



Derin Öğrenme ile Türkçede Adıl Çözümleme

Pronoun Resolution in Turkish with Deep Learning

Mehmet TAZE

İzmir Ekonomi Üniversitesi

Lisansüstü Eğitim Enstitüsü

İzmir - Türkiye

mehmettaze89@gmail.com

ORCID: 0000-0001-6835-302X

Senem KUMOVA METİN

İzmir Ekonomi Üniversitesi

Mühendislik Fakültesi

İzmir - Türkiye

senem.kumova@ieu.edu.tr

ORCID: 0000-0002-9606-3625

Öz

Dilde, bir sözcüğün/sözcük öbeğinin sürekli tekrar eden kullanımını önlemek için, ilgili öncül sözcüğe/sözcük öbeğine atıfta bulunan adillerin kullanımına sık rastlanır. Bir adılın atıfta bulunduğu öncül ile eşleştirilmesi adıl çözümleme olarak adlandırılır. Bu çalışmada Türkçe metinlerde adillerin çözümlenmesinde derin öğrenme yöntemlerinin başarımı değerlendirilmiştir. Çalışma kapsamında 10 Türkçe çocuk hikayesi kullanılarak bir veri kümesi derlenmiş, deneylerde kullanılmak üzere 12 öznitelik belirlenmiştir. Çok katmanlı algılayıcı, evrimsel (konvüsyonel) ve tekrarlayan sinir ağları nöron ve katman sayılarının değiştiği bir dizi farklı konfigürasyonla uygulanarak F1 ölçüsü ile başarımlar ölçülmüştür. Sonuçlar, Türkçe adıl çözümlemesinde en yüksek başarımları, çok fazla nöron kullanan orta sayıda katmana sahip çok katmanlı algılayıcı sinir ağı tarafından elde edildiğini göstermiştir.

Anahtar sözcükler: Doğal dil işleme, Adıl çözümleme, Evrimsel sinir ağları, Çok katmanlı algılayıcı, Tekrarlayan sinir ağları

Abstract

In language, to avoid repetitive use of a word/phrase, it is common to use pronouns that refer to the corresponding antecedent word/phrase. Matching a pronoun with the antecedent to which it refers is called pronoun resolution. This study evaluates the performance of deep learning methods in pronoun resolution in Turkish texts. Within the scope of the study, a dataset was compiled using 10 Turkish children's stories and 12 features were identified to be used in the

experiments. Multilayer perceptron, convolutional and recurrent neural networks were tested in several different

configurations with varying numbers of neurons and layers and the performance was measured with F1 score. The results show that the highest performance in Turkish pronoun resolution is achieved by a multilayer perceptron neural network with a medium number of layers using too many neurons.

Keywords: Natural language processing, Pronoun resolution, Convolutional neural networks, Multilayer Perceptron, Recurrent neural networks

1. Giriş

Günlük yaşamda dinleyici ve okuyucuların konuşma ve yazıları kolay takip edebilmesi ve akıcılığın sağlanması için aynı sözcük veya sözcük öbeklerinin tekrar tekrar kullanılmasından kaçınılır. Bu sebeple sıklıkla daha önce/sonra bahsedilen varlıklara atıfta bulunan veya onların yerini alan ifadeler kullanılır. Geriye/ileriye atıfta bulunan bu tür ifadeler gönderim ve işaret ettikleri varlıklar ise öncül olarak adlandırılır. Öncül-gönderim çiftlerini tanımlama görevi ise doğal dil işleme alanında önemli bir yere sahip olan gönderim çözümlemesi görevidir.

Alanyazında farklı dillerde gönderim kavramını tartışan ve farklı bakış açılarına odaklanan çeşitli çalışmalar bulunmaktadır. 1995 yılında Grosz gönderimin biçimsel tanımını yapmıştır [1]. Daha önce bir adıl ya da ad öbeği ile bahsedilen bir olay ya da nesneye atıfta bulunmanın gönderim/anafor olduğunu belirtmiştir. [2]'de ise gönderimin daha önce bahsedilmiş bir şeyi göstererek bağdaşıklık sağlama görevi olduğunu söylemiştir. Valin ve LaPolla [3] metnin bağdaşıklığının mekân, zaman ve eylemin ayrınlığı/benzerliği ile şekillenebileceğini belirtmiştir. Bir

metindeki ya da konuşmadaki bağdaşıklık sayesinde insanlar bir metni ya da konuşmayı bütünleşik bir şekilde algırlarlar. Söylemdeki bağdaşıklık sayesinde ise gönderim ögesi ile atıfta bulunulan öğeler belirlenebilir. Bu sayede insanlar daha önce karşılaştıkları bir şey hakkında yeni bilgiler edinebilirler [4]. Örneğin, aşağıdaki cümlelerde gönderim ilişkisi "onlar" ve "çiçekler" kelimeleri arasındadır.

Çiçekler solmuş. Onları sulamak gerek.

Bu örnekte, "onlar" adılı "çiçekler" adına atıfta bulunmaktadır. Basitçe, ilk cümle "çiçekler" ögesinin mevcut durumunu vermekte ve ikinci tümce aynı öğeyle ilgili yeni bir bilgi (gerçekleştirilecek bir eylem) eklemektedir.

Gönderim-öncül ikililerinin tespiti bir metnin veya tümcenin anlaşılmasını geliştirebilir, ancak gönderimlerin belirlenmesi bile çeşitli göstergeler olmadığı zaman zordur [5]. Bazı durumlarda, sıklıkla gönderim olarak gözlenen sözcüklerin (örneğin adılar), farklı amaçlarla kullandığı gözlenebilir.

Gönderim çözümleme, makine çevirisi, otomatik özetleme, metin özetleme, bilgi çıkarımı ve soru yanıtlama sistemleri gibi birçok doğal dil anlama ve işleme uygulamasında pratik olarak kullanılmaktadır [6]. Gönderimsel ilişkilerin çözümlenmesini gerektiren bu geniş görev yelpazesi nedeniyle, bu problemi çözmek için yıllar içinde çeşitli çalışmalar sunulmuştur. Devam eden bu çalışmalar, sözdizimsel, anlamsal ve gerçek dünya bilgilerini yoğun olarak kullanan bilgi yoğun tekniklerden, metinden elde edilen bilgilere dayanan yaklaşımlara kadar çok çeşitli çözümleme yöntemlerini içermektedir [7]. Gönderimsel ilişkilerin sınıflandırılmasında ise ilgili ilişkinin yönü, gönderim ögesinin ses içeriğine dönüşebilme yetkinliği gibi çeşitli unsurlar söz konusu olur. Gönderimsel ilişkilerde yön kavramı gönderim ögesi ve öncülün metin içindeki konumları ile ilişkilidir. Gönderim yönlerinden ilki, önceki örnek tümcelerde olduğu gibi gönderim ögesinin daha önce bahsedilen bir öğeye atıfta bulunması anlamına gelen art gönderimdir. Diğeri ise ögenin daha sonra bahsedilecek bir öğeye atıfta bulunduğu ön gönderimdir. Örneğin aşağıda verilen ilk tümce de yer alan "onun" sözcüğü ikinci tümce deki "Ayşe" adına atıfta bulunan bir ön gönderimdir.

Sabahtan beri onun hakkında konuşuyoruz. Ama Ayşe henüz gelmedi.

Ses içeriği bakımından gönderimler değerlendirildiğinde gönderim ögesinin metin içinde açıkça yer alması veya gizli olarak sunulması şeklinde iki durum mevcuttur. Örneğin aşağıdaki tümcelerde üç gönderim yer almaktadır. İkinci tümcedeki "Onu" sözcüğü ses içeriğine sahiptir ancak Ø1 ve Ø2 sembolü ile gösterilen gönderim öğeleri ses içeriğine sahip değildir. Başka bir deyişle, "Öğretmen" ve "Mehmet" sözcüklerine gönderme yapan iki öğe ikinci tümce de açıkça verilmemiştir. Bu nedenle Ø1 ve Ø2 "gizli gönderim" olarak tanımlanır.

Tümce 1. Öğretmen1, Mehmet2'e bir kalem3 verdi.

Tümce 2. Ø1 Ø2 Onu3 ders sonunda geri vermesini istedi.

Gizli gönderimlerin kullanımı dile göre değişmektedir. Örneğin, Kornfilt [8] Türkçede gönderim öğelerinin (özellikle

adıların) mümkün olduğunda silinmesinin tercih edildiğinden bahsetmiştir. Öte yandan, İngilizcede gönderim öğelerinin sese dönüşmesi yani öğelerin metinde yer alması çok daha yaygındır [9]. Türkçede gizli gönderim öğelerinin sık kullanılması nedeniyle, çözümleme görevinin de İngilizceye kıyasla çok daha zor olduğu belirtilmektedir [10].

Bu çalışmada gönderim ögesi ifadesi aynı veya önceki tümcelerde yer alan bir öğeye atıfta bulunan Türkçe adılar ile sınırlandırılmıştır. Çalışma kapsamına yer alan adıl örnekleri Çizelge-1'de verilmiştir. Gönderim çözümlemeye (bu çalışma özelinde adıl çözümlemeye) 5 açık adıl türü göz önünde tutulmuştur. Deneylerimizde, söz konusu adıların olası tüm dilbilgisel durumları (örneğin, belirtme, yönelme, bulunma, ayrılma, yalın) dikkate alınmıştır.

Çizelge-1: Çalışma kapsamında değerlendirilen adılar

Adıl türü	Örnekler
Kişi adıları	ben, sen, o, biz, siz, onlar
Dönüşlülük adıları	kendim, kendin, kendi, kendimiz, kendiniz, kendileri
İşaret adıları	bu, şu, o, bunlar, şunlar, onlar, bura, şura, ora, buralar, şuralar, oralar, öbürü, ötekisi, berikisi, diğeri
Belgisiz (belirsiz) adılar	kimi, kimileri, kimse, biri, birisi, birileri, başkası, başkaları, bazısı, birçoğu, birkaçı, birazı, herkes, hepsi, hepimiz, hiçbirisi, herhangi biri, her biri, şey, çoğu
İşteş adıları	Birbiri, birbirimiz, birbirinizi, birbirleri

Deneylerimizde, adıl-öncül çözümlemesini ikili sınıflandırma görevi olarak kabul ederek, 593 olumlu ve en çok 17.790 olumsuz örnekten oluşan bir adıl-öncül veri kümesi oluşturulmuştur. Sınıflandırmada 12 öznitelik (örneğin öncülde büyük harf kullanımı, sözcük sayısı, karakter sayısı) kullanılmıştır.

Çalışmada temel amacımız yapay sinir ağlarında nöron ve katman sayısındaki değişimin ilgili sınıflandırma problemindeki etkisinin incelenmesi ve en başarılı kurulumla sahip sinir ağının tespit edilmesidir. Bu doğrultuda öznitelik vektörleri bir dizi farklı konfigürasyon ile kurulan üç farklı derin sinir ağına girdi olarak sunulmuştur. Başarım değerlendirmesi ağırlıklı F1 ölçüsü ile gerçekleştirilmiştir.

Çalışmanın geri kalanı şu şekilde düzenlenmiştir. Bölüm 2'de Türkçedeki adılar hakkında temel bilgiler verilmektedir. Adıl çözümlemesi ile ilgili çalışmalar 3. bölümde verilmiştir. Önerilen yöntem ve deneysel düzenek sırasıyla bölüm 4 ve 5'te verilmiştir. Sonuçlar bölüm 6'da sunulmuş ve makale bölüm 7'de sonlandırılmıştır.

2. Türkçede Adılar

Türkçe, sözcüklerin geleneksel olarak adlar, önadlar, belirteçler, adılar, ilgeç, bağlaçlar, ünlemler ve eylemler olarak sınıflandırıldığı sondan eklemeli bir dildir. Adılar sınıfı, genellikle daha önce bahsedilmiş veya daha sonra bahsedilecek bir ad veya ad öbeğine atıfta bulunmak için kullanılan küçük bir kelime kümesini belirtir. Türkçede adılar, atıf yapılan adların türüne ve/veya adların gönderme şekline

göre sınıflandırılabilir. Bu sınıflar ve kısa tanımları aşağıda verilmiştir.

1. *Kişi adılları*: Bu tür adılar tümce içinde özne olarak bir kişi ya da gruba atıfta bulunmak için kullanılır ve genellikle atlanırlar.
2. *Dönüşlülük adılları*: Türkçede "kendi" dönüşlülük adılı eylemi kimin yaptığını vurgulamak için kullanılır ve sıklıkla kişi adılları yerine kullanılır.
3. *İşaret adılları*: Kişi adları dışındaki tüm nesne/kavram adlarının yerini alan kapalı sözcük sınıfıdır.
4. *Belgisiz adıllar*: Tanımlanmamış kişileri, yerleri, miktarları veya şeyleri gösterirler.
5. *Soru adılları*: İsimleri soru biçiminde temsil eden ve soru olarak ifade eden sözcükler soru adıllarıdır. Soru adıllarına örnek olarak kim (kim), kaç tane (kaç tane) verilebilir.
6. *İyelik adılları*: İyelik adılları -im (örn. kalekim), -in (örn. kalemin) gibi bazı son eklerin kullanımıyla sahiplik gösterirler.
7. *İlgi -ki adılları*: İngilizcedeki "who (kim)", "whom (kimi)" gibi ilgi adıllarından farklı olarak Türkçe ilgi adılları -ki eki kullanılarak sahiplik ifade eder. Örnek olarak benimki (benim), onlarınki (onların) sözcükleri verilebilir.

Bu kategorilere ek olarak, işteş adılar olarak bilinen bir adıl türü daha vardır. İşteş adılara örnekler Çizelge-1'de verilmiştir.

3. Önceki Çalışmalar

Adıl ya da daha genel olarak gönderim çözümlemesindeki temel amaç, belirli bir metin ya da konuşmadaki gönderim ve öncülleri belirlemektir. Önerilen yaklaşımlar genellikle söylemdeki her bir gönderim/adıl için olası öncülleri bulur ve en uygun olanı en yüksek başarımla seçer. Çalışmalar, hedefe ulaşmak için izledikleri yol, çözümleme sürecinde kullanılan yöntem ve öznitelikler bakımından farklılık göstermektedir. Ayrıca deneylerin yapıldığı dil de çözümleme sisteminin yapısını değiştirebilecek önemli bir etkidir. Çözümleme çalışmaları bilgi tabanlı, öğrenme tabanlı ve melez yaklaşımlar olmak üzere üç ana grupta toplanabilir.

Bilgi tabanlı yaklaşımlar biçimsel, söz dizimsel, sözcüksel ve söylem kaynağı bilgilerine dayanır. Bilgi tabanlı yaklaşımların ilk örneklerinden ikisi [11]'de önerilmiştir. Bunlardan ilki, metindeki bir tümcenin ayrıştırma ağaçları üzerinde çalışan basit ve etkili bir yöntemdir. Algoritma, ağaçları soldan sağa doğru dolaşır ve adıl le birlikte doğru bir sayı ve cinsiyet içeren bir ad öbeği arar. Hobbs [11] deneylerinde Arthur Hailey'in bir romanını ve Newsweek dergisinin bir bölümünü kullanmış ve %88,3 başarı oranı elde etmiştir. Hobbs tarafından önerilen ikinci yaklaşım ise ilk yaklaşıma anlam analizinin eklendiği geliştirilmiş bir versiyonudur [11]. Bu yaklaşım ile %91,7 başarı oranı elde edildiği bildirilmektedir.

Lappin ve Leass [12], söz dizimsel yapı ve çarpıcı durum modelinden elde edilen dikkate değer ölçütleri kullanan anafora çözümleme sürecini (RAP) tanıtmıştır. RAP'da, çözümleme işlemi sırasında, bazı aday ad öbeklerini elemek için biçimsel ve söz dizimsel filtreler kullanılır. Daha sonra, kalan adaylara önceden tanımlanmış kriterler aracılığıyla belirli değerler atanır. Bu algoritmada kullanılan kriterler

yakınlık, özne önceliği, varlık önceliği ve ad öbeği önceliğinden oluşmaktadır. Tümceler arası için %72, tümce içi için ise %89 başarı oranı elde edildiği bildirilmiştir [12]. Kenndy ve Boguraev'in [13] çalışmasında RAP metnin derinlemesine bir şekilde sözdizimsel ayrıştırılmasına ihtiyaç duymayacak şekilde değiştirilmiştir. Değiştirilmiş algoritmada, aday öncüller dilbilgisel roller yerine sadece sözdizimsel bilgilere göre sıralanmaktadır. Kenndy ve Boguraev üçüncü şahıs adılları üzerinde %75 başarı oranı elde etmiştir [13].

Öğrenme tabanlı yaklaşımlar, çözümleme için makine öğrenimi tekniklerini kullanır. [14] çalışmasında karar ağacına dayalı denetimli bir öğrenme sistemi önerilmiştir. İlgili sistem sözlük, sözdizimi, anlambilim ve konum gibi 66 özellik ile eğitilmiştir. Sistem tarafından %90 başarı oranı elde edildiği rapor edilmiştir [14]. McCarthy ve Lehnert [15] tarafından sunulan RESOLVE sisteminde ise sekiz anlamsal özellik bir karar ağacına verilmiş ve deneyler MUC-5 metinlerinden elde edilen veri seti üzerinde gerçekleştirilmiştir. Başarı oranları budanmış ağaçlar için %87,6, budanmamış ağaçlar için ise %92,4 olarak sunulmuştur. Soon, Ng ve Lim'in [16] çalışmasında karar ağaçları 12 özellik (sözcüksel, söz dizimsel, anlamsal ve konum özellikleri) ile kullanılmış ve deneyler MUC-6 ve MUC-7 derlemleri üzerinde gerçekleştirilmiştir. Her bir özelliğin çözümleme görevine katkısı da [16]'da ölçülmüş ve sadece 3 özellik kullanılarak %68'lik bir başarı oranı elde edildiği raporlanmıştır.

Öğrenme ve bilgi tabanlı yaklaşımların bir arada kullanılmasıyla ise melez çözümleme yaklaşımları ortaya çıkmıştır. [17] ve [18] melez yaklaşımlara örnek olarak verilebilir. 2002 yılında Mitkov tam otomatik çözümleme sistemi MARS'ı (Mitkov's Anaphora Resolution System) tanıtmıştır [17]. MARS'a girdi olarak ham yazılı metin verilir. Ardından, sistem sözdizimsel ayrıştırma ile verilen metindeki ad öbeklerini belirler. Daha sonra makine öğrenmesi yöntemleri ile gönderimsel (anaforik) öğeler belirlenir. Tüm anaforik öğeler için sistem iki tümce ye kadar aday öncülleri toplar ve kişi, sayı ve cinsiyet entegrasyon filtreleri uygular. Filtrelenen aday öncüller çeşitli faktörler kullanılarak sıralanır ve son olarak gerçek öncül belirlenir. Mitkov, Evans ve Orasan, tam otomatik MARS ile %61,6'lık bir başarı oranı elde edildiğini bildirmiştir [17]. Preiss, algoritmayı makine öğrenmesi yöntemleriyle değiştirmiş ve bu şekilde bellek tabanlı yaklaşımın etkinliğini ölçmüştür. Deneylerde dört adet son teknoloji olasılıksal ayrıştırıcı kullanılmıştır. Düzeltmelerden sonra sistemin asıl kural tabanlı yaklaşımla yaklaşık olarak aynı başarımla gösterdiği bildirilmiştir [18].

Bildiğimiz kadarıyla, Türkçede gönderim/anafor çözümlemesi üzerine yapılan en eski çalışma [19]'da sunulmuştur ve burada Merkezleme Kuramına dayalı bir sistem önerilmiştir. İngilizceden farklı olarak, Türkçede sıralama dilbilgisi kurallarına dayalı sıralama yerine tematik rollere dayalı olmalıdır. [19]'un amacı, Türkçede konuya özel vurgu yaparak ifadeler arasındaki anaforik ilişkileri içeren söylem tutarlılığının bir yönünü tespit etmektir. Bu çalışmada, söylem anafora ilişkin birbirini tamamlayan üç soru bulunmaktadır [19]. İlk soru, sonraki gönderim ile ilişkili olabilecek mevcut ifadelerin incelenmesi üzerinedir. İkincisi, Türkçede, bir dizi potansiyel öncül arasında, gönderim için en belirgin olanı

gösteren faktörlerin belirlenmesidir. Sonucusu ise belirli bir göndergesel ifadenin işlevlerini tespit etmek üzerinedir.

[20]'de, yerel ve yerel olmayan gönderimleri modellemek için bağlama ilişkileri ve merkezleme teorisi için bir dizi kuralı birleştiren bir sistem önerilmiştir. Sistemde, bağlama kuralına uygun olarak, dönüşlü adılların öncülü yerel etki alanında olmalıdır. Önerilen sistem, anaför, adıllar ve gönderme yapan ifadelerin enterpolasyonu ile ilgilenen Bağlama Teorisine [21] ve Merkezleme Kuramına [1][20] dayanmaktadır.

[6] çalışmasında öncülleri tahmin etmek için makine öğrenmesi tabanlı bir yaklaşım sunulmuştur. Önerilen yöntem, topluluk öğrenme yöntemiyle birleştirilmiş bir karar ağacı sınıflandırma tekniğidir ve amaç şahıs adıllarının öncüllerini tespit etmektir. [10]'da Hobbs'un Naive algoritması [11] Türkçe için kullanılmak üzere değiştirilmiştir. Sistem, verilen adılların olası öncüllerini toplamak için söz dizimsel bilgiyi kullanarak nominal seviye bilgisini işler.

Daha önceki çalışmalardan birinde [22], bir tümce deki açık veya boş adıl in nasıl ve nerede kullanılabileceğini açıklamak üzerine tartışmıştır. [23] çalışmasında, bağlamsal faktörlerin Türkçe anafora çözümlemesinde yüksek öneme sahip olduğundan bahsedilmiştir. Ayrıca, Türkçede adıl sel anaför çözümlemesinde kullanılan durum kuramsal bir hesaplama ortamı olan BABY-SIT'i tanımlamışlardır.

4. Adıl Çözümlemesinde Derin Sinir Ağlarının Kullanımı

Bu bölümde Türkçe metinlerde adıl çözümlemesinde derin sinir ağlarının kullanımı için yürüttüğümüz çalışmamıza ilişkin ayrıntılar sunulacaktır. Sırasıyla yapay sinir ağlarına girdi olarak sunulan öznitelik vektörlerinin içeriği, deneylerde yer alan üç sinir ağı kısaca tanıtılacaktır.

4.1. Öznitelikler

Yapay sinir ağları, verilen örnekleri sınıflandırmak için bir grup özneliğin kullanılmasını gerektirir. Bu çalışmada adıl-öncül çiftlerinin çözümlenmesinde 12 öznitelikten oluşan vektörler ağlara girdi olarak sunulacaktır. Bu öznitelikler aşağıda kısaca açıklanmıştır.

1. *Öncülde büyük harf kullanımı (CLU)*: CLU, öncülün büyük harfle başlaması durumunda 1, aksi takdirde 0 değerini alan ikili bir öznitelik olarak tanımlanmıştır. CLU 1 ise, adıl-öncül çiftinin doğru bir çift olma eğiliminde olması beklenir.
2. *Sözcük sayısı (NT)*: NT özelliği, öncül adayı ile adıl arasındaki sözcük sayısını (iki boşluk arasındaki herhangi bir dize) temsil eder. NT değerlerinin aralığı pencere boyutu ile sınırlıdır.
3. *Karakter sayısı (NC)*: NC, öncül adayı ile adıl arasındaki karakter sayısıdır (boşluklar hariç). Örneğin, aşağıdaki tümceler göz önüne alındığında
Ali eve geldi. Onu özlediğimi biliyordum.
(Onu, Ali) çifti için $NC(\text{Onu}, \text{Ali})=9$ olarak hesaplanır.

4. *Benzersiz karakter sayısı (NUC)*: NUC özelliği, öncül adayı ile adıl arasındaki benzersiz (farklı) karakterlerin (boşluklar hariç) sayısını temsil eder. Örneğin, önceki örnek tümcelerdeki (Onu, Ali) çifti için $NUC(\text{Onu}, \text{Ali})=6$ 'dır.

5. *Adıl ve öncül arasında büyük harf kullanımı (CLUB)*: CLUB özelliğinde, adıl ve öncül arasındaki büyük harf sayısı sayılır. CLUB değerinin düşük olması durumunda adıl ve öncül adayı arasında daha az sayıda özel ad kullanılması beklenir ki bu da ilgili öncülün gerçek öncül olduğunu düşündürür.

6. *Ad sayısı (NN)*: NN, öncül adayı ile adıl arasındaki ad etiketli sözcüklerin sayısıdır. Doğru adıl-öncül çiftleri arasında daha düşük NN değerleri gözlemlenmesi beklenir. Örneğin, aşağıdaki tümcelerde "Ali" öncül adayı ve "Onu" adıl i için $NN(\text{Ali}, \text{Onu}) = 2$ ("Ali" ve "Onu" arasındaki adlar "eve" ve "çantasıyla" sözcükleridir).

Ali eve çantasıyla geldi. Onu unuttuğunu bilmiyordu.

7. *Adıl sayısı (NP)*: NP özniteliği adıl ile öncül adayı arasındaki adıl sayısını ifade eder. İlgili öncül ile adıl arasında çok sayıda adıl varsa, adıl-öncül çiftinin olumsuz/yanlış örnek olmaya daha yakın olduğu varsayılır.

8. *Noktalama sayısı (NPU)*: Bu çalışmada, verilen öncül adayı ve adıl arasındaki noktalama sayısı arttıkça, çiftin bileşenlerinin bağlanma şansını kaybetmesi beklenmektedir. Sonuç olarak NPU özelliğinde adıl ve öncül arasındaki noktalama sayısı sözcük türü etiketleyici çıktıları baz alınarak ölçülmektedir.

9. *Öncülün Uzunluğu (LA)*: Zipf, dildeki en az çaba ilkesi nedeniyle insanların daha kısa sözcükleri daha sık kullanma eğiliminde olduğunu belirtmektedir [24]. Zipf'in iddiasından yola çıkarak, uzun sözcüklerin kısa biçimleriyle, başka bir deyişle adıllarla değiştirilmeye daha yatkın olduğu söylenebilir. Bu bakış açısından yola çıkarak, adıl çözümleme özelliği olarak öncül adaylarının uzunluğu kullanılmıştır. LA, öncül adayının karakter cinsinden uzunluğunu temsil etmektedir.

10. *Çoğul adıl (PP)*: Bu çalışmada, çok sözcüklü öncüller için, sınıflandırıcı çok sözcüklü bir öncülün kurucu sözcüklerinden birini ilgili adıl le eşleştirirse, eşleşme doğru pozitif / isabet olarak kabul edilir. Örneğin, aşağıdaki tümcelerde "onlar" adıl i "Ali ve Ayşe" çok sözcüklü öncülüne gönderme yapmaktadır.

Ali ve Ayşe eve geldi. Onlar çok yorgundu.

Sonuç olarak, (Onlar, Ali) (Onlar ve) ve (Onlar, Ayşe) çiftlerinden herhangi biri adıl-önad çifti olarak atanırsa, bu bir başarı (isabet) olarak kabul edilir. PP özelliği bu varsayıma dayanmaktadır. Eğer çift çoğul adıl içeriyorsa PP değeri 1 olarak atanır, çünkü çoğul adıllar için pozitif çiftlerin sayısı tekil adıllara kıyasla daha fazla olacaktır. Başka bir deyişle, çoğul adıllarla eşleştirilen öncül adaylarının adıl tarafından gerçekten ifade edilme şansı daha yüksektir.

11. *Sayı uyumu (NUMA)*: Sayı uyumu özelliğinde adil ve öncül adayının çoğulluk durumu açısından uyuşup uyuşmadığına bakılır. NUMA değeri NA1, NA2, ..., NA6 olmak üzere altı sayı uyumu göstergesinin bir kombinasyonudur. Bu parametreler şunlardır

NA1: Adil çoğulsa ve öncül adayı "-lar", "-ler" çoğul eklerini içeriyorsa veya öncül bir sayı ise NA1=1; aksi takdirde NA1=0 olur.

NA2: Adil çoğulsa ve öncül adayının ardından "ve", "ile" ve "," gibi bir bağlaç belirteci geliyorsa NA2=0,75; aksi takdirde NA2=0 olur.

NA3: Adil çoğulsa ancak çoğul bir öncülün söz dizimsel veya morfolojik kanıtı yoksa NA3=0,5; aksi takdirde NA3=0 olur.

NA4: Adil tekil ise ve çoğul bir öncül olduğuna dair kanıt yoksa NA4=0,75; aksi takdirde NA4=0 olur.

NA5: Adil tekil ise ve öncül adayı "-lar" veya "-ler" çoğul ekini içeriyorsa, NA5=0,25; aksi takdirde NA5=0 olur.

NA6: Adil tekilse ve öncül adayı "ve", "ile" ve "," gibi bir bağlaç belirteci izliyorsa veya öncül adayı bir sayı ise NA6=0,25; aksi takdirde NA6=0 olur.

NUMA, altı NA değerinin toplanması ve NUMA aralığını [0 1]'e yerleştirebilmek için toplama sonucunun 3,5'e bölünmesiyle elde edilir.

12. *Çoğul öncül (PA)*: PA, öncül adayının çoğul olup olmadığını veya çoğul bir ifadenin parçası olup olmadığını temsil eden bir özelliktir. Bu özellik iki durum göz önünde bulundurularak ölçülür:

Durum 1: Eğer öncül adayı "-lar" veya "-ler" çokluk eklerinden biriyle bitiyorsa PA1 değeri 1'e, bitmiyorsa 0'a atanır.

Durum 2: Eğer öncül adayından sonra gelen belirteç "ve" veya "ile" ise PA2= 1, aksi takdirde PA2=0 olarak atanır.

Bu durumlar göz önüne alındığında, PA1PA2 değerlerinin dört farklı olası kombinasyonu ("00", "01", "10" ve "11") gözlemlenebilir. Eğer kombinasyon "10", "01" veya "11" ise, aday öncül çoğul olarak kabul edilir ve karşılık gelen PA değerleri 1 olur; aksi takdirde PA=0 olarak atanır.

4.2. Çok Katmanlı Algılayıcı Ağlar

Derin ileri beslemeli ağlar olarak da bilinen çok katmanlı algılayıcı sinir ağları, giriş ve çıkış katmanları arasında bir veya daha fazla katmana sahip sinir ağlarıdır. Çok katmanlı algılayıcıların en basit şekli tek bir gizli katmana sahiptir [25]. Standart çok katmanlı algılayıcıda, sonlu sayıda nörona sahip birden fazla ardışık gizli katman vardır ve katmandaki her nöron, giriş katmanındakiler hariç doğrusal olmayan bir aktivasyon fonksiyonuna sahiptir. Çok katmanlı algılayıcı ağlar eğitimde geri yayılım algoritmasını kullanır.

Çok katmanlı algılayıcı ağlarda, bir katmandaki her bir nöron, bir sonraki ve bir önceki katmandaki her bir nörona bağlanır. Bu bağlantılar sinapsis olarak adlandırılır. Bir katmandan diğerine bilgi akışı bu sinapsisler tarafından sağlanır ve bu akış

ileri besleme olarak adlandırılır. Son olarak, ağıncı çıktısı çıkış katmanları tarafından aktarılır [26]. Eğitimde, hata geri yayılım algoritması çok katmanlı algılayıcı ağlarda yaygın olarak kullanılır. Basitçe, geri yayılım, ağıdaki hatayı ortadan kaldırmak için ağırlıkları ve eşikleri geriye doğru bir şekilde yeniden hesaplar. Geri yayılım sırasında hata örüntüleri sırayla yayılır. Sınıflandırma görevinde, eğitilen sinir ağı ileri doğru çalışır. Ancak, öğrenme kuralları tarafından kullanılan ağırlık ayarları, gizli katmanlar aracılığıyla çıkış katmanından giriş katmanına geriye doğru yayılır [27]. Genel olarak, eşleme (giriş ve çıkış arasındaki) hatası kümülatiftir ve tüm eğitim seti üzerinden hesaplanır.

4.3. Evrimsel Sinir Ağları

Evrimsel sinir ağı, en az ön işlem gerektirecek şekilde tasarlanmış çok katmanlı algılayıcı modelinin bir çeşididir. Evrim katmanları, alt örnekleme katmanları ve tam bağlı gizli katmanları içeren bir dizi çoklu gizli katman tarafından oluşturulur. Bu tür sinir ağlarının görüntü ve video tanıma, konuşma tanıma, sınıflandırma ve doğal dil işleme gibi birçok farklı alanda yaygın kullanımının olduğu belirtilmektedir [28]. Evrimsel sinir ağlarının belirli bir resimdeki nesnelere tanımlama veya yüz tanıma konusunda çok başarılı olduğu bilinmektedir ve el yazısı karakterlerinin tanınmasında, robot ve sürücüsüz araba teknolojilerinde vizyonu güçlendirmek için kullanılmıştır [29].

4.4. Tekrarlayan Sinir Ağları

Tekrarlayan sinir ağı, kısa ve/veya uzun vadeli bağımlılıkları keşfetmek için sıralı bilgilerin kullanıldığı ileri beslemeli sinir ağının değiştirilmiş bir versiyonudur [30]. Basitçe, geleneksel sinir ağlarında, her t zamanında, sınıflandırmada yalnızca mevcut girdi dikkate alınır. Başka bir deyişle, önceki girişlerin veya sonraki girişlerin sonuçlarından kaçınılır ve yalnızca mevcut ağa beslenen giriş değerlendirilir. Öte yandan, tekrarlayan sinir ağlarında, geçmiş çıktılar girdi olarak kullanılmasına izin verilen gizli durumlar vardır.

5. Deneysel Düzenek

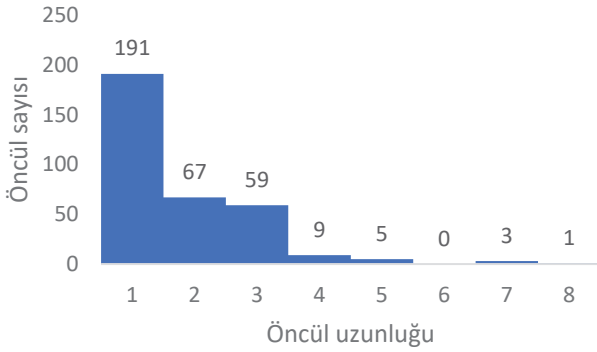
Bu bölümde deneylerimizde kullanılan veri seti, deney parametreleri ve değerlendirme ölçütleri sırasıyla sunulacaktır.

5.1. Veri Kümesi

Bu çalışmada, 10 Türkçe çocuk öyküsünden oluşan bir koleksiyon (toplam 1274 tümce -11613 sözcük) adil -öncül çiftleri veri kümesini oluşturmak için kullanılmıştır. Hikayeler web kaynaklarından elde edilmiş ve rastgele seçilmiştir. Çocuk hikayeleri diğer yazı türlerine göre daha anlaşılır ve kısa oldukları için tercih edilmiştir. Veri kümesinin oluşturulmasından önce lokmalama (tokenizasyon), sözcük türü (İng. Part of Speech-POS) belirleme ([30]'da sunulan POS etiketleyici ile) ve elle etiketleme görevleri gerçekleştirilmiştir.

Elle etiketlemede POS belirleyicisi tarafından adil olarak etiketlenen her sözcük (toplam 442 sözcük) incelenir. Bu inceleme aslında adil olarak etiketlenen sözcüğün uygun bir öncülü olup olmadığını kontrol etmek için yapılmaktadır.

Toplam 107 adıl etiketli sözcüğün metinde geçerli bir öncüle atıfta bulunmadığı gözlemlenmiştir. Bu atıfta bulunmayan sözcükler hariç tutulduğunda, geriye kalan adıl etiketli sözcükler (kısaca adıllar) öncülleri ile eşleştirilerek pozitif adıl-öncül çift örnekleri elde edilmiştir. Bu eşleştirme işlemi sırasında bazı adılların tek bir sözcük yerine bir sözcük öbeğine gönderme yaptığı görülmüştür. Şekil 1'de öncüllerin sözcük sayısı cinsinden uzunluklarına ilişkin istatistikler verilmiştir. Yatay eksen öncüllerin uzunluğunu (içerdikleri sözcük sayısını), dikey eksen ise atıfta bulunan adılların toplam sayısını göstermektedir. Örneğin, 191 adıl in tek sözcüklük öncüllere sahip olduğu, 67 adıl in ise iki sözcüklük öncüllere atıfta bulunduğu görülmektedir.



Şekil-1: Öncül istatistikleri (öncülde bulunan sözcük sayısı cinsinden)

Adıl in birden fazla sözcükten oluşan bir öbeğe atıfta bulunduğu durumlarda, ilgili öbek sözcüklere bölünmüştür. İfadedeki her sözcük için, sözcük-adıl çiftinin bir isabet (doğru bir sınıflandırma/doğru çift) olduğu varsayılarak pozitif örnekler eklenmiştir.

Aşağıda, ikinci tümce deki "onu" adıl inin birinci tümce deki "güllerinin en güzeli" ifadesine gönderme yaptığı böyle bir duruma örnek verilmiştir.

Güllerinin en güzeli solmamış. İyi yürekli peri, her gün onu evinin penceresinden seyrediyormuş.

Yukarıdaki örnek için (onu, güllerinin), (onu, en) (onun, güzeli) çiftleri veri kümesine pozitif örnek olarak eklenmiştir.

Sonuç olarak veri kümesi, 227 benzersiz öncül ve 56 benzersiz adıl in bulunduğu 593 adıl-öncül çifti içermektedir. Gerçek öncülleri ile birlikte 593 adıl deneylerimizde pozitif örnekler olarak kullanılmaktadır. Negatif örnekler kümesi, farklı pencere boyutları kullanılarak deneylerde oluşturulmuştur. Buradaki pencere boyutu terimi, negatif bir çift örneği oluşturmak için adıl le eşleştirilecek sözcük sayısını (adıl ve gerçek öncülü arasında) temsil eder. Pencere boyutu (W) deneylerimizde 6 farklı değere (W= 5, 10, 15, 20, 25 ve 30) ayarlanmıştır. Kısaca, pencere boyutu W=n olduğunda, her adıl in önceki (en yakın) n sözcüğü negatif çiftlerde kullanılmıştır. Bu durumda gerçek öncül verilen pencere içindeyse, negatif örnek sayısı n-1 olmaktadır.

Çizelge 2'de verilen örnek metin göz önüne alındığında, W=n sütunları, Sözcük No=17 olan "onları" adıl i için pozitif ve negatif örnek çiftlerini göstermektedir. Örneğin W=5

olduğunda (onları, güllerini) pozitif, (onları, sabah), (onları, onu), (onları, sever), (onları, de) çiftleri ise negatif çiftler olarak etiketlenmiştir.

Çizelge 3, farklı pencere boyutlarıyla oluşturulan her bir veri kümesindeki pozitif/negatif örneklerin (çiftlerin) nihai sayısını vermektedir.

Çizelge 2. Pencere tabanlı veri seti oluşturma - Örnek metin

Sözcük No	Sözcük	W=5	W=10	W=15
1	Güller	-	-	-
2	öyle	-	-	Negatif
3	güzellermiş	-	-	Negatif
4	ki	-	-	Negatif
5	gören	-	-	Negatif
6	herkes	-	-	Negatif
7	perinin	-	Negatif	Negatif
8	güllerine	-	Negatif	Negatif
9	hayran	-	Negatif	Negatif
10	kalırmış	-	Negatif	Negatif
11	Peri	-	Negatif	Negatif
12	de	Negatif	Negatif	Negatif
13	güllerini	Pozitif	Pozitif	Pozitif
14	sever	Negatif	Negatif	Negatif
15	her	Negatif	Negatif	Negatif
16	sabah	Negatif	Negatif	Negatif
17	onları	Adıl	Adıl	Adıl
18	sularmış	-	-	-

Çizelge 3. Veri kümelerindeki pozitif/negatif örnek sayısı (çiftler)

Veri kümesi	Pencere Büyüklüğü	Pozitif Çift	Negatif Çift	Toplam
Küme-5	5	593	2881	3474
Küme-10	10	593	5714	6307
Küme-15	15	593	8575	9168
Küme-20	20	593	11468	12061
Küme-25	25	593	14369	14962
Küme-30	30	593	17298	17891

5.2. Deneysel Ayarlar

Bu çalışmada deneyler Tensorflow-Keras [32] ve DeepLearning4j araçları [33] kullanılarak gerçekleştirilmiştir. Tüm sınıflandırma deneylerinde, aşırı uyum sorununun üstesinden gelmek için n-kat çapraz doğrulama uygulanmıştır. Kısaca, n-kat çapraz doğrulamada veri kümesi (hem pozitif hem de negatif örnekler dahil) eşit büyüklükte n alt kümeye bölünür. n-1 alt küme eğitimde ve bir bölme testte kullanılır. Her denemede (katta), test alt kümesi daha önce test kümesi olarak kullanılmamış olan eğitim alt kümelerinden biriyle değiştirilir.

Her bir veri kümesi için (Küme-5 ... Küme-30), tüm deneyler farklı sayıda katmana ve/veya farklı sayıda nörona sahip üç farklı derin öğrenme makinesi için ayrı ayrı

gerçekleştirilmiştir. Çok katmanlı algılayıcı ağa özgü deneysel ayarlar aşağıda sunulmuştur:

- Çıktı sayısı=1
- Yineleme sayısı=1000
- Parti (İng. batch) büyüklüğü= Girdi büyüklüğünün %10'u
- Tohum değeri=6
- Algoritma öğrenme oranı =0.1

Çok katmanlı algılayıcı modeli için alanyazında yer alan çok sayıda farklı aktivasyon fonksiyonu arasından tanh aktivasyon fonksiyonu diğer aktivasyon fonksiyonlarına kıyasla daha hızlı yakınsadığı ve çok katmanlı algılayıcı sinir ağlarında daha iyi doğruluk sonuçları elde ettiği belirtildiği için deneylerimizde tercih edilmiştir [34]. Xavier yöntemi, ağırlıkların doğru olduğundan emin olmak için birçok katman boyunca sinyalleri makul bir değer aralığında tuttuğu ve çıkış aktivasyonlarının giriş aktivasyonları noktasındaki dağılımını dikkate aldığı için ağırlık ilk değere atamada kullanılmıştır [35].

Çok katmanlı algılayıcı derin sinir ağlarında iki ana parametre vardır. Birincisi ağdaki gizli katman sayısı, ikincisi ise her katmandaki nöron sayısıdır. Katman sayısındaki değişikliğe bağlı olarak performanstaki değişimi incelemek için aşağıdaki ağ türleri üzerinde deneyler yapılmıştır.

Ağ FL. Az katmanlı (1 girdi + 1 çıktı + 2 gizli katman)

Ağ ML. Orta sayıda katmanlı (1 girdi + 1 çıktı + 7 gizli katman)

Ağ TL. Çok katmanlı (1 girdi + 1 çıktı + 18 gizli katman)

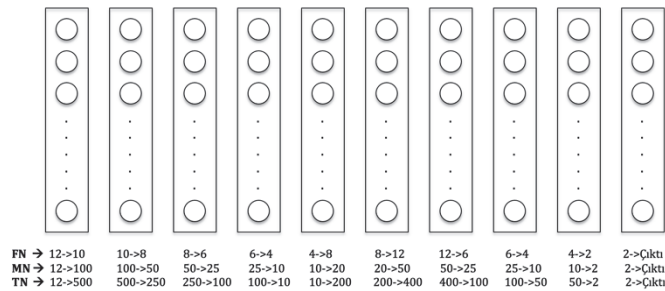
Şekil 2, nöron sayısını değiştirerek oluşturulan farklı ağ türlerini (Ağ FN, Ağ MN, Ağ TN) göstermektedir. Bu deneylerde ağdaki katman sayısı 10 olarak ayarlanmıştır. Farklı ağ türleri şu şekilde adlandırılmıştır:

Ağ FN. Az nöronlu

Ağ MN. Orta sayıda nöronlu

Ağ TN. Çok nöronlu

Her ağ/konfigürasyon için ilk katman girdiyi alır ve önce nöron boyutunu azaltır, sonra nöron boyutu artırılır ve daha sonra ikili sınıflandırma sonucunu döndürmek için önceden kararlaştırılan çıkış katmanı için nöron sayısı tekrar azaltılır. Örneğin, Ağ FN'de ilk katman 12 giriş değeri alır ve 10 tane üretir. İkinci katmanda 10 girdi 8 çıktı değerine düşürülür ve bu şekilde devam eder.



Şekil-2: Çok katmanlı algılayıcılar (Ağ FN, MN ve TN)

Çalışmamızdaki ikinci yöntem olan evrişimli sinir ağı, aşağıda verilen parametre ve deneylerimizde gerekli başlangıç değerleri ile kurulmuştur. Bunlar

- Girdi sayısı=12 (öznitelik sayısı)
- Çıktı sayısı=1
- Yineleme sayısı=1000
- Parti (İng batch) büyüklüğü= Giriş büyüklüğünün %10'u,
- Tohum değeri=6
- Algoritma öğrenme oranı =0.01

şeklinde listelenir.

Önceki çalışmalarda evrişimli sinir ağı modelleri için çok farklı aktivasyon fonksiyonları kullanılmasına rağmen, hangi aktivasyon fonksiyonunun hangi tip problemlerde daha verimli olduğu kesin değildir. [36]'da evrişimli sinir ağlarının genellikle çok katmanlı algılayıcı sinir ağlarına kıyasla daha yüksek zaman karmaşıklığına sahip olduğu, bu nedenle tasarımda daha düşük maliyetli bir fonksiyonun tercih edilmesinin gerektiği belirtilmektedir. Bu sebeple, çalışmamızda sigmoid ve tanh aktivasyon fonksiyonlarına kıyasla hesaplama açısından daha düşük maliyetli olduğu bilinen relu aktivasyon fonksiyonu kullanılmıştır. Çok katmanlı algılayıcı ağına benzer şekilde ilk ağırlık değeri atamada Xavier yöntemi kullanılmıştır. Bilişsel sinir ağı için optimizasyon algoritması olarak ise stokastik kademeli azalma (İng. stochastic gradient descent) tercih edilmiştir.

Çok katmanlı algılayıcı deneylerine benzer şekilde, evrişimli derin sinir ağı deneyleri için katman sayısı ve nöron sayısı ana parametreler olarak kabul edilmiş ve 6-kat çapraz doğrulama yaklaşımı kullanılmıştır. Deneylerimizde ilk olarak katman sayısı filtre değerleri üzerinden değiştirilmektedir. İkinci olarak, toplam katman sayısı sabitlenerek nöron sayıları değiştirilmiştir.

Evrişimli katman ve alt örnekleme katmanlı bir evrişimli sinir ağı oluşturulurken katman sayısının belirlenmesi için kernel ve stride değerlerinin önceden hesaplanması gerekmektedir çünkü her katmanın çıkış hacminin boyutu, kernel ve kaydırma adım (İng. stride) değeri ile hesaplandığında sistemdeki katman sayısını değiştirebilmektedir. Katman sayısındaki değişime bağlı olarak performanstaki değişimi incelemek için çeşitli ağ türleri üzerinde deneyler gerçekleştirilmiştir. Bu ağ türleri

Ağ VFL. Çok az katmanlı (1 girdi + 1 çıktı + 1 gizli katman)

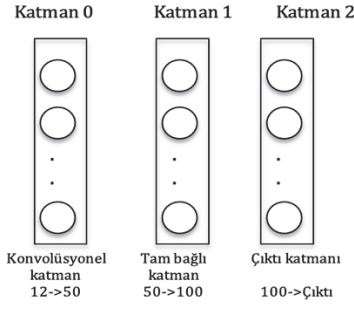
Ağ FL. Az katmanlı (1 girdi + 1 çıktı + 2 gizli katman)

Ağ ML. Orta sayıda katmanlı (1 girdi + 1 çıktı + 6 gizli katman)

Ağ TL. Çok katmanlı (1 girdi + 1 çıktı + 18 gizli katman)

şeklinde kurgulanmıştır.

Katman sayısındaki değişimin etkisinin incelenebilmesi için her katmandaki nöron sayısı tek bir değere sabitlenmiştir. Şekil 3'te Ağ VFL örnek olarak verilmiştir. Ağ VFL'de toplam 3 katman bulunmaktadır. Bilişsel katman olan 0. katman, çekirdek değeri 12 ve adım değeri 1 olan 12 girişe sahiptir. Bu katman 12 değer arasında herhangi bir korelasyon bulmaya çalışır ve daha sonra tam bağlantılı katmana 50 girdi üretir. Tam bağlı katman 50 girdi alır ve girdi hacmini 100'e çıkarır ve çıktı değerine karar vermek için çıktı katmanına gönderir.



Şekil-3: Ağ VFL – Çok az katmanlı evrişimli sinir ağı

Şekil 4, katman sayısının 7'ye sabitlendiği, nöron sayısının değiştirilmesiyle oluşturulan farklı evrişimli sinir ağı konfigürasyonlarını göstermektedir. Bu ağlar aşağıda listelenmiştir:

Ağ FN. Az nöronlu

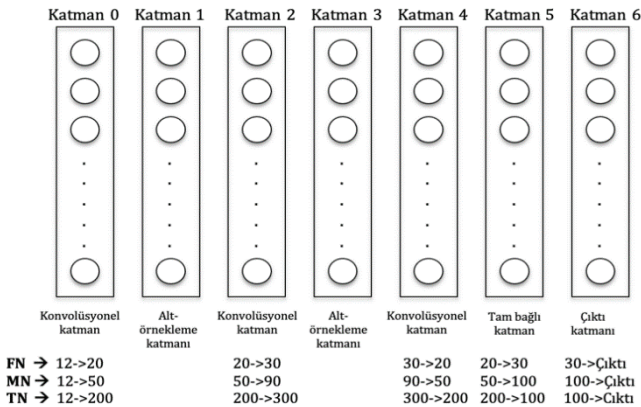
Ağ MN. Orta sayıda nöronlu

Ağ TN. Çok nöronlu

Tüm ağ türlerinde, sıfırıncı katman, çekirdek değeri 2 ve adım değeri 2'ye eşit olan 12 giriş değerini alan evrişimli katmandır. Sonraki tüm evrişimli ve havuzlama katmanlarında, çekirdek ve adım değeri sırasıyla 2 ve 1'e eşit olarak ayarlanır.

Çalışmamızda yer alan üçüncü makine tekrarlayan sinir ağıdır. Deneylerimizde tekrarlayan sinir ağının parametreleri ve gerekli başlangıç değerleri aşağıdaki gibi ayarlanmıştır:

- Girdi sayısı=12 (öznitelik sayısı)
- Çıktı sayısı=1
- Gizli katman(lar)daki birim sayısı =4
- Tohum değeri=6
- Çıktı dizisi= Doğru(çıktı dizisindeki son çıktı geri döndürülür)



Şekil-4: Evrişimli sinir ağları (Ağ FN, MN ve TN)

Tekrarlayan sinir ağı deneylerinde, çok katmanlı algılayıcıya kıyasla ağlarla ilgili karmaşık yapıları nedeniyle katman sayısı ana parametre olarak kabul edilmektedir. Çalışmamızda 4 kat çapraz doğrulama yaklaşımını kullanılmış; giriş katmanı bir gömme katmanı olarak ayarlanmış ve çıkış katmanı, çıkış sayısını bire düşüren yoğun bir katman olarak tasarlanmıştır. Tekrarlayan ağlarda gizli katman sayısındaki değişikliğe bağlı olarak performanstaki değişimi incelemek için aşağıdaki ağ türleri ile deneyler gerçekleştirilmiştir:

Ağ FL. Az katmanlı (1 giriş + 1 çıkış + 2 gizli katman)

Ağ ML. Orta sayıda katmanlı (1 giriş + 1 çıkış + 7 gizli katman)

Derin öğrenme ile adıl çözümlene performansının ölçümü için F1- ölçüsü kullanılmıştır. F1 ölçüsü aşağıdaki şekilde ifade edilir.

$$F1 = \frac{2TP}{2TP+FN+FP} \quad (1)$$

Burada TP doğru pozitiflerin sayısı (doğru çiftler olarak sınıflandırılan ve gözlemlenen çiftler), FP yanlış pozitiflerin sayısı (yanlış bir şekilde adıl -öncül çifti olarak sınıflandırılan çiftler) ve FN yanlış negatiflerin sayısıdır (aslında doğru çiftler olan ancak yanlış çiftler olarak sınıflandırılan çiftler).

6. Deney Sonuçları

Bu çalışmada n-kat çapraz doğrulama yöntemiyle toplam 15 farklı ağ türü ile sınıflandırma deneyleri gerçekleştirilmiştir. Aşağıda sırasıyla çok katmanlı algılayıcı, evrişimli ve tekrarlayan sinir ağlarının deneysel sonuçları ağırlıklandırılmış F1 ölçüsü ile verilecektir. Son olarak, üç derin sinir ağının sınıflandırma performansları karşılaştırılacaktır.

Çizelge 4'te farklı katman sayısına sahip çok katmanlı algılayıcı sinir ağlarının performans sonuçları verilmiştir. Çizelge 4'te W, veri kümelerindeki negatif örnek sayısını değiştiren pencere boyutunu temsil etmektedir ve sunulan performans değerleri 6-katlı çapraz doğrulamanın ağırlıklı F1 ortalamalarından oluşmaktadır. Her sütundaki altı çizili değerler ilgili ölçümde elde edilen en yüksek değerleri, son satır ise tüm pencere boyutları için ortalama değerleri vermektedir. Son satırdaki koyu değerler, üç ağ türü dikkate alındığında ilgili ölçütün en yüksek ortalama değerini göstermektedir.

Çizelge-4: Deney sonuçları – Farklı katman sayılarına sahip çok katmanlı algılayıcı sinir ağları

W	Ağ FL	Ağ ML	Ağ TL
5	0,936	0,943	0,939
10	0,868	0,881	0,879
15	0,824	0,830	0,827
20	0,784	0,798	0,785
25	0,759	0,765	0,797
30	0,745	0,748	0,731
Ortalama	0,819	0,828	0,826

Farklı sayıda katmana sahip çok katmanlı algılayıcı ağlar üzerinde yapılan deneyler göstermiştir ki

1. F1 ölçüsü dikkate alındığında, (tüm ağ türlerinde) derin ağ pencere boyutu 5 olarak ayarlandığında en yüksek başarı elde edilmiştir.
2. Ortalama değerler incelendiğinde (Çizelge 4'te son satır), en yüksek F1 ölçüsü sinir ağının orta katman sayısına sahip olduğu durumda (Ağ ML) elde edildiği görülmüştür.

Farklı nöron sayılarına sahip çok katmanlı algılayıcı ağının değerlendirme sonuçları Çizelge-4'e benzer şekilde Çizelge 5'te verilmiştir.

Çizelge-5. Deney sonuçları – Farklı nöron sayılarına sahip çok katmanlı algılayıcı sinir ağları

W	Ağ FN	Ağ MN	Ağ TN
5	0,937	0,942	0,946
10	0,868	0,883	0,888
15	0,822	0,834	0,842
20	0,788	0,798	0,804
25	0,749	0,775	0,785
30	0,659	0,738	0,749
Ortalama	0,804	0,828	0,836

Farklı nöron sayısına sahip çok katmanlı algılayıcı ağlar üzerinde yapılan deneyler aşağıdaki önemli sonuçları ortaya çıkarmaktadır:

1. Tüm ağ türlerinde, en yüksek F1 ölçüsü pencere boyutu (W=5) en küçük olduğunda elde edilmektedir.
2. F1 ölçüsünün ortalama değerleri göz önüne alındığında, çok fazla nöron kullanıldığında (Ağ TN) sınıflandırma sisteminin en yüksek başarıma ulaştığı görülmektedir.
3. F1 ortalama değerlerine bakıldığında, farklı ayarlarda doğruluk değerleri benzer olsa da çok fazla nöron kullanıldığında F1 değerinin (F1=0,836) çok daha yüksek olduğu görülmektedir.

Evrışimli sinir ağı deneylerinde çapraz doğrulamada her kat için tekrarlanacak toplam 7 (4 farklı katman sayısı + 3 farklı nöron sayısı) deney bulunmaktadır. Farklı katman sayılarıyla (Tip VFL, FL, ML, TL) yapılan deneylerin sonuçları Çizelge-6'da verilmiştir. Burada W pencere boyutunu, sütunlar 6 katın ağırlıklı F1 ortalamalarını göstermektedir. Çizelge-6'da altı çizili değerler ilgili sütun için en yüksek değerlerdir. Ortalama satırı sütunlardaki değerlerin ortalamalarını gösterir ve mean yüksek ortalama değer(ler) koyu olarak verilmiştir.

Çizelge-6: Deney sonuçları – Farklı katman sayılarına sahip evrişimli sinir ağları

W	Ağ VFL	Ağ FL	Ağ ML	Ağ TL
5	0,934	0,932	0,933	0,929
10	0,872	0,868	0,858	0,860
15	0,826	0,817	0,820	0,820
20	0,786	0,784	0,773	0,780
25	0,758	0,761	0,746	0,729
30	0,742	0,744	0,738	0,712
Ortalama	0,820	0,818	0,811	0,820

Farklı katman sayısına sahip evrişimli sinir ağları üzerinde yapılan deneyler, F1 ölçüsü dikkate alındığında ağın tüm türler için W=5 olduğunda en başarılı olduğunu göstermektedir. Ortalama performans değerleri göz önüne alındığında (Çizelge 6'da son satır), tek bir ağ türünün diğerlerinden anlamlı derecede daha başarılı olmadığı görülmüştür. Sonuç olarak, adil çözümleme probleminde evrişimli ağda çok fazla katmana ihtiyaç olmadığı söylenebilir.

Farklı nöron sayılarının etkisini gözlemlemek için evrişimli sinir ağı modeli üzerinde yapılan deneylerin sonuçları Çizelge- 6 ile aynı notasyon/yapı kullanılarak Çizelge-7'de gösterilmiştir. Çizelge-7'deki sonuçlar incelendiğinde, en yüksek F1 değerlendirme puanlarını elde etmek için W=5 penceresinin gerekli olduğu açıkça görülmektedir. Farklı nöron sayıları dikkate alındığında tüm farklı ağlar için neredeyse aynı maksimum performans sonuçları elde edilmiştir.

Çizelge-7'deki ortalama değerler incelendiğinde, evrişimli ağlarda az sayıda nöron yerine orta sayıda (Ağ MN) veya çok fazla sayıda nöron (Ağ TN) kullanılmasının tercih edilmesi gerektiği görülmektedir.

Çizelge-7: Deney sonuçları – Farklı nöron sayılarına sahip evrişimli sinir ağları

W	Ağ FN	Ağ MN	Ağ TN
5	0,933	0,935	0,935
10	0,857	0,860	0,862
15	0,809	0,820	0,821
20	0,767	0,769	0,773
25	0,738	0,743	0,746
30	0,735	0,737	0,736
Ortalama	0,807	0,811	0,812

Çizelge 8'de önceki çizelgelere benzer şekilde az ve orta sayıda katmana sahip tekrarlayan sinir ağı deneylerinin F1 sonuçları sunulmaktadır. Çizelge-8'de W pencere boyutunu ifade etmektedir. Ortalama satırı F1 değerlerinin ortalamalarını gösterir ve en yüksek ortalama değer koyu olarak verilmiştir.

Çizelge-8: Deney sonuçları – Farklı katman sayılarına sahip tekrarlayan sinir ağları

W	Ağ FL	Ağ ML
5	0,916	0,918
10	0,733	0,764
15	0,630	0,657
20	0,409	0,525
25	0,327	0,441
30	0,351	0,100
Ortalama	0,561	0,567

Farklı sayıda katmana sahip tekrarlayan sinir ağları üzerinde yapılan deneyler, W arttıkça F1 ölçüsünün önemli ölçüde azaldığını göstermektedir. Tekrarlayan sinir ağlarının birçok doğal dil işleme probleminde başarılı olduğu bilinmesine rağmen, çalışmamızda önceden belirlenmiş özniteliklerin bağlılık bilgisi içermediği bilindiği için deneysel sonuçlarımız şaşırtıcı değildir.

Bilişsel, tekrarlayan ve çok katmanlı algılayıcı sinir ağlarının adil çözümlemedeki performans karşılaştırmasında farklı ağ türleri için elde edilen F1 sonuçları dikkate alınmıştır. Bu karşılaştırmada kullanılacak olan F1 değerleri farklı büyüklükteki veri kümelerinden elde edilen puanların ortalamasıdır. Çizelge-9 ve Çizelge-10'da sırasıyla farklı katman sayısı ve farklı nöron sayısına sahip ağlar için F1

değerleri verilmiştir. Çizelge 9-10'daki ortalama satırları ilgili sütundaki değerlerin ortalamasını, koyu değerler ise ilgili ölçütte elde edilen en yüksek değerleri temsil etmektedir.

Çizelge-9: Farklı katman sayısına sahip evrişimli, çok katmanlı algılayıcı ve tekrarlayan ağlar için ortalama F1 değerleri

	Bilişsel	Çok katmanlı algılayıcı	Tekrarlayan
Ağ VFL	0,844	-	-
Ağ FL	0,843	0,839	0,561
Ağ ML	0,837	0,851	0,567
Ağ TL	0,832	0,844	-
Ortalama	0,839	0,845	0,564

Çizelge-10: Farklı nöron sayısına sahip evrişimli ve çok katmanlı algılayıcı ağlar için ortalama F1 değerleri

	Bilişsel	Çok katmanlı algılayıcı
Ağ FN	0,833	0,831
Ağ MN	0,837	0,849
Ağ TN	0,838	0,858
Ortalama	0,836	0,846

Çizelge 9'da yer alan

1. En yüksek F1 değeri orta sayıda katmanla oluşturulan çok katmanlı algılayıcı ile elde edilir
2. En yüksek ortalama F1 puanı çok katmanlı algılayıcı ile elde edilir

sonuçları değerlendirildiğinde çok katmanlı algılayıcı sinir ağının evrişimli ve tekrarlayan ağlardan daha yüksek başarımlar gösterdiği sonucu çıkarılabilir.

Çizelge 10'da verilen deneysel sonuçlara göre ise en yüksek başarımların çok fazla nörona sahip çok katmanlı algılayıcı ağ ile elde edildiği (Çizelge 10, Ağ TN'de değerlerin altı çizilmiştir) ve ortalama değerler açısından da bu ağın evrişimli sinir ağına göre daha başarılı olduğu görülmektedir. Çizelge 9 ve 10'daki sonuçlar göz önüne alındığında, çok fazla nörona (Ağ TN) ve orta katman sayısına (Ağ ML) sahip çok katmanlı algılayıcı sinir ağının en yüksek performansı elde ettiği söylenebilir. Sonuç olarak, daha uzun çalışma süresi gerektiren evrişimli sinir ağı yerine, çok katmanlı algılayıcı sinir ağının Türkçede daha önce bahsedilen özelliklerle adil çözümleme probleminde kullanılabileceği görülmüştür.

7. Sonuç

Bu makalede, adil çözümlemesi ikili bir sınıflandırma problemi olarak kabul edilerek Türkçe adil-öncül çifti adaylarının öznelik değerleri derin öğrenme makinelerine girdi olarak verilmiştir. Derin öğrenme makinelerinin ilgili problemi çözmedeki başarımlarını araştırılmıştır. Deneyleri gerçekleştirmek için 10 çocuk hikayesinden oluşan bir koleksiyon derlenmiş ve 593 pozitif adil-öncül çiftinden oluşan bir veri kümesi hazırlanmıştır. Veri kümesi, farklı boyutlarda veri kümeleri elde etmek için farklı sayıda negatif örneklerle büyütülmüştür. Derin öğrenmede kullanılmak üzere 12 öznelik tanımlanmıştır.

Çok katmanlı algılayıcı, evrişimli ve tekrarlayan sinir ağları olmak üzere üç derin öğrenme makinesinin performansı farklı konfigürasyonlar dikkate alınarak ölçülmüştür. İlk deney setinde, ağlardaki katman sayısı değiştirilmiştir. Performanslar (çok) az, orta ve yüksek sayıda katman için ölçülmüştür. İkinci sette, her katmandaki nöron sayısı az, orta veya yüksek olarak ayarlanmıştır. Çok fazla nöron kullanan orta sayıda (9) katmana sahip çok katmanlı algılayıcı sinir ağının en yüksek başarımlar sonuçlarını verdiği gözlemlenmiştir.

Kaynakça

- [1] Grosz B. J., Weinstein S., Joshi A.K., *Centering: A Framework for Modelling the Local Coherence of Discourse Centering: A Framework for Modelling the Local Coherence of Discourse*, Computational Linguistics, 1995, 21: 203–225.
- [2] Halliday M.A., Hasan R., *Cohesion in English 1st ed.*, Routledge, <https://www.routledge.com/Cohesion-in-English/Halliday-Hasan/p/book/9780582550414> UK, 1976, 1-392.
- [3] Van Valin R.D., LaPolla R.J., *Syntax: Structure, Meaning, and Function 1st ed.*, Cambridge University Press, UK, 1997, 1-744.
- [4] Van Valin R.D., *A Summary of Role and reference Grammar*, Papers of the Summer Institute of Linguistics, University of North Dakota Session, 2005, 37(5):1–30.
- [5] Smeaton A.F., *Progress in the application of natural language processing to information retrieval tasks*, Computing Journal, 1992, 35(268):268-278.
- [6] Yıldırım S., Kılıcaslan Y., *A Machine Learning Approach to Personal Pronoun Resolution in Turkish*, FLAIRS Conference, USA:Florida, 2007, 269–270.
- [7] Kılıcaslan Y., Guner E.S., Yıldırım S., *Learning-based pronoun resolution for Turkish with a comparative evaluation*, Computers Speech and Language, 2009, 23: 311–331.
- [8] Kornfilt J., *Turkish 1st ed.*, Routledge, UK, 1997, 1-608.
- [9] Hamann J., Hamann J., *On the Syntax and Morphology of Double Agreement in Lavukaleve*, In agreement to Sebastian Bank, Doreen Georgi & Jochen Trommer (eds.) Linguistische Arbeits Berichte, 2010, 197–225.
- [10] Tufekci P., Kılıcaslan Y., *A Computational Model for Resolving Pronominal Anaphora in Turkish Using Hobbs' Naive Algorithm*, Journal of Computing and Information Science in Engineering, 2007, 1: 854–858.
- [11] Hobbs J.R., *Resolving pronoun references*, Lingua, 1978, 44: 311–338.
- [12] Lappin S., Leass H.J., *An Algorithm for Pronominal Anaphora Resolution*, Computational Linguistics, 1994, 20: 535–561.
- [13] Kennedy C., Boguraev B., *Anaphora for Everyone : Pronominal Anaphora Resolution without a Parser*, Proceedings of the 16th International Conference on Computational Linguistics, 1996, 113–118.
- [14] Aone C., Bennett S., *Evaluating automated and manual acquisition of anaphora resolution strategies*, Proceedings of 33rd Annual Meeting of the Association for Computational Linguistics, Massachusetts, 1995, 122–129.
- [15] McCarthy J.F., Lehnert W.G., *Using Decision Trees for Coreference Resolution*, Proceedings of Fourteenth International Joint Conference of Artificial Intelligence, Montreal, Quebec, Canada, 1995, 1–5.

- [16] Soon W.M., Ng H.T., Lim D.C.Y., *A Machine Learning Approach to Coreference Resolution of Noun Phrases*, Computational Linguistics, 2001, 27: 521–544.
- [17] Mitkov R., Evans R., Orasan C., *A New, Fully Automatic Version of Mitkov's Knowledge-Poor Pronoun Resolution Method*, Proceedings of CICLing2002, Mexico City, Mexico, 2002, 2276: 168–186.
- [18] Preiss J., *Choosing a Parser for Anaphora Resolution*, Proceedings of 4th Discourse Anaphora Anaphora Resolution Colloq. (DAARC 2002), Lisbon, 2002, 175–180.
- [19] Turan U.D., *Null vs. Overt Subjects in Turkish Discourse: A Centering Analysis*, Phd. Thesis, University of Pennsylvania, Pennsylvania, 1995, 1-27.
- [20] Yuksel O., Bozsahin C., *Contextually appropriate reference generation*, 2002, Natural Language Engineering, 8: 69–89.
- [21] Chomsky N., *on Binding*, Linguist Inquires; 1980, 11: 1–46.
- [22] Erguvanli-Taylan E., *Pronominal versus zero representation of anaphora in Turkish*, Studies in Turkish Linguistics, 1986, 209–231.
- [23] Tin E., Akman V., *Situated Processing of Pronominal Anaphora*, Verarbeitung Natürlicher Spr, 1998, 369–378.
- [25] Elizondo D., *The Linear Separability Problem: Some Testing Methods*, IEEE Transactions Neural Network, 2006, 17: 330–344.
- [24] Zipf, G. K. *Human behavior and the principle of least effort*, Cambridge: Addison Wesley Press, 1949.
- [26] Pinkus A., *Approximation theory of the MLP model in neural networks*, Acta Numerica, 1999, 8(143): 143-195.
- [27] Zurada, J.M., *Introduction to Artificial Neural Systems* 1st ed., West Group, Minnesota:St. Paul,1992, 1-759.
- [28] Oord A.V.D., Dieleman S., Schrauwen B., *Deep content-based music recommendation*, Advances in Neural Information Processing Systems 26 (NIPS 2013), Nevada, 2013, 643-2651.
- [29] Sermanet P., Eigen D., Zhang X., et al., *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*, 2nd International Conference on Learning Representations (ICLR 2014), Banff, Canada, 2014, 1-17.
- [30] Marhon, S.A., Cameron, C.J.F., Kremer, S.C., *Recurrent Neural Networks*, Handbook on Neural Information Processing. Intelligent Systems Reference Library, Springer, Berlin, Heidelberg, 2013, 49:29-65.
- [31] Kisla T., Karaoglan B., *A hybrid Statistical Approach to Stemming in Turkish: An Agglutinative Language*, Anadolu University-Journal of Science and Technology, 2016, 17: 401–412.
- [32] Abadi, M., Agarwal, A., Barham, P., et.al., *Tensorflow: large-scale machine learning on heterogeneous distributed systems*, Arxiv:1603.04467, 2016.
- [33] <https://deeplearning4j.konduit.ai>. Access date: 29.12.2021.
- [34] Ruck D.W., Rogers S.K., Kabrisky M., et al., *Letters: The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function*, IEEE Transactions on Neural Networks, 1990, 1: 296–298.
- [35] Glorot X., Bengio Y., *Understanding the difficulty of training deep feedforward neural networks*, Proceedings of 13th International Conference on Artificial Intelligence and Statistics (AISTATS), Italy, 2010, 9: 249–256.
- [36] Krizhevsky A., Sutskever I., Geoffrey E.H., *ImageNet Classification with Deep Convolutional Neural Networks*, Advances in Neural Information Processing Systems, 2012, 25:1–9.