

İstem Mühendisliği

Bilgisayar Mühendisliği 4. Sınıf Seçmeli Dersi

Dersin Amacı

Bu ders, öğrencilere **Büyük Dil Modelleri (BDM, Large Language Models - LLM)** ile etkili çalışma becerisi kazandırmayı hedefler. Öğrenciler:

- BDM'lerin temel çalışma prensiplerini ve mimarisini anlayacak
- Farklı problem türleri için optimize edilmiş istemler tasarlayabilecek
- API entegrasyonları ve yerel model kurulumları yapabilecek
- Erişim Artırımlı Üretim (Retrieval-Augmented Generation - RAG) sistemleri geliştirebilecek
- BDM tabanlı ajanlar oluşturabilecek
- Güvenlik, etik ve performans değerlendirmesi konularında bilgi sahibi olacak
- Gerçek dünya problemlerine BDM tabanlı çözümler üretebilecek

Ders hem **teorik kavramları** hem de **pratik uygulamaları** içerir. İlk yarıda temel konular işlenirken, ikinci yarıda ileri seviye konular ve proje geliştirme aşamaları yer alır.

Değerlendirme Kriterleri

Bileşen	Açıklama
Ara Sınav	Klasik ve test sınav formatı
Ödev ve Uygulama	Haftalık küçük uygulamalar ve ödevler
Final Projesi	Grup projesi + Sunum + Rapor

Ara Sınav Detayları

- İçerik:** Vizeye kadar olan teorik konular
- Format:** Kısa cevaplı sorular, kod parçaları yazma, ,istem tasarımları, test soruları
- Süre:** 60 dakika

Final Projesi Detayları

- Ekip:** 2-3 kişilik gruplar
- İstem mühendisliği ve RAG'ı birleştirecek ve bir gerçek dünya problemini çözecek uygulama

Haftalık Detaylı Plan

Hafta 1: Üretken Yapay Zeka'ya Giriş

Öğrenme Hedefleri:

- Yapay zeka, makine öğrenmesi ve derin öğrenme kavramlarını hatırlamak

- Üretken Yapay Zeka'nın (Generative AI) ayırt edici yapay zekadan (Discriminative AI) farkını anlamak
- BDM'lerin günlük hayatı uygulamalarını keşfetmek
- Dönüştürücü (Transformer) mimarisinin temel mantığını kavramak

Konu Başlıkları:

- Yapay Zeka tarihsel perspektif
- Ayırt Edici (Discriminative) vs Üretken (Generative) modeller
- Üretken Yapay Zeka çeşitleri (Metin, Görüntü, Ses, Video)
- BDM örnekleri: GPT-4, Claude, Gemini, LLaMA
- Dönüştürücü (Transformer) mimarisine giriş
- Uygulamalar: ChatGPT, GitHub Copilot, Midjourney

Ders İçi Aktiviteler:

- Farklı BDM'leri deneme ve karşılaştırma
- Grup tartışması: Türkiye'de üretken yapay zeka kullanım alanları
- Mini ödev: Seçilen bir BDM ile 5 farklı görev denemesi ve sonuçların analizi

Hafıza 2: Büyük Dil Modellerinin Çalışma Prensibi

Öğrenme Hedefleri:

- Dönüştürücü (Transformer) mimarisinin detaylarını öğrenmek
- Öz-dikkat (Self-attention) mekanizmasını anlamak
- Tokenizasyon (Tokenization) işleminin önemini kavramak
- BDM'lerin "şıradaki token (belirteç) tahmini" (next token prediction) mantığını anlamak

Konu Başlıkları:

- Transformer mimarisi detaylı inceleme
 - Encoder-Decoder yapısı
 - Çok başlı öz-dikkat (Multi-head self-attention)
 - Konum kodlama (Position encoding)
- Tokenizasyon yöntemleri (BPE, WordPiece, SentencePiece)
- Ön Eğitim (Pre-training) ve İnce Ayar (Fine-tuning) süreçleri
- Model boyutları ve parametreler (7B, 13B, 70B modeller)
- Sıcaklık (Temperature), Top-p, Top-k gibi hiperparametreler

Ders İçi Aktiviteler:

- Jupyter Notebook: Basit bir attention mekanizması implementasyonu
- Tokenizasyon deneyleri (farklı tokenizörler ile aynı metni tokenizasyon)
- Hiperparametre değişimlerinin çıktıya etkisini gözleme

Ödev:

- Bir dönüştürücü (transformer) modülünü açıklayan interaktif notebook tamamlama
- Sıcaklık ve Top-p parametrelerinin etkisini gösteren mini rapor

Hafta 3: Temel İstem Bileşenleri ve API Kullanımı

Öğrenme Hedefleri:

- Sistem, kullanıcı, asistan mesaj rollerini kullanabilmek
- Sıfır örnekli (Zero-shot), az örnekli (few-shot) ve düşünce zinciri (chain-of-thought) teknikleri uygulayabilmek
- OpenAI/Anthropic API'lerine bağlanabilmek
- Python ile temel BDM çağrıları yapabilmek

Konu Başlıkları:

- İstem anatomisi:
 - Sistem mesajı (model davranışını belirleme)
 - Kullanıcı mesajı (kullanıcı girdisi)
 - Asistan mesajı (model yanıtı)
- İstem bileşenleri:
 - Instructions (talimatlar)
 - Primary content (ana içerik)
 - Supporting content (destekleyici bilgi)
 - Output format (çıkıtı formatı)
- İstemleme stratejileri:
 - Sıfır örnekli (Zero-shot) öğrenme
 - Az örnekli (Few-shot) öğrenme (1-örnekli, 5-örnekli, vb.)
 - Düşünce Zinciri (CoT) istemleme
- API Entegrasyonu:
 - API anahtarı oluşturma
 - Environment variables kullanımı
 - Python SDK kurulumu (openai, anthropic)
 - Rate limiting ve token yönetimi

Uygulama:

- **API Setup:**
 - OpenAI API anahtarı alma
 - `.env` dosyası oluşturma
 - Python virtual environment kurulumu
- **İstem Denemeleri:**
 - Zero-shot: "Translate to Turkish: Hello World"
 - Few-shot: Duygu analizi için örnekler verip yeni cümle analizi
 - CoT: Matematiksel problem çözümünde adım adım düşünme

Ödev:

- Farklı istem stratejilerini karşılaştırılan Python scripti yazma
- Kendi seçtiğiniz bir görev için en etkili istemi bulma ve raporlama

Hafta 4: İleri İstem Teknikleri

Öğrenme Hedefleri:

- Öz-tutarlılık (Self-consistency), Düşünce Ağacı (Tree of Thoughts) gibi ileri teknikleri kullanabilmek
- İstem biçimlendirme (prompt formatting) ve yapılandırılmış en iyi pratiklerini uygulayabilmek
- Sınırlayıcı (delimiter) kullanımı ve enjeksiyon (injection) saldırılara karşı koruma
- Rol yapma (role-playing) ve kişiliğe dayalı (persona-based) istemleme

Konu Başlıkları:

- İleri İstemleme Teknikleri:
 - Öz-tutarlılık (Self-Consistency) (birden fazla akıl yürütme yolu)
 - Düşünce Ağacı (Tree of Thoughts - ToT)
 - Erişim tabanlı istemleme (Retrieval-based prompting)
 - En azdan çoğu (Least-to-Most) istemleme
- İstem Mühendisliği En İyi Uygulamalar:
 - Açık ve spesifik talimatlar
 - Sınırlayıcı kullanımı (###, "", XML etiketleri)
 - Çıktı formatı belirleme (JSON, Markdown, CSV)
 - Kısıtlamalar ve sınırlamalar belirleme
- İstem Enjeksiyonu ve Güvenlik:
 - İstem enjeksiyonu nedir?
 - Kısıtlama delme (Jailbreaking) örnekleri
 - Savunma mekanizmaları
- Rol Yapma ve Kişilik (Persona):
 - Uzman rolü verme (Python uzmanı gibi davran)
 - Ton ve stil ayarlama

Uygulama:

- Öz-tutarlılık ile matematik problemi çözme
- JSON ve XML formatında yapılandırılmış çıktı (structured output) alma
- İstem enjeksiyonu örnekleri ve savunma mekanizmaları deneme

Ödev:

- Seçtiğiniz bir alan için (hukuk, tıp, finans) uzman persona istemi tasarlama
- ToT tekniği ile kompleks bir problem çözme

Hafta 5: Çok Modlu Büyük Dil Modelleri

Öğrenme Hedefleri:

- Görsel-Dil (Vision-Language) modelleri anlama (GPT-4V, Gemini Vision)
- Görüntüden metne (Image-to-text) ve metinden görüntüye (text-to-image) görevlerini kullanabilme
- Çok modlu istem mühendisliği stratejileri

Konu Başlıkları:

- Çok Modlu Yapay Zeka'ya (Multimodal AI) Giriş
- Görsel-Dil Modelleri:
 - OpenAI GPT
 - Google Gemini
 - Claude Opus
 - LLaVA
- Görüntüden Metne Görevleri:
 - Görüntü betimleme
 - Görsel Soru Cevaplama (VQA)
 - OCR ve belge ayrıştırma
 - Grafik/Tablo analizi
- Text-to-Image Modeller (kısa giriş):
 - DALL-E 3
 - Stable Diffusion
 - Midjourney
- Çok Modlu İstem Stratejileri:
 - Görüntü + metin birleştirme
 - Görsellerle çok adımlı akıl yürütme
 - Görsel düşünce zinciri

Uygulama:

- GPT-4V, Gemini API kullanarak görüntü analizi
- Grafik/tablodan veri çıkarma
- Belge ayrıştırma (fatura, fiş)
- Görsel Soru Cevaplama denemeleri

Ödev:

- Çok modlu bir kullanım durumu belirleme ve prototip geliştirme (örn: Fatura okuma ve analiz sistemi)

Hafta 6: Yerel BDM Çatıları

Öğrenme Hedefleri:

- Açık kaynak BDM'leri yerel olarak çalıştırabilme
- Ollama, LM Studio, OpenRouter kullanabilme
- Hugging Face Transformers kütüphanesi ile model çalışma
- API maliyetlerinden kaçınma stratejileri

Konu Başlıklarları:

- Yerel BDM'lerin Avantajları:
 - Veri gizliliği ve güvenlik
 - API maliyetlerinden kaçınma
 - İnternet bağımlılığı olmama
 - Tam kontrol ve özelleştirme
- Yerel BDM Araçları:

- **Ollama:** Komut satırı arayüzü, model yönetimi
- **LM Studio:** GUI tabanlı, kullanıcı dostu
- **OpenRouter:** Bulut tabanlı BDM aracı
- Hugging Face Ecosystem:
 - Transformers kütüphanesi
 - Pipeline API
 - Model Hub'dan model indirme
 - Nicemleme (Quantization) (4-bit, 8-bit)
- Popüler Açık Kaynak Modeller:
 - LLaMA
 - Mistral
 - Phi
 - Gemma
 - Qwen

Uygulama:

1. Ollama kullanımı
2. LM Studio Kullanımı
3. HuggingFace Kullanımı
4. OpenRouter kullanımı

Ödev:

- Yerel bir model kurarak kendi kullanım durumunuz için performans testi

Hafta 7: Ara Sınav (Vize)

Sınav Kapsamı:

- Hafta 1-6 arası tüm teorik konular
- BDM temel konseptleri ve mimari
- İstemleme teknikleri
- API kullanımı
- Çok modlu modeller
- Yerel BDM'ler

Sınav Formatı:

- Teorik sorular
- İstem tasarımlı soruları
- Kod yazma/okuma soruları
- Test soruları

Hafta 8: Retrieval-Augmented Generation (RAG) - Temeller

Öğrenme Hedefleri:

- RAG konseptini ve neden gerekli olduğunu anlamak

- Vektör veritabanı (vector database) ve gömmeleri (embeddings) kavramak
- Basit bir RAG pipeline'ı kurabilmek
- Belge parçalama (document chunking) stratejileri

Konu Başlıklarları:

- RAG'e Neden İhtiyaç Var?
 - BDM'lerin bilgi kesintisi (knowledge cutoff) problemi
 - Sanrı (Hallucination) azaltma
 - Alana özgü (Domain-specific) bilgi ekleme
 - Gerçek zamanlı bilgi entegrasyonu
- RAG Mimarisi:
 - Belge alımı (Document ingestion)
 - Metin gömme (Text embedding - vektör temsili)
 - Vektör veritabanı depolama
 - Anlamsal arama (Semantic search)
 - Bağlam ekleme (Context injection)
 - Yeniden sıralama (Reranking)
 - Anahtar Kelime Araması (BM25)
 - BDM üretimi (LLM generation)
- Gömme Modelleri:
 - OpenAI text-embedding-ada-002
 - Sentence Transformers
 - all-MiniLM-L6-v2
 - Google embeddinggemma-300m
- Vektör Veritabanları:
 - Pinecone
 - Weaviate
 - Chroma
 - FAISS
 - Qdrant
- Belge Parçalama (Chunking):
 - Sabit boyutlu parçalama
 - Cümle tabanlı parçalama
 - Anlamsal parçalama
 - Parça örtüşme (Chunk overlap) stratejileri

Uygulama:

1. **RAG Pipeline Kurulumu:**
2. **Farklı Chunking Stratejileri Deneme:**
 - 500, 1000, 2000 token chunk sizes
 - Overlap 0%, 10%, 20%
 - Performans karşılaştırması

Hafıza 9: RAG İleri Konular

Öğrenme Hedefleri:

- Sorgu dönüşümü ve genişletme teknikleri
- Yeniden sıralama (Re-ranking) ve filtreleme stratejileri
- Çoklu sorgu ve füzyon erişimi
- RAG optimizasyonu

Konu Başlıkları:

- RAG Optimizasyon Teknikleri:
 - **Sorgu Dönüşümü (Query Transformation):**
 - Sorgu genişletme (Query expansion)
 - Sorgu yeniden yazma (Query rewriting)
 - Varsayımsal Belge Gömmeleri (HyDE)
 - Geç Parçalama (Late Chunking)
 - **Erişim Geliştirme (Retrieval Enhancement):**
 - Çoklu sorgu erişimi (Multi-query retrieval)
 - Füzyon erişimi (hibrit arama: yoğun + seyrek)
 - **Yeniden Sıralama (Re-ranking):**
 - Cohere Rerank
 - Cross-encoder models
 - MMR (Maksimum Marjinal Farklılık)
 - **Filtreleme:**
 - Metadata filtreleme
 - Zaman tabanlı filtreleme
 - Alaka eşiği (Relevance threshold)
- İleri RAG Mimarileri:
 - Naive RAG
 - İleri RAG (sorgu dönüşümü ile)
 - Modüler RAG (bileşen tabanlı)
 - Etmen tabanlı RAG (araç kullanan etmenler)
- RAG Değerlendirme:
 - Erişim metrikleri (Kesinlik, Duyarlılık, MRR)
 - Üretim metrikleri (Sadakat, Alaka)
 - RAGAS framework

Uygulama:

- HyDE implementasyonu
- Multi-query retrieval
- Yeniden Sıralama (Rerank) Kullanımı
- Farklı RAG pattern'lerinin karşılaştırılması

Hafta 10: BDM Ajanları

Öğrenme Hedefleri:

- Ajan kavramını ve mimarisini anlamak
- Araç/Fonksiyon çağrıma (Tool/Function calling) kullanabilmek
- ReAct deseni ile ajan geliştirmek

- LangChain ajanları kullanabilmek

Konu Başlıkları:

- BDM Ajanı Nedir?
 - Otonom karar verme
 - Araç kullanımı / Fonksiyon çağırma
 - Çok adımlı akıl yürütme
 - Hafıza ve durum yönetimi
- Ajan Mimarileri:
 - **ReAct (Akıl yürütme + Eylem):**
 - Düşünce → Eylem → Gözlem döngüsü
 - Düşünce zinciri + araç kullanımı
 - **Planla-ve-Yürüt (Plan-and-Execute):**
 - İlk önce plan yap, sonra yürüt
 - **Yansıma (Reflection):**
 - Öz-eleştiri ve yinelemeli iyileştirme
- Araç/Fonksiyon Çağırma:
 - OpenAI Fonksiyon Çağırma API
 - Araç tanımı ve şeması
 - Araç yürütme (Tool execution)
- Ajan Çatıları (Agent Frameworks):
 - LangChain, LangFlow
 - Microsoft AutoGen
 - CrewAI
- Ajan Kullanım Durumları:
 - Kod üretim ajanları
 - Araştırma asistanları
 - Müşteri destek botları
 - Veri analizi ajanları

Uygulama:

1. Basit Araç Çağırma:
2. ReAct Ajanı:

Hafta 11: BDM Güvenliği ve Etik

Öğrenme Hedefleri:

- BDM güvenlik risklerini anlama
- İstem enjeksiyonu saldırılarını tanıma ve önleme
- Yanlılık ve adililik konularını kavrama
- Etik kullanım prensipleri

Konu Başlıkları:

- BDM'ler için OWASP İlk 10:
 - LLM01: İstem Enjeksiyonu (Prompt Injection)

- LLM02: Güvensiz Çıktı İşleme
 - LLM03: Eğitim Verisi Zehirleme
 - LLM04: Model Hizmet Reddi
 - LLM05: Tedarik Zinciri Güvenlik Açıkları
 - LLM06: Hassas Bilgi İfşası
 - LLM07: Güvensiz Eklenti Tasarımı
 - LLM08: Aşırı Yetkilendirme
 - LLM09: Aşırı Güven
 - LLM10: Model Hırsızlığı
- İstem Enjeksiyonu Detayları:
 - Doğrudan enjeksiyon
 - Dolaylı enjeksiyon (zehirli veri)
 - Kısıtlama delme (Jailbreaking)
 - Savunma mekanizmaları
 - Yanlılık ve Adillik (Bias ve Fairness):
 - Cinsiyet yanlılığı
 - Irksal yanlılık
 - Kültürel yanlılık
 - Azaltma stratejileri
 - Gizlilik ve Veri Koruma:
 - Kişisel Tanımlanabilir Bilgi (PII) sızıntısı
 - Eğitim verisi ezberleme
 - Diferansiyel gizlilik
 - Etik Kullanım:
 - Academic integrity
 - Deepfake ve yanlış bilgi
 - Telif hakkı ve atıf
 - Sorumlu Yapay Zeka prensipleri

Uygulama:

- İstem enjeksiyonu örnekleri ve tespit
- Yanlılık testi (cinsiyet, ırk, din)
- PII tespiti ve filtreleme
- İçerik denetimi API kullanımı

Hafta 12: Güvenilirlik ve Performans Değerlendirme

Öğrenme Hedefleri:

- BDM çıktılarını değerlendirme metrikleri
- Sanrı (hallucination) tespiti
- Kiyaslama testleri (Benchmarks) ve değerlendirme çatıları
- A/B testi ve insan değerlendirme

Konu Başlıkları:

- BDM Değerlendirme Zorlukları:

- Öznel vs nesnel metrikler
- Göreve özgü değerlendirme
- İnsan değerlendirmesi maliyeti
- Değerlendirme Metrikleri:
 - **Otomatik Metrikler (Automatic Metrics):**
 - BLEU, ROUGE (text generation)
 - Exact Match, F1 (QA tasks)
 - Perplexity
 - BERTScore
 - **Yargıcı Olarak BDM (LLM-as-Judge):**
 - GPT-4 ile otomatik değerlendirme
 - Rubrik tabanlı puanlama
 - **İnsan Değerlendirmesi (Human Evaluation):**
 - Alaka, tutarlılık, akıcılık
 - Olgu doğruluğu (Factual accuracy)
 - Kullanıcı memnuniyeti
- Sanrı Tespiti (Hallucination Detection):
 - Doğruluk kontrol yöntemleri
 - Güven puanlaması (Confidence scoring)
 - Alıntılama ve temellendirme (Citation ve grounding)
- Kiyaslama Testleri (Benchmarks):
 - MMLU (Massive Multitask Language Understanding)
 - HellaSwag
 - TruthfulQA
 - HumanEval (code generation)
 - LMSYS Chatbot Arena
- Testing Frameworks:
 - PromptFlow
 - LangSmith
 - Phoenix
 - Weights & Biases

Uygulama:

- BLEU/ROUGE hesaplama
- GPT-4 ile otomatik değerlendirme
- Sanrı tespiti denemeleri
- A/B testing için framework kurulumu

Hafta 13: Proje Sunumları - 1

Amaç:

Grupların geliştirdiği projeleri sınıfı sunması ve teknik detayları paylaşması.

Sunum Formatı:

- **Süre:** Grup başına 15-20 dakika (10-12 dk sunum + 5-8 dk soru-cevap)
- **İçerik:**

1. Problem tanımı ve motivasyon
 2. Kullanılan teknolojiler ve araçlar
 3. Sistem mimarisi (diyagram)
 4. İstem mühendisliği stratejileri
 5. Canlı demo
 6. Karşılaşılan zorluklar ve çözümler
 7. Sonuçlar
-

Hafta 14: Proje Sunumları - 2 ve Ders Özeti

Amaç:

- Kalan grupların sunumları
- Dönem boyunca öğrenilenlerin özeti
- Gelecek trendleri ve kariyer fırsatları

Ek Konular:

- Trendler:
 - Daha uzun bağlam pencereleri (100K+ tokens)
 - Çok modlu BDM'ler
 - Uç BDM'ler (Edge LLMs - cihaz üzeri YZ)
 - Özelleşmiş alan BDM'leri
 - YZ ajanları ve otonom sistemler
 - Kariyer Fırsatları:
 - İstem Mühendisi (Prompt Engineer)
 - BDM Uygulama Geliştiricisi
 - YZ Güvenlik Araştırmacısı
 - ML Mühendisi (LLMops)
 - Yararlı Kaynaklar:
 - Takip edilmesi gereken araştırmacılar
 - Açık kaynak projeler
 - Online topluluklar
-

Proje Gereksinimleri

Proje Teslim Edilecekler

1. **Çalışan Kod:** GitHub repository (README dahil)
2. **Dokümantasyon:**
 - Sistem mimarisi
 - Kurulum talimatları
 - Kullanım kılavuzu
 - İstem mühendisliği stratejileri
3. **Demo Video:** 3-5 dakika (YouTube/Loom)
4. **Sunum:** PDF formatında
5. **Nihai Rapor:** 5-10 sayfa LaTeX/Word

Proje Önerileri (Örnek Konular)

- **RAG Tabanlı Soru-Cevap Sistemi:** Üniversite ders notları üzerine
- **Kod Gözden Geçirme Asistanı:** Pull request'leri analiz eden ajan
- **Veri Analizi Sohbet Robotu:** SQL sorguları üreten ve çalıştırın sistem
- **İçerik Üreticisi:** Blog yazısı, sosyal medya içeriği oluşturma
- **Akıllı Eğitim Asistanı:** Öğrenci sorularını yanıtlayan ve quiz oluşturan sistem
- **Medikal Belge Ayırıştırıcı:** Tahlil/rapor okuma ve özetleme
- **E-ticaret Ürün Önerisi:** Kullanıcı tercihlerine göre öneri
- **E-posta Asistanı:** Otomatik email kategorize etme ve yanıt önerisi
- **Çok Dilli Çevirmen:** Bağlam farkındalıklı çeviri sistemi
- **Akıllı Görev Yöneticisi:** Doğal dilde görev yönetimi

Kullanılacak Araçlar ve Teknolojiler

Bulut BDM API'leri

- OpenAI GPT Modelleri
- Anthropic Claude Modelleri
- Google Gemini Modelleri

Açık Kaynak BDM'ler

- Meta LLaMA
- Mistral
- Microsoft Phi
- Google Gemma
- Alibaba Qwen

Yerel Çalıştırma Araçları

- **Ollama** - CLI tabanlı model yönetimi
- **LM Studio** - GUI desktop uygulaması

Vektör Veritabanları

- Chroma
- Pinecone
- Weaviate
- Qdrant
- FAISS

Ana Kaynaklar

Resmi Dokümantasyonlar

1. **OpenAI Dokümantasyonu** - <https://platform.openai.com/docs>
2. **Anthropic Claude Docs** - <https://docs.anthropic.com>

3. **Google Gemini API** - <https://ai.google.dev/docs>
4. **Hugging Face Transformers** - <https://huggingface.co/docs/transformers>

Online Kurslar ve Rehberler

1. **LearnPrompting.org** - Kapsamlı istem mühendisliği rehberi
2. **PromptingGuide.ai** - İstem mühendisliği en iyi uygulamalar
3. **DeepLearning.AI - ChatGPT Prompt Engineering** (Andrew Ng)
4. **Microsoft Learn - Prompt Engineering**

Akademik Makaleler

1. Attention is All You Need (Vaswani et al., 2017) - Transformer
2. Language Models are Few-Shot Learners (Brown et al., 2020) - GPT-3
3. Chain-of-Thought Prompting (Wei et al., 2022)
4. ReAct: Reasoning and Acting (Yao et al., 2023)
5. Tree of Thoughts (Yao et al., 2023)
6. Retrieval-Augmented Generation (Lewis et al., 2020)

Kitaplar

1. "**Prompt Engineering for Generative AI**" - James Phoenix & Mike Taylor
2. "**Build a Large Language Model (From Scratch)**" - Sebastian Raschka

Blog ve Newsletter'lar

- Hugging Face Blog
- OpenAI Blog
- Anthropic Research
- Sebastian Raschka's Newsletter

Ders Çıktıları

Bu dersi başarıyla tamamlayan öğrenciler:

1. Büyük Dil Modellerinin temel çalışma prensiplerini açıklayabilir
2. Farklı problem türleri için etkili istemler tasarlayabilir
3. BDM API'lerine entegre Python uygulamaları geliştirebilir
4. Yerel ve bulut tabanlı BDM'leri kullanabilir
5. RAG sistemleri tasarlayıp geliştirebilir
6. BDM tabanlı etmenler geliştirebilir
7. BDM güvenlik risklerini tanıyabilir ve önlem alabilir
8. BDM çıktılarını değerlendirebilir ve optimize edebilir
9. Gerçek dünya problemlerine BDM tabanlı çözümler üretebilir
10. BDM etik ve sorumlu Yapay Zeka prensiplerini uygulayabilir

Son Güncelleme: Ocak 2026
Versiyon: 2.0

Bu müfredat, 4. sınıf Bilgisayar Mühendisliği öğrencilerinin modern BDM teknolojilerini hem teorik hem pratik açıdan derinlemesine öğrenmelerini sağlamak üzere tasarlanmıştır. İçerik, sektörün güncel ihtiyaçları ve akademik standartlar göz önünde bulundurularak hazırlanmıştır.

Sorularınız için: asimyuksel@sdu.edu.tr