FILKOM
FAKULTAS ILMU KOMPUTER

Text Mining

Pembobotan Kata (*Term Weighting*)

Team Teaching
Universitas Brawijaya

Outline

- Document indexing
  - Bag-of-words model
- Pembobotan Kata (*Term weighting*)
  - Binary model
  - Raw term-frequency model
  - Log-frequency model
  - Document frequency/Inverse document frequency
  - Tf-idf model

Document indexing

- Tahapan ***preprocessing*** menghasilkan sekumpulan ***term*** yang akan dijadikan sebagai **indeks**
- **Indeks** merupakan perwakilan dari dokumen dan merupakan **fitur** dari dokumen tersebut
- **Indeks** menjadi dasar untuk pemrosesan selanjutnya dalam *text mining* maupun *information retrieval*

Bag of words model

- Indeks dari suatu dokumen dibuat hanya berdasarkan kemunculan kata, tanpa memperhatikan urutan kata
- Sebagai contoh, terdapat dua dokumen sebagai berikut:
  - **d1** : Kucing makan ikan
  - **d2** : Ikan makan kucing
- Kedua dokumen tersebut memiliki indeks yang sama, yaitu : kucing, makan, ikan
- Metode pembuatan indeks seperti ini disebut dengan **bag of words model**

Pembobotan kata

- Dalam pembentukan indeks berdasarkan data dokumen, setiap **kata** perlu diberi **nilai/bobot**
- Terdapat berbagai macam cara pemberian bobot pada masing-masing term pada indeks
- Pemberian **nilai/bobot** pada masing-masing term pada **indeks** disebut dengan **term weighting**

Metode pembobotan kata

- Beberapa metode pembobotan kata :
  1. Binary term weighting
  2. Raw-term frequency
  3. Log-frequency weighting
  4. Term-frequency inverse document frequency

Binary term-weighting

- Pada matriks bobot, **dokumen** berada pada **kolom** dan **term** berada di **baris**
- Tiap dokumen diwakili oleh sebuah vektor biner
- Bobot suatu term pada binary term weighting adalah **1** (jika term tersebut muncul pada suatu dokumen) atau **0** (jika term tersebut tidak muncul di dokumen)
- $w_{t,d} = \begin{cases} 1, jika\ d\ mengandung\ t \\ 0, jika\ d\ tidak\ mengandung\ t \end{cases}$
- Binary term weighting tidak memperhatikan frekuensi kemunculan kata pada sebuah dokumen

Binary term-weighting

Kelebihan :

- Mudah diimplementasikan

Kekurangan :

- Tidak dapat membedakan term yang sering muncul ataupun term yang hanya sekali muncul

Contoh dokumen

d1

Sekarang saya sedang suka memasak. Masakan kesukaan saya sekarang adalah nasi goreng. Cara memasak nasi goreng adalah nasi digoreng

d2

Ukuran nasi sangatlah kecil, namun saya selalu makan nasi

d3

Nasi berasal dari beras yang ditanam di sawah. Sawah berukuran kecil hanya bisa ditanami sedikit beras

d4

Mobil dan bus dapat mengangkut banyak penumpang. Namun, bus berukuran jauh lebih besar dari mobil, apalagi mobil-mobilan

d5

Bus pada umumnya berukuran besar dan berpenumpang banyak, sehingga bus tidak bisa melewati persawahan

Contoh term dari dokumen setelah preprocessing

d1	d2	d3
suka, masak, nasi, goreng	ukur, nasi, makan	nasi, beras, tanam, sawah

d4	d5
mobil, bus, angkut, tumpang, ukur	bus, ukur, sawah, tumpang

Binary term weighting

	D1	D2	D3	D4	D5
suka	1	0	0	0	0
masak	1	0	0	0	0
nasi	1	1	1	0	0
goreng	1	0	0	0	0
ukur	0	1	0	1	1
makan	0	1	0	0	0
beras	0	0	1	0	0
tanam	0	0	1	0	0
sawah	0	0	1	0	1
mobil	0	0	0	1	0
bus	0	0	0	1	1
angkut	0	0	0	1	0
tumpang	0	0	0	1	1

Raw term frequency weighting

- Bobot suatu term pada sebuah dokumen merupakan jumlah kemunculan term tersebut pada dokumen
- $w_{t,d} = tf_{t,d}$
- $tf_{t,d}$ = jumlah kemunculan (frekuensi) term t pada dokumen d

Raw term frequency weighting

	D1	D2	D3	D4	D5
suka	2	0	0	0	0
masak	3	0	0	0	0
nasi	3	2	1	0	0
goreng	3	0	0	0	0
ukur	0	1	0	1	1
makan	0	1	0	0	0
beras	0	0	2	0	0
tanam	0	0	2	0	0
sawah	0	0	2	0	1
mobil	0	0	0	4	0
bus	0	0	0	2	2
angkut	0	0	0	1	0
tumpang	0	0	0	1	1

d1

suka, masak, nasi, goreng

d2

ukur, nasi, makan

d3

nasi, beras, tanam, sawah

d4

mobil, bus, angkut, tumpang, ukur

d5

bus, ukur, sawah, tumpang

Raw term frequency weighting

Kelebihan :

- Memperhatikan frekuensi kemunculan term. Suatu term yang muncul 10x dalam sebuah dokumen memiliki bobot yang lebih tinggi dari term yang hanya muncul 1x

Kekurangan:

- Raw TF memberikan bobot yang terlalu tinggi pada term yang terlalu sering muncul
- Tingkat kepentingan (bobot) suatu term pada dokumen tidak seharusnya linear terhadap frekuensi kemunculan term

Log frequency weighting

- Bobot term pada sebuah dokumen merupakan logaritma dari frekuensi kemunculan term pada dokumen
- $w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, jika\ tf_{t,d} > 0 \\ 0, jika\ tf_{t,d} = 0 \end{cases}$
- Logaritma (log) berfungsi mengurangi perbedaan *tf* yang terlalu besar

tf	$1+\log_{10}$tf
0	0
1	1
2	1.3
100	3

Log frequency weighting

	D1	D2	D3	D4	D5
suka	1.301	0	0	0	0
masak	1.477	0	0	0	0
nasi	1.477	1.301	1.000	0	0
goreng	1.477	0	0	0	0
ukur	0	1.000	0	1.000	1.000
makan	0	1.000	0	0	0
beras	0	0	1.301	0	0
tanam	0	0	1.301	0	0
sawah	0	0	1.301	0	1.000
mobil	0	0	0	1.602	0
bus	0	0	0	1.301	1.301
angkut	0	0	0	1.000	0
tumpang	0	0	0	1.000	1.000

Log frequency weighting

Kelebihan

- Perbedaan frekuensi term tidak berpengaruh secara signifikan terhadap bobot term

Kekurangan

- Hanya memperhatikan kemunculan term pada sebuah dokumen saja

Document frequency

- Kata-kata yang muncul di banyak dokumen adalah kata yang tidak penting, karena tidak bisa membedakan isi dokumen-dokumen tersebut
- Meskipun telah dilakukan filtering, masih terdapat kata-kata yang sering muncul
- Contoh : merupakan, tinggi, bisa, dll
- Kata-kata tersebut kurang informatif

Document frequency

- Di sisi lain, kata-kata langka (yang hanya muncul di sebagian kecil dokumen) justru lebih informatif
- Contoh : kata Meganthropus mampu membedakan dokumen sejarah dengan dokumen olahraga dan ekonomi karena kata Meganthropus hanya muncul di dokumen sejarah

Document Frequency

	D1	D2	D3	D4	D5	df
suka	1.301	0	0	0	0	1
masak	1.477	0	0	0	0	1
nasi	1.477	1.301	1.000	0	0	3
goreng	1.477	0	0	0	0	1
ukur	0	1.000	0	1.000	1.000	3
makan	0	1.000	0	0	0	1
beras	0	0	1.301	0	0	1
tanam	0	0	1.301	0	0	1
sawah	0	0	1.301	0	1.000	2
mobil	0	0	0	1.602	0	1
bus	0	0	0	1.301	1.301	2
angkut	0	0	0	1.000	0	1
tumpang	0	0	0	1.000	1.000	2

Document frequency

- *Document frequency* (df_t) merupakan jumlah dokumen yang mengandung term *t*
- *Rare terms* merupakan term yang memiliki nilai *df* yang kecil
- *Frequent terms* merupakan term yang memiliki nilai *df* besar
- *Rare* terms seharusnya memiliki bobot yang lebih besar dari *Frequent terms* karena *rare terms* lebih informatif

Inverse document frequency weight

- df_t = Document frequency of *t* (jumlah dokumen yang mengandung term *t*)
  - df_t merupakan ukuran kebalikan dari keinformatifan term *t*
  - $df_t \leq N$ (Nilai df_t lebih kecil atau sama dengan jumlah dokumen)
- idf (Inverse document frequency) dari *t* adalah :

$$idf_t = \log_{10}(\frac{N}{df_t})$$

  - Perhitungan idf_t dapat menggunakan logaritma basis berapapun

Inverse document frequency weight

	D1	D2	D3	D4	D5	idf
suka	1.301	0	0	0	0	0.699
masak	1.477	0	0	0	0	0.699
nasi	1.477	1.301	1.000	0	0	0.222
goreng	1.477	0	0	0	0	0.699
ukur	0	1.000	0	1.000	1.000	0.222
makan	0	1.000	0	0	0	0.699
beras	0	0	1.301	0	0	0.699
tanam	0	0	1.301	0	0	0.699
sawah	0	0	1.301	0	1.000	0.398
mobil	0	0	0	1.602	0	0.699
bus	0	0	0	1.301	1.301	0.398
angkut	0	0	0	1.000	0	0.699
tumpang	0	0	0	1.000	1.000	0.398

tf-idf weighting

- Nilai *tf-idf* dari sebuah *term t* merupakan perkalian antara nilai *tf* dan nilai *idf* nya.

$$w_{t,d} = \left(1 + \log_{10}\left(tf_{t,d}\right)\right).\log_{10}\left(\frac{N}{df_t}\right)$$

- tf-idf merupakan *term weighting* yang paling populer
  - Catatan : tanda "-" pada notasi tf-idf adalah tanda hubung, bukan pengurangan!
- Term yang sering muncul di satu dokumen dan jarang muncul pada dokumen lain akan mendapatkan nilai tinggi

tf-idf weighting

	D1	D2	D3	D4	D5
suka	0.909	0	0	0	0
masak	1.032	0	0	0	0
nasi	0.328	0.289	0.222	0	0
goreng	1.032	0	0	0	0
ukur	0	0.222	0	0.222	0.222
makan	0	0.699	0	0	0
beras	0	0	0.909	0	0
tanam	0	0	0.909	0	0
sawah	0	0	0.518	0	0.398
mobil	0	0	0	1.120	0
bus	0	0	0	0.518	0.518
angkut	0	0	0	0.699	0
tumpang	0	0	0	0.398	0.398

tf-idf weighting

- Variasi bobot tf-idf

Term frequency		Document frequency		Normalization	
n (natural)	$\mathrm{tf}_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(\mathrm{tf}_{t,d})$	t (idf)	$\log \frac{N}{\mathrm{df}_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2+w_2^2+\ldots+w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times \mathrm{tf}_{t,d}}{\max_t(\mathrm{tf}_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-\mathrm{df}_t}{\mathrm{df}_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^{\alpha}$, $\alpha < 1$
L (log ave)	$\frac{1+\log(\mathrm{tf}_{t,d})}{1+\log(\mathrm{ave}_{t \in d}(\mathrm{tf}_{t,d}))}$				

FILKOM UB

Tugas

- Carilah paper/jurnal tentang *text* mining
  - Jelaskan proses *preprocessing* yang digunakan dalam *paper*/jurnal tersebut!
  - Pembobotan apa yang digunakan dalam *paper*/jurnal tersebut! Sebutkan kekurangan dan kelebihannya.
- Sumber *paper* **HANYA DIPERBOLEHKAN** dari:
  - https://ieeexplore.ieee.org/
  - https://www.sciencedirect.com/
- Pengerjaan kelompok, minimal 2 anggota dan maksimal 3 anggota
- **Paper yang digunakan dalam 1 kelas harus berbeda. Jika ada 2 atau lebih kelompok menggunakan *paper* yang sama, maka tidak akan dinilai!**
- Tugas dikumpulkan dalam bentuk *hard copy.* Maksimal dikumpulkan **9 September 2019 pukul 09.45 WIB di kelas** (sebelum perkuliahan dimulai). Keterlambatan pengumpulan tugas tidak dinilai!