Richard Rafer Guy – 222117056

Tree of Thoughts (ToT) Prompting

I. Penjelasan Dataset yang Digunakan

Pada percobaan yang dilakukan, terdapat 2 dataset yang digunakan. Berikut adalah penjelasan mengenai 2 dataset (dari paper utama teknik ToT) tersebut.

A. GameOf24 : merupakan dataset untuk permainan untuk membentuk angka 24 dari 4 angka dengan menggunakan operasi matematika (*, /, +, -). Pada percobaan yang dilakukan, terdapat 15 data saja dari dataset tersebut untuk dieksekusi oleh program. Pada dataset tersebut telah terdapat pertanyaan (berupa 4 angka yang dapat digunakan). Dataset tersebut didapat dari scrape pada 4nums.com. Dataset ini disimpan dalam format CSV dengan field utama yaitu, Puzzles (berisi angka yang dapat digunakan), dan beberapa informasi tambahan seperti Solved Rate, Rank, dan lainnya.

B. CreativeWriting : merupakan dataset yang dihasilkan dari sampling kalimat random pada randomwordgenerator.com. Pada percobaan yang dilakukan, data yang digunakan hanya 15 data saja. Dataset tersebut disimpan pada file txt, setiap baris pada txt tersebut merupakan sebuah data / record, sehingga total terdapat 15 baris kalimat sebagai data pada file txt tersebut.

II. Penjelasan Library yang Digunakan

Berikut adalah daftar library yang digunakan dalam kode program, beserta penjelasan fungsi/kegunaannya masing-masing:

A. os : digunakan untuk berinteraksi dengan sistem operasi (misalnya mendapatkan value dari env variable).
B. google.generativeai : digunakan untuk mengatur API key, inisialisasi model, dan mengirim prompt serta menerima hasil dari model.
C. python-dotenv : merupakan library untuk membaca dan memuat isi dari file .env.
D. json : digunakan untuk membaca dan menyimpan data dalam format JSON.
E. argparse : digunakan untuk membaca argumen yang diberikan melalui command-line saat mengeksekusi program.
F. re : library ini digunakan untuk pemrosesan regex, untuk ekstraksi atau pembersihan teks.
G. time : library ini membantu memberikan delay proses, untuk menghindari rate limit API.
H. pandas : digunakan untuk membaca dataset dengan format CSV.

III. Penjelasan Kode Program & Cara Kerja

Bagian ini akan membahas program yang digunakan, cara kerja, hingga peran masing-masing komponen yang mereferensi ke dokumentasi kode yang telah dilakukan :

A. File **utils/gemini_client.py,** digunakan untuk menyiapkan konfigurasi gemini client dengan API Key yang terdapat pada .env, dan juga menyiapkan function untuk mengirim

promp dan mendapatkan hasil response dari LLM. Selain itu **utils/io_utils.py**, sebagai penyedia function untuk load & save data dalam format JSON.

B. File **main.py** berfungsi sebagai titik awal program. Program ini menerima 3 argument yaitu, --output (untuk memilih file penyimpanan hasil output pemrosesan), --dataset (untuk memilih dataset), --n-sample (dapat berguna untuk memilih jumlah sample yang dibuat pada proses ToT).

C. Function **main (main.py – Line 267-277),** merupakan function yang pertama kali dipanggil ketika main.py dieksekusi. Function ini akan mendapatkan argument pada command-line, dan memproses tugas tertentu berdasarkan dataset yang dipilih.

Cara Kerja Utama

Terdapat 2 function handler utama yaitu handle_game24 dan handle_writing. Berikut adalah penjelasan pada masing-masing function tersebut :

Function handle_game24 (Line 39-139): Berikut adalah cara kerja utama pada function handler untuk puzzle Game of 24

1. Mendapatkan dataset menggunakan bantuan os dan pandas.
2. Menyiapkan variable penyimpanan (seperti current_input_list, new_target_input, generated_samples, dan output log info).
3. Setelah itu akan membaca & memproses data puzzle dari dataset satu per satu. Berikut adalah proses yang dilakukan pada setiap data puzzle:
   - Mencatat initial log (berisi **input**, **index**, **steps** kosong, dan **final_score** = 0)
   - Memproses puzzle dengan metode ToT **(dijelaskan pada poin 4)**

4. **ToT untuk Game-of-24:**
   - **Template prompt** yang dipakai terdapat pada /prompts/gameof24_template.py
   - **Input** yang diberikan adalah daftar angka yang terdapat pada current_input_list
   - Akan dilakukan penelusuran sampai 3 kali iterasi (depth = 3). Penelusuran menggunakan metode BFS (Breadth First Search).
   - Pada setiap iterasi akan diawali dengan proses **generate sample.** Daftar angka yang terdapat pada current_input_list akan menghasilkan sample dengan menggabungkan 2 angka menjadi 1 menggunakan operasi *, /, +, -. Dari sebuah daftar angka akan dihasilkan sample sejumlah n sesuai dengan yang telah ditentukan pada argument command-line (default = 3). Pembuatan sample akan menggunakan bantuan LLM dan template prompt `generate_prompt`.

     Contoh : Jika `n = 5` dan `current_input_list = ["1 2 3 4"]` maka akan dihasilkan 5 sample, tetapi misalnya `current_input_list = ["2 3 4", "1 2 3"]`, maka akan menghasilkan 2x5=10 sample.

   - **Evaluasi** : Setiap hasil sample akan disimpan dan dilakukan evaluasi apakah sample tertentu memungkinkan untuk dilanjutnya, penilaian dilakukan dengan memberikan value "sure" += 10, "likely" += 1, dan "impossible" += 0.01. Evaluasi juga akan dilakukan dengan bantuan LLM dan template prompt `evaluate_prompt`.

- **Final**: Setelah itu sample yang dihasilkan akan diurutkan untuk mendapatkan Top3 sample terbaik dan dianggap paling memungkinkan untuk dilanjutkan.
- Hasil Top3 Sample tersebut akan menjadi input pada iterasi selanjutnya. Hasil sample pada setiap iterasi akan memiliki jumlah angka yang berkurang 1 dari iterasi sebelumnya. Sehingga pada iterasi ke 3 selesai, didapatkan top3 final answer yang terbaik untuk menyelesaiakn game of 24 dari 4 angka puzzle yang diberikan diawal.

5. Setiap hasil pemrosesan puzzle akan disimpan secara detail pada setiap stepnya. Selain itu evaluasi keberhasilan juga akan disimpan pada file yang telah ditentukan sebagai output file.

Function handle_writing (Line 144-266) : berikut adalah cara kerja utama pada function handler untuk creative wriing.

1. Membaca dataset dari file dengan format .txt
2. Menyediakan **function extract_plan_only**, untuk mendapatkan hasil plan untuk creative writing pada output LLM.
3. Menyediakan **function extract_passage_only**, untuk mendapatkan hasil passage untuk creative writing pada output LLM.
4. Menyediakan **split_with_dot,** untuk membuat format kalimat input dari dataset menjadi lebih rapih.
5. Setelah itu akan membaca dan memproses data dari dataset satu per satu. Berikut adalah proses yang akan dilakukan pada setiap data.
   - Mencatat initial log (berisi id, input yang diberikan dari record dataset tertentu, dan steps yang masih kosong).
   - Memproses pembuatan creative writing dengan ToT **(dijelaskan pada poin 5)**.

6. **ToT untuk Creative Writing:**
   - **Template prompt** yang dipakai terdapat pada /prompt/writing_template.py
   - **Input** yang diberikan adalah 4 kalimat yang dipisah dengan ".", dan tugas yang harus dilakukan adalah membuat 4 paragraf yang memiliki akhir kalimat dari masing masing kalimat yang diberikan.
   - Pada creative writing, akan digunakan teknik ToT dengan depth = 2, yang terpisah menjadi proses Generate Plan, dan Generate Passage. Penelusuran dilakukan dengan metode BFS (Breadth First Search)
   - **Generate Plan**, merupakan langkah awal untuk membuat plan cerita pada setiap paragraf dengan bantuan LLM dan template prompt `generate_sample_plan_prompt` pembuatan plan ini akan diulangi sebanyak jumlah n_sample, dan semuanya disimpan pada sebuah variable.
   - **Evaluate Plan**, langkah selanjutnya adalah untuk mengevaluasi plan yang telah dihasilkan, dengan melakukan voting. Voting akan dilakukan dengan meminta LLM dan bantuan template prompt `plan_vote_prompt` untuk memilih 1 plan terbaik. Voting akan dilakukan sebanyak jumlah plan yang dimiliki.
   - **Generate Passage,** plan terbaik akan digunakan untuk generate passage secara keseluruhan. Proses pembuatan passage akan diulang sebanyak n_sample, sehingga akan terdapat passage yang utuh sebanyak n_sample

- **Evaluate Passage,** beberapa hasil passage yang dihasilkan akan di evaluate dengan voting (dengan mekanisme sama seperti evaluate plan). Hasil passage terbaik akan menjadi jawaban akhir dari proses ToT pada creative writing tersebut.

7. Setiap hasil pemrosesan data akan disimpan secara detail setiap stepnya pada file output yang telah ditentukan diawal.

IV. Template Prompting

A. **/prompts/gameof24_template.py**:
   1. **generate_prompt**: Input berupa daftar angka, Output adalah menghasilkan sample dengan jumlah tertentu.
   2. **evaluate_prompt**: Input berupa daftar angka. Output akan menilai kemungkinan daftar angka membentuk angka 24 dengan pesan “sure”/”likely”/”impossible”.
   3. **final_prompt:** Input adalah daftar angka dari record dataset tertentu, dan jawaban yang dihasilkan, Output akan menilai ketepatan jawaban dengan value “sure”/”likely”/”impossible”.

B. **/prompts/writing_template.py**:
   1. **generate_sample_plan_prompt:** Input berupa data dari dataset, Output akan menghasilkan sebuah plan pada masing masing paragraf.
   2. **generate_sample_passage_prompt:** Input berupa plan dan Output berupa passage dari plan yang diberikan
   3. **plan_vote_prompt:** untuk voting 1 plan terbaik dari beberapa plan yang diberi
   4. **passage_vote_prompt:** untuk voting 1 passage terbaik dari beberapa passage yang diberikan.
   5. **score_template:** sebagai template yang dapat digunakan jika ingin melakukan penilaian terhaap hasil output dari creative writing.

V. Mekanisme Perhitungan Performansi

Berikut adalah penjelasan mengenai hasil perhitungan performansi dari teknik ToT pada 2 dataset dan task yang berbeda :

A. **Pada Dataset Game Of 24** : perhitungan score dilakukan dengan memberikan menghitung jumlah keberhasilan pemecahan masalah (membentuk angka 24) dibandingkan dengan total soal:
   - Berhasil / Passed : 10
   - Gagal / Failed : 5
   - Score : 66.66%

B. Pada Dataset Creative Writing : pada dataset ini saya tidak melakukan perhitungan score karena keterbatasan resource dan juga memerlukan expert dalam melakukan penilaian terhadap hasil yang diberikan. Pada paper utama, penilaian dilakukan dengan menggunakan model GPT berbayar untuk memberikan nilai 1-10 pada hasil output passage yang dihasilkan, dan meminta human expert dalam bidang tersebut untuk memberikan penilaian.