CSCE604135 • Perolehan Informasi
Semester Ganjil 2021/2022
Fakultas Ilmu Komputer, Universitas Indonesia

Tugas 2

Tenggat Waktu: 7 Oktober 2021, 23.55 WIB

Ketentuan:

1. Dataset yang digunakan pada tugas ini telah disediakan di SCeLe.
2. Buatlah program Jupyter Notebook yang menjawab pertanyaan sesuai dengan perintah soal yang disediakan.
3. Program Jupyter Notebook yang telah dibuat dikumpulkan dengan format penamaan **TugasX_NPM_Nama.ipynb**
   Contoh: Tugas2_1706043361_Rd Pradipta Gitaya Samiadji.ipynb
4. Kumpulkan dokumen tersebut pada submisi yang telah disediakan di SCeLe sebelum tanggal **7 Oktober 2021, 23.55 WIB**. Keterlambatan pengumpulan akan dikenakan pinalti.
5. Tugas ini dirancang sebagai **tugas mandiri**. Plagiarisme tidak diperkenankan dalam bentuk apapun. Adapun kolaborasi berupa diskusi (tanpa menyalin maupun mengambil jawaban orang lain) dan literasi masih diperbolehkan dengan mencantumkan **kolaborator** dan **sumber**.
6. Untuk soal-soal pemrograman Anda dibebaskan menggunakan bahasa pemrograman apa saja tetapi untuk mempermudah, kami merekomendasikan bahasa Python.

A - Edit Distance (40 Poin)

1. **[10]** Buatlah sebuah fungsi "**edit_distance**". Fungsi tersebut memiliki **dua** buah parameter string yaitu **string_1** dan **string_2**. Fungsi ini dapat menghitung **Levenshtein Distance** antara string_1 dengan string_2. Lebih jelasnya lagi, fungsi tersebut akan mengembalikan **nilai terkecil** yang dibutuhkan untuk **mentransformasi** string_1 menjadi string_2.

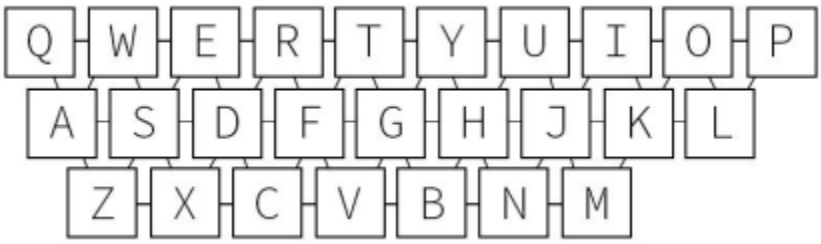
2. **[5]** Menggunakan fungsi yang telah dibuat pada soal sebelumnya, carilah **nilai edit_distance** dari pasangan kata berikut ini:
    a. phasmophobia - puafnilhotik
    b. genetik - ganteng

***Catatan A1 dan A2:**

- Diperbolehkan membuat *helper method* (jika diperlukan)
- Tidak diperbolehkan untuk menggunakan *library* apapun
- Untuk detail terkait algoritma Levenshtein Distance, telah disediakan penjelasan singkat mengenai algoritma tersebut pada bagian Lampiran **(Credit: Tim Asdos IR 2019)**

3. **[13]** Salah satu metode untuk menghitung *edit distance* secara heuristik adalah menggunakan "***keyboard distance***". Penentuan "**bobot**" atau "**weight**" pada metode ini dapat bervariasi. Penggunaan **graf** dari *keyboard* (asumsikan QWERTY *keyboard*) merupakan salah satu metode yang dapat dilakukan.

    Pada graf yang akan digunakan, satu ***node*** merepresentasikan satu ***key***. Kemudian, satu ***edge*** menandakan bahwa dua *key* **berdekatan** yang artinya memiliki **bobot** *edit distance* sebesar 1. Berikut adalah tampilan graf yang dihasilkan dari suatu *keyboard* QWERTY.



Menggunakan ilustrasi graf tersebut, buatlah **matriks *edit distance* M** dari suatu QWERTY keyboard! **M(*ii*, *jj*)** menyatakan **jarak terpendek** antara huruf "***ii***" ke huruf "***jj***" dalam graf QWERTY *keyboard*. Contoh matriks **M** yang dihasilkan dapat dilihat pada gambar di bawah.

	a	**s**	**d**	**...**	**l**	**p**
a	0	1	2	...	8	9
s	1	0	1	...	7	8
d	2	1	0	...	6	7
...	...	...	...	0	...	...
l	8	7	6	...	0	1
p	9	8	7	...	1	0

***Catatan A3:**

Kamu diperbolehkan membuat matriks M secara manual dengan mengetikkannya secara langsung ke dalam *notebook*. Tetapi, hal ini tentunya tidak dapat memperoleh nilai maksimal.

4. **[5]** Menggunakan matriks **M** yang telah dibuat pada soal sebelumnya, carilah **nilai edit_distance** dari pasangan kata berikut ini:
    a. phasmophobia - puafnilhotik
    b. genetik - ganteng

5. **[7] Bandingkan** hasil yang kamu temui pada nomor A2 dan A4. Apa yang dapat kamu **simpulkan** terkait kedua metode *edit distance* ini? Selain itu, apakah kedua algoritma *edit distance* ini sudah **sempurna**? Jelaskan analisis singkat kamu minimal dalam 3 kalimat.

B - N-Gram Language Model (30 Poin)

Perhatikan empat kalimat berikut

K1 : orang itu menjadi kepala rumah sakit di desa
K2 : kepala orang itu sedang sakit di rumah sakit
K3 : kepala rumah sakit itu memakan kelapa bersama warga desa
K4 : orang sakit itu memakan kelapa bersama kepala rumah sakit

1. **[12]** Jika diberikan ***threshold*** sebesar 10^{-12} dengan menggunakan *language model* **bigram** dan **trigram**, manakah di antara kalimat berikut ini yang memiliki kecenderungan pada ***spelling error***? Tunjukkan peluang dari setiap kalimat berikut dan tentukan kalimat yang mengandung *spelling error*!
    a. warga desa sedang sakit kelapa di rumah sakit
    b. orang itu sedang berada di rumah sakit
    c. rumah sakit di desa itu tempat warga memakan kepala
    d. orang sakit sedang memakan kelapa bersama warga desa

*Catatan B1

Terapkan *1-smoothing* jika diperlukan. Kamu dibolehkan mengerjakan dengan perhitungan manual tanpa implementasi kode.

2. **[8]** Dalam kasus umum (tidak terbatas pada soal ini), menurut kamu, manakah yang **lebih baik**; *language model* unigram, bigram, atau trigram? Berikan alasan pendukung jawaban kamu minimal dalam 3 kalimat.

3. **[10]** Menurut kamu apakah N-gram *language model* yang standar **sudah baik** untuk diterapkan pada bahasa yang kaya secara morfologis, seperti Bahasa Indonesia? Apa saja **masalah** berkaitan dengan morfologis yang mungkin ditemukan pada kasus ini? Ceritakan **strategi** kamu dalam mengatasinya dikaitkan dengan pengetahuan yang telah didapatkan pada bagian *Text Processing* minimal dalam 3 kalimat.

C - Pemetaan Dokumen dan Teks (30 Poin)

Perhatikan tiga buah dokumen berikut ini!

Dokumen A

> Sivitas Akademika Universitas Indonesia menjalankan aktivitas perkuliahan secara daring yang diikuti oleh mahasiswa dari seluruh jenjang. Hal ini diakibatkan karena adanya pandemi COVID-19 di Indonesia selama satu setengah tahun belakang.

Dokumen B

> Masyarakat memiliki respon berbeda dalam menyikapi pandemi COVID-19. Salah satu mahasiswa bernama Dipsy mengatakan kesulitan untuk belajar dengan maksimal karena sulit untuk berdiskusi dengan orang yang lebih pandai.

Dokumen C

> Salah satu kunci untuk menghadapi musibah adalah dengan bersabar dan yakin semua musibah akan berlalu pada waktunya. Badai pasti berlalu, jadi harus tetap semangat.

1. **[10]** Buatlah **pemetaan** seluruh kata yang sudah dalam bentuk kata dasar *lowercase* dan **bukan termasuk dalam *stopwords*** dengan jumlah kemunculan kata tersebut di setiap dokumen. Kamu dibebaskan bagaimana cara menampilkannya, (tidak harus dalam bentuk tabel, bisa saja dalam bentuk *key-value pair*).

   Stopwords dapat mengacu pada: https://github.com/masdevid/ID-Stopwords

***Catatan C1**

Ilustrasi mengenai pemetaan dokumen dan teks dapat dilihat pada bagian Lampiran

2. **[10]** Jika menjalankan masing-masing *query* berikut, dokumen manakah yang akan **ditemukan**?
   a. akibat AND sulit
   b. waktu OR santai
   c. (masyarakat AND mahasiswa) OR sabar
   d. (libur OR respon) OR (waktu AND jenjang)
   e. (kunci OR musibah) AND NOT (aktivitas AND pandemi) OR maksimal

3. **[10]** Apakah metode pemetaan dokumen dan teks dapat **memberikan manfaat** dalam proses perancangan sistem IR apabila dilihat dari perspektif **keakuratan hasil** yang akan dihasilkan? Jelaskan analisis singkat kamu minimal dalam 3 kalimat.

Lampiran

Algoritma Levenshtein Distance

Algoritma Levenshtein Distance

Edit distance digunakan untuk mengukur jumlah minimal perubahan antara satu string ke string lain. Operasi pada edit distance meliputi menambahkan (insert), menghilangkan (delete), mengganti (replace), dan menukar (transposition).

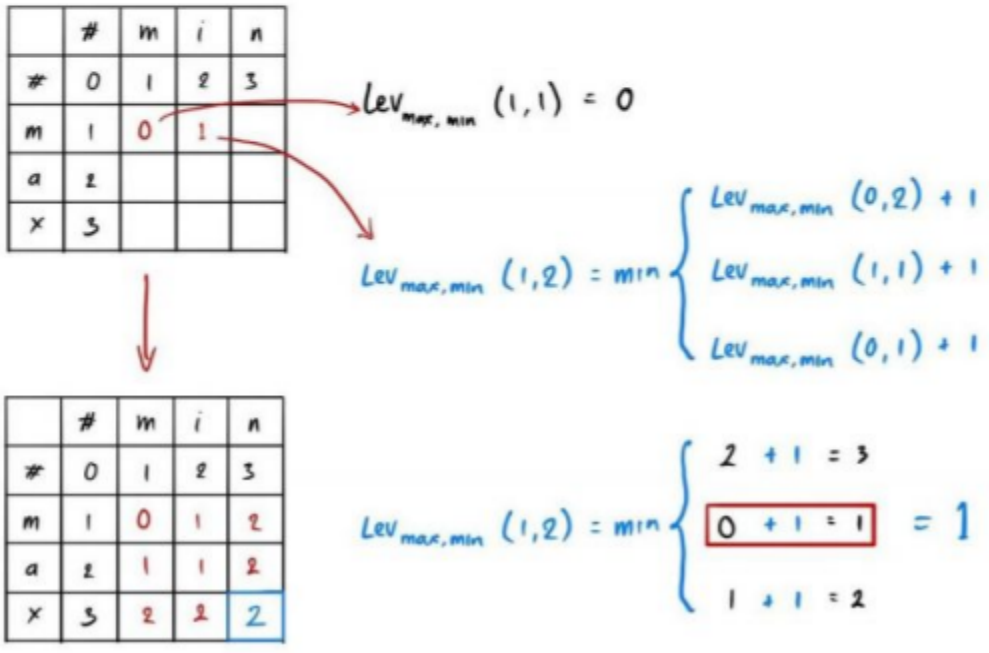
$$\text{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1,j) + 1 \\ \text{lev}_{a,b}(i,j-1) + 1 \\ \text{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases}$$

Contoh:

rani → rini (subsitusi a ke i) sehingga $\text{lev}_{\text{rani,rini}} = 1$

max → min (subsitusi a ke i dan subsitusi x ke n) sehingga $\text{lev}_{\text{max,min}} = 2$

Dalam melakukan implementasi persamaan diatas dapat menggunakan ilustrasi matriks sebagai berikut:



Contoh Pemetaan Dokumen dan Teks

Dokumen 1

Tiket kereta api
kelas ekonomi
sudah terjual habis.

Dokumen 2

Menjelang **lebaran**
pengunjung pusat
perbelanjaan sangat
meningkat
jumlahnya.

Kata	Dokumen1	Dokumen2
api	1	0
belanja	0	1
ekonomi	1	0
habis	1	0
jelang	1	0
jual	1	0
jumlah	0	1
kelas	1	0
kereta	1	0
kunjung	0	1
lebaran	0	1
pusat	0	1
tiket	1	0
tingkat	0	1