API

TEXT

CLENSING

GILANG RANU ASANAGARI

PENDAHULUAN

Media social Twitter merupakan media social yang sangat digemari untuk mencari dan menyebarkan informasi. Untuk penggunanya, Twitter dapat mengirim pesan (Tweet) serta membaca dan mengomentari pesan dari pengguna lain. Namun dalam penyampaian pesan tersebut, terkadang mempunyai makna yang negative dan positif. Hal yang dapat digunakan untuk menganalisa berbagai tulisan atau pesan tersebut memerlukan clensing data yang berguna untuk mengindentifikasi pesan twitter yang bersifat positif dan negative.

METODE PENELITIAN

1. Deskripsi data
2. Metode analysist data
3. Metode statistic/ machine learning/ visualisasi yang dipakai

API TEXT PROCESSING

Untuk membuat API, disini menggunakan library :

- Flask untuk membuat API
- Pandas untuk load csv file
- Regex untuk replace text
- Swagger sebagai UI dan API
- Sqlite3 sebagai tempat penyimpanan data yang telah diproses
- Json format file berbasis teks digunakan dalam proses pertukaran data antara server dan klien

Hasil yang didapat dari kode tersebut adalah :

- Menghilangkan kata Abusive berdasarkan referensi data yang diberikan (abusive.csv)
- Membenarkan kata yang telah salah berdasarkan refensi (new_kamusalay.csv)
- Menghilangkan kata yang tidak perlu seperti (website)
- Menyimpan hasil dari proses cleansing ke dalam sqlite3

```python
@swag_from("C:/Users/skyne/Documents/docs/text_processing.yml", methods=['POST'])
@app.route('/text-processing', methods=['POST'])
def text_processing():
    global text,new_list
    text = request.form.get('text')

    text = re.sub('\n',' ',text)
    text = re.sub('rt',' ',text)
    text = re.sub('RT',' ',text)
    text = re.sub('user',' ',text)
    text = re.sub('USER',' ',text)
    text = re.sub('((www\.[^\s]+)|(http?://[^\s)+))',' ',text)
    text = re.sub(' +',' ',text)
    return text

    json_response = {
        'status_code': 200,
        'description': "Teks yang sudah diproses",
        'data': re.sub(r'[^a-zA-Z0-9]',' ', text)
    }

    response_data = jsonify(json_response)
    return response_data
```

Data cleansing

Hasil yang didapat dari kode tersebut adalah :

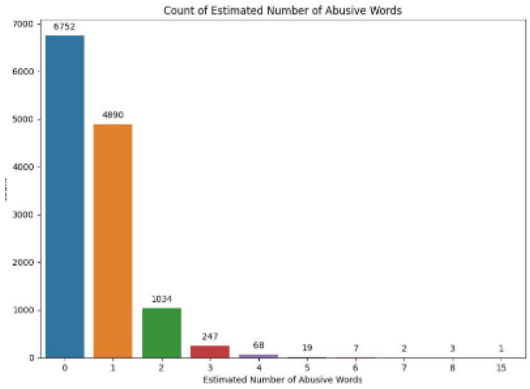
- Menghilangkan kata Abusive berdasarkan referensi data yang diberikan (abusive.csv)
- Membenarkan kata yang telah salah berdasarkan refensi (new_kamusalay.csv)
- Menghilangkan kata yang tidak perlu seperti (website)
- Menyimpan hasil dari proses cleansing ke dalam sqlite3

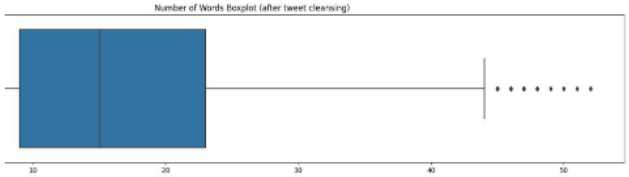
Namun untuk melakukan cleansing data sebanyak 13.701 row membutuhkan waktu yang tidak sedikit sehingga diperlukan kesabaran.

```
# DEFINE ENDPOINTS: POST FOR TEXT PROCESSING FROM FILE
@swag_from("C:/Users/skyne/Documents/docs/file_processing.yml", methods=['POST'])
@app.route('/text-processing-file', methods=['POST'])
def text_processing_file():
    global post_df

    # USING REQUEST TO GET FILE THAT HAS BEEN POSTED FROM API ENDPOINT
    file = request.files.get('file')

    # IMPORT FILE OBJECT INTO PANDAS DATAFRAME (YOU CAN SPECIFY NUMBER OF ROWS IMPORTED USING PARAMETER nrows=(integer value) )
    post_df = pd.read_csv(file, encoding='latin-1')

    # SET THE TWEET COLUMN ONLY FOR THE DATAFRAME
    post_df = post_df[['Tweet']]

    # DROP DUPLICATED TWEETS
    post_df.drop_duplicates(inplace=True)

    # CREATE NEW NUMBER OF CHARACTERS (NO_CHAR) COLUMN THAT CONSISTS OF LENGTH OF TWEET CHARACTERS
    post_df['no_char'] = post_df['Tweet'].apply(len)

    # CREATE NEW NUMBER OF WORDS (NO_WORDS) COLUMN THAT CONSISTS OF NUMBER OF WORDS OF EACH TWEET
    post_df['no_words'] = post_df['Tweet'].apply(lambda x: len(x.split()))

    # CREATE A FUNCTION TO CLEAN DATA FROM ANY NON ALPHA-NUMERIC (AND NON-SPACE) CHARACTERS, AND STRIP IT FROM LEADING/TRAILING SPACES
    def tweet_cleansing(x):
        tweet = x
        cleaned_tweet = re.sub(r'[^a-zA-Z0-9 ]','',tweet).strip()
        return cleaned_tweet

    # APPLY THE TWEET_CLEANSING FUNCTION ON TWEET COLUMN, AND CREATE A NEW CLEANED_TWEET COLUMN
    post_df['cleaned_tweet'] = post_df['Tweet'].apply(lambda x: tweet_cleansing(x))

    # CREATE NEW NO_CHAR, AND NO_WORDS COLUMNS BASED ON CLEANED_TWEET COLUMN
    post_df['no_char_2'] = post_df['cleaned_tweet'].apply(len)
    post_df['no_words_2'] = post_df['cleaned_tweet'].apply(lambda x: len(x.split()))

    # CREATE A FUNCTION TO COUNT NUMBER OF ABUSIVE WORDS FOUND IN A CLEANED TWEET
    def count_abusive(x):
        cleaned_tweet = x
        matched_list = []
        for i in range(len(df_abusive)):
            for j in x.split():
                word = df_abusive['ABUSIVE'].iloc[i]
                if word==j.lower():
                    matched_list.append(word)
        return len(matched_list)
```

DATA VISUALISASI

Berikut adalah hasil visualisasi data menggunakan bar chart untuk mengetahui jumlah positif dan negative komentar tersebut dari sebuah pesan (Tweet)





KESIMPULAN

Berdasarkan hasil analysis pesan tweet yang tidak menggunakan kata Abusive terdapat 6752, dimana terdiri dari :

- Tweet yang menggunakan 1 kata Abusive yaitu sebanyak 4890
- Tweet yang menggunakan 2 kata Abusive yaitu sebanyak 1034
- Tweet yang menggunakan 3 kata Abusive yaitu sebanyak 247
- Tweet yang menggunakan 4 kata Abusive yaitu sebanyak 68
- Tweet yang menggunakan 5 kata Abusive yaitu sebanyak 19
- Tweet yang menggunakan 6 kata Abusive yaitu sebanyak 7
- Tweet yang menggunakan 7 kata Abusive yaitu sebanyak 2
- Tweet yang menggunakan 8 kata Abusive yaitu sebanyak 3
- Tweet yang menggunakan 15 kata Abusive yaitu sebanyak 1

-