

YALPI YIG'ILISH MA'RUZALARI

DOĞAL DİL İŞLEMEDEKİ GELİŞMELER

Eşref Adalı,
Prof. Dr.
İstanbul Teknik Üniversitesi
Bilgisayar ve Bilişim Fakültesi
Türkiye
adali@itu.edu.tr

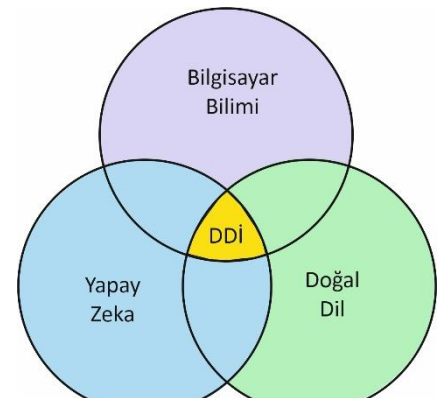
1. Doğal Dil İşlemenin Kısa tarihi

Konuştığımız dilin (doğal dil) bilgisayar ile işlenmesine Doğal Dil İşleme diyoruz. DDİ çalışmalarının ne zaman başladığı konusu, 1954 yılına kadar gitmektedir. O yıllarda, Rusça bilimsel yayınların İngilizceye bilgisayar ile çevrilmesi amaçlanmıştır. Bu amaçla ALPAC (Automatic Language Processing Advisory Committee) kurulmuştur. Dönemin bilgisayarlarının yetenekleri bu çalışmayı sürdürmeye yetmemiştir. Bu dönemin önemli bir çalışması IBM Georgetown Demonstration (1954) ve aynı yıl ilk dergi (Mechanical Translation- daha sonra Computational Linguistics) sayılır.

DDİ nedir sorusunun yanıtı şöyle verilir:

- İnsan dilini anlamaya, yorumlamaya ve işlemeye yardımcı olan bir Yapay Zeka dalı.
- Doğal dili işlemek ve anlamak için kullanılan araçlar, teknikler, yöntemler ve algoritmalar hakkında her şey.
- Bilgisayarların insan dilini anlamasını ve işlemesini sağlamaya odaklanan Yapay Zeka alt alanı.

Şekil-1'de DDİ ile bilgisayar bilimi, yapay zeka ve doğal dil arasındaki ilişki gösterilmiştir.



Şekil-1: DDİ - Bilgisayar bilimi -
Yapay zeka - Doğal dil ilişkisi

2. DDİ'nin Kısa Tarihi

DDİ'nin kısa tarihi aşağıda verilmiştir [1]:

1950'ler

- Erken Makine Çeviri çalışmaları
- Dil bilgisi üzerinde çalışmalar

1960-80'ler

- Uygulamalar

- BEYZBOL: Beyzbol oyunlariyla ilgili bilgileri veri tabaninda aramak uzere hazirlanmistir.
- ELIZA: Çok basit soru-yanit uygulaması
- SHREDLU: Belli sayıda harf içeren sözcük üretme uygulaması
- Mesaj anlama: Gazetelerde yayımlanan yazılardan terörizmle ilgili olanları bulmak ile ilgili bir çalışma
- Makine çevirisi çalışmaları.

• Yöntemler

- ATN (artırılmış geçiş ağları): Bağlamdan bağımsız dilbilgisi yöntemleri ile ilgili çalışmalar.
- Dilbilgisi çalışmaları (etken, nesne, vb.)
- DCG-Belirli Tümce Dilbilgisi
- Bağımlılık dilbilgisi: Öğeler arası bağımlılık araştırması ile ilgili çalışmalar.

1990 ve sonrası

- İstatistiksel yöntemler
- Konuşma tanıma
- Makine Çevirisi Sistemleri
- Soru Yanıtlama
- Büyük Dil Modelleri (BDM-LLMs)

Bazı Güncel DDİ Uygulamaları

Günümüzde çalışılan konulardan bazıları şunlardır:

- Soru Yanıtlama
- Makine Çevirisi
- Duygu Analizi
- Özet Çıkarma
- Metni anlama
- Doğal dil benzeri tümce oluşturma

3. Bir Tümce veya Metni Anlama

Bir tümce veya metni anlama ve sorunları aşağıda kısa açıklanmıştır.

Tümce veya Metni Anlama

- Metin ya da tümceyi (girdi) yararlı gösterime eşleme.
- Dilin farklı yönlerini analiz etme.

Doğal Dil Üretimi

- Doğal dile yakın biçimde, anlamlı ifadeler ve tümceler üretme süreci: Şu konuları içerir:
 - Metni Düzenleme (bilgi kaynağından ilgili içeriği getirme)
 - Tümce Düzenleme (gerekli sözcükleri seçme, anlamlı ifadeler oluşturma, tümcenin tonunu ayarlama)
 - Metni Gerçekleştirme (tümce düzenini tümce yapısına eşleme)

Anlam Belirsizliği

Belirsizlik

- **Sözcüsel/Biçim Bilimsel** (Lexical/Morphological): değişim (Eylem, Ad - V,N), eğitim (Eylem, Ad - V,N), çift (Önad, Belirteç -ADJ,ADV)...
- **Söz Dizimsel**: İngilizcede sorun “Helicopter powered by human flies”, Türkçede sorun yok “İnsan yönetiminde uçan helikopter”
- **Anlamsal**: İngilizcede sorun “He saw a man on the hill with a telescope”, Türkçede sorun yok “ Tepedeki bir adamı teleskopla gördü”
- **Söylem**: anaför ,her dilde sorun. Ekmek sözcüğünün Türkçe çok farklı kullanımı var

Çözüm Yöntemleri

- Daha önceki adımda bulunmuş çözümü kullanmak.
- İstatistiksel yöntem kullanmak
- Dikkat vektörü kullanmak
- Sözcük kesesi kullanmak

4. DDİ'nin Temel Öğeleri

DDİ çalışmalarının temel taşlarını oluşturan öğeler Şekil-2'de sıradüzensel biçimde gösterilmiştir [3].



Şekil 2: DDİ'nin temel öğeleri

Şekil-3'te bir DDİ sürecinde işlemlerin izlediği akış (İş Akışı) gösterilmiştir.



Şekil 3: DDİ İş hattı

DDİ İş hattında görülen işlem ile ilgili açıklamalar aşağıda verilmiştir.

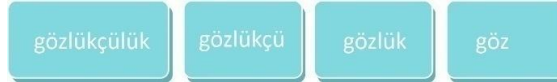
Lokmalama (Tokenization)

- Karmaşık bir tümcelerine ayırmak
- Tümce açısından her bir sözcüğün önemini belirtmek
- Tümceden yapısal bir açıklama oluşturmak
- Noktalama işaretini göstermek



Sözcüğün Kökünü Bulma (Yapım eklerinden arındırma) (Stemming)

Bir sözcüğün kökünü bulma işlemidir. Türkçe gibi eklemeli dillerde yapım ekiyle türetilmiş sözcük kök sözcüğün temel anlamını taşır.



Sözcüğün Çekimli Biçimleri (Lemmatization)

Türkçede sözcüğü çekim eklerinden arındırmak anlamında düşünülebilir. İngilizce gibi kuralsız dillerde anlamı farklıdır. Örneğin “go, going, went” ve “gone” sözcüklerinin temeli “go” dur. Benzer biçimde “better, best” sözcüklerinin temeli “good” sözcüğüdür. Türkçe gibi kurallı dillerde bu tür gariplikler görülmez. Türkçe gibi kurallı dillerde sözcüğün (eylem ya da ad) çekim eklerinden arındırılması olarak yorumlanabilir.

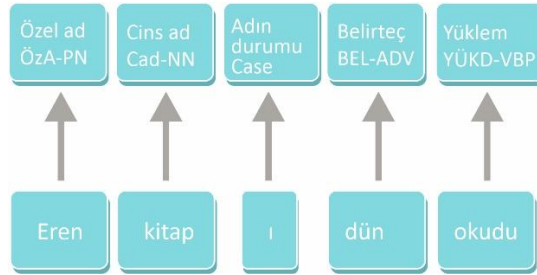


Sağır Sözcüklerin Kaldırılması

Sağır sözcükler, makine öğrenmesinde gürültüye neden olabilir bu nedenle metnin işlenmesinden önce veya sonra kaldırılır.

POS

POS, sözcüklerin söz dizimsel bağamlarına ve rollerine göre atanması sürecidir.



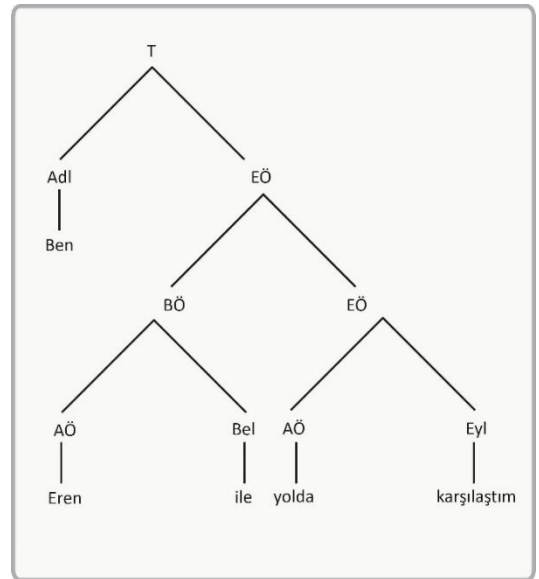
Parçalara ayırma

Bir tümceyi öğelerine ayırma işlemi



Ayrıştırma

Bir cümleyi en küçük bileşenlerine ayırma işlemi



5. Dil Modelleri

Bir dilin özellikleri çıkarabilmek üzere dillerin modelleri çıkarılmaya çalışılmıştır. İlk dönemlerde, bir sözcükten sonra gelebilecek sözcük ya da sözcüklerin nasıl tahmin edileceği üzerinde çalışılmıştır. Bu amaçla n-gram ve Matkov zincirleri geliştirilmiştir. Örneğin “*Her şey çok olacak*” tümcesinde boş olan yere “*güzel, kötü, berbat, muhteşem, ucuz, pahalı*” gibi sözcüklerden biri gelebilir. Bu sözcüklerden hangisinin geleceğini o dilin derlemine bakılarak karar verilir. Derlemde en sık geçen sözcüğün bu boş alana yazılması en doğru seçenektir.

Tanım olarak bir derlemin, bir dilde üretilmiş olan tüm yayınları içermesi beklenir ancak bunu sağlamak çok kolay değildir. Bunun yerine, farklı konularda, farklı yazarlar tarafından üretilmiş metinlerden belli boyutlarda örnek metinler alınarak derlem oluşturulabilir. Böylesine hazırlanmış olan bir derleme “**Dengeli Derlem**” adı verilmektedir. Dillerin zaman içinde geliştiği düşünüldüğünde, dengeli derlemde yer alacak metinlerin belli bir zaman kesitinde yayımlanmış olanlardan seçilmesi gerekir.

Bir dilde yayımlanmış yayınları rastgele toplayarak da derlem hazırlanabilir. Bu durumda hazırlanmış olan derleme “**Dengesiz Derlem**” adı verilir.

Derlem bir dildeki tüm yayınları barındıracak biçimde hazırlanmış ise bu tür derlemlere “**Genel Derlem**”, yalnızca belli bir alanı kapsayan yayımlardan oluşturulmuş ise “**Özel Derlem**” adı verilmektedir.

Diller arası çevirilerde kullanılmak üzere de derlemler hazırlanmaktadır. “**Koşut Derlem**” olarak anılan bu derlemlerde yer alan tümceler birbirinin karşılığı olacak biçimde yer alırlar [2].

Yakın zaman, dil modelleri üzerindeki çalışmalar yoğunlaşmıştır. N-gram kullanmak yerine Büyük Dil Modelleri (LLMs) geliştirilmeye başlanmıştır. BDM'lerini üretmek için genellikle çok büyük miktarda metin bir araya getirilir; bir anlamda dengesiz derlem oluşturulur. Bu veri kümesi üzerinde, bir sonraki sözcük, bir sonraki tümce hatta bir sonraki metin tahmin edilmeye çalışılır. Bazı veri kümeleri metinlere ek olarak resim de içerir. Böylece bir resim de değerlendirilir. Bir sonraki sözcük, tümce veya metni tahmin etmek üzere sinir ağının gelişmiş bir biçimi olan **Derin Öğrenme** yöntemi kullanılır.

5.1 Büyük Dil Modelleri Nelerdir?

Büyük dil modelleri, çok büyük veri kümeleri üzerinde eğitim sırasında edinilen bilgiye dayanarak metni tanıyabilen, çıkarım yapabilen, özetleyebilen, tahmin edebilen ve üretebilen derin öğrenme algoritmalarıdır. Ayrıca dil modelleri adı verilen daha genel bir teknolojinin alt kümesidir. Dil modellerinin ortak özellikleri doğal dilde yazılmış metinleri işleyebilmeleri ve doğal dilde tümceler kurabilmeleridir. Büyük boyuttaki veri kümesi metin ve resim barındırdığından

büyük veri olarak kabul edilir. BDM'leri, bu büyük veri kümesi üzerinde eğitilirler. Bu eğitim aşaması çok sayıda eğitilebilir katsayı (parametre) içerir. Günümüzde çok bilinen iki BDM'nin veri kümesi boyut Çizelge-1'de verilmiştir [4].

Çizelge-1: MT-NLG ve GPT-3'ün Karşılaştırılması

Büyük Dil Modeli	Katsayı (Parametre) sayısı	Eğitim veri kümesindeki imleç (token) sayısı
NVIDIA (MT-NLG)	530 milyar	270 milyar
OpenAI Model: GPT-3 Davinci Model	175 milyar	499 milyar

Çizelge-1'de görülen dil modelleri çok büyük veri üzerinde eğitilirler ve soruları yanıtlamak, özetleri oluşturmak, tümceleri tamamlamak ve diller arası çeviri yapmak amacıyla kullanılırlar.

Bir modelin başarımı, bir anlamda kalitesi, eğitim verisinin boyutuna bağlıdır. Bu nedenle, daha büyük dil modelleri genellikle daha küçük dil modellerine göre daha doğru başarılıdır. Ancak, şunu da hatırlatmakta yarar vardır: büyük dil modellerinin başarımı yalnızca model boyutuna veya veri miktarına bağlı değil; verinin kalitesine de bağlıdır. Örneğin, hakemler tarafından değerlendirilmiş bildiri ya da makaleler veya editör tarafından düzeltilmiş hikaye ve romanlar üzerinde eğitilmiş alan BDM'lerin başarımı sosyal medyadan toplanmış mesajlar, yorumlar veya diğer incelenmemiş içerikler üzerinde eğitilmiş alan BDM'lerden daha başarılı sonuçlar üretirler. Belli bir konuda çalışmak üzere hazırlanan dil modelleri doğal olarak, ilgili alandaki veriler ile oluşturulur. Belli bir alanı kapsayacak biçimde oluşturulan BDM'leri boyut olarak genel BDM'lerinden daha küçük olurlar. Ayrıca, bu veri kümesi üzerinde ince ayar yapılarak daha başarılı sonuçlar üretirler.

5.2 Büyük Dil Modellerinin Evrimi

Yapay zeka (YZ) alanında, geçmişte yapılmış olan çalışmalar verileri işlemek ve analiz etmek amaçlı iken yakın zamandaki çalışmalar anlamak ve doğal dilde tümce kurmak üzerinedir. Böylece Algılayıcı ve Üretken YZ çalışmaları ağırlık kazanmıştır. Günümüzdeki BDM'lerini anlayabilmek için büyük dil modellerinden önceki DDİ modellerini gözden geçirmekte yarar vardır.

1. **Dönüştürücü Kullanmayan DDİ:** Bu çalışmalarda DDİ görevlerini yerine getirmek için makine öğrenmesi yöntemleri yerine insan tarafından işaretlenen modeller kullanılmıştır. Doğal olarak tüm kuralların bilindiği ve modele işlendiği varsayılır. Bu modeller, metin sınıflandırması çalışmaları için yeterli olmuştur. Makine çevirisi uygulamalarında, kaynak ve hedef dil aynı dil ailesinden ise bu modeller yeterince başarılı olmuşlardır ancak farklı dil ailesinde yer alan diller arasındaki çevirilerde başarılı olamamışlardır. Bilinmeyen kurallar söz konusu olduğunda başarısız olurlar. Varlığı bilinmeyen kuralların neden olduğu sorunlar

çözebilmek amacıyla RNN'ler ve LSTM'ler gibi sinir ağılarıyla çözülmeye çalışılmıştır. RNN'ler ve LSTM'ler önceki veriler üzerinden çıkarımlarda bulunarak ya da önceki sonuçları belleyerek tahmin ve sınıflandırma yapabiliyordu. RNN'ler ve LSTM'lerin tahminler kısa geçmişteki verilere dayandığından uzun metinler için tahmin yapamazlardı.

2. Dönüştürücü Kullanan DDİ: Dönüştürücü mimarisi 2017'de önerildi ve kullanılmaya başlandı. Dönüştürücüler, o zamana kadar kullanılan RNN ve LSTM'lere oranla daha iyi genelleme yapabiliyor daha fazla bağlam yakalayabiliyor ve daha fazla veriyi işleyebiliyordu. Dönüştürücü kullanımı ile DDİ modellerinin daha uzun veri dizilerini anlaması ve çok daha görev yapabilmesine olanak sağlandı. Dönüştürücü kullanan DDİ uygulamaları ilk aşamada araştırmacılar tarafından kullanılmaya başlandı; ticari olarak alanda uygulanmadılar.

3. BDM Kullanan DDİ: Büyük Dil Modellerini kullanan DDİ'ler 2020'de OpenAI'nin GPT-3'ün tanıtılması ile dikkat çekmeye başladı. GPT-3 ve benzeri BDM'leri, önceki dil modellerine göre daha doğru ve kapsamlı sonuçlar üretebiliyordu. Bunun nedeni kullandıkları veri kümesinin çok büyük olması ve bu modelin eğitilebilmesidir. BDM'leri çeşitli DDİ işlevlerini yalnızca doğal dil istemlerini kullanarak çözebilen teknik olmayan kullanıcılar için kolay erişilebilir hale getirdi.

Büyük Dil Modeline geçişin aşamalı biçimde olduğu yukarıdaki açıklamalardan anlaşılmaktadır. Bu gelişmelerde sinir ağlarının özellikle derin öğrenmenin çok etkin olduğu bir gerçektir. Dönüştürücüler bu yöntemler kullanılarak gerçekleştirilebilmiştir. Bu teknik ve yöntemlerin öğrenmekte yarar vardır.

5.3 Yapay Sinir Ağları

Yapay Sinir ağları YSA (SA) (NN) ya da kısaca sinir ağları, insan beyninin çalışma ilkesine temel alarak modellenmiş makine öğrenmesi algoritmalarıdır. İnsan beynine benzetilerek sinir ağları düğümlerden (nöron) oluşur ve bunlar girdiyi işlemekten çıktı üretmeye kadar tüm işlevleri yürütürler. Her sinir ağında en az üç katman bulunur:

- **Giriş katmanı:** Verileri kabul eder ve ağın geri kalanına iletir.
- **Gizli katman veya birden fazla gizli katman:** SA'dan beklenen işlevleri yerine getirir. Bu işlevler, verileri tanımlamayı veya sınıflandırmayı, yeni veri üretmeyi ve belirlenmiş olan DDİ işlevini yerine getiren işlevlerdir.
- **Çıkış katmanı:** Girdiye dayalı bir tahmin veya sınıflandırma sonucunu üretir.

RNN'ler ve LSTM'ler gibi sinir ağlarının, önceki kural tabanlı yöntemlere göre en büyük üstünlükleri, çok az ya da hiç insan katkısı olmadan kendi başlarına

öğrenebilmeleri idi. Bu yeni yöntemler kuralları öğrenip daha sonra verilere uygulamak yerine, kendi kurallarını oluşturmak için verileri analiz edebiliyorlardır. Çalışma biçimleri aşağıdaki örnekte görülmektedir:

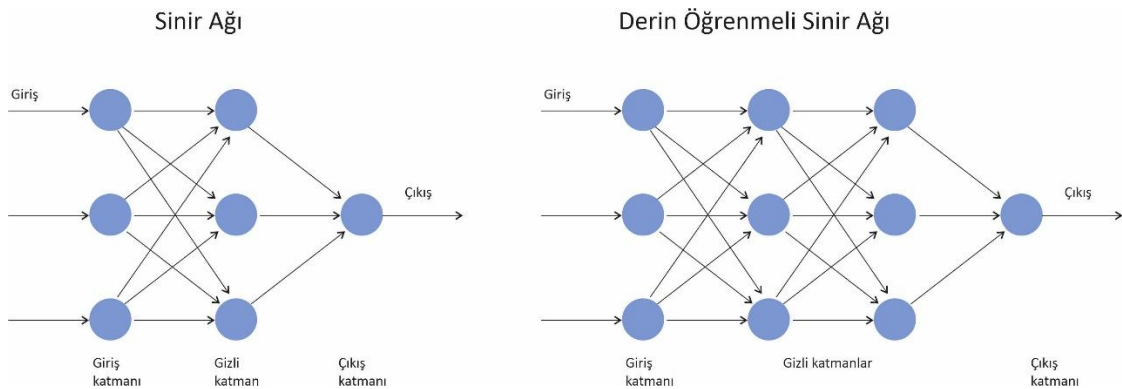
“kedi,” “kediler,” kuş,” “kuşlar”

Bu örnekte görülen lokmalar analiz ettikten sonra, model aşağıdaki biçimde modeli eğitir:

Çoğul adlar “-lar” eki alırlar.

Model daha sonra bu modeli kullanarak başka adların çoğullarını bulabilir veya üretebilir. Örneğin, “masa” dan “masalar”, “ev” den “evler”i türetebilir. Benzer biçimde diğer dil bilgisi kurallarını çıkarabilir. Bu tür özellik çıkarma, diğer adıyla gösterim çıkarma Türkçe gibi kurallı diller için çok kullanışlıdır. Sinir ağları ne kadar çok nörona sahipse, o kadar karmaşık gösterimler çıkarabilirler. Bu nedenle, BDM’leri birden fazla gizli katmana ve dolayısıyla daha fazla nörona sahip derin öğrenme sinir ağlarını kullanır [7].

Şekil-4’te tek katmanlı bir sinir ağı ile derin öğrenme sinir ağı çizilmiştir.



Şekil-4: Tek Katmanlı ve Derin Öğrenmeli Sinir Ağı

Günümüzde çok sayıda gizli katman içeren sinir ağlarının kullanılabilmesi Grafik İşlemci Birimlerinin (GPU) GPU’ların 1999’dan sonra kullanılmaya başlanması ve 2012 de yaygınlaşması ile sağlanmıştır. GPU’lar, sinir ağındaki bulunan katmanlar ile işlemleri paralel biçimde işleyebilme yeteneğine sahiptirler. Böylece işlemlerin süresi kısalmıştır. GPU’ların kullanımındaki başarının bir sonucu olarak dönüştürücüler gündeme gelmiştir.

5.4 Dönüştürücüler

Daha önce açıklandığı gibi RNN’ler ve LSTM’ler diller arası çevirilerde yetersiz kalıyorlardı. Bu eksiklikleri gidermek üzere dönüştürücü adı verilen yeni, özel sinir

ağları tasarlanmıştır. Dönüştürücüler ilk olarak 2017 yılında Vaswani ve arkadaşları tarafından "**Dikkat İhtiyacınız Olan Tek Şey**" başlıklı bir makalede tanıtılmıştır [5]. Bu makalede, dönüştürücülerin temel bileşeni haline gelecek olan “**dikkat mekanizması**”na atıfta bulunmuştur. Önerilen yöntem tekrarlama ve evrişimleri ortadan kaldırmıştır. Dikkat mekanizmaları, modellerin girdiyi işlerken, girdinin belirli bölümlerine özellikle dikkat etmesini isteyerek yetersiz bağlam işleme sorununu çözmüştür. Tüm bağlamı bir adımda yakalamak yerine, belirli bir görevle ilgili en önemli lokmalara odaklandırılmıştır.

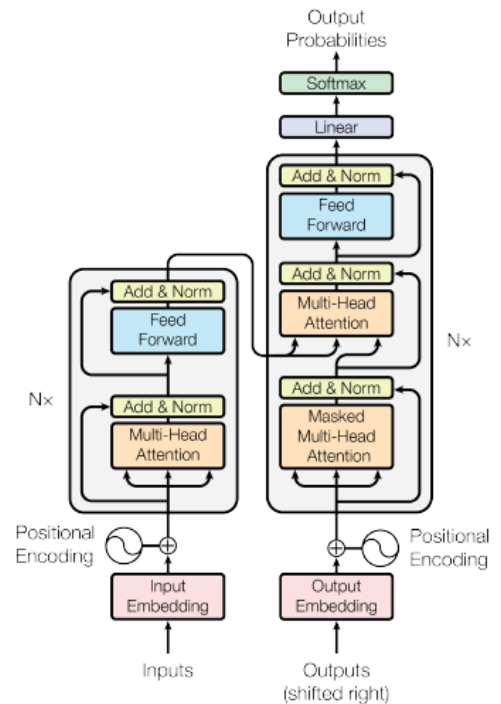
Dikkat mekanizmalarına dayanan öz-dikkat katmanları daha önce girilen her sözcük için dikkat ağırlıklarını hesaplar. Dikkat ağırlıkları her bir lokmanın önemini temsil eder, bu nedenle bir lokmaya ne kadar fazla ağırlık atanırsa o kadar önemli kabul edilir. Örneğin,

Eren’in bir kedisi var.

Tümcesinde dikkat mekanizması "*kedi*" lokmasına "*bir*" lokmasından daha fazla ağırlık verebilir. Bunun nedeni “*kedi*” lokmasının sonuca etkisinin “*bir*” lokmasının etkisinden büyük olmasıdır. Her bir lokmanın ne kadar önemli olduğunu belirlemek için, öz-dikkat katmanları bir metindeki bir lokmanın diğer lokmalarla ilişkilerini değerlendirir. Bir lokmanın diğer lokmalarla çok sayıda ilişkisi varsa, o lokma önemli kabul edilir.

LSTM'ler ve RNN'ler girdiyi ardışık olarak işledikleri için girişe karşılık çıkışın üretilmesi uzun zaman alır. Buna karşın dönüştürücüler paralel çalışırlar, dolayısıyla işlem süresi çok kısadır. Ancak, LSTM'ler ve RNN'ler yavaş olmalarına karşın, lokmaları ardışık işlemleri nedeniyle hangi lokmanın önce ya da sonra geldiğini, yani lokmaların ya da sözcüklerin sırasını bilirlir. Dönüştürücülerin böyle bir özellikleri yoktur. Bu eksikliğin üstesinden gelmek için dönüştürücüler, lokmaları konum bilgilerine yardımcı olan konumsal kodlamalar kullanır. Konumsal kodlamalar, her bir lokmayla ilişkilendirilen ek girdiler veya vektörlerdir.

Şekil-5'te dönüştürücü mimarisi gösterilmiştir. Şekil “Dikkat İhtiyacınız Olan Tek Şey” bildirisiinden alınmıştır.



Şekil-5: Dönüştürücü mimarisi

Araştırmacılar ve uygulamacılar dönüştürücülerini uygulamaya başlamışlardır. Uygulamalardan biri Google tarafından 2018 yılında BERT' adıyla duyurulmuştur.

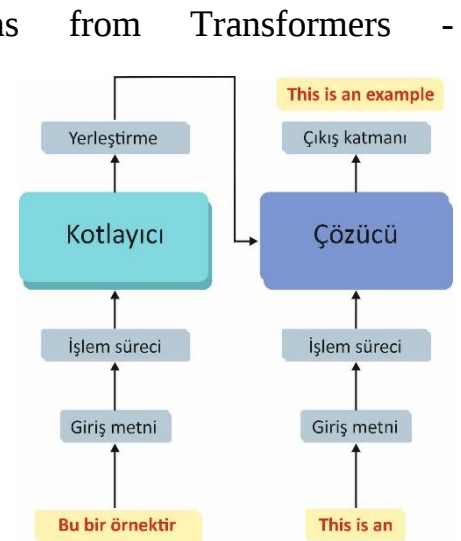
Dil çevirisi için derin öğrenme modeli olan orijinal dönüştürücü mimarisinin basitleştirilmiş bir çizimi Şekil-6'da görülmektedir. Şekildeki dönüştürücü iki bölümden oluşmaktadır: Birinci bölüm giriş metnini işleyen ve metnin bir yerleştirme gösterimini (farklı boyutlarda birçok farklı etkenin sayısal gösterimi) üreten bir kodlayıcı, ikinci bölüm çevrilen metni kod çözücünün her seferinde bir kelime üretmek için kullanabileceği bölümdür. Bu şekil, kod çözücünün yalnızca son sözcüğü üretmesi gereken çeviri sürecinin son aşamasını gösterir.

5.4.1 BERT

BERT (Bidirectional Encoder Representations from Transformers - Dönüştürücülerden Çift Yönlü Kodlayıcı Gösterimleri) ilk olarak tanıtılan dönüştürücü tabanlı dil modellerinden biridir [8]. Maskeli dil modelidir (MLM), yani maskeli lokmalar içeren tümcelerle eğitilmiştir. Modelin, maskeli lokmayı, çevresindeki bağlamı dikkate alarak tahmin etmesi amaçlanmıştır. Yapıyı açıklamak için aşağıdaki örnek tümcenin giriş olarak verildiğini düşünelim:

Askıda bir *şapka* var.

Bu örnek tümcede “*şapka*” sözcüğünün maskelendiğini ve dönüştürücünün bulmasını isteyelim. Şapka sözcüğünün öncesinde “*bir*” ve sonrasında “*var*” lokmaları bulunuyor. Dönüştürücü, bu tümceden önceki tümcelere ve bağlamı inceleyerek maskelenen sözcüğün “*şapka*” olduğunu bulur. Tek yönlü çalışan modellerde maskelenmiş lokmayı bulmak daha zordur. BERT, çift yönlü çalışan ilk dönüştürücü modelidir.



Şekil-6: Basitleştirilmiş
dönüştürücü mimarisi

6. Diğer Büyük Dil Modelleri

Dönüştürücüler kullanılmaya başlamasıyla farklı özellikte BDM'leri oluşturmaya başlanmıştır. Çizelge-2'de bazı önemli BDM'leri verilmiştir.

Çizelge-2: Bazı Önemli BDM'leri

Model	Üretici Firma	Yıl	Kısa açıklama
GPT-2	OpenA1	2019	İnsan benzeri metinler üretmek ve dil çevirisi, özetleme ve soru yanıtlama gibi DDİ işlevlerini gerçekleştirmek için tasarlanmıştır.
RoBERTa	Facebook	2019	BERT temelli bir BDM'dir. Daha büyük bir veri kümesi üzerinde eğitilmiş, daha verimli bir eğitim yöntemi kullanılmıştır. Bu nedenle başarıyı yükseltilmiştir.

DeBERTa	Microsoft	2020	BERT temelli olan bu uygulamada kodlayıcı ve kod çözücü birbirinden ayrılarak DDİ işlevlerinin başarımı yükseltilmiştir.
GPT-3	OpenAI	2020	Daha büyük bir veri kümesi üzerinde eğitilen ve daha başarılı çıktılar üretebilen GPT-2'nin bir üst modelidir.

6.1 GPT ve BERT Karşılaştırması

GPT (Generative Pre-trained Transformer - Üretken Ön-eğitilmiş Dönüştürücü) ve BERT yaygın olarak kullanılan ancak mimarileri farklı BDM'lerdir. GPT, bir tümce içindeki bir sözcük verildiğinde bir sonraki sözcüğü tahmin etmek üzere eğitilen bir modeldir. Bu ön eğitim sayesinde tutarlı ve akıcı tümceler üretebilmektedir. Buna karşın, BERT, tümceleri veya lokmaları duygu analizi, adlandırılmış varlık tanıma ve metin sınıflandırması gibi farklı ulamlara ayırmak üzere eğitilmiş bir modeldir. Bir sözcüğün anlamını çıkarmak için tümcenin önceki ve sonraki bağlamlarını dikkate alan iki yönlü bir modeldir. Duygu analizi ve soru yanıtlama gibi uygulamalarda başarılıdır.

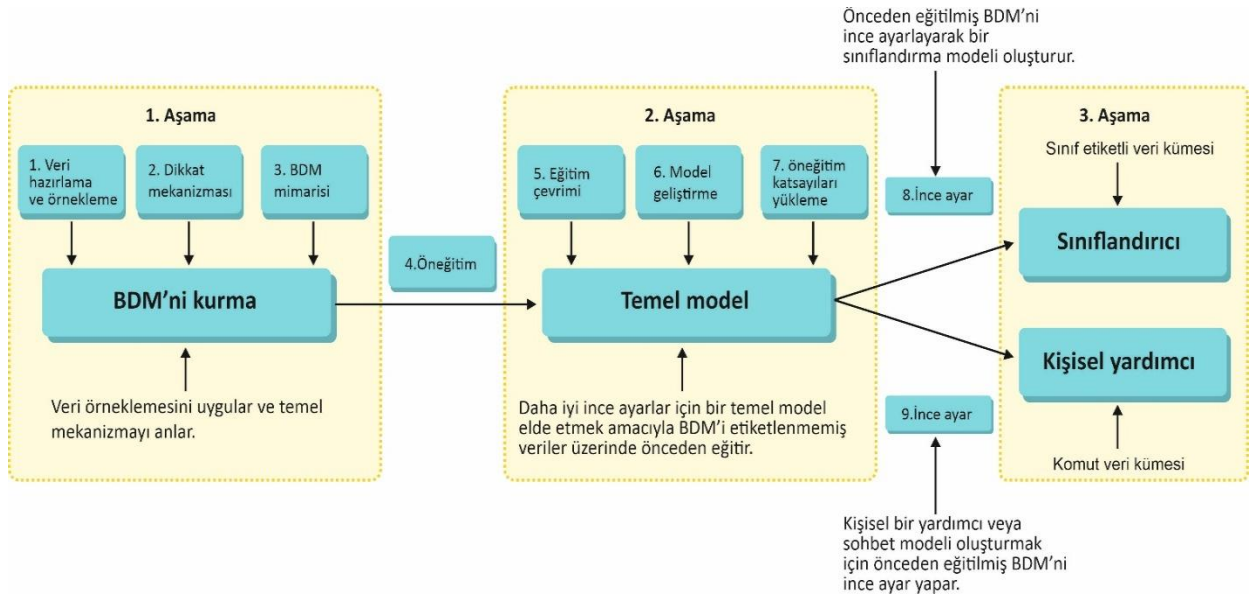
GPT'nin dönüştürücü mimarisi tek yönlü olmasına karşın BERT'inki çift yönlüdür. Bunun sonucu olarak bir sözcüğü tahmin ederken yalnızca önceki sözcüğe bakarken, BERT önceki ve sonraki sözcüklere bakarak tahminde bulunur.

6.2 BDM Oluşturma

Sıfırdan büyük dil modelleri oluşturmak, özellikle temel işi yapay zeka veya DDİ olmayan işletmeler için mantıklı değildir. BDM oluşturma süreci zaman alıcı ve emek gerektireceğinden işletmelerin mevcut BDM'leri gereksinimlerine göre özelleştirmeleri salık verilir. Mevcut BDM'leri özelleştirmek için aşağıda açıklanan üç adımın izlenmesi yerinde olur [9,10].

- 1. İşletmenin gereksinimlerini karşılayabilecek modeli araştırmak:** Bu araştırma sırasında modelin boyutu, veri kümesi ve sağlayıcısını araştırmak gerekir.
- 2. Modelin ince ayarını yapmak:** Temel BDM'lerinin işletmenin gereksinimlerine ve kullanım biçimine göre ince ayarlarının yapılması gerekir. Örneğin, metin sınıflandırmaya, duygu analizine veya yasal mevzuata uyumu sağlamak üzere ince ayar yapılır.
- 3. Modeli en iyi biçime getirmek:** Genel amaçlı olarak hazırlanmış olan modellerde işletme için gereksiz olabilecek veri kümeleri veya yetenekler bulunabilir. Bunlar modelden çıkarılarak bilgisayarın iş yükü azaltılabilir.

Bir BDM'nin oluşturulması Şekil-7'da ayrıntılı biçimde gösterilmiştir.



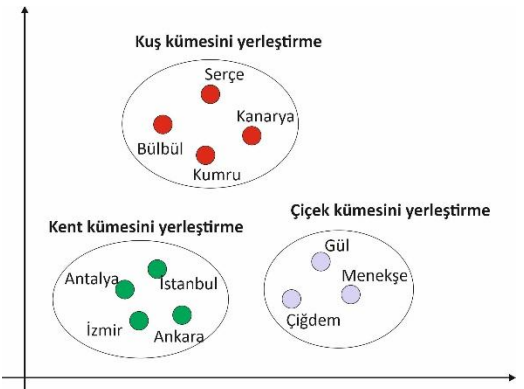
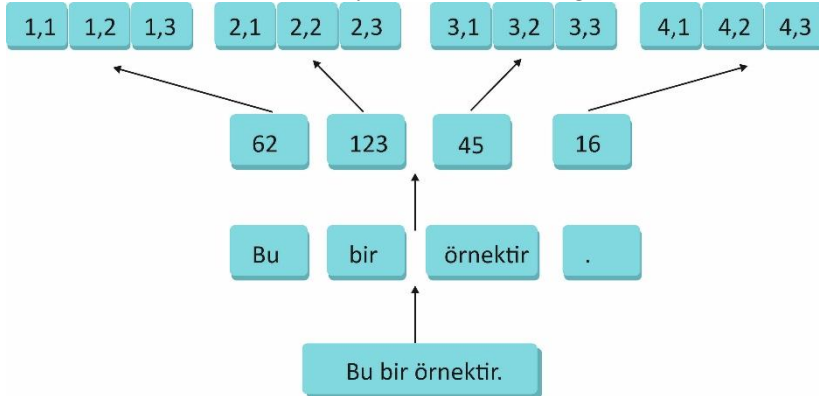
Şekil-7: Bir BDM'nin geliştirilme aşamaları

Şekil-7’de görülen işlevler aşağıda teker teker açıklanmıştır.

Veri Hazırlama ve Örnekleme

Bu ilk aşamanın görevi, girilen metnin lokmalara ayrılması, lokmaların sayılar ile temsili ve yerleştirilmesidir. Lokmaların vektör biçiminde temsili yerleştirme (embedding) olarak adlandırılır. Yerleştirme için farklı algoritmalar kullanılmaktadır. Bu algoritmalarından biri Word2Vec olarak anılır. Word2Vec’in temel fikri benzer bağlamlarda görünen sözcüklerin benzer anlamlara sahip olma eğiliminde olmasıdır. Şekil-8’de bu fikir tanıtılmıştır.

Yerleştirme öncesi metin lokmalarına ayrılır. Lokma, metin içinde görünen sözcükler ve noktalama işaretleridir. Tüm metinde geçen sözcükler alfabetik olarak sıralanır; tekrarlananların sırası kaydırılır. Bu aşamadan sonra her sözcüğe sıra numarası atanır. Bir sonraki aşamada her bir lokma için vektör ataması (yerleştirme) yapılır. Şekil-9’da basitleştirilmiş lokmalama ve yerleştirme süreci gösterilmiştir.



Şekil-8: İki boyutta yerleşme örneği

Dikkat Mekanizması

Bir metin içinde bulunan her bir sözcüğün metin içindeki önemi ya da ağırlığını belirtir. Dikkat mekanizmasının türleri şunlardır:

- Basit öz dikkat
- Öz dikkat
- Nedensel dikkat
- Çok başlı dikkat

BDM Mimarisi

GPT gibi BDM'leri, her seferinde bir sözcük (veya lokma) olmak üzere yeni metin üretmek üzere tasarlanmış büyük derin sinir ağı mimarileridir. Ancak, büyük boyutlarına karşın, model mimarisi çok karmaşık değildir. Çünkü bileşenlerinin çoğu tekrarlanır. Şekil-10'da GPT benzeri BDM'nin mimarisi çizilmiştir.

Derin öğrenme ve GPT gibi BDM'lerinde, "parametreler" terimi modelin eğitilebilir ağırlıklarını ifade eder. Bu ağırlıklar aslında, belirli bir kayıp fonksiyonunu en aza indirmek için eğitim süreci sırasında ayarlanan ve en iyi biçime getirilen modelin iç değişkenleridir.

Ön Eğitim, Eğitim Çevrimi, Model Geliştirme ve Ön Eğitim Katsayılarını Yükleme

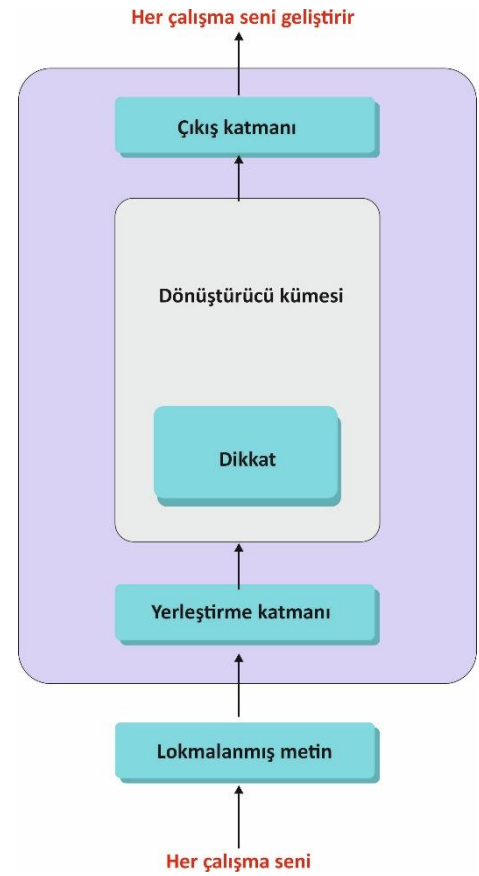
Şeki-7'deki 4, 5, 6, 7. Adımlar BDM'nin eğitilmesi ile ilgilidir. Bu aşamalarda, lokmalar arasındaki ağırlık katsayıları hesaplanır ve sonuçlar tekrar güncellenir. Bu işlemlerin sonunda, DDİ'ye yakın tümce oluşturulur.

İnce Ayar

BDM'nin araştırmannın ya da işletmenin gereksinimlerine göre ayıklanması işlemine ince ayar denilmektedir.

Kaynakça

- [1] K. S. Jones, Natural Language Processing: A Historical Review, Computational Linguistics, 1994
- [2] E. Adalı, Türkçe Doğal Dil İşleme, Akçağ Yayınevi, 2020, ISBN 97860553425519
- [3] F.M.Kom, Natural Language Processing Introduction, theory and application, Universitas Gadjah Mada



Şekil-10: BDM Mimarisi

- [4] A. Chockalingam, A. Patel, S. Verma, T. Yeung, A beginner's Guide to Large Language Models, NVIDIA Corporation. 2023
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, 31st Conference on Neural Information Processing Systems (NIPS 2017)
- [6] S.J.D. Prince, Understanding Deep Learning, The MIT Press, March 26, 2025
- [7] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, A. Mian, A Comprehensive Overview of Large Language Models, arXiv:2307.06435v10 [cs.CL] 17 Oct 2024
- [8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, arXiv:1810.04805v2 [cs.CL] 24 May 2019
- [9] J. Kunz, Understanding Large Language Models, Linköping Studies in Science and Technology Dissertation No. 2364
- [10] S. Raschka, Build a Large Language Model From Scratch, Manning Publications Co. 2025, ISBN: 9781633437166