Laporan Tugas NLP Minggu ke-3 / TF-IDF & PPMI

Muhammad Asjad Adna Jihad 1301170242

- Deskripsi Masalah

Untuk topik artikel dengan topik 1 dipilih dengan topik 'Sneakers Sampah Nike Space Hippie' sedangkan untuk artikel dengan topik 2 dengan topik 'Kolaborasi Travis Scott dengan McDonald's' yang kemudian akan dirancang penghitungan nilai TF-IDF, pembentukan matriks TF-IDF, pembentukan matriks Co-Occurrence, Probability & PPMI , serta beberapa eksperimen.

- Perancangan Sistem

i. Pembacaan file & melakukan penggabungan menjadi satu file .txt

Untuk tahap ini, digunakan library glob untuk membaca 20 file artikel, dimana 10 artikel pertama dengan topik 1 digabungkan terlebih dahulu menjadi resultA.txt dan 10 artikel berikutnya digabung menjadi resultB.txt kemudian dua file tersebut digabung menghasilkan output file resultAll.txt yang didalamnya merupakan gabungan 20 file tersebut.

ii. Tokenisasi

Pada tahap tokenisasi, data dilakukan pembersihan (lowercasing, penghapusan tanda baca & space berlebih) kemudian membagi kalimat per-kata serta memunculkan frekuensinya disertai kata yang sering muncul.

- Tahapan Perancangan

Berikut tahap-tahap perancangan dalam mendapatkan nilai dan matriks yang dibutuhkan, diantaranya seperti berikut :

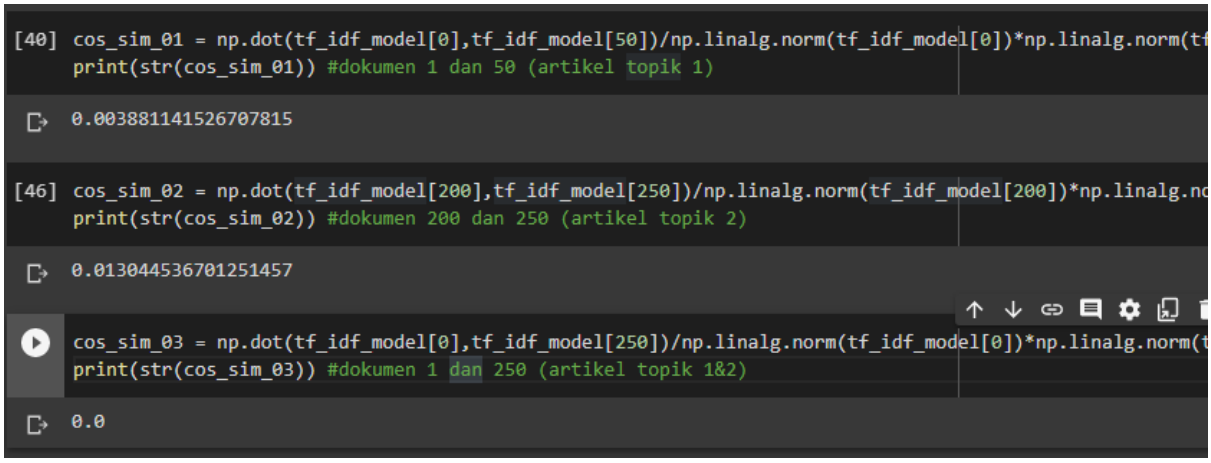
a. Menghitung Nilai TF-IDF
b. Membentuk Matriks TF-IDF
c. Membentuk Matriks Co-Occurrence (Output disajikan dalam file .txt (co-occurrence.txt) dikarenakan ukurannya yang cukup besar apabila ditampilkan langsung)
d. Membentuk Matriks Probabilitas (Output disajikan dalam file .txt (probability_matrix.txt) dikarenakan ukurannya yang cukup besar apabila ditampilkan langsung)
e. Membentuk Matriks PPMI (Output disajikan dalam file .txt (pmi_matrix.txt) dikarenakan ukurannya yang cukup besar apabila ditampilkan langsung)
f. Melakukan beberapa eksperimen diantaranya :
    i. Menghitung berapa persen elemen matriks Tf-IDF dan PPMI yang bernilai tidak sama dengan 0
    ii. Menghitung nilai cosine similarity antar dokumen dengan topik yang sama berdasar matriks TF-IDF, dan antar dokumen dengan topik yang berbeda.
    iii. Menghitung nilai cosine similarity antar kata berdasar matriks TF-IDF, dengan contoh pasangan kata yang berasal dari dokumen dengan topik yang sama dan contoh pasangan kata yang berasal dari dokumen dengan topik berbeda.
    iv. Melakukan perhitungan nilai cosine similarity antar kata berdasarkan Matriks Co-Occurrence
    v. Memeriksa nilai PPMI antar kata berdasar matriks PPMI

- Analisis

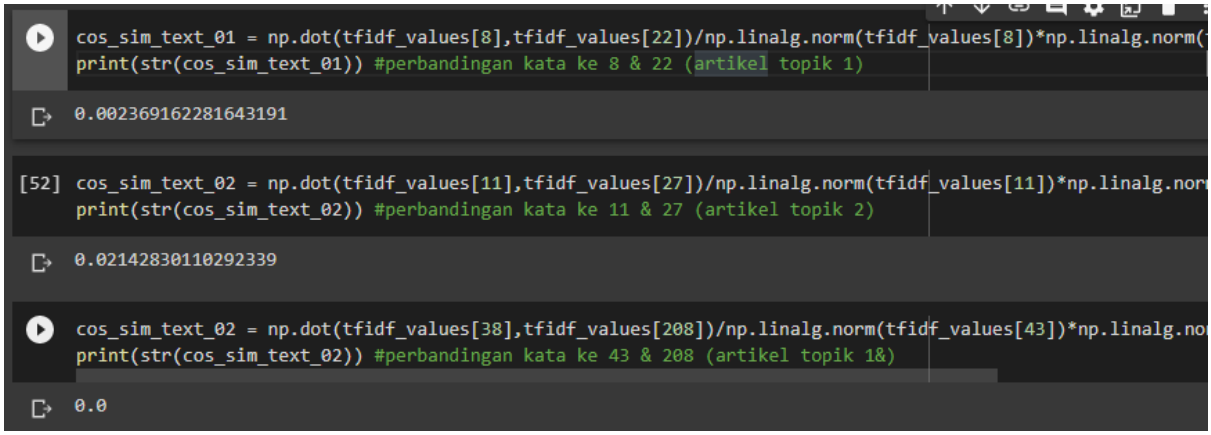
Setelah berhasil melakukan perancangan matriks TF-IDF, matriks Co-Occurrence, Matriks Probabilitas, dan Matriks PPMI dapat dilihat ukuran matriks dimana ukuran matriks TF-IDF adalah 295x500 sedangkan untuk 3 matriks terakhir dapat dilihat pada keluaran file .txt nya berdasarkan dengan nama yang sama.

Untuk eksperimen, pertama didapatkan persentase elemen matriks TF-IDF yang bernilai tidak sama dengan 0 adalah 2,87%. Kemudian untuk penghitungan nilai cosine similarity berdasar matriks TF-IDF dapat dilihat seperti berikut.

- Nilai per dokumen



- Nilai per kata



Untuk eksperimen selanjutnya mungkin terjadi kesalahan sehingga tidak mendapatkan hasil sesuai ekspektasi namun dapat dilihat seperti berikut.

4. Berdasar matriks co-occurrence term-context. Lakukan perhitungan nilai cosine similarity antar kata seperti poin nomor 3.

```
[84] def cos_similarity_word(a,b) :
       x,y,z = 0, 0, 0

       for woken in wordfreq.keys() :
         x += (co_occurrence_mat[(a, token)]*co_occurrence_mat[(b, token)])
         y += pow(co_occurrence_mat[(a,word)],2)
         z += pow(co_occurrence_mat[(b,word)],2)

       try:
         result = x / (math.sqrt(y) * math.sqrt(z))
       except ZeroDivisionError:
         result = 0

       return result

     cos_sim_text_01_ppmi = cos_similarity_word('hippie','sneakers')
     cos_sim_text_02_ppmi = cos_similarity_word('mcdonald','cactus')
     cos_sim_text_03_ppmi = cos_similarity_word('sampah','mcd')
```

5. Berdasar matriks PPMI, periksa nilai PPMI antar kata sesuai dengan yang digunakan pada eksperimen nomor 3 dan 4.

```
def ppmi_word(token_1, token_2):
  token_1_prob = bigram_count[token_1]/sum_term_context
  token_2_prob = bigram_count[token_2]/sum_term_context

  if term_context_prob[(token_1,token_2)] > 0:
    ppmi = max(round(math.log2(term_context_prob[(token_1,token_2)]/(token_1_prob*token_2_prob)), 4), 0)
  else: # kalau nilai probability = 0, tidak bisa dihitung log 2 -nya
    ppmi = None
  return ppmi

ppmi_01 = ppmi_word('hippie','sneakers')
ppmi_02 = ppmi_word('mcdonald','cactus')
ppmi_03 = ppmi_word('sampah','mcd')

print('PPMI antar kata (artikel 1) : ', str(ppmi_01))
print('PPMI antar kata (artikel 2) : ', str(ppmi_02))
print('PPMI antar kata (artikel 1&2) : ', str(ppmi_03))
```

```
PPMI antar kata (artikel 1) :  None
PPMI antar kata (artikel 2) :  None
PPMI antar kata (artikel 1&2) :  None
```

- Kesimpulan

Pada penghitungan frekuensi munculnya sebuah pasangan kata disarankan melakukan penghitungan berdasarkan matriks Co-Occurrence karena nilai cosine similarity yang dimiliki lebih besar jika dibandingkan dengan nilai yang berdasarkan matriks TF-IDF maupun berdasarkan matriks PPMI. Untuk penghitungan berdasarkan Matriks Co-Occurrence, tidak ada hasil cosine similarity yang bernilai 0, bahkan ada yang tidak dapat didefinisikan atau none.