NLP Kütüphaneleri ve Terimler Rehberi

📚 Kütüphaneler

TikToken

- **Tanım**: OpenAI tarafından geliştirilen hızlı tokenizer kütüphanesi
- **Kullanım Alanı**: GPT modelleri için tokenization işlemleri
- **Avantajları**: Yüksek performans, OpenAI modelleriyle tam uyumluluk
- **Desteklenen Encoding'ler**:
  - `gpt2`: GPT-2 modeli için (50,257 vocab)
  - `cl100k_base`: GPT-3.5/4 için (100,277 vocab)
  - `p50k_base`: Code modelleri için

Transformers (Hugging Face)

- **Tanım**: En popüler açık kaynak NLP kütüphanesi
- **Kullanım Alanı**: Hazır eğitilmiş modelleri kullanma ve fine-tuning
- **Desteklenen Modeller**: BERT, GPT, T5, Gemma, LLaMA, Claude vs.
- **Ana Sınıflar**:
  - `AutoModel`: Model otomatik yükleme
  - `AutoTokenizer`: Tokenizer otomatik yükleme
  - `AutoProcessor`: Çoklu modalite (metin+görsel) işleme

PyTorch (Torch)

- **Tanım**: Facebook'un derin öğrenme framework'ü
- **Kullanım Alanı**: Neural network oluşturma, tensor işlemleri
- **Özellikler**: Dinamik computation graph, GPU desteği
- **Neden Gerekli**: Transformers kütüphanesi PyTorch backend'ini kullanır

Datasets (Hugging Face)

- **Tanım**: NLP veri setlerini yönetme kütüphanesi
- **Özellikler**: Otomatik caching, lazy loading, streaming
- **Popüler Dataset'ler**: IMDB, GLUE, Squad, Common Crawl

Matplotlib

- **Tanım**: Python'un temel veri görselleştirme kütüphanesi

- **Kullanım Alanı**: Grafik çizme, model performans analizi
- **Alternatifler**: Seaborn, Plotly, Bokeh

Hugging Face Hub

- **Tanım**: Hugging Face platformu ile etkileşim kütüphanesi
- **Özellikler**: Model upload/download, authentication, repository yönetimi
- **API Token**: Private modellere erişim için gerekli

🔤 Terimler ve Kavramlar

Tokenization (Token'laştırma)

- **Tanım**: Metni daha küçük parçalara (token) ayırma işlemi
- **Amaç**: Bilgisayarların metni işleyebilir hale getirme
- **Türleri**:
  - **Word-level**: Kelimelere ayırma (`"Hello world"` → `["Hello", "world"]`)
  - **Subword-level**: Alt kelimelere ayırma (`"running"` → `["run", "ning"]`)
  - **Character-level**: Karakterlere ayırma (`"Hi"` → `["H", "i"]`)

Vocabulary (Sözlük Dağarcığı)

- **Tanım**: Model tarafından bilinen tüm token'ların listesi
- **Vocabulary Size**: Token sayısı (örn: GPT-2'de 50,257)
- **Trade-off**:
  - Büyük vocab = Daha detaylı anlama + Daha fazla memory
  - Küçük vocab = Daha az memory + Daha az detaylı anlama

Token ID

- **Tanım**: Her token'a atanan benzersiz sayısal kimlik
- **Örnek**: `"The"` → `464`, `"cat"` → `2574`
- **Amaç**: Bilgisayarların metinle matematiksel işlem yapabilmesi

Encoding (Kodlama)

- **Tanım**: Metni token ID'lere çevirme işlemi
- **Örnek**: `"Hello"` → `[15496]`
- **Metod**: `tokenizer.encode(text)`

Decoding (Kod Çözme)

- **Tanım**: Token ID'leri tekrar metne çevirme işlemi
- **Örnek**: `[15496]` → `"Hello"`
- **Metod**: `tokenizer.decode(token_ids)`

Attention Mechanism

- **Tanım**: Modelin farklı token'lara ne kadar odaklanacağını belirleme
- **Türleri**:
  - **Self-Attention**: Token'ların birbirleriyle ilişkisi
  - **Cross-Attention**: Farklı sequence'lar arası ilişki
  - **Multi-Head Attention**: Paralel attention hesaplaması

Context Window (Bağlam Penceresi)

- **Tanım**: Modelin bir seferde işleyebileceği maksimum token sayısı
- **Örnekler**:
  - GPT-3.5: 4,096 token
  - GPT-4: 8,192-32,768 token
  - Claude-3: 200,000 token
- **Etki**: Uzun metinlerde performans sınırı

🧠 Model Türleri ve Mimariler

Encoder-Only Modeller

- **Örnekler**: BERT, RoBERTa, DeBERTa
- **Kullanım**: Classification, sentiment analysis, NER
- **Özellik**: Bidirectional (çift yönlü) attention

Decoder-Only Modeller

- **Örnekler**: GPT serisi, LLaMA, Gemma
- **Kullanım**: Text generation, chat, completion
- **Özellik**: Causal (nedensel) attention, autoregressive

Encoder-Decoder Modeller

- **Örnekler**: T5, BART, mT5
- **Kullanım**: Translation, summarization, Q&A
- **Özellik**: Hem encoding hem decoding katmanları

Vision-Language Modeller

- **Örnekler**: CLIP, LayoutLM, BLIP
- **Kullanım**: Image captioning, visual Q&A, OCR
- **Özellik**: Görsel ve metin verilerini birlikte işler

⚙️ Eğitim ve Optimizasyon

Pre-trained Model (Önceden Eğitilmiş Model)

- **Tanım**: Büyük veri setlerinde önceden eğitilmiş model
- **Avantaj**: Sıfırdan eğitim yerine transfer learning
- **Örnekler**: GPT-4, BERT, Gemma, LLaMA

Fine-tuning

- **Tanım**: Önceden eğitilmiş modeli belirli görev için uyarlama
- **Amaç**: Domain-specific performansı artırma
- **Yöntem**: Model ağırlıklarını küçük öğrenme oranıyla güncelleme
- **Türleri**:
  - **Full Fine-tuning**: Tüm parametreleri güncelleme
  - **LoRA**: Sadece ek katmanları eğitme
  - **Adapter**: Küçük adaptör katmanları ekleme

Learning Rate (Öğrenme Oranı)

- **Tanım**: Model ağırlıklarının ne hızda güncelleneceği
- **Tipik Değerler**: 1e-5 ile 1e-3 arası
- **Scheduler**: Eğitim boyunca learning rate'i değiştirme

Loss Function (Kayıp Fonksiyonu)

- **Cross-Entropy**: Classification görevleri için
- **MSE (Mean Squared Error)**: Regression görevleri için
- **Perplexity**: Language modeling kalitesi ölçümü

📊 Performans Metrikleri

BLEU Score

- **Kullanım**: Machine translation kalitesi
- **Aralık**: 0-100 (yüksek = iyi)

- **Hesaplama**: N-gram overlap ile reference metinler

ROUGE Score

- **Kullanım**: Text summarization kalitesi
- **Türleri**: ROUGE-1, ROUGE-2, ROUGE-L
- **Hesaplama**: Recall ve precision kombinasyonu

Perplexity

- **Kullanım**: Language model kalitesi
- **Formül**: 2^(cross-entropy)
- **Yorum**: Düşük perplexity = iyi model

F1 Score

- **Kullanım**: Classification görevleri
- **Hesaplama**: 2 × (precision × recall) / (precision + recall)
- **Balanced**: Precision ve recall dengesini sağlar

🛠️ Pratik Uygulama Teknikleri

Prompt Engineering

- **Tanım**: Model'den istenen çıktıyı almak için giriş metnini optimize etme
- **Teknikler**:
  - **Zero-shot**: Örnek vermeden görev tanımı
  - **Few-shot**: Az sayıda örnek verme
  - **Chain-of-Thought**: Adım adım düşünme sürecini gösterme

RAG (Retrieval-Augmented Generation)

- **Tanım**: Dış bilgi kaynaklarından bilgi alarak text generation
- **Bileşenler**:
  - **Retriever**: İlgili belgeleri bulma
  - **Generator**: Bulunan bilgiyle metin üretme
- **Avantaj**: Model'in bilgi tabanı genişletme

Quantization

- **Tanım**: Model ağırlıklarını daha az bit ile temsil etme
- **Türleri**:

- **8-bit**: Hafıza kullanımını ~50% azaltır
- **4-bit**: Hafıza kullanımını ~75% azaltır

- **Amaç**: Memory kullanımını azaltma, inference hızlandırma

Gradient Accumulation

- **Tanım**: Küçük batch'leri biriktirip büyük batch etkisi yaratma
- **Kullanım**: Memory sınırı olduğunda
- **Formül**: Effective batch size = batch_size × accumulation_steps

---

🔧 Optimizasyon Teknikleri

Adam Optimizer

- **Tanım**: Adaptive learning rate ile momentum kombinasyonu
- **Parametreler**: lr, betas, weight_decay
- **Avantaj**: Çoğu NLP görevi için stabil sonuçlar

AdamW

- **Tanım**: Weight decay düzeltmesi yapılmış Adam
- **Kullanım**: Transformer modelleri için önerilen
- **Fark**: Weight decay'i gradient'ten ayırır

Warmup

- **Tanım**: Learning rate'i yavaşça artırıp sonra azaltma
- **Amaç**: Eğitimin başında instabiliteyi önleme
- **Tipik**: İlk %10 adımda warmup

Gradient Clipping

- **Tanım**: Gradient'lerin belirli değeri aşmasını engelleme
- **Amaç**: Exploding gradient problemini çözme
- **Değer**: Genellikle 1.0 civarı

📱 Deployment ve Serving

ONNX (Open Neural Network Exchange)

- **Tanım**: Model formatları arası dönüşüm standardı
- **Avantaj**: Framework bağımsız deployment

- **Desteklenen**: PyTorch, TensorFlow, Transformers

TensorRT

- **Tanım**: NVIDIA'nın inference optimizasyon kütüphanesi
- **Amaç**: GPU'da maksimum hız için optimizasyon
- **Kullanım**: Production deployment'larda

TorchScript

- **Tanım**: PyTorch modellerini C++ ortamında çalıştırma
- **Avantaj**: Python dependency'si olmadan serving
- **Metod**: torch.jit.script() veya torch.jit.trace()

Model Parallelism

- **Tanım**: Büyük modeli birden fazla GPU'ya bölme
- **Türleri**:
  - **Pipeline Parallelism**: Katmanları farklı GPU'lara
  - **Tensor Parallelism**: Ağırlık matrislerini bölme

🧪 Gelişmiş Konular

Knowledge Distillation

- **Tanım**: Büyük modelin bilgisini küçük modele aktarma
- **Amaç**: Performansı koruyarak model boyutunu küçültme
- **Yöntem**: Teacher-student training paradigması

Reinforcement Learning from Human Feedback (RLHF)

- **Tanım**: İnsan geri bildiriminden öğrenme
- **Kullanım**: ChatGPT, Claude gibi modellerin eğitimi
- **Aşamalar**: SFT → Reward Model → PPO

Constitutional AI

- **Tanım**: AI sisteminin davranışlarını prensiplerle sınırlama
- **Amaç**: Güvenli ve faydalı AI oluşturma
- **Yöntem**: Self-critique ve revision döngüleri

Mixture of Experts (MoE)

- **Tanım**: Farklı uzmanlık alanları için ayrı model parçaları

- **Avantaj**: Parametreleri artırmadan kapasiteyi genişletme
- **Örnekler**: Switch Transformer, GLaM

📈 Evaluation ve Benchmarking

GLUE Benchmark

- **Tanım**: 9 farklı NLP görevinden oluşan değerlendirme seti
- **Görevler**: Sentiment, similarity, inference, QA
- **Kullanım**: Genel NLP performansını ölçme

SuperGLUE

- **Tanım**: GLUE'nun daha zor versiyonu
- **Amaç**: İnsan seviyesi performansı test etme
- **Görevler**: Reading comprehension, reasoning

HellaSwag

- **Tanım**: Commonsense reasoning değerlendirmesi
- **Format**: Cümle tamamlama görevi
- **Zorluk**: İnsan seviyesi performans gerektirir

TruthfulQA

- **Tanım**: Modelin doğru bilgi verme kabiliyetini test etme
- **Amaç**: Hallucination ve misinformation tespiti
- **Format**: Açık uçlu sorular

🔍 Debugging ve Monitoring

Attention Visualization

- **Tanım**: Model'in hangi token'lara odaklandığını görselleştirme
- **Araçlar**: BertViz, Captum
- **Amaç**: Model davranışını anlama

Gradient Analysis

- **Tanım**: Gradient akışını analiz etme
- **Sorunlar**: Vanishing/exploding gradients
- **Çözümler**: Residual connections, normalization

Loss Curves

- **Tanım**: Training ve validation loss'unun izlenmesi
- **Signaller**:
  - **Overfitting**: Validation loss artmaya başlar
  - **Underfitting**: Her iki loss da yüksek kalır
  - **Good fit**: Her iki loss da stabil azalır

Perplexity Tracking

- **Tanım**: Language modeling kalitesinin sürekli izlenmesi
- **Kullanım**: Training progress ve model comparison
- **Hedef**: Düşük ve stabil perplexity

Bu rehber, modern NLP'de tokenization'ın temellerini ve pratik uygulamalarını kapsamaktadır. Her konu detaylı açıklanmış ve güncel teknolojiler dahil edilmiştir.