**CSGE601020**

**Dasar-Dasar Pemrograman 1**

# Tugas Pemrograman 3: Sentiment Analysis

(Batas Tenggat Akhir: 29 November 2021)

## 1. Tujuan

Tugas Pemrograman 3 memberikan pengalaman terhadap penggunaan fungsi, string, *text processing*, **list**, **dictionary**, dan **set** dalam menyelesaikan permasalahan populer. Peserta juga diperkenalkan dengan library python (**matplotlib**) yang berguna dalam bidang *data science* dan sering digunakan untuk visualisasi properti data.

## 2. Penilaian

Kebenaran program (60%), Penjelasan saat demo di depan Asisten dosen (30%), dokumentasi & kerapihan (10%).

## 3. Deskripsi

*Sentiment* merupakan ekspresi subjektif manusia yang dialamatkan untuk suatu entitas tertentu, seperti produk, film, atau layanan. Membuat rangkuman dari banyak ekspresi *sentiment* yang dihasilkan oleh banyak pengguna media sosial tentunya merupakan pengetahuan yang sangat berharga bagi perusahaan penghasil suatu produk atau perusahaan penyedia suatu jasa. Berikut adalah contoh beberapa kalimat yang mengandung *sentiment* dan orientasinya (domain review sebuah film).

1. a thoughtful, provocative, insistently humanizing film. (orientasi: positif)
2. a gentle, compassionate drama about grief and healing. (orientasi: positif)
3. an odd, haphazard, and inconsequential romantic comedy. (orientasi: negatif)
4. the movie is a mess from start to finish. (orientasi: negatif)

Peserta diminta untuk membuat program Python yang melakukan analisis kalimat-kalimat yang memiliki sentiment, dan kemudian membangun program sederhana untuk memprediksi jenis orientasi dari suatu kalimat subjektif.

### 3.1 Bagian Pertama (*Normalized Difference Sentiment Index*)

Di bagian pertama, Anda diminta untuk melengkapi file **ndsi.py**. Anda diberikan beberapa elemen berikut:

1. Folder **sent-polarity-data** yang berisi dua buah file teks, yaitu **rt-polarity.neg** dan **rt-polarity.pos**. rt-polarity.neg berisi **5000** kalimat subjektif yang berorientasi

negatif. Sebuah baris pada file tersebut merepresentasikan sebuah kalimat. Serupa dengan rt-polarity.neg, **rt-polarity.pos** berisi 5000 kalimat subjektif yang berorientasi positif.[1]

2. File **stopwords.txt** yang berisi *stop words*, seperti yang sudah Anda gunakan di Tugas Pemrograman 2.

Inti dari bagian pertama ini adalah Anda diminta untuk membuat file teks yang berisi daftar sebuah kata beserta nilai ***Normalized Difference Sentiment Index*** **(NDSI)**. NDSI adalah sebuah nilai bekisar antara **-1** dan **+1**, yang merepresentasikan orientasi sentiment dari sebuah kata. Nilai -1 artinya sangat negatif nilai +1 artinya sangat positif. Sebagai contoh, kata "baik" diharapkan mempunyai NDSI > 0 dan kata "buruk" mempunyai NDSI < 0. NDSI dari sebuah kata dihitung dengan

$$NDSI(word) = \frac{freq^{+}(word) - freq^{-}(word)}{freq^{+}(word) + freq^{-}(word)},$$

dimana:

- $freq^{+}(word)$ adalah banyaknya kemunculan sebuah kata pada dokumen rt-polarity.pos
- $freq^{-}(word)$ adalah banyaknya kemunculan sebuah kata pada dokumen rt-polarity.neg

Intuisi dari NDSI ini adalah jika sebuah kata lebih sering muncul di dokumen yang berisi kalimat-kalimat positif dibandingkan negatif, maka kata tersebut mempunyai kecenderungan bersifat positif; dan sebaliknya.

Pada file **ndsi.py**, ada beberapa fungsi atau fitur yang harus Anda implementasikan:

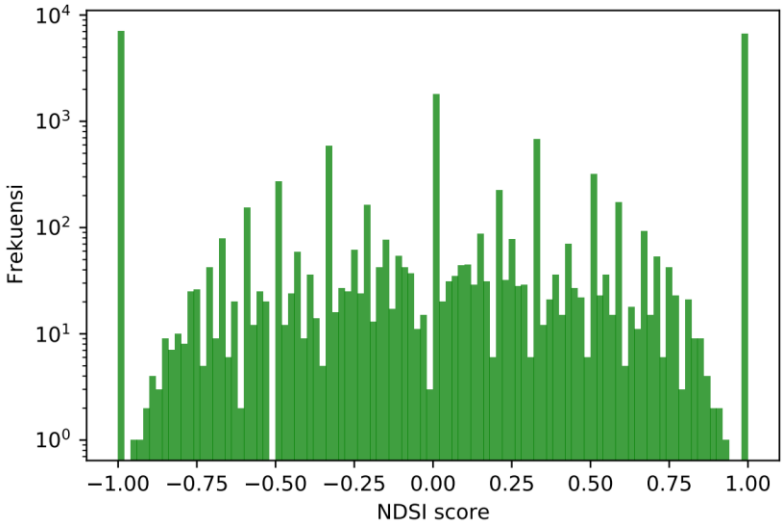
1. `load_stop_words(filename)`: Fungsi menerima nama file yang berisi daftar stopwords (**stopwords.txt**), kemudian memuat semua stopwords ke dalam struktur data **set**. Perhatikan bahwa semua stopwords yang ada di dalam file sudah dalam bentuk huruf kecil semua.
2. `count_words(filepath, stop_words)`: Fungsi ini akan scan semua baris (semua kalimat) yang ada di file yang terletak di `filepath` dan kemudian mengakumulasikan frekuensi dari setiap kata yang muncul pada file tersebut. Fungsi mengembalikan sebuah *dictionary*, dimana *key* adalah kata (string) dan *value* adalah frekuensi (int) dari kemunculan kata tersebut. Kata-kata yang berupa *stopwords* atau tanda baca akan diabaikan.
3. `compute_ndsi(word_freq_pos, word_freq_neg)`: Fungsi yang menghitung NDSI dari kata-kata diberikan dua buah dictionary, yaitu `word_freq_pos` yang berisi frekuensi kata-kata pada file rt-polarity.pos dan `word_freq_neg` yang berisi frekuensi kata-kata pada file rt-polarity.neg. Silakan lihat deskripsi fungsi di file kode **ndsi.py**.
4. Program utama yang menghasilkan sebuah file teks bernama **ndsi.txt** (note: berbeda dengan ndsi.py). Isi dari file ini adalah daftar kata-kata beserta nilai NDSI-nya

[1] https://www.cs.cornell.edu/people/pabo/movie-review-data/

(dipisahkan dengan **spasi**), diurutkan dari yang paling kecil (paling negatif) ke yang paling besar (paling positif). Contoh isi dari file **ndsi.txt**:

```
larded -1.0
sands -1.0
rough-hewn -1.0
andie -1.0
flat -0.9444444444444444
mediocre -0.9230769230769231
mindless -0.8888888888888888
stale -0.8888888888888888
boring -0.8867924528301887
stupid -0.8666666666666667
...
rewards 0.7142857142857143
glimpse 0.7142857142857143
son 0.7142857142857143
richer 0.7142857142857143
sweetness 0.7142857142857143
vitality 0.7142857142857143
decade 0.7142857142857143
nuance 0.7142857142857143
...
```

Anda juga diberikan sebuah fungsi siap pakai `show_ndsi_histogram(word_ndsi)` yang menampilkan distribusi (dalam bentuk histogram) dari nilai-nilai NDSI yang dihasilkan. Seandainya Anda benar mengimplementasikan fungsi `compute_ndsi`, seharusnya histogram yang dihasilkan adalah (kira-kira) seperti berikut:




### 3.2 Bagian Kedua (Prediksi)

Di bagian kedua, Anda diminta untuk melengkapi file **predict.py**. Anda akan menggunakan daftar kata-kata beserta nilai NDSI yang dihasilkan untuk memprediksi orientasi sentiment dari kalimat-kalimat yang tidak diketahui labelnya (apakah positif atau negatif). Anda diberikan sebuah file bernama **sent-unknown-label.txt** yang berisi **662** kalimat yang ingin diprediksi jenis orientasinya.

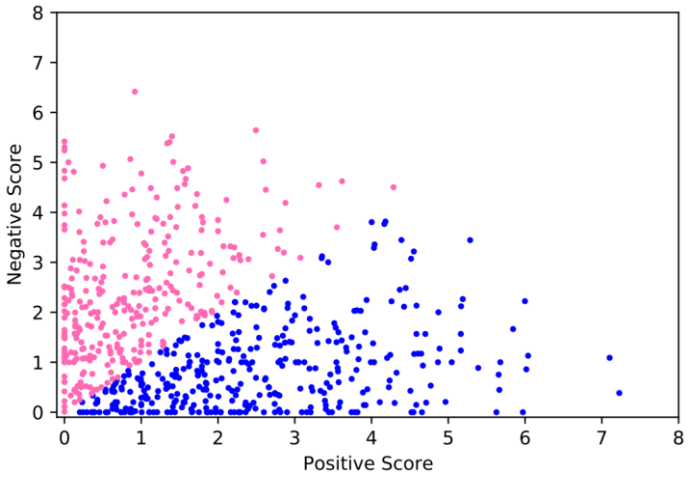
Pada file **predict.py**, ada dua fungsi atau fitur yang harus Anda implementasikan:

1. `load_ndsi(ndsi_filename)`: Fungsi ini memuat daftar kata-kata dan nilai NDSI yang bersesuaian ke dalam sebuah dictionary (dari file **ndsi.txt**), dimana key adalah kata (string) dan value adalah NDSI score (float) dari kata tersebut.
2. `compute_score(filename, word_ndsi)`: Fungsi ini mengembalikan *list of pairs*, dimana setiap elemen merupakan pasangan (**`positive score, negative score`**) untuk sebuah kalimat. Sebuah kalimat akan diklasifikasikan sebagai positif jika `positive score > negative score`; dan sebaliknya. Jika kedua nilai sama, kalimat diklasifikaskan sebagai **netral**. Silakan lihat deskripsi fungsi pada file **predict.py** untuk penjelasan lebih lanjut.

Contoh output yang dihasilkan di layar:

```
...
sentence 165 -- pos: 0.622  neg: 0.415  prediction:pos
sentence 166 -- pos: 1.450  neg: 1.339  prediction:pos
sentence 167 -- pos: 1.615  neg: 1.136  prediction:pos
sentence 168 -- pos: 0.976  neg: 2.390  prediction:neg
sentence 169 -- pos: 0.606  neg: 0.796  prediction:neg
sentence 170 -- pos: 0.784  neg: 4.356  prediction:neg
sentence 171 -- pos: 0.000  neg: 3.656  prediction:neg
...
```

Anda juga diberikan sebuah fungsi siap pakai `show_scatter_plot` yang menampilkan *scatter plot* untuk semua kalimat di **sent-unknown-label.txt**. Sumbu X merupakan nilai positif dan sumbu Y merupakan nilai negatif. Jika Anda benar dalam mengimplementasikan fungsi `compute_score`, kira-kira akan muncul seperti berikut:




Titik-titik warna biru yang berada di bawah garis **Y = X** merupakan kalimat-kalimat yang diprediksi sebagai kalimat ber-sentiment positif, sedangkan yang berwarna merah diprediksi sebagai kalimat ber-sentiment negatif.