1906003132015

# Doğal Dil İşleme

BAİBÜ Bilgisayar Müh.

Dr. Öğr. Üyesi İsmail Hakkı Parlak

ismail.parlak@ibu.edu.tr

Oda: 335

# TF-IDF

- Dolgu sözcükleri: ben, şey, ama, yani, …
- Dolgu sözcükleri üzerinde çalışılan veri setine göre değişebilir.
- Sinema: film, yönetmen; matematik: toplam, çarpım; alışveriş: ürün, satıcı; …
- Dolgu sözcüklerine nasıl karar veriyoruz?
- Önemli bir sözcüğü yanlışlıkla dolgu sözcüğü kümesine eklemiş olabilir miyiz?

# TF-IDF

- TF-IDF veri setinde sık bulunan sözcüklerin, doküman vektörlerindeki önemini azaltmak için kullanılır.
- TF-IDF = TF (Term Frequency) x IDF (Inverse Document Frequency)
- *tfidf*(t, d) = *tf*(t, d) x *idf*(t), t: term, d: document
- tf(t, d) = terim *t*'nin doküman *d*'deki frekansı.
- tf(t, d) = $\frac{count(t,d)}{len(d)}$ idf(t) = log($\frac{D}{D(t)}$)

# TF-IDF

- tf(t, d) = $\frac{count(t,d)}{len(d)}$
  - count(t, d): t teriminin d dokümanında bulunma sayısı
  - len(d): d dokümanındaki toplam terim sayısı

- idf(t) = log($\frac{D}{D(t)}$)
  - D: veri setindeki toplam doküman sayısı
  - D(t): veri setindeki, içinde t terimi geçen toplam doküman sayısı

# TF-IDF

- Örn: d0, d1, d2'yi TFIDF kullanarak vektörleştirelim.
  - d0: [a, b, a, a, b, a]
  - d1: [b, b, a, c, c]
  - d2: [a, c, d, d, d]

TF=

| | **a** | **b** | **c** | **d** |
|---|---|---|---|---|
| **d0** | 4/6 | 2/6 | 0/6 | 0/6 |
| **d1** | 1/5 | 2/5 | 2/5 | 0/5 |
| **d2** | 1/5 | 0/5 | 1/5 | 3/5 |

IDF=

| | |
|---|---|
| **a** | log(3/3) = **0** |
| **b** | log(3/2) ≈ **0.41** |
| **c** | log(3/2) ≈ **0.41** |
| **d** | log(3/1) ≈ **1.1** |

# TF-IDF

**d0**: [a, b, a, a, b, a], **d1**: [b, b, a, c, c], **d2**: [a, c, d, d, d]

TF=

| | **a** | **b** | **c** | **d** |
|---|---|---|---|---|
| **d0** | 4/6 | 2/6 | 0/6 | 0/6 |
| **d1** | 1/5 | 2/5 | 2/5 | 0/5 |
| **d2** | 1/5 | 0/5 | 1/5 | 3/5 |

IDF=

| | |
|---|---|
| **a** | log(3/3) = **0** |
| **b** | log(3/2) ≈ **0.41** |
| **c** | log(3/2) ≈ **0.41** |
| **d** | log(3/1) ≈ **1.1** |

TF-IDF=

| | **a** | **b** | **c** | **d** |
|---|---|---|---|---|
| **d0** | 4/6 x 0 = 0 | 2/6 x 0.41 ≈ 0.13 | 0/6 x 0.41 = 0 | 0/6 x 1.1 = 0 |
| **d1** | 1/5 x 0 = 0 | 2/5 x 0.41 ≈ 0.16 | 2/5 x 0.41 ≈ 0.16 | 0/5 x 1.1 = 0 |
| **d2** | 1/5 x 0 = 0 | 0/5 x 0.41 = 0 | 1/5 x 0.41 ≈ 0.08 | 3/5 x 1.1 = 0.66 |

TF-IDF=

| | **a** | **b** | **c** | **d** |
|---|---|---|---|---|
| **d0** | 0 | 0.13 | 0 | 0 |
| **d1** | 0 | 0.16 | 0.16 | 0 |
| **d2** | 0 | 0 | 0.08 | 0.66 |

# Python'da hesaplama

TF=

| | a | b | c | d |
|---|---|---|---|---|
| **d0** | 0.66 | 0.33 | 0 | 0 |
| **d1** | 0.2 | 0.4 | 0.4 | 0 |
| **d2** | 0.2 | 0 | 0.2 | 0.6 |

IDF=

| | |
|---|---|
| **a** | 0 |
| **b** | 0.41 |
| **c** | 0.41 |
| **d** | 1.1 |

```
tf = np.array([ [0.66, 0.33, 0, 0],
                [0.2, 0.4, 0.4, 0],
                [0.2, 0, 0.2, 0.6]])

idf = np.array([0, 0.41, 0.41, 1.1])

tf_idf = tf * idf
```

```
array([   [0. , 0.1353, 0. , 0. ],
          [0. , 0.164 , 0.164 , 0. ],
          [0. , 0. , 0.082 , 0.66 ]])
```