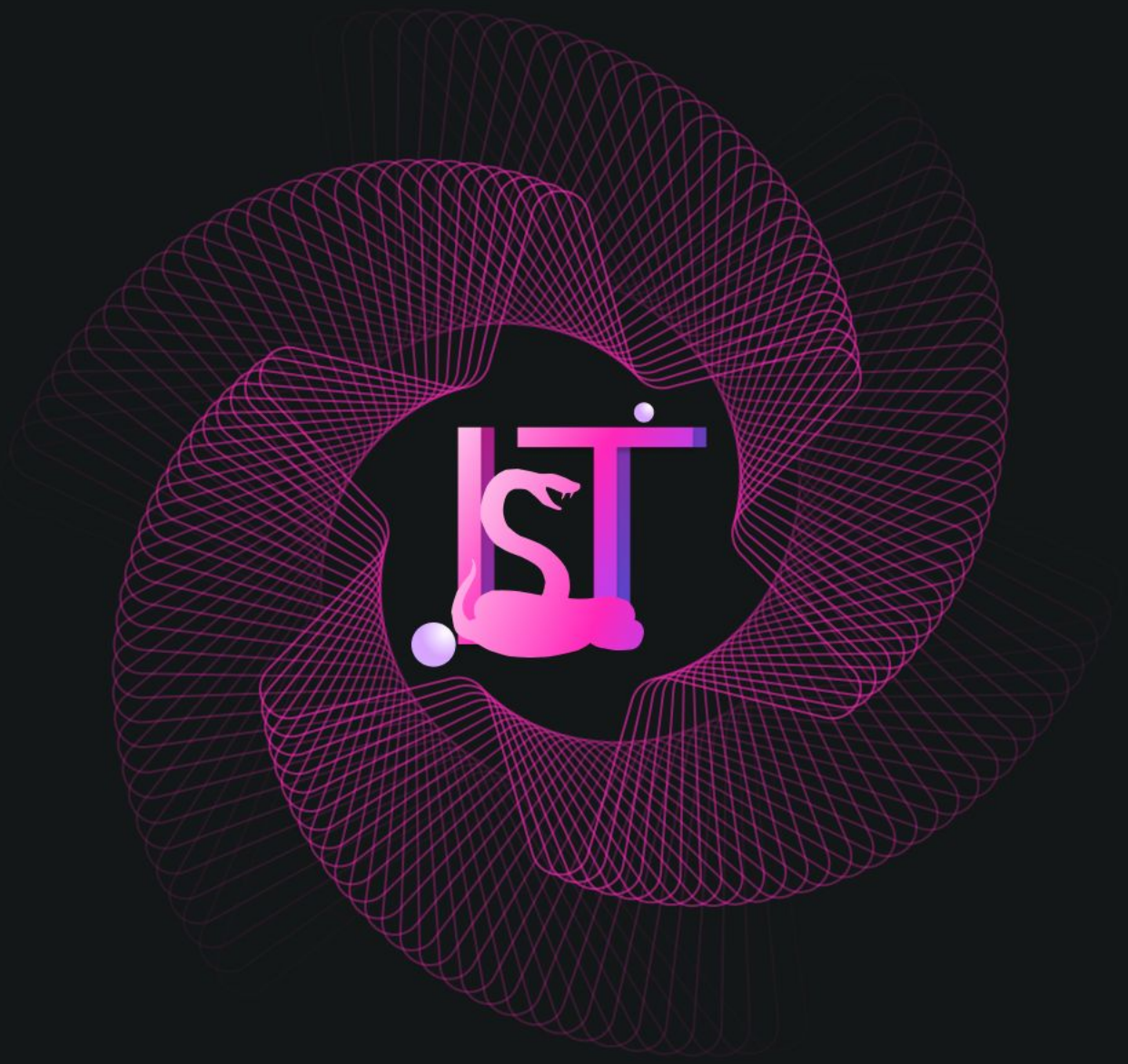


IT.py

SEO assist

KOKOC GROUP

Задача

Разработать сервис автоматического определения тематики веб-страниц из заданного списка

Технические трудности:

- Определение тематики недоступных страниц
- Парсинг страниц с защитой от ботов
- Сбор содержимого страниц, непроиндексированных поисковыми роботами
- Парсинг html-страниц большого размера
- Сложности в определении тематики страниц с неоднородным содержимым.

Решение

Чтобы сделать сервис максимально быстрым и точным, будем смотреть на тематику web-страницы не с точки зрения человека, а с точки зрения поисковой машины.

Описание страницы, если она **проиндексирована**, уже хранится и обновляется в БД поисковой машины. Если его нет, поисковая машина **сама пытается выделить ключевые фразы** для описания

1 Решение
Этап поиска инфо о странице

Для автоматического сбора информации о содержимом и метаинформации используем Google custom search API

Для работы API необходимо создать свою поисковую машину

Для получения результата отправляем строку с url web-страницы
в качестве текста запроса

2 Решение
Этап сбора информации

Как показывает практика, модель машинного обучения ChatGPT достаточно неприхотлива к чистоте текста при решении задачи определение его темы.

Текст для последующего запроса формируется из исходного url, описания, сформированного поисковой машиной, и title (если был найден)

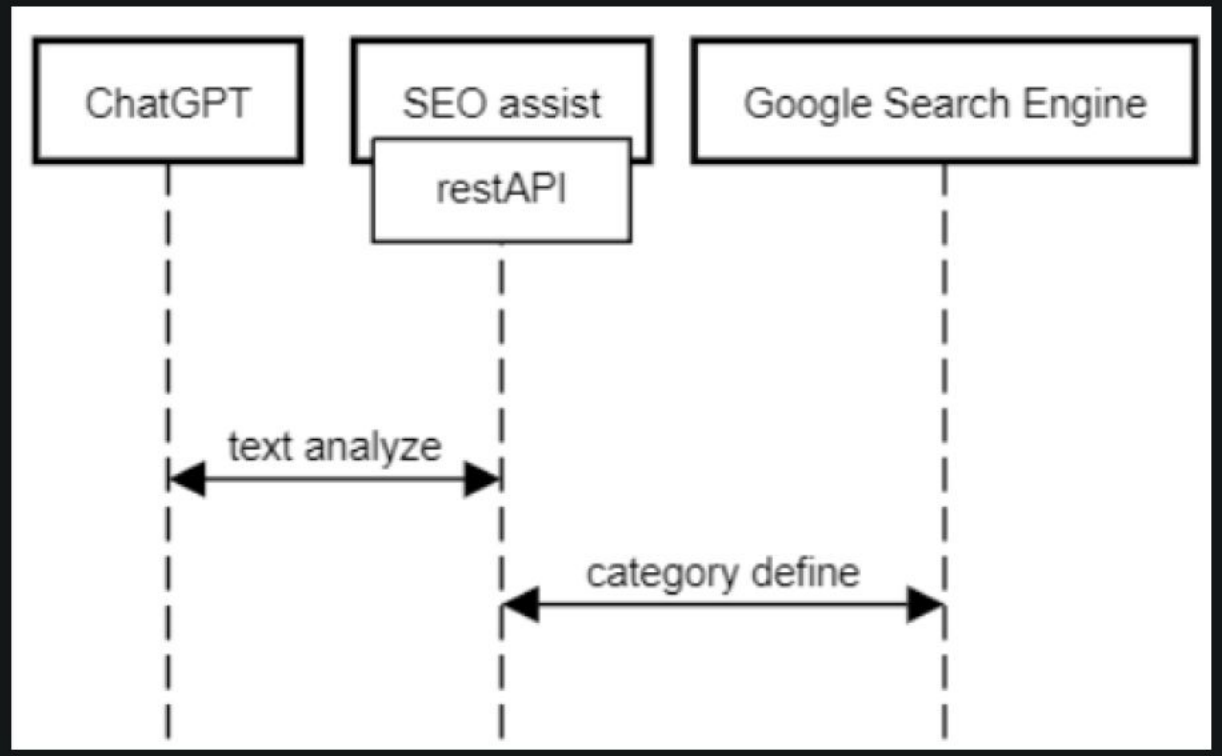
ChatGPT query text
=
url
+
search engine description
+
title

3 Решение

Этап получения тематики

Привет! Выбери одну наиболее релевантную тематику из перечисленных для следующего текста: {site_info} В ответе верни только название тематики строго как она записана в списке перечисленных в точности до символа.

Архитектура



- **/check_url** - получить тематику для запрашиваемого url

```
request_body: application/json = {
  url : string
}
response_body: application/json = {
  category: string,
  theme: string
}
```

- **/check_urls** - получить тематики для запрашиваемого списка url-ов

```
request_body: application/json = [string]
response_body: application/json = [{
  category: string,
  theme: string,
  url : string
}]
```

Стек: Python 3.11 + FastApi async

Преимущества решения

- Высокая производительность (rps) за счет асинхронности и отсутствия ресурсоемких вычислений (Всё делегируется сторонним API)
- “Лёгкость” сервиса: малый размер образа, малая ресурсоёмкость, высокая скорость CI/CD процессов,
  простота внесения изменений и поддержки
- Достаточная точность определения тематики
  для недоступных страниц, страниц с защитой от парсинга,
  страниц с объёмными статическими ресурсами
  (для анализа используются данные только от поискового движка)

Недостатки решения

- Использование платных сторонних сервисов
- Неустойчивость результата к ошибкам написания во входных данных по темам и категориям

Ссылки:

Репозиторий: https://git.codenrock.com/kokoc-2023/cnrprod-team-50824/theme-web-resources-kokoc

Сервис развернут: 51.250.69.175:5000/check_url

Демонстрация работы:
https://drive.google.com/file/d/1lBMZxafy97bc0AT554VFopFbHNUEwe7b/view