Laporan Analisis QuestionAnswering System

1. Berikan analisis dugaan Anda mengapa tipe *question* ABBR kinerjanya paling rendah (sulit dideteksi), sementara tipe *question* LOC mempunyai kinerja paling tinggi?

Karena ABBR(abbreviation) berupa singkatan jadi sulit sekali memahami maksud dari singkatan tersebut sedangkan tipe question LOC(Location) mudah karena bersifat pasti seperti lokasikantor lebih mudah dicari saat googling dari pada mencari singaktan tertentu karena bisa menyebabakan ambigu sehingga bisa menyebabkan makna ganda

2. Apakah menurut Anda dapat diterapkan metode gabungan dengan pendefinisian aturan secara manual untuk mendeteksi satu atau beberapa tipe question? Sebutkan contoh aturan yang dapat diterapkan untuk mendeteksi tipe question (dalam bentuk pseudocode/algoritma).

Menurut saya dapat dilakukan dengan metode gabungan karena banyak sistem yang kinerjanya lebih tinggi menggunakan pendekatan gabungan/hybrid seperti DeepQA dan IBM Watson

If head_noun_class[0] == “ABBR” then
    If head_noun is an Abbreviation then
        Return (“ABBR”, ‘exp’)
    Return head_noun_class

3. Terapkan aturan tambahan yang didefinisikan secara manual tersebut untuk mengidentifikasi tipe *question* pada dataset uji yang sama, lalu amati hasil evaluasinya!

```
for i in range(len(questions)):
  questions= [i.split('\n', 1)[0] for i in questions]
  list_questions = list(next(zip(*map(str.split, questions))))
  if list_questions[i]=='What':
    labels[i] = "ENTY"
  elif list_questions[i]=='Who':
    labels[i] = "HUM"
  elif list_questions[i]=='Where':
    labels[i] = "LOC"
  elif list_questions[i]=='When':
    labels[i] = "NUM"
  elif list_questions[i]=='How':
    labels[i] = "NUM"
  else:
    # list_questions[i]=='Why':
    labels[i] = "DESC"
```

```
[47] print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
DESC	1.00	0.05	0.10	19
ENTY	0.84	1.00	0.92	125
HUM	1.00	0.92	0.96	24
LOC	1.00	0.67	0.80	9
NUM	0.91	0.91	0.91	23
accuracy			0.88	200
macro avg	0.95	0.71	0.74	200
weighted avg	0.89	0.88	0.84	200

Gambar1: Klasifikasi dengan Aturan tambahan

```
[25] print(classification_report(y_test, y_predict))
```

	precision	recall	f1-score	support
ABBR	0.00	0.00	0.00	4
DESC	0.71	0.54	0.62	37
ENTY	0.52	0.67	0.59	51
HUM	0.78	0.76	0.77	50
LOC	0.85	0.88	0.86	32
NUM	0.80	0.77	0.78	26
accuracy			0.70	200
macro avg	0.61	0.60	0.60	200
weighted avg	0.70	0.70	0.70	200

Gambar2: Klasifikasi tanpa aturan tambahan

Berdasarkan hasil evaluasi setelah ditambahkan aturan akurasi yang dihasilkan meningkat dari sebelumnya 70% menjadi 88% yang menandakan jika diberikan aturan mampu meningkatkan akurasi. Untuk precision berhasil mendapatkan macro avg 95% dan weighted avg 89% yang menggambarkan tingkat keakuratan antara data yang diminta dengan hasil prediksi yang diberikan, untuk recall berhasil mendapatkan macro avg 71% dan weighted avg 88% yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi, untuk f1-score berhasil mendapatkan macro avg 88% dan weighted avg 84% yang mengindikasikan bahwa model klasifikasi kita punya precision dan recall yang baik

4. Kira-kira fitur apa lagi yang dapat ditambahkan untuk meningkatkan kinerja identifikasi tipe question?
Lengkapi jawaban poin ini dengan referensi.
- Fitur BERT karena dapat menginkatkan kinerja sebuah dokumen dengan menyematkan question dan dokumen dengan cara yang lebih canggih (Kana, 2020)
- Dengan embebeddings kontekstual memungkinkan model menjawab pertanyaan berdasarkan Embeddings kontekstual BERT dan arsitektur tranformator untuk mencapai ketepatan yang lebih tinggi. (Jurafsky & Martin, 2020)