Pre-processing data Teks menggunakan R

(Pelatihan data sains menggunakan R dan Gephi)

Ujang Fahmi

Pelajaran ke-3

Salam kenal dan selamat datang.
Semoga kita semua bisa saling berbagi pengalaman dan pengetahuan. Saya adalah Ujang Fahmi, Co-founder dan mentor Sadasa Academy.

Jika anda berada dan sedang membaca tutorial ini, maka kemungkinan anda adalah orang yang sedang ingin belajar data sains, atau mungkin ditugaskan untuk mempelajari R oleh institusi atau organisasi anda. Sama seperti saya dulu, dimana tanpa latar belakang enginering saya didiharuskan untuk belajar R, demi menyelesaikan tugas akhir dan akhirnya jadilah seperti saya sekarang ini.

Satu hal yang pasti, ini adalah langkah pertama dari banyak langkah yang harus dilalui, entah melalui lembaga resmi atau belajar secara mandiri. Jadi selamat belajar!!!

Ujang Fahmi,
Yogyakarta, 2021-09-17

Text Mining?

Text mining juga merujuk pada text data mining mirip dengan teks analitik, yaitu sebuah proses mendapatkan informasi yang berkualitas dari teks. Hal ini melibatkan penggunaan komputer untuk observasi, dan **informasi yang sebelumnya belum diketahui**.
Text mining melibatkan teknik dan metode interdisiplin seperti pemerograman komputer untuk memproses data tidak terstruktur dalam jumlah besar dan linguistik untuk memahami dan mengambil intiusi tentang penggunaan bahasa.
source: text.ai

Metode analisis

1. Term-based Method (Each term is associated with a value, known as weight)
2. Phrase-based Method (This method analyses a document based on phrases which carry more information than a single term)
3. Concept-based Method (This model tries to analyses a term on a document or sentence level by finding a significant matching term)
4. Pattern Taxonomy Method (Relation between terms to form taxonomy, which is a tree-like structure)
5. Metode hibrida pengolahan teks (e.g CADS)

Figure 1: Buzz words

Teknik analisis

1. **Extraction of Information** (The information extraction technique focuses a lot on identifying the extraction of attributes, entities, along with their relationship with unstructured or semi-structured texts)
2. **Retrieval of Information** (This technique makes use of information retrieval systems that make use of various algorithms that track and monitor user behaviour and also determine related data accordingly)
3. **Categorization** (This is a text mining technique which is a `supervised` learning form where the usual language texts are set to a pre-defined bunch of topics depending on their content)
4. **Clustering** (Clustering helps identify structures that are intrinsic in nature within text information and organize them in clusters or relevant subgroups for further analysis)
5. **Summarization** (Text summarization allows you to browse through various text sources in order to create summaries of texts which contain large amounts of information that are insightful in a concise format)

Proses Analisis

1. Mendapatkan data (umumnya tidak/belum terstruktur)
2. Melakukan pre-proccessing (melibatkan data wrangling dan cleansing)
3. Membuat data terstruktur
4. Menyimpan dan menganalisis

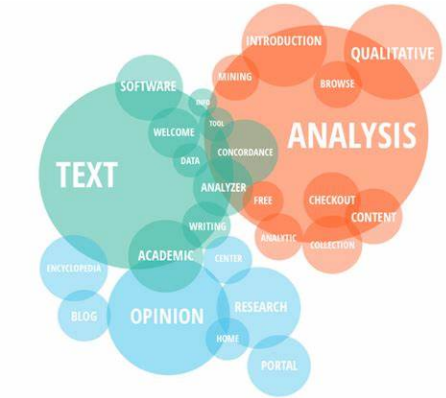

Figure 2: Buzz words

Memanfaatkan Regex (Regular Expression)

A regular expression (shortened as regex or regexp; also referred to as rational expression) is a sequence of characters that specifies a search pattern. Usually such patterns are used by string-searching algorithms for “find” or “find and replace” operations on strings, or for input validation. It is a technique developed in theoretical computer science and formal language theory. wikipedia

Penggunaan regex

Character	What does it do?	Example	Matches
^	Matches beginning of line	^abc	abc, abcdef.., abc123
$	Matches end of line	abc$	my:abc, 123abc, theabc
.	Match any character	a.c	abc, asg, a2c
|	OR operator	abc|xyz	abc or xyz
(...)	Capture anything matched	(a)b(c)	Captures 'a' and 'c'
(?:...)	Non-capturing group	(a)b(?:c)	Captures 'a' but only groups 'c'
[...]	Matches anything contained in brackets	[abc]	a,b, or c

source: computerhope

Pra-pemerosesan data teks

Data teks yang umumnya tidak terstruktur biasanya juga masih banyak elemen yang tidak diperlukan dalam analisis. Oleh karena itu biasanya kita perlu membersihkan noise nya terlebih dahulu.

Noise removal

Noise removal disini termasuk tanda html (html tags, spasi yang lebih, tanda baca dan sambung, serta menyergamkan teks)

Remove HTML tags

html tags yang umum ada dalam teks adalah url. Urls biasanya diawali dengan kelompok karakter spesifik, seperti:

- http
- https
- www
- etc

Buku untuk belajar regex di R/Handling Strings With R

```
teks = "Bpbd...
        https://instagram.com/p/CT5xmXMP_xq/?utm_medium=twitter,
        www.dodoremi.com,
        pic.twitter..."
# teks

library(stringr)
teks = gsub(" ?(f|ht)(tp)(s?)(://)(.*)[.|/](.*)", "", teks)
```

Remove extra whitespaces

```
teks = " jagalah 3m, Menjaga Jarak,
         kebersihan, dan menjaga hatii......."
teks

library(textclean)

# extra sapce
replace_white(teks)

# white space di depan/belakang
str_trim(string = teks, side = "both")
```

Remove special characters

```
teks = " jagalah 3m, Menjaga jarak,
         kebersihan, dan menjaga hatii......."

library(textclean)
teks = replace_non_ascii(teks)
teks
```

Remove Punctuations

```
teks = " jagalah 3m, menjaga jarak,
         kebersihan, dan menjaga hatii......."

teks = gsub(pattern = "[[:punct:]]", replacement = "", teks)
teks
```

Lower atau Upper Case

```
teks = "jagalah 3m, menjaga jarak"

teks = toupper(teks)
teks
teks = tolower(teks)
teks
```

Pembersihan kata sambung

Untuk membersihkan kata sambung kita membutuhkan kamus kata sambung. Kamus ini biasanya sudah dibuat terlebih dahulu. Misalnya adalah sebagai berikut:

```
stopwords_id = readr::read_csv("https://raw.githubusercontent.com/eppof
```

sumber url-stopword

kata
ada
adanya
adalah
adapun
agak

Konsep 1

```
stopwords_id$kata <- paste0("\\b", stopwords_id$kata, "\\b")
stopwords_id$to <- ""
pattern <- as.character(stopwords_id$kata)
replacement <- as.character(stopwords_id$to)

teks = "yang aku bawa adalah buku pelajaran dan makanan buat siang nant

library(textclean)
mgsub_regex(teks, pattern = pattern, replacement = replacement,
    fixed = FALSE)
```

```
## [1] "  bawa  buku pelajaran  makanan  siang "
```

Konsep 2

Pembersihan kata sambung juga bisa menggunakan filter sama seperti kita memilih baris. Tapi untuk bisa mengerjakan hal ini, kita harus telebih dahulu mentokenisasi data yang akan dibersihkan.

Tokenisasi

Apa itu tokenisasi?

> *Tokenisasi adalah proses mengubah kalimat menjadi token/term individu atau sesuai dengan aturan token yang ditentukan.*

Data yang akan digunakan bisa diambil **dari sini**

```
library(tidyverse)

data_twit = readr::read_csv("https://raw.githubusercontent.com/eppofahm

glimpse(data_twit[, 1:6])
```

```
## Rows: 10
## Columns: 6
## $ X1              <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
## $ id              <dbl> 1.364073e+18, 1.363931e+18, 1.363868e+18, 1.
## $ permalink       <chr> "https://twitter.com/CynthiaEllenna/status/1
## $ timestamp       <dbl> 1614055338, 1614021494, 1614006552, 16140033
## $ created_at      <dttm> 2021-02-23 11:42:00, 2021-02-23 02:18:00
```

One Gram

1 gram berarti 1 token. Di sini kita akan mempraktikannya pada data yang sudah diimpor sebelumnya dan disimpan dalam objek `data_twit`.

```
library(tidytext)

data_twit_token = data_twit %>%
    select(full_text_clean) %>%
    unnest_tokens(kata, input = full_text_clean, token = "words",
        to_lower = TRUE, drop = FALSE)
```

full_text_clean	kata
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	ada
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	yang
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	bisa
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	jelas
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	kronologi

Bigram

```
library(tidytext)

data_twit = readr::read_csv("https://raw.githubusercontent.com/eppofahm

data_twit$full_text_clean = textclean::replace_non_ascii(data_twit$full

data_twit_bigram = data_twit %>%
    unnest_tokens(ngram, full_text_clean, token = "ngrams", n = 2,
        drop = FALSE) %>%
    select(full_text_clean, ngram)
```

full_text_clean	ngram
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	ada yang
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	yang bisa
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	bisa jelas
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	jelas kronologi
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	kronologi kasus

Trigram

```
library(tidytext)

data_twit = readr::read_csv("https://raw.githubusercontent.com/eppofahm

data_twit$full_text_clean = textclean::replace_non_ascii(data_twit$full

data_twit_trigram = data_twit %>%
    unnest_tokens(ngram, full_text_clean, token = "ngrams", n = 3,
        drop = FALSE) %>%
    select(full_text_clean, ngram)
```

full_text_clean	ngram
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	ada yang bisa
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	yang bisa jelas
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	bisa jelas kronologi
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	jelas kronologi kasu
ada yang bisa jelas kronologi kasus fee formula e yang besar 560 milyar itu	kronologi kasus fee

Table of Contents