

# Hafta 10 - Vektör Uzay Modelleri

## BGM 565 - Siber Güvenlik için Makine Öğrenme Yöntemleri

Bilgi Güvenliği Mühendisliği  
Yüksek Lisans Programı

**Dr. Ferhat Özgür Çatak**  
ozgur.catak@tubitak.gov.tr

İstanbul Şehir Üniversitesi  
2018 - Bahar

## 1 Text Mining

- Giriş

- NLTK
- Tf-Idf
- Vektör Uzay Modeli

## 1 Text Mining

- Giriş

- NLTK
- Tf-Idf
- Vektör Uzay Modeli

# Metin Madenciliği - Doğal Dil İşleme

Text Mining - Natural Language Processing

## Doğal Dil

- ▶ “Doğal dil” ile, insanlar tarafından günlük iletişim için kullanılan bir dil; Türkçe, İngilizce veya Almanca gibi diller
- ▶ Programlama dilleri gibi yapay dillerden farklı olarak evrimleşerek günümüze gelmişlerdir.
- ▶ **Bütün kurallarını oluşturabilmek neredeyse imkansızdır.**

## NLTK I

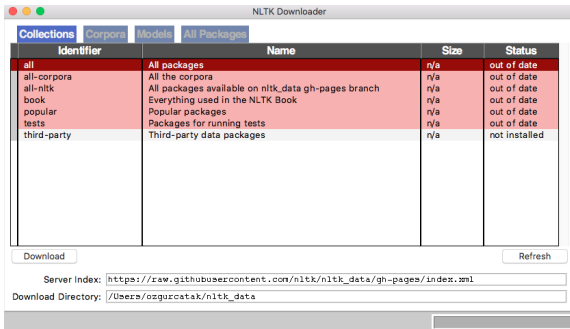
## Natural Language Toolkit

## NLTK

- ▶ Doğal dil verileriyle çalışan kütüphane

```
import nltk  
nltk.download()
```

## Natural Language Toolkit



## NLTK III

## Natural Language Toolkit

```
In [1]: from nltk.book import *
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
In [2]: text1
Out[2]: <Text: Moby Dick by Herman Melville 1851>
```

## NLTK IV

## Natural Language Toolkit

## Kelime Dağarcığı (Vocabulary)

- ▶ NLTK üzerinde `len(text7)` ifadesi ile `text7`: Wall Street Journal içinde yer alan tokenler (kelimeler, noktalama işaretleri) sayılmaktadır.
- ▶ **token**: "uzun", "onun", ":"

```
In [1]: len(text7)
```

```
Out[1]: 100676
```

```
In [2]: text7[1200:1220]
```

```
Out[2]:
```

```
['yield', '.', 'The', 'average', 'seven-day', 'simple', 'yield', 'of',  
'the', '400', 'funds', 'was', '8.12', '%', 'down', 'from', '8.14',  
'%', '.']
```



## Tokens

```
In [1]: sorted(set(text7))[0:40]
Out[1]:
['!', '#', '$', '%', '&', '"', "'", '"', '30s', '40s', '50s', '80s', '82',
 '86', 'S', 'd', 'll', 'm', 're', 's', 've', '*', '*-1', '*-10',
 '*-100', '*-101', '*-102', '*-103', '*-104', '*-105', '*-106', '*-107',
 '*-108', '*-109', '*-11', '*-110', '*-111', '*-112', '*-113', '*-114',
 '*-115']
In [2]: len(set(text7))
Out[2]: 12408
```

- Her bir token ortalama 8 kere metin içerisinde yer almaktadır.

```
In [4]: 100*text7.count("U.S.)/len(text7)
Out[4]: 0.21951607135762247
```

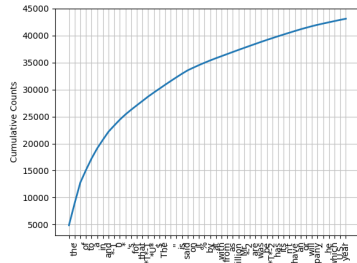
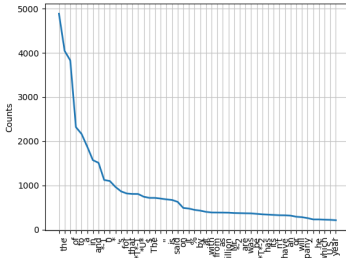
## NLTK VII

## Natural Language Toolkit

## Frekans Dağılımları

- Metnin konusu ve türü hakkında en bilgilendirici olan bir tokenlar

```
In [1]: fdist1 = FreqDist(text7)
In [2]: vocab1 = fdist1.keys()
In [3]: list(vocab1)[0:20]
Out[3]: ['Pierre', 'Vinken', ',', ', ', '61', 'years', 'old', 'will', 'join',
'the', 'board', 'as', 'a', 'nonexecutive', 'director', 'Nov.', '29', ' ',
'Mr.', 'is', 'chairman']
In [4]: fdist1.plot(50, cumulative=False)
In [5]: fdist1.plot(50, cumulative=True)
```



## Tf-Idf

- ▶ **Tf-Idf** : Term Frequency – Inverse Document Frequency
- ▶ Bir dökümanda yer alan kelimelerin (tokenların) ne kadar önemli olduğunun ölçümü
- ▶ Textual representation information  $\Rightarrow$  Vector Space Model (VSM)
- ▶ **VSM**: metinsel bilgiyi bir vektör olarak temsil eden modeldir.
  - ▶ bir terimin önemi (tf – idf)
  - ▶ Bir terimin belgede varlığı veya yokluğu (Bag of Words)

# Vektör Uzay Modeli I

## Vector Space Model

Train Document Set:

d1: The sky is blue.

d2: The sun is bright.

Test Document Set:

d3: The sun in the sky is bright.

d4: We can see the shining sun, the bright sun.

- ▶ Sözlük (index vocabulary) oluşturulması
- ▶ index vocabulary:  $E(t)$ ,  $t$ : term

$$E(t) = \begin{cases} 1, & \text{if } t \text{ is "blue"} \\ 2, & \text{if } t \text{ is "sun"} \\ 3, & \text{if } t \text{ is "bright"} \\ 4, & \text{if } t \text{ is "sky"} \end{cases}$$

- ▶ "is", "the" gibi kelimeler **stopwords** olduklarından ihmal edilirler.
- ▶ bana, bazıları, beni, böyle, bundan, bütün, edecek, ederek, hangi, kadar, o, olarak, olduklarını, onları

# Vektör Uzay Modeli II

## Vector Space Model

- ▶ Vektör uzayında her bir terimi temsil etmek için **terim frekansı (term-frequency)** kullanılır

$$tf(t, d) = \sum_{x \in d} fr(x, t)$$

$$fr(x, t) = \begin{cases} 1, & \text{if } x = t \\ 0, & \text{otherwise} \end{cases}$$

- ▶  $tf(\text{"sun"}, d_4) = 2$
- ▶ dokuman vektörü:  $\mathbf{v}_{d_n} = \{tf(t_1, d_n), tf(t_2, d_n), \dots, tf(t_m, d_n), \}$

$$\mathbf{v}_{d_3} = (0, 1, 1, 1)$$

$$\mathbf{v}_{d_4} = (0, 2, 1, 0)$$

- ▶ The matrix representation of the vectors

$$\mathcal{X} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 2 & 1 & 0 \end{bmatrix}$$

## Python

```
>>> train_set = ("The sky is blue.", "The sun is bright.")
>>> test_set = ("The sun in the sky is bright.",
                "We can see the shining sun, the bright sun.")
>>> from sklearn.feature_extraction.text import CountVectorizer
>>> vectorizer = CountVectorizer()
# Sozluk olustur
>>> vectorizer.fit_transform(train_set)
>>> print(vectorizer.vocabulary_)
{'the': 5, 'sky': 3, 'is': 2, 'blue': 0, 'sun': 4, 'bright': 1}
# sparse matrix of test set
>>> smatrix = vectorizer.transform(test_set)
>>> print(smatrix)
(0, 1)      1
(0, 2)      1
(0, 3)      1
(0, 4)      1
(0, 5)      2
(1, 1)      1
(1, 4)      2
(1, 5)      2
>>> smatrix.todense()
matrix([[0, 1, 1, 1, 1, 2],
        [0, 1, 0, 0, 2, 2]])
```

# Ters Döküman Frekansı

## Ters döküman frekansı

- ▶ Her terime eşit önem.
- ▶ *sun* kelimesi her dokümanda mevcut.
- ▶ Sözcüğün ne kadar bilgi sağladığı, yani terimin tüm belgeler arasında yaygın mı yoksa nadir mi olduğu ölçüsüdür.
- ▶ Terim  $t$ 'nin derlemde yer alan  $N$  adet mesajda görülme frekansı, doküman frekansı,  $df_t$ , olarak adlandırılır. Ters dokuman frekansı:

$$idf_t = \log \frac{N}{df_t}$$