyesi kodlayıcı özelliklerine bağlı olarak etiket dizileri üzerinde bir olasılık dağılımını hesaplar. Bu

da $P\left[y_2, y_3 \ldots y_{n-1} | \{h_i^{Enc}\}\right]$ ile ifade edilmiş olur. Burada popüler olan Koşullu Rastgele Alanlar (Conditional Random Fields-CRF) (Lafferty *et al*. 2001) algoritması etiket çözücü olarak kullanılmıştır. Denklem 2, bu algoritmanın açıklayan denklem gösterilmidir. Bu denklemdeki W, A, b; öğrenebilen parametreler, t_i ise vektörün koordinatlarıdır:

$$P\left[y_2, y_3 \ldots y_{n-1} | \{h_i^{Enc}\}\right] \propto \exp\left(\sum_{i=2}^{n-1} \{W h_i^{Enc} + b\}_{t_i} + A_{t_{i-1} t_i}\right) \quad (2)$$

Sözü geçen çalışmada öğrenme süreci birden fazla turdan oluşturulmuştur. Her turun başında, aktif öğrenme algoritması cümleleri önceden tanımlanmış limite kadar açıklanacak şekilde seçmiştir. Ek açıklamaları aldıktan sonra, artırılmış veri kümesi üzerinde eğitim alarak model parametreleri güncellenmiş ve bir sonraki tura geçilmiştir. Bir cümleyi açıklama maliyetinin cümledeki kelime sayısıyla orantılı olduğu ve seçilen cümledeki her kelimenin bir kerede açıklanması gerektiği varsayılmıştır. Kısmen açıklamalı cümlelere izin verilmemiş veya açıklanmamıştır. Ayrıca, bu çalışmada veri seti olarak CoNLL-2003 English (Sang and Meulder 2003) ve OntoNotes-5.0 English (Pradhan *et al.* 2013) kullanılmıştır.

Mikolov *et al.* (2013) makalesinde kurulan algoritmadaki ana önemli etken hız etkenidir. Sonuç bölümünde verilen çizelgelere göre makale test setindeki verilerde epoch başına 11 saniye gibi aynı alandaki çalışmalara göre çok önemli bir hıza ulaşmıştır. Ayrıca eğitim setinde de epoch başına 22 saniyelik hız ile aynı alandaki çalışmalara göre en iyi hıza ulaşmıştır. Hız anlamında eşit olduğu çalışmalarda da F1-skoru başarısı (%90,69) daha yüksek olarak gözlenmiştir.

Ahmet'in babası beyaz koyunu araba ile köye getirdi.

a) Normal metin

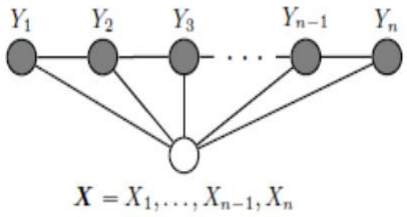
<İsim>Ahmet<İsim><TamlayanEki>in<TamlayanEki><İsim>baba<İsim><TamlananEki>sı<TamlananEki><Sıfat>beyaz<Sıfat><İsim>koyun<İsim><NesneEki>u<NesneEki><İsim>araba<İsim><DiğerZarflar>ile<DiğerZarflar><İsim>köy<İsim><DolaylıTümleçEki>e<DolaylıTümleçEki><Yüklem>getir<Yüklem><ZamanEki>di<ZamanEki>

b) Uygun Söz dizimsel metin

Şekil 4. Söz dizimsel analiz kullanarak etiketlenme örneği.

Özkaya and Diri (2011) makalesinde Şartlı Rastgele Alan (Conditional Random Field - CRF) (Wallach 2004) yöntemi kullanılarak, kural tabanlı adlandırılmış varlık tanıma işlemi gerçekleştirilmiş ve eposta metinlerindeki cümlelerde adlandırılmış varlık tanıma üzerine çalışılmıştır. Sıralı veri, söz dizimsel analiz kullanılarak etiketlenmiştir (Şekil 4).

Şartlı Rastgele Alan algoritmasında dizili kelimeleri işaretlemek veya bölümlendirmek amacıyla kullanılan, Maksimum Gizli Markov Modeli ve Entropi Markov Modelinin genel durumunu gösteren bir olasılık ortaya çıkar. Şartlı Rastgele Alanda Şekil 5'de gösterildiği gibi, *y* gibi belirli bir işaret dizisinin *x* değeriyle şartlı olasılığını hesaplamak için, yönsüz çizge modeli kullanılmaktadır.



Şekil 5. *y* işaret dizisinin *x* değeriyle şartlı olasılığını hesaplamak için yönsüz çizge modeli.

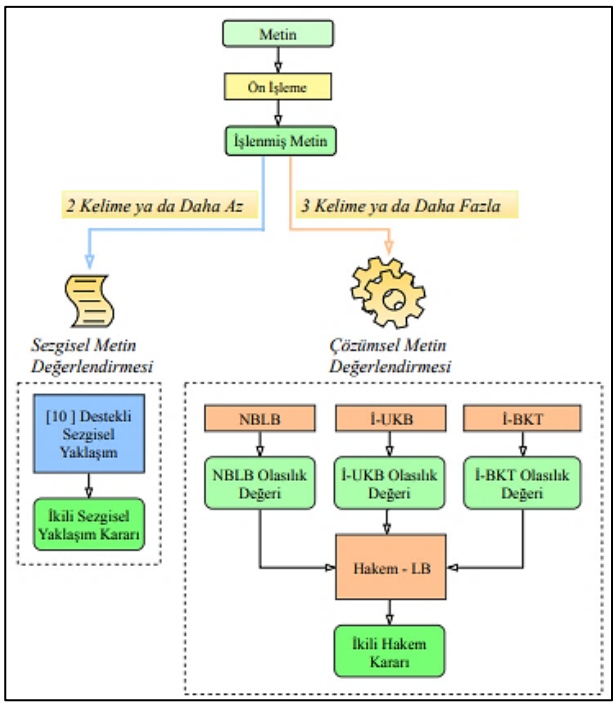
Özkaya and Diri (2011) çalışmasında veri seti, akademik, kurumsal ve kişisel olmakla, toplam 150 mail metni ile çalışılmıştır. Önemli özelliklerin ortaya konmasına yardımcı olan unvanların, bazı özel kelimelerin ve kısaltmaların olduğu sözlüklere de ihtiyaç duyulmuştur. O yüzden, bu çalışmada 175 adet kısaltma listesi (İng., Fr., Tşk, Sok., Mah., TD, Sn, A.Ş., Ltd., ...), 35 adet unvan listesi (Prof., Dr., Av., Gen., …), ve 32 adet özel kelimeler listesi (Sayın, Hanım, Bey, Hocam, Üniversite, Bakanlık, Hastane, Dağ, Tepe, ...) kullanılmıştır. Çalışmada e-posta metinlerindeki kelime özelliklerini elde ederken 3 ana yapıdan yararlanılmıştır. Bunlar başlık bilgisi, etiketlenebilen özellikler ve kural tabanlı özelliklerdir. Başlık bilgisi "From, To, Cc, Bcc, Send, Sender, In-Reply-to, Reply-to, Forwarded by" gibi başlık kalıplarından sonra varlık adı geleceği düşünülerek, o kısımlardaki varlık isimleri araştırılmıştır. Etiketlenen özellikler kısmında kelimenin büyük harf ile başlayıp başlamaması,

içinde nokta, virgül gibi noktalama işareti içerip içermemesi, cümlenin ilk kelimesi olup olmaması, kelimenin başlık bilgisinde olup olmaması, kelime unvan listesinde veya özel kelimeler listesinde yer alıyor mu almıyor mu gibi filtreler ile özellikler oluşturulmuştur. Kural tabanlı özelliklerde her kelimeden önce ve sonra gelen kelimeler belirlenerek isimler, soy isimler yakalanmaya çalışılmıştır. Daha sonra 1'li, 2li ve 3'lü n-gramlara bakılarak bazı son ekler belirlenmeye çalışılmıştır. Ayrıca kelime uzunluğuna bakılarak da o kelimenin kısaltma veya unvan adı olup olmayacağının tahmininde işe yarayacağı düşünülmüştür. Son olarak da art arda gelen kelimelerde isim olarak nitelendirilebilecek kelimenin hangisinin isim, hangisinin soy isim olabileceği belirlenmeye çalışılmıştır. Bu çalışmada üç farklı sınıf (yer, kişi, kurum ismi) için 50 farklı eposta taranmıştır. Adlandırılmış varlık tanınmasında en başarılı sonuçlar %87 ile kurumsal epostalar olmuştur. En düşük doğru tanıma oranı ise %72 ile kişilerde olmuştur. Bu çalışmada doğru tanımadaki en büyük etkenlerden, genellikle noktalama işaretlerinin doğru yerde kullanılması ve yazım yanlışı yapılmaması olduğu vurgulanmıştır.

Schiersch *et al.* (2020) çalışmasında, sokak adları, durak adları ve güzergâh adları gibi coğrafi varlıklarla ve standart adlandırılmış varlık türleriyle açıklamalı Almanca veri seti oluşturulmuştur. Ayrıca, kazalar, trafik sıkışıklıkları, satın almalar ve grevler gibi trafik ve endüstri ile ilgili n-li (birbiri ile ilişkili n adet varlık adının bulunması) ilişkileri ve olayları içeren bir dizi 15 açıklama eklenmiştir. Veri seti genel olarak online gazete, radyo istasyonları, polis ve demiryolu şirketlerinden gelen haber metinleri, Twitter mesajları ve trafik raporlarından oluşturulmuştur. Bu çalışma, coğrafi varlıkların açıklanmasını amaçlayan hem adlandırılmış varlık tanıma algoritmalarının hem de n-li ilişki çıkarma sistemlerinin eğitimine ve değerlendirilmesine olanak tanıyan bir çalışma olmuştur. Veri setini oluşturmak için Twitter Search API, uberMetrics Search API, RSS Feeds gibi metin sağlayıcı servisler kullanılmıştır. Sadece Almanca metinleri çekebilmek için Python ile yazılmış "langid" kütüphanesi kullanılmıştır. Adlandırılmış varlık tanıma çalışması için Stanford Core NLP (Manning *et al.* 2014) araçlarından yararlanılmıştır. Ayrıca varlıklar arasında ilişki çıkarımı yapmak için DARE algoritması kullanılmıştır. DARE, serbest metinler üzerinde ilişki çıkarımı için minimal denetimli bir makine öğrenme sistemidir. DARE algoritması, tüm ilişki argümanlarını birbirine bağlayan minimum bağımlılık alt graflarını öğrenir (Krause *et al*. 2012).
Sarı ve Aktaş (2018) makalesinde ders içeriği olarak hazırlanmış coğrafya ve tarih metinlerinin içerisinde adlandırılmış varlık adı bulunmaya çalışılmıştır. Bu çalışmada amaç, girdi metinlerde varlık adlarını belirleyerek terimler sözlüğü oluşturabilmekten ibarettir. Oluşturulan sistem, cümleler üzerine çalışan ve kural tabanlı bir sistemdir. Cümleleri belirlemek için öncelikle gereksiz boşluk, karakter ve semboller çıkarılmış, sonra cümle sonlarını belirlemek için regex (regular expression) kullanılmıştır. Regex işlemi sırasında oluşabilecek hatalar için Türkçe kısaltmaların olduğu bir sözlükten yararlanılmıştır. Cümle sonları belirlendikten sonra birimleştirme (tokenize) işlemi yapılmıştır. Sözcükleri birimleştirmek için her sözcüğün; büyük harf içerip içermediği, tamamının büyük harften oluşup oluşmadığı, son karakterinin nokta olup olmaması, sayı içerip içermemesi (içerdiği sayıların gün ay yıl sayısı olması da işaretlenmiş), kesme işareti, virgül, noktalı virgül, parantez içermesi, yüzde içermesi dikkate alınmıştır.

Yi *et al.* (2021) çalışmasında FastText modelini kullanarak sadece kelimelerin anlamlarını değil aynı zamanda morfoloji bilgilerini de dikkate almaktadırlar. Bunun yanında, LSTM modeli kullanılarak bağlam akışı konusunda eğitim yapılmış, önceki çalışmalarda önerilen metodolojilerle tespit edilemeyen küfürler tespit edilmiştir. Önerilen yöntemle 40005 küfürlü ve 40254 küfürsüz cümleden 40126 cümle küfür, 40133 cümle küfür değil olarak tahmin edilmiştir. Sınıflandırma performans testi göstergelerine göre doğruluk oranı %96,15, geri çağırma oranı %96,29 ve kesinlik %96 olarak tespit edilmiştir. Önceki bir çalışmada önerilen düzenleme mesafesi (edit distance) algoritması ile bu çalışmada önerilen yöntem arasındaki karşılaştırmalı analizin sonucu, önerilen

yöntemin daha doğru küfür tespiti yapabildiğini doğrulamıştır.



Şekil 6. Türkçe küfür tespit için örnek model yapısı (Çelik ve Yıldırım 2020).

Ratadiya and Mishra (2019) küfür tespiti çalışmasında, gösterilen katkılar üç kısımda sıralanmıştır. Çoğu yaklaşımda kullanılan tekrarlama mekanizması tamamen atlanmasına rağmen iyi sonuçlar elde edilmiştir. Tekrarlama olmadığında diziyle ilgili bilgileri sağlamak için konumsal kodlama etkin bir şekilde kullanılmıştır. Son olarak, birleştirme kullanarak padding'e rağmen bir dizide bulunan maksimum bilginin etkili bir şekilde tutulduğu gösterilmiştir. Önerilen yöntemde, her modelin tahminine ve modelin doğruluğuna göre bir ağırlık atanmıştır. Bireysel ağırlık değeri 0 ile 1 arasındadır ve tüm modellere atanan ağırlıkların toplamı 1'e eşit olmalıdır. Tahminlerin ağırlıklarla çarpımlarının toplamı nihai tahmin olarak kabul edilmektedir. Ağırlıklı ortalama tahminleri denklem 3'te gösterilmiştir. P_i, i'inci modelin tahminini ve W_i ise i'inci modele atanan katsayıyı göstermektedir.

$$AğrılıklıOrtalamaTahmini = \sum_{i=1}^{n} P_i W_i \qquad (3)$$

Çelik ve Yıldırım (2020) küfür tespiti çalışmasında, 3 faklı yapay zekâ modelinin döndürdüğü olasılıklara bakarak, hakem modelin karar vermesi sağlanmıştır. Şekil 6'da çalışmanın ana yapısı gösterilmektedir.

Çelik ve Yıldırım (2020) çalışmasında, metin ön işleme aşamasında kelimeler küçük harf yapılmış, http benzeri bağlantılar kaldırılmış, hatalı kelimeler düzenlenmiş, dolgu kelimeleri yok edilmiş, metnin içerisinde geçen rakamlar silinmiş, kelimelerdeki yinelenen harfler yeniden düzeltilmiş, kısaltma olarak yazılmış kelimeler yeniden düzeltilmiş, noktalama işaretleri silinmiş, tek harfli heceler ve tüm fazla olan boşluklar kaldırılmıştır. Yazarlar çalışmalarının sezgisel model kısmında Ratcliff-Obershelp algoritması yardımıyla metin içerisindeki kelimeler ile küfür listesi içerisindeki kelimelerin dizi benzerliğini (string similarity) bulmuştur. Elde edilen dizi benzerliği belirlenen eşiği aşarsa sözcük küfür olarak tanımlanmıştır. Ayrıca Java tabanlı Türkçe metin işleme kütüphanesi olan Zemberek kullanılarak kelimelerin sonuna farklı Türkçe ekler getirilmiş ve dizi benzerlikler o şekilde elde edilmiştir. Yapay zekâ tabanlı model kısmında ise 3 farklı model kullanılmıştır. Bunlar: Naif Bayes tabanlı Lojistik Bağlanım (NBLB), İki Yönlü Uzun Kısa Süreli Bellek (İUKB) ve İki Yönlü Kapılı Tekrarlayan Hücre (İBKT) modelleridir. Son kısımda ise lojistik tabanlı hakem bir model ile karar verilmektedir.

Yılmaz vd. (2022) çalışmasında Twitter platformundan elde edilen bir veri seti oluşturulmuştur. Türkçe tweet metinlerden oluşan bu veri seti, etiketleyiciler tarafından el ile etiketlenmiş ve LSTM ve GRU modellerinin sınıflandırma performansları karşılaştırılmıştır. Türkçe için saldırgan dil konusunda çoklu sınıflandırmanın yapıldığı ilk çalışmadır. Burada Word2vec yöntemi ile kelime temsilleri elde edilmiştir. Böylece genişletilmiş korpus kullanımının sınıflandırma performanslarına katkısı karşılaştırılmıştır. Ayrıca toplanan veri setinde genişletilmiş derlem ve etiketli veri olmak üzere iki farklı veri kullanılmıştır. Bu verilerin farkı, ilkinin görece olarak daha fazla veriden oluşması ve diğerinin ise etiketli olmasıdır. Amaç, algoritmaların çalışma başarılarını ve performanslarını karşılaştırmaktır. Sınıflandırma işlemi ise üç aşamalıdır. Birinci aşamada "saldırgan" ve "saldırgan değil" olarak ikili sınıflandırma yapılmıştır. İkinci aşamada saldırgan olan tweet metni "hedefli" ve "hedefsiz" olarak ayrılmıştır.

Böylelikle iki aşamada sonucunda, "hedefli", "hedefsiz" ve "saldırgan değil" olarak çoklu sınıflandırma yapılmıştır. Üçüncü aşamada ise, hedefli olan tweet metni "birey", "grup", "diğer" olarak ayrılmıştır. Tüm aşamalar sonucunda ise "saldırgan değil", "hedefsiz", "birey", "grup" ve "diğer" olarak çoklu sınıflandırma yapılmıştır (Yılmaz vd. 2022).

3. Materyal ve Metot

3.1 NER Etiketleme Yöntemleri

Adlandırılmış varlık tanıma çalışmalarında kullanan pek çok etiketleme yöntemi vardır. Bunlardan bazıları part of speech (POS) (Schmid 1999), IO, IOB (Ramshaw and Marcus 1995) ve IOB2 gibi pek çok etiketleme yöntemi bulunmaktadır. POS yöntemi daha çok kelimelerin cümledeki görevini anlamak için kullanılır. IO, IOB ve IOB2 gibi etiketleme yöntemleri ise kelimeleri gruplandırmak veya sınıflandırmak için kullanılmaktadır. Bu etiketleme yöntemi sayesinde hedef varlık adını ve diğer grupları birbirinden ayırabiliriz.

3.1.1 Part of Speech (POS) Tagging

Adlandırılmış varlık tanımlama uygulamalarında kelimeleri daha iyi tanımlamak ve anlamak için en çok kullanılan yöntemlerden bir tanesi de konuşma bölümlerini etiketlemedir (part of speech (POS) tagging). Konuşma bölümlerini etiketleme, bir cümledeki her bir kelimenin, o kelimenin konuşmanın hangi kısmı (örneğin, İsim, Fiil, Sıfat, vb.) olduğunu etiketlemek için kullanılır. Bunu yapan bir bilgisayar programı, bir metin alanı girdisi almakta, metni bir kelime listesine ayrıştırmakta ve ardından her bir kelime ile birlikte bu kelimenin konuşmanın hangi bölümü olduğunu içeren, aynı boyutta bir liste döndürmektedir. Konuşma bölümlerini etiketleme, hem insanların hem de bilgisayarların aynı kelimenin farklı kullanımlarını ayırt etmesine ve bir kelime hakkında daha fazla bağlam vermesine olanak tanımaktadır. Örneğin, "kaz" kelimesi "kaz gördüm" cümlesinde fiil iken "toprağı kazıyorum" cümlesinde isimdir. Konuşma bölümlerini etiketlemenin tüm uygulamaları, üzerinde eğitim almak için etiketlenmiş bir bütünlük gerektirir. Bu bütünlük, çoğu kelimenin net olduğunu, yani kelimenin herhangi bir kullanımında aynı etikete sahip olduğunu gösterir. Bir örnek olarak "güzel" kelimesinin her zaman bir sıfat olması olabilir.

3.1.2 IOB Tagging

IOB formatı, (iç, dış, başlangıç kısaltması), hesaplamalı dilbilimde bir yığınlama görevinde belirteçleri etiketlemek için yaygın bir etiketleme biçimidir (Collobert *et al.* 2011). Bu yöntem ilk defa, Ramshaw and Marcus (1995) tarafından sunulmuştur. Bir etiketin önündeki "B", etiketin bir yığının başlangıcında olduğunu gösterir. Bir "O" etiketi, bir belirtecin hiçbir parçaya ait olmadığını gösterir. Bir etiketten önceki "I" öneki, etiketin, aralarında "O" etiketleri olmayan diğer bir parçayı hemen takip eden bir yığının içinde olduğunu belirtir. "O" etiketinden sonra bir yığın geldiğinde, yığının ilk simgesi "B" önekini alır. Örnek olarak Şekil 7'de, IOB ve başka bir etiketleme yapısı olan IO (içinde-dışında) etiketleme yapıları gösterilmiştir. Yaygın olarak kullanılan diğer bir benzer biçim, her yığının başında B etiketinin kullanılması dışında IOB biçimiyle aynı olan IOB2 biçimidir (yani, tüm parçalar B etiketi ile başlar).

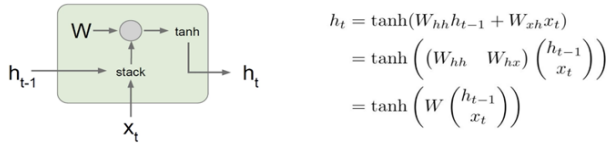
	IO Kodlama	IOB Kodlama
Mehmet	PER	B-PER
Edvard	PER	B-PER
Munch	PER	I-PER
'un	O	O
resmini	O	O
Ahmet	PER	B-PER
'e	O	O
gösterdi	O	O

Şekil 7. IOB etiketleme örneği.

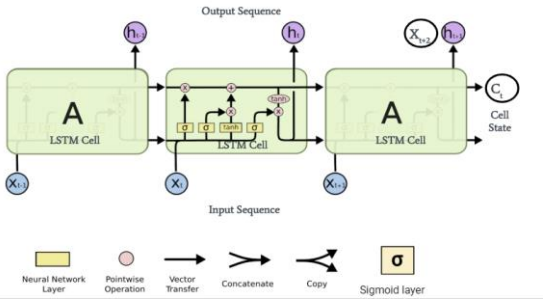
Çalışmamızda IOB etiketleme yöntemi kullanılmıştır. POS etiketleme yöntemi cümlenin ögelerinin anlaşılması gereken durumlarda kullanılmaktadır. Küfürlerin cümlenin hangi ögesi olacağı değişiklik göstereceği için bu yöntem tercih edilmemiştir. Tek başına küfür olan bir kelimenin B ile etiketlenmesi halinde kelimelerin başlangıçlarını yakalamaya faydası olacağı düşünüldüğünden, bu çalışmada IOB yöntemi kullanılmıştır.

3.2 LSTM (Long Short Term Memory)

LSTM, yinelenen sinir ağının (RNN) bir alt dalıdır. LSTM genellikle sıralı veya zamana bağlı olarak değişen dinamik yapıları tahmin etmek için kullanılır. Klasik sinir ağlarındaki tüm veri girişleri ve bu girdilere bağlı oluşan sonuçlar diğer giriş ve çıkışlardan ayrı veya kopuk olarak oluşmaktadırlar. Bununla birlikte genellikle, doğal dil işleme konusu gibi sıralı bilgi içeren yapılarda klasik sinir ağları pek iyi sonuçlar ortaya koyamamaktadır. Böyle bir durumda devreye yinelenen sinir ağları (RNN) girmekte ve eldeki dizinin her bir elemanı için aynı işi, önceki sonuçları dikkate alarak gerçekleştirmektedir. Bu sayede dizili durumdaki girdilerin bütün sıralama yapısı veya başka bir deyişle şeması muhafaza edilmiş olur. Yinelenen sinir ağlarındaki tüm çıktılar kendinden önceki elemanların çıktılarına bağlı olarak değişmekte veya meydana gelmektedir. Fakat kelimeler arası mesafe arttıkça RNN'in önceki verileri kullanması zorlaşır.



Şekil 8. Yinelenebilir Sinir Ağı (RNN) yapısı.



Şekil 9. Uzun Kısa Vadeli Hafıza Ağları (Long Short Term Memory-LSTM) yapısı.

Şekil 8'de, X_t girdi, h(t-1) bir önceki adımda üretilen gizli durum (hidden state), W ağırlık matrisi ve *tanh* ise ağrılık fonksiyonunu oluşturmaktadır. RNN yapısı, "kayıp gradyan" (vanishing gradient) ve "taşan gradyan" (exploding graident) problemleri yüzünden, yani oluşan çıktı değerinin çok küçük veya çok büyük çıkması yüzünden, günümüzde çok tercih edilmemektedir. RNN'in bir alt yapısı olan LSTM (Long Short Term Memory) ise sağladığı avantajlardan ötürü daha çok kullanılmaktadır (Şekil 9).

Şekil 9'da görüleceği üzere, birleşen oklar vektörlerin bir araya gelmesini, okların ayrılması ise kopyalanarak iki vektörün meydana gelmesini göstermektedir. Ayrıca, LSTM yapısında duruma göre birden farklı aktivasyon fonksiyonu da kullanılabilmektedir. Bunlara ek olarak hücre durumu (cell state), LSTM için çok önemli bir elemandır. Hücre durumu, bir hücreden diğer hücreye veri geçişini düzgün bir biçimde gerçekleştirmektedir. Ayrıca, hücre durumu güncellenerek optimize edebilmektedir. LSTM kendini güncel tutmak için bazı geçitlere sahiptir. Bunlar, unutma kapısı (forget gate), girdi kapısı (input gate) ve çıktı kapısı (output gate) kapılarıdır.

$$f_t = \sigma\left(W_f.[h_{t-1}, x_t] + b_f\right) \quad (4)$$

$$i_t = \sigma(W_i.[h_{t-1}, x_t] + b_i) \quad (5)$$

$$\acute{C}_t = tanh(W_c.[h_{t-1}, x_t] + b_c) \quad (6)$$

$$C_t = f_t * C_{t-1} + i_t * \acute{C}_t \quad (7)$$

Unutma kapısı, kendisine gelen verilerin hangisini unutacağına *sigmoid* fonksiyonu ile denklemde gösterildiği gibi karar verir. Çıkan değerler 0'a yakınsa, unutmaya yakın davranır. Değerler 1'e yakınsa, hiç değişiklik yapmaz ya da yakınlık oranına uygun olarak az değiştirir. Girdi kapısı, genellikle hangi bilgilerin sonradan kullanılacağına karar verir. Sigmoid fonksiyonu sayesinde Girdi kapısı hangi değerlerin kullanılması gerektiğine (5) numaralı fonksiyonu kullanarak karar verir. (6) numaralı *tanh* fonksiyonunda ise hücre durumu üzerine eklenmeye aday olan verilerden bir vektör oluşturulur. Sonrasında, bu oluşan iki vektör birleştirilerek Hücre durum vektörü üzerine eklenir. Daha sonra da Girdi ve Unutma kapılarından gelen bilgiler kullanılarak (7) numaralı denklemde gösterildiği gibi hücre durumu güncellenir. Hücre durumuna göre de her seferinde (8) ve (9) denklemleri ile çıktı vektörü güncellenir:

$$o_t = \sigma(W_o.[h_{t-1}, x_t] + b_0) \quad (8)$$

$$h_t = o_t * tanh(C_t) \quad (9)$$

Ayrıca LSTM'de geri besleme (backpropagation) yapılırken de RNN'in aksine, her seferinde aynı W ağırlık matrisi ile çarpılmak yerine, her adımda farklı bir unutma kapısı ile çarpıldığı için gradyan kaybı veya taşması sorunundan da kurtulmuş olunur.

3.3 Bi-LSTM

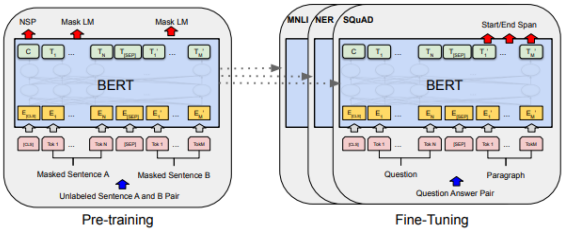
Bi-LSTM (Bidirectional Long Short Term Memory) modelinde, tek bir model eğitilmek yerine iki model eğitilmektedir. İlk model sağlanan girdinin sırasını öğrenir ve ikinci model bu sıranın tersini öğrenir. Eğitilmiş iki model olduğundan dolayı, ikisini birleştirmek için bir mekanizma oluşturulması gerekiyor. Bu adım genellikle birleştirme adımı olarak adlandırılır.

3.4 BERT

BERT (Bidirectional Encoder Representations from Transformers), transformatörlerden çift yönlü enkoder temsilleri anlamına gelir ve Google tarafından geliştirilmiş bir dil gösterimi modelidir (Devlin *et al.* 2018). BERT modeli, doğal dil çıkarımı ve yorumlama gibi cümle seviyesindeki görevleri, bunları bütünsel olarak analiz ederek, belirteçlerin tanımlanması ve soru cevaplaması gibi belirteç düzeyinde görevleri, modellerin ince taneli (fine-grained) çıktı üretmesi için gerekli olduğu belirteç seviyesi görevleri gerçekleştirebilir.

BERT modeli, temelde maskelenmiş dil modeli mantığını kullanır. Maskelenmiş dil modelinde, rastgele bazı belirteçler girişten maskelidir ve amaç, sadece bağlamsal, yani içerik olarak, maskeli kelimenin orijinal kelime kimliğini tahmin etmektir. Örneğin, Türkçe'de ki "yaz" kelimesi hem fiil hem de isim anlamına gelebilmektedir. Model, buradaki ayrımı yaparak "yaz" kelimesinin anlamına göre bir vektör ortaya çıkarmaktadır. Yani kelime mevsim olan "yaz" ise ayrı, fiil olan "yaz" kelimesi ise ayrı bir vektör oluşturmaktadır. Ayrıca, sağdan sola dil modelinin aksine BERT modeli, hedef kelimenin hem sağından hem solundan örnekleri birleştirerek çift yönlü derin dönüştürücülerin (bidirectional deep transformers) eğitimine izin verir. Bunlara ek olarak, BERT modeli bir sonraki cümle tahmini için de kullanılabilmektedir.

BERT, ön eğitimli modellerin göreve özgü mimariye olan ihtiyacı azalttığını göstermektedir. BERT, çok görevli mimarinin daha iyi performans gösteren, hem büyük bir cümle düzeyi ve hem de belirteç seviyesi görevleri üzerine son teknoloji performansı elde eden ilk ince ayar (fine-tuning) temelli gösterim modelidir.



Şekil 10. BERT için genel ön eğitim ve ince ayar prosedürleri (Devlin *et al.* 2018).

Şekil 10'da BERT için genel ön-eğitim ve ince ayar prosedürleri gösterilmektedir. Çıktı katmanlarının yanı sıra, aynı mimariler hem eğitim öncesi hem de ince ayarlarda kullanılmaktadır. Önceden eğitilmiş model parametreleri, farklı aşağı akış görevleri için modelleri başlatmak amacıyla da kullanılır. İnce ayarlama sırasında, tüm parametreler ince ayarlanmaktadır. Şekilde, [CLS], her giriş örneğinin önüne eklenen özel bir semboldür ve [SEP], özel bir ayırıcı belirteçtir (örneğin, soruları / cevapları ayırır).

4. Bulgular

4.1. Veri Seti ve Ön işlemler

Bu çalışmada, derin öğrenme tabanlı Bi-LSTM ve BERT modelleri kullanarak Türkçe tweet metinleri üzerinde küfür, hakaret veya aşağılayıcı kelime tespitine çalışılmıştır. Çalışmada kullandığımız veri seti Github üzerinden duygu durumu analizi ile alakalı bir çalışmadan alınmıştır (1-internet kaynakları). Bu veri seti, Twitter'dan alınmış Türkçe tweetlerden oluşmaktadır. Veri seti aslında cümlede duygu analizi gerçekleştirmek için oluşturulmuş bir veri setidir. Ancak içerisinde Türkçe hakaret ve küfür içeren tweetler bulunduğu için amaca uygun olarak etiketleme yapılmış ve küfür tespitine çalışılmıştır. Veri setinde toplamda 15110 cümle bulunmaktadır. Çizelge 3'te, veri setindeki birkaç örnek cümle verilmiştir. Cümle içerisinde geçen küfürlü kelimeleri tespit etmek amacıyla kelimeler IOB

tagging yöntemine göre etiketlenmiştir (Çizelge 4). Ayrıca, cümleler içerisindeki kelimeleri etiketlemek için kullanılan küfürlü kelimelere internet kaynakları bölümünden ulaşılabilir (2- internet kaynakları).

Çizelge 3. Veriden alınmış örnek tweetler.

Şerefsizlik, sözde sanatçıların vazgeçemediği bir değerdir
Kendisi de bilmiyordur çünkü beyinsiz
En uzun yolculuklar bile, tek bir adımla başlar. Geleceğin mimarları gençlerimiz için atılan ilk adımları var gücümüzle desteklemeye devam ediyoruz.
Merhaba, konuyla ilgili yardımcı olmak isteriz. İrtibat numaranızı direkt mesaj olarak bize iletir misiniz?
Çok güzel bir çalışma olduğuna inanıyoruz. Yazımızda iki bölümün de avantaj ve dezavantajlarını değerlendirmeye çalıştık. Unutmayınız, bu bir üstünlük konusu değil. İşimiz ve amacımız bilim.

Çizelge 4: Etiketlenmiş örnek cümle.

Kelime	Etiket
şerefsizlik	B-Profanity
sözde	O
sanatçıların	O
vazgeçemediği	O
bir	O
değerdir	O

Veri ön hazırlık aşamasında aşağıdaki işlemler yapılmıştır:

- Cümle içerisindeki tüm noktalama işaretleri kaldırılmıştır,
- Tüm harfler küçük hale getirilmiştir,
- "@" işareti ile başlayan ve twitterda kişi etiketleme için kullanılan bahsetmeler kaldırılmıştır.

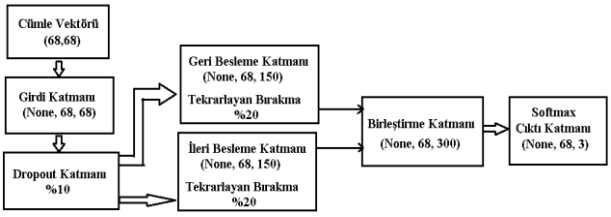
Veri etiketleme aşamasında aşağıdaki işlemler yapılmıştır:

- Ön hazırlıktan geçen kelimeler tokenleştirilerek ayrı tokenler haline getirilmiştir.
- Her token ve sonrasında gelen kelime bir küfür veya hakaret içeren söze eşit mi diye kontrol edilerek eldeki küfür sözlüğüyle tokenler eşlenmiştir. Etiketleme işlemi IOB etiketleme sistemine göre gerçekleştirilmiştir.
- Etiketlenmeyen veya sözlükte bulunmayan kelimeler manuel olarak kontrol edilmiş ve eşlenme sağlanmıştır.

4.2 Model Mimarisi

Çalışmamızda BİLSTM Model Mimarisi kullanılmıştır. Model mimarisi oluşturulurken tüm cümleler en uzun cümle ile aynı boyda vektörlere dönüştürülmüştür. Vektör boylarını eşitlemek için vektörler sıfır ile doldurulmuştur. Cümle içerisindeki kelimeler vektörleştirilirken kategorik vektörleştirme uygulanmıştır. Her benzersiz sözcük eşsiz bir tek sayı ile temsil edilmiştir. Oluşturulan cümle vektörleri modellerin giriş katmanına en uzun cümlenin boyu kadar, yani 68 sayı ile verilmiştir. Girdi katmanından sonra ezberlemeyi engellemek için yüzde 10'luk bir bırakma (dropout) katmanı ilave edilmiştir.

BiLSTM modeli hafıza yapısına sahip olduğu için ezberlemeyi önlemek amacıyla sinir ağı katmanına %20 tekrarlayan bırakma (recurrent dropout) eklenmiştir. Çift Yönlü beslemeli sinir ağı katmanından sonra da birleştirme katmanı eklenmiştir. İleri ve geri yöndeki sinir ağları 150 adet nörondan oluşturulmuştur. Bu sayede birleştirme katmanı tek yönlü katmanların iki katına, 300 adet nörona çıkarılmıştır. Daha sonra çıktı katmanına softmax aktivasyon fonksiyonu eklenmiştir. Bu katman ise 68 satır 3 sütundan oluşan tahmin vektörünü döndürmektedir (Şekil 11).



Şekil 11. BİLSTM Model Mimarisi

BiLSTM modelinde aktivasyon fonksiyonu olarak softmax kullanılmıştır. Yinelemeli aktivasyon fonksiyonu olarak ise sigmoid fonksiyonu tercih edilmiştir. Sigmoid fonksiyonu dönüş değerleri 0 ile 1 aralığında olduğundan çıktı katmanı hızlı bir şekilde yakınsamayı sağlamıştır. 10 ve 11 numaralı fonksiyonlarda sırasıyla sigmoid ve softmax fonksiyonları verilmiştir.

$$f(x) = \frac{1}{1+e^{-x}} \tag{10}$$

$$\sigma(\boldsymbol{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{11}$$

4.3. Değerlendirme metrikleri

Bu çalışmadaki farklılıklardan biri, cümleyi bir bütün olarak ele alıp tüm cümleyi hakaret içeriyor veya içermiyor diye değerlendirmek yerine, cümleyi tokenleştirerek her kelimenin hakaret olup olmama olasılığı belirlenmeye çalışılmıştır. Modelin sonunda elde edilen, hakaret içeren kelimeler sansürlenmiş, cümlenin geri kalanı ise olduğu gibi gösterilmiştir. Değerlendirme ölçüsü olarak doğru pozitif (True Positive-TP), doğru negatif (True Negative-TN), yanlış pozitif (False Positive-FP), yanlış negatif (False Negative-FN) değerleri baz alınmıştır (Çizelge 5).

Çizelge 5. Karışıklık Matrisi (Confusion Matrix).

	Actually Positive	**Actually Negative**
Predicted Positive	True Positives (TP)	False Positive (FP)
Predicted Negative	False Negatives (FN)	True Negatives (TN)

Modellerin keskinlik (precision), hassasiyet (recall) ve F1-skor ölçümleri, TP, TN, FP ve FN değerleri kullanılarak aşağıdaki şekilde hesaplanmıştır:

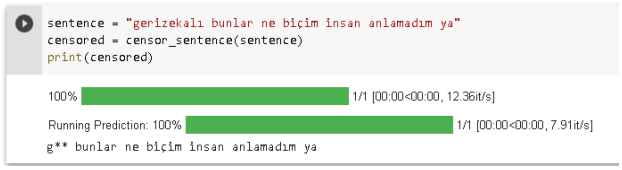
$$Kesinlik = \frac{TP}{TP+FP} \quad (12)$$

$$Hassasiyet = \frac{TP}{TP+FN} \quad (13)$$

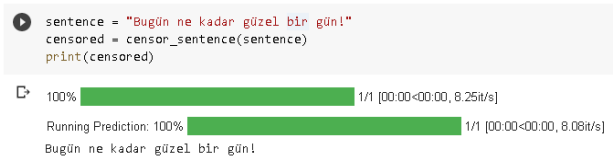
$$F_1 = 2 * \frac{Kesinlik*Hassasiyet}{Kesinlik+Hassasiyet} \quad (14)$$

5. Tartışma ve Sonuç

Kullanılan modellerin eğitimi, Google Colab üzerinde sağlanan GPU (Graphical Processor Unit) Tesla K80 isimli sanal makine ile gerçekleştirilmiştir. Sanal makine, 33 MB CPU (Central Processing Unit) ve 1.4 GB GPU'ya sahiptir. Eğitim ve test setleri %80'e %20 olacak şekilde ayarlanmıştır. Şekil 12 ve 13'te, örnek model çıktıları gösterilmiştir. Eğitilmiş modele verilen cümleler önce tokenleştirilmiş, sonra da modelin tahminleme yapması sağlanmış ve yapılan tahminlere göre sansürlenmiş cümle şekline döndürülmüştür.



Şekil 12. Hakaret içeren cümle ve sansürlenmiş model tahmin çıktısı.



Şekil 13. Normal bir cümle ve model tahmin çıktısı.

Çizelge 6 ve 7'de, Bi-LSTM ve BERT modellerinin çalışma sonuçları gösterilmiştir. Bi-LSTM modeli hem eğitimde hem de test aşamasında %99'a yakın doğruluk oranı (accuracy) sergilemiştir. BERT modeli ise eğitim aşamasında %99 civarında doğruluk oranı gösterirken, test doğruluk oranı %95 civarında gözlemlenmektedir. BERT modeli özelinde test başarısının eğitim başarısına göre düşüş nedenlerinden birisi, eğitim yapılırken seçilen epoch sayısının azlığı olarak düşünülmektedir. Ayrıca iki modeli eğitim süreleri açısından karşılaştırmak gerekirse; Bi-LSTM modeli her adımı 840 milisaniye civarında gerçekleştirirken, BERT modeli her adımı 2,2 saniye civarında tamamlamıştır. Bi-LSTM modelinin yaklaşık olarak 3 kat daha hızlı olduğu görülmektedir.

Çizelge 6. Bi-LSTM modelinin çalışma performansı.

Bi-LSTM	**Loss**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Train	0.0010	0.9997	0.9997	0.9997	0.9996
Test	0.0036	0.9992	0.9992	0.9992	0.9992

Çizelge 7. BERT modelinin çalışma performansı.

BERT	**Loss**	**Accuracy**	**F1-Score**	**Precision**	**Recall**
Train	0.0026	0.988	0.9884	0.9876	0.9893
Test	0.0316	0.952	0.9336	0.9223	0.9453

Sonuç olarak, bu çalışmada Türkçe tweetler kullanılarak hakaret veya küfür içeren kelime ve kelime öbeklerinin tespiti problemi ele alınmıştır. Bu probleme, NER problemi olarak yaklaşılmıştır ve çözümü için derin öğrenme tabanlı Bi-LSTM ve BERT modelleri kullanılmıştır. Modellerin karşılaştırmalı sonuçları analiz edilmiştir. Bi-LSTM modeli hem eğitimde hem de test aşamasında %99'a yakın doğruluk oranı sergilemiştir. BERT modeli ise eğitim aşamasında %99 civarında başarı gösterirken, test başarısının %95 civarında olduğu gözlemlenmiştir. Çalışma hızı açısından, Bi-LSTM modelinin BERT

modelinden yaklaşık olarak 3 kat daha hızlı olduğu görülmüştür.

Çalışmada kullanılan veri seti, genel olarak sosyal medya mesajlarından oluşmaktadır. Yapılan çalışma, incelenen literatüre göre, Türkçe'de makine öğrenmesi temelli yöntemlerle yapılan ilk kelime bazlı küfür tespiti çalışmasıdır. Genel itibariyle çalışmalar siber zorbalık tespiti üzerine ve cümle bazlı çalışmalardır. Çelik and Yıldırım, (2020) çalışmasında olduğu gibi, literatürdeki çalışmalar cümle içerisinde küfürlü bir sözün geçip geçmediğini tespit etme üzerine kuruludur.

Yapılan bu çalışmada, Türkçe'de kelime bazlı bir küfür tespit yöntemi derin öğrenme yöntemleri ile birlikte uygulanmıştır ve başarılı sonuçlar elde edilmiştir. Bu sayede sosyal medya ve benzeri platformlarda zaman geçiren kişilerin direkt olarak ofansif sözlere maruz kalmadan içeriklere ulaşabilmesi mümkündür. Ayrıca listeye dayalı yöntemlerde olduğunun aksine, liste dışı olan veya kısaltılmış söz öbekleri de başarılı şekilde yakalanmıştır.

Bu çalışma farklı uygulama alanları, yani dergi, kitap veya gazete gibi diğer basın kuruluşlarından veri alarak daha genel metinlerde istenmeyen kelimeleri sansürleme amaçlı geliştirilebilir. Ayrıca, ileri çalışmalar olarak, kullanılan metnin yanında diğer ek özelliklerinin de dikkate alınarak daha yüksek performanslı algoritmaların geliştirilmesi düşünülebilir.

6. Kaynaklar

Bowden, K. K., Wu, J., Oraby, S., Misra, A., and Walker, M., 2018. SlugNERDS: A named entity recognition tool for open domain dialogue systems. *arXiv preprint arXiv:1805.03784*.

Çelik, A. and Yıldırım, B., 2020. Turkish profanity detection enhanced by artificial intelligence. *28th Signal Processing and Communications Applications Conference (SIU)*, 1-4. IEEE.

Deepak, G., Teja, V., and Santhanavijayan, A., 2020. A novel firefly driven scheme for resume parsing and matching based on entity linking paradigm. *Journal of Discrete Mathematical Sciences and Cryptography*, **23(1),** 157-165.

Devlin, J., Chang, M. W., Lee, K., and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Grishman, R., 1995. *The NYU System for MUC-6 or Where's the Syntax?*, New York Unıv, Ny, Dept. Of Computer Scıence.

Guo, J., Xu, G., Cheng, X., and Li, H., 2009. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval,* 267-274.

Güneş, A., and Tantuğ, A. C., 2018. Turkish named entity recognition with deep learning. In *2018 26th Signal Processing and Communications Applications Conference (SIU),* 1-4. IEEE.

Han, J., Sun, A., Cong, G., Zhao, W. X., Ji, Z., and Phan, M. C., 2017. Linking fine-grained locations in user comments. *IEEE Transactions on Knowledge and Data Engineering*, **30(1),** 59-72.

Hochreiter, S., and Schmidhuber, J., 1997. Long short-term memory. *Neural computation*, **9(8),** 1735-1780.

Krause, S., Li, H., Uszkoreit, H., and Xu, F., 2012. Large-scale learning of relation-extraction rules with distant supervision from the web. In *International Semantic Web Conference,* 263-278, Springer, Berlin, Heidelberg.

Krupka, G., 1995. SRA: Description of the SRA system as used for MUC-6. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland,* November 6-8, 1995.

Laboreiro, G., and Oliveira, E., 2014. What we can learn from looking at profanity. In *International Conference on Computational Processing of the Portuguese Language,* 108-113, Springer, Cham.

Lafferty J., McCallum A., Pereira F., *et a*l., 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning*, ICML, **1**, 282–289.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D., 1990. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems,* 396-404.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86(11),** 2278-2324.

Lee, H. S., Lee, H. R., Park, J. U., and Han, Y. S., 2018. An abusive text detection system based on enhanced abusive and non-abusive word lists. *Decision Support Systems*, **113**, 22-31.

Luo, F., Xiao, H., and Chang, W., 2011. Product named entity recognition using conditional random fields. In *2011 Fourth international conference on business intelligence and financial engineering,* 86-89. IEEE.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D., 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 55-60.

Meng, X., Wei, F., Liu, X., Zhou, M., Li, S., and Wang, H., 2012. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining,* 379-387.

Mikolov T., Chen K., Corrado G., and Dean J., 2013. Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781.*

Mikolov T., Sutskever I., Chen K., Corrado G.S., and Dean J., 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Nasiboglu R. and Gencer M., 2021. Comparison of Spacy And Stanford Libraries' Pre-Traıned Deep Learnıng Models for Named Entıty Recognıtıon, *Journal of Modern Technology and Engineering,* **6(2),** 104-111.

Özkaya, S., and Diri, B. 2011. Named entity recognition by conditional random fields from Turkish informal texts. In *2011 IEEE 19th Signal Processing and Communications Applications Conference (SIU),* 662-665). IEEE.

Pawar, S., Srivastava, R., and Palshikar, G. K., 2012. Automatic gazette creation for named entity recognition and application to resume processing. In *Proceedings of the 5th ACM COMPUTE Conference: Intelligent & scalable system technologies,* 1-7.

Pradhan S., Moschitti A., Xue N., Ng H.T., Björkelund A., Uryupina O., Zhang Y., and Zhong Z., 2013. Towards robust linguistic analysis using ontonotes. In *CoNLL*, 143–152.

Ramshaw, L., and Marcus, M. P., 1995. Text Chunking Using Transformation-Based Learn. *ACL Third Workshop on Very Large Corpora*, June 1995, 82-94.

Ratadiya, P., and Mishra, D., 2019. An attention ensemble based approach for multilabel profanity detection. In *2019 International Conference on Data Mining Workshops (ICDMW),* 544-550. IEEE.

Rau L.F., 1991. Extracting company names from text*,* In *Proceedings of Seventh IEEE Conference on Artificial Intelligence Applications*, **1,** 29-32: IEEE.

Sang E. F. and Veenstra J., 1999. Representing text chunks, In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, 173-179. Association for Computational Linguistics.

Sarı, Ö. C., ve Aktaş, Ö., 2018. Türkçe Ders Metinleri İçin Özelleştirilmiş Bir Varlık İsmi Tanıma Yapısı. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, **11(2)**, 52-68.

Schiersch, M., Mironova, V., Schmitt, M., Thomas, P., Gabryszak, A., and Hennig, L., 2020. A german corpus for fine-grained named entity recognition and relation extraction of traffic and industry events. arXiv preprint arXiv:2004.03283.

Schmid, H., 1999. Improvements in part-of-speech tagging with an application to German. In *Natural language processing using very large corpora,* 13-25. Springer, Dordrecht.

Shen, Y., Yun, H., Lipton, Z. C., Kronrod, Y., and Anandkumar, A., 2017. Deep active learning for

named entity recognition. arXiv preprint arXiv:1707.05928.

Sienčnik, S. K., 2015. Adapting word2vec to named entity recognition. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, 239-243.

Sood, S. O., Antin, J., and Churchill, E., 2012. Using crowdsourcing to improve profanity detection. In *2012 AAAI Spring Symposium Series*, 69-74.

Sood, S., Antin, J., and Churchill, E., 2012. Profanity use in online communities. In *Proceedings of the SIGCHI conference on human factors in computing systems,* 1481-1490.

Subramaniam, L. V., Faruquie, T. A., Ikbal, S., Godbole, S., and Mohania, M. K., 2009. Business intelligence from voice of customer. In *2009 IEEE 25th International Conference on Data Engineering,* 1391-1402). IEEE.

Su, H. P., Huang, Z. J., Chang, H. T., and Lin, C. J., 2017. Rephrasing profanity in chinese text. In *Proceedings of the First Workshop on Abusive Language Online,* 18-24.

Teh, P.L., and Cheng, C.B., 2020. Profanity and hate speech detection. *International Journal of Information and Management Sciences*, **31(3),** 227-246.

Sang E.F. and Meulder F., 2003. Introduction to the conll-2003 shared task: Language independent named entity recognition. In *Proceedings of CoNLL-2003, Edmonton, Canada,* **4,** 142–145. Association for Computational Linguistics.

Wallach, H.M., 2004. Conditional Random Fields: An Introduction, *University of Pennsylvania CIS Technical Report*.

Yılmaz, Ş.Ş., Özer, İ., ve Gökçen, H., 2022. Twitter Platformundan Elde Edilen Türkçe Saldırgan Dil Derlemi. *Mühendislik Bilimleri ve Araştırmaları Dergisi*, **4(2),** 304-316.

Yi, M., Lim, M., Ko, H., and Shin, J., 2021. Method of profanity detection using word embedding and LSTM. *Mobile Information Systems*, Article ID: 6654029, https://doi.org/10.1155/2021/6654029

İnternet kaynakları

1-https://github.com/ezgisubasi/turkish-tweets-sentiment-analysis/tree/main/data, (10.05.2022)

2-https://github.com/d35k/Turkish-Swear-Words/blob/master/swears.txt, (10.05.2022)