Euroasia Journal of Mathematics, Engineering, Natural & Medical Sciences
International Indexed and Refereed
ISSN 2667-6702

Türkçe Metinler İçin Pos Etiket Bilgisi İle Cümle Sonu Belirlenmesinde Derin Öğrenme Yöntemlerinin Başarısı

Success of Deep Learning Methods in Determining the End of Sentence with Pos Tag Information in Turkish Texts

Yasin Bektaş[1*], Selma Ayşe Özel[2]

[1] Öğr. Gör. Dr., Mersin Üniversitesi, Erdemli Meslek Yüksekokulu, Bilgisayar Teknolojileri Bölümü, Mersin, Türkiye
[2] Prof. Dr., Çukurova Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Adana, Türkiye

** Corresponding author: yasinbektas@mersin.edu.tr*

Geliş Tarihi / Received: 12.06.2023
Kabul Tarihi / Accepted: 11.07.2023
Araştırma Makalesi/Research Article
DOI: 10.5281/zenodo.8233545

ÖZET

Günümüzdeki teknolojik gelişmelerin bir sonucu olarak dijital dünyada yazılı ve sözlü metinler hızla artmıştır. Bununla birlikte Doğal Dil İşleme (DDİ) uygulamaları günümüzde büyük önem kazanmıştır. DDİ uygulamalarında çözülmesi gereken ilk ve en önemli konu metindeki cümle sonlarının doğru bir şekilde belirlenmesidir. Çoğunlukla cümle sonunda bulunan nokta, ünlem, soru işareti gibi noktalama işaretleri sadece cümle sonunu belirlemede kullanılmazlar. Bu yüzden noktalama işaretlerinin kullanım amacının belirginleştirilmesi de bir problem olarak karşımıza çıkmaktadır. Daha önce yapılan çalışmalarda POS (Part-Of-Speech) etiket bilgilerinin cümle sonuna etkileri incelenmiş ve klasik sınıflandırıcılar ile başarılı sonuçlar elde edilmiştir. Bu çalışmada ise kural tabanlı oluşturulmuş olan 9 adet niteliğe farklı sayılarda POS etiket bilgileri eklenmiş ve Uzun Kısa Süreli Bellek (Long Short Term Memory- LSTM) ve Çift Yönlü Uzun Kısa Süreli Bellek (Bidirectional Long Short Term Memory- BiLSTM) olarak isimlendirilen derin öğrenme yöntemleri ile deneyler gerçekleştirilmiştir. Deneylerde Türkçe Ulusal Derlemi (TUD) ve SETimes isimli paralel derlem kullanılmıştır. TUD 1990-2009 dönemini kapsayan, çok fazla alan ve türden oluşmuş 50 milyon kelimelik bir derlemdir. SETimes ise 9'u güneydoğu Avrupa'ya biri ise İngilizceye ait olmak üzere toplam 10 dilden oluşan bir paralel derlemdir. Belirtilen derlemlerden gelişigüzel seçilen cümle sonu olan ve olmayan 30000 örnekli dengeli alt veri setleri oluşturulmuş ve deneylerde kullanılmıştır. Yapılan deneyler ile Geri Beslemeli Sinir Ağı (Back Propagation Neural Network), RBF (Radial Basis Function) Ağı, Naive Bayes sınıflayıcısı, Karar Ağacı ve Destek Vektör Makinesi (Support Vector Machine) gibi klasik sınıflandırıcılar ile LSTM ve BiLSTM gibi derin öğrenme yöntemleri kıyaslanmıştır. Bunun sonucunda derin öğrenme yöntemlerinin başarısının belirgin bir şekilde klasik sınıflandırıcılardan iyi olduğu gözlenmiştir.

Anahtar Kelimeler: Derin Öğrenme, Doğal Dil İşleme, Cümle Sınırı Tespiti, Derlem

ABSTRACT

As a result of today's technological developments, written and spoken texts have increased rapidly in the digital world. However, Natural Language Processing (NLP) applications have gained great importance today. The first and most important issue to be solved in NLP applications is to determine the sentence boundary of the text correctly. Punctuation marks such as periods, exclamation points, and question marks that are generally seen at the end of sentences are not only used to determine the boundary of sentences in the text. Therefore, the disambiguation of the purpose of using punctuation

marks is a problem. In previous studies, the effects of POS (Part-Of-Speech) tag information at the end of the sentence were examined and successful results were obtained with classical classifiers. In this study, different numbers of POS tag information were added to 9 rules-based attributes and experimental evaluations were carried out with deep learning methods called Long Short Term Memory(LSTM) and Bidirectional Long Short Term Memory(BiLSTM). For the experiments, the Turkish National Corpus (TNC) and the parallel corpus called SETimes were used. TNC is a 50 million-word corpus of many fields and genres covering the period 1990-2009. SETimes is a parallel corpus of 10 languages, 9 of which belong to southeast Europe and one to English. Balanced sub-datasets with 30000 samples, with and without sentence endings, randomly selected from the specified corpus, were created and these datasets were used for testing. With the experiments performed, classical classifiers such as Back Propagation Neural Network, RBF (Radial Basis Function) Network, Naive Bayes classifier, Decision Tree and Support Vector Machine; and deep learning methods such as LSTM and BiLSTM were compared. As a result, it has been observed that the success of deep learning methods is significantly better than classical classifiers.

Keywords: Deep Learning, Natural Language Processing, Sentence Boundary Detection, Corpus

1. GİRİŞ

Doğal Dil İşleme (DDİ), temel olarak Yapay Zeka ve Dilbilim'in bir alt alanı olup, insanların günlük yaşamda kullandığı doğal dillerin bilgisayarlar aracılığı ile işlenmesini ve kullanılmasını sağlar. Bilgisayarlar ve insanlar arasındaki doğal dil aracılığı ile etkileşim 1950'lerde başlamıştır (Grissman, 1986). Daha sonra DDİ çalışmalarının kullanımı belge çevirisi, soru-cevaplama veya metin özetleme gibi birçok araştırma alanında hızla artmıştır. Tüm bu süreçlerde karşılaşılan en büyük sorun, doğal dillerin karmaşıklığıdır. Hızlı teknolojik gelişmelerle birlikte, bu karmaşıklık bilgisayarlar aracılığı ile çözülmeye çalışılmıştır (Eric, 1997). DDİ çalışmalarında çözülmesi gereken en temel sorunlardan biri, metni oluşturulan cümle sınırlarının doğru bir şekilde belirlenmesidir.

Cümle sınırlarının belirlenmesi (CSB), DDİ çalışmalarındaki en önemli ön işleme adımıdır. Bir kelimenin birden fazla anlamı arasından doğru anlamın bulunabilmesi, iyi bir cümle analiziyle belirlenebilir. Aynı zamanda iyi bir çeviri sistemi veya soru-cevap sistemi, temel birim olarak cümleyi ele alır (Kiss ve Struck, 2006). Tüm DDİ çalışmalarında, CSB aşaması başarıyı en önemli derecede etkiler. Yazılı metinlerde cümlelerin sonunda kullanılan karakterler, nokta (.), üç nokta (...) işareti, soru işareti (?), ünlem işareti (!) ve iki nokta üst üste (:) gibi noktalama işaretleridir (TDK, 2023). Ancak, bu işaretler sadece cümlenin sonunu belirtmez. Ayrıca, metinde farklı amaçlarla da bulunabilirler. CSB sürecinin temelinde, bu karakterlerin kullanım amacının netleştirilmesi gerekmektedir.

CSB süreçlerini iki grupta incelemek mümkündür: Bunlardan ilki, kural tabanlı yaklaşımdır. Bu yaklaşımda, özellikle kısaltmaların tespiti için el ile oluşturulan kural listeleri kullanılır. Ayrıca, kullanılan kuralların anlaşılması zor olabilir ve bu kurallar çalışmalarda kullanılan metinlere sınırlıdır (Dinçer ve Karaoğlan, 2004). Diğer yaklaşım grubu ise makine öğrenimi yöntemlerini içerir. Bu yaklaşımda, sistemin öğrenmesi için kapsamlı bir derlem gereklidir. Makine öğrenimi yöntemlerinde dünya genelinde yaygın olarak kullanılan İngiliz İngilizcesi ve Amerikan İngilizcesi üzerinde hazırlanan derlemelerle karşılaşırız. Ayrıca, Türk Dilbilimi ve DDİ çalışmalarında kullanılmak üzere Türkçe için oluşturulmuş sınırlı sayıda derlemler de mevcuttur.

Farklı doğal diller için çalışabilen genel amaçlı CSB çalışmalarında, çeşitli İngilizce derlemlerle eğitilen sistemlerin, f-ölçeği değerlerinin %97 ile %98,80 arasında olduğu bildirilmiştir. Genel amaçlı CSB sistemlerinin Türkçe derlemler üzerinde uygulandığı çalışmalarda ise bu oranın %91 ile %99,80 arasında olduğu görülmüştür. Genel amaçlı sistemlerin Türkçe için yüksek f-ölçeği değerlerine sahip olmasının nedenleri incelendiğinde, kullanılan Türkçe derlemlerin çoğunlukla gazete, haber ve

benzeri makalelerden oluştuğu ve benzer cümle yapılarına sahip olduğu tespit edilmiştir (Aksan ve Ark., 2014). Türkçe'nin farklı kullanım kalıplarının olduğu Türkçe Ulusal Derlemi (TUD)'nin bir alt derlemiyle yapılan bir çalışmada, bazı yaygın olarak kullanılan CSB sistemleri test edilmiş, en yüksek doğruluk oranının %88 olduğu tespit edilmiştir. Türkçe'nin farklı kullanımlarına sahip büyük bir derlemle bildirilen test sonuçlarının düşük olması, Türkçe için bir CSB sisteminin gerekliliğini ortaya koymaktadır (Aksan ve Ark., 2014).

Daha önce yapılan bir çalışmada POS (Part-Of-Speech) etiketlerinden elde edilen niteliklerin klasik sınıflandırıcılar ile cümle sonu belirlenmesine etkisi incelenmiştir (Bektaş ve Özel, 2018). POS etiketi, metnin her bir sözcüğüne atanan bir etikettir. Bu etiket, ilgili sözcüğün zaman, durum, sayı (çoğul/tekil) gibi dilbilgisel bilgilerini belirtir. POS etiketleri, derlem aramalarında, metin analiz araçlarında ve algoritmalarında kullanılır. Bektaş ve Özel (2018) tarafından yapılan çalışmada kural tabanlı oluşturulan niteliklere belirli kelime birimlerinin POS etiket bilgileri eklenmiştir. Çok bilinen klasik sınıflandırıcılar ile yapılan testlerde başarılı sonuçların elde edildiği gözlenmiştir.

Bu çalışmada, Türkçe dili için CSB çalışmalarına bir katkı sağmak amaçlanmıştır. POS etiket bilgilerinin yinelemeli sinir ağları ile oluşturulmuş bir derin öğrenme modelinde nasıl bir fayda sağlayacağı incelenmiş ve başarılı sonuçlar elde edilmiştir. Yapılan deneylerde klasik sınıflandırıcılara ek olarak LSTM (Long-Short Term Memory) ve BiLSTM (Bidirectional Long-Short Term Memory) derin ağ modelleri kullanılmış ve iki farklı Türkçe derlemde sonuçlar sunulmuştur.

2. MATERYAL ve METOT

2.1. Veri Kümeleri

Bilgisayar teknolojilerindeki ilerlemeler sayesinde, doğal dillerin farklı kullanımlarını içeren büyük derlemler oluşturulmuştur. Bir dil kaynağı olarak bir derlem, belirli amaçlar temelinde yapılandırılmış metinler/konuşmalardan oluşan bir dizi veridir. Bu derlemler, belirli bir dil veya dil çalışma alanında dil kurallarının istatistiksel analizi ve doğrulaması için kullanılır. Bu koleksiyonlar aracılığıyla dilbilim ve bilgi teknolojileri alanında yapılan çalışmalar sayesinde, diğer yöntemler ve araçlarla görülemeyen, doğal dilin birçok önemli özellikleri ortaya çıkarılmıştır. Günümüzde, birçok dilin özel veya genel amaçlı derlemesi oluşturulmuş ve kullanıcılara sunulmuştur (Lee, 2010). Çalışmada TUD ve SETimes derlemleri olmak üzere iki adet Türkçe derlem kullanılmıştır.

2.1.1. Türkçe Ulusal Derlemi (TUD)

Çalışmada kullanılan ilk veri kümesi, aynı dağıtım kriterlerine sahip Türkçe Ulusal Derlemi (TUD)'nin alt derlemidir (Aksan, 2012). TUD, çağdaş Türkçe'nin büyük ölçekli genel bir derlemesini oluşturmayı hedefleyen bir proje kapsamında geliştirilmiştir. 50 milyon kelime içermekte olup, 20 yıllık bir dönemi (1990-2009) kapsar. TUD'un büyük bir kısmı (%98) metin örneklerinden oluşurken, geri kalan kısmı konuşma verilerinin yazılı bir sürümüdür. TUD alt derlemi, TUD ile aynı dağılım kriterlerine göre örneklenmiş, yaklaşık 10 milyon kelime içeren bir derlemdir ve alt derlemde cümle bitimleri bir araştırmacı tarafından el ile etiketlenmiştir (Demirhan, 2013).

2.1.2. SETimes Derlemi

SETimes derlemi, www.setimes.com haber sitesinde yayınlanan içeriğe dayanan, haber metinlerinden oluşan bir paralel derlemdir ve içinde Türkçe bir derlem de içermektedir (Tyers ve Alperen, 2010). Türkçe ve İngilizce metinlerle birlikte, 9 güneydoğu Avrupa dilinde metinler içeren bir derlemedir.

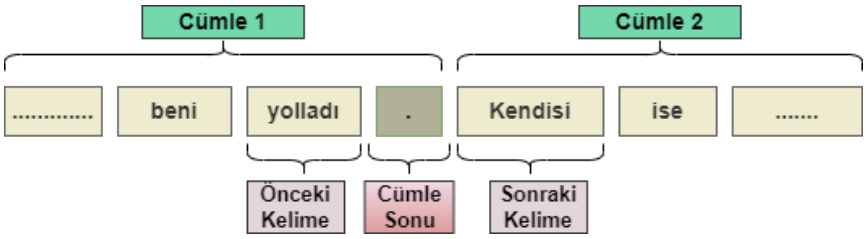
2.1.3. Çalışmada Kullanılan Veri Kümeleri

Bu çalışmada, TUD ve SETimes derlemlerinden elde edilen 2 adet CSB veri kümesi kullanılmıştır. İlk CSB veri kümesi için, TUD derleminden gelişigüzel 30000 adet örnek seçilmiştir. Bu örneklerin 15000 tanesi cümle sonu örneği olan pozitif örneklerdir. 15000 tanesi ise cümle sonu karakterinin başka amaçlar için kullanıldığı negatif örneklerdir. Eğitim ve test işlemleri için belirlenen örnek dağılımı Tablo 1'de gösterilmiştir. Eğitim ve test işlemlerinde veri kümesi içinde pozitif ve negatif sınıfların dağılım dengesi gözetilmiştir. Aynı işlemler SETimes derlemi için de tekrarlanmış, toplam 30000 örnekli bir veri kümesi de SETimes derleminden Çizelge 1' deki örnek dağılımlarını sağlayacak şekilde elde edilmiştir.

Çizelge 1. Veri kümesi örnek dağılımları

	Pozitif	***Negatif***	***Toplam***
Eğitim	10.500	10.500	**21.000**
Test	4.500	4.500	**9.000**
Toplam	**15.000**	**15.000**	**30.000**

Veri seti oluşturulurken, öncelikle Whong ve Chao tarafından hazırlanan çalışmadan esinlenerek belirlenmiş 9 adet kural tabanlı nitelik kullanılmıştır (Whong ve Chao, 2010). Bu niteliklere ek olarak Şekil 1'de gösterildiği gibi potansiyel cümle sonu karakterinin öncesindeki ve sonrasındaki birime ait iki adet POS etiket bilgisi de nitelik kümesine eklenmiştir.



Şekil 1. Cümlede kullanılan POS etiket yerleşimi

POS etiket bilgisinin de eklemesi sonucunda toplam 11 adet nitelik ile çalışılmıştır. Bu niteliklerin listesi Çizelge 2'de verilmiştir. Klasik sınıflandırıcılar ile yapılan deneylerde potansiyel cümle sonu karakterleri için Çizelge 2'de yer alan niteliklerden oluşan veri kümesi oluşturulup, kullanılmıştır. YSA ile yapılan deneylerde ise ileride bölüm 2.2.6'da ifade edildiği gibi cümle sonu karakterinin öncesindeki ve ardındaki birimleri de kapsayan ardışık yapının her biri için Çizelge 2'deki bilgilere sahip vektörlerden oluşan veri seti elde edilmiş ve kullanılmıştır.

Çizelge 1. Temel nitelik listesi

Nitelik No	***Nitelik Açıklaması***
1	Önceki kelimenin ilk harfi büyük harfle mi başlıyor?
2	Sonraki kelimenin ilk harfi büyük harfle mi başlıyor?
3	Önceki kelime tamamen büyük harflerden oluşuyor mu?
4	Sonraki kelime tamamen büyük harflerden oluşuyor mu?
5	Sonraki kelime rakamlardan oluşuyor mu?
6	Önceki kelimenin uzunluğu 5 karakterden daha kısa mı?

Nitelik No	*Nitelik Açıklaması*
7	Önceki kelimenin uzunluğu kaç karakterdir?
8	Sonraki kelimenin uzunluğu kaç karakterdir?
9	Öncesindeki çift tırnak sayısının mod değeri nedir?
10	Önceki birim POS etiket bilgisi
11	Sonraki birim POS etiket bilgisi

2.2. Kullanılan Yöntemler

Daha önce POS etiket bilgilerinin cümle sonuna etkisinin incelendiği Bektaş ve Özel'e ait çalışmada klasik sınıflandırıcılar kullanılmıştır (Bektaş ve Özel, 2018). Bu sınıflandırıcılar; Geri Beslemeli Sinir Ağı (Back Propagation Neural Network), RBF (Radial Basis Function) Ağı, Naive Bayes sınıflayıcısı, Karar Ağacı ve Destek Vektör Makinesi (Support Vector Machine)' dir. Derin öğrenme yöntemlerinin başarısını ölçmek adına alınan sonuçlar bu sınıflandırıcılar ile karşılaştırılmıştır. İlgili çalışmadaki sonuçlar TUD derlemi ile yapılmış deneylerin sonuçlardır. Buna ek olarak, bu çalışmada SETimes derlemi ile de aynı yöntemler kullanılmış ve sonuçlarından 3. bölümde bahsedilmiştir. Varsayılan değerler ile kullanılan bu klasik sınıflayıcı yöntemleriyle, cümlelerin ardışık dizilimlerine uygun olan yinelemeli sinir ağları karşılaştırılmıştır.

2.2.1. Geri Beslemeli Sinir Ağı (GBSA)

Geri beslemeli sinir ağı (GBSA) yöntemi, çok katmanlı bir ileri beslemeli sinir ağıdır ve oluşan hata geri doğru besleme algoritmasına göre eğitilir. GBSA, Rumelhart (1985) tarafından tanıtılmış olup en yaygın kullanılan sinir ağı modellerinden biridir.

İleri besleme aşamasında, giriş verisi alınır ve ilgili çıktı üretilir. Bu süreçte bilgi ağ boyunca ileriye doğru akar. Giriş verisi, her katmandaki gizli birimlere iletilir ve sonunda çıktı üretilir. Bu süreç ileri yayılma (forward propagation) olarak adlandırılır.

Eğitim sırasında, ileri yayılma, tek bir skaler maliyet üretene kadar devam edebilir. Geri besleme algoritması, ileri yayılmadan meydana gelen maliyetten gelen bilginin ağ boyunca geri akmasına izin verir ve iki durum arasındaki farkın hesaplanmasını sağlar (Rumelhart, 1986).

2.2.2. Radyal Tabanlı Foksiyon (RTF) Ağı

Radyal tabanlı fonksiyon (RTF) ağı, radyal tabanlı fonksiyonu bir aktivasyon fonksiyonu olarak kullanan yapay sinir ağıdır. RTF ağı, giriş katmanı, gizli katman ve çıkış katmanı olmak üzere üç ana katmandan oluşur ve besleme tabanlı yapay sinir ağı yapısıyla benzerlik gösterir (Broomhead et al., 1988).

Giriş katmanından gizli katmana olan dönüşüm, RTF aktivasyon fonksiyonuyla gerçekleştirilen doğrusal olmayan bir sabit dönüşümdür. RTF fonksiyonu, girişle bir dizi merkez vektörü arasındaki benzerliği hesaplar. Bu merkez vektörleri, giriş uzayındaki prototip noktaları veya küme merkezlerini temsil eder. Gizli katmandan çıkış katmanına doğrusal bir dönüşüm gerçekleşir. RTF ağında uyum sağlanabilen serbest parametreler merkez vektörleri, merkezi fonksiyonların genişliği ve çıkış katmanı ağırlıklarıdır.

2.2.3. Naive Bayes Sınıflayıcı

Bayes sınıflayıcılar, istatistiksel yöntemle birlikte denetimli öğrenme yöntemini sınıflandırma için sunar. Bu yöntem, özellik vektörü tarafından tanımlanan belirli bir örneği en olası sınıfa atar (Hand ve Keming, 2001). Naive Bayes sınıflayıcısı Bayes teoreminin bağımsızlık önermesiyle basitleştirilmiş bir versiyonudur. , Tüm özelliklerin sınıf etiketlerinden bağımsız olduğu varsayımını

yapar. Bu gerçekçi olmayan varsayıma rağmen, Naive Bayes sınıflandırıcısı, pratikte olağanüstü başarılı olup genellikle çok daha karmaşık yöntemlerle rekabet eder (Hilden, 1984). Naive Bayes yönteminin, metin sınıflandırma, tıbbi teşhisler ve sistem performans yönetimi gibi birçok pratik uygulamada etkili olduğu bilinmektedir.

2.2.4. Karar Ağacı

Veri Madenciliği ile sınıflandırmada en çok kullanılan yöntemlerden birisi de karar ağaçlarıdır. Karar ağacının yapısında her düğüm bir özelliği temsil eder. Dallar ve yapraklar ağaç yapısının unsurlarıdır (Quinlan, 1986). En üstteki öğeye kök, en alttaki öğeye yaprak ve aralarındaki öğelere ara düğümler denir. Karar ağacında kurallar, IF-THEN kuralları biçiminde kökten yaprağa doğru yazılır. Karar ağacı yönteminde bir olayı sonlandırırken probleme verilen cevaba göre hareket edilir.

Karar ağacı algoritmalarının başarılı bir şekilde birçok farklı alanda uygulamaları vardır. Testlerimizde kullandığımız karar ağacı uygulaması C4.5 algoritmasıdır. C4.5 algoritması, en tanınmış karar ağacı algoritmalarından biridir (Quinlan, 1993). C4.5 algoritmasında test özelliği seçim kriteri olarak bilgi kazancı oranı kullanılır ve her küme için en yüksek bilgi kazancı oranına sahip özellik seçilir. C4.5 algoritması, ID3 algoritmasına dayanan bir yöntem olup bu algoritmanın bazı sınırlamalarını ortadan kaldırır (Quinlan, 1993). C4.5 algoritması hem sürekli hem de ayrık özelliklerle çalışabilir. Ayrıca, eksik özellik değerlerini içeren eğitim veri kümeleriyle de çalışabilir. Bununla birlikte, karar ağacı oluşturma sırasında veya sonrasında bazı düğümleri veya alt ağaçları budama yaparak aşırı tahmini önler ve eğitim setindeki istisnai ve gürültülü değerlerin çıkarılmasını sağlar (Niuniu ve Ark., 2010).

2.2.5. Destek Vektör Makineleri(DVM)

İki sınıflı problemlerin çözümü için geliştirilen ve sıklıkla kullanılan en başarılı makine öğrenimi algoritmalarından biri Destek Vektör Makineleri (DVM) (Cortes ve Vapnik, 1995)'dir. DVM, birçok sınıflandırma problemi üzerinde başarılı bir şekilde uygulanmış ve yüksek genelleme performansıyla etkili ve verimli bir makine öğrenimi algoritması olarak literatürde yerini almıştır.

DVM'lerin en önemli avantajı, sınıflandırma problemini ikinci dereceden bir optimizasyon problemine dönüştürüp çözmesidir. Bu sayede, problemi çözme sürecinde işlem sayısı azalır ve diğer algoritmalara kıyasla daha hızlı bir çözüm elde edilir (Nitze ve Ark., 2012). Yöntemin bu özelliği, özellikle büyük hacimli veri kümelerinde büyük bir avantaj sağlar. Ayrıca, optimizasyon temeline dayandığı için, sınıflandırma performansı, hesaplama karmaşıklığı ve kullanışlılık açısından diğer tekniklere göre daha başarılıdır.

DVM'lerin uygulanmasında, sınıflandırma probleminin çözümü için çekirdek fonksiyonu seçimi ve parametre optimizasyonu önemli bir rol oynar. Literatürdeki uygulamalarda genellikle daha iyi sonuçlar verdiği düşünülerek RTF'nun çekirdek fonksiyonu olarak kullanıldığı görülmüştür. Bu çalışmada da varsayılan parametreler ve RTF çekirdeği seçilerek deneyler yapılmıştır.

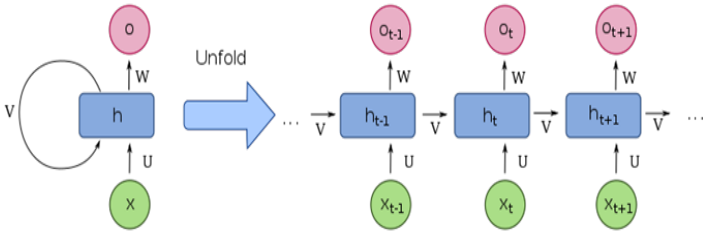
2.2.6. Yinelemeli Sinir Ağları

Yinelemeli Sinir Ağları (YSA), derin öğrenme mimarilerinden biri olarak, Mikolov ve Ark. tarafından 2010 yılında konuşma tanıma problemine uygulanmıştır (Mikolov et al., 2010). Sonuçlar, YSA'nın n-gram tekniklerine göre daha iyi olduğunu göstermektedir. YSA'nın dil modellemesindeki avantajı, önceki durumu mevcut durumun hesaplanmasında kullanmasıdır; bu, çoğu doğal dilin bağlamına benzer. Çünkü YSA'lar ardışık olarak sıralı bilgileri kullanmaktır. Geleneksel sinir ağlarında, tüm girişlerin ve çıkışların birbirinden bağımsız olduğu varsayılır. Ancak birçok işlem için bu iyi bir fikir değildir. Bir cümledeki bir sonraki kelimenin ne olduğunu tahmin etmek için önceki kelimelerin ve hangi sıraya sahip oldukları önemlidir. YSA'ları tekrarlayıcı olarak adlandırılır, çünkü dizi elemanlarının her birinin değerinin hesaplanması önceki hesaplamalara bağlıdır. Tipik bir YSA mimarisi Şekil 2'de gösterilmiştir (Wiki, 2023). Şekil sırasıyla aşağıdaki katmanları içerir;

Giriş : x_t, t zamanında ağa giriş olarak alınır. Örneğin, x_1 bir cümlenin bir kelimesine karşılık gelen tek bir etkin vektör olabilir.

Gizli Durum : h_t, t zamanında bir gizli durumu temsil eder ve ağın "belleği" görevini görür. h_t en son girişe ve önceki zaman adımının gizli durumuna dayanarak hesaplanır: $h_t = f(Ux_t + Wh_{t-1})$ olarak hesaplanır. Burada f fonksiyonu, tanh, ReLU gibi doğrusal olmayan bir dönüşüm olarak alınır.

Çıkış : o_t , ağın çıktısını temsil eder.



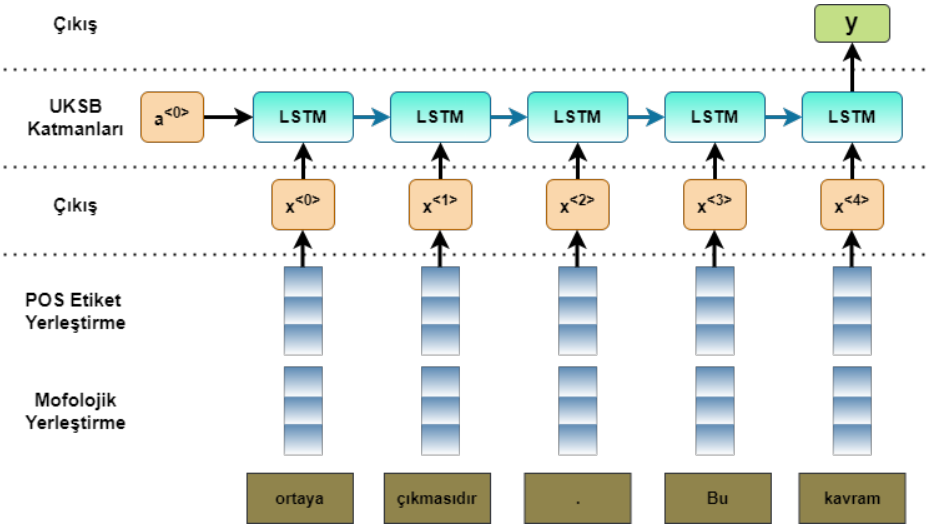
Şekil 2. Yinelemeli Sinir Ağları Mimarisi(Wiki, 2023)

2.2.7 Uzun Kısa Süreli Bellek

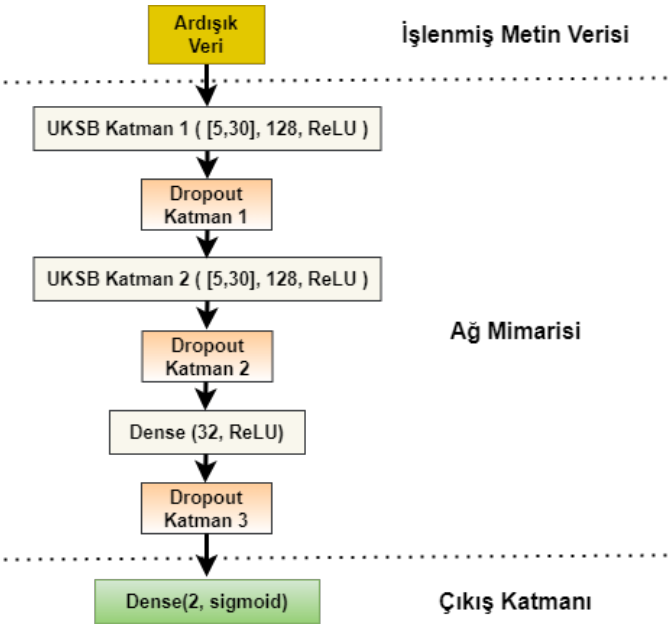
Uzun Kısa Süreli Bellek (UKSB) mimarisi, mevcut YSA'larındaki hata akışının analizi sonucunda ortaya çıkmıştır. Bu analizde uzun zaman gecikmelerinin mevcut mimarilerle ulaşılamaz olduğu bulunmuştur çünkü YSA'larda geriye yayılan hata üssel olarak azalır veya tamamen bozulur (Hochreiter ve Schmidhuber, 1997), (Gers ve Ark., 2002).

Çalışmamızda ardışık olarak sıralı sözcük birimlerinin her birisi için kural tabanlı oluşturulan nitelikler ve POS etiket bilgilerinin oluşturduğu nitelik vektörü birleştirilmiştir. Klasik sınıflandırıcıların verisinden farklı olarak burada cümle sonu karakterinin öncesindeki ve sonrasındaki birimler içinde aynı vektörler oluşturulup tek seferde ağa verilmiştir. Bu vektörler Şekil 3 'de sembolize edildiği gibi UKSB ağına verilmiştir. Burada, cümle sonu belirlemesi ikili bir sınıflandırma gerektirmektedir. Bu nedenle, oluşturulacak ağ, çoktan-bire bir YSA mimarisi ile tasarlanmıştır.

Bu çalışma için özel olarak oluşturulan bu UKSB ağı diyagramı Şekil 4' te gösterilmiştir.



Şekil 3. Çoktan-Bire UKSB Mimarisi



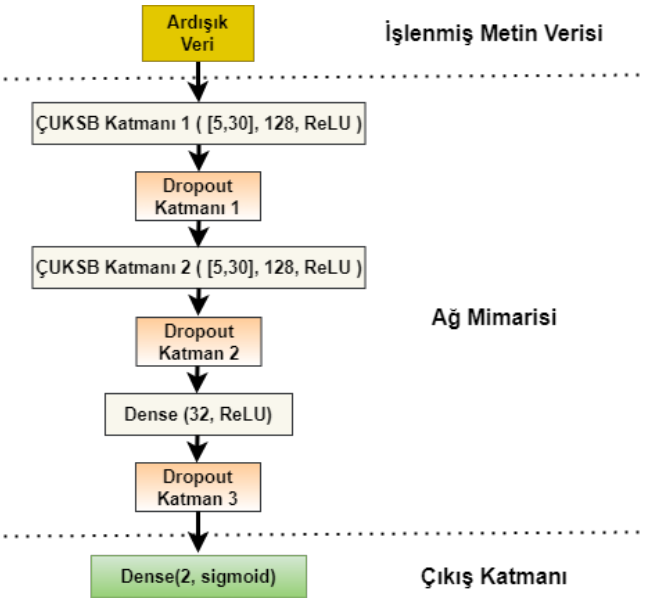
Şekil 4. Kullanılan UKSB ağ modeli

2.2.8 Çift Yönlü Uzun Kısa Süreli Bellek

Çift yönlü UKSB (ÇUKSB), doğal dil işlemede başlıca kullanılan bir tekrarlayan sinir ağıdır. Standart UKSB'den farklı olarak, giriş her iki yönde de akar ve her iki taraftan bilgi kullanabilir. Aynı zamanda, sıralamanın her iki yönde de kelimeler ve ifadeler arasındaki ardışık bağımlılıkları modellemek için güçlü bir araçtır (Schuster ve Paliwal, 1997).

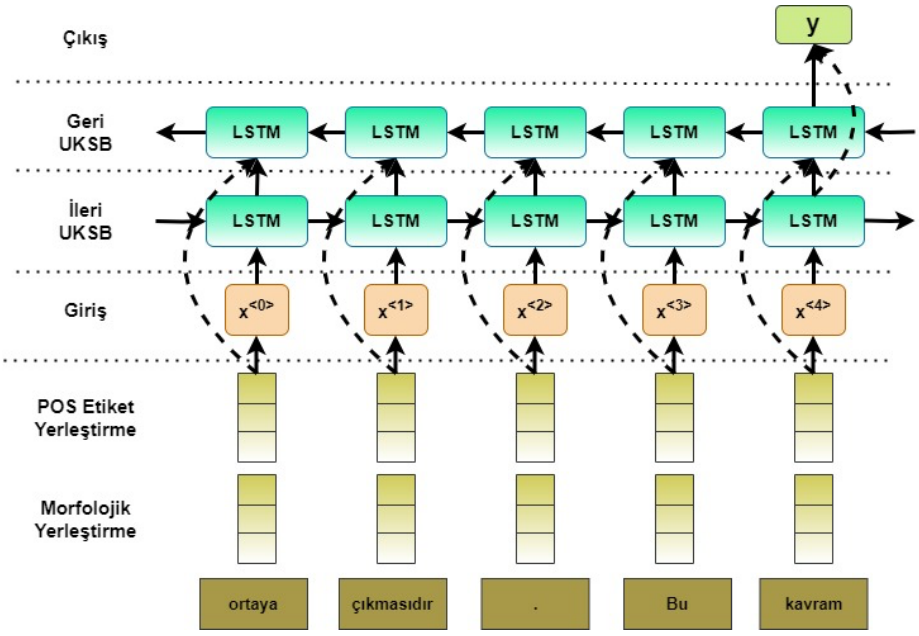
Özetlemek gerekirse, ÇUKSB, bilginin akış yönünü tersine çeviren başka bir UKSB katmanı ekler. Bu nedenle, ek UKSB katmanında giriş sırası ters yönde akar. Daha sonra, her iki UKSB katmanının çıktıları ortalama, toplam, çarpma veya birleştirme gibi çeşitli yöntemlerle birleştirilir.

Çalışmada, cümle dizilerinin iki yönlü anlamlı ilişkilerine dayanarak ÇUKSB ağı uygulanabileceği öngörülmektedir. Bu nedenle, aynı veri setleri ÇUKSB katmanlarıyla birlikte YSA ağında test edilmiştir. İlgili yapı Şekil 5' te gösterilmektedir.



Şekil 5. Kullanılan ÇUKSB ağ modeli

ÇUKSB ağı, önceki bölümde kullanılan UKSB ile aynı yapıdadır. UKSB katmanları iki yönde çalışacak şekilde düzenlenirken, diğer katmanlar tüm parametreleriyle aynı bırakılmıştır. Buna göre hazırlanan ağ Şekil 6' da gösterilmektedir.

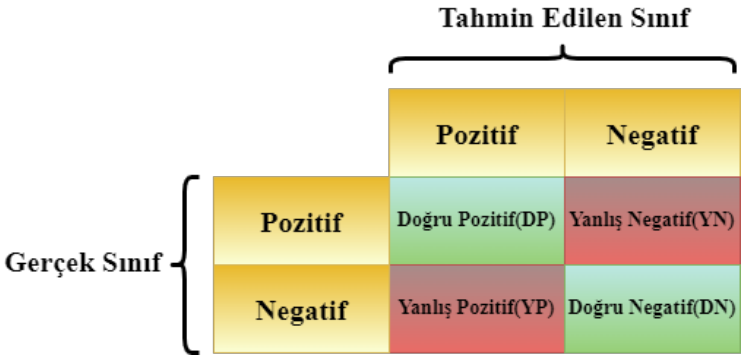


Şekil 6. Çoktan-Bire ÇUKSB Mimarisi

2.3 Değerlendirme Ölçütü

Bu çalışmada, cümle sonunun tespiti için sınıflandırıcıların performansını değerlendirmek için F-ölçeği (F-measure) kullanılmıştır (Han ve Kamber, 2006; Liu, 2011; Mundluru, 2008). Bilgi geri kazanımı alanında sıkça karşılaşılan bu değerlendirme ölçeği, doğru sınıfa atanan örnek sayısı ve yanlış sınıfa atanan örnek sayısıyla ilgilidir. Test sonucunda elde edilen performans değerlendirmesi,

Şekil 7' de gösterilen karmaşıklık matrisi ile ifade edilir. Karmaşıklık matrisinde, satırlar test setindeki her sınıftaki örneklerin gerçek sayısını gösterirken, sütunlar modelin tahminini göstermektedir.



Şekil 7. Karmaşıklık Matrisi

Karmaşıklık matrisine göre bir sınıflayıcının ne ölçüde doğru ve yanlış tahmin ettiği DP, YN, YP ve DN değerlerine göre hesaplanır. Bu hesaplamadan sonra, bu çalışmada sınıflandırıcıların performanslarını değerlendirmek amacıyla kullanılan F-ölçeği eşitlik 1'deki gibi hesaplanır.

$$F - Ölçeği = \frac{2DP}{2DP+YP+YN} \quad (1)$$

3. BULGULAR ve TARTIŞMA

Çalışmada, klasik sınıflandırıcıların yanı sıra yinelemeli sinir ağlarından UKSB ve ÇUKSB ağları ile testler yapılmıştır. Bu testler için, Bölüm 2.1'de detaylarından bahsedilen TUD ve SETimes derlemlerinden 30000 'er örnekli veri kümeleri hazırlanmıştır. Bu veri kümelerindeki her bir örnek için öncelikle 9 adet kural tabanlı nitelik kullanılarak, örnekler vektörlere dönüştürülmüş ve elde edilen veri kümeleri üzerinde bahsedilen tüm sınıflandırıcıların CSB başarısı test edilmiştir. Ardından potansiyel cümle sonu karakterinin öncesindeki ve sonrasındaki POS etiket bilgileri de eklenerek, veri kümesindeki her bir örnek 11 nitelikten oluşan vektörlere dönüştürülmüş, 11 nitelikten oluşan veri kümeleri ile tekrar aynı sınıflandırıcıların CSB başarıları test edilmiştir. Tüm bu testlerin neticesinde ulaşılan sonuçlar Çizelge 3'de sunulmuştur.

Bektaş ve Özel tarafından daha önce yapılan çalışmada TUD derlemi için klasik sınıflandırıcılarda POS etiket bilgilerinin CSB başarısına olumlu etkisinden bahsedilmektedir (Bektaş ve Özel, 2018). Burada aynı testler SETimes derlemi ile de tekrarlanmıştır. Ve yine RBF ağı dışındaki tüm sınıflandırıcılarda CBS başarısında iyileştirme elde edilmiştir. Ayrıca UKSB ve ÇUKSB yinelemeli ağları her iki derlemde de kural tabanlı 9 nitelikli veri setinde iyi sonuçlar vermiştir. POS etiket bilgisi de eklendiğinde UKSB ve ÇUKSB'nin CSB başarıları daha da artmış ve %94,8 F-ölçeği değeri ile ÇUKSB en iyi sonucu vermiştir.

Çizelge 3. Tüm sınıflayıcıların F-ölçeği değerleri

	TUD Alt Derlem		**SETimes Derlemi**	
Sınıflandırıcılar	**9 Nitelik**	**11 Nitelik**	**9 Nitelik**	**11 Nitelik**
Geri beslemeli ağ	84,3	85,8	89,9	90,6
DVM	78,4	80,5	85,4	87,4
RTF ağı	75,4	74,6	80,3	80,1
Naive Bayes	77,0	77,6	82,5	83,2
C4.5	84,7	86,2	90,1	92,7
UKSB	86,1	89,8	92,2	93,1
ÇUKSB	**86,7**	**90,5**	**93,6**	**94,8**

4. SONUÇLAR

Türkçe metinler için dilin özelliklerini önceleyen bir yöntem üretmenin başarıya olan etkisi bu çalışmada elde edilen en önemli çıktıdır. Genel olarak tüm deney sonuçlarına bakıldığında CSB işlemlerinde POS etiket bilgilerinin göz ardı edilemeyeceği ortaya çıkmıştır. POS etiket bilgilerinin eklendiği testlerde, RBF ağı dışındaki tüm yöntemlerde olumlu etki göstermiştir. Ayrıca POS etiket bilgilerinin eklendiği bir ardışık veri yapısı ile yinelemeli sinir ağları gayet tatmin edici bir başarıya ulaşmıştır. SETimes yerine TUD gibi dilin çok çeşitli kullanım şekillerini barındıran derlemle çalışmanın faydalı olacağı genel test sonuçlarına bakılarak anlaşılmıştır. Gazete metinleri gibi daha tek düze derlemlerle yapılan testler başarıyı yüksek göstermesine rağmen, dilin çeşitli kullanımlarının bulunduğu derlemlerde başarı düşmektedir.

Tüm bu sonuçlar, Türkçe metinlerdeki CSB uygulamalarında POS etiketlerinin daha etkili kullanılması gerekliliğini ortaya koymaktadır. Bununla birlikte temel sınıflandırıcıların parametrelerinde iyileştirmeler yapılarak daha iyi sonuçlar elde edilebilir. Ayrıca YSA ile yapılan testlerde, ağ derinlikleri arttırılarak ve parametreleri optimize edilerek daha iyi bir başarı sağlanabileceği ön görülmektedir.

KAYNAKLAR

Aksan, Y., Aksan, M., Koltuksuz, A., Sezer, T., Mersinli, Ü., Demirhan, U. U., ... and Kurtoğlu, Ö., 2012. Construction of the Turkish national corpus (TNC). In Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12), 3223-3227.

Aksan, Y., Özel, S. A., Bektaş, Y., Aksan, M., Demirhan, U. U., Mersinli, Ü., and Yılmazer, H., 2014. Türkçe Tümcelerin Sonunu Belirlemede Açık Kaynak/Ücretsiz Yazılımlar ve Performans Analizleri. Akademik Bilişim, Mersin 727-734.

Bektaş, Y., and Özel, S. A., 2018. The Effect of POS Tag Information on Sentence Boundary Detection in Turkish Texts. In 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), Adana, 1-5.

Broomhead, D. S., and Lowe, D., 1988. Radial basis functions, multi-variable functional interpolation and adaptive networks. Royal Signals and Radar Establishment Malvern (United Kingdom).

Cortes, C., and Vapnik, V., 1995. Support-vector networks. Machine learning, 20(3), 273-297.

Demirhan, U. U., 2013. A description of the verb gel-with special reference to pattern grammar (Master's thesis, Sosyal Bilimler Enstitüsü).

Dinçer, B. T., and Karaoğlan, B., 2004. Sentence boundary detection in Turkish. In International Conference on Advances in Information Systems, Springer, Berlin, Heidelberg. 255-262.

Gers, F. A., Schraudolph, N. N., and Schmidhuber, J., 2002. Learning precise timing with LSTM recurrent networks. Journal of machine learning research, 3(Aug), 115-143.

Grishman, R., 1986. Computational linguistics: an introduction. Cambridge University Press, New York, 193p.

Han, J., and Kamber, M., 2006. Data Mining Concepts and Techniques, 2nd ed., Morgan Kaufmann Publishers, San Francisco, p800.

Hand, D. J., and Yu, K., 2001. Idiot's Bayes—not so stupid after all?. International statistical review, 69(3), 385-398.

Hilden, J., 1984. Statistical diagnosis based on conditional independence does not require it. Computers in biology and medicine, 14(4), 429-435.

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

Kiss, T., and Strunk, J., 2006. Unsupervised multilingual sentence boundary detection. Computational linguistics, 32(4), 485-525.

Lee, D. Y., 2010. What corpora are available. The Routledge handbook of corpus linguistics, Roudledge Press, New York, 650p.

Liu, B., 2011. Web data mining: exploring hyperlinks, contents, and usage data (Vol. 1). Berlin: springer.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., and Khudanpur, S., 2010. Recurrent neural network based language model. In Interspeech, Vol. 2, No. 3, 1045-1048.

Mundluru, D., 2008. Automatically constructing wrappers for effective and efficient Web information extraction. University of Louisiana at Lafayette.

Niuniu, X., and Yuxun, L., 2010. Notice of Retraction: Review of decision trees. In 2010 3rd international conference on computer science and information technology, Vol. 5, 105-109).

Quinlan, J. R., 1986. Induction of decision trees. Machine learning, 1(1), 81-106.

Quinlan, J. R., 1993. C4. 5: Programming for machine learning. Morgan Kauffmann, 38(48), 49.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J., 1986. Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Schuster, M., and Paliwal, K. K., 1997. Bidirectional recurrent neural networks. IEEE transactions on Signal Processing, 45(11), 2673-2681.

TDK, 2023.http://www.tdk.gov.tr/icerik/yazim-kurallari/noktalama-isaretleri-aciklamalar, 01/06/2023

Tyers, F. M., and Alperen, M. S., 2010. South-east european times: A parallel corpus of balkan languages. In Proceedings of the LREC workshop on exploitation of multilingual resources and tools for Central and (South-) Eastern European Languages, 49-53.

Wiki, 2023, https://en.wikipedia.org/wiki/File:Recurrent_neural_network_unfold.svg#filehistory, 01/06/2023

Wong, F., and Chao, S., 2010. iSentenizer: An incremental sentence boundary classifier. In Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering (NLPKE-2010), 1-7