

Text Mining

Budi Susanto

Materi

- Pengertian Text Mining
- Pemrosesan Text
 - Tokenisasi
 - Lemmatization
 - Vector Document

Pengertian Text Mining

- Text mining merupakan penerapan konsep dan teknik data mining untuk mencari pola dalam teks
 - proses penganalisisan teks guna menyorikan informasi yang bermanfaat untuk tujuan tertentu.
- Proses data mining untuk data dokumen atau teks memerlukan lebih banyak tahapan, mengingat data teks memiliki karakteristik yang lebih kompleks daripada data biasa.

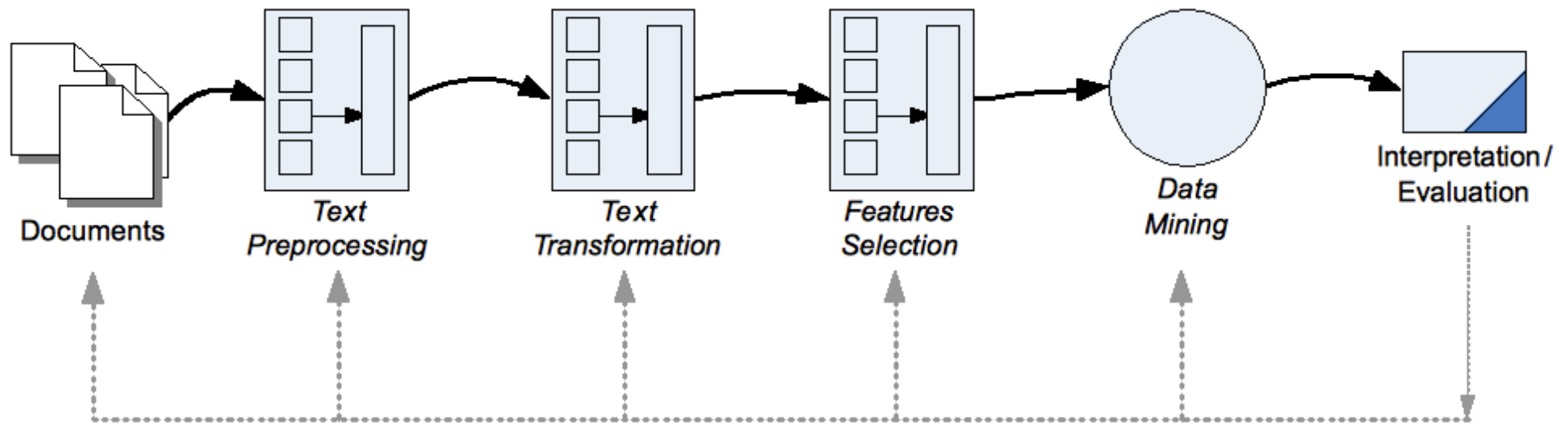
Karakteristik Dokumen Teks

- Menurut Loretta Auvil dan Duane Sears Smith dari University of Illinois, karakteristik dokumen teks:
 - database teks yang berukuran besar,
 - memiliki dimensi yang tinggi, yakni satu kata merupakan satu dimensi,
 - mengandung kumpulan kata yang saling terkait (frase) dan antara kumpulan kata satu dengan lain dapat memiliki arti yang berbeda,
 - banyak mengandung kata ataupun arti yang bias (ambiguity),
 - dokumen email merupakan dokumen yang tidak memiliki struktur bahasa yang baku, karena di dalamnya terkadang muncul istilah slang seperti "r u there?", "helllooo boss, whatzzzzzz up?", dan sebagainya.

Proses Text Mining

- Berdasarkan ketidakteraturan struktur data teks, maka proses text mining memerlukan beberapa tahap awal yang pada intinya adalah mempersiapkan agar teks dapat diubah menjadi lebih terstruktur.
- Bentuk perubahan yang dilakukan adalah ke dalam spreadsheet, kolom menunjuk dokumen dan baris menunjuk kata, sedangkan selnya menunjuk frekuensi kata dalam dokumen.

Proses Text Mining



Dokumen

- Plain text
- Format Elemen
 - XML, HTML, RTF, ODT, email, dsb.
- Format Biner
 - PDF, DOC, dsb.

Tokenisasi

- Tokenisasi secara garis besar memecah sekumpulan karakter dalam suatu teks ke dalam satuan kata.
 - bagaimana membedakan karakter-karakter tertentu yang dapat diperlakukan sebagai pemisah kata atau bukan.
 - Sebagai contoh karakter whitespace, seperti enter, tabulasi, spasi dianggap sebagai pemisah kata.
- Namun untuk karakter petik tunggal ('), titik (.), semikolon (;), titik dua (:) atau lainnya, dapat memiliki peran yang cukup banyak sebagai pemisah kata.
 - Sebagai contoh antara “tahu, tempet dan sambal” dengan “100,56”.

Tokenisasi

- Dalam memperlakukan karakter-karakter dalam teks sangat tergantung sekali pada konteks aplikasi yang dikembangkan.
- Pekerjaan tokenisasi ini akan semakin sulit jika juga harus memperhatikan struktur bahasa (*grammatikal*).

Tokenisasi: Bagaimana dengan ini?

- Karakter Nonalphanumeric
 - contoh: Yahoo!, AT&T, dsb.
- Sebuah titik (.) biasanya untuk tanda akhir kalimat, tapi dapat juga muncul dalam singkatan, inisial orang, alamat internet
 - Contoh: Sdr., S.Kom., 192.168.1.1, ukdw.ac.id
- Tanda hyphen (-) biasanya muncul untuk menggabungkan dua token yang berbeda untuk membentuk token tunggal. Tapi dapat pula ditemukan untuk menyatakan rentang nilai, kata berulang, dsb.
 - Contoh: x-ray, 32-120, lari-lari.

Tokenisasi: Bagaimana dengan ini?

- Karakter slash (/) sebagai pemisah file atau direktori atau url ataupun untuk menyatakan “dan atau”
 - Contoh: /opt/rapidminer, www.google.com/search?num=100&q=text+mining, Ibu/Bapak.
- URL.
- Format nomor telepon.
- Emoticon
- Format angka
- Frase

Tokenisasi

Lemmatization

- Setelah deretan karakter telah disegmentasi ke dalam kata-kata (token), langkah berikut yang mungkin dilakukan adalah mengubah setiap token ke bentuk standard.
 - Proses ini disebut menerapkan *stemming* dan atau *lemmatization*.
 - Tujuan: untuk mendapatkan bentuk dasar umum dari suatu kata.
 - Contoh:
 - Am, are, is => be
 - Car, cars, car's, cars' => car

Lemmatization

- Stemming
 - Proses heuristic yang memotong akhir kata, dan sering juga membuang imbuhan.
- Lemmatization
 - Serupa dengan stemming, hanya lebih baik hasilnya.
 - Memperhatikan kamus dan analisis morfologi.
 - Menghasilkan kata dasar (*lemma*)

Porter Stemming

- Algoritma stemming Porter didasarkan pada ide bahwa akhiran dalam bahasa Inggris sebagian besar terbentuk dari kombinasi akhiran yang lebih kecil dan sederhana.
- Proses penanggalan dikerjakan pada serentetan langkah, yang mensimulasikan perubahan bentuk dan penurunan dari sebuah kata.
- Pada setiap langkah, sebuah akhiran tertentu dibuang berdasar aturan substitusi.
- Sebuah aturan substitusi diterapkan ketika sekumpulan kondisi/batasan untuk aturan tersebut terpenuhi.
- Salah satu contoh kondisi adalah jumlah minimal dari hasil stem (disebut juga ukuran (measure)).
- Kondisi sederhana lain dapat berupa apakah akhir dari stem konsonan atau vokal.

Porter Stemming

(F)	Rule		Example
	SSES	→ SS	caresses → caress
	IES	→ I	ponies → poni
	SS	→ SS	caress → caress
	S	→	cats → cat

Terdapat banyak aturan lain. Kunjungi:

<http://snowball.tartarus.org/algorithms/porter/stemmer.html>

Contoh Perbandingan Stemmer

Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Porter stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Paice stemmer: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

<http://www.cs.waikato.ac.nz/~eibe/stemmers/>

<http://www.comp.lancs.ac.uk/computing/research/stemming/>

ZIPF's LAW

- Kita menggunakan sedikit kata lebih sering dan jarang untuk sebagian besar kata lain.
 - Rata-rata 20% kata-kata berperan sebagai mayoritas kata dalam suatu teks.
- Kita dapat memilih kata-kat sehingga kita mengkomunikasikan pesan dengan jumlah kata yang lebih sedikit.
- *the product of the frequency of a word and its rank will be approximately the same as the product of the frequency and rank of another word.*

Membangun Vektor untuk Prediksi

- Karakteristik ciri/sifat sebuah dokumen dinyatakan oleh token atau kata-kata di dalamnya.

Input:

ts, all the tokens in the document collection
k, the number of features desired

Output:

fs, a set of k features

Initialize:

hs := empty hashtable

for each tok in ts **do**

If hs contains tok **then**

i := value of tok in hs

increment i by 1

else

i := 1

endif

store i as value of tok in hs

endfor

sk := keys in hs sorted by decreasing value

fs := top k keys in sk

output fs

Membangun Vektor untuk Prediksi

- Himpunan ciri-ciri yang terkumpul disebut sebagai kamus (dictionary).
- Token-token atau kata-kata dalam kamus membentuk dasar untuk membuat sebuah matrik angka yang sangat berkaitan dengan kumpulan dokumen yang di analisis.
- Sehingga, setiap sel berisi ukuran dari sebuah ciri/sifat (kolom) untuk sebuah dokumen (baris).

Membangun Vektor untuk Prediksi

- Dimensi kamus yang dihasilkan tentu saja akan berukuran sangat besar, sehingga perlu dilakukan proses transformasi untuk mengurangi ukuran dimensinya.
- Beberapa proses transformasi yang dapat diterapkan antara lain:
 - Stopwords,
 - Frequent Words, dan
 - pengurangan token (Stemming atau Sinonim).

Mengubah Dokumen ke sebuah matrix

Input:

fs, a set of k features

dc, a collection of n documents

Output: ss, a spreadsheet with n rows and k columns**Initialize:** $i := 1$ **for each** document d in dc, **do** $j := 1$ **for each** feature f in fs, **do** $m :=$ number of occurrences of f in d**if** ($m > 0$) **then** ss(row=i, col=j) := 1;**else** ss(row=i, col=j) := 0 ;**endif**

increment j by 1

endfor

increment i by 1

endfor

output ss

Mengubah Dokumen ke sebuah matrix

- Untuk memberikan ketepatan prediksi, perlu dilakukan transformasi tambahan.
 - Menghitung tingkat peran kata dalam corpus.
 - tf-idf (term frequency-inverse document frequency).

$$w_{ij} = TFIDF(t_i, e_j) = TF_{i,j} \log \frac{N}{DF_i}$$

$$w_{ij} = \frac{TFIDF(t_i, e_j)}{\sqrt{\sum_{s=1}^{|V|} (TFIDF(t_s, e_j))^2}}$$

Pengukuran Lain

Information Gain:

$$IG(t_i, c_j) = P(t_i, c_j) \log \frac{P(t_i, c_j)}{P(c_j)P(t_i)} + P(\bar{t}_i, c_j) \log \frac{P(\bar{t}_i, c_j)}{P(c_j)P(\bar{t}_i)}$$

Chi-Squared Measure:

$$\chi^2(t_i, c_j) = \frac{N[P(t_i, c_j)P(\bar{t}_i, \bar{c}_j) - P(t_i, \bar{c}_j)P(\bar{t}_i, c_j)]^2}{P(t_i)P(\bar{t}_i)P(c_j)P(\bar{c}_j)}$$

Contoh

1	D1	Human machine interface for computer applications
2	D2	A survey of user opinion of computer system response time
3	D3	The EPS user interface management system
4	D4	System and human system engineering testing of EPS
5	D5	The generation of random, binary and ordered trees
6	D6	The intersection graph of paths in trees
7	D7	Graph minors: A survey

Matrik Dokumen

```
1 === Raw Term Frequencies ===
2           D1      D2      D3      D4      D5      D6      D7
3     binary  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
4     computer 1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
5 computer system 0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
6     engineering 0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000
7         eps 0.0000  0.0000  1.0000  1.0000  0.0000  0.0000  0.0000
8     generation 0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
9         graph 0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  1.0000
10        human 1.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000
11    interface 1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
12 intersection 0.0000  0.0000  0.0000  0.0000  0.0000  1.0000  0.0000
13        machine 1.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
14    management 0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
15        minors 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  1.0000
16    opinion 0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
17    ordered 0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
18    random 0.0000  0.0000  0.0000  0.0000  1.0000  0.0000  0.0000
19    response 0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
20    survey 0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  1.0000
21    system 0.0000  0.0000  1.0000  2.0000  0.0000  0.0000  0.0000
22    testing 0.0000  0.0000  0.0000  1.0000  0.0000  0.0000  0.0000
23        time 0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
24        user 0.0000  1.0000  0.0000  0.0000  0.0000  0.0000  0.0000
25 user interface 0.0000  0.0000  1.0000  0.0000  0.0000  0.0000  0.0000
```

Matrik Dokumen TF-IDF

```
1 === Inverse Document Frequency ===
2           D1      D2      D3      D4      D5      D6      D7
3     binary  0.0000  0.0000  0.0000  0.0000  0.2500  0.0000  0.0000
4     computer 0.2656  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
5 computer system 0.0000  0.1735  0.0000  0.0000  0.0000  0.0000  0.0000
6     engineering 0.0000  0.0000  0.0000  0.1977  0.0000  0.0000  0.0000
7         eps 0.0000  0.0000  0.2167  0.1512  0.0000  0.0000  0.0000
8     generation 0.0000  0.0000  0.0000  0.0000  0.2500  0.0000  0.0000
9         graph 0.0000  0.0000  0.0000  0.0000  0.0000  0.4333  0.3023
10        human 0.2031  0.0000  0.0000  0.1512  0.0000  0.0000  0.0000
11    interface 0.2656  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
12 intersection 0.0000  0.0000  0.0000  0.0000  0.0000  0.5667  0.0000
13        machine 0.2656  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
14    management 0.0000  0.0000  0.2833  0.0000  0.0000  0.0000  0.0000
15        minors 0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.3953
16    opinion 0.0000  0.1735  0.0000  0.0000  0.0000  0.0000  0.0000
17    ordered 0.0000  0.0000  0.0000  0.0000  0.2500  0.0000  0.0000
18    random 0.0000  0.0000  0.0000  0.0000  0.2500  0.0000  0.0000
19    response 0.0000  0.1735  0.0000  0.0000  0.0000  0.0000  0.0000
20    survey 0.0000  0.1327  0.0000  0.0000  0.0000  0.0000  0.3023
21    system 0.0000  0.0000  0.2167  0.3023  0.0000  0.0000  0.0000
22    testing 0.0000  0.0000  0.0000  0.1977  0.0000  0.0000  0.0000
23        time 0.0000  0.1735  0.0000  0.0000  0.0000  0.0000  0.0000
24        user 0.0000  0.1735  0.0000  0.0000  0.0000  0.0000  0.0000
25 user interface 0.0000  0.0000  0.2833  0.0000  0.0000  0.0000  0.0000
```

Feature Selection

- Teknik pemilihan sebuah subset *feature* yang relevan untuk membentuk model yang baik.