

Büyük Dil Modelleri

Prof.Dr.Hüseyin ÇAKIR
Gazi Üniversitesi

Büyük Dil Modelleri (Large Language Models – LLM)

Yapay Zeka ve Uygulamaları - Copyright © 2026

LLM Nedir?



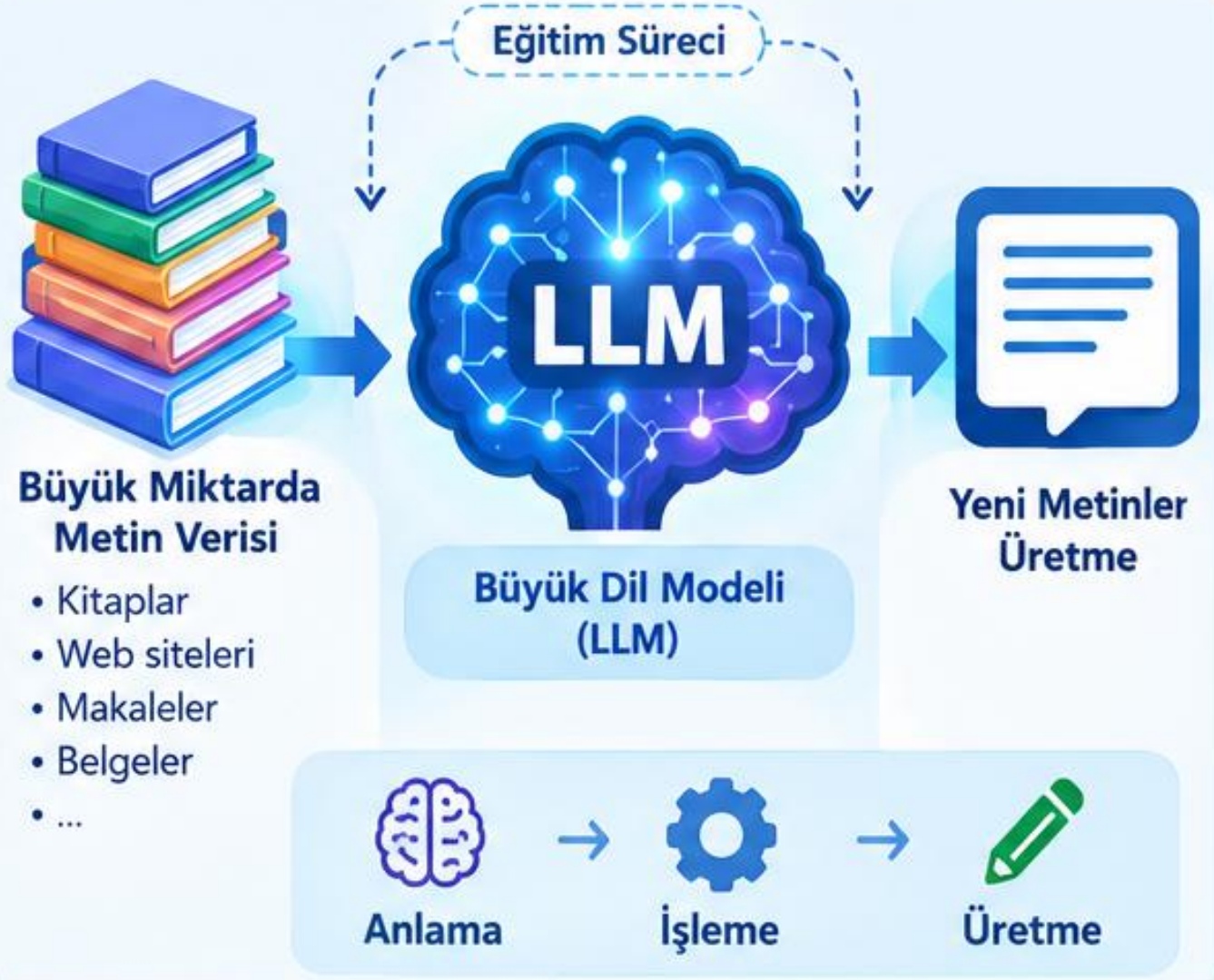
LLM (Large Language Model – Büyük Dil Modeli), **çok büyük miktarda metin verisi** kullanılarak eğitilmiş ve **insan dilini işleyebilen, anlayabilen** ve **yeni metinler üretebilen** yapay zekâ modelidir.



Büyük Dil Modelleri

Büyük Dil Modelleri (Large Language Models – LLM), çok büyük miktarda metinsel veri üzerinde eğitilerek **insan dilini anlamaya, işlemeye ve yeni metinler üretmeye** yönelik yapay zekâ modelleridir.

LLM'ler günümüzde **üretken yapay zekânın temel teknolojilerinden biri** olarak kabul edilir.



Yapay Zeka ve Uygulamaları - Copyright © 2026

LLM'lerin Özellikleri

Doğal dil anlama
Metni doğru şekilde kavrar.

Bağlamı yorumlama
Cümlelerin anlamını içinde bulunduğu bağlama göre değerlendirir.

Metin üretme
Tutarlı, anlamlı ve amaca uygun yeni metinler oluşturur.

Çok yönlü kullanım
Soru-cevap, özetleme, çeviri, kod yazma, yaratıcı içerik üretme vb.

LLM Nedir?

LLM, insan dilindeki örüntüleri öğrenerek verilen bir metnin devamını tahmin edebilen, soruları yanıtlayabilen ve yeni metinler oluşturabilen büyük ölçekli yapay zekâ modelidir.



Büyük Veri

İnternet, kitaplar, makaleler, web sayfaları gibi milyonlarca metin verisiyle eğitilir.

→



LLM

Bu verilerden dilin örüntülerini öğrenir.

→



Metin Üretimi

Öğrendiği örüntülere göre en uygun kelime ve cümleleri tahmin eder.

→



Sonuç

Soruları yanıtlar, açıklamalar yapar, yeni metinler oluşturur.

Büyük Dil Modelleri

Kullanıcı:

"Yapay zekânın eğitimde kullanım alanlarını açıkla."

LLM Yanıtı:

Yapay zekâ, eğitimde birçok alanda kullanılmaktadır.
Başlıca kullanım alanları şunlardır:

1. **Kişiselleştirilmiş öğrenme**: Öğrencinin seviyesine ve hızına göre içerik sunar.
2. **Akıllı tutorlar**: Soruları yanıtlar, rehberlik eder.
3. **Otomatik değerlendirme**: Sınav ve ödevleri hızlıca değerlendirir.
4. **İçerik üretimi**: Ders materyalleri, özetler ve etkinlikler oluşturur.
5. **Erişilebilirlik**: Dil, engel veya coğrafi sınırlamaları azaltır.
6. **Veri analizi**: Öğrenme süreçlerini takip edip iyileştirme önerileri sunar.

Bu sayede daha etkili, esnek ve kapsayıcı bir eğitim ortamı oluşturulabilir."

Eğitimde Başlıca Kullanım Alanları

Kişiselleştirilmiş Öğrenme
Öğrencinin ihtiyaçlarına göre uyarlanmış içerik ve öğrenme yolları.

Akıllı Tutorlar
Soru-cevap, rehberlik ve anlık destek.

Otomatik Değerlendirme
Sınav, ödev ve performans analizi.

İçerik Üretimi
Ders notları, özetler, örnekler ve etkinlikler.

Erişilebilirlik
Engelli bireyler ve farklı diller için destek.

Veri Analizi
Öğrenme süreci takibi ve iyileştirme önerileri.

LLM Nasıl Çalışır?

Büyük verilerle eğitilen modeller, kullanıcıdan gelen isteği (prompt) anlayarak en olası ve anlamlı yanıtı üretir.

Büyük Dil Modelleri

1 Büyük Veri Kümeleri



İnternet, kitaplar, makaleler, web siteleri, forumlar ve daha birçok kaynaktan toplanan çok büyük metin kümeleri kullanılır.

2 Model Eğitimi



Bu veriler, derin öğrenme teknikleriyle eğitilen bir sinir ağı modeline verilir. Model, metin içindeki yapı ve ilişkileri öğrenir.

3 Dil Örüntülerinin Öğrenilmesi



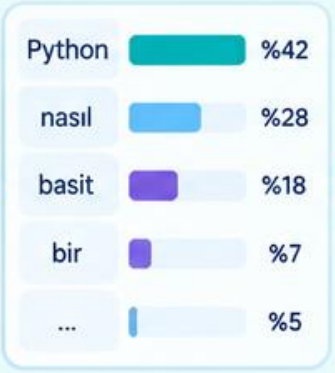
Model, kelimeler arasındaki ilişkileri, ifade kalıplarını ve dilin yapısını öğrenir.
Bu süreçte bağlam, anlam ve gramer kuralları dahil çok sayıda örüntü kazanır.

4 Kullanıcı Promptu



Kullanıcının yazdığı soru veya talep (prompt), modele gönderilir. Model bu metni analiz ederek bağlamını anlamaya çalışır.

5 Modelin Olasılıkları Hesaplaması



Model, her bir kelime veya kelime grubunun gelme olasılığını hesaplar. En yüksek olasılığa sahip sözcükler seçilerek yanıt adım adım oluşturulur.

6 Yanıtın Oluşturulması



Hesaplanan olasılıklara göre en uygun kelimeler birleştirilir ve kullanıcıya anlamlı, tutarlı ve doğal bir yanıt sunulur.

LLM'ler Neler Yapabilir?

Büyük Dil Modelleri

Yetenek	Örnek
Metin üretimi	Makale, rapor, e-posta
Soru-cevap	Soruları yanıtlamak
Özetleme	Uzun metni kısaltmak
Çeviri	Türkçe → İngilizce
Metin dönüştürme	Akademik metni sadeleştirme
Kod üretimi	Python, JavaScript vb.
Akıl yürütme	Problemleri çözmeye çalışma
Eğitim	Konu anlatımı ve soru hazırlama
Veri analizi	Verileri yorumlama
Kişiselleştirme	Öğrenciye uygun içerik oluşturma

Önemli LLM Örnekleri

GPT	Gemini	Claude	Llama	Mistral	Qwen
OpenAI	Google	Anthropic	Meta	Mistral AI	Alibaba
Metin, kod, görsel ve çoklu modalite desteğiyle güçlü bir genel amaçlı LLM'dir.	Google'ın çok modelli yapay zekâ platformudur. Metin, görsel, video ve daha fazlasını işleyebilir.	Güvenli, etik ve insan odaklı yapay zekâ yaklaşımıyla bilinen güçlü bir LLM'dir.	Açık kaynaklı büyük dil modeli ailesidir. Geliştiriciler için esnek ve özgür bir çözündür.	Yüksek performanslı, hızlı ve verimli büyük dil modelleri geliştiren Fransız yapay zekâ şirketidir.	Çin merkezli Alibaba tarafından geliştirilen güçlü ve çok dilli büyük dil modelidir.

LLM ile Chatbot Aynı Şey midir?



LLM =
Yapay zekâ modeli

Büyük veri üzerinde eğitilmiş, metin, görsel, ses gibi verileri anlayıp üretebilen ileri seviye yapay zekâ modelidir.

≠



Chatbot =
Kullanıcının modeliyle etkileşmesini sağlayan uygulama

LLM'yi bir arayüz üzerinden kullanır ve kullanıcı ile sohbet edebilmesini sağlar.

Örneğin:

GPT modeli (LLM) + **Sohbet arayüzü** (kullanıcı ile etkileşim) + **Araçlar** (ör. internet, eklentiler) + **Sistem talimatları** (davranış ve kurallar) = **ChatGPT** benzeri bir uygulama

Yapay Zeka ve Uygulamaları - Copyright © 2026

LLM'lerin Sınırlılıkları

LLM'ler güçlü olmalarına rağmen:

Yapay Zeka ve Uygulamaları - Copyright © 2026

Büyük Dil Modelleri



1 Yanlış bilgi üretebilir.



2 Halüsinasyon (hallucination) oluşturabilir.



3 Kaynakları yanlış aktarabilir veya var olmayan kaynaklar üretebilir.



4 Güncel bilgiye erişimi kullanılan sisteme bağlıdır.



5 Önyargılı çıktılar üretebilir.



6 Kullanıcının verdiği bilgiler açısından gizlilik ve güvenlik riskleri oluşturabilir.


LLM
Eğitimde LLM Kullanımı
LLM'ler eğitimde öğretmen ve öğrencilere
farklı alanlarda destek sağlayabilir.

Büyük Dil Modelleri

Öğretmen için:



Ders planı → Soru bankası → Ders materyali → Rubrik → Geri bildirim

Öğrenci için:



Konu anlatımı → Örnek → Alıştırma → Soru-cevap → Kişiselleştirilmiş öğrenme

Büyük Dil Modelleri

Nasıl Çalışır?

Metin Verisi

Kitaplar, makaleler, web içerikleri, kodlar gibi büyük veri kümeleri toplanır.

Eğitim Süreci

Veriler temizlenir, düzenlenir ve model eğitilir.

Örüntüleri Öğrenme

Model, dildeki örüntüleri ve bağlam ilişkilerini öğrenir.

Yanıt Üretimi

Verilen metne uygun yeni ve anlamlı metinler oluşturur.

En basit tanımıyla

LLM, insan dilindeki örüntüleri öğrenerek verilen metne uygun yanıtlar oluşturabilen büyük ölçekli yapay zekâ modelidir.



LLM'nin Temel Özellikleri

Yapay Zeka ve Uygulamaları - Copyright © 2026

Büyük Dil Modelleri

1 Metin oluşturabilir.



İstenen konuda özgün ve anlamlı metinler yazabilir.

"Güneş enerjisinin avantajlarını anlatan bir paragraf yaz."

2 Soruları yanıtlayabilir.



Kullanıcının sorularını anlayarak doğru ve ilgili yanıtlar verebilir.

"Türkiye'nin başkenti neresidir?"

3 Metinleri özetleyebilir.



Uzun metinleri kısa ve anlaşılır biçimde özetleyebilir.

"Bu makalenin kısa bir özetini çıkar."

4 Çeviri yapabilir.

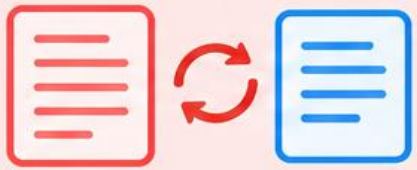


Farklı diller arasında çeviri gerçekleştirebilir.

"Bu cümleyi İngilizceye çevir: 'Eğitim, daha iyi bir gelecek için önemlidir.'"



5 Metinleri farklı biçimlerde yeniden yazabilir.



Aynı içeriği farklı üslup, ton ve yapıda ifade edebilir.

"Bu metni daha resmi bir dille yeniden yaz."

6 Kod oluşturabilir ve açıklayabilir.



Programlama diliyle kod yazabilir, kodları açıklayabilir ve hata ayıklamada yardımcı olabilir.

"Python'da bir dizinin ortalamasını hesaplayan kod yaz ve açıkla."

7 Eğitim materyalleri hazırlayabilir.



Ders notları, soru bankaları, çalışma planları ve görsel materyaller oluşturabilir.

"Yapay zekâ konulu 5 soruluk çalışma soruları hazırla."

8 Karmaşık problemlere yönelik çözüm üretebilir.



Analiz yaparak çok adımlı problemlerde mantıklı ve uygulanabilir çözümler sunabilir.

"Daha verimli bir öğrenme planı nasıl oluşturulabilir?"

LLM Nasıl Öğrenir?

Büyük miktarda metin verisini analiz ederek dilin yapısını, örüntülerini ve ilişkilerini öğrenir. Bu sayede, yeni sorulara uygun yanıtlar üretebilir.

Büyük Dil Modelleri

1 Çok büyük metin veri setleri



İnternet, kitaplar, makaleler, forumlar ve daha birçok kaynaktan toplanan devasa metin verileri kullanılır.

2 Model eğitimi



Veriler, derin öğrenme teknikleriyle işlenir ve model, metindeki ilişkileri ve yapıyı öğrenir.

3 Dil örüntülerinin öğrenilmesi



Model, kelimeler arasındaki ilişkileri, cümle yapılarını ve dilin örüntülerini öğrenir.

4 Kullanıcı promptu



Kullanıcı, bir soru veya talep (prompt) gönderir. Model, bu metni analiz ederek bağlamını anlar.

5 Uygun yanıtın oluşturulması



Model, öğrendiği dil örüntülerine dayanarak en uygun ve anlamlı yanıtı üretir.

Örnek LLM'ler

Farklı şirketler ve kuruluşlar tarafından geliştirilen popüler büyük dil modellerinden bazıları:

Büyük Dil Modelleri

Model	Geliştirici
GPT Genel amaçlı, çok yönlü dil modeli.	**OpenAI** (ABD)
Gemini Metin, görsel, video ve çok modlu yeteneklere sahip model.	**Google** (ABD)
Claude Güvenli, faydalı ve etik odaklı model.	**Anthropic** (ABD)
Llama Açık kaynaklı, güçlü ve esnek model.	**Meta** (ABD)
Mistral Verimli ve yüksek performanslı model.	**Mistral AI** (Fransa)
Qwen Çok dilli ve açık kaynaklı model.	**Alibaba** (Çin)

Model Eğitimi (Model Training)

Model eğitimi, bir yapay zekâ modelinin büyük miktardaki verilerden **örüntüleri, ilişkileri** ve **kuralları** öğrenmesini sağlayan süreçtir.

Veri Toplama



Kitaplar, makaleler, web siteleri, kodlar ve diğer kaynaklardan büyük miktarda veri toplanır.

Veri Hazırlama



Toplanan veriler temizlenir, düzenlenir ve modelin eğitimine uygun hale getirilir.

Model Eğitimi



Veriler, modelin parametrelerini ayarlamak üzere kullanılır. Model, verilerden örüntüleri öğrenir.

Test ve Değerlendirme



Eğitilen model, test verileriyle denenir ve performansı değerlendirilir.

Kullanıma Alma



Başarılı olan model, gerçek dünyada görev yapmak üzere kullanıma alınır.

Basitçe:

Model eğitimi = Yapay zekâya veriler üzerinden nasıl görev yapacağını öğretme sürecidir.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Model Eğitimi Nasıl Gerçekleşir?

Makine öğrenmesi modelleri, belirli adımlarla toplanan verilerden öğrenerek tahmin yapabilir hale gelir. Genel süreç şu şekilde düşünülebilir:

Yapay Zeka ve Uygulamaları - Copyright © 2026

Büyük Dil Modelleri

1 Veri Toplama



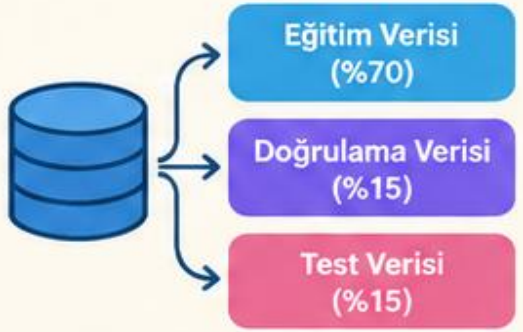
Modelin öğrenmesi için gerekli veriler, farklı kaynaklardan toplanır. (ör. veritabanları, dosyalar, web, sensörler vb.)

2 Veri Temizleme ve Hazırlama



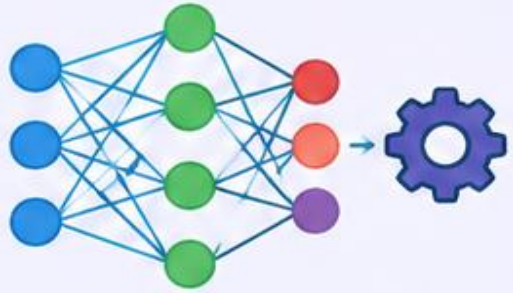
Toplanan veriler, hatalı, eksik veya aykırı değerlerden arındırılır. Gerekli dönüşümler yapılarak model için uygun hale getirilir.

3 Eğitim Verisinin Oluşturulması



Hazırlanan veri seti, modelin öğrenmesi ve genelleme yapabilmesi için eğitim, doğrulama ve test verisi olarak ayrılır.

4 Modelin Eğitilmesi



Seçilen algoritma ile eğitim verisi kullanılarak modelin parametreleri öğrenilir.
Bu aşamada model, verilerdeki örüntüleri keşfeder.

Büyük Dil Modelleri

5 Modelin Test Edilmesi



Eğitim sırasında öğrenilmeyen test verisi ile modelin tahminleri kontrol edilir. Modelin genelleme yapma yeteneği değerlendirilir.

6 Performans Değerlendirmesi



- Doğruluk (Accuracy)
- Kesinlik (Precision)
- Duyarlılık (Recall)
- F1 Skoru
- Kayıp (Loss)

Modelin başarısı, belirli metriklerle ölçülür. Gerekirse parametreler düzenlenerek iyileştirme yapılır.

7 Modelin Kullanıma Alınması



Test ve değerlendirme süreci tamamlanan model, gerçek ortamda kullanılmak üzere devreye alınır. Böylece, yeni verilerle çalışarak tahmin yapmaya başlar.

Kedi ve Köpek Sınıflandırma Örneği

Bir modele kedi ve köpek görsellerini ayırt etmeyi öğretelim.

Büyük Dil Modelleri

1 Eğitim Verileri

Modele binlerce kedi ve köpek görseli gösterilir.

Kedi Görselleri



...

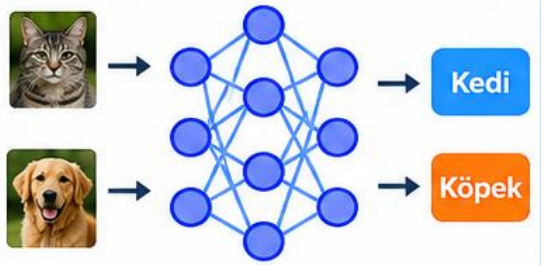
Köpek Görselleri



...

2 Öğrenme Süreci

Model, bu görsellerdeki özellikleri (renk, kulak şekli, yüz yapısı, tüy yapısı vb.) öğrenir.



Makine Öğrenmesi Modeli

3 Yeni Görsel

Daha sonra modele daha önce görmediği bir görsel verilir.

"Bu kedi mi, köpek mi?"



Bu kedi mi, köpek mi?

4 Tahmin

Model, öğrendiği örüntülere göre tahmin yapar.



Girdi Görseli



Trained Model
(Öğrendiği özelliklere göre analiz yapar)

Tahmin:

Kedi ✓

Güven Skoru

%92

LLM'lerde Model Eğitimi

Büyük Dil Modellerinde (LLM) süreç, çok büyük ölçekte veri ile gerçekleştirilir.
Model, milyonlarca sayıda metin üzerinde eğitilerek dilin yapısını, bağlamını ve örüntülerini öğrenir.

Büyük Dil Modelleri

1 Büyük Veri Toplama

Model, çok çeşitli ve büyük miktarda metin verisi ile eğitilir.



- Kitaplar
- Makaleler
- Web siteleri
- Kod depoları
- Forumlar, belgeler vb.

2 Tokenlaştırma

Metinler, modelin anlayabileceği küçük parçalara (token) ayrılır.

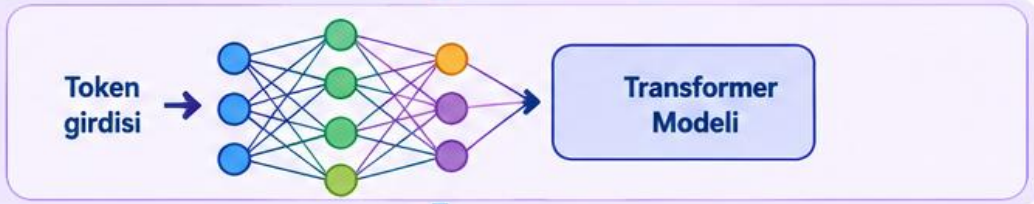


Her token, bir sayısal ID ile temsil edilir.

Örn: [3821, 721, 11, 4502, ...]

3 Model Eğitimi

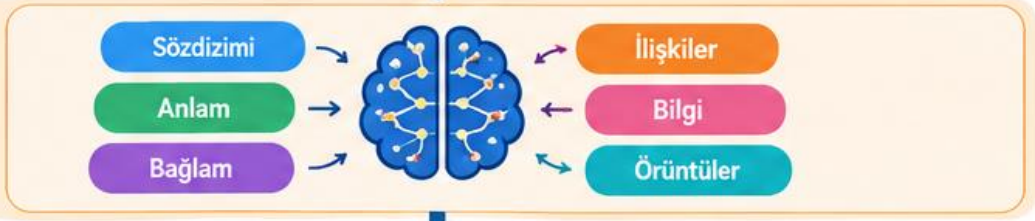
Token dizileri, devasa sinir ağı mimarileri (Transformer) ile işlenir.



- Dikkat (Attention) mekanizması
- Çok katmanlı ağ yapısı
- Milyarlarca parametre
- Yüksek hesaplama gücü (GPU/TPU)

4 Dil ve Bağlam Örüntülerinin Öğrenilmesi

Model, kelimeler arasındaki ilişkileri, bağlamı ve dilin yapısını öğrenir.



Bu süreçte model, metinlerdeki örüntüleri istatistiksel olarak öğrenir ve genelleme yapabilir.

5 Yeni Metin Oluşturabilme

Eğitimi tamamlanan model, öğrendiği örüntülere dayanarak yeni, tutarlı ve anlamlı metinler üretebilir.



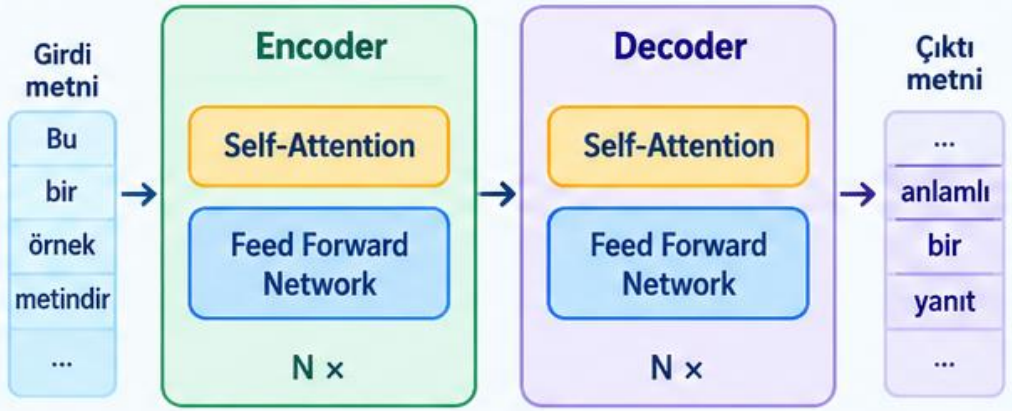
Model, öğrendiklerini kullanarak sorulara yanıt verir, özetler, çeviri yapar, kod yazar ve daha fazlasını gerçekleştirebilir.

LLM'lerde Önemli Bir Kavram: Transformer

Modern LLM'lerin önemli bir bölümü **Transformer mimarisi** kullanır.



Transformer Mimarisinin Temel Yapısı



Transformer mimarisi sayesinde model:

- Kelimeler arasındaki ilişkileri,
- Cümle bağlamını,
- Uzun metinlerdeki bağlantıları,
- Kelimelerin birbirleriyle olan ilişkilerini

daha etkili şekilde öğrenebilir.

Eğitim ≠ Kullanım

Önemli bir ayrım:

Model eğitimi:

Modelin verilerden öğrenmesi.



≠

Model kullanımı (Inference):

Eğitilmiş modelin yeni bir kullanıcı girdisine cevap vermesi.



Model Eğitimi Türleri

Yapay zekâ modelleri, farklı aşamalarda ve yöntemlerle eğitilerek daha güçlü ve faydalı hale gelir.

Büyük Dil Modelleri

1 Ön Eğitim (Pre-training)

Model çok büyük ve genel veri kümeleri üzerinde eğitilir.

Amaç:
Genel dil ve bilgi örüntülerini öğrenmek.

- Web siteleri
- Kitaplar
- Makaleler
- Belgeler
- Diğer kaynaklar

→ **Genel LLM**
(Önceden eğitilmiş model)

Milyarlarca kelime, çok çeşitli kaynaklar ve genel bilgiler.

2 İnce Ayar (Fine-tuning)

Önceden eğitilmiş model belirli bir görev veya alan için yeniden eğitilir.

Örneğin:

Genel LLM → **Tıp alanına yönelik model**
- Tıbbi sorular
- Hastalık teşhisi
- Klinik rehberler
- ...

veya

Genel LLM → **Hukuk alanına yönelik model**
- Mevzuat analizi
- Hukuki danışmanlık
- Sözleşme inceleme
- ...

Amaç: Belirli alanlarda daha doğru, uzman ve faydalı yanıtlar üretmek.

3 İnsan Geri Bildirimiyle Hizalama

Modelin insan beklentilerine daha uygun yanıtlar vermesi için insan değerlendirmelerinden yararlanılabilir.

Örneğin:

Soru:
"Diyabet hastaları için beslenme önerileri nelerdir?"

Yanıt A ✓
Diyabet hastaları için şunlar önerilir: ...
(Doğru, açık, güvenli ve bilimsel bilgiler içerir.)
★★★★★
İyi

Yanıt B ✗
Kesinlikle şeker tüketmelisiniz, bu hastalığı tamamen geçirir.
(Yanlış veya yetersiz bilgi içerir.)
★☆☆☆☆
Uygun değil

İnsan değerlendirmeleri, modelin daha uygun, doğru ve güvenli yanıtlar üretmesine yardımcı olur.

Veri Nedir?

Veri, yapay zekâ modelinin öğrenmesi için kullanılan bilgi ve örneklerdir.

Metin

Yazılı içerikler, cümleler, makaleler, sohbetler vb.

"Yapay zekâ, insan benzeri zekâ gerektiren görevleri gerçekleştirebilen sistemlerdir. Günümüzde eğitim, sağlık, ulaşım gibi birçok alanda kullanılmaktadır."

Örnek: Blog yazısı, haber, sohbet

Görsel

Fotoğraf, resim, çizim, grafik vb.



Örnek: Fotoğraf, illüstrasyon, infografik

Ses

Konuşma, müzik, ses kayıtları vb.



Örnek: Podcast, konuşma, müzik

Video

Hareketli görüntüler, videolar, animasyonlar vb.



Örnek: Eğitim videosu, film, animasyon

Sayısal Veriler

Sayılar, ölçümler, istatistikler, sensör verileri vb.

Yaş	Sıcaklık (°C)	Nem (%)	Adım Sayısı
25	23.5	45	8240
30	24.1	50	9120
28	22.8	48	7630

Örnek: Tablo, istatistik, sensör verisi

Kod

Programlama dilleriyle yazılmış kod parçaları.

```python
def topla(a, b):
    return a + b

sonuc = topla(5, 3)
print(sonuc) # 8
```

Örnek: Python, Java, C++, JavaScript

Dokümanlar

PDF, Word, Excel, sunumlar vb. dosyalar.

Örnek: Makale, rapor, e-kitap, sunum

Model Eğitiminin Temel Aşamaları

Yapay Zeka ve Uygulamaları - Copyright © 2026

Büyük Dil Modelleri

1 Veri Toplama



Modelin öğrenmesi için uygun veriler toplanır.

2 Veri Temizleme



- Hatalı veriler
- Tekrarlanan veriler
- Uygun olmayan veriler

Hatalı, gereksiz, tekrarlanan veya uygun olmayan veriler temizlenir.

3 Veri Hazırlama



- Özellik çıkarımı
- Normalizasyon
- Dönüşüm

Veriler modelin kullanabileceği formata dönüştürülür.

4 Eğitim (Training)



Model, veriler üzerinde tekrar tekrar çalışarak parametrelerini günceller.

5 Doğrulama (Validation)



Modelin öğrenme sürecindeki performansı değerlendirilir.

6 Test (Testing)



Modelin daha önce görmediği veriler üzerindeki başarısı ölçülür.

7 Dağıtım (Deployment)



Başarılı model gerçek kullanıcıların kullanımına sunulur.

1 Veri Toplama

İlk aşamada modelin öğrenmesi için gerekli veriler toplanır.

Örnek:

Bir görsel sınıflandırma modeli için:
50.000 kedi + **50.000** köpek görseli toplanabilir.

Kedi Görselleri
50.000 adet



Köpek Görselleri
50.000 adet



LLM için

ise çok büyük miktarda
metinsel veri kullanılabilir.



Yapay Zeka ve Uygulamaları - Copyright © 2026

2 Veri Kaynakları

Yapay zekâ modelleri, farklı kaynaklardan elde edilen verilerle beslenir.



Akademik yayınlar



Kitaplar



Web içerikleri



Kurumsal dokümanlar



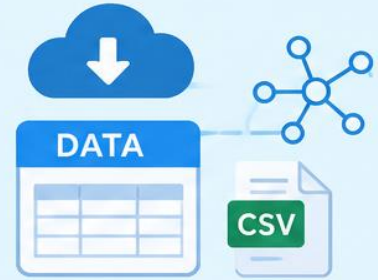
Sensörler



Veritabanları



Kullanıcı etkileşimleri



Açık veri setleri

Yapay Zeka ve Uygulamaları - Copyright © 2026

3 Veri Temizleme

Ham veriler doğrudan modele verilmez.
Verilerde aşağıdaki sorunlar bulunabilir:

- Eksik bilgiler
- Tekrarlanan kayıtlar
- Hatalı bilgiler
- Gürültülü veriler
- Gereksiz içerikler
- Yanlış etiketler

Bu nedenle verilerin temizlenmesi gerekir.

Ham Veri

ID	Ad	Yaş	Şehir	Etiket
1	Ayşe	25	İstanbul	Kedi
2	Mehmet	-	Ankara	Kopek
2	Mehmet	-	Ankara	Kopek
4	Elif	32	İzmir	Kedi
5	Ali	120	Bursa	Kopek
6	Zeynep	28	İstanbul	Kedi
7	Deniz	?	Antalya	Kopek
8	Mert	35	-	Kedi
9	Ayşe	25	İstanbul	Kedi
10	Can	40	İstanbul	Köpek

Veri Temizleme Süreci

- Eksik verileri tamamla veya çıkar.
- Tekrarlanan kayıtları kaldır.
- Hatalı verileri düzelt.
- Gürültülü verileri temizle.
- Gereksiz içerikleri sil.
- Yanlış etiketleri düzelt.

Temiz Veri

ID	Ad	Yaş	Şehir	Etiket
1	Ayşe	25	İstanbul	Kedi
2	Mehmet	30	Ankara	Köpek
4	Elif	32	İzmir	Kedi
5	Ali	28	Bursa	Köpek
6	Zeynep	28	İstanbul	Kedi
7	Deniz	22	Antalya	Köpek
8	Mert	35	Ankara	Kedi
10	Can	40	İstanbul	Köpek

Yapay Zeka ve Uygulamaları - Copyright © 2026

Veri Etiketleme

Bazı yapay zekâ modellerinde verilerin **etiketlenmesi (labeling)** gerekir.

Örneğin:

Görsel	Etiket
 Görsel 1	Kedi
 Görsel 2	Köpek
 Görsel 3	Kedi

Model zamanla:
Görsel özellikleri → **Etiket** ilişkisini öğrenir.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Verinin Bölünmesi

Makine öğrenmesi sürecinde veri genellikle üç ana gruba ayrılır:

Eğitim Verisi (Training Data)

Modelin **öğrenmesi** için kullanılır.



Doğrulama Verisi (Validation Data)

Modelin eğitim sırasında **performansını değerlendirmek** ve **ayarlamak** için kullanılır.



Test Verisi (Test Data)

Modelin daha önce görmediği veriler üzerinde **gerçek performansını ölçmek** için kullanılır.



Basit bir örnek:



Model Eğitimi

Hazırlanan eğitim verileri modele verilir. Model, veriyi görür, tahmin yapar, hatasını hesaplar ve parametrelerini günceller. Bu işlem çok sayıda tekrar gerçekleştirilir.

Örnek:



Girdi: Öğrencinin özellikleri → **Model:** Başarı tahmini = %60 → **Gerçek sonuç:** Başarılı → **Hata hesaplama** → **Model parametrelerini güncelleme**

Modelin Eğitim Süreci



Eğitim Verisi
(Önceden hazırlanmış)

→



Model
(Başlangıç parametreleri)

→ **Tahmin**
(Model çıktısı)

→ **Hata**
(Loss)

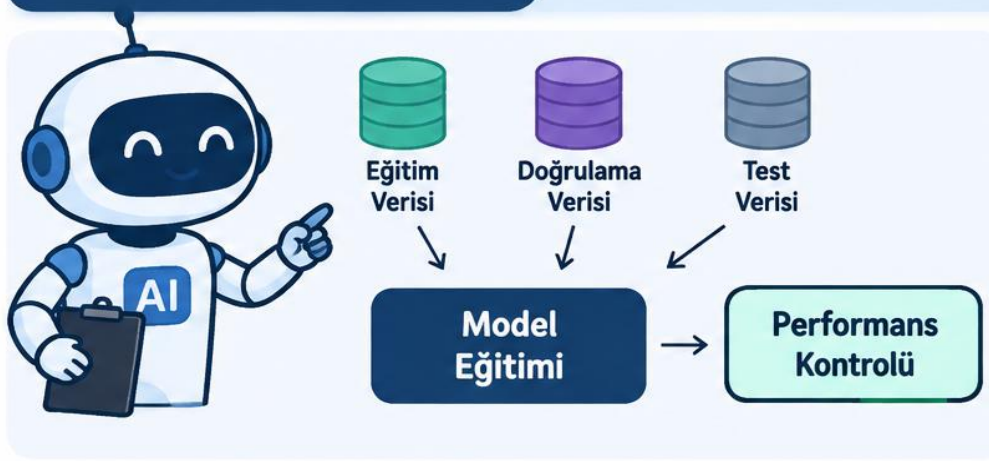
→ **Parametre Güncelleme**
(Optimizer)

Tekrar Et

Doğrulama

Modelin eğitim sürecindeki performansı doğrulama verileriyle kontrol edilir.

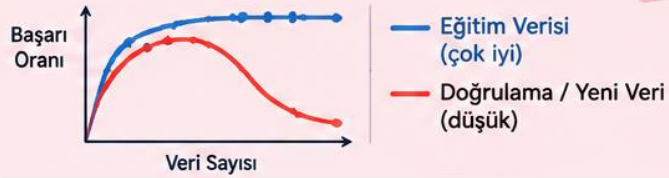
Burada amaç:
"Model sadece eğitim verisini ezberliyor mu, yoksa gerçekten öğreniyor mu?"
sorusuna cevap vermektir.



Overfitting (Aşırı Öğrenme)

Model eğitim verisini çok iyi öğrenir ancak yeni verilerde başarısız olursa overfitting ortaya çıkabilir.

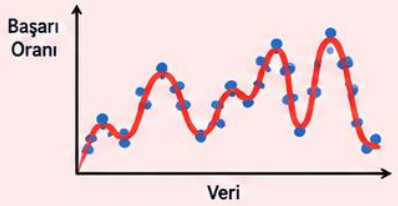
Model ezberliyor, öğrenmiyor!



Ezberleme ≠ Öğrenme

Bu, yapay zekâ eğitiminde en önemli problemlerden biridir.

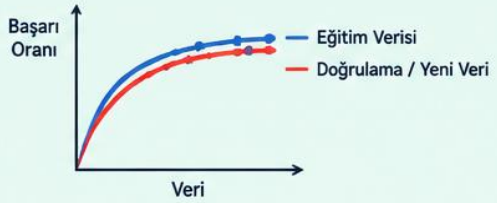
Ezberleme (Overfitting)



- Eğitim verisinde çok yüksek başarı sağlar.
- Yeni verilerde başarısı düşer.
- Genelleme yapamaz.
- Sadece gördüğü veriyi hatırlar.

Sadece ezberler, öğrenmez.

Gerçek Öğrenme



- Eğitim verisinde iyi sonuç verir.
- Yeni verilerde de başarılı olur.
- Genelleme yapar.
- Veriyi anlamlandırır, kurallar öğrenir.

Öğrenir, geneller, başarıyı sürdürür.

Yapay Zeka ve Uygulamaları - Copyright © 2026

Test

Model eğitimini tamamladıktan sonra daha önce görmediği test verileri kullanılır.

Test verileri, modelin gerçek dünyadaki performansını ölçmek için çok önemlidir.

Eğitim Aşaması

Model, **100.000** görsel ile eğitilir.



Eğitim Verisi
100.000 görsel

Test Aşaması

Daha önce görmediği **10.000** görsel gösterilir.



Test Verisi
10.000 görsel

Performans Ölçümü

Modelin test verilerindeki başarısı ölçülür.



- Doğruluk (Accuracy)
- Kesinlik (Precision)
- Duyarlılık (Recall)
- F1 Skoru

Modelin gerçek dünya performansı oranlanır.

Özetle:

Eğitim Verisi (100.000 görsel) → Model Eğitilir → Test Verisi (10.000 görsel) → Performans Ölçülür

Test verisi, modelin gerçekten öğrenip öğrenmediğini ortaya koyar.

Yapay Zeka ve Uygulamaları - Copyright © 2026



Model Değerlendirmesi

Modelin başarısı probleme göre farklı metriklerle ölçülür.



Örneğin:

- Accuracy (Doğruluk)
- Precision (Kesinlik)
- Recall (Duyarlılık)
- F1-Score
- ROC-AUC
- MAE
- RMSE

Accuracy (Doğruluk)

Modelin genel olaraık doğru tahmin oranıdır.

$$\text{Accuracy} = \frac{\text{Doğru Tahmin}}{\text{Toplam Tahmin}}$$

TP (Doğru Pozitif)	FP (Yanlış Pozitif)
FN (Yanlış Negatif)	TN (Doğru Negatif)

Örn: %92 doğruluk

Precision (Kesinlik)

Pozitif tahminler arasında doğru olanların oranıdır.

$$\text{Precision} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif + Yanl ş Pozitif}}$$

Örn: Alarmların %95'i gerçekten doğru.



Doğru pozitif tahminler

Recall (Duyarlılık)

Gerçek pozitiflerin ne kadarının doğru tahmin edildiğini gösterir.

$$\text{Recall} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif + Yanl ş Negatif}}$$

Örn: Hastaların %88'i teşhis edildi.



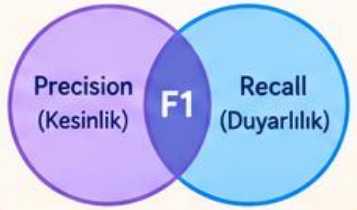
Doğru tespit edilenler

F1-Score

Precision ve Recall'in harmonik ortalamasıdır.

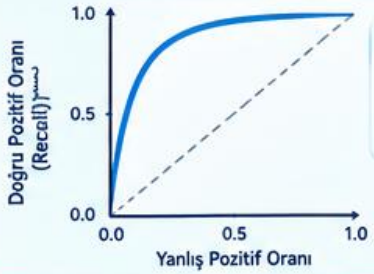
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Örn: %90 dengeli başarı.



ROC-AUC

Modelin sınıfları ayırt etme kabiliyetini gösterir.



AUC = 0.92 (Mükemmel performans)

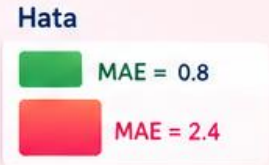
AUC değeri 1'e ne kadar yakınsa o kadar iyidir.

MAE (Ortalama Mutlak Hata)

Tahminlerin gerçek değerlere ortalama uzaklığını gösterir.

$$MAE = \frac{1}{n}\sum |y_i - \hat{y}_i|$$

Düşük değerler daha iyi performans demektir.



RMSE (Kök Ortalama Kare Hatası)

Hataların karelerinin ortalamasının kareköküdür. Büyük hatalara daha fazla önem verir.

$$RMSE = \sqrt{\frac{1}{n}\sum (y_i - \hat{y}_i)^2}$$

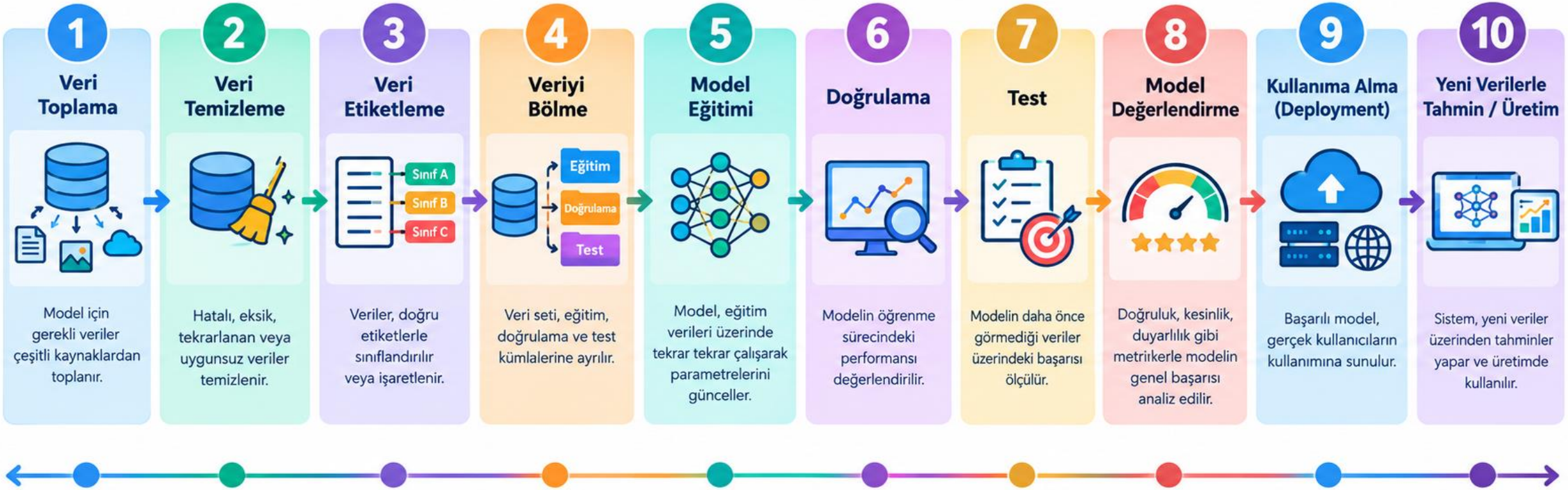
Düşük değerler daha iyi performans demektir.



Basit Bir Örnek: Öğrenci Başarı Tahmin Sistemi

Yapay Zeka ve Uygulamaları - Copyright © 2026

Büyük Dil Modelleri



Veriden Değere: Yapay Zeka ile Daha Akıllı Çözümler

Büyük Dil Modelleri

Veri

Model, geçmiş öğrenci verilerinden öğrenir.

- **Sınav notları** (vize, final vb.)
- **Devam durumu** (yüzde)
- **Ödev puanları** (ortalamalar)
- **Çalışma süresi** (saat/hafta)
- **Önceki başarı** (geçmiş dönem notu)

Model Eğitimi

Toplanan verilerle makine öğrenmesi modeli eğitilir.



Eğitim verileri

Öğrüntü (Öğrenilen Kural)

Model, eğitim verilerindeki örüntülerden şu sonucu öğrenebilir:

"Düzenli çalışan ve yüksek devam oranına sahip öğrencilerin başarılı olma olasılığı daha yüksek."



Daha fazla çalışma +
Daha yüksek devam =
Daha yüksek başarı olasılığı

Yeni Öğrenci

Model, yeni öğrencinin verilerini analiz eder.



Sınav notları	85
Devam durumu	%92
Ödev puanları	88
Çalışma süresi	12 saat/hafta
Önceki başarı	78

Tahmin Sonucu

Model, yeni öğrenci için tahminini yapar.



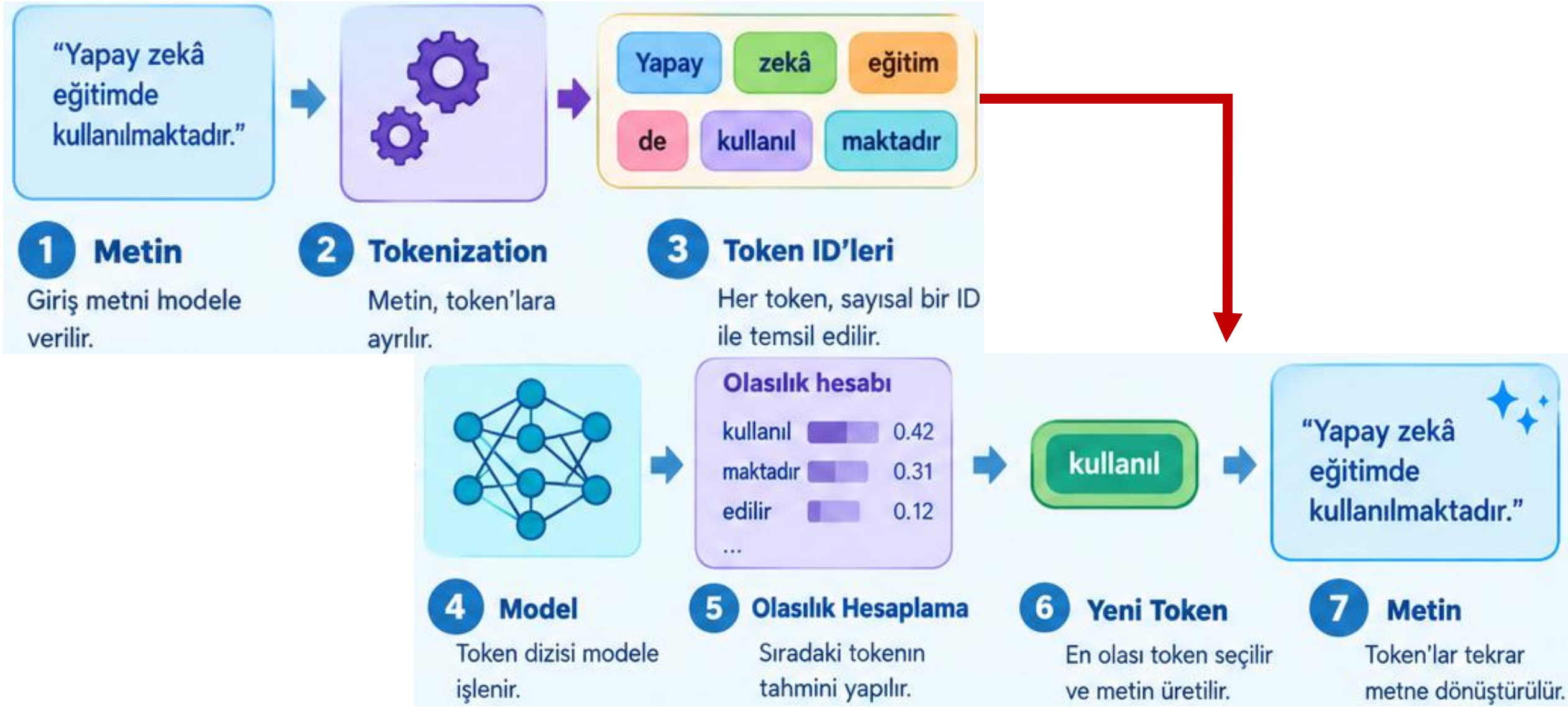
Yüksek başarı olasılığı

LLM'ler metni **token** adı verilen parçalara ayırarak işler ve bağlama göre hangi tokenların gelmesinin daha olası olduğunu hesaplayarak çıktı oluşturur.



Büyük Dil Modelleri

LLM'de Token Süreci



Örnek: "Merhaba dünya!"



Kısaca: **Token** = LLM'nin metni okuyup, işleyip ve üretebilmek için kullandığı sayısallaştırılmış metin parçasıdır.

Token Neden Önemlidir?

Bağlam Uzunluğu
Modelin aynı anda işleyebileceği token sayısını belirler.

Maliyet
API kullanımında genellikle girdi ve çıktı token'ları maliyet hesabında kullanılır.

Hız
Daha fazla token genellikle daha fazla hesaplama anlamına gelir.

Çıktı Uzunluğu
Modelin üretebileceği metin token sınırından etkilenir.



Nasıl Çalışır?



Model, bu tokenleri sayısal temsillere dönüştürerek işler ve anlamı öğrenir.

Token her zaman bir kelime değildir.

Bir token;

- Tam bir kelime,
- Kelimenin bir bölümü,
- Bir ek,
- Noktalama işareti,
- Sayı,
- Boşlukla birlikte bir karakter dizisi olabilir.

Örnek: Tokenizasyon

Girdi metni:

"Yapay zekâ, geleceği şekillendiriyor!"

Oluşan tokenler:

Yapay | zek | â | , | gelece | ği | şekil | lendir | iyor | !

Doğru tokenizasyon, modelin metni daha iyi anlamasını ve daha doğru sonuçlar üretmesini sağlar.

Farklı örnekler

"kitap"	→	kitap	(tam kelime)
"yazılım"	→	yaz + ılım	(kelimenin bölümü)
"-ler"	→	-ler	(ek)
"."	→	.	(noktalama işareti)
"2025"	→	2025	(sayı)
" " (boşluk)	→		(boşlukla birlikte karakter dizisi)

Örnek Ayrıştırma

“Yapay zekâ öğreniyor.”



Yapay	zekâ	öğren	iyor	.
Token	Token	Token	Token	Token



Önemli Not

Bu ayrıştırma, kullanılan modele ve tokenizer’a göre değişir.



Dolayısıyla “1 kelime = 1 token” şeklinde düşünmek doğru değildir.

Farklı Tokenizasyon Örnekleri

Aynı cümle, farklı modeller ve tokenizer'lar tarafından farklı şekillerde parçalanabilir.

BPE (Byte Pair Encoding)

Yapay | zek | â | öğren | iyor | .

6 token

BPE gibi yöntemlerde, sık görülen kelime parçaları birleştirilerek token'lara dönüştürülür.

WordPiece

Yapay | zek | ##â | öğren | ##iyor | .

6 token

WordPiece, özellikle alt kelime parçalarını (subword) kullanır. (## işareti, parçanın devamını gösterir.)

SentencePiece

_Yapay | _zekâ | _öğren | iyor | .

5 token

SentencePiece, boşlukları da token'lara dahil edebilir ve farklı parçalama stratejileri kullanır.

Büyük Dil Modelleri

Örnek “ Yapay zekâ gelecektir.”

Metin

“Yapay zekâ gelecektir.”

→

Tokenization
(Tokenlara ayırma)

Yapay | zekâ | gelecek | tir | .

→

Token ID’leri

101	(Yapay)
532	(zekâ)
784	(gelecek)
203	(tir)
7	(.)

→

Model işlemesi



Model, ID’ler üzerinden metni işler.

→

Yeni tokenlar

101 | 532 | 784 | 203 | 7

Model, bir sonraki tokenları üretir.

→

Yanıt

“Yapay zekâ gelecektir.”

Önemli: Tokenizasyon yöntemi modele ve kullanılan tokeniter’a göre değişir.
Bu yüzden “1 kelime = 1 token” şeklinde düşünmek doğru değildir.

Token ve Kelime Aynı Şey Değildir

Bu konu özellikle önemlidir.

Kelime ≠ Token

İngilizcede Örnek

"unbelievable" kelimesi bazı tokenizasyon yöntemlerinde şu şekilde parçalanabilir:

unbelievable → **un** (token) + **believe** (token) + **able** (token)

Türkçede Örnek

Türkçede **eklemeli dil yapısı** nedeniyle uzun **kelimeler birden fazla tokena** ayrılabilir.

Örneğin: **"evlerimizden"** kelimesi model tarafından tek bir token yerine birden fazla parçaya ayrılabilir.

evlerimizden → **ev** (token) + **ler** (token) + **imiz** (token) + **den** (token)

Önemli

Bu nedenle:

100 kelime = 100 token ✗

demek doğru değildir.

Kısaca



- **Kelime**, anlamlı bir birimdir (sözlükteki kelime).
- **Token**, modelin metni işlediği küçük parçalardır.
- Bir kelime birden fazla tokena ayrılabilir.
- Bazı kelimeler tek bir token olabilir.
- Bu nedenle, token sayısı **her zaman kelime sayısına eşit değildir.**

Token ve LLM Bağlam Penceresi

LLM'lerin bir seferde işleyebileceği maksimum token miktarı vardır. Buna **context window (bağlam penceresi)** denir.

Context Window
128.000 token

Context Window Nedir?

Örneğin bir modelin bağlam penceresi:

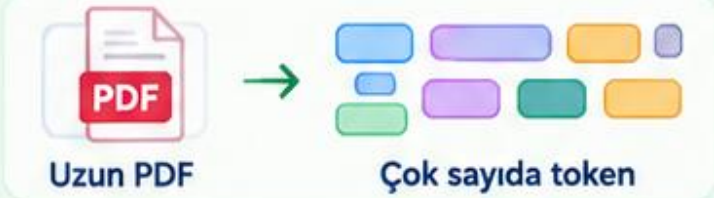
128.000 token

ise model, tek bir etkileşim bağlamında belirli bir **üst sınıra** kadar metin ve diğer desteklenen girdileri işleyebilir.



Token Sayısı Neden Önemli?

- Her kelime bir token değildir.
- Metin, farklı tokenizasyon yöntemlerine göre farklı sayıda tokena ayrılabilir.
- Uzun ve karmaşık metinler (özellikle PDF'ler) daha fazla token üretir.



Problemin Ortaya Çıkışı



Uzun PDF
veya büyük metin

→



Çok sayıda
token

→

Bağlam sınırı (128.000 token) aşılabilir.

→ Bu durumda model, tüm içeriği tek **seferde işleyemez** ve bilgi kaybı, kesinti veya hata ortaya çıkabilir.

RAG Sistemlerinde Neden Önemli?

RAG (Retrieval-Augmented Generation), harici kaynaklardan (ör. doküman, veritabanı) bilgileri alarak LLM'e ek bağlam sağlar.

Ancak bu eklenen içerikler de token tüketir.
Bu nedenle RAG sistemlerinde bağlam penceresi sınırı dikkatle yönetilmelidir.



Harici kaynaklardan
bilgi alınır
(retrieval)

+

LLM

Soru + Alınan bilgi
(bağlam)

Context Window sınırı

→



Daha doğru,
güncel ve kapsamlı
yanıt

Token ve Maliyet

API ile kullanılan birçok LLM hizmetinde ücretlendirme **token miktarına** göre yapılabilir.

Genellikle:

Input Tokens

Kullanıcının modele gönderdiği içerik.

+

Output Tokens

Modelin oluşturduğu içerik.

=

Toplam Token Kullanımı

Toplam token

Örnek:

Bir öğrenci için vize ve final notlarını hesaplayan Python programı yaz.

İşte Python kodu... </>

Input Tokens	:	350
Output Tokens	:	620
Toplam Token	:	970

Yapay Zeka ve Uygulamaları - Copyright © 2026

Dikkat

Dolayısıyla daha uzun promptlar ve daha uzun cevaplar **daha fazla token** kullanımına yol açabilir.

Daha uzun prompt (daha fazla token) + **Daha uzun cevap** (daha fazla token) → **Daha yüksek maliyet**

Unutmayın

Ancak her model ve hizmetin fiyatlandırma sistemi farklıdır.



Token ve RAG İlişkisi

Özellikle sizin akademik PDF/RAG çalışmalarınız açısından token kavramı çok önemlidir. Token'lar, metnin anlamını koruyarak belgeleri parçalara ayırır, arama ve ilişkilendirme süreçlerinin temelini oluşturur.

Büyük Dil Modelleri

1 — 10.000 sayfalık PDF



Akademik makale, tez, rapor veya kitap gibi büyük bir PDF dosyası yüklenir.

Büyük belge

2 — Metin çıkarma



PDF'den metinler çıkarılır (OCR / parser işlemi)

Ham metin

3 — Chunk'lara bölme

Chunk 1
Chunk 2
Chunk 3
...

Metin, anlam bütünlüğü korunarak daha küçük parçalara (chunk) ayrılır.

Parçalı metin

4 — Tokenizasyon

Akademik | araştırma
yapılması | gereken
... | ...

Her chunk, modelin anlayacağı token'lara ayrılır.

12 Token dizisi

5 — Embedding oluşturma

[0.12 0.34 0.56 ...]
[0.23 0.11 0.67 ...]
[0.45 0.32 0.21 ...]

Token'lar, anlamsal vektörlere dönüştürülür.

Vektör temsili

6 — Vektör veritabanına kaydetme



Oluşturulan embedding'ler, vektör veritabanına kaydedilir ve RAG sürecinde kullanılır.

Vektör veritabanı

Token'lar RAG'de Nasıl Kullanılır?

Kullanıcı bir soru sorduğunda, sistem önce sorunun **embedding'ini** oluşturur. Vektör veritabanında **en ilgili parçalar (chunk'lar) bulunur.** Bu parçalar, token'lara ayrılmış metinlerden oluşur ve modelin bağlamına eklenir. Böylece LLM, **daha güncel ve doğru bilgilere dayalı yanıt üretebilir**

RAG Süreci (Özet)



Unutmayın!

- Token sayısı, maliyeti ve modelin bağlam kapasitesini etkiler.
- PDF/RAG projelerinde doğru chunklama ve tokenizasyon, başarı için kritik öneme sahiptir.
- Token'lar, sadece "kelime" değil; modelin anladığı en küçük anlamlı parçadır.





Tokenization Simülatörü

Türkçe cümleler token'lara nasıl bölünür? Hemen dene!

Cümlenizi Yazın:

Öğrencilerimiz teknolojiye adapte oluyorlar.

Örnek 1 | Örnek 2 | Örnek 3 | Örnek 4

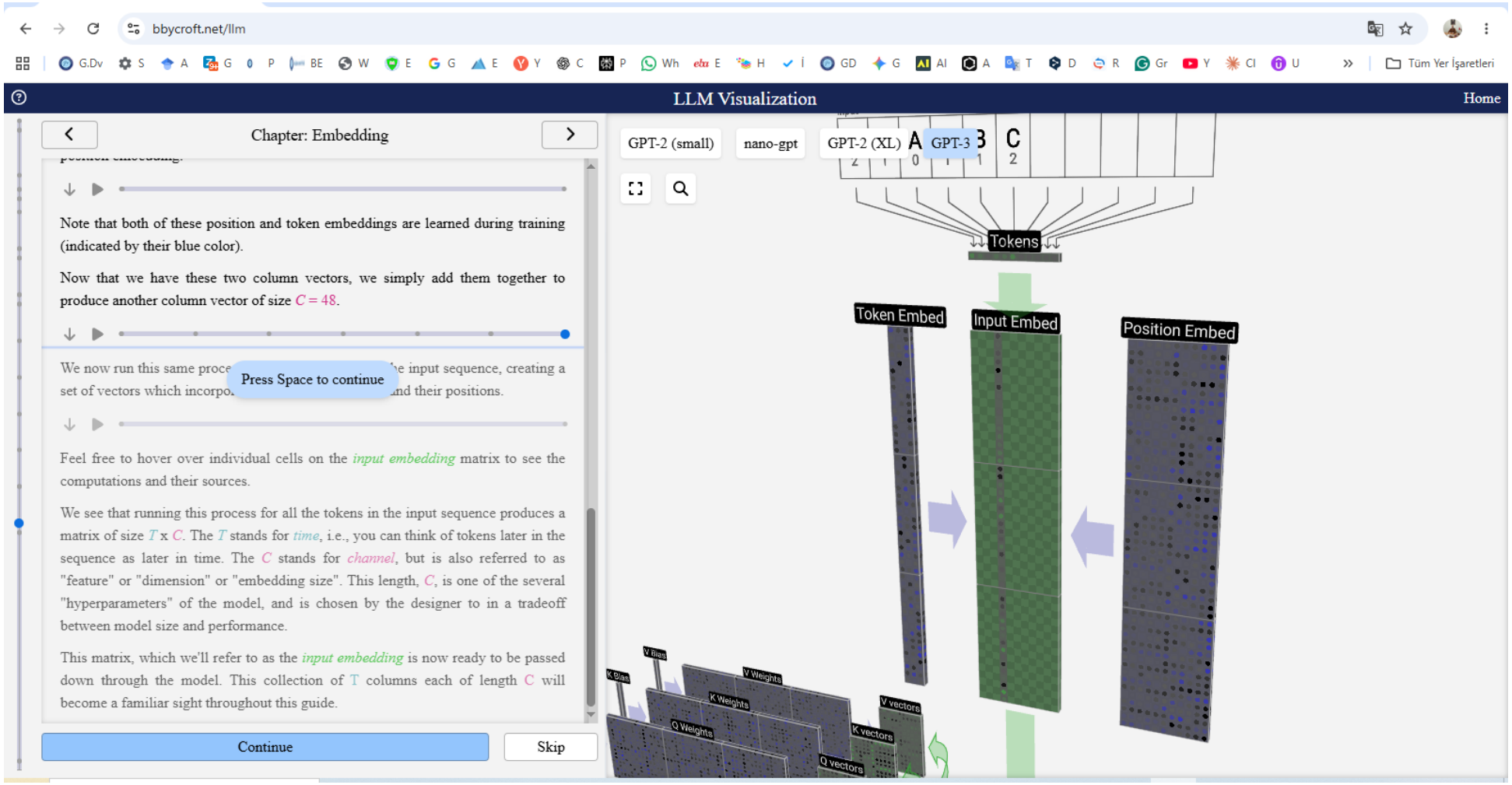
Token Sayısı: 12

Token'lar:

öğren #8978	ci #3174	ler #7033	imiz #6181	teknoloji #5485	ye #3852	adap #8946
te #3697	ol #3549	uyor #5383	lar #6909	. #46		

https://www.brick.institute/nasil-calisir/llm-nasil-calisir

Büyük Dil Modelleri



https://bbycroft.net/llm

Bağlam (Context) Kavramı

Bağlam (Context), bir yapay zekâ modelinin verilen bir soruyu veya talimatı doğru yorumlamak için dikkate aldığı **önceki ve mevcut bilgi bütünüdür.**



En basit ifadeyle:

Bağlam = Yapay zekânın bir isteği anlamlandırırken "bildiği ve dikkate aldığı" ilgili bilgiler

Yapay Zeka ve Uygulamaları - Copyright © 2026

Büyük Dil Modelleri

Bağlam Nerelerden Gelir?

- **Önceki mesajlar** (sohbet geçmişi)
- **Belge ve dokümanlar** (dosyalar, web sayfaları)
- **Kullanıcı bilgileri** (ad, rol, tercihler)
- **Mevcut durum** (oturum, zaman, görev)
- **Harici bilgiler** (ör. internet, veritabanı)

Bağlam



Tüm bu bilgiler bir araya gelir ve yapay zekâ modeline **"bağlam"** olarak sunulur.

Yapay Zekâ Modeli



Model, bağlamı dikkate alarak isteği doğru şekilde yorumlar ve yanıt üretir.

Doğru ve Anlamlı Yanıt



Sonuçta daha doğru, kişiselleştirilmiş ve bağlamsal bir yanıt elde edilir.

Basit Örnek

Kullanıcı: “Ahmet bugün derse gelmedi.”
Ardından: “Neden?”



Ahmet bugün derse gelmedi.

Kullanıcı

Bağlam

Model, önceki cümledeki **“Ahmet”** ve **“derse gelmedi”** bilgilerini **bağlam** olarak alır.



Neden?

Ahmet bugün derse gelmediği için muhtemelen hastadır, ulaşım sorunu yaşamıştr veya başka bir sebebi olabilir.

Yapay Zekâ Modeli

Bağlam Varken



Model, önceki cümleyi hatırlayarak soruyu doğru şekilde yorumlar.

Bağlam Olmadan



Bağlam olmadığı için "Neden?" sorusunun neyle ilgili olduğu belirsizdir.

LLM'lerde Bağlam

Bir LLM ile konuşurken önceki mesajlar, dokümanlar ve kullanıcının verdiği bilgiler modelin **bağlamını** oluşturabilir.

Örnek Konuşma

Kullanıcı: Ben Python öğreniyorum. 10:24

Kullanıcı: Bana başlangıç seviyesinde bir proje öner. 10:26

LLM: Tabii! Python ile başlangıç seviyesinde yapabileceğin bazı proje önerileri şunlar... 10:27

Modelin Bağlamı

- **Önceki mesajlar** "Ben Python öğreniyorum."
- **Dokümanlar** (varsa)
- **Kullanıcının verdiği bilgiler** (ilgi alanları, seviyeleri, vb.)

Tüm bu bilgiler modelin bağlamını oluşturur.

Bağlam Sayesinde Daha Doğru Yanıt



- ✓ Python'a uygun
- ✓ Başlangıç seviyesi
- ✓ Adım adım
- ✓ Kolay proje önerileri

Önerilen Projeler:

1. Basit bir hesap makinesi
2. Not defteri uygulaması
3. Hava durumu bilgisi çeken uygulama
4. Sayı tahmin oyunu
5. Kitap öneri listesi uygulaması

Bağlam Penceresi (Context Window)

LLM'lerin aynı anda işleyebileceği bağlam miktarına **bağlam penceresi (context window)** denir.

Bağlam genellikle **token** cinsinden ifade edilir.

Örneğin:

128K token context window

ifadesi, modelin ilgili kullanım senaryosunda yaklaşık 128.000 tokenlık bir bağlamı işleyebilecek kapasiteye sahip olduğunu belirtir.

Kesin davranış modele ve uygulamaya göre değişebilir.

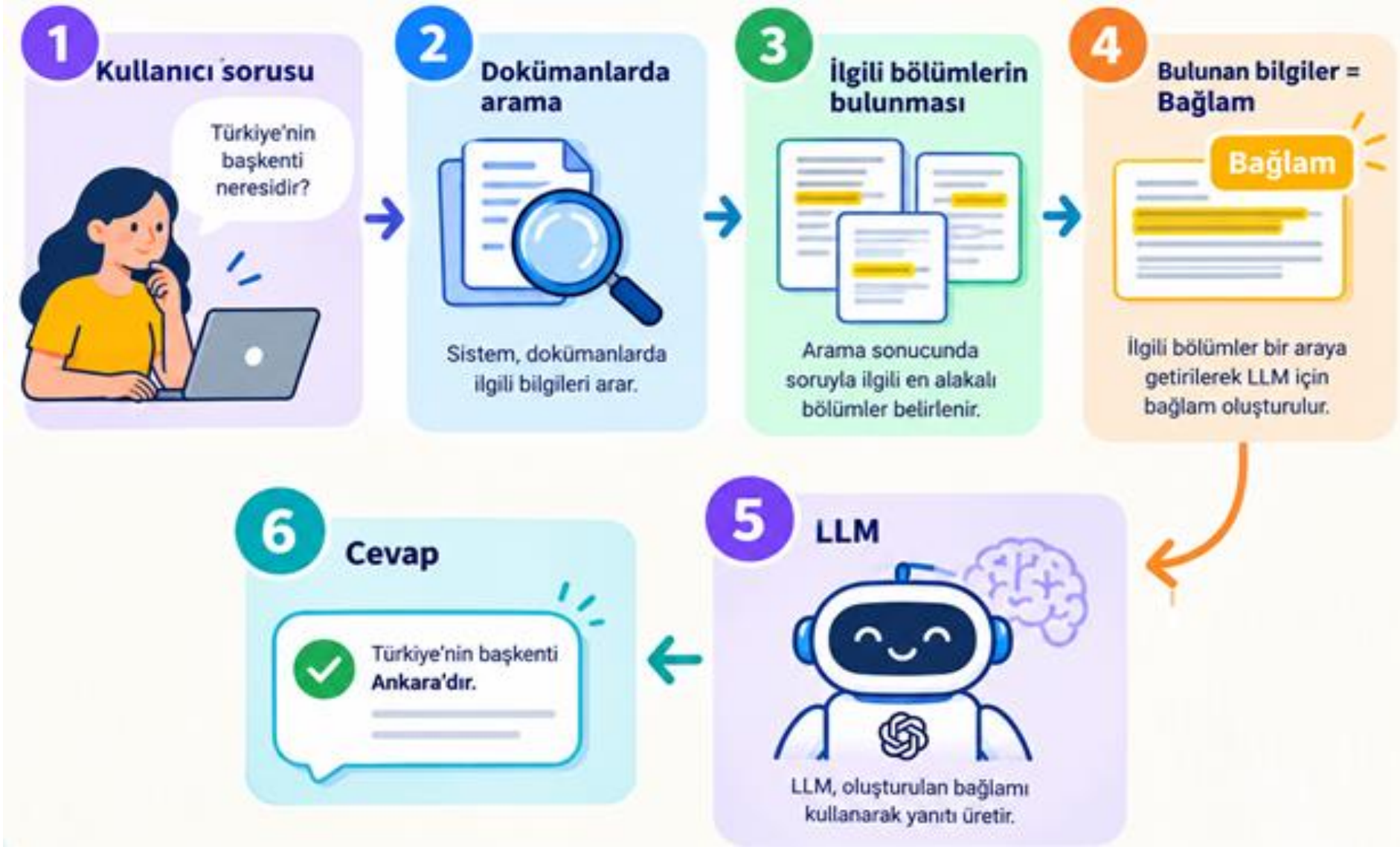


RAG Sistemlerinde Bağlam

RAG (Retrieval-Augmented Generation) sistemlerinde **bağlam** özellikle önemlidir.



Büyük Dil Modelleri



RAG Nasıl Çalışır?

1 Kullanıcı Sorusu

"Üniversitenin sınav yönetmeliğine göre bütünleme sınavı nasıl uygulanır?"

Kullanıcı, sorusunu sisteme yazar.

2 Arama (Retrieval)

- ✔ Yönetmelik
- ✔ Dokümanlar
- ✔ Veritabanı
- ✔ İnternet

Sistem, ilgili kaynaklarda anahtar kelimelerle arama yapar ve en uygun bölümleri bulur.

3 Bölümler (Context)

İlgili yönetmelik bölümü

Bulunan bilgiler, LLM için bağlam (context) olarak hazırlanır.

4 LLM

LLM

LLM, soruyu ve ilgili bağlamı birlikte değerlendirerek cevabı üretir.

5 Cevap

Kullanıcıya, ilgili yönetmelik bilgisine dayalı doğru ve anlaşılır bir yanıt sunulur.

Örnek Akış

Kullanıcı Sorusu

"Üniversitenin sınav yönetmeliğine göre bütünleme sınavı nasıl uygulanır?"

RAG

İlgili yönetmelik bölümü bulunur.

LLM

Bağlamla birlikte değerlendirilir.

Cevap

Doğru ve kaynaklı bir yanıt üretilir.

RAG'de doğru bağlamın bulunması, doğru cevap üretmek açısından kritik öneme sahiptir.



Doğru bilgi + Doğru bağlam = Güvenilir cevap

Bağlam ve Token İlişkisi

Bu iki kavram birbirine doğrudan bağlıdır.

Büyük Dil Modelleri







LLM'lerde Temel Süreç

Kullanıcıdan gelen girdiden, yanıtın tamamlanmasına kadar olan adımlar.

Büyük Dil Modelleri

1 Kullanıcı Girdisi



Kullanıcı, sorusunu veya isteğini sisteme iletir.

2 Tokenizasyon

Türkiye'nin başkenti neresidir? → Türkiye | 'nin | başkenti | neresidir | ?

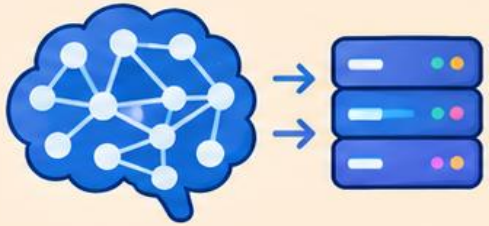
Metin, modelin anlayabileceği küçük parçalara (token) ayrılır.

3 Bağlamın Analizi



Gerekli bilgiler ve önceki bağlam incelenir, bağlam oluşturulur.

4 Modelin Hesaplamaları



Model, bağlamı kullanarak olasılıkları hesaplar.

5 Sonraki Tokenın Tahmini

Ankara	%72
İstanbul	%12
İzmir	%8
Diğer	%8

Model, bir sonraki token için en olası kelimeyi tahmin eder.

6 Tokenın Üretilmesi



Tahmin edilen token, çıktıya eklenir.

7 Tekrar Tahmin



Yeni bağlamla süreç tekrar başlar.

8 Yanıtın Tamamlanması

Türkiye'nin başkenti Ankara'dır.

Belirli bir token dizisi oluşturulunca yanıt tamamlanır.

Model Çıktısının Oluşumunda Temel Unsurlar

Bir yapay zekâ modelinin verdiği yanıt, birden fazla bileşenin birlikte çalışmasıyla oluşur.

Büyük Dil Modelleri





1 Prompt

Kullanıcının modele verdiği soru veya talimattır.



Kullanıcı

"Üretken yapay zekâyı 100 kelimeyle açıkla."

2 Bağlam (Context)

Modelin dikkate aldığı önceki mesajlar, dokümanlar veya ilgili bilgiler.

- Önceki sohbetler
- Yüklenen dokümanlar
- İlgili bilgiler

→ Bağlam

3 Tokenlar

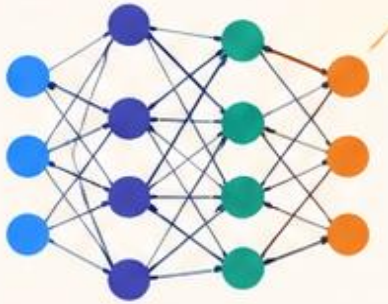
Modelin işlediği ve ürettiği temel metin birimleridir.

Üretken	yapay	zekâ	yı	...

Her kelime, alt kelime veya özel karakter bir token olarak işlenir.

4 Model Parametreleri

Modelin eğitim sırasında öğrendiği çok sayıdaki sayısal değerlerdir. Bunlar modelin dil ve örüntüleri temsil etmesinde rol oynar.



Parametreler

0.3721
-1.2847
0.5632
...

5 Olasılık Hesaplama

Model, mevcut bağlama göre hangi tokenın gelmesinin daha uygun olduğunu hesaplar.

yapay	%42
zekâ	%28
teknoloji	%15
bir	%8
...	%7

En yüksek olasılıklı token seçilir.

6 Token Üretimi

Seçilen token çıktıya eklenir ve model yeni bağlam üzerinden bir sonraki tokenı üretmeye devam eder.

Üretken | yapay | zekâ | ...

Yeni bağlam
(bir sonraki token için)

Büyük Dil Modelleri

1 Başlangıç Metni

Kullanıcı bir metin yazar. Model, bu metni bağlam olarak kullanır.

"Bugün hava çok...

Kullanıcı

2 Olası Devamlar Hesaplanır

Model, bağlama göre bir sonraki token için olası seçenekleri ve olasılıklarını hesaplar.

Olası tokenlar ve olasılıkları

güzel	%42
sıcak	%24
soğuk	%18
yağmurlu	%16

3 En Uygun Token Seçilir

Model, hesaplanan olasılıklara göre en uygun tokeni seçer.

güzel (%42)

4 Yeni Metin Oluşur

Seçilen token, metne eklenir. Bağlam güncellenir ve model yeniden bir sonraki tokeni hesaplar.

"Bugün hava çok güzel..."

Bağlam güncellenir

5 İşlem Tekrar Edilir

Bu süreç, yanıt tamamlanıncaya kadar devam eder.

1. Bugün
2. hava
3. çok
4. güzel
5. ...

Örnek Devam Süreci

"Bugün hava çok..." → güzel (%42) → güzel → ... → Tamamlanan Yanıt "Bugün hava çok güzel..."

AI **Model:** Bağlamı anlar, olasılıkları hesaplar, en uygun tokeni seçer. → **Sonuç:** Doğal ve tutarlı bir metin üretilir.

LLM'leri Yerel Olarak Çalıştırmak İçin En İyi Araçların Karşılaştırması

Yapay Zeka Aracı	En İyi Kullanım	Özellikler
Ollama	Yerel model sunumu ve geliştirici entegrasyonları	Model yönetimi, CLI, yerel sunucu ve uygulama entegrasyonları
LM Studio	Masaüstünde model keşfi, sohbet ve yerel API'ler	Model indirme, çevrimdışı sohbet, belgelerle sohbet ve geliştirici sunucusu
Open WebUI	Paylaşımlı, kendi sunucunuzda barındırabileceğiniz bir yapay zekâ arayüzü	Ollama ve API bağlantıları, bilgi, kullanıcılar ve araçlar
AnythingLLM	Belge çalışma alanları ve cihaz üzerinde üretkenlik	Masaüstü ve kendi sunucunuzda barındırabileceğiniz sürümler, bilgi erişimi, aracılar ve belge bilgisi
Jan	Açık kaynaklı bir masaüstü sohbet alternatifi	Yerel ve çevrimiçi modeller, masaüstü arayüzü ve açık geliştirme
Msty Studio	Yeniden kullanılabilir bir yapay zekâ çalışma alanında modelleri karşılaştırma	Bölünmüş Sohbetler, Bilgi Yığınları, kişilikler, yerel ve çevrimiçi modeller
LocalAI	Birden fazla yapay zekâ biçimi için kendi sunucunuzda barındırabileceğiniz API'ler	Uyumlu API'ler, modüler arka uçlar, metin, konuşma ve görüntü
llama.cpp	Doğrudan çıkarım ve donanım denetimi	GGUF, kuantizasyon, CPU/GPU arka uçları, CLI ve sunucu
GPT4All	Belgelerle kolayca yerel sohbet	Masaüstü uygulaması, LocalDocs, yerel modeller, Windows/macOS/Linux
TextGen	Gelişmiş masaüstü denemeleri ve isteğe bağlı eğitim	Taşınabilir GGUF derlemeleri, API ve görüntü desteği; eğitim için daha kapsamlı kurulum