Python NLTK Kütüphanesi

Ahmet Göktuğ Özdemir

19360859070

Sunum İçeriği

- Doğal Dil İşleme Nedir?
- Doğal Dil İşlemenin Kullanım Alanları
- Doğal Dil İşlemenin Adımları
- Veri Temizleme Şekilleri
- Türkçedeki ve İngilizcedeki Etkisiz Kelimeler
- Algoritma Tasarlanması
- Kural Bazlı Ve Makine Öğrenmesine Dayalı Sistem Farkı
- Doğal Dil İşlemedeki Zorluklar
- Python Nltk Nedir?
- Python Nltk Proje Örneği

Doğal Dil İşleme Nedir?

- Doğal Dil İşleme insanların kendi aralarında anlaşmak için kullandıkları dili insan-bilgisayar etkileşimini en üst düzeye çıkarabilmek veya farklı doğal dilleri kullanan insanlar arasında iletişimi güçlendirmek üzere çözümler üreten bilim alanıdır.
- Bilgisayarlarda TextBlob, NLTK, spaCy, Genism, Pattern gibi programlar ve kütüphaneler ile yapılabilir.

Doğal Dil İşlemenin Kullanım Alanları

- E-mail Filtireleri
- Sanal Asistanlar
- Çeviri programları
- Chat botlar
- Kişisel Reklamlar
- Ses Tanıma

Doğal Dil İşlemenin Adımları

- Doğal Dil İşlemenin iki temel adımı vardır. Bu adımlar verinin temizlenmesi ve algoritmanın tasarlanmasıdır.
- Verinin temizlenmesi, makinelerin analiz edebilmesi için metin verilerinin hazırlanmasını ve "temizlenmesini" içerir. Veri temizlemesi, verileri uygulanabilir bir forma sokar ve metinde algoritmanın çalışabileceği özellikleri vurgular.
- Algoritma tasarlanması ön işlenmiş veriden istenen bilgilerin çıkarılması için adımları planlamaktır.

Veri Temizleme Şekilleri

- Tokenizasyon

Tokenizasyon, metnin çalışmak için daha küçük birimlere ayrılmasıdır.

Ali uçağını kaçırmamak için arabaya bindi. Bu cümleyi ‘Ali’, ‘uçağını’, ‘kaçırmamak’, ‘için’, ’arabaya’, ’bindi’, ‘.’ şekline getirmek bir Tokenizasyon örneğidir.

- Etkisiz Kelime Kaldırma

Etkisiz Kelime Kaldırma, ortak kelimelerin metinden çıkarıldığı ve metin hakkında en fazla bilgiyi sunan benzersiz kelimelerin kaldığı zamandır.

Ali uçağını kaçırmamak için arabaya bindi. Bu cümledeki etkisiz kelimeleri kaldırırsak 'Ali', 'uçağını', 'kaçırmamak', 'arabaya', 'bindi', ‘.’ sonucunu alırız.

Veri Temizleme Şekilleri

- Lemmatization ve kökleme(stemming)

Bu işlemler, kelimelerin işlenmek üzere kök biçimlerine indirgendiği zamandır.

Lemmatization ve kökleme arasındaki fark Lemmatization kelimenin kullanıldığı yere bakarak köke ayırırken kökleme sadece köküne ayırır.

Örnek olarak buluşma kelimesi hem bir fiil hem bir isim olarak kullanılabilir. Lemmatization bunu anlama göre indirgemeye çalışırken stemming anlamdan bağımsız olarak indirger.

- Sözcük Türlerini Belirleme

Bu, kelimelerin isimler, fiiller ve sıfatlar gibi gruplara ayrıldığı zamandır.

Türkçedeki Etkisiz Kelimeler

- **A:** acaba, ama, ancak, artık, asla, aslında, az
- **B:** bana, bazen, bazı, bazıları, bazısı, belki, ben, beni, benim, beş, bile, bir, birçoğu, birçok, birçokları, biri, birisi, birkaç, birkaçı, birşey, birşeyi, biz, bize, bizi, bizim, böyle, böylece, bu, buna, bunda, bundan, bunu, bunun, burada, bütün
- **C:** çoğu, çoğuna, çoğunu, çok, çünkü
- **D:** da, daha, de, değil, demek, diğer, diğeri, diğerleri, diye, dolayı
- **E:** elbette, en
- **F:** fakat, falan, felan, filan
- **G:** gene, gibi
- **H:** hangi, hangisi, hani, hatta, hem, henüz, hep, hepsi, hepsine, hepsini, her, her biri, herkes, herkese, herkesi, hiç, hiç kimse, hiçbiri, hiçbirine, hiçbirini
- **İ:** için, içinde, ile, ise, işte
- **K:** kaç, kadar, kendi, kendine, kendini, ki, kim, kime, kimi, kimin, kimisi
- **M:** madem, mı, mi, mu, mü
- **N:** nasıl, ne, ne kadar, ne zaman, neden, nedir, nerde, nerede, nereden, nereye, nesi, neyse, niçin, niye
- **O:** ona, ondan, onlar, onlara, onlardan, onların, onu, onun, orada, oysa, oysaki

Türkçedeki Etkisiz Kelimeler

- Ö: öbürü, ön, önce, ötürü, öyle
- S: sana, sen, senden, seni, senin, siz, sizden, size, sizi, sizin, son, sonra, seobilog
- Ş: şayet, şey, şimdi, şöyle, şu, şuna, şunda, şundan, şunlar, şunu, şunun
- T: tabi, tamam, tüm, tümü
- Ü: üzere
- V: var, ve, veya, veyahut
- Y: ya, ya da, yani, yerine, yine, yoksa
- Z: zaten, zira

İngilizcedeki Etkisiz Kelimeler

a	able	about	above	abroad	abst
accordance	according	accordingly	across	act	actually
added	adj	adopted	affected	affecting	affects
after	afterwards	again	against	ago	ah
ahead	ain't	all	allow	allows	almost
alone	along	alongside	already	also	although
always	am	amid	amidst	among	amongst
amount	an	and	announce	another	any
anybody	anyhow	anymore	anyone	anything	anyway
anyways	anywhere	apart	apparently	appear	appreciate
appropriate	approximately	are	aren	arent	aren't

İngilizcedeki Etkisiz Kelimeler

arise	around	as	a's	aside	ask
asking	associated	at	auth	available	away
awfully	b	back	backward	backwards	be
became	because	become	becomes	becoming	been
before	beforehand	begin	beginning	beginnings	begins
behind	being	believe	below	beside	besides
best	better	between	beyond	bill	biol
both	bottom	brief	briefly	but	by
c	ca	call	came	can	cannot
cant	can't	caption	cause	causes	certain

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a g

Algoritma Tasarlanması

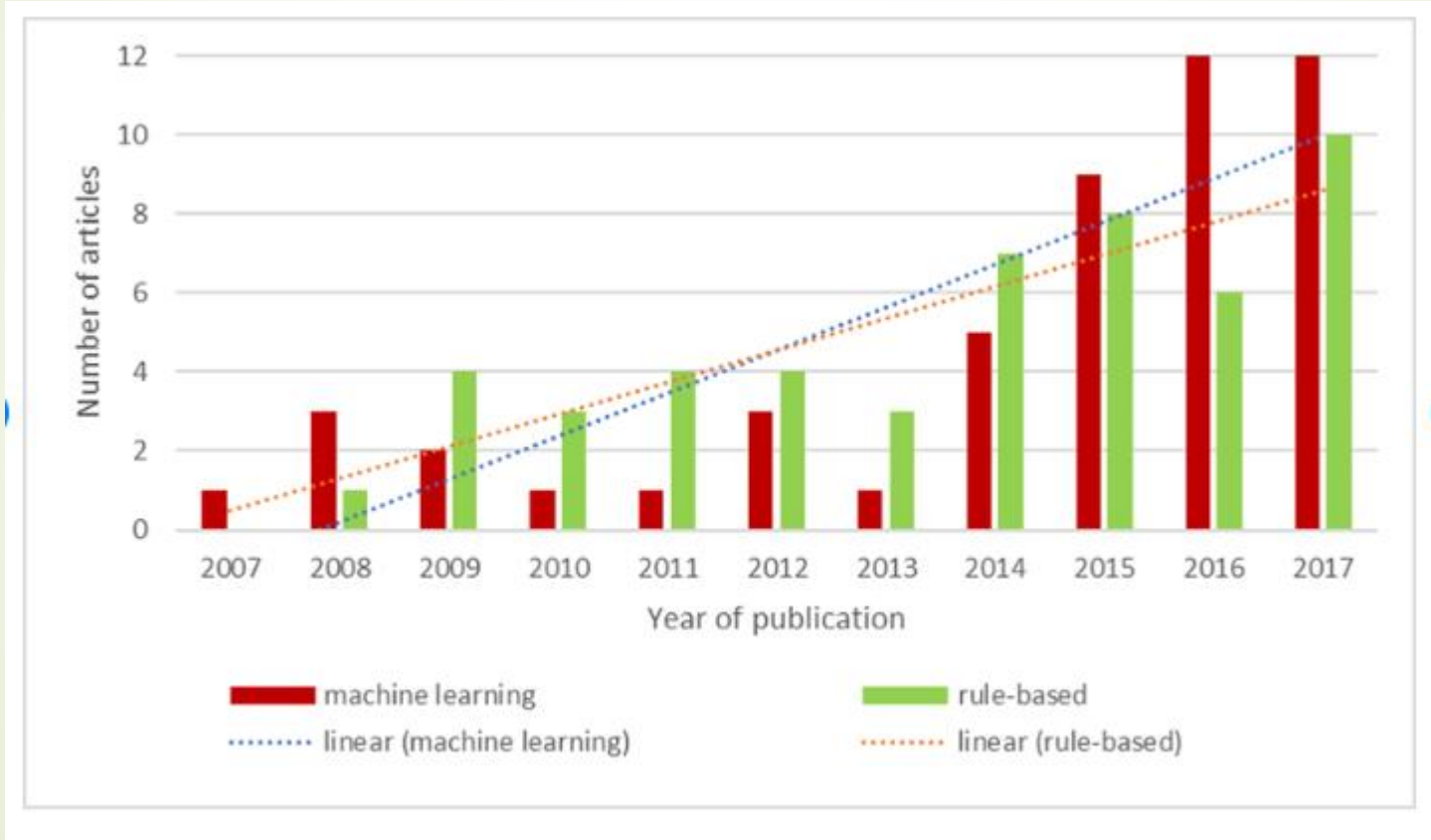
- Algoritma tasarlamak için en çok kullanılan iki tip kural tabanlı sistem ve makina öğrenmesine bağlı sistemdir.
- Kural Tabanlı Sistem

Bu sistem belirlenmiş dilsel kurallara göre bir algoritma tasarlamamızı sağlar. Grammer kurallarının belirlenerek kodlanması ile yapılır. Bu yaklaşım, daha çok doğal dil işlemenin geliştirilmesinde erken dönemde kullanılmasına rağmen halen kullanılmaktadır.

- Makina Öğrenmesine Bağlı Sistem

Makine öğrenimi algoritmaları istatistiksel yöntemler kullanır. Beslenilen eğitim verilerine dayalı olarak görevleri gerçekleştirmeyi öğrenirler ve daha fazla veri verildiği sürece yöntemlerini geliştirirler.

Kural Bazlı Ve Makine Öğrenmesine Dayalı Sistem



Doğal Dil İşlemedeki Zorluklar

- Bağlamsal kelimeler, deyimler ve eş anlamlı kelimeler
- İroni ve alay kullanımı
- Yazıdaki belirsizlik
- Metin veya konuşmadaki yazım hataları
- Konuşma dili ve argo arasındaki fark
- Çok kullanılmayan diller
- Araştırma ve geliştirme eksikliği

Python Nltk Nedir?

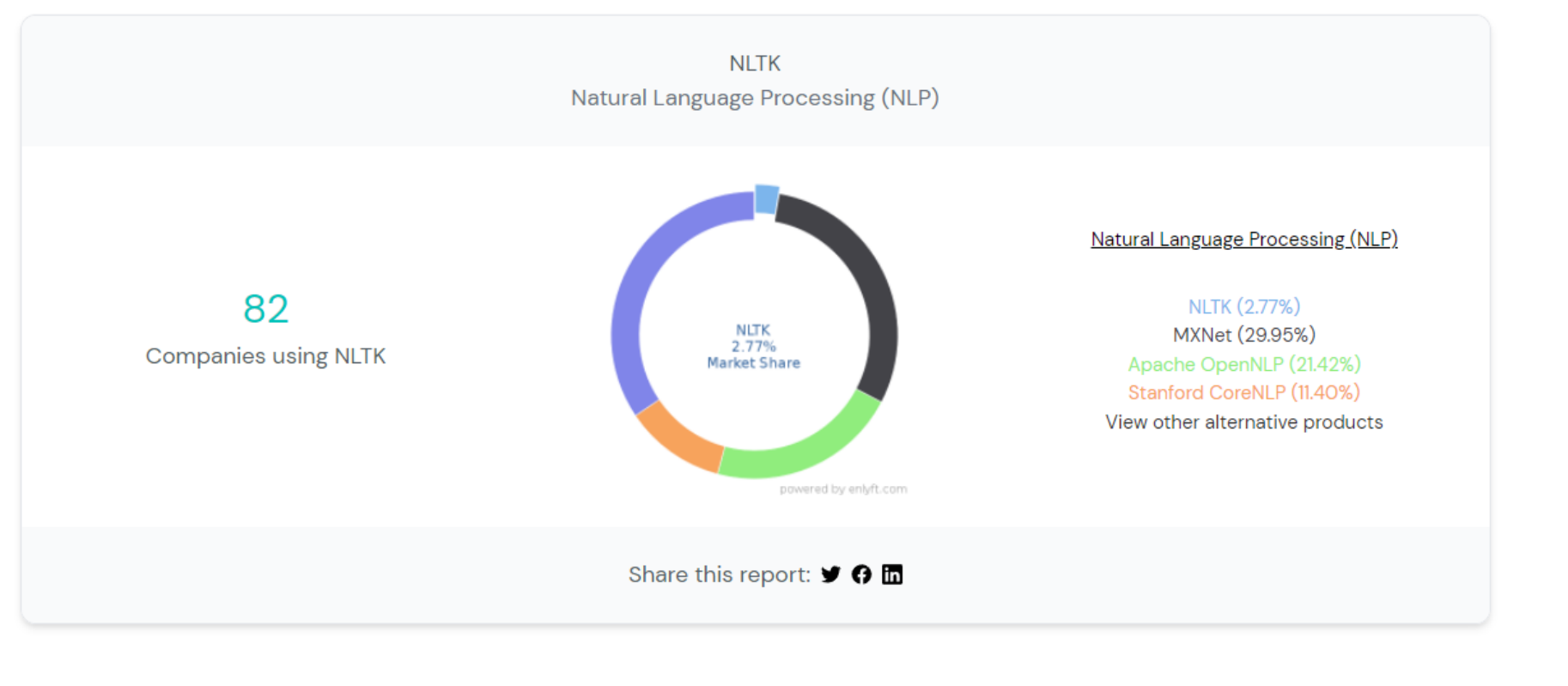
- Python Nltk (Natural Language Toolkit) açık kaynaklı bir Python kütüphanesidir. Python dilinde doğal dil işleme işlemini yapmamızı sağlar.
- Pensilvanya Üniversitesinde Steven Bird ve Edward Loper tarafından geliştirilmiştir. 2001 yılında kullanıma açılmıştır.
- Günümüzde hala desteklenmektedir.
- Doğal dil işleme konusunda en çok kullanılan

kütüphanelerden biridir.

Neden Python Nltk?

- Python Nltk doğal dil işleme konusunda kullanım oranı yüksek olduğundan bir çok hazır kütüphane bulundurur ve en çok dili desteklenmesidir. Örnek olarak stemming için RSLPStemmer (Portuguese), ISRIStemmer (Arabic), ve SnowballStemmer (Danish, Dutch, English, Finnish, French, German, Hungarian, Italian, Norwegian, Portuguese, Romanian, Russian, Spanish, Swedish) bulunur. Ancak öğrenimi ve kullanımı diğer kütüphanelere göre daha zordur, yavaştır ve sinir ağı modellini desteklemez.

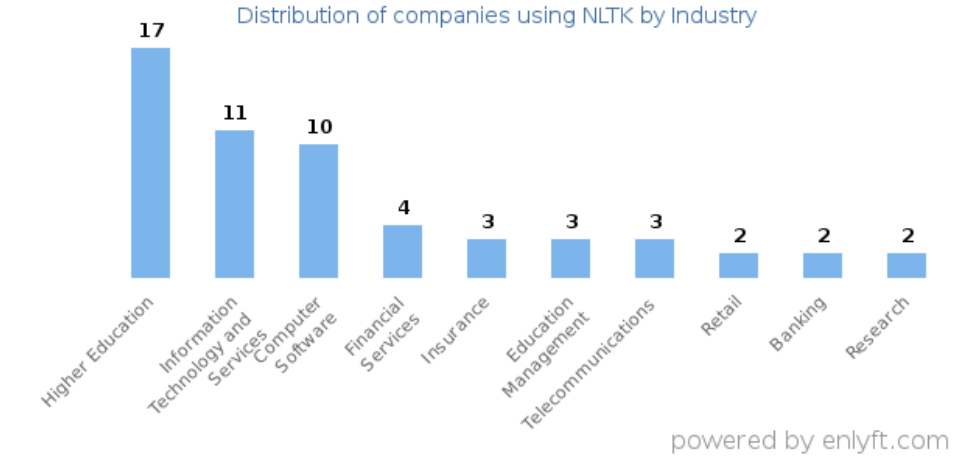
Python Nltk Nedir?



Python Nltk Nedir?

Top Industries that use NLTK

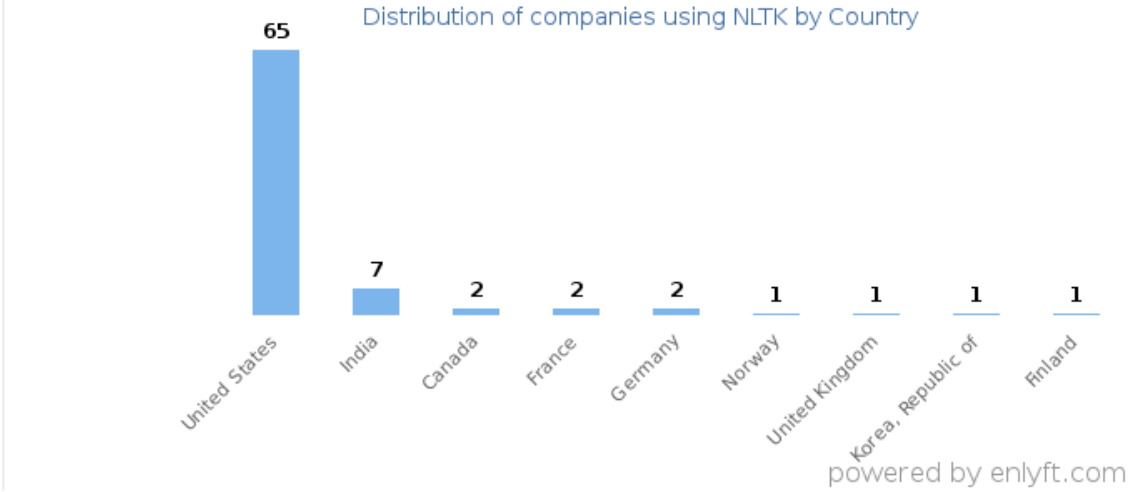
Looking at NLTK customers by industry, we find that Higher Education (20%), Information Technology and Services (12%) and Computer Software (11%) are the largest segments.



Python Nltk Nedir?

Top Countries that use NLTK

78% of NLTK customers are in United States and 7% are in India.



Python Nltk Proje Örneği

```
import nltk
from nltk.stem import PorterStemmer
nltk.download('punkt')
from nltk.tokenize import word_tokenize, sent_tokenize
```

Python Nltk Proje Örneği

```
sentence="Bu cümle seminer dersi için testir. Tokenizasyon işlemi yapılacaktır"
print(sent_tokenize(sentence))
print(word_tokenize(sentence))
```

```
['Bu cümle seminer dersi için testir.', 'Tokenizasyon işlemi yapılacaktır']
['Bu', 'cümle', 'seminer', 'dersi', 'için', 'testir', '.', 'Tokenizasyon', 'işlemi', 'yapılacaktır']
```

Python Nltk Proje Örneği

```
kokbul2 = TurkishStemmer()
print([kokbul2.stem('vardır'), kokbul2.stem('var')])
print([kokbul2.stem('bildirgeyi'), kokbul2.stem('bildirgede')])
print([kokbul2.stem('herkes'), kokbul2.stem('herkesin')])
```

```
['var', 'var']
['bildirge', 'bildirge']
['herkes', 'herk']
```

Python Nltk Proje Örneği

```
words = ["rocks", "corpora", "better"]
for w in words:
    print(w, " : ", ps.stem(w))
lemmatizer = WordNetLemmatizer()
print("rocks :", lemmatizer.lemmatize("rocks"))
print("corpora :", lemmatizer.lemmatize("corpora"))
print("better :", lemmatizer.lemmatize("better", pos="a"))
```

```
rocks  :  rock
corpora  :  corpora
better  :  better
rocks : rock
corpora : corpus
better : good
```

Python Nltk Proje Örneği

```
ps = PorterStemmer()
words = "reading,playing,watching,seeing,eating"
words = word_tokenize(words)
for word in words:
    print(word + ":" + ps.stem(word))
```

```
reading:read
,:,
playing:play
,:,
watching:watch
,:,
seeing:see
,:,
eating:eat
```

Python Nltk Proje Örneği

```
from nltk.corpus import stopwords
stopWords = set(stopwords.words('turkish'))
sentence="Size bunları anlattım çünkü bir şey yapmanızı bekledim."
words = word_tokenize(sentence)
wordsFiltered = []
for w in words:
    if w not in stopWords:
        wordsFiltered.append(w)
print(wordsFiltered)
```

```
['Size', 'bunları', 'anlattım', 'bir', 'yapmanızı', 'bekledim', '.']
```