

Graf Pengetahuan dan Sistem RAG

Tugas Proyek II IF4070 Representasi Pengetahuan dan Penalaran



RAG—anggap saja itu adalah suatu mekanisme yang membuat model tidak sok tahu. Model tersebut layaknya seorang turis yang perlu membaca panduan atau peta setiap kali ingin melakukan sesuatu. Ini penting di era saat ini karena terkadang model cenderung menghafal sesuatu dalam konteks umum, bukan spesifik domain. Misalkan kita dihadapkan dengan pertanyaan, “Apakah kita bisa terbang?” Jika dikaitkan dengan kasus umum, tentu saja tidak. Namun, jawabannya berbeda bergantung pada domain. Pada domain bandara, jawabannya bergantung pada status maskapai pada waktu tertentu. Pada domain hukum, jawabannya merujuk pada undang-undang terkait penerbangan. Pada domain fiksi, jawabannya dapat berupa barang atau kondisi yang memungkinkan hal tersebut terjadi meskipun itu bertentangan dengan dunia nyata (misalnya gravitasi). Konteks diperlukan, dan di sinilah peran pengetahuan. Model memerlukan hal tersebut dengan representasi yang dapat dipahami sehingga dapat melakukan penalaran yang tepat.

Dalam tugas ini, hal yang perlu dikembangkan adalah menyusun graf pengetahuan, kemudian mengembangkan sistem RAG berdasarkan graf tersebut. Semoga berhasil!

(Versi: 1.0.1)

Spesifikasi Umum

- Tugas dikerjakan dengan kelompok yang sama dengan Tugas Proyek I.
- Tugas dikerjakan dengan domain yang sama dengan Tugas Proyek I.
- Setiap kelompok menyusun graf pengetahuan dan memanfaatkannya untuk sistem RAG yang menerima masukan pengguna.
- Tugas diselesaikan dengan mengumpulkan
 - berkas laporan berformat PDF,
 - berkas cadangan (*backup/dump*) Neo4j yang dapat diimpor, dan
 - kode sumber sistem RAG yang telah dimodifikasi¹.
- Pengumpulan dilakukan paling lambat pada 9 Desember 2025 (Selasa), pukul 16.59 WIB.

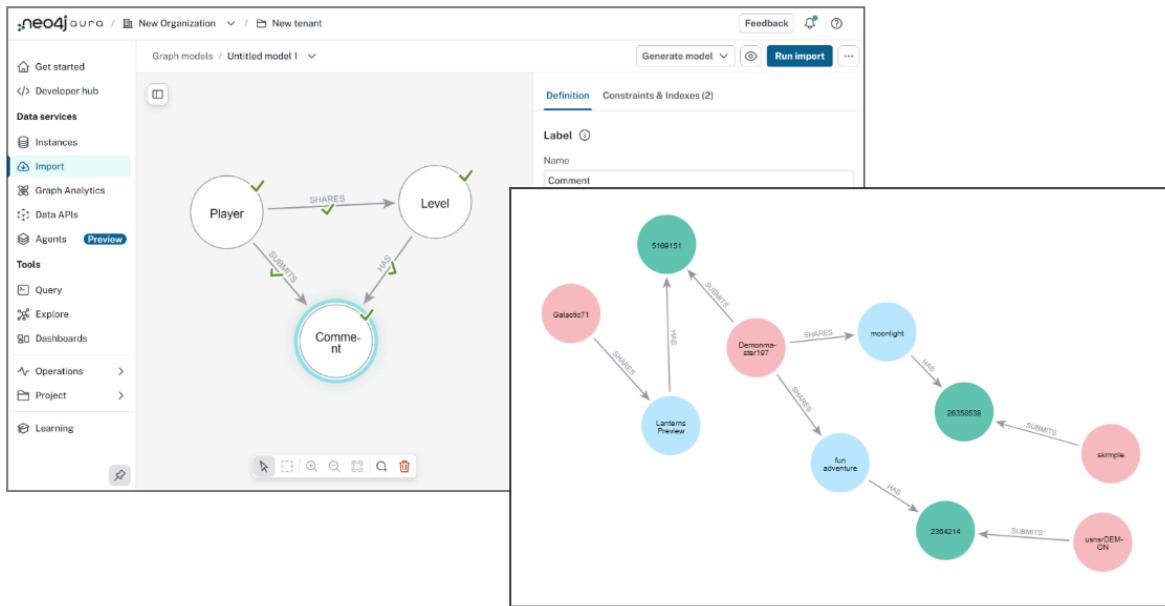
Kumpulkan hasil dari tugas di sini:

<https://forms.gle/5EoZVTSg4dAychuH9>

¹ Sebagai alternatif, tautan repositori GitHub dapat digunakan sebagai rujukan kode sumber. Dalam alternatif ini, rilis (*release*) perlu dilakukan (tutorial di [sini](#)), dan asisten hanya akan menilai hasil rilis tersebut. Agar setara dengan cara biasa, hasil rilis yang diterima adalah hasil yang dibuat sebelum batas waktu dan sebelum pengumpulan tugas.

Spesifikasi Rinci

Graf Pengetahuan



Graf pengetahuan adalah, ya, graf yang merepresentasikan pengetahuan. Pada tugas sebelumnya, ontologi disusun sehingga memungkinkan adanya asersi untuk instansi-instansi tertentu, yaitu dengan cara menetapkannya ke suatu konsep (*concept*) dan menghubungkannya dengan suatu peran (*role*). Protégé dan Prolog agak sulit untuk menjadi barang produksi, jadi teknologinya diganti dengan sesuatu yang lebih mutakhir: Neo4j!

Terdapat tiga hal utama yang perlu dilakukan untuk menyusun graf pengetahuan:

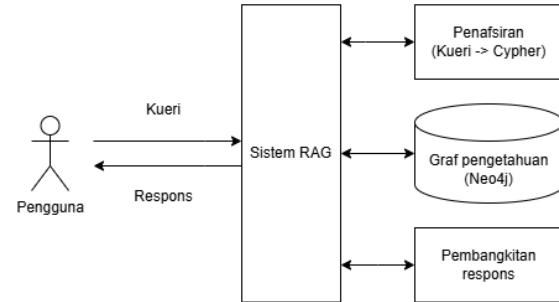
- 1) Tentukan skema atau model dengan menerjemahkan pengetahuan yang terlibat dengan simpul (*node*) dan sisi (*edge/arc*), termasuk atribut-atributnya.
- 2) Kumpulkan data-data yang relevan ke Neo4j berdasarkan skema tersebut.
- 3) Eksport cadangan (*dump/backup*) hasil implementasi.

Perhatikan jumlah data yang digunakan. Ukuran maksimum hasil tugas yang dapat dikumpulkan adalah 10 MB. Jika cadangan hasil eksport lebih dari 5 MB, pertimbangkan untuk melakukan pengurangan data². Tidak perlu lengkap. Intinya adalah peserta bisa menyusun graf pengetahuan.

² Batasnya mungkin lebih kecil dari itu jika ukuran laporan dan kode sumber lebih besar.

Sistem RAG

RAG adalah singkatan dari *retrieval-augmented generation*³. Ketika sistem memperoleh masukan dari pengguna, proses temu balik (*retrieval*) dilakukan dan hasilnya diaugmentasi bersama dengan masukan pengguna untuk melakukan pembangkitan (*generation*). Dengan kata lain, sistem selalu cari konteks dulu sebelum membuat jawaban. Ini dapat mengurangi kemungkinan halusinasi dari suatu model bahasa.



Pada tugas ini, kode sumber awal akan diberikan untuk membantu pengembangan sistem RAG⁴. Hal utama yang perlu dilakukan adalah sebagai berikut:

- Atur skema Neo4j yang akan digunakan⁵.
- Uji sistem RAG.
- Analisis hasil yang diberikan.
- Buat antarmuka yang dapat menerima masukan pengguna.⁶

Kode sumber awal dapat diakses di sini:

<https://github.com/AbdiHaryadi/graph-rag-starter-kit/>

Perhatikan bahwa kode sumber awal ini jauh dari kinerja sempurna. Skema yang diberikan mungkin kurang tergunaan, dan hasil yang diberikan mungkin tidak tepat dengan masukan pengguna⁷. Mungkin modelnya perlu diganti? Mungkin modelnya perlu dilatih sendiri? Mungkin kueri-kueri yang digunakan untuk Neo4j dibuat tetap saja dan sistemnya cukup memilih salah satu dari kueri tersebut? Mungkin perlu antarmuka yang lebih baik? Asisten serahkan hal itu kepada kelompok masing-masing.

³ Alternatif padanan dalam bahasa Indonesia: ‘pembangkitan beraugmentasi temu balik’

⁴ Jalur implementasi yang lebih umum adalah memanfaatkan pustaka LangChain dan OpenAI. Namun, tugas ini tidak mewajibkan kedua teknologi ini karena API dari OpenAI bukanlah hal yang gratis secara bawaan. Langkah ini menghindari penyelesaian tugas yang “pay-to-win”. Namun, jika ada yang ingin menggunakan jalur ini, silakan.

⁵ Set data Neo4j (<https://huggingface.co/datasets/neo4j/text2cypher-2025v1>) menjadi acuan untuk membuat skema yang dapat diterima sesuai dengan domain masing-masing.

⁶ Ini juga bisa dilakukan sebelum pengujian.

⁷ Peserta mungkin perlu paham membaca Cypher, bahasa kueri Neo4j, untuk menganalisis lebih lanjut terkait keandalan dari sistem RAG. Set data sebelumnya menunjukkan contoh kuerinya.

Laporan

Seperti pada tugas sebelumnya, laporan pada dasarnya adalah media untuk menyatakan hal yang telah dilakukan. Format dibebaskan. Laporan diharapkan dapat mencakup hal-hal berikut:

- Graf pengetahuan
 - Pemanfaatan secara garis besar
 - Deskripsi label dan simpul yang terlibat
 - Keterkaitan dengan ontologi
- Sistem RAG
 - Fungsionalitas penting dari sistem RAG yang dikembangkan
 - Pemanfaatan graf pengetahuan
- Pembahasan
 - Hasil dari sistem RAG (bukti bahwa sistemnya bekerja sesuai yang dirancang)
 - Perbandingan antara sistem RAG dan sistem berbasis pengetahuan
 - Keterbatasan graf pengetahuan dan RAG
- Kesimpulan
- Lampiran wajib: matriks kontribusi

Perhatikan bahwa ini bukanlah struktur yang harus diikuti secara tepat, jadi bisa diubah-ubah. Asisten akan menggunakan struktur ini sebagai acuan awal dalam menilai laporan. Intinya adalah laporan yang bagus adalah laporan yang menjelaskan segala proses yang terjadi dalam penggeraan tugas yang menjadi dasar pengembangan sistem RAG. Nilainya tetap bagus jika menggunakan format yang berbeda, jadi sesuaikan dengan kebutuhan.

Matriks kontribusi perlu ditunjukkan dalam laporan. Kuantitas kontribusi seharusnya tidak memengaruhi nilai jika perbedaannya tidak signifikan.⁸ Bisa saja orang yang mengerjakan paling sedikit justru mengalami kesulitan tertinggi. Tetap jujur, ya!

⁸ Sebagai kasus khusus, untuk peserta yang benar-benar merugikan kelompok, sampaikan ke asisten secara pribadi.

Penutup

Sama seperti tugas sebelumnya, kebebasan merupakan hal utama dari tugas ini, dan spesifikasi hanya menjadi panduan untuk memperoleh pengalaman yang baik. Kalau ada kesulitan, langsung sampaikan ke tanya jawab. Semangat, semangat!

Ini adalah tugas terakhir dalam mata kuliah ini untuk semester ini. Asisten berharap, peserta mendapatkan sesuatu yang menarik dalam setelah menyelesaikan dua tugas yang telah diberikan. Asisten sadar bahwa kedua tugas yang diberikan memiliki kekurangan, jadi silakan beritahukan saat mengumpulkan hasil tugas ini. Hal ini opsional, tetapi itu akan membantu asisten saat ini dan asisten selanjutnya.

Peringatan

Jangan terlambat, jangan plagiat, dan jangan gunakan kakas AI generatif sebagai proses utama. (Cek spesifikasi tugas sebelumnya untuk rincian.)