

HashtagHolidays Design Document

Table of Contents

- [Objective](#)
- [System Overview](#)
- [System Architecture](#)
- [Detailed Workflow](#)
- [API Endpoints](#)
- [Integration with DialogFlow](#)
- [Database Design](#)
- [Error Handling](#)
- [Future Enhancements](#)

Objective

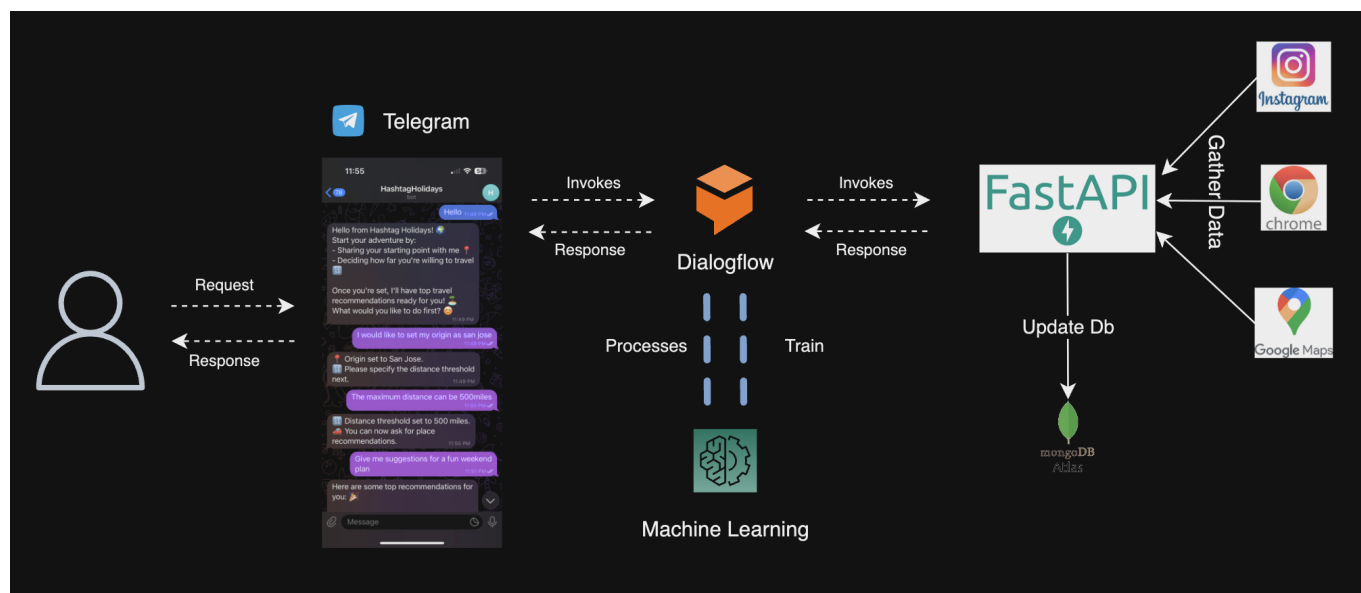
The objective of this system is to provide travel recommendations based on Instagram interactions, personal browsing history and google maps location history.

System Overview

The HashtagHolidays recommendation engine is designed to extract data from Instagram, process it, identify travel-related posts, and, based on the user's preferences and browsing history, provide relevant travel recommendations. The engine is built upon the FastAPI framework for backend services. It integrates with DialogFlow to offer chatbot capabilities, which are further extended to the Telegram platform for broader user accessibility. The system also leverages users' Google Maps location history to refine recommendations, ensuring they are both fresh and aligned with users' travel patterns.

System Architecture

High-level architecture diagram



The system can be broken down into a series of stages:

1. **Data Acquisition:** Extracting data from Instagram, including posts that are liked or saved.
2. **Data Processing:** Checking for existence in MongoDB, fetching post details, and classifying based on the content.
3. **Analysis:** Using Named Entity Recognition (NER) to identify locations and computing scores based on frequency and recency using a decay function.
4. **Incorporating Browsing History:** Enhances the scores based on personal browsing history if the place is recently searched.
5. **Recommendation Generation:** Calculates distances between potential destinations and the user's origin. If within a threshold, these are added to the recommendation list.
6. **API Endpoints:** FastAPI is used to expose the system's functionalities.
7. **Integration with DialogFlow:** For chatbot capabilities.

Detailed Workflow

Data Acquisition:

- Gathers data from Instagram, specifically saved and liked posts.
- Extracts URLs of these Instagram posts for further processing.

Data Processing:

- The system uses the `instaloader` library to extract post metadata, including captions and hashtags.
- After collecting the data, the system delves deeper into each URL, extracting post captions. These captions, rich in information, provide insights into the nature and context of the post. Preprocessing is then applied to remove any hashtags, ensuring a cleaner dataset for subsequent stages.

Classification & Entity Recognition:

- Utilizes a zero-shot classification model to determine if a post is travel-related.
- Employs FLAIR's NER model to extract entities such as locations from the post content.
- A zero-shot classification model is employed to discern the context of each post, particularly identifying if it's travel-related. Once identified, FLAIR's NER model is deployed to pinpoint specific entities within the post, like location names, ensuring the system knows exactly where the user's interests lie.

Scoring Mechanism:

The scoring mechanism is pivotal in generating relevant recommendations. It is divided into three main components:

1. Frequency and Recency Scoring:

- Attributes scores to locations based on frequency and timestamp using a decay function.
- By leveraging a decay function, HashtagHolidays determines the relevance of locations based on how frequently they appear and how recent the interactions are. This ensures that the system's suggestions remain fresh and in sync with the user's evolving interests.

2. Browsing History Adjustments:

- Integrates personal browsing history to fine-tune these scores, further personalizing recommendations.
- The system adjusts scores based on how often users research or show interest in certain destinations through their browsing history, ensuring a tailored experience.

3. Google Maps Location History Adjustments:

- Extracts places visited by the user from their Google Maps Location History.
- If a place has been visited before, its score is reduced by a certain percentage (e.g., 20%) to promote new recommendations.

Recommendation Engine:

- The backend of HashtagHolidays, powered by FastAPI, serves as the core of the recommendation system. This robust and responsive framework handles vast datasets, processes user queries, and fetches personalized travel recommendations.
- For more dynamic user interactions, the system integrates seamlessly with Dialogflow. This allows users to set various parameters, ask specific queries, and receive personalized feedback. The final recommendations can be sorted based on various criteria like score, distance, or time, giving users the freedom to plan their travels as per their convenience.

API Endpoints

- **/set-origin/**: Sets the user's origin.
- **/set-distance/**: Sets the user's preferred distance threshold.
- **/get-top-recommendations/**: Provides the top travel recommendations based on user preferences.
- **/places**: Retrieves all places from the database.
- **/places/scores**: Retrieves the scores for all places.
- **/dialogflow-webhook/**: Endpoint to handle requests from DialogFlow and provide appropriate responses.

Integration with DialogFlow and Telegram

HashtagHolidays integrates seamlessly with DialogFlow to handle chatbot interactions. This ensures dynamic interactions, enabling users to set parameters, make queries, and receive tailored feedback.

Furthermore, the chatbot is deployed on Telegram, a popular messaging platform. This allows users to interact with the recommendation engine in a conversational manner, directly from their Telegram accounts. The integration ensures a seamless experience, enabling users to fetch travel recommendations, set preferences, and interact with the system in real-time.

Database Design

The system uses MongoDB with two primary collections:

- **instagram_posts**: Stores information about Instagram posts, including URL, caption, hashtags, category, and timestamp.

- **places:** Stores recognized places along with their timestamps and associated post ID.

Error Handling

- The system handles various potential errors such as failed metadata fetches, bad responses from the `instaloader`, and JSON decode errors from the Google Maps API.
- For certain operations like fetching post details, random proxies are set to avoid potential rate limits or bans.