

Dokumentation QueryGenerator

Unser LLM-QueryGenerator ist ein auf dem Gemini 2.5 Flash Modell basierender Query Generator um auf Basis bereits vorhandener Suchabfragen, Folgeanfragen möglichst nah an einem realen Nutzer zu erzeugen.

Der User nimmt insgesamt drei Attribute entgegen.

- `user_type`: Steuert die Promptstruktur und legt fest, welcher Stil für die Generierung verwendet wird ("basic", "question", oder "boolean").
- `query_file`: Pfad zu der csv Datei mit alten Suchanfragen
- `user`: Für die Filterung der zu verwendenden Queries

Die Funktion „`generate_query_list`“ liest anhand von Nutzernamen und Thema passende Anfragen aus der Datei und gibt diese sortiert zurück. Daraufhin erzeugt die „`classify_query`“ Funktion zehn neue Folgeanfragen pro Prompt an das LLM.

Der Prompt hängt wie oben bereits erwähnt vom `user_type` ab und wird jeweils im Query Generator definiert. Technisch ändert sich hierbei direkt nichts, nur das LLM erhält andere Anweisungen. Zusätzlich erhält das LLM ein bis zwei Beispiele: „*Example: "Farmers Income Problems in philippines", "Pierre Bourdieu theory of State cultural capital symbolic capital analysis"*“

Ziele der verschiedenen Prompts sind wie folgt:

- "basic": Neutrale, stichwortartige Suchanfragen
- "question": Natürliche Fragen, wie von unerfahrenen Nutzern.
- "boolean": Boolische Anfragen, strukturiert, erfahrener Nutzer

Die Funktion „`get_next_query`“ stellt die generierten Queries der Reihe nach bereit. Alle generierten Strings werden durch „`sanitize_and_split_queries`“ bereinigt und in einfacher Textform zurückgegeben.

Die Einbindung erfolgt über eine XML-User-Konfiguration. Die Klasse `TripleLLMQueryGenerator` wird dort mit den oben genannten Parametern initialisiert. Der Rest des Nutzers basiert funktional auf dem `PredeterminedUser`, mit einem festen `StoppingDecisionMaker`, `Logger`, `UserContext` und einfachen Textklassifizierern.

Timo Busch (11160094), Patrick Knippenberg (11159646)