

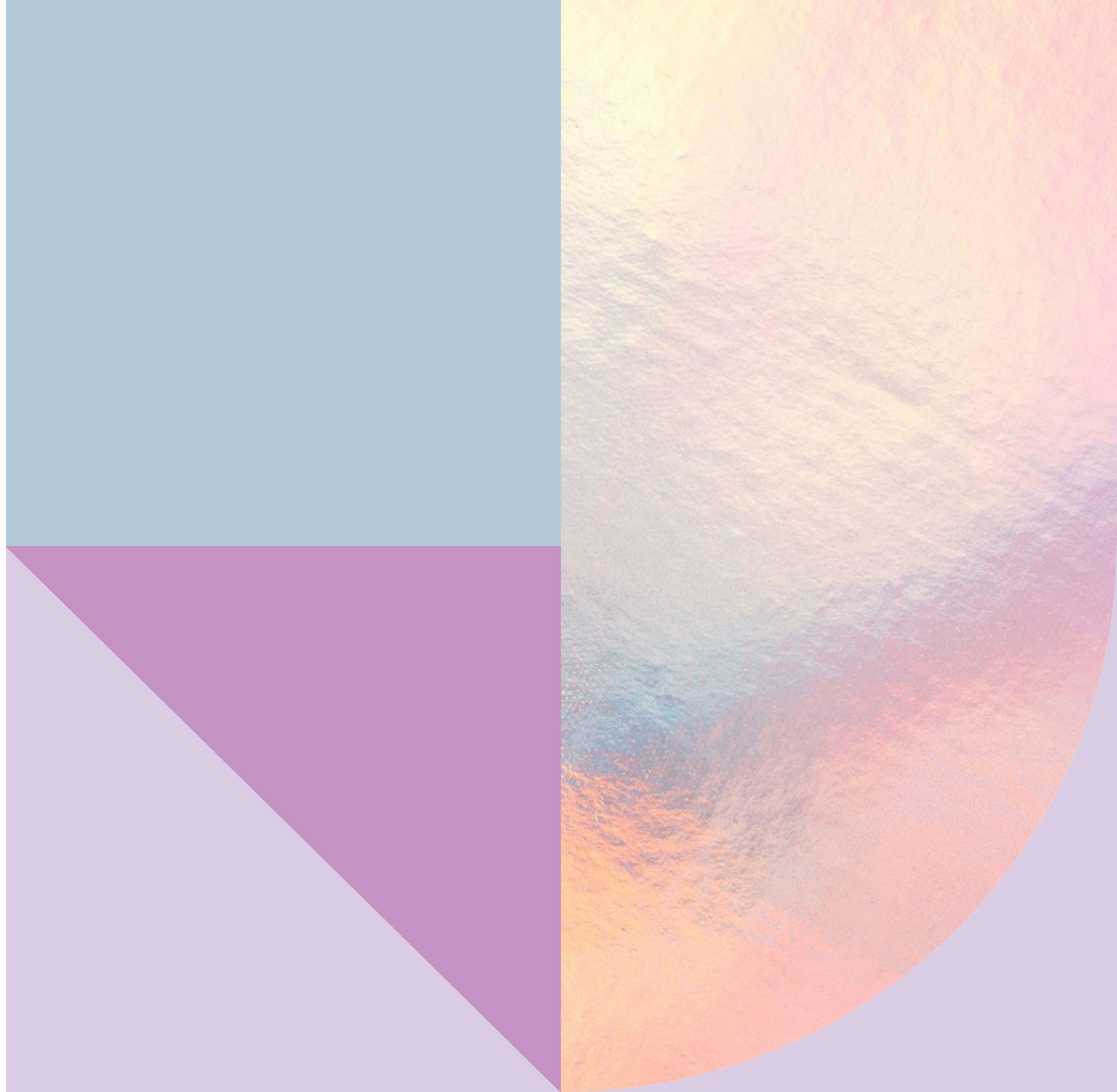
Doğal Dil İşleme ile Flair

Haticenur Yalman

20360859011

3.SINIF

BİLGİSAYAR MÜHENDİSLİĞİ



İçindekiler

- Doğal Dil İşleme
- Doğal Dil İşlemenin Tarihçesi
- Doğal Dil İşlemenin Uygulama Alanları
- Doğal Dil İşlemenin Temel Kavramları
- Flair nedir?
- Flair ile NLP Görevleri

Doğal Dil İşleme (NLP)

Doğal dil işleme (NLP), bilgisayarların insan dilini anlamasını ve üretmesini sağlayan bir yapay zeka alanıdır.

Metin sınıflandırma, duygu analizi, makine çevirisi ve metin özetleme gibi görevlerde kullanılır.

Tarihçe

- 1950'ler: Alan Turing'in "Makineler Düşünebilir mi?" sorusu ve Turing Testi.
- 1950'ler-90'lar: Bilgisayarların dili nasıl işlediğini aydınlatmayı hedefleyen kuralların kullanıldığı sistemlerdir.
- 1990'lar: İstatistiksel yöntemlerin yükselişi.
- 2000'ler ve Sonrası: Makine öğrenimi ve derin öğrenme ile büyük ilerlemeler.

Uygulama Alanları

1.Sosyal Medya İçerik Yönetimi:

- Spam Filtreleme
- Yorum Analizi

2. Sanal Asistanlar ve Chatbotlar

- Siri, Google Asistan, Alexa
- Müşteri Hizmetleri Chatbotları

3. Çeviri Hizmetleri

- Google Translate

4. Arama Motorları :

- Google Arama

5. Metin Düzenleme ve Yazma Yardımcıları

- Grammarly
- Otomatik Tamamlama

NLP'nin Temel Kavramları

Tokenizasyon

- Metinleri kelime, cümle veya karakter gibi daha küçük parçalara ayırma süreci.
- Kelime Bazlı Tokenizasyon: "Doğal dil işleme" -> ["Doğal", "dil", "işleme"]
- Cümle Bazlı Tokenizasyon: "Merhaba. Nasılsın?" -> ["Merhaba.", "Nasılsın?"]

Part-of-Speech (POS) Etiketi

- Kelimelerin dilbilgisel kategorilerini belirleme süreci (isim, fiil, sıfat vb.).
- Örnek: " Köpeğim havlıyor." -> [("Köpeğim" -> İsim,"havlıyor" -> Fiil)]

Ad Öbeği Tanıma (NER)

- Metindeki özel isimleri tanıma ve etiketleme süreci.
- Örnek: "Barack Obama, 1961'de Hawaii'de doğdu." -> [("Barack Obama", "Kişi"), ("Hawaii", "Yer")]

Duygu Analizi

- Metnin duygusal tonunu belirleme (olumlu, olumsuz, tarafsız).
- Örnek: "Bu ürün harika!" -> ["Olumlu"]
"Bu film berbattı!" -> ["Olumsuz"]

Flair Nedir?

- FLAIR, açık kaynaklı bir DDI kütüphanesidir.
- Python dilinde yazılmıştır ve kullanımı kolaydır.
- Önceden eğitilmiş modeller sunar ve kendi modellerinizi eğitmenize olanak tanır.
- Metin sınıflandırma, duygu analizi gibi görevler için kullanılabilir.

Flair ile NLP Görevleri

- 1. Metin Sınıflandırma**
- 2. Varlık İsmi Tanıma (NER)**
- 3. POS Etiketi (Part-of-Speech)**
- 4. Duygu Analizi**

Metin Sınıflandırma

- Doğal dil işleme (NLP) tekniklerini kullanarak metin verilerini belirli kategorilere veya sınıflara ayırma işlemidir.

- ```
from flair.models import TextClassifier
from flair.data import Sentence
Duygu analizi modelini yükle
classifier = TextClassifier.load('en-sentiment')
Sınıflandırmak istediğimiz metni oluştur
sentence = Sentence('Flair is amazing!')
Metni sınıflandır
classifier.predict(sentence)
Sonucu yazdır
print(sentence.labels)
Örnek çıktı: [POSITIVE (0.998)]
```

# Varlık İsmi Tanıma (NER)

- Metindeki adları (kişiler, yerler, kuruluşlar vb.) belirleme görevidir.
- ```
from flair.data import Sentence  
from flair.models import SequenceTagger
```

```
# Model yükleme
```

```
tagger = SequenceTagger.load('ner')
```

```
# Metni etiketleme
```

```
sentence = Sentence("Barack Obama was born in Hawaii.")
```

```
tagger.predict(sentence)
```

```
print(sentence.to_tagged_string())
```

Örnek çıktı: Barack <B-PER> Obama <I-PER> was born in Hawaii <B-LOC> .

POS Etiketi (Part-of-Speech)

- Kelimelerin dilbilgisel kategorilerinin belirlenmesi.
- `from flair.data import Sentence`
`from flair.models import SequenceTagger`

`# Model yükleme`

`tagger = SequenceTagger.load('pos')`

`# Metni etiketleme`

`sentence = Sentence("Flair is very useful.")`

`tagger.predict(sentence)`

`print(sentence.to_tagged_string())`

`# Örnek çıktı: Flair <NNP> is <VBZ> very <RB> useful <JJ> .`

Duygu Analizi

- Metnin duygusal tonunu belirleme. Örneğin, bir yorumun olumlu, olumsuz veya tarafsız olup olmadığını analiz etmek.
- `from flair.data import Sentence`
`from flair.models import TextClassifier`

`# Model yükleme`

`classifier = TextClassifier.load('en-sentiment')`

`# Metni sınıflandırma`

`sentence = Sentence("I love using Flair for NLP tasks!")`

`classifier.predict(sentence)`

`print(sentence.labels)`

`# Örnek çıktı: [POSITIVE (0.98)]`

Kaynakça

- <https://flairnlp.github.io/docs/intro>
- <https://dev.to/admantium/python-nlp-library-flair-41f4>
- <https://github.com/flairNLP/flair>

Teşekkürler!

- **Soru-Cevap**