UNIVERSITAS INDONESIA
Veritas, Probitas, Iustitia

FAKULTAS ILMU KOMPUTER

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

Tugas 3

Tenggat Waktu: 10 November 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: Tugas1_1706979341_Lulu Ilmaknun Qurotaini.ipynb
4. Kumpulkan dokumen tersebut pada submisi yang telah disediakan di SCeLe sebelum tanggal **10 November 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.

Petunjuk Pengerjaan Tugas

Pada tugas ini, Anda diminta untuk mengolah korpus menggunakan korpus abstrak Skripsi Fasilkom UI yang telah diberikan: "**korpus_abstrak.csv**". Untuk tugas ini (selain bagian E), silakan fokus pada kolom **Abstrak** saja. Buatlah satu file Jupyter notebook dan kerjakan soal-soal berikut. Untuk soal yang membutuhkan penjelasan, Anda cukup menambahkan sel baru dengan format markdown/text.

A - Preprocessing (5 Poin)

Pada bagian ini, Anda diminta untuk melakukan *preprocessing* pada korpus Anda. Gunakan hasil *preprocessing* pada bagian-bagian setelah ini.

1. **[3]** Lakukan *lowercasing* dan tokenisasi. *Lowercasing* dilakukan dengan fungsi lower() dan tokenisasi dilakukan dengan *word_tokenize* dari NLTK. Tampilkan 10 abstrak teratas!
2. **[2]** Lakukan penghapusan semua karakter selain alfanumerik menggunakan fungsi isalnum(). Tampilkan 10 abstrak teratas!

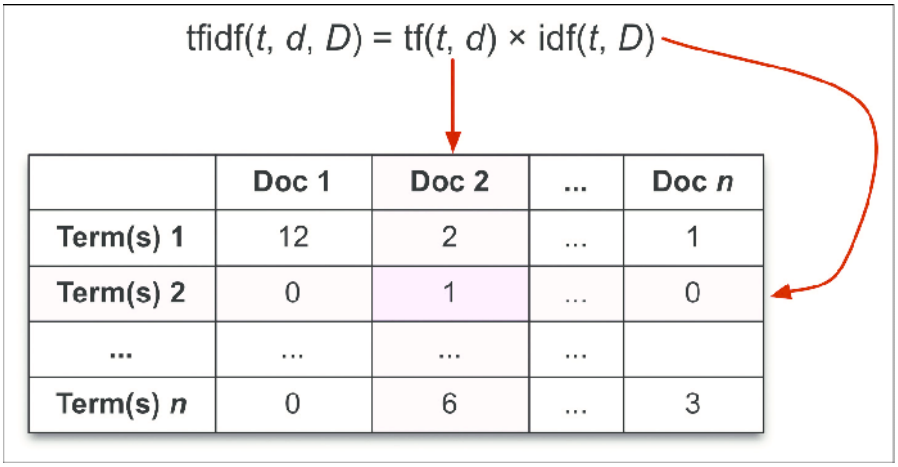
B - BM-25 (30 Poin)

Pada bagian ini, Anda akan dipandu untuk melakukan implementasi *retrieval* menggunakan BM25 secara langkah demi langkah pada korpus yang telah melalui tahapan *pre-processing*.

1. **[2]** Untuk setiap abstrak yang telah di-*preprocessing*, buatlah *dictionary* dengan *key* berupa token dan *value* berupa *term frequency*. Anda diperkenankan menggunakan *library* collections, namun Anda tidak wajib menggunakan *library* ini. Tampilkan *dictionary* untuk masing-masing dari 10 abstrak teratas.
2. **[6]** Buatlah daftar *vocabulary* (token yang unik) dari seluruh abstrak dan hitung *document frequency* serta *inverse document frequency* (idf) untuk setiap *vocabulary*. Untuk perhitungan *idf*, gunakan rumus berikut:

$$idf = log(1 + \frac{(N-df+0.5)}{df+0.5})$$

Anda dilarang menggunakan *library* yang dapat menghitung nilai *idf* secara langsung tetapi diperbolehkan menggunakan fungsi untuk membantu perhitungan seperti fungsi math.log (*natural logarithm*). Berikut adalah contoh ilustrasi tf-idf untuk beberapa dokumen dan term.



	Doc 1	Doc 2	...	Doc *n*
Term(s) 1	12	2	...	1
Term(s) 2	0	1	...	0
...	...	...	...	
Term(s) *n*	0	6	...	3

3. **[4]** Carilah *document length* (*dl*) dari setiap abstrak dan *average document length* (*adl*) dari keseluruhan abstrak. Tampilkan nilai *dl* dari 10 abstrak teratas.

4. **[12]** Buatlah fungsi *score*(*query, doc, k, b*) yang memberi *output* nilai skor relevansi antara *query* dengan masing-masing abstrak (*doc*). Skor relevansi tersebut dihitung dengan menggunakan rumus:

$$score(query,\ doc) = \sum_{t \in query} \left(\frac{tf_{t,doc}}{tf_{t,doc} + k[(1-b+b(\frac{dl}{adl})]} \times idf_t\right)$$

Pemecahan *query* menjadi beberapa *term* t harus dilakukan dengan *word_tokenize* dari NLTK.

5. **[6]** Gunakan fungsi *score* yang telah Anda buat untuk mencari 10 abstrak dengan relevansi tertinggi untuk *query*, nilai *k*, dan *b* berikut ini:

   a. query: *information retrieval*, k=1.2, b=0.75
   b. query: sistem manajemen pengetahuan, k=1.2, b=0.5

c. query: *knowledge discovery*, k=2, b=0.5

Tampilkan 10 *tokenized* abstrak dengan nilai relevansi tertinggi untuk masing-masing dari (a), (b), dan (c). Anda dilarang menggunakan *library* yang merupakan implementasi BM25 secara langsung.

C - Neural Embedding Word2Vec (30 Poin)

1. [**2**] Menggunakan data *preprocessing*, lakukan ***stemming*** dengan menggunakan ***library*** **Sastrawi** untuk masing-masing token pada setiap baris data.

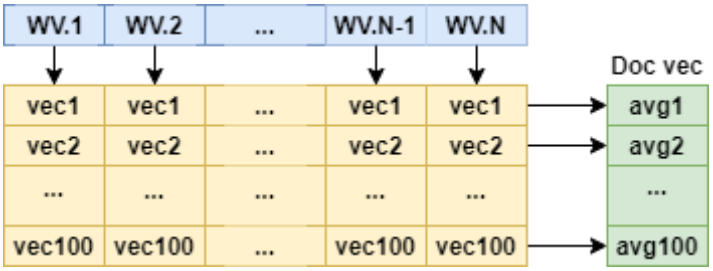
2. [**5**] Menggunakan data hasil *stemming*, buatlah sebuah model **Word2Vec** dengan modul yang disediakan oleh **gensim** (silahkan baca dokumentasi lebih lanjut) dengan ketentuan sebagai berikut:
   - ukuran dimensi Word2Vec adalah 2
   - dibebaskan untuk menggunakan model skip-gram ataupun CBOW
   - jumlah iterasi (epoch) sebesar 10.
   - untuk parameter lainnya dibebaskan kepada mahasiswa

3. [**2**] Gunakan model yang Anda buat untuk mencari representasi word2vec dari term berikut ini:
   a. algoritma
   b. interface

4. [**3**] Plot nilai vektor dari kedua term yang teman-teman dapatkan (soal nomor 3) dalam grafik 2 dimensi *euclidean space* (tidak perlu menggunakan *dimensionality reduction* karena vektor sudah berada pada 2 dimensi).

   DISCLAIMER: Hasil plot yang didapat mungkin saja tidak sesuai dengan ekspektasi teman - teman mengingat kecilnya nilai epoch yang digunakan dan korpus yang digunakan untuk kebutuhan tugas ini.

5. [**12**] Buatlah sebuah fungsi yang dapat mengembalikan dokumen relevan menggunakan Word2Vec berdasarkan **abstrak** jika diberikan sebuah query. Cara kerjanya dapat tapi tidak terbatas seperti berikut:
    a. Untuk menghitung vektor suatu dokumen atau kumpulan kata dapat dilakukan dengan menggabungkan vektor kata (*word embedding*). Penggabungan juga dapat bervariasi. Yang umum dilakukan adalah dengan menghitung rata-rata dari kumpulan vektor kata seperti visualisasi berikut:



    Catatan gambar: WV.N berarti vektor kata ke-N yang terdiri atas 100 dimensi vektor {vec1, ..., vec100} (dimensi disesuaikan dengan model yang kalian buat). {avg1, ..., avg100} merupakan vektor dokumen yang dihasilkan dari rerata vektor kata dengan dimensi bersesuaian.
    b. Mencari dokumen yang relevan dapat dilakukan dengan menghitung *similarity* (contoh: *cosine similarity*) antara representasi vektor query dan vektor dokumen. Seharusnya dokumen dengan *similarity* tinggi cenderung memiliki relevansi yang tinggi juga.

    Catatan: Anda bebas berkreasi untuk implementasi ini jika dirasa ada yang tidak sesuai, tidak terbatas pada mekanisme yang tertulis di atas.

6. [**6**] Gunakan fungsi yang telah Anda buat untuk mencari 10 abstrak dengan relevansi tertinggi untuk query:
    a. query: *information retrieval*
    b. query: sistem manajemen pengetahuan
    c. query: *knowledge discovery*

7. **[10] BONUS:** Anda dapat menggunakan pretrained model word2vec yang sudah dilatih menggunakan korpus wikipedia bahasa Indonesia. Setelah berhasil me-load model tersebut, Anda dapat mencoba menggunakan model tersebut untuk menjawab soal nomor 5 dan 6. Coba ceritakan perbedaan apa yang teman-teman dapatkan tanpa dan dengan pretrained model?

   Tips: Penggunaan pretrained model pada gensim tidak begitu kompleks. Kalian dapat menjalankan kode yang sama seperti di soal-soal sebelumnya, hanya saja dengan model model yang berbeda. Pretrained model dapat diperoleh pada link berikut ini:

   - word2vec_id ➜ pretrained model word2vec menggunakan korpus wikipedia Bahasa Indonesia. Terdapat 3 model pretrained word2vec dengan dimensionalitas embedding yang berbeda (100, 200, 300). Teman-teman cukup menggunakan model pretrained dengan dimensionalitas 100 (242 MB dalam format .zip).
   - word2vec_bahasa_indonesia ➜ Cara menggunakan pretrained model (credit to @deryrahman).

D - Dimensionality Reduction (10 Poin)

1. **[2]** Buatlah representasi matriks TF-IDF dengan mengalikan nilai *tf* pada setiap terms di setiap dokumen dan *idf* dari dokumen tersebut. Gunakan nilai *tf* dan *idf* yang Anda dapatkan pada nomor B2. Kemudian tampilkan hasilnya, berikut salah satu contohnya (tidak harus persis seperti ini):

```
Count Vectorizer

      blue  bright  sky  sun
Doc1     1       0    1    0
Doc2     0       1    0    1

TD-IDF Vectorizer

          blue    bright       sky       sun
Doc1  0.707107  0.000000  0.707107  0.000000
Doc2  0.000000  0.707107  0.000000  0.707107
```

2. **[3]** Berdasarkan hasil yang sudah didapatkan pada nomor 1, lakukan pemprosesan menggunakan PCA (2 komponen), kemudian tampilkan seluruh dokumen dalam bentuk visualisasi PCA di plot *euclidean space*.

   Tips: Bayangkan setiap kata pada korpus berada di posisi kolom, sedangkan nomor dokumen berada di posisi baris, kemudian semua kata yang ada pada korpus akan direduksi menjadi 2 *principal component*.

3. **[3]** Berdasarkan hasil yang sudah didapatkan pada nomor 1, lakukan pemprosesan menggunakan SVD (2 komponen). Anda dapat mencoba menggunakan **TruncatedSVD** pada library scikit-learn untuk menyelesaikannya.
4. **[2]** Bagaimana hasil yang Anda dapatkan setelah menggunakan PCA dan SVD ? Apakah ada hal unik yang bisa Anda temukan ? Jelaskan analisis singkat Anda minimal 3 kalimat.

E - IR Model Evaluation (25 Poin)

Pada bagian ini, Anda diminta untuk mengevaluasi hasil retrieval dari sistem yang telah Anda buat pada bagian B dan C dan memberikan analisis Anda berdasarkan hasil evaluasi yang Anda dapatkan. Gunakan hanya query "*information retrieval*" (query nomor **a** pada B5 dan C6) di bagian ini.

1. [**5**] Lakukan *human judgment* (proses manual dalam menilai apakah dokumen yang diperoleh merupakan dokumen yang relevan dengan query yang diberikan) pada 10 abstrak untuk masing-masing hasil retrieval menggunakan **BM25** (bagian B) dan **Word2Vec** (bagian C). Berikan nilai 1 jika Anda anggap abstrak tersebut relevan dengan query awal dan 0 jika tidak. Untuk tugas ini, tidak masalah jika Anda tidak mengetahui secara pasti apakah query benar-benar relevan dengan dokumen yang diperoleh. Yang penting adalah Anda konsisten dalam menilai kedua hasil retrieval. Tampilkan hasil *human judgement* yang telah Anda buat!

2. [**10**] Berdasarkan hasil dari nomor 1, hitung P@1 (precision-at-1), P@3, P@5, Mean Average Precision (MAP) dengan micro average, dan MAP dengan macro average dari hasil retrieval Anda untuk BM25 dan Word2Vec. Anda boleh melakukan perhitungan ini secara manual ataupun otomatis.

3. [**7**] Andaikan pada *ground truth* terdapat 5 item yang relevan dan berdasarkan hasil dari nomor 1, hitung R@1 (recall-at-1), R@3, R@5 dari hasil retrieval Anda untuk BM25 dan Word2Vec. Anda boleh melakukan ini secara manual maupun otomatis.

4. [**3**] Berdasarkan hasil dari nomor-nomor sebelumnya di bagian ini, lakukan analisis perbandingan mana yang lebih baik antara model BM25 atau Word2Vec. Sertakan penjelasan singkat dalam 3-5 kalimat terkait hasil yang didapatkan tersebut.