NLP and Web Applications Final - SearchPTT

電信所博三 許博竣 f07942095

GitHub Repository

https://github.com/BogiHsu/SearchPTT

Slide

https://github.com/BogiHsu/SearchPTT/blob/master/slide.pdf

Motivation

PTT, the most influential social platform in Taiwan for the past two decades, updates tens of thousands of posts every day. The extensive contents include almost everything from gossip, sports, and movies to career, finance, and military affairs. These posts not only serve as sources of numerous news but also bring up the discussion of different issues and affect the trend of the whole society. Although extensive and in-depth articles can be found in various fields, the electronic bulletin board system PTT uses only provides simple search functions, making it difficult for users to absorb and analyze new knowledge quickly. Motivated by the limitation, this project aims to build a search and analysis system that allows users to access the content of interest rapidly.

Impact

This project establishes a PTT search and analysis system with various features. According to the conditions given by the user, the system will search and list related posts. In addition, the system also presents NLP analysis results for either a single post or all search results. This project helps users access and analyze specific issues more efficiently and provides a good web application framework, including the frontend UI and the backend server. Developers can also present more different results on the web page by changing the NLP analysis algorithm or adding new functions on the backend server. Overall, this project can analyze and track PTT posts from different aspects and present the results, helping people understand current hot topics and trends more quickly.

Design

Features

1. Search

   Users can search for posts on PTT given the following conditions:

   (1) Board (required): Select a board from the top ten popular boards to search for posts on. The list of the top ten popular boards will refresh when loading the web page.

   (2) Keyword (optional): Search for posts with a specific word in the title.

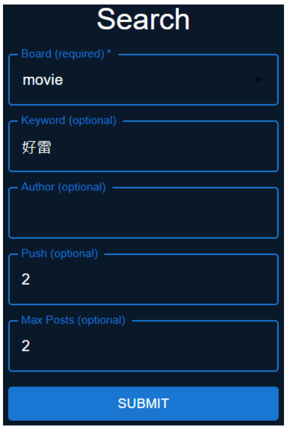
   (3) Author (optional): Search for posts from a specific author.

   (4) Push (optional): Search for posts with pushes greater than a specific number.

   (5) Max Posts (optional): Specify how many posts to list.

   The search results list different information of the posts, including (1) time, (2) title, (3) author, (4) summary, (5) keyword, (6) AID, and (7) URL. The AID is a unique tag for each post and can be used in 3. Podcast.

   Below is an example of this feature.



Search

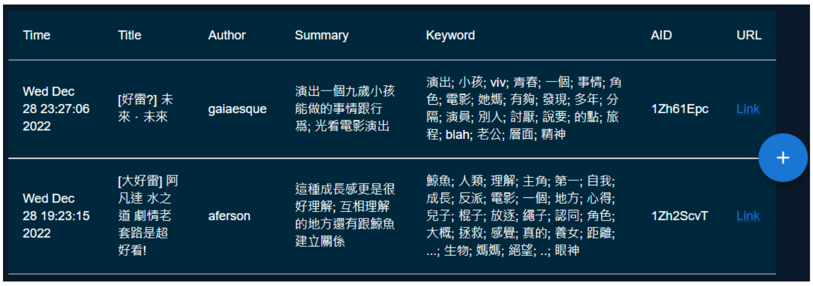
Board (required) *: movie

Keyword (optional): 好雷

Author (optional):

Push (optional): 2

Max Posts (optional): 2



Time	Title	Author	Summary	Keyword	AID	URL
Wed Dec 28 23:27:06 2022	[好雷?] 未來 · 未來	gaiaesque	演出一個九歲小孩能做的事情跟行爲; 光看電影演出	演出; 小孩; viv; 青春; 一個; 事情; 角色; 電影; 她媽; 有夠; 發現; 多年; 分隔; 演員; 別人; 討厭; 說要; 的點; 旅程; blah; 老公; 層面; 精神	1Zh61Epc	Link
Wed Dec 28 19:23:15 2022	[大好雷] 阿凡達 水之道 劇情老套路是超好看!	aferson	這種成長感更是很好理解; 互相理解的地方還有跟鯨魚建立關係	鯨魚; 人類; 理解; 主角; 第一; 自我; 成長; 反派; 電影; 一個; 地方; 心得; 兒子; 棍子; 放逐; 繩子; 認同; 角色; 大概; 拯救; 感覺; 真的; 養女; 距離; ...; 生物; 媽媽; 絕望; ..; 眼神	1Zh2ScvT	Link

2. Analyze

   The previous feature provides information of each post, while this one analyzes the trends of all search results.

   The search interface is the same as in 1. Search. Given the conditions, the system will analyze all the searched posts and show the trends. The results include:

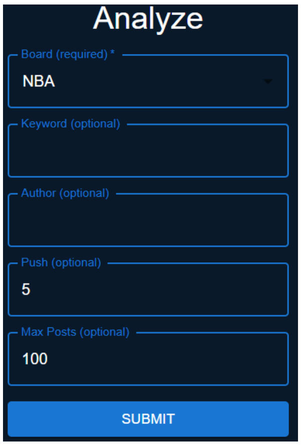
   (1) Keyword: Concatenate all post contents and analyze the keywords.

   (2) Sentiment analysis: Analyze the average sentiment scores of the post contents and the replies, respectively.

   (3) Word cloud of keywords: Collect all keywords in each post and generate a word cloud figure to show the most frequent ones.

   (4) Word cloud of posting locations: Collect the posting locations of each post and generate a word cloud figure to show the most frequent ones.

   Below is an example of this feature.



Analyze

Board (required) * NBA

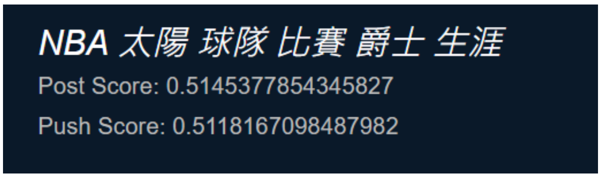
Keyword (optional)

Author (optional)

Push (optional) 5

Max Posts (optional) 100

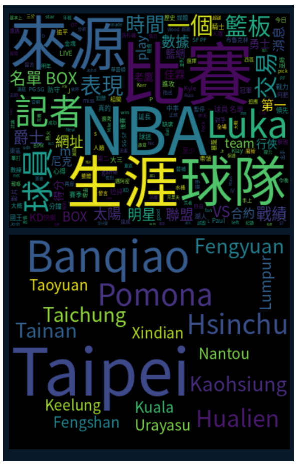
SUBMIT



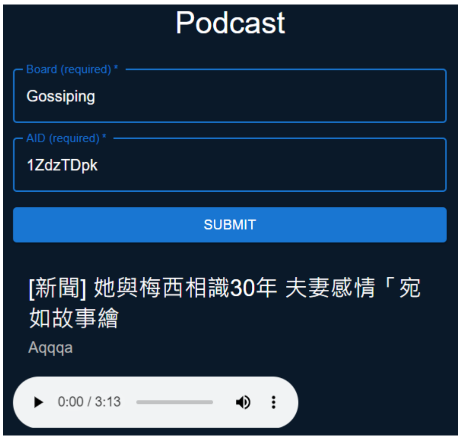
NBA 太陽 球隊 比賽 爵士 生涯

Post Score: 0.5145377854345827

Push Score: 0.5118167098487982



3. Podcast
   This feature converts a post into an audio clip. Given the board and the AID, users can "play" the selected post, like listening to a podcast.
   Below is an example of this feature.



Podcast

Board (required) * Gossiping

AID (required) * 1ZdzTDpk

SUBMIT

[新聞] 她與梅西相識30年 夫妻感情「宛如故事繪

Aqqqa

0:00 / 3:13

Architecture

1. Workflow
   The whole system consists of two parts, Frontend and Backend, which will be discussed in the following paragraphs. Overall, Frontend provides an interface for users to specify the conditions for post-searching, and the interface also visualizes the processed results. Backend, on the other hand, performs all the crawling, searching, analyzing, and speech generation algorithms. Frontend and Backend can communicate with each other through the local network or the Internet.
2. Frontend
   Modified from the assignment 2, Frontend is built with React. Users can access features and give conditions mentioned in *Features* through this interface. All the components, such as buttons, navigators, and input spaces, are based on the MUI library. Users' requests (features and conditions) are sent to Backend using Axios and the HTTP GET method.
   The code is in the directory SearchPTT/webpage/ of the GitHub repository.

3. Backend
   Backend is built with Python and the Flask library. The Flask library provides a simple way to construct APIs accessible with the HTTP GET method. Once an API receives the request, it will call the corresponding algorithm based on the specified feature and given conditions.
   The code is in the directory SearchPTT/server/ of the GitHub repository. Specifically, all the APIs are constructed in SearchPTT/server/src/main.py. Running this file will also start the Backend service.

Method

This section introduces the main methods in Backend to provide the services in *Feature*. All the methods are implemented in Python.

PTT Crawler

The top ten popular boards are crawled from https://www.ptt.cc/bbs/hotboards.html using the Requests library.
PyPtt is a library that provides an interface to interact with PTT. This project uses the functions from PyPtt to log in to and search posts on PTT.
The functions are implemented in SearchPTT/server/src/ptt.py.

IP Analysis

The IP of each post is sent to https://ipinfo.io/ through the HTTP GET method. The API service analyzes the IP address and returns the location information, including country, city, organization, and geographic coordinates.
The function is implemented in analyze_ip in SearchPTT/server/src/analyze.py.

NLP Analysis

Each post is first converted into simplified Chinese using the OpenCC library and analyzed using the SnowNLP library. The analysis results include summaries, keywords, and sentiment scores of the post. The final results are converted back to traditional Chinese.
The functions are implemented in snow_analysis and hot_keyword_and_cloud in SearchPTT/server/src/analyze.py.

Word Cloud

Given a collection of the keywords or cities of different posts, the word cloud figures are generated using the WordCloud library.
The functions are implemented in hot_keyword_and_cloud and city_chart in SearchPTT/server/src/analyze.py.

Text-to-Speech (TTS)

The TTS function is built using gTTS, a library to interface with Google Translate text-to-speech API. Given the text, one can generate speech with different accents in a single function. In this project, the speech is generated with the `zh-tw` tag, leading to a Taiwanese accent.
The function is implemented in `text2speech` in `SearchPTT/server/src/tts.py`.

Data and Ethics

All the data used in this project is crawled from PTT. The copyright of the contents might be reserved by the authors or the PTT official. All the contents and the analysis results are not allowed for commercial use. The user agreement in https://www.ptt.cc/index.ua.html provides more detailed information.