

LAPORAN

PRAKTIKUM BIG DATA ANALYTIC

Pertemuan Ke - 7



Dosen :
Sri Redjeki, S.Si., M.Kom.

Disusun oleh :
RAHADIYAN BONDAN PERMADI
215411119

Universitas Teknologi Digital Indonesia
UTDI
YOGYAKARTA
2022

DATA PREPROCESSING

Dasar Teori

Tahap Text Preprocessing adalah tahapan dimana aplikasi melakukan seleksi data yang akan diproses pada setiap dokumen. Proses preprocessing ini meliputi (1) case folding, (2) tokenizing, (3) filtering, dan (4) stemming

1. Case Folding

Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu, peran Case Folding dibutuhkan dalam mengkonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar (biasanya huruf kecil atau lowercase). Sebagai contoh, user yang ingin mendapatkan informasi “KOMPUTER” dan mengetik “KOMPOTER”, “KomPUter”, atau “komputer”, tetap diberikan hasil retrieval yang sama yakni “komputer”. Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter.

2. Tokenizing

Tahap Tokenizing adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. 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. Dalam memperlakukan karakter-karakter dalam teks sangat tergantung pada konteks aplikasi yang dikembangkan. Pekerjaan tokenisasi ini akan semakin sulit jika juga harus memperhatikan struktur bahasa (grammatikal).

3. Filtering

Tahap Filtering adalah tahap mengambil kata-kata penting dari hasil token. Bisa menggunakan algoritma stoplist (membuang kata kurang penting) atau wordlist (menyimpan kata penting). Stoplist/stopword adalah kata-kata yang tidak deskriptif yang dapat dibuang dalam pendekatan bag-of-words. Contoh stopwords adalah “yang”, “dan”, “di”, “dari” dan seterusnya. Kata-kata seperti “dari”, “yang”, “di”, dan “ke” adalah beberapa contoh kata-kata yang berfrekuensi tinggi dan dapat ditemukan hampir dalam setiap dokumen (disebut sebagai stopwords). Penghilangan stopwords ini dapat mengurangi ukuran index dan waktu pemrosesan. Selain itu, juga dapat mengurangi level noise. Namun terkadang stopping tidak selalu meningkatkan nilai retrieval. Pembangunan daftar stopwords (disebut stoplist) yang kurang hati-hati dapat memperburuk kinerja sistem Information Retrieval (IR). Belum ada suatu kesimpulan pasti bahwa penggunaan stopping akan selalu meningkatkan nilai retrieval, karena pada beberapa penelitian, hasil yang didapatkan cenderung bervariasi.

4. Stemming

Pembuatan indeks dilakukan karena suatu dokumen tidak dapat dikenali langsung oleh suatu Sistem Temu Kembali Informasi atau Information Retrieval System (IRS). Oleh karena itu, dokumen tersebut terlebih dahulu perlu dipetakan ke dalam suatu representasi dengan menggunakan teks yang berada di dalamnya. Teknik Stemming diperlukan selain untuk memperkecil jumlah indeks yang berbeda dari suatu dokumen, juga untuk melakukan pengelompokan kata-kata lain yang memiliki kata dasar dan arti yang serupa namun memiliki bentuk atau form yang berbeda karena mendapatkan imbuhan yang berbeda. Sebagai contoh kata bersama, kebersamaan, menyamai, akan distem ke root word-nya yaitu

“sama”. Namun, seperti halnya stopping, kinerja stemming juga bervariasi dan sering tergantung pada domain bahasa yang digunakan. Proses stemming pada teks berbahasa Indonesia berbeda dengan stemming pada teks berbahasa Inggris. Pada teks berbahasa Inggris, proses yang diperlukan hanya proses menghilangkan sufiks. Sedangkan pada teks berbahasa Indonesia semua kata imbuhan baik

itu sufiks dan prefiks juga dihilangkan. 3 Yang perlu diingat bahwa serangkaian proses diatas tidak selalu harus seperti itu. Proses yang dilakukan dalam preprocessing sangat bervariasi dan tergantung pada kebutuhan data anda.

Kebutuhan Alat

1. Python
2. Jupyter Notebook
3. Module Sastrawi
4. Module nltk
5. Module stopwords
6. Module string

Langkah – Langkah dalam praktikum install modul pandas dan xlrd pada python

1. Install modul Sastrawi, nltk, stopwords dan string (ERROR INSTALL STRING).

```
File Edit View Bookmarks Settings Help
~: bash -- Konsole

rhp@localhost:~$ zypper pip in Sastrawi
zypper: command 'pip' is not available.
Type 'zypper help' to get a list of global options and commands.

In case 'pip' is not a typo it's probably not a built-in command, but provided as a subcommand or plug-in (see 'zypper help subcommand').
In this case a specific package providing the subcommand needs to be installed first. Those packages are often named 'zypper-pip' or 'zypper-pip-plugin'.
rhp@localhost:~$ pip install Sastrawi
Defaulting to user installation because normal site-packages is not writeable
Collecting Sastrawi
  Downloading Sastrawi-1.8.1-py2.py3-none-any.whl (209 KB)
    209 KB 287 KB/s
Installing collected packages: Sastrawi
Successfully installed Sastrawi-1.8.1
rhp@localhost:~$ pip install nltk
Defaulting to user installation because normal site-packages is not writeable
Collecting nltk
  Downloading nltk-3.6.7-py3-none-any.whl (1.5 MB)
    1.5 MB 410 KB/s
Collecting regex==2021.8.3
  Downloading regex-2022.3.15-cp36-cp36m-manylinux_2_17_x86_64_manylinux2014_x86_64.whl (749 KB)
    749 KB 289 KB/s
Collecting tqdm
  Downloading tqdm-4.64.0-py2.py3-none-any.whl (78 KB)
    78 KB 540 KB/s
Requirement already satisfied: click in /usr/lib/python3.6/site-packages (from nltk) (8.0.4)
Collecting joblib
  Using cached joblib-1.1.0-py2.py3-none-any.whl (386 KB)
Requirement already satisfied: importlib-metadata in /usr/lib/python3.6/site-packages (from click->nltk) (4.8.3)
Collecting importlib-resources
  Using cached importlib_resources-5.4.0-py3-none-any.whl (28 KB)
Requirement already satisfied: typing-extensions==3.8.2 in /usr/lib/python3.6/site-packages (from importlib-metadata->click->nltk) (4.1.1)
Requirement already satisfied: zipp==3.5 in /usr/lib/python3.6/site-packages (from importlib-metadata->click->nltk) (3.7.0)
Installing collected packages: importlib-resources, tqdm, regex, joblib, nltk
Successfully installed importlib-resources-5.4.0 joblib-1.1.0 nltk-3.6.7 regex-2022.3.15 tqdm-4.64.0
rhp@localhost:~$ pip install stopwords
Defaulting to user installation because normal site-packages is not writeable
Collecting stopwords
  Downloading stopwords-1.0.0-py3-none-any.whl (37 KB)
Installing collected packages: stopwords
Successfully installed stopwords-1.0.0
```

2. Pastikan instalasi masuk kedalam REPL Py3 berhasil.

```
File Edit View Bookmarks Settings Help
~: bash -- Konsole

rhp@localhost:~$ python3
Python 3.6.15 (default, Sep 23 2021, 15:41:43) [GCC] on linux
Type "help()", "copyright()", "credits()" or "license()" for more information.
>>> import Sastrawi
>>> import nltk
>>> import stopwords
>>> import string
>>>
rhp@localhost:~$
```

3. Tahap preprocessing untuk data kosong.

- mengambil data insurance.csv
- Tampilkan 20 data insurance.csv

```

In [1]: import pandas as pd
import numpy as np

data_alah = "insurance.csv"
dataset = pd.read_csv(data_alah)
dataset.head(20)

Out[1]:
   age  sex  bmi  children  smoker  region  charges
0  18  male  21.01    0      yes  northwest  16864.60000
1  18  male  35.770    1      no  southeast  1725.55230
2  28  male  31.000    3      no  southeast  4449.46200
3  33  male  22.705    0      no  northwest  23984.47961
4  32  male  26.880    0      no  northwest  3668.86820
5  31  female  25.740    0      no  southeast  3756.82260
6  48  female  31.440    1      no  southeast  8240.58960
7  37  female  27.740    3      no  northwest  7281.50590
8  37  male  26.880    3      no  northwest  6498.40270
9  40  female  25.840    0      no  northwest  28923.13692
10 25  male  26.220    0      no  northwest  2721.32080
11 62  female  26.290    0      yes  southeast  27908.72510
12 23  male  34.400    0      no  southwest  1826.84300
13 56  female  39.820    0      no  southwest  11090.71780
14 27  male  42.130    0      yes  southeast  39611.75770
15 19  male  24.600    1      no  southwest  1837.23700
16 52  female  30.780    1      no  northwest  10787.33620
17 23  male  23.845    0      no  northwest  2395.17155
18 56  male  40.300    0      no  southwest  10602.38500
19 30  male  35.300    0      yes  southwest  36837.46700

```

4. Data pada insurance diatas tidak terdapat missing value (data kosong) yang jika kosong ditandai dengan "NaN" pada beberapa kolom/atribut/field. Data ini akan menjadi masalah besar dalam pengolahan data apabila jumlah missing data banyak sekali. Sebelum dilakukan pengolahan data sebaiknya dilakukan pengecekan data missing value pada dataset yang kita miliki dengan coding dan output sebagai berikut

```

In [2]: dataset.isnull().sum()

Out[2]:
age      0
sex      0
bmi      0
children 0
smoker   0
region   0
charges  0
dtype: int64

```

Dari output menunjukkan bahwa pada kolom masing-masing kolom data tidak terdapat data missing value. Selanjutnya panggil **data3.csv**

```

In [13]: import pandas as pd
import numpy as np

data3 = "data3.csv"
data = pd.read_csv(data3)
data

Out[13]:
   Nilai UTS  Nilai UAS
0      NaN      87.0
1      78.0      81.0
2      72.0      70.0
3      66.0      NaN
4      78.0      90.0
5      74.0      78.0
6      NaN      87.0
7      76.0      78.0
8      73.0      81.0
9      66.0      NaN
10     89.0      94.0
11     75.0      86.0
12     NaN      78.0

```

5. Untuk checking missing value :

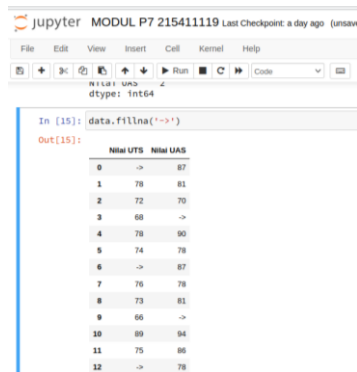
```

In [14]: data.isnull().sum()

Out[14]:
Nilai UTS    3
Nilai UAS    2
dtype: int64

```

Memberi tanda untuk datamissingvalue menggunakan -> memudahkan melihat data kosong



```

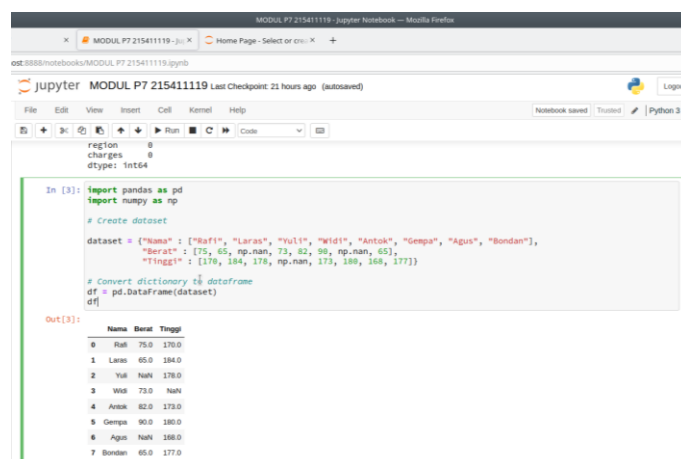
In [15]: data.fillna('->')
Out[15]:

```

	Nilai UTS	Nilai UAS
0	->	87
1	78	81
2	72	70
3	68	->
4	78	90
5	74	78
6	->	87
7	76	78
8	73	81
9	66	->
10	89	94
11	75	86
12	->	78

6. Membuat dataset baru :

Demikian laporan Pertemuan Ke-Lima yang dapat saya rangkum dan saya kerjakan, saya dapat mempraktekkan penggunaan scraping web dan penggunaan library urllib3.



```

In [3]: import pandas as pd
import numpy as np

# Create dataset
dataset = {'Nama': ["Rafi", "Laras", "Yuli", "Widi", "Antok", "Gempa", "Agus", "Bondan"],
"Berat": [75, 65, np.nan, 73, 82, 98, np.nan, 65],
"TINGGI": [170, 184, np.nan, 178, 188, 168, 177]}

# Convert dictionary to dataframe
df = pd.DataFrame(dataset)

Out[3]:

```

	Nama	Berat	Tinggi
0	Rafi	75.0	170.0
1	Laras	65.0	184.0
2	Yuli	NaN	178.0
3	Widi	73.0	NaN
4	Antok	82.0	173.0
5	Gempa	90.0	180.0
6	Agus	NaN	168.0
7	Bondan	65.0	177.0

Mengganti nilai yang awalnya “NaN” menggunakan nilai rata – rata pada variabel Berat dengan menggunakan script :



```

In [4]: df['Berat'] = df['Berat'].fillna(df['Berat'].mean())
df

Out[4]:

```

	Nama	Berat	Tinggi
0	Rafi	75.0	170.0
1	Laras	65.0	184.0
2	Yuli	75.0	178.0
3	Widi	73.0	NaN
4	Antok	82.0	173.0
5	Gempa	90.0	180.0
6	Agus	75.0	168.0
7	Bondan	65.0	177.0

Mengganti nilai yang awalnya “NaN” menggunakan nilai rata – rata pada variabel Tinggi dengan menggunakan script :



```

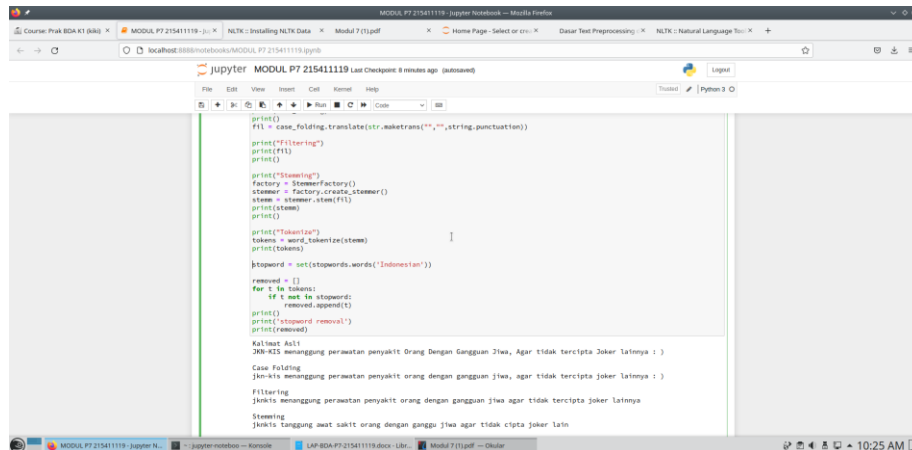
In [5]: df['TINGGI'] = df['TINGGI'].fillna(df['TINGGI'].mean())
df

Out[5]:

```

	Nama	Berat	Tinggi
0	Rafi	75.0	170.000000
1	Laras	65.0	184.000000
2	Yuli	75.0	178.000000
3	Widi	73.0	175.714286
4	Antok	82.0	173.000000
5	Gempa	90.0	180.000000
6	Agus	75.0	168.000000
7	Bondan	65.0	177.000000

7. Implementasi casefolding , filtering, tokenize dan stemming :



```
print()
f1 = case_folding.translate(str.maketrans("", "", string.punctuation))
print("Filtering")
print(f1)
print()

print("Stemming")
factory = StemmerFactory()
stemmer = factory.create_stemmer()
stem = stemmer.stem(f1)
print(stem)
print()

print("Tokenize")
tokens = word_tokenize(stem)
print(tokens)

stopword = set(stopwords.words('Indonesian'))

removed = []
for t in tokens:
    if t not in stopword:
        removed.append(t)

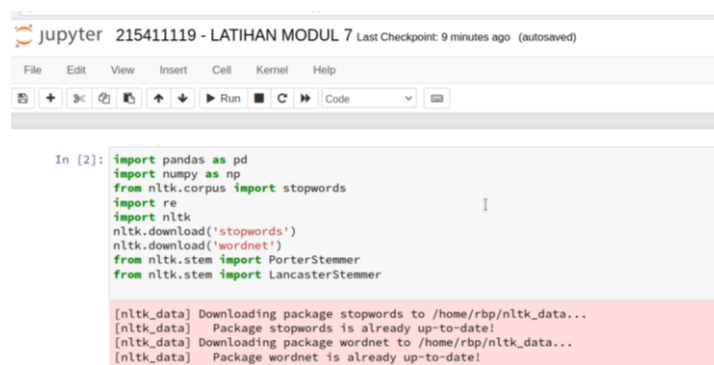
print('stopword removed')
print(removed)

Kalimat Asli
Jkn-kls menanggapi perawatan penyakit orang dengan gangguan jiwa, agar tidak tercapai joker lainnya : )
Case Folding
jkn-kls menanggapi perawatan penyakit orang dengan gangguan jiwa, agar tidak tercapai joker lainnya : )
Filtering
jknkls menanggapi perawatan penyakit orang dengan gangguan jiwa agar tidak tercapai joker lainnya
Stemming
jknkls tangng awat sakit orang dengan ganggu jiwa agar tidak cipta joker lain
```

Adalah tahapan Preprocessing dimana mesin melakukan seleksi data yang akan diproses dalam setiap dokumen yang meliputi :

1. Casefolding : berfungsi untuk memfilter atau mengkonversi keseluruhan teks dalam dokumen kedalam bentuk standar (huruf kecil).
2. Filtering : berfungsi mengambil kata – kata penting dari hasil token.
3. Stemming : untuk memperkecil jumlah indeks yang berada di suatu dokumen dan untuk melakukan pengelompokan kata.
4. Tokenize : tahap pemotongan string input berdasarkan tiap kata yang menyusun

LATIHAN 7 :



```
In [2]: import pandas as pd
import numpy as np
from nltk.corpus import stopwords
import re
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
from nltk.stem import PorterStemmer
from nltk.stem import LancasterStemmer

[nltk_data] Downloading package stopwords to /home/rbp/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /home/rbp/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
```

Import Data :

Data yang kita gunakan kali ini adalah data sepeda hasil dari scrapping web dan datanya dalam Bahasa Inggris.

```
df= pd.read_excel('datasepeda.xlsx', header=0)
```

```
df.head() #panggil data teratas
```



```
In [9]: df= pd.read_csv('datasepeda.csv', header=0)
df.head() #panggil data teratas

Out[9]:
```

	bike_name	description
0	S-Works Shiv Disc	We've never been ones for dogmatic rules. And ...
1	S-Works Aethos - Dura Ace Di2	For once, we're not in it to win. With Aethos,...
2	S-Works Stumpjumper	When we say "The Ultimate Trail Bike," we mean...
3	Shiv Expert Disc	We've never been ones for dogmatic rules. And ...
4	Aethos Pro - Ultegra Di2	The Aethos line promises three things: unprece...

Kolom data yang ingin kita pre-processing adalah hanya kolom description saja.

`df.info()`

```
In [10]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 164 entries, 0 to 163
Data columns (total 2 columns):
 #   Column      Non-Null Count  Dtype  
---  -
 0   bike_name    164 non-null   object 
 1   description  164 non-null   object 
dtypes: object(2)
memory usage: 2.7+ KB
```

`df.describe()`

```
In [11]: df.describe()

Out[11]:
```

	bike_name	description
count	164	164
unique	153	156
top	Shiv Sport	All bam-burner and no benchwarmer, the Rockho...
freq	2	3

Case Folding

jupyter 215411119 - LATIHAN MODUL 7 Last Checkpoint: 22 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help

In [12]:

```
def clean_lower(lwr):
    lwr = lwr.lower() # lowercase text
    return lwr # Buat kolom tambahan untuk data description yang telah
df['lwr'] = df['description'].apply(clean_lower)
casefolding=pd.DataFrame(df['lwr'])
casefolding
```

Out[12]:

	lwr
0	we've never been ones for dogmatic rules. and ...
1	for once, we're not in it to win. with aethos...
2	when we say "the ultimate trail bike," we mean...
3	we've never been ones for dogmatic rules. and ...
4	the aethos line promises three things: unprece...
...	...
159	you have places to go, people to see, and fitn...
160	sirrus x is your ticket to riding more, and to...
161	the rugged pitch comp 1x is built to get you w...
162	that spot where the bike path turns from concr...
163	a mountain bike is freedom: the ability to pic...

Remove Punctuation

jupyter 215411119 - LATIHAN MODUL 7 Last Checkpoint: 27 minutes ago (unsaved changes)

File Edit View Insert Cell Kernel Help

In [17]:

```
#Remove Punctuation
clean_spcl = re.compile('[/(){}\\[\]@,;]')
clean_symbol = re.compile('[^0-9a-z]')

def clean_punct(text):
    text = clean_spcl.sub('', text)
    text = clean_symbol.sub(' ', text)
    return text # Buat kolom tambahan untuk data description yang telah di removepunctuation

df['clean_punct'] = df['lwr'].apply(clean_punct)
df['clean_punct']
```

Out[17]:

0	we ve never been ones for dogmatic rules and ...
1	for once we re not in it to win with aethos w...
2	when we say the ultimate trail bike we mean ...
3	we ve never been ones for dogmatic rules and ...
4	the aethos line promises three things unprece...
...	...
159	you have places to go people to see and fitnes...
160	sirrus x is your ticket to riding more and to ...
161	the rugged pitch comp 1x is built to get you w...
162	that spot where the bike path turns from concr...
163	a mountain bike is freedom the ability to pic...

Name: clean_punct, Length: 164, dtype: object

Demikian laporan Pertemuan Ke-Tujuh yang dapat saya rangkum dan saya kerjakan, saya dapat mempraktekkan penggunaan Casefolding, Stemming, Filtering dan Tokenize dalam tahapan PreProcessing dokumen.

=====Terimakasih=====