20875 – Software Engineering

Gooper - Search Engine for Influencers

MSc in Artificial Intelligence – Bocconi University

Giacomo Cirò

Fall 2024

Introduction

The rise of influencer marketing has revolutionized how brands engage with their target audiences. Yet, identifying the ideal influencer who aligns with a brand's values and goals remains a complex and time-consuming challenge. Existing tools offered by marketing agencies rely on basic filters such as tags, keywords, and location, which often lack precision and fail to meet the nuanced needs of modern marketing campaigns.

This project introduces *Gooper*, a search engine powered by a custom chatbot designed to deliver a more advanced, accurate, and reliable influencer search experience. Users interact with *Gooper* by means of natural language, describing their brand and marketing needs. The system then identifies and recommends the most suitable influencer from a database, explaining the reasons behind the choice while mimicking a natural conversation with the user. The core of this model is a Retrieval-Augmented Generation (RAG) pipeline, which leverages open-source Large Language Models (LLM) and a custom-built database to make suggestions based on user input.

The system is referred to as a *Search Engine* rather than a recommendation system because user interactions are entirely initiated and driven by user-entered queries. Suggestions are not based on predefined algorithms or user history, but on specific user-provided criteria. The goal is to address the influencer discovery challenge.

System Architecture

Gooper is presented as a fully integrated application, combining a web-based interface, a custom API, and a scalable vector database. In the following paragraphs, we explain the general workflow, as well as each component of the system in detail. Finally, we explore how the database was constructed.

Workflow A user interacts with the system by typing a brand description and marketing needs in plain English inside a chatbot-like interface. The user's input, as depicted in Figure 1, is processed in three sequential stages:

1. *Validation* → a discriminator model establishes whether the input is well-written and ready to be used for the search. If not, the user is prompted to try again.
2. *Retrieval* → based on the content of the input, the most relevant influencer is retrieved from the database.
3. *Generation* →based on the information retrieved, the system generates a recommendation for the user.

The user can then type another query and the process is repeated as in a natural one-to-one discussion.

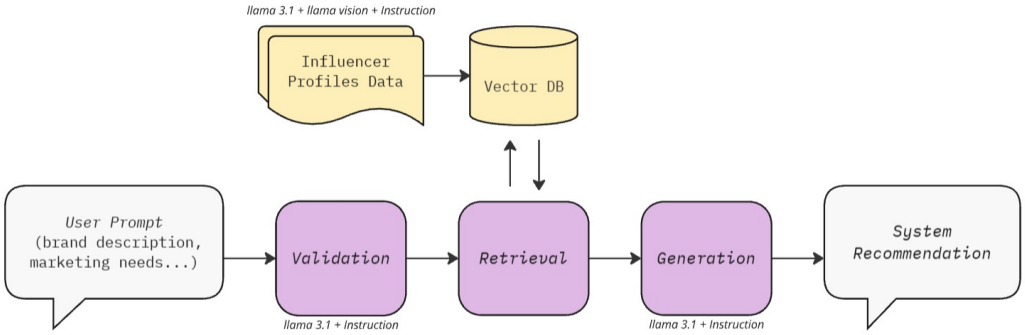

Figure 1: The *Gooper* workflow

Frontend The user interface is a custom website developed with *HTML*, *CSS* and *JavaScript*, hosted through *GitHub Pages*. It features a chat-box and some information about the system, such as model's version, the number of unique influencers available in the database and the name of the backbone LLM used, this information is retrieved by querying the *Gooper* API. The key front-end requirement was for the layout to be responsive and adapt to different screens, which is achieved using flex-boxes and *Bootstrap5*'s grid system.

Back-end The custom *Gooper* API is built using *Python* and *Flask* and hosted on *PythonAnywhere*. It handles communication between the front-end interface, the core model, and the database. It consists of a few endpoints, providing all the required functionalities. The main endpoint is `/generate`, which accepts user prompts via POST requests and returns the model response containing influencer recommendation and motivation. A detailed tutorial on the API usage and the other endpoints is made available at the API landing page through custom documentation.

Model The model is implemented in *Python*, leveraging *Together AI* for LLM inference and *Supabase* for vector database hosting. The core of the system is the `GooperModel` class, which operates through two main agents: a discriminator and a generator. The discriminator ensures that user inputs are valid for processing, while the generator creates the final response after having retrieved the relevant influencer from the database. Both agents use the same LLM backbone, but are guided by distinct sets of system instructions.

One of the key challenges of this project was to develop an efficient way to represent an influencer's profile, capturing information (such as target audience, content of posts, type of message, style etc.) from publicly available data, store it and retrieve the most appropriate profile upon request. To do so, the main abstraction on which the system operates is the profile's description, a concise paragraph summarizing an influencer's profile in natural language, directly extracted from the influencer's publicly available data. Each profile description is embedded into a vector space using a RAG-specific embedding model, and the embeddings are stored in a vector database. Details on how these descriptions are obtained are provided in the following section.

During the *Retrieval* stage, the system computes the cosine similarity between the user input and the stored embeddings, identifying the influencer whose description achieves the highest similarity score as the best match. In the final *Generation* stage, the model uses the selected profile's content to create a recommendation for the user, clearly explaining and justifying the choice.

Inference cost on LLMs varies significantly: different models were tried and the one eventually deployed strikes a good balance between costs and performance, in terms of response quality and latency.

Database The performance of similarity search is one of the main bottlenecks of the system, choosing an efficient vector database provider was crucial to make *Gooper* functional. The database for *Gooper* is hosted on *Supabase*, an open-source scalable platform for managing relational and vector data using *PostgreSQL*. It employs spatial data structures to enable fast similarity search in high-dimensional spaces. Besides performance, another key challenge was to design and implement the ETL pipeline to populate the database from publicly available information, in a RAG-friendly format.

For the scope of this project, I focus on the influencers' Instagram profiles. However, this pipeline can be easily generalized to more social networks or sources of knowledge, and is what I am planning to do with future improvements to the system.

The ETL pipeline is implemented in *Python* and works as follows. A web scraping tool is used to scrape data directly from Instagram and obtain a well-formatted JSON file containing profile's username, biography, recent posts images, recent posts captions etc. A vision-capable LLM is used to generate a text description of the Instagram post images. These are used together with the rest of the data as input for a text LLM instructed to generate a concise description of the overall profile. Finally, the descriptions are projected onto a vector space and stored in the database, ready to be used for RAG.

Conclusion & Future Work

This project represents the first public release of *Gooper*, and as such there are plenty of features and refinements I would love to add. For example:

1. I am planning to improve the front-end with nice widgets and influencer profile cards, user authentication features, a dashboard with user-related data and history of recent searches (specific data-collecting pipelines need to be implemented).
2. In order to make this tool actually useful to companies, I would like to scale the database to at least 100,000 entries and implement a feedback model to help the users modify the prompt when it is not valid.
3. Security measures should also be implemented, such as rate-limiting and more comprehensive input sanitization.

The system is designed to improve as it scales: more users interacting with the platform means more queries and validated responses, which can be used to fine-tune LLMs and align them more effectively with specific use cases, rather than relying solely on instructions.

In conclusion, *Gooper* represents a scalable and innovative solution to improve influencer discovery, leveraging LLMs and their open-world knowledge to meet modern marketing needs with precision and efficiency.