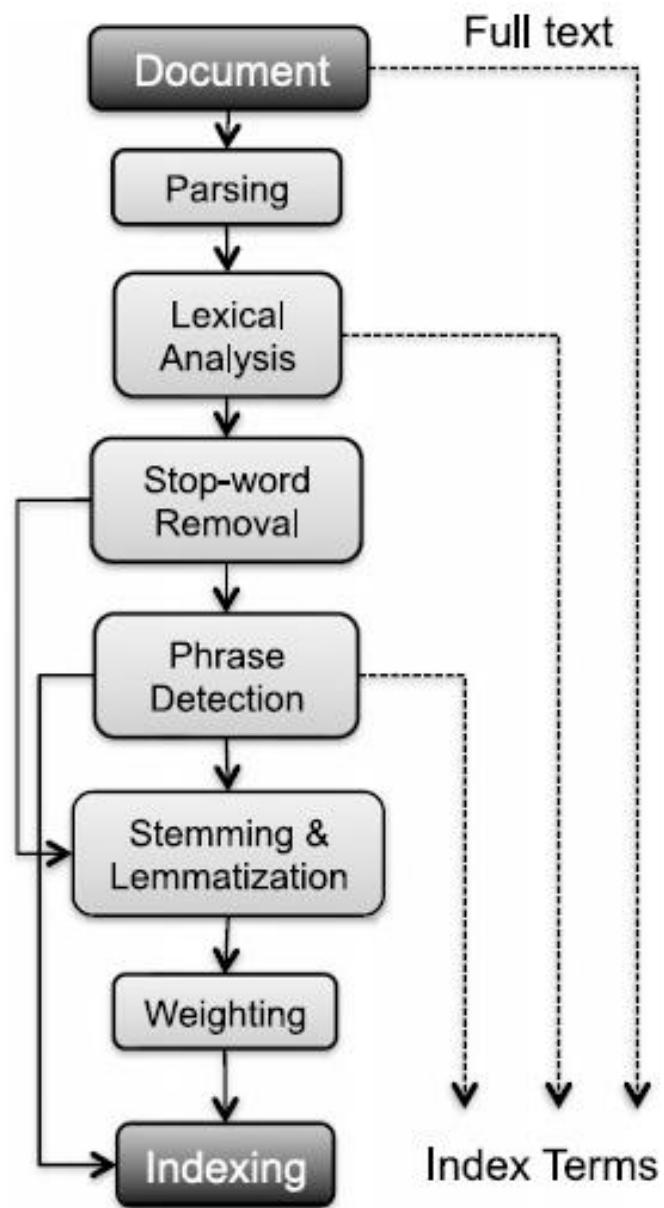


# *Document Indexing dan Term Weighting*

M. Ali Fauzi



# *Document Indexing*

- Setelah melakukan preprocessing, kita akan mendapatkan sebuah **set term** yang bisa kita jadikan sebagai **indeks**.

Indeks adalah perwakilan dari dokumen.

Indeks memudahkan proses selanjutnya dalam teks mining ataupun IR.

# *Document Indexing*

- Setelah melakukan preprocessing, kita akan mendapatkan sebuah **set term** yang bisa kita jadikan sebagai **indeks**.
- **Indeks** adalah **perwakilan** dari dokumen.
- **Indeks memudahkan** proses selanjutnya dalam teks mining ataupun IR.

# Document Indexing

Hasil Token	Hasil Filtering	Hasil Stemming	Type	Term
they	-	-	-	-
are	-	-	-	-
applied	applied	apply	apply	apply
to	-	-	-	-
the	-	-	-	-
words	words	word	word	word
in	-	-	-	-
the	-	-	-	-
texts	texts	text	text	text

# Document Indexing

Hasil Token	Hasil Filtering	Hasil Stemming	Type	Term
namanya	namanya	nama	nama	nama
adalah	-	-	-	-
santiago	santiago	santiago	santiago	santiago
santiago	santiago	santiago	-	-
sudah	-	-	-	-
memutuskan	memutuskan	putus	putus	putus
untuk	-	-	-	-
mencari	mencari	cari	cari	cari
sang	-	-	-	-
alkemis	alkemis	alkemis	alkemis	alkemis

# *Document Indexing*

- Dalam membuat sebuah indeks, secara umum kita **tidak memperhatikan urutan kata**

“John is quicker than Mary” dan “Mary is quicker than John” memiliki representasi yang sama

Ini disebut bag of words model.

# *Document Indexing*

- Dalam membuat sebuah indeks, secara umum kita **tidak memperhatikan urutan kata**
- “*John is quicker than Mary*” dan “*Mary is quicker than John*” **memiliki representasi yang sama**

Ini disebut bag of words model.

# *Document Indexing*

- Dalam membuat sebuah indeks, secara umum kita **tidak memperhatikan urutan kata**
- “*John is quicker than Mary*” dan “*Mary is quicker than John*” **memiliki representasi yang sama**
- Ini disebut **bag of words** model.

# Document Indexing

Hasil Token	Hasil Filtering	Hasil Stemming	Type	Term
they	-	-	-	-
are	-	-	-	-
applied	applied	apply	apply	apply
to	-	-	-	-
the	-	-	-	-
words	words	word	word	word
in	-	-	-	-
the	-	-	-	-
texts	texts	text	text	text

# Document Indexing

Hasil Token	Hasil Filtering	Hasil Stemming	Type	Term
namanya	namanya	nama	nama	nama
adalah	-	-	-	-
santiago	santiago	santiago	santiago	santiago
santiago	santiago	santiago	-	-
sudah	-	-	-	-
memutuskan	memutuskan	putus	putus	putus
untuk	-	-	-	-
mencari	mencari	cari	cari	cari
sang	-	-	-	-
alkemis	alkemis	alkemis	alkemis	alkemis

# Term Weighting

Teks Mining

# *Term Weighting*

- Dalam membuat sebuah indeks, setiap **kata/term** memiliki **bobot/nilai** masing-masing

Ada banyak metode untuk memberikan bobot pada masing-masing term pada indeks

# *Term Weighting*

- Dalam membuat sebuah indeks, setiap **kata/term** memiliki **bobot/nilai** masing-masing
- Ada banyak metode untuk **memberikan bobot** pada masing-masing term pada indeks

# *Term Weighting*

**Term Weighting** : Metode untuk **memberikan nilai/bobot** pada masing-masing **term indeks**.

# *Term Weighting*

- Beberapa **metode** Term Weighting yang popular :
  - Binary Term Weighting
  - (Raw) Term-frequency
  - Logarithmic Term-frequency
  - TF-IDF

# *Binary Term Weighting*

Metode Term Weighting

# *Binary Term Weighting*

- Masing-masing dokumen direpresentasikan oleh sebuah **binary vector**

Dokumen diwakili oleh **kolom**, dan term diwakili oleh **baris**

Jika kata/term berada pada dokumen tertentu, maka nilainya 1, jika tidak, maka nilainya 0

# *Binary Term Weighting*

- Masing-masing dokumen direpresentasikan oleh sebuah **binary vector**
- **Dokumen** diwakili oleh **kolom**, dan **term** diwakili oleh **baris**

Jika kata/term berada pada dokumen tertentu, maka nilainya 1, jika tidak, maka nilainya 0

# *Binary Term Weighting*

- Masing-masing dokumen direpresentasikan oleh sebuah **binary vector**
- **Dokumen** diwakili oleh **kolom**, dan **term** diwakili oleh **baris**
- Jika kata/term **berada** pada dokumen tertentu, maka nilainya 1, jika tidak, maka nilainya 0

# *Binary Term Weighting*

Metode term weighting ini **tidak memperhatikan jumlah kemunculan** kata pada 1 dokumen.

# *Binary Term Weighting*

Misal : terdapat **6 dokumen** : **Antony and Cleopatra, Julius Caesar, The Tempest, Hamlet, Othello, dan Macbeth**

dan **7 kata/term** : **Antony, Brutus, Caesar, Calpurnia, Cleopatra, mercy, dan worser**

# *Binary Term Weighting*

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

# (Raw) Term-frequency

Metode Term Weighting

# (Raw) Term-frequency

- Seperti halnya binary, hanya saja mempertimbangkan **jumlah kemunculan kata** pada dokumen: **count vector**

Term frequency  $tf_{t,d}$  dari term  $t$  dalam dokumen  $d$  didefiniskan sebagai jumlah kemunculan term  $t$  pada dokumen  $d$ .

# (Raw) Term-frequency

- Seperti halnya binary, hanya saja mempertimbangkan **jumlah kemunculan kata** pada dokumen: **count vector**
- Term frequency  $\text{TF}_{t,d}$  dari term  $t$  dalam dokumen  $d$  didefiniskan sebagai jumlah kemunculan term  $t$  pada dokumen  $d$ .

# (Raw) Term-frequency

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	157	73	0	0	0	0
Brutus	4	157	0	1	0	0
Caesar	232	227	0	2	1	1
Calpurnia	0	10	0	0	0	0
Cleopatra	57	0	0	0	0	0
mercy	2	0	3	5	5	1
worser	2	0	1	1	1	0

# (Raw) Term-frequency

- Term frequency  $\text{TF}_{t,d}$  dari term  $t$  dalam dokumen  $d$  didefiniskan sebagai jumlah kemunculan term  $t$  pada dokumen  $d$ .

$$\text{TF}_{\textit{Anthony, Antony and Cleopatra}} = 157$$

$$\text{TF}_{\textit{Anthony, Julius Caesar}} = 73$$

$$\text{TF}_{\textit{Mercy, Macbeth}} = 1$$

# (Raw) Term-frequency

- Raw term frequency **kurang relevan**:
  - Sebuah term yang muncul 10 kali pada sebuah dokumen memang lebih penting dalam mewakili dokumen dibandingkan dengan term yang muncul cuma 1 kali.
  - Tapi tidak berarti 10 kali lebih penting.

# **(Raw) Term-frequency**

**Relevance does not increase proportionally with term frequency.**

# *Log Term-frequency*

Metode Term Weighting

# *Log Term-frequency*

- Log Term-frequency dari term  $t$  dalam  $d$  adalah

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1.3, 10 \rightarrow 2, 1000 \rightarrow 4,$   
etc.

# *Log Term-frequency*

- Log Term-frequency dari term  $t$  dalam  $d$  adalah

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

- $0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1.3, 10 \rightarrow 2, 1000 \rightarrow 4,$   
etc.

# Log Term-frequency

W <sub>tf</sub>	Antony & Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	$1 + 10^{\log(157)}$	$1 + 10^{\log(73)}$	0	0	0	0
Brutus	$1 + 10^{\log(4)}$	$1 + 10^{\log(157)}$	0	$1 + 10^{\log(1)}$	0	0
Caesar	$1 + 10^{\log(232)}$	$1 + 10^{\log(227)}$	0	$1 + 10^{\log(2)}$	$1 + 10^{\log(1)}$	$1 + 10^{\log(1)}$
Calpurnia	0	$1 + 10^{\log(10)}$	0	0	0	0
Cleopatra	$1 + 10^{\log(57)}$	0	0	0	0	0
Mercy	$1 + 10^{\log(2)}$	0	$1 + 10^{\log(3)}$	$1 + 10^{\log(5)}$	$1 + 10^{\log(5)}$	$1 + 10^{\log(1)}$
Worser	$1 + 10^{\log(2)}$	0	$1 + 10^{\log(1)}$	$1 + 10^{\log(1)}$	$1 + 10^{\log(1)}$	0

# Log Term-frequency

W <sub>ff</sub>	Antony & Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	3.195899652	2.86332286	0	0	0	0
Brutus	1.602059991	3.195899652	0	1	0	0
Caesar	3.365487985	3.356025857	0	1.301029996	1	1
Calpurnia	0	2	0	0	0	0
Cleopatra	2.755874856	0	0	0	0	0
Mercy	1.301029996	0	1.477121255	1.698970004	1.698970004	1
Worser	1.301029996	0	1	1	1	0

# **TF-IDF**

Metode Term Weighting

# **TF-IDF**

- Nilai *TF-IDF* dari sebuah term adalah **perkalian** antara nilai **(Log)TF** and nilai **IDF**-nya.

$$\text{TF-IDF} = \text{TF} \times \text{IDF}$$

Catatan: tanda “-” dalam tf-idf adalah tanda hubung, bukan minus!. Alternative : TF.IDF, TF x IDF

# **TF-IDF**

- Nilai *TF-IDF* dari sebuah term adalah **perkalian** antara nilai **(Log)TF** and nilai **IDF**-nya.
- $\text{TF-IDF} = \text{TF} \times \text{IDF}$
- Catatan: tanda “-” dalam tf-idf adalah tanda hubung, bukan minus!. **Alternative : TFf.IDF, TF x IDF**

# **TF-IDF**

- Apa itu **IDF**?
- IDF : **Inverse Document Frequency** atau Kebalikan dari Document Frequency

# *Document frequency*

- **Document frequency** ( $df_t$ ) adalah jumlah dokumen yang mengandung term t

$$df_t \leq N$$

$N$  = Jumlah Dokumen

# *Document frequency*

- **Document frequency** ( $df_t$ ) adalah jumlah dokumen yang mengandung term t
- $df_t \leq N$
- $N$  = Jumlah Dokumen

# Document frequency

	Antony & Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	df <sub>t</sub>
<b>Antony</b>	157	73	0	0	0	0	2
<b>Brutus</b>	4	157	0	1	0	0	3
<b>Caesar</b>	232	227	0	2	1	1	5
<b>Calpurnia</b>	0	10	0	0	0	0	1
<b>Cleopatra</b>	57	0	0	0	0	0	1
<b>Mercy</b>	2	0	3	5	5	1	5
<b>Worser</b>	2	0	1	1	1	0	4

# *Document frequency*

- **Document frequency** ( $df_t$ ) adalah jumlah dokumen yang mengandung term t
- **Rare terms** adalah term memiliki nilai **df yang kecil**

*Frequent terms* adalah term memiliki nilai **df yang besar**

# *Document frequency*

- **Document frequency** ( $df_t$ ) adalah jumlah dokumen yang mengandung term t
- **Rare terms** adalah term memiliki nilai **df yang kecil**
- **Frequent terms** adalah term memiliki **nilai df yang besar**

# *Inverse Document frequency*

- Apa itu **IDF**?
- IDF : **Inverse Document Frequency** atau Kebalikan dari Document Frequency

# *Inverse Document frequency*

- Kata-kata yang **muncul di banyak dokumen** adalah kata yang "**tidak penting**"

Misal kata : dan, di, atau, merupakan, tinggi, bisa

Sering muncul di hampir semua dokumen

Kata-kata seperti ini kurang informative

# *Inverse Document frequency*

- Kata-kata yang **muncul di banyak dokumen** adalah kata yang "**tidak penting**"
- Misal kata : dan, di, atau, merupakan, tinggi, bisa
- Sering muncul di hampir semua dokumen

Kata-kata seperti ini kurang informative

# *Inverse Document frequency*

- Kata-kata yang **muncul di banyak dokumen** adalah kata yang "**tidak penting**"
- Misal kata : dan, di, atau, merupakan, tinggi, bisa
- Sering muncul di hampir semua dokumen
- Kata-kata seperti ini **kurang informatif**

# *Inverse Document frequency*

- Di sisi lain, **kata-kata langka** yang hanya muncul di sedikit dokumen, **lebih informatif**

Misal, kata *Meganthropus* yang hanya muncul di dokumen sejarah, hampir tidak pernah muncul di dokumen-dokumen lain seperti dokumen olahraga, ekonomi, maupun politik.

# *Inverse Document frequency*

- Di sisi lain, **kata-kata langka** yang hanya muncul di sedikit dokumen, **lebih informatif**
- Misal, kata **Meganthropus** yang hanya muncul di dokumen **sejarah**, hampir tidak pernah muncul di dokumen-dokumen lain seperti dokumen olahraga, ekonomi, maupun politik.

# *Inverse Document frequency*

- **Rare terms** (Kata-kata langka yang hanya muncul di dokumen-dokumen tertentu) **lebih informatif** dibandingkan dengan **Frequent terms** (kata-kata yang muncul di banyak dokumen)

Oleh karena itu, Rare terms harus memiliki bobot/nilai yang lebih besar daripada Frequent terms

# *Inverse Document frequency*

- **Rare terms** (Kata-kata langka yang hanya muncul di dokumen-dokumen tertentu) **lebih informatif** dibandingkan dengan **Frequent terms** (kata-kata yang muncul di banyak dokumen)
- Oleh karena itu, *Rare terms* harus memiliki **bobot/nilai yang lebih besar** daripada *Frequent terms*

# *Inverse Document frequency*

- **df<sub>t</sub>**, adalah ukuran kebalikan dari keinformatifan term t
- idf (inverse document frequency) dari sebuah term t didefinisikan:

$$\text{idf}_t = \log_{10} (N/\text{df}_t)$$

# *Inverse Document frequency*

$$\text{idf}_t = \log_{10} (N/\text{df}_t)$$

- Digunakan **log (N/df<sub>t</sub>)** dibanding **N/df<sub>t</sub>** untuk “**mengecilkan**” efek dari IDF.

Menggunakan log berbasis berapapun  
tidak masalah

# *Inverse Document frequency*

$$\text{idf}_t = \log_{10} (N/\text{df}_t)$$

- Digunakan **log (N/df<sub>t</sub>)** dibanding **N/df<sub>t</sub>** untuk “**mengecilkan**” efek dari IDF.
- Menggunakan **log berbasis berapapun tidak masalah**

# Latihan

Term	$df_t$	$idf_t$
calpurnia	1	
animal	100	
sunday	1,000	
fly	10,000	
under	100,000	
the	1,000,000	

Berbeda dengan TF, sebuah term hanya memiliki satu nilai IDF.

# *Inverse Document frequency*

$$\text{idf}_t = \log_{10} (N/\text{df}_t)$$

	$\text{df}_t$	$\text{idf}_t$	$\text{idf}_t$
<b>Antony</b>	2	${}^{10}\log (6/2)$	0.47712125
<b>Brutus</b>	3	${}^{10}\log (6/3)$	0.30103
<b>Caesar</b>	5	${}^{10}\log (6/5)$	0.07918125
<b>Calpurnia</b>	1	${}^{10}\log (6/1)$	0.77815125
<b>Cleopatra</b>	1	${}^{10}\log (6/1)$	0.77815125
<b>Mercy</b>	5	${}^{10}\log (6/5)$	0.07918125
<b>Worser</b>	4	${}^{10}\log (6/4)$	0.17609126

# **TF-IDF**

- Nilai *TF-IDF* dari sebuah term adalah **perkalian** antara nilai **(Log)TF** and nilai **IDF**-nya.
- $\text{TF-IDF} = \text{TF} \times \text{IDF}$

Term weighting paling populer

# **TF-IDF**

- Nilai *TF-IDF* dari sebuah term adalah **perkalian** antara nilai **(Log)TF** and nilai **IDF**-nya.
- $\text{TF-IDF} = \text{TF} \times \text{IDF}$
- Term weighting **paling populer**

# TF-IDF

O TF-IDF = TF x IDF

$$\text{TF - IDF}_{t,d} = \log(1 + \text{tf}_{t,d}) \times \log_{10}(N / \text{df}_t)$$

Term yang sering muncul di satu dokumen dan jarang muncul pada dokumen lain akan mendapatkan nilai tinggiaaaaaaaaaa

# TF-IDF

- $\text{TF-IDF} = \text{TF} \times \text{IDF}$

$$\text{TF-IDF}_{t,d} = \log(1 + \text{tf}_{t,d}) \times \log_{10}(N / \text{df}_t)$$

- Term yang sering muncul di satu dokumen dan jarang muncul pada dokumen lain akan mendapatkan nilai tinggi

# TF-IDF

$$\text{TF-IDF}_{t,d} = \log(1 + \text{tf}_{t,d}) \times \log_{10}(N / \text{df}_t)$$

TF-IDF	Antony & Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1.524831652	1.366152196	0	0	0	0
Brutus	0.482268112	0.962061659	0	0.301029996	0	0
Caesar	0.266483532	0.265734309	0	0.103017176	0.079181246	0.07918
Calpurnia	0	1.556302501	0	0	0	0
Cleopatra	2.144487465	0	0	0	0	0
Mercy	0.103017176	0	0.116960302	0.134526562	0.134526562	0.07918
Worser	0.22910001	0	0.176091259	0.176091259	0.176091259	0

# Variasi *TF-IDF*

Metode Term Weighting

# Variasi *TF-IDF*

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

# *Term Weighting Lain*

Metode Term Weighting

# *Term Weighting Lain*

- Masih ada banyak Term Weighting lain
  - Information Gain
  - Latent semantic indexing
  - Mutual information
  - TF.IDF.ICF
  - Dsb.