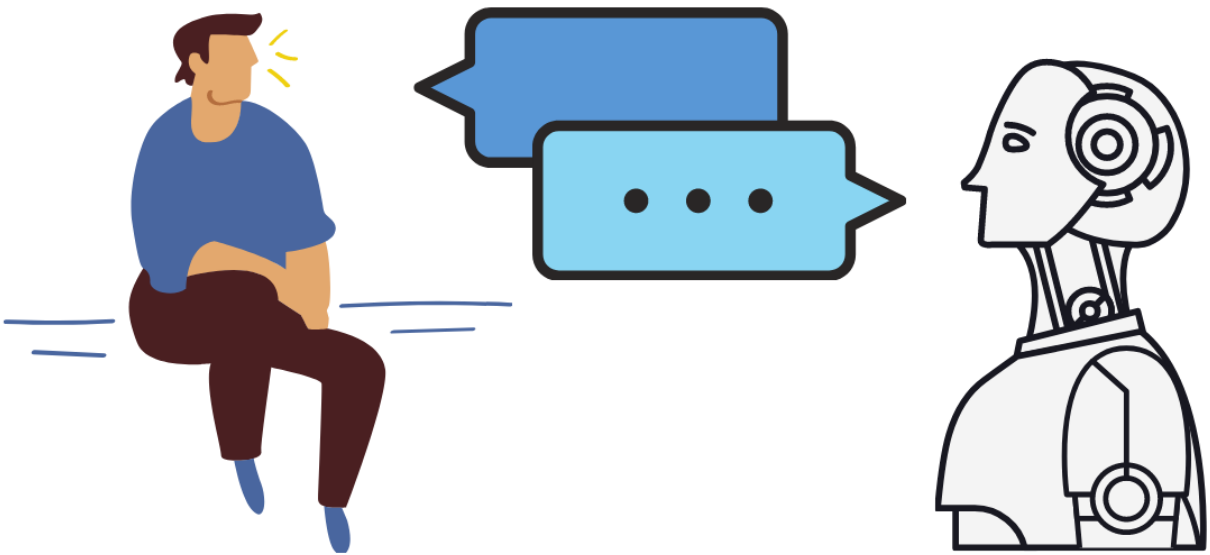
Pemrosesan Bahasa Alami

Analisis Karakteristik Teks

Berdasarkan Komposisi *POS Tag*

Maria Veronica Claudia M., M.T.

Semester Genap 2020/2021



PETUNJUK:

1. Kerjakan berkelompok, masing-masing kelompok terdiri dari dua orang.
2. Untuk kali ini, Anda tidak diberi *template*, tetapi Anda tetap diberi contoh dan petunjuk. Silakan buat modul dan fungsi masing-masing, beri nama yang DESKRIPTIF agar mudah dipahami, jangan asal-asalan.
3. Beri penjelasan lengkap berupa komen pada kode program Anda.
4. Satukan seluruh pekerjaan Anda ke dalam folder dengan format penamaan T04xxyyy_xxyyy. Kumpulkan dalam bentuk zip.

Pendahuluan

Setiap jenis teks memiliki karakteristik yang beragam. Karakteristik dapat dilihat dari berbagai hal, salah satu contoh dari apa yang telah dipelajari di minggu-minggu yang lalu adalah bentuk penulisan. Jika dibandingkan dengan teks formal, teks yang berasal dari media sosial relative lebih berantakan dan tidak beraturan, banyak terdapat singkatan, *slank*, dan karakter-karakter spesial.

Minggu ini, Anda telah mendapatkan materi mengenai POS *tag*. Untuk itu, pada praktikum kali ini Anda akan melakukan eksperimen untuk membuktikan apakah karakteristik teks dapat dibedakan menurut komposisi *POS tag*. Jenis teks yang digunakan adalah teks formal dengan tujuan berbeda. Satu dokumen memuat teks-teks yang bertujuan untuk *product review*, sedangkan dokumen yang lain memuat teks cerita pendek.

Hipotesa yang akan dibuktikan dalam eksperimen ini adalah sebagai berikut.

> Secara logika, komposisi POS *tag* pada teks *product review* akan didominasi oleh kata benda dan kata sifat. Sedangkan pada cerita pendek, seharusnya kata kerja lebih mendominasi.

Tugas Anda adalah menganalisis komposisi POS *tag* kedua jenis teks tersebut dan membuktikan apakah hipotesa di atas dapat diterima.

Eksperimen

Bagian ini merupakan petunjuk untuk eksperimen Anda. Eksperimen terdiri dari 3 bagian, yaitu penyiapan data, analisis, dan visualisasi.

Penyiapan Data

Dengan memanfaatkan petunjuk dan contoh-contoh di bawah ini, buatlah sebuah fungsi untuk menyiapkan data dan menambahkan fitur POS *tag* ke dalam teks.

Anda telah diberi dua buah dokumen, yaitu *SelfishGiant.txt* dan *SamsungReview.txt*. Untuk membuka dan membaca file *txt*, Anda tidak perlu menggunakan *pandas*. Cukup menggunakan baris kode di bawah.

```
In [1]: open("yourFile","r").read()
```

Kedua teks yang disediakan sudah bersih sehingga Anda tidak perlu membuat fungsi *cleaning text* lagi. Tanda baca perlu dihapus, tetapi setelah tokenisasi paragraph menjadi kalimat. Hasil membaca sebuah file *txt* akan berupa String, simpanlah kedua teks tersebut ke dalam variabel.

Untuk memisahkan kalimat-kalimat dalam sebuah teks (khususnya yang berbahasa Inggris), Anda dapat menggunakan *sent_tokenize* dari *nltk library*. Perhatikan contoh berikut.

```
In [5]: from nltk import sent_tokenize

In [6]: para = "This is a paragraph. This paragraph has several sentences. This is
the third sentence. And this is the last one."

In [7]: stcList = sent_tokenize(para)

In [8]: stcList
Out[8]:
['This is a paragraph.',
 'This paragraph has several sentences.',
 'This is the third sentence.',
 'And this is the last one.']
```

Setelah Anda mendapatkan *list of sentences* dan membersihkan tanda baca, Anda dapat mulai menambahkan fitur POS *tag* ke dalam setiap kalimat. Ingat, Anda perlu melakukan tokenisasi terlebih dahulu untuk memisahkan kata-kata di setiap kalimat. Untuk eksperimen kali ini, Anda tidak perlu membuat sendiri tokenisasi dan POS *tagger* yang diperlukan. Silakan gunakan *word_tokenize* dan *pos_tag* dari *nltk library*. Perhatikan contoh berikut.

```
In [14]: from nltk import word_tokenize, pos_tag

In [15]: tokenizedSentences = [word_tokenize(sentence) for sentence in stcList]

In [16]: print(tokenizedSentences)
[['This', 'is', 'a', 'paragraph', '.'], ['This', 'paragraph', 'has', 'several',
'sentences', '.'], ['This', 'is', 'the', 'third', 'sentence', '.'], ['And', 'this',
'is', 'the', 'last', 'one', '.']]

In [17]: taggedSentences = [pos_tag(stc) for stc in tokenizedSentences]
```

Anda dapat memodifikasi contoh di atas dengan menggabungkan [15] dan [17] menjadi satu baris perintah (*word_tokenize* langsung dipanggil saat pemberian *pos_tag*). Anda juga dapat mengganti jenis *tag* menjadi *tagset* universal dengan menambahkan parameter *tagset* pada *pos_tag*.

```
pos_tag(stc, tagset='universal')
```

Fungsi *pos_tag* pada contoh di atas akan mengembalikan *list* of *list-of-sets*, dimana setiap *list* berisi beberapa *set*, dan setiap *set* berisi kata dan POS *tag* untuk kata tersebut. Perhatikan isi dari variabel *posTagged* di bawah ini (menggunakan *tagset default*, bukan *universal*).

```
In [18]: print(taggedSentences)
[[('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('paragraph', 'NN'), ('.', '.')],
[('This', 'DT'), ('paragraph', 'NN'), ('has', 'VBZ'), ('several', 'JJ'),
('sentences', 'NNS'), ('.', '.')], [('This', 'DT'), ('is', 'VBZ'), ('the', 'DT'),
('third', 'JJ'), ('sentence', 'NN'), ('.', '.')], [('And', 'CC'), ('this', 'DT'),
('is', 'VBZ'), ('the', 'DT'), ('last', 'JJ'), ('one', 'NN'), ('.', '.')]]
```

Analisis

Dengan memanfaatkan petunjuk dan contoh-contoh di bawah ini, buatlah sebuah fungsi untuk menghitung komposisi POS *tag* dalam sebuah dokumen.

Komposisi POS *tag* yang sudah disinggung di awal modul dapat dihitung sebagai berikut.

$$Comp(t) = \frac{\sum_{i=0}^{n}[tag_i = t]}{count(T)}, tag_i \in T$$

Sebagai contoh, jika dalam dokumen terdapat kata dan POS *tag* sebagai berikut:

('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('paragraph', 'NN'), ('.', '.')

dengan mengabaikan '.', maka komposisi POS *tag* yang didapat adalah sebagai berikut:

$$Comp(DT) = \frac{2}{4};\ Comp(VBZ) = \frac{1}{4};\ Comp(NN) = \frac{1}{4}$$

Kembali ke kode program, hasil yang Anda dapatkan dari fungsi *pos_tag* sebelumnya adalah sebagai berikut.

```
In [18]: print(taggedSentences)
[[('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('paragraph', 'NN'), ('.', '.')],
[('This', 'DT'), ('paragraph', 'NN'), ('has', 'VBZ'), ('several', 'JJ'),
('sentences', 'NNS'), ('.', '.')], [('This', 'DT'), ('is', 'VBZ'), ('the', 'DT'),
('third', 'JJ'), ('sentence', 'NN'), ('.', '.')], [('And', 'CC'), ('this', 'DT'),
('is', 'VBZ'), ('the', 'DT'), ('last', 'JJ'), ('one', 'NN'), ('.', '.')]]
```

Untuk mengakses nilai di dalam *set,* Anda dapat langsung menggunakan variabel sejumlah elemen di dalam *set* tersebut. Perhatikan contoh berikut.

```
In [19]: listOfSet = [(1,2), (3,4), (5,6)]

In [20]: for a, b in listOfSet:
    ...:     print(a+b)
    ...:
3
7
11
```

Setelah Anda menemukan cara untuk menghitung kemunculan setiap POS *tag* dan komposisinya, simpanlah hasil perhitungan Anda ke dalam sebuah *dictionary*. Sebagai contoh:

{'DT':0.5, 'VBZ':0.25, 'NN': 0.25}

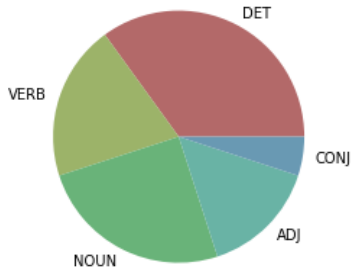
Reminder: Anda dapat menambahkan *key* baru pada *dictionary* dan mengubah *value* dari sebuah *key* yang sudah ada di *dictionary*. Perhatikan contoh berikut.

```
In [21]: dicti = {'a':1}

In [22]: lst = [('a', 2),('b',3)]

In [23]: for first, second in lst:
    ...:     if first in dicti.keys():
    ...:         dicti[first] += second
    ...:     else:
    ...:         dicti[first] = second
    ...:

In [24]: print(dicti)
{'a': 3, 'b': 3}
```

Visualisasi

Untuk mempermudah pengguna dalam menganalisis hasil eksperimen, buatlah sebuah fungsi untuk menampilkan hasil penghitungan komposisi dokumen dengan memanfaatkan visualisasi. Visualisasi yang digunakan bebas (*bar chart, pie chart*, dan sebagainya), yang menurut Anda sesuai dan dapat digunakan untuk melihat perbandingan komposisi kedua jenis dokumen.

Hasil akhir yang diharapkan adalah dengan melihat visualisasi komposisi, kesimpulan penerimaan / penolakan hipotesa dapat diambil. Berikut contoh hasil visualisasi komposisi POS *tag* (dari paragraf yang digunakan dalam contoh-contoh di atas, dengan *tagset* universal).



Daftar kode POS *tag* universal dapat Anda lihat di https://universaldependencies.org/u/pos/

Petunjuk tambahan: salah satu *library* yang dapat Anda gunakan untuk membuat visualisasi adalah *matplotlib*. Untuk membuat *chart*, Anda perlu memisahkan terlebih dahulu *key* dan *value* pada *dictionary* dan menyimpan masing-masing ke dalam *list* yang berbeda. Perhatikan contoh berikut.

```
In [44]: dicti
Out[44]: {'a': 3, 'b': 3}

In [45]: label = list(dicti.keys())

In [46]: label
Out[46]: ['a', 'b']

In [47]: values = list(dicti.values())

In [48]: values
Out[48]: [3, 3]
```

SELAMAT MENGERJAKAN

** SUMBER DATASET

SamsungReview: https://www.techradar.com/reviews/samsung-galaxy-s20-full-review

SelfishGiant: https://americanliterature.com/author/oscar-wilde/short-story/the-selfish-giant