

API FOR TEXT CLEANSING TWITTER COMMENTS

Adlan Muhammad Faras

Binar Academy

Pendahuluan

Latar Belakang

Menurut berbagai sumber, Indonesia merupakan salah satu negara dengan jumlah pengguna Twitter yang signifikan. Pada September 2023, Indonesia memiliki 24 juta pengguna aktif Twitter, menjadikannya negara Twitter terbesar kelima [1]. Jakarta juga adalah kota Twitter paling aktif di dunia, dan 7,5% dari seluruh tweet secara global berasal dari Indonesia [2]. Hal ini menunjukkan bahwa twitter memiliki dampak yang signifikan di kehidupan sehari-hari masyarakat Indonesia.

Sebuah studi menemukan bahwa ketika pengguna twitter membaca komentar bernuansa negatif di media sosial, efek penularannya akan meningkatkan kemarahan dan kecemasan, yang dapat menyebabkan lebih banyak hal negatif [3]. Mengingat relevansi twitter di kehidupan sehari-hari masyarakat Indonesia, diperlukan sebuah metode sederhana untuk membersihkan tweet dari kata abusive dan hate speech.

Rumusan Masalah

Dari fakta di atas, muncul beberapa masalah yang jadi perlu dibahas, yaitu:

1. Bagaimana persebaran sentimen tweet di twitter Indonesia
2. Bagaimana cara melakukan pembersihan tweet dari sentimen negatif dengan cara yang sederhana

Tujuan Penelitian

Oleh karena itu penelitian ini bertujuan untuk menganalisis persebaran sentimen komentar pada twitter. Selain itu, sebuah API yang mudah digunakan juga akan dibuat untuk membersihkan tweet dari sentimen negatif

Metode Penelitian

Deskripsi Data

Data pada penelitian ini bersumber dari [Kaggle](#). Lebih spesifik lagi data yang dianalisis adalah *multi-label hate speech and abusive language dataset* dari twitter indonesia.

Metode EDA/Statistik

Metode EDA yang digunakan adalah *Univariate Analysis* dengan juga dalam setiap prosesnya menerapkan metode Visualisasi. Visualisasi dilakukan untuk mencari tahu persebaran data secara visual.

Code Function

pembersihan tweet yang sederhana dari sentimen negatif bisa dilakukan dengan pembuatan API text cleansing. Ada dua endpoint yang dibuat yaitu text cleansing dan csv cleansing API.

Text cleansing API (text dari input user kemudian dibersihkan dari kata abusive dan kata alay)

```
#form cleaning api
@swag_from('docs/cleansing_form.yml', methods=['POST'])
@app.route('/cleansing_form', methods=['POST'])
def cleansing_form():
    # Get text from input user
    raw_text = request.form["raw_text"]

    # Text cleansing
    clean_text = text_cleansing(raw_text)
    result_response = {"raw_text": raw_text, "clean_text": clean_text}

    # Insert result to database
    db_connection = db_connect()
    db_insert_cleaned_form(db_connection, raw_text, clean_text)
    return jsonify(result_response)
```



```
def text_cleansing(text):
    # Menghilangkan Non-alphabet dan angka
    cleaned_text = re.sub(r'[^a-zA-Z0-9\s]', '', text)

    # Mengecilkan kata
    cleaned_text = cleaned_text.lower()

    # Mengubah kata abusive menjadi ***
    regex_pattern = re.compile(r'\b(?:' + '|'.join(map(re.escape, abusive_list)) + r')\b', flags=re.IGNORECASE)
    replacement_string = "***"
    cleaned_text = regex_pattern.sub(replacement_string, cleaned_text)

    # Mengganti kata alay dengan kata formal
    cleaned_text = " ".join(dict_alay.get(word, word) for word in cleaned_text.split())

    return cleaned_text
```

CSV cleansing API (file csv dari upload user kemudian dibersihkan dari kata abusive dari kata abusive dan kata alay)

```
#csv cleaning api
@swag_from('docs/cleansing_csv.yml', methods=['POST'])
@app.route('/cleansing_csv', methods=['POST'])
def cleansing_csv():
    # Get file from user upload
    uploaded_file = request.files["upload_file"]

    # CSV cleansing
    df_cleansing = file_cleansing(uploaded_file)

    # Insert result to database
    db_connection = db_connect()
    db_insert_cleaned_csv(db_connection, df_cleansing)
    print("Upload result to database success!")
    result_response = df_cleansing.T.to_dict()
    return jsonify(result_response)
```



```
def file_cleansing(file_upload):
    # Read csv file
    df_upload = pd.read_csv(file_upload, encoding="latin-1")

    # Ambil kolom text
    df_upload = pd.DataFrame(df_upload.iloc[:,0])

    # Rename kolom menjadi "raw_text"
    df_upload.columns = ["raw_text"]

    # Bersihkan text menggunakan fungsi text_cleansing dan simpan di kolom "clean_text"
    df_upload["clean_text"] = df_upload["raw_text"].apply(text_cleansing)
    print("Cleansing text success!")
    return df_upload
```

Contoh penggunaan Text cleansing API

Contoh penggunaan Text cleansing API

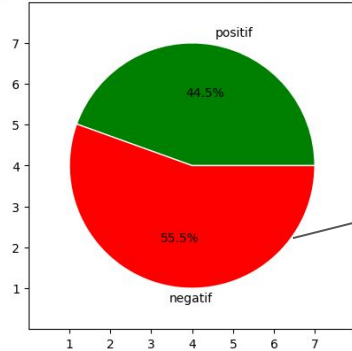
```
{
  "clean_text": "dasar *** kamu",
  "raw_text": "dasar babi loe!"
}
```

Contoh penggunaan CSV cleansing API

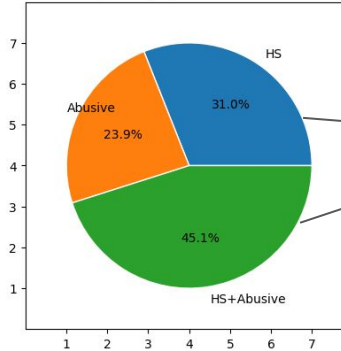
[illegible]

Visualisasi dan Analisis Data

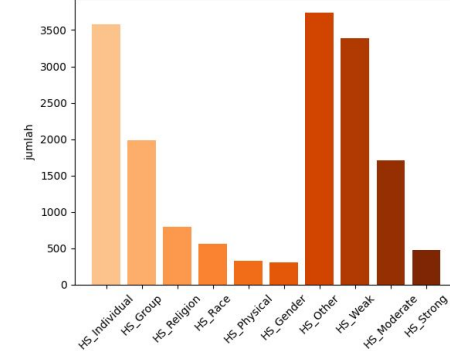
persebaran komentar bersentimen positif v negatif



persebaran kata Hate Speech (HS) dan Abusive dalam komentar negatif



distribusi jenis hate speech pada komentar negatif yang berisi hate speech



Univariate analysis dengan visualisasi diatas digunakan untuk memahami persebaran sentimen komentar di twitter Indonesia. Dari 13169 sampel tweet, sebesar 55.5% memiliki sentimen negatif (mengandung *hate speech* atau kata *abusive*). Kemudian, persebaran dari itu, kebanyakan memiliki *hate speech* dan kata *abusive* sebesar 45.1%. Sampel tweet yang mengandung *hate speech* dibagi lagi menjadi beberapa jenis. Paling banyak adalah yang berjenis *hate speech* other dan individual.

Hasil dan Kesimpulan

Berdasarkan analisis yang sudah kita lakukan dapat hasilnya dapat dijabarkan sebagai berikut::

Dalam visualisasi menunjukkan persebaran sentimen tweet di twitter Indonesia:

- Persebaran sentimen dari sampel tweet indonesia adalah 55.5% negatif dan 44.5% positif
- Sampel tweet negatif 23.9% hanya mengandung kata abusive, 31% hate speech, dan 45.1% mengandung kata abusive dan hate speech

pembersihan tweet dari sentimen negatif dengan cara yang sederhana bisa dilakukan dengan API text cleansing. Hal ini dibuktikan dengan contoh penggunaan api yang dapat dilakukan dengan mudah dan intuitif