DATA SCIENCE CHALLENGE CLEANSING API

Ardhini Hendiani

LATAR BELAKANG

Untuk mendapatkan analisis data yang optimum, langkah penting yang harus dilakukan sebelum proses analisis adalah Pembersihan data atau Data Cleaning. Ini adalah proses memperbaiki atau menghapus data yang tidak benar, salah format, data-data duplikat, atau tidak lengkap dalam sebuah kumpulan data. Pada tahapan pengumpulan data dari berbagai macam sumber, sangat memungkinkan terjadi duplikasi data atau pemberian label yang salah. Jika data tidak benar, maka hasil analisis tidak akurat.

Salah satu analisis penting adalah analisis data teks. Data teks dapat menyimpan insight berharga dalam mempelajari analisis sentimen seperti analisis review atau cuitan sosial media. Namun, sebelum proses analisis, penting untuk memastikan bahwa data teks bersih dan siap untuk analisis. Data teks sering kali berantakan, bising, dan tidak konsisten, mengandung elemen seperti kesalahan ejaan, bahasa gaul, singkatan, emotikon, tanda baca, dan bahasa yang berbeda. Membersihkan data teks melibatkan penghapusan noise yang tidak perlu, mentransformasi teks ke dalam format yang konsisten, dan menangani segala ketidaksesuaian yang dapat menghambat analisis. Langkah-langkah ini penting untuk meningkatkan kualitas dan akurasi visualisasi data, serta meningkatkan kinerja alat analisis dan algoritma.

Source:

- https://www.tableau.com/learn/articles/what-is-data-cleaning
- https://medium.com/@pawan329/text-data-preprocessing-made-easy-steps-to-clean-text-data-using-python-81a138a0e0e3
- https://www.linkedin.com/advice/0/what-text-data-cleaning-techniques-most-effective-maxxc#:~:text=others%20are%20saying-,1%20Why%20clean%20text%20data%3F,your%20analytical%20tools%20and%20algorithms.

RUMUSAN MASALAH

01 Bagaimana pengaruh proses pembersihan data (Data Cleaning) dan transformasi data text sebelum dan sesudah dibersihkan?

02 Apa langkah-langkah yang dapat dilakukan dalam pembersihan data teks?

03 Bagaimana mengidentifikasi dan menghapus elemen-elemen tidak relevan dalam data teks?

04 Bagaimana cara mengatasi masalah seperti kesalahan ejaan, slang, dan variasi format dalam pembersihan data teks?

TUJUAN PENELITIAN

Bagaimana pengaruh proses pembersihan data (Data Cleaning) dan transformasi data text sebelum dan sesudah dibersihkan?

01

Apa langkah-langkah yang dapat dilakukan dalam pembersihan data teks?

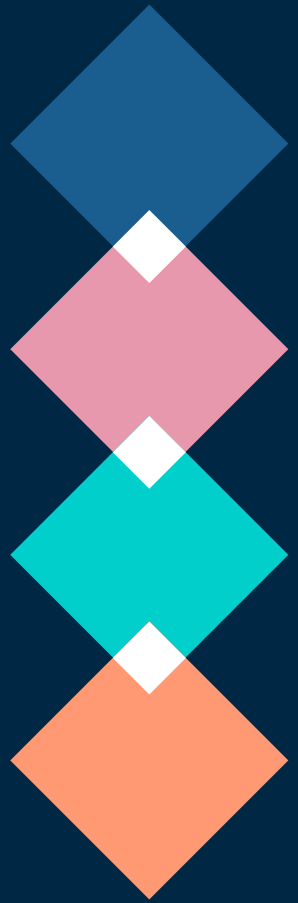
02

Bagaimana mengidentifikasi dan menghapus elemen-elemen tidak relevan dalam data teks?

03

Bagaimana cara mengatasi masalah seperti kesalahan ejaan, slang, dan variasi format dalam pembersihan data teks?

04



SUMBER DATA

Data.csv

- Dataset merupakan kumpulan tweets dengan unsur kebencian
- Dataset diberi label berdasarkan jenis tweet kebencian

new_kamusalay.csv

- Dataset kamus kata-kata typo dan slang untuk normalisasi teks
- Dataset terdiri dari dua kolom kata-kata typo dan slang, dan kolom kedua berisi kata-kata formal)

abusive.csv

- Daftar leksikon ini dapat membantu dalam mengidentifikasi konten yang mengandung bahasa kasar atau ujaran kebencian dalam data teks.

Source: https://aclanthology.org/W19-3506.pdf

PROSES DATA CLEANING

LIBRARY DAN KOMPONEN

```python
#API
from flask import Flask, jsonify
import re
import sqlite3

from flask import request
from flasgger import Swagger, LazyString, LazyJSONEncoder
from flasgger import swag_from


# Install Libraries/Packages
import re
import pandas as pd
import string

#text
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory


# Time
import time
import datetime
from datetime import datetime

# NLTK
import nltk
from nltk.tokenize import word_tokenize
nltk.download('punkt')
nltk.download('stopword')
from nltk.corpus import stopwords
```

- **Flask**: Micro web framework untuk di Python.
- **Flasgger**: Untuk membuat dokumentasi API Swagger
- **SQLite3**: Database berbasis RDMS
- **Pandas**: Alat manipulasi data
- **Sastrawi**: Perpustakaan untuk stemming kata dalam bahasa Indonesia.
- **NLTK**: The Natural Language Toolkit

Function-function

```python
# Remove special text using regex
def remove_text_special(text):
    # Remove non-ascii characters from the string
    text = re.sub(r'[^\x00-\x7f]', r'', text)
    # Replace 2+ dots with space
    text = re.sub(r'\.{2,}', ' ', text)
    # Remove newline
    text = text.replace("\\n", "")
    # Remove hashtags
    text = re.sub(r'#', '', text)
    # Remove single character
    text = re.sub(r"\b[a-zA-Z]\b", "", text)
    # Remove number
    text = re.sub('[0-9]+', '', text)
    # Remove url
    text = re.sub(r"http\S+", "", text)
    # Strip space, " and ' from tweet
    text = text.strip(' "\'')
    # Replace multiple spaces with a single space
    text = re.sub(r'\s+', ' ', text)
    # Remove url uncomplete
    text = text.replace("http://", " ").replace("https://", " ")
    # Remove punctuation
    text = text.translate(str.maketrans("","",string.punctuation))

    return text

# Remove the word 'USER'
def remove_user(df, column_name):
    df[column_name] = df[column_name].str.replace(r'USER', '', regex=True)
    return df[column_name]

# Remove the word 'RT'
def remove_RT(df, column_name):
    df[column_name] = df[column_name].str.replace(r'RT', '', regex=True)
    return df[column_name]

# Lowercase the letters
def lowercase_letters(df, column_name):
    df[column_name] = df[column_name].str.lower()
    return df[column_name]
```

```python
# Remove abusive words
def remove_abusive_words(df, column_name):
    # Load abusive words from the CSV file
    abusive_words_df = pd.read_csv('abusive.csv', encoding='latin-1')
    abusive_words = abusive_words_df['ABUSIVE'].tolist()

    # Convert specified column to string type
    df[column_name] = df[column_name].astype(str)

    # Replace or remove abusive words from the DataFrame
    for word in abusive_words:
        df[column_name] = df[column_name].str.replace(word, '')

    return df[column_name]

# Remove stopwords
def remove_stopwords(df, column_name):
    factory = StopWordRemoverFactory()
    stopword = factory.create_stop_word_remover()

    df[column_name] = df[column_name].apply(lambda x: " ".join(stopword.remove(x) for x in x.split()))
    return df[column_name]
```

- **remove_text_special**: Menghapus karakter khusus dari teks seperti karakter dan tanda baca
- **remove_user**: Menghapus kata 'USER' dari kolom teks dalam dataframe.
- **remove_RT**: Menghapus kata 'RT' (retweet) dari kolom teks dalam dataframe.
- **lowercase_letters**: Mengubah semua huruf dalam kolom teks menjadi huruf kecil.
- **remove_abusive_words**: Menghapus kata-kata kasar dari abusive.csv
- **remove_stopwords**: Menghapus kata-kata umum (stopwords) dari kolom teks dalam dataframe menggunakan modul Sastrawi.

Function

```python
# Tokenizing
def word_tokenize_wrapper(text):
    if isinstance(text, str):
        return word_tokenize(text)
    else:
        return text

# Convert slang words
    # Read slang vocabulary dictionary
convert_slang_word = pd.read_csv("new_kamusalay.csv", encoding='latin-1')

    # Create a variable in the form of a dictionary that will store the results of convert slang word function
convert_slang_word_dict = {}

for index, row in convert_slang_word.iterrows():
    if row[0] not in convert_slang_word_dict:
        convert_slang_word_dict[row[0]] = row[1]

    # Function for convert slang word
def convert_slang_word_term(document):
    if isinstance(document, float):
        return document
    else:
        return [convert_slang_word_dict[term] if term in convert_slang_word_dict else term for term in document]

# Stemming
def stemming_process(df, column_name):
    # Record the start time
    start_time = datetime.now()

    # create stemmer
    factory = StemmerFactory()
    stemmer = factory.create_stemmer()

    # stemming process
    df[column_name] = df[column_name].apply(lambda x: stemmer.stem(' '.join(x)) if isinstance(x, list) else stemmer.stem(x))
    return df[column_name]

    # Record the end time
    end_time = datetime.now()

    # Print the duration
    duration = end_time - start_time
    print("Stemming process took:", duration)
```

- **convert_slang_word_term**: Mengonversi kata-kata slang menjadi kata formal berdasarkan kamus slang dari new_kamusalay.csv
- **stemming_process**: Melakukan stemming pada teks untuk mengubah kata-kata menjadi bentuk dasarnya menggunakan Sastrawi.

Drop Null Data & Save to Database

```
df_data.dropna(subset=['Tweet_cleaned'], inplace = True)

 # Save the cleaned dataframe to SQLite3
conn = sqlite3.connect('cleansing.db')
df_data.to_sql(name='Tweets', con=conn, if_exists='replace', index=False)

# Close database
conn.close()

# Extract the 'Tweet' column from the DataFrame
texts = df_data['Tweet_cleaned'].tolist()
```

- **drop.na**: digunakan untuk menghapus baris atau kolom yang memiliki nilai yang kosong atau null data
- **df_data.to_sql**: Menyimpan hasil cleaning ke database sqlite3

EXPLORATORY DATA ANALYSIS (EDA)

Perbandingan Hasil Tweet

Out[10]:

	Tweet_cleaned	Tweet
0	di saat semua cowok usaha lacak perhati gue ka...	- disaat semua cowok berusaha melacak perhatia...
1	siapa telat beri tau kamu gue gaul cigax jifla...	RT USER: USER siapa yang telat ngasih tau elu?...
2	kadang aku pikir aku tetap percaya tuhan padah...	41. Kadang aku berfikir, kenapa aku tetap perc...
3	aku aku tau mata lihat mana aku	USER USER AKU ITU AKU\n\nKU TAU MATAMU SIPIT T...
4	kaum ce kafir sudah lihat nya awal tambah haha	USER USER Kaum cebong kapir udah keliatan dong...
5	dan kawan kawan xfxfxxxfxfxxxfxfxx	USER Ya bani taplak dkk \xf0\x9f\x98\x84\xf0\x...
6	deklarasi pilih kepala daerah aman anti hoaks ...	deklarasi pilkada 2018 aman dan anti hoax warg...
7	gue baru saja selesai rewatch aldnoah zero pal...	Gue baru aja kelar re-watch Aldnoah Zero!!! pa...
8	nah admin belanja satu port baik nak makan ais...	Nah admin belanja satu lagi port terbaik nak m...
9	enak lagi kalau sambil	USER Enak lg klo smbil ngewe'
10	gue punya jari tengah buat kamu belum gue ukur...	Setidaknya gw punya jari tengah buat lu, sebel...
11	kaleng malu tidak jawab peanyaan hari lalu nyu...	USER USER USER USER BANCI KALENG MALU GA BISA ...
12	kalau ajar ekonomi mesti jago privatisasi hati...	Kalo belajar ekonomi mestinya jago memprivatis...
13	aktor huru hara prabowo kan perintah jokowi nyata	Aktor huruhara 98 Prabowo S ingin lengserkan p...
14	bu guru enak jadi guru sekolah dasar sihkayakn...	USER Bu guru enakan jadi jablay atau guru esde...
15	lawan bicara gue tidak intelek kayak kamu yang...	USER USER USER USER USER USER Lawan bicara gw ...
16	belakang kok pikir banget	Belakangan ini kok fikiran ampas banget ya'
17	ari sama bek adalah rapi xfxfxxxfxfxx	Ari sarua beki mah repeh monyet\xf0\x9f\x98\x8...
18	jadi cowok gantle kalau tidak gantle nama	Jadi cowo itu harus Gantle kalo ga Gantle itu ...
19	alga mnr bom xfxfxx	USER Slga mnr bom \xf0\x9f\x98\x82'
20	anjing gue jarang ambek takut wkwk gue kan bud...	Asw ya tapi gua jarang ngambek, tacut wkkwkwkw...

Perbandingan Info Data Sebelum dan Sesudah drop.na

In [7]:
```
# Check dataset info
df_data.info()
```

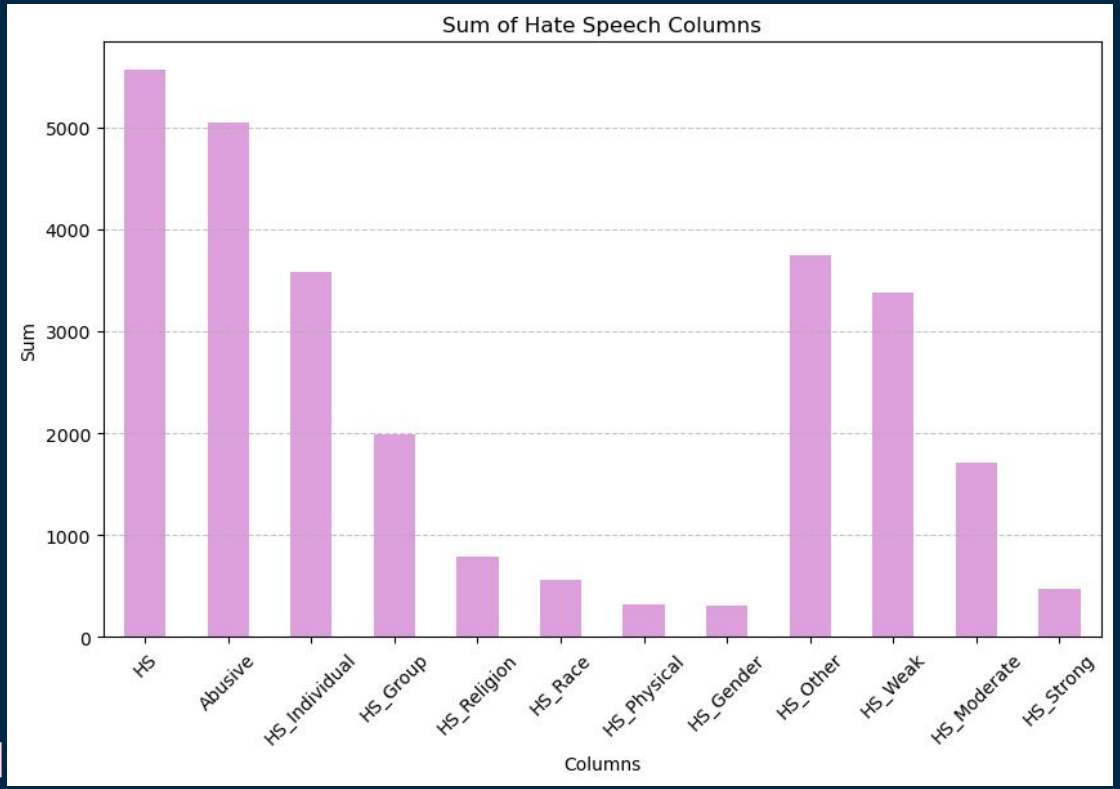
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13169 entries, 0 to 13168
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Tweet           13169 non-null  object
 1   HS              13169 non-null  int64
 2   Abusive         13169 non-null  int64
 3   HS_Individual   13169 non-null  int64
 4   HS_Group        13169 non-null  int64
 5   HS_Religion     13169 non-null  int64
 6   HS_Race         13169 non-null  int64
 7   HS_Physical     13169 non-null  int64
 8   HS_Gender       13169 non-null  int64
 9   HS_Other        13169 non-null  int64
 10  HS_Weak         13169 non-null  int64
 11  HS_Moderate     13169 non-null  int64
 12  HS_Strong       13169 non-null  int64
dtypes: int64(12), object(1)
memory usage: 1.3+ MB
```

In [14]:
```
# Check dataset info
df_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13169 entries, 0 to 13168
Data columns (total 14 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Tweet           13169 non-null  object
 1   HS              13169 non-null  int64
 2   Abusive         13169 non-null  int64
 3   HS_Individual   13169 non-null  int64
 4   HS_Group        13169 non-null  int64
 5   HS_Religion     13169 non-null  int64
 6   HS_Race         13169 non-null  int64
 7   HS_Physical     13169 non-null  int64
 8   HS_Gender       13169 non-null  int64
 9   HS_Other        13169 non-null  int64
 10  HS_Weak         13169 non-null  int64
 11  HS_Moderate     13169 non-null  int64
 12  HS_Strong       13169 non-null  int64
 13  Tweet_cleaned   13169 non-null  object
dtypes: int64(12), object(2)
memory usage: 1.4+ MB
```

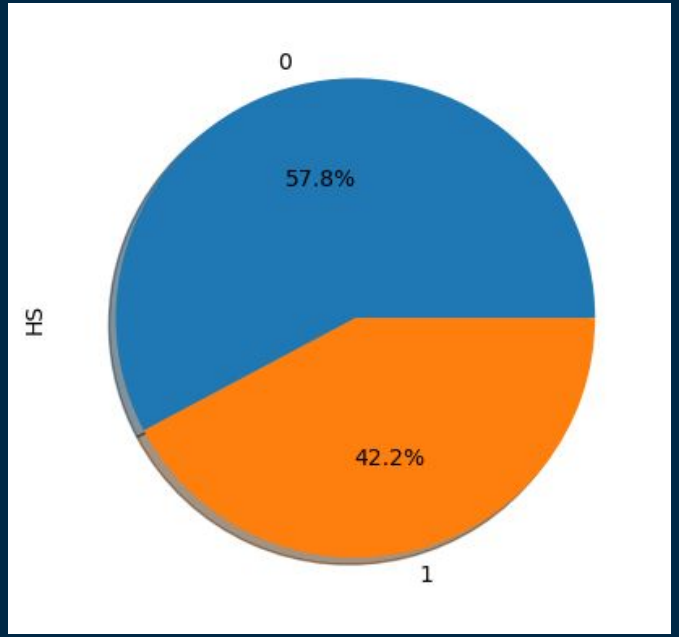
Terlihat bahwa tidak ada row yang dihilangkan artinya tidak ada data null sebelum dan sesudah cleansing

Perbandingan Sum Hate Speech



Terlihat dari hasil perbandingan jumlah Tweet yang mengandung Hate Speech dan abusive sama-sama tinggi. Namun secara specific, Hate speech kategori Other, Weak, dan Group tertinggi. Sedangkan terendah ada pada Race, Physical, Gender

Persentase Hate Speech

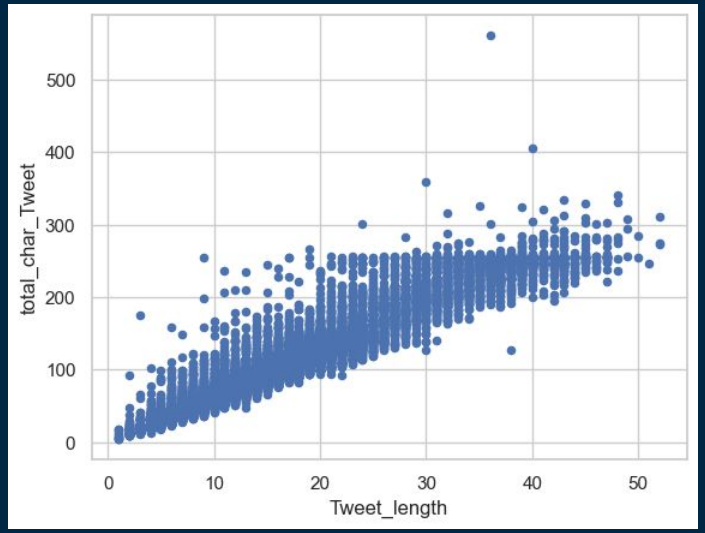


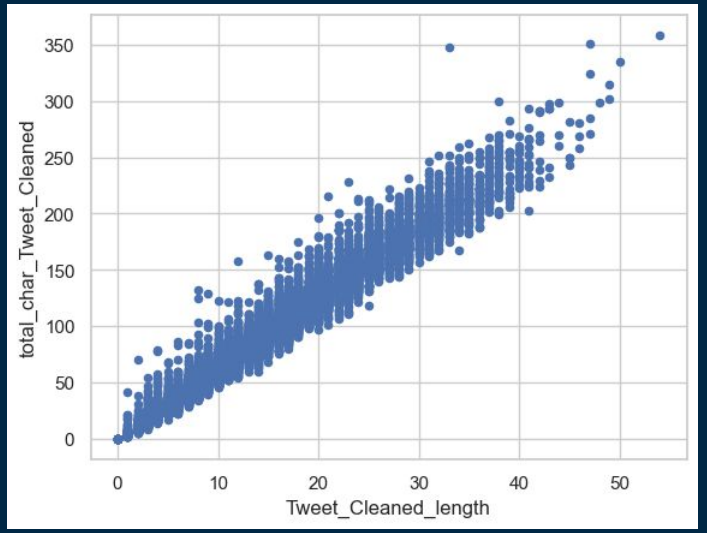
Dari data.csv, 42.2% tweet dikategorikan sebagai hate speech, sedangkan 57.8% lainnya tidak mengandung hate speech sama sekali. Artinya lebih dari setengah data bukan merupakan hate speech namun terdapat kemungkinan terdapat kata abusive

WordCloud Tweets yang Sudah Dibersihkan

Dari WordCloud data, kata-kata yang paling menonjol tidak menunjukkan sentiment tertentu, seperti 'Yang', 'Kamu', 'Tidak'. Namun, ada juga kata-kata yang terkait dengan topik tertentu, terutama politik dan negara. Misalnya, 'Jokowi', 'Prabowo', dan 'Ahok' mencerminkan topik politik, sementara frase seperti 'ganti presiden' dan 'berantas korupsi', serta penyebutan lembaga pemerintah seperti 'presiden', 'wakil rakyat', 'kepala daerah', menunjukkan sentiment politik. Mayoritas tweet dalam data.csv berfokus pada ranah politik. Selain itu, terdapat kata-kata yang merepresentasikan agama seperti 'Islam' dan 'Kristen'. Hate speech terkait agama juga dapat ditemukan dalam kolom 'HS_Religion'.

Scatter Jumlah Kata dengan Jumlah Karakter





Terlihat bahwa scatter antara jumlah kata dengan jumlah karakter tweet original dengan yang sudah dibersihkan sama-sama positif. Namun terlihat Tweet sudah dibersihkan lebih padat sendangkan tweet yang original terdapat beberapa outlier yang cukup jauh.