



Nama : Muhammad Reza

Matkul : Temu Kembali Informasi

1. Tugas temu Kembali untuk evaluasi hasil kueri adalah :

Hasil kueri yang dipilih adalah “data”

- Process kueri

	Term	Dok	tf
0	data	D1	1
1	data	D2	1
2	data	D3	1
3	data	D4	0
4	data	D5	0
...
79	data	D80	0
80	data	D81	0
81	data	D82	0
82	data	D83	0
83	data	D84	0
84 rows x 4 columns			

Gambar df kueri 1

Gambar diatas menunjukkan bahwa terdapat 84 total baris, serta 4 kolom, maka dari itu dilakukannya evaluasi menggunakan *precision and recall*.Sebelum dilakukan evaluasi maka dilakukan :

```
not_relevant = df_kueri[df_kueri["tf"] == 0]
```

```
relevant = df_kueri[df_kueri["tf"] != 0]
```

```
not_relevant = not_relevant["tf"]
```

```
relevant = relevant["tf"]
```

Melakukan partisi antara document yang relevant dan tidak dengan eliminasi angka selain 0 yaitu *relevant*, lalu mengambil semua angka 0 yaitu *not_relevant*, karena ketidak seimbangan antara kedua array maka dilakukan :

```
dummy_zeroes_relevant = len(hasil_tf_) - len(relevant)
```

```
dummy_zeroes_notRelevant = len(hasil_tf_) - len(not_relevant)
```

```
dumy_number_relevant = np.zeros(dummy_zeroes_relevant, dtype = int)
```

```
dumy_number_notRelevant = np.zeros(dummy_zeroes_notRelevant, dtype = int)
```

```
relevant = np.insert(dumy_number_relevant, 0, relevant)
```

```
not_relevant = np.insert(dumy_number_notRelevant, 0, not_relevant)
```

Memberikan angka 0 agar hasil partisi yang tidak seimbang menjadi seimbang dan bisa dilakukan evaluasi

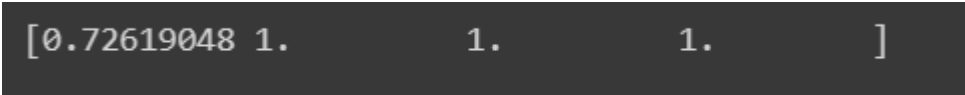
Evaluasi :

Evaluasi dengan menggunakan *precision & recall* sebagai berikut :

Recall

```
from sklearn.metrics import recall_score  
  
print(recall_score(y_true = not_relevant, y_pred = relevant, average =  
None, zero_division = 1))
```

Menggunakan library *scikit-learn* agar dapat kalkulasi dengan cara yang mudah, maka mendapat hasil pada gambar 2



```
[0.72619048 1. 1. 1. ]
```

gambar 2

Precision

```
from sklearn.metrics import precision_score  
  
print(precision_score(y_true = not_relevant, y_pred = relevant, average  
= None, zero_division = 1))
```

Mendapatkan hasil pada gambar 3:



```
[1. 0. 0. 0.]
```

gambar 3

Rocchio

$\alpha = 1$

$\beta = 0.9$

$\gamma = 0.4$

$query_optimal = \alpha * querys + \beta * relevant - \gamma * not_relevant$

[illegible]

gambar 4

Pada gambar 4 merupakan hasil menghitung optimal dari kueri yang telah dilakukan partisi

```
for v in relev["Dok"]:  
    optimal = np.dot(queri["kueri"],relev["tf"])  
    print(f" {v} Hasil rochioo optimal: {optimal} ")
```

Perhitungan rochio optimal dari masing-masing dokumen menghasilkan seperti pada gambar 5:

D1 Hasil rochio optimal: 1

D1	Hasil	rochioo	optimal: 1
D2	Hasil	rochioo	optimal: 1
D3	Hasil	rochioo	optimal: 1
D6	Hasil	rochioo	optimal: 1
D7	Hasil	rochioo	optimal: 1
D8	Hasil	rochioo	optimal: 1
D10	Hasil	rochioo	optimal: 1
D11	Hasil	rochioo	optimal: 1
D12	Hasil	rochioo	optimal: 1
D14	Hasil	rochioo	optimal: 1
D15	Hasil	rochioo	optimal: 1
D16	Hasil	rochioo	optimal: 1
D17	Hasil	rochioo	optimal: 1
D18	Hasil	rochioo	optimal: 1
D19	Hasil	rochioo	optimal: 1
D20	Hasil	rochioo	optimal: 1
D26	Hasil	rochioo	optimal: 1
D30	Hasil	rochioo	optimal: 1
D34	Hasil	rochioo	optimal: 1
D46	Hasil	rochioo	optimal: 1
D69	Hasil	rochioo	optimal: 1
D73	Hasil	rochioo	optimal: 1
D77	Hasil	rochioo	optimal: 1

gambar 5

Perhitungan rochio non-optimal dari masing-masing dokumen menghasilkan seperti pada gambar 6 dan kode sebagai berikut:

```
for v in relev["Dok"]:  
    optimal = np.dot(kueri_optimal["Kueri_optimal"],relev["tf"])  
    print(f" {v} Hasil rochioo optimal: {optimal} ")
```

```
D1 Hasil rochioo optimal: 64.0  
D2 Hasil rochioo optimal: 64.0  
D3 Hasil rochioo optimal: 64.0  
D6 Hasil rochioo optimal: 64.0  
D7 Hasil rochioo optimal: 64.0  
D8 Hasil rochioo optimal: 64.0  
D10 Hasil rochioo optimal: 64.0  
D11 Hasil rochioo optimal: 64.0  
D12 Hasil rochioo optimal: 64.0  
D14 Hasil rochioo optimal: 64.0  
D15 Hasil rochioo optimal: 64.0  
D16 Hasil rochioo optimal: 64.0  
D17 Hasil rochioo optimal: 64.0  
D18 Hasil rochioo optimal: 64.0  
D19 Hasil rochioo optimal: 64.0  
D20 Hasil rochioo optimal: 64.0  
D26 Hasil rochioo optimal: 64.0  
D30 Hasil rochioo optimal: 64.0  
D34 Hasil rochioo optimal: 64.0  
D46 Hasil rochioo optimal: 64.0  
D69 Hasil rochioo optimal: 64.0  
D73 Hasil rochioo optimal: 64.0  
D77 Hasil rochioo optimal: 64.0
```

gambar 6

Lampiran kode : [individu /Tugas individu sesi 13.ipynb at main · ackermanjayjay/individu \(github.com\)](https://github.com/ackermanjayjay/individu/blob/main/individu_Tugas_individu_sesi_13.ipynb)