VERİ BİLİMİ DERSİ

NLP ve Hugging Face

Hafta 13 · Modül 13
Doğal Dil İşleme Temelleri ve Hugging Face Ekosistemi

Dr. Murat Altun

Veri Bilimi ve Yapay Zekâ Eğitimi · 2026

6
Saat

3
Notebook

1M+
HF Model

İçindekiler

Doğal Dil İşleme (NLP) Nedir?

BÖLÜM 1

Tanım

NLP (Natural Language Processing), bilgisayarların insan dilini anlama, yorumlama ve üretme yeteneğidir. Yapay zekânın en hızlı büyüyen alt alanlarından biridir.

$43B
NLP Pazar Büyüklüğü (2025)

%25+
Yıllık Büyüme Oranı

Chatbot & Asistan

Siri, Alexa, ChatGPT

Makine Çevirisi

Google Translate, DeepL

Metin Özetleme

Haber, makale, belge özetleri

Duygu Analizi

Sosyal medya, müşteri yorumları

NLP Tarihçesi

1950 Turing Testi
1966 ELIZA — ilk chatbot
2013 Word2Vec devrimi
2017 Transformer mimarisi
2018 BERT — bidirectional
2022 ChatGPT — LLM çağı

Metin Ön İşleme Pipeline

1. **Tokenization**
   Metni kelime veya alt-kelime parçalarına ayırma

2. **Lowercasing**
   Tüm karakterleri küçük harfe çevirme

3. **Stop Words**
   Anlam taşımayan kelimeleri çıkarma (ve, bir, ile...)

4. **Stemming**
   Kelimeleri kök formuna indirgeme (koşuyordum → koş)

5. **Lemmatization**
   Sözlük temelli kök bulma (better → good)

Tokenization Yöntemleri

Word Tokenization

Metni boşluk ve noktalama işaretlerine göre ayırır. Basit ama etkili yöntem.

```
from nltk.tokenize import word_tokenize
tokens = word_tokenize("NLP çok güçlü bir alan")
# ['NLP', 'çok', 'güçlü', 'bir', 'alan']
```

Subword Tokenization

BPE, WordPiece gibi yöntemlerle kelimeleri alt parçalara ayırır. BERT ve GPT bu yöntemi kullanır.

```
from transformers import AutoTokenizer
tok = AutoTokenizer.from_pretrained("bert-base-uncased")
# ['un', '##believ', '##able']
```

Karşılaştırma

Özellik	Word Tokenization	Subword Tokenization
Bilinmeyen kelimeler	OOV (tanımsız)	Alt parçalara ayırır
Sözlük boyutu	Çok büyük	Kompakt (~30K)
Kullanım	NLTK, spaCy	BERT, GPT, T5

Bag of Words (Kelime Torbası)

BoW Yaklaşımı

Her belgeyi, içerdiği kelimelerin frekanslarıyla temsil eder. Kelime sırası dikkate alınmaz. Basit ama güçlü bir başlangıç noktasıdır. Metin sınıflandırma ve bilgi çıkarma için yaygın kullanılır.

BoW Matris Örneği

Belge	nlp	güçlü	alan	model
D1	1	1	1	0
D2	1	0	0	1
D3	0	1	1	1

```
from sklearn.feature_extraction.text import CountVectorizer

corpus = [
    "NLP güçlü bir alan",
    "NLP model eğitimi",
    "güçlü alan model"
]

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(corpus)
print(X.toarray())
```

TF-IDF: Kelime Önem Ağırlıklandırma

TF-IDF Formülü

$$TF\text{-}IDF(t,d) = TF(t,d) \times IDF(t)$$

TF (Term Frequency): Kelimenin belgede kaç kez geçtiği
IDF (Inverse Document Frequency): log(Toplam belge / Kelimeyi içeren belge)

Neden TF-IDF?

- BoW'dan daha bilgilendirici
- Yaygın kelimelerin ağırlığını düşürür
- Nadir ama önemli kelimeleri öne çıkarır
- Metin sınıflandırma ve bilgi erişiminde standart

```
from sklearn.feature_extraction.text import TfidfVectorizer

corpus = [
    "makine öğrenmesi ile doğal dil işleme",
    "doğal dil işleme uygulamaları",
    "derin öğrenme modelleri"
]

tfidf = TfidfVectorizer()
matrix = tfidf.fit_transform(corpus)
print(tfidf.get_feature_names_out())
```

Word Embeddings: Kelimeler Vektör Uzayında

Word2Vec (2013 — Mikolov et al.)

Her kelimeyi sabit boyutlu bir vektörle temsil eder (genellikle 100-300 boyut).

Anlamsal ilişkileri yakalar:
kral − erkek + kadın ≈ kraliçe
Ankara − Türkiye + Fransa ≈ Paris

İki mimari:
- CBOW: Bağlamdan kelime tahmin
- Skip-gram: Kelimeden bağlam tahmin

2013 — **Word2Vec**
Statik, kelime düzeyi, hızlı eğitim

2014 — **GloVe**
Eş-oluşum matrisi tabanlı, global istatistik

2016 — **FastText**
Alt-kelime bilgisi, nadir kelimelerde güçlü

2018 — **ELMo**

```
from gensim.models import Word2Vec

model = Word2Vec(sentences, vector_size=100, window=5, min_count=1)
similar = model.wv.most_similar("yapay_zeka", topn=5)
```

BERT: Bidirectional Encoder Representations

BERT Nedir? (Google, 2018)

Transformer mimarisinin encoder kısmını kullanan, çift yönlü (bidirectional) dil modeli.

Pre-training görevleri:
1. Masked Language Model (MLM): Rastgele kelimeleri maskele, tahmin et
2. Next Sentence Prediction (NSP): İki cümle ardışık mı?

Fine-tuning ile her NLP görevine uyarlanabilir.

Parametre	110M / 340M
Katman	12 / 24
Gizli Boyut	768 / 1024
Attention Head	12 / 16

Transformer Avantajları

Self-Attention

Her kelime diğer tüm kelimelere bakabilir

Paralel İşleme

RNN'den farklı olarak sıralı değil, paralel eğitim

Transfer Learning

Bir kez eğit, her göreve uyarla (fine-tuning)

Hugging Face Ekosistemi

T

🤗 Transformers

PyTorch ve TensorFlow ile 100K+ model. pipeline() ile tek satırda NLP. AutoModel, AutoTokenizer sınıfları.

H

Model Hub

1M+ açık kaynak model. Filtrele, dene, indir. Topluluk katkıları ve model kartları.

D

Datasets

50K+ hazır veri seti. load_dataset() ile anında yükle. Streaming desteği büyük veriler için.

S

Spaces

Gradio ve Streamlit ile demo oluştur. Ücretsiz GPU/TPU. Model denemeleri ve paylaşım.

pipeline() ile Tek Satırda NLP

Hugging Face pipeline() fonksiyonu, model yükleme, tokenization ve inference işlemlerini tek satırda gerçekleştirir.

```
# Duygu Analizi
from transformers import pipeline

classifier = pipeline("sentiment-analysis")
result = classifier("Bu film harikaydı!")
```

```
# İsim Varlık Tanıma (NER)
ner = pipeline("ner",
  grouped_entities=
True)

result = ner("Ankara Türkiye'nin
  başkentidir.")
```

```
# Metin Özetleme
summarizer = pipeline("summarization")

summary = summarizer(long_text,
  max_length=130, min_length=30)
```

```
# Çeviri (EN → FR)
translator = pipeline("translation_en_to_fr")

result = translator(
  "Natural Language Processing
   is amazing!"
)
```

Duygu Analizi Uygulaması

Twitter/X Duygu Analizi Senaryosu

1. Twitter API ile tweet toplama
2. Metin ön işleme (temizleme, tokenization)
3. HF pipeline ile duygu sınıflandırma
4. Sonuçları görselleştirme (pasta, bar grafik)

%62	%28	%10
Pozitif	Negatif	Nötr

```
from transformers import pipeline
import pandas as pd

classifier = pipeline("sentiment-analysis")

tweets = pd.read_csv("tweets.csv")
tweets["duygu"] = tweets["text"].apply(
    lambda x: classifier(x[:512])[0]["label"]
)
print(tweets["duygu"].value_counts())
```

NER: İsim Varlık Tanıma

Named Entity Recognition (NER)

Metindeki özel isimleri (kişi, kurum, yer, tarih, miktar) otomatik olarak tespit eder ve kategorize eder.

PER	Kişi	Dr. Murat Altun
ORG	Kurum	Hugging Face, Google
LOC	Yer	Ankara, Türkiye

Örnek Çıktı

"Dr. Murat Altun Ankara'da Hugging Face eğitimi verdi." → PER | LOC | ORG

```
from transformers import pipeline

ner = pipeline("ner", grouped_entities=True)
entities = ner("Dr. Murat Altun Ankara'da eğitim verdi.")
for e in entities:
    print(f"{e['word']}: {e['entity_group']} ({e['score']:.2f})")
```

Metin Özetleme ve Çeviri

Abstractive Summarization

Orijinal metni yeniden ifade ederek özet üretir. Kopyalama değil, anlama ve yeniden yazma. T5, BART, Pegasus gibi modeller kullanılır.

```
# Metin Özetleme
from transformers import pipeline

summarizer = pipeline(
  "summarization",
  model=
"facebook/bart-large-cnn"
)

text = "Uzun makale metni..."
summary = summarizer(text,
  max_length=150)
```

Machine Translation

Bir dilden diğerine otomatik çeviri. Helsinki-NLP ve MarianMT modelleri. 500+ dil çifti destekli.

```
# Makine Çevirisi
from transformers import pipeline

translator = pipeline(
  "translation",
  model=
"Helsinki-NLP/
    opus-mt-en-tr"
)

result = translator(
  "Deep learning has
   transformed NLP."
)
```

Türkçe NLP: Zorluklar ve Çözümler

dbmdz/bert-base-turkish-cased

Türkçe BERT modeli (MDZ Digital Library). 35GB Türkçe metin ile eğitildi. Duygu analizi, NER, soru cevaplama ve metin sınıflandırmada İngilizce BERT'ten çok daha başarılı Türkçe sonuçlar.

Türkçe NLP Zorlukları

- Sondan eklemeli dil (agglutinative)
- Kelime sayısı çok fazla
- Serbest söz dizimi (SOV)
- Etiketli veri kıtlığı
- Özel karakter sorunları (İ/ı, Ş/ş)

Morphological Analysis

Zemberek kütüphanesi ile Türkçe morfolojik çözümleme. Kök bulma ve ek ayrıştırma.

Transfer Learning

Çok dilli modeller (mBERT, XLM-R) fine-tune ederek Türkçe'ye uyarlama.

Veri Artırma

Back-translation ve paraphrase ile Türkçe eğitim verisini zenginleştirme.

```
from transformers import pipeline

classifier = pipeline("sentiment-analysis", model="savasy/bert-base-turkish-sentiment-cased")
print(classifier("Bu ürün gerçekten harika!"))
```

SMS Spam Tespiti: NLP + ML

NLP + ML Pipeline

1. SMS verisi yükleme (ham/spam etiketli)
2. Metin temizleme ve ön işleme
3. TF-IDF ile özellik çıkarma
4. Naive Bayes / SVM / Random Forest ile sınıflandırma
5. Precision, Recall, F1-Score ile değerlendirme

%97	%95	5.5K
Accuracy	F1-Score	SMS Veri

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline
from sklearn.metrics import classification_report

# Pipeline: TF-IDF → Naive Bayes
model = make_pipeline(TfidfVectorizer(), MultinomialNB())
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

NLP Projeleri İş Akışı

1. **Veri Toplama** — Web scraping, API, hazır veri setleri. Metin verisi toplama ve etiketleme.
2. **Ön İşleme** — Temizleme, tokenization, stop words, lemmatization. Veri kalitesi her şeydir.
3. **Özellik Çıkarma** — BoW, TF-IDF, Word Embeddings. Metni sayısal vektörlere dönüştürme.
4. **Model Seçimi** — Geleneksel ML (NB, SVM) veya Transformer (BERT, GPT). Görevin karmaşıklığına göre karar.
5. **Değerlendirme** — Accuracy, Precision, Recall, F1-Score. Confusion matrix ve cross-validation.
6. **Deploy** — FastAPI / Flask + Hugging Face Spaces / Docker. Model servisi ve API oluşturma.

Haftalık Notebook'lar

1 Notebook 1: NLP Temelleri

hafta13_nlp_temel.ipynb

- NLTK ile tokenization ve stop words
- CountVectorizer ile Bag of Words
- TfidfVectorizer ile TF-IDF
- Kelime bulutu (WordCloud) görselleştirme

2 Notebook 2: Duygu Analizi

hafta13_duygu_analizi.ipynb

- HF pipeline ile İngilizce duygu analizi
- Türkçe BERT ile Türkçe duygu analizi
- Tweet veri seti ile toplu analiz
- Sonuçları Matplotlib ile görselleştirme

3 Notebook 3: Spam Tespiti

hafta13_spam_tespiti.ipynb

- SMS Spam Collection veri seti
- TF-IDF + Naive Bayes pipeline
- Model karşılaştırma (NB vs SVM vs RF)
- Classification report ve confusion matrix

Ödev ve Kaynaklar

Ödev 1: Duygu Analizi (1000+ Tweet)

- Twitter/X API veya hazır veri seti kullanın
- En az 1000 tweet toplayın
- HF pipeline ile duygu sınıflandırma
- Pozitif/negatif/nötr dağılımı görselleştirin
- En yaygın kelimeler için WordCloud
- Teslim: Jupyter Notebook + rapor

Ödev 2: SMS Spam Tespiti Raporu

- SMS Spam Collection veri setini indirin
- TF-IDF + en az 3 farklı sınıflandırıcı
- Performans karşılaştırma tablosu
- En iyi modelin confusion matrix'i
- Yanlış sınıflandırılan örnekleri analiz edin
- Teslim: Jupyter Notebook + rapor

Kaynaklar ve İleri Okuma

Hugging Face Docs huggingface.co/docs/transformers
NLTK Book nltk.org/book — Doğal Dil İşleme ile Python
Speech and Language Processing Jurafsky & Martin (3rd Ed.) — web.stanford.edu/~jurafsky/slp3
Türkçe NLP github.com/Firat-Yilmaz/TurkishNLPResources
Papers With Code paperswithcode.com/area/natural-language-processing

Hafta 13 — Özet

1. NLP, bilgisayarların insan dilini anlamasını sağlayan yapay zekânın en dinamik alanıdır.
2. Tokenization, BoW, TF-IDF ve Word Embeddings metin temsil yöntemlerinin temelini oluşturur.
3. BERT ve Transformer mimarisi, NLP'de devrim yaratarak transfer learning'i mümkün kıldı.
4. Hugging Face pipeline() ile tek satırda duygu analizi, NER, özetleme ve çeviri yapılabilir.
5. Türkçe NLP için özel modeller (bert-base-turkish) ve morfolojik analiz araçları gereklidir.

“Dil, düşüncenin giysisidir. — Samuel Johnson”

Dr. Murat Altun · Veri Bilimi ve Yapay Zekâ Eğitimi · 2026