Alvaro Austin - 2106752180

Laporan TPK 4

1. Format Dokumen: Saya menggunakan kedua title dan body text sebagai text yang akan di proses.
2. Pemrosesan Data:
   Pertama, teks dari dokumen (gabungan judul dan isi) maupun query diproses dengan *contractions* (misalnya, "can't" menjadi "cannot"), normalisasi huruf kecil, serta penghapusan karakter non-alfabet. Setelah itu, tokenisasi dilakukan untuk memisahkan teks menjadi kata-kata. Stopwords dihapus, kecuali kata-kata tanya penting seperti "what" atau "why", guna mempertahankan konteks pertanyaan. Saya tidak melakukan stemming karena pada saat dicoba hasil evaluasi re-ranker menjadi lebih buruk. Untuk dokumen, saya menggabungkan judul dan body textnya.

3. Pembuatan BiEncoder (mengikuti template):
```
def _biencoder_apply(dataframe):
  query_embs = bimodel.encode(dataframe['query'].values)
  doc_embs = bimodel.encode(dataframe['processed_text'].values)
  scores = cos_sim(query_embs, doc_embs)
  return scores[0]
```
4. Pembuatan Cross Encoder
```
def _crossencoder_apply(dataframe):
  return crossmodel.predict(list(zip(dataframe['query'].values, dataframe['processed_text'].values)))
```

5. Intepretasi Skor Evaluasi

	name	map	P@10	nDCG@10	map +	map -	map p-value	P@10 +	P@10 -	P@10 p-value	nDCG@10 +	nDCG@10 -	nDCG@10 p-value
0	BM25	0.364973	0.202857	0.449196	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	BM25 >> Bi-encoder	0.205385	0.157143	0.260837	7.0	26.0	0.000401	7.0	14.0	0.030335	5.0	26.0	0.000151
2	BM25 >> Cross-encoder	0.318294	0.214286	0.420224	13.0	18.0	0.171420	10.0	6.0	0.401599	13.0	15.0	0.379009
3	Bi-encoder >> Cross-encoder	0.279152	0.205714	0.373175	14.0	19.0	0.065873	11.0	9.0	0.855797	15.0	16.0	0.121551
4	BM25 >> MonoT5	0.344227	0.208571	0.440983	15.0	16.0	0.495198	9.0	6.0	0.720643	16.0	11.0	0.799767

   Dari hasil eksperimen reranking yang dilakukan, BM25 sebagai baseline menunjukkan performa yang cukup baik dengan:
   a. MAP = 0.364973
   b. P@10 = 0.202857
   c. nDCG@10 = 0.449196

Ketika dilakukan reranking menggunakan bi-encoder terhadap hasil BM25, terjadi penurunan performa yang signifikan (p-value = 0.000401:

a. MAP = 0.205385
b. P@10 = 0.157143
c. nDCG@10 = 0.260837

Nilai p-value yang jauh di bawah 0.05 ini mengindikasikan bahwa penurunan performa tersebut bukan kebetulan, melainkan memang bi-encoder secara konsisten memberikan hasil yang lebih buruk.

Saat menggunakan cross-encoder untuk reranking hasil BM25, penurunan performa tidak se-signifikan bi-encoder (p-value = 0.171420 untuk MAP, 0.401599 untuk P@10, dan 0.379009 untuk nDCG@10). Nilai p-value yang lebih besar dari 0.05 ini menunjukkan bahwa perbedaan performa antara BM25 dan BM25>>Cross-encoder tidak signifikan secara statistik. Ini berarti cross-encoder mampu mempertahankan kualitas ranking yang setara dengan BM25, bahkan dengan sedikit peningkatan pada P@10.

Pipeline yang menggunakan bi-encoder diikuti cross-encoder menunjukkan penurunan yang cukup signifikan untuk MAP (p-value = 0.065873, mendekati threshold 0.05), sangat signifikan untuk nDCG@10 (p-value = 0.121551), namun tidak signifikan untuk P@10 (p-value = 0.855797). Ini mengindikasikan bahwa meskipun pipeline ini dapat mempertahankan presisi di 10 dokumen teratas, secara keseluruhan kualitas rankingnya masih di bawah baseline BM25.

Pipeline bonus yang menggunakan bm25 diikuti oleh Mono T5 menunjukkan penurunan performa yang mirip dengan pipeline kedua (bm25 >> cross-encoder), terlihat bahwa pipeline ini juga memiliki performa yang lebih baik pada map dan juga nDCG@10, namun memiliki performa yang sedikit lebih buruk pada presisi namun tetap lebih baik dari baseline biasa. Nilai p-value yang **jauh** lebih besar dari 0.05 ini menunjukkan bahwa perbedaan performa antara BM25 dan BM25>>Mono T5 tidak signifikan secara statistik. Ini berarti mono T5, sama dengan cross encoder, mampu mempertahankan kualitas ranking yang setara dengan BM25, bahkan dengan sedikit peningkatan pada P@10. Menurut saya, pipeline inilah yang **dapat digunakan** sebagai **re-ranker yang stabil** (bahkan ada kemungkinan untuk di tuning agar lebih baik).

6. Analisis (Efektivitas dan Efisiensi)

Dalam menganalisis efektivitas dan efisiensi eksperimen reranking ini, kita bisa melihat beberapa pola menarik. BM25 sebagai baseline menunjukkan performa yang cukup meyakinkan dengan MAP 0.364973, membuktikan bahwa pendekatan berbasis term matching masih reliable untuk kasus ini. Kekuatan BM25 terletak pada kemampuannya menangkap relevansi dokumen berdasarkan distribusi term dan panjang dokumen, sambil tetap menjaga efisiensi komputasi yang tinggi.

Ketika kita mengimplementasikan bi-encoder sebagai reranker, terjadi penurunan efektivitas yang signifikan dengan MAP turun ke 0.205385. Hal ini mungkin disebabkan oleh beberapa faktor seperti ketidaksesuaian domain data training, representasi vektor yang terlalu general, atau

masalah semantic drift dimana makna yang ditangkap model terlalu jauh dari intensi query asli. Meskipun bi-encoder menawarkan efisiensi komputasi yang moderat karena hanya perlu melakukan encoding sekali, trade-off terhadap akurasi terlalu besar untuk diabaikan.

Pipeline dengan cross-encoder menunjukkan hasil yang lebih menjanjikan, mampu mempertahankan efektivitas dengan MAP 0.318294. Keunggulan cross-encoder terletak pada kemampuannya melakukan attention langsung antara query dan dokumen, memungkinkan penangkapan interaksi yang lebih kompleks dan scoring yang lebih presisi. Namun, hal ini datang dengan trade-off signifikan dalam hal computational cost, karena setiap pasangan query-dokumen harus diproses secara individual.

Yang menarik, MonoT5 mengungguli kedua reranker sebelumnya (bi-encoder dan cross-encoder). Dibandingkan dengan bi-encoder yang mengalami penurunan signifikan dan cross-encoder yang mempertahankan sekitar 87% performa baseline, MonoT5 mampu mempertahankan sekitar 94% performa baseline BM25. Keunggulan ini kemungkinan disebabkan oleh arsitektur T5 yang lebih sophisticated dalam memahami konteks dan relevansi, serta kemampuannya dalam sequence-to-sequence learning yang lebih baik untuk tugas reranking.

Akan tetapi, ketidakefektifan reranker dalam eksperimen ini bisa dijelaskan dari beberapa aspek. Pertama, kemungkinan adanya gap antara data training dan data aplikasi, termasuk potensi domain mismatch dan perbedaan distribusi relevansi. Tidak hanya itu, arsitektur bi-encoder mungkin terlalu simplistic untuk menangkap relevansi, sementara cross-encoder mungkin mengalami overtraining pada fitur tertentu.