Let’s Beat Google!!!

主題:傳說鬼怪故事

110305075 企管四 繆孟珊

110308035 風管四 曾惠瑜

109405229 新聞五 王苡謙

112208077 經濟二 邱晨綠

Introduction

1. Motivation

Every Halloween, spooky topics gain immense popularity, and many intriguing ghost stories circulate on historical campuses. However, for timid individuals, these stories may cause discomfort. We aim to create a "NCCU Ghost Story Search Engine" specifically designed for the timid. Through a fun and light-hearted approach, users can safely explore ghost stories on the NCCU campus while enabling data collection and classification, allowing more people to gradually search for scary content.

2. User Needs and Pain Points

Target Users: Timid individuals who are curious but hesitant to explore.
According to some psychological studies, watching horror films helps individuals develop coping mechanisms for stress and anxiety, akin to exposure therapy. To make scary search results acceptable for timid users, our solution focuses on a gradual, progressive horror experience. By starting with lighthearted stories or non-scary text, users are gently guided to more thrilling content, eventually leading to scarier videos. This avoids overwhelming users with excessively frightening material in one go.

Our search engine prioritizes providing a balanced experience. Carefully curated and optimized search results introduce just enough thrill to engage users without causing excessive fear. This allows users to explore paranormal stories in a relatively comfortable and controlled environment, reducing the likelihood of abandoning their exploration due to fright.

This approach not only enhances users' sense of security but also encourages them to delve deeper into and learn about paranormal topics.

System Architecture Analysis

1. System Overview

This system is built on the Spring Boot framework and designed as a web application that searches for specific keywords. It fetches and sorts Google search results, providing users with a gradual, progressive experience tailored for a "timid audience" exploring spooky content.

---

2. Key Module Analysis

1. **Controller Layer**
   - **GoogleSearchController**
     - Serves as the interface between the user interface and backend services.
     - Features:
       - @GetMapping("/"): Displays the homepage.
       - @PostMapping("/search"): Handles search requests and returns processed results to the results page.
       - @GetMapping("/api/boogle"): Provides search results in JSON format via an API endpoint.
     - This layer handles user requests and passes data to the service layer for processing.
2. **Service Layer**
   - **GoogleQueryService**
     - Handles the fetching and processing of Google search results.
     - Key functionalities:
       - **Content Fetching**: JSoup is used to retrieve titles and links from Google search pages.
       - **Result Sorting**: Scores and ranks results based on keyword weights.
   - **KeywordWeights**
     - Defines the weight of each keyword, which determines the importance of the content in the ranking. This reflects the design of a progressive user experience.
3. **Data Structure Layer**
   - **WebNode and WebTree**
     - Use a tree structure to represent the hierarchy and relationships between web pages, with methods to calculate node scores.
     - Perform post-order traversal to compute the scores of the entire tree.
   - **WebPageHeap**
     - Implements both min-heap and max-heap for efficient sorting, ensuring results are output in descending order of scores.

4. **Utility Layer**
    - **WordCounter**
        - Provides functionality for counting keyword occurrences, aiding in the scoring of web pages.

---

3. Workflow

1. The user submits a query request to /search.
2. **GoogleQueryService** fetches Google search results and uses JSoup to parse page content.
3. Scores are calculated based on keyword weights defined in **KeywordWeights**.
4. **WebTree** and **WebPageHeap** are constructed, and post-order traversal is used to compute scores and rank results.
5. Results are returned to the user or provided in JSON format via /api/boogle.

---

4. System Features and Advantages

1. **Structured Design**
    - The clear layered architecture facilitates independent development and testing of functional modules.
2. **Flexible Sorting Logic**
    - Results can be further optimized by adjusting **KeywordWeights** to meet different user needs.
3. **Gradual Experience**
    - With keyword weights and tree structures, the system provides a progressive spooky experience aligned with psychological design principles.

---

5. Future Improvement Directions

1. **Dynamic Keyword Adjustment**
    - Incorporate learning algorithms to dynamically adjust keyword weights based on user behavior.
2. **Multi-language Support**
    - Expand support for searching in multiple languages to enhance user experience.
3. **Real-time Results Updates**
    - Use WebSocket to implement real-time search result updates.

Key ideas

1. **Keyword Weight Configuration**

   By setting keywords and assigning different weights, search results that are highly relevant to the main topic can be prioritized. For highly relevant keywords (e.g., "ghost stories," "supernatural"), a higher weight (such as 1.0) is assigned to ensure the search results closely match the user's needs. For indirectly related keywords (e.g., "thriller," "creepy"), a moderate weight (such as 0.5) is given, allowing users to gradually explore different levels of horror, from less frightening content to more intense stories.

   Additionally, consider less common keywords, such as "corpse water." These keywords may not strongly evoke horror, but they could appear lower in search results. Assigning them a higher weight ensures they appear higher in the results, enhancing the system's accuracy.

2. **Efficient Search Result Sorting and Filtering**

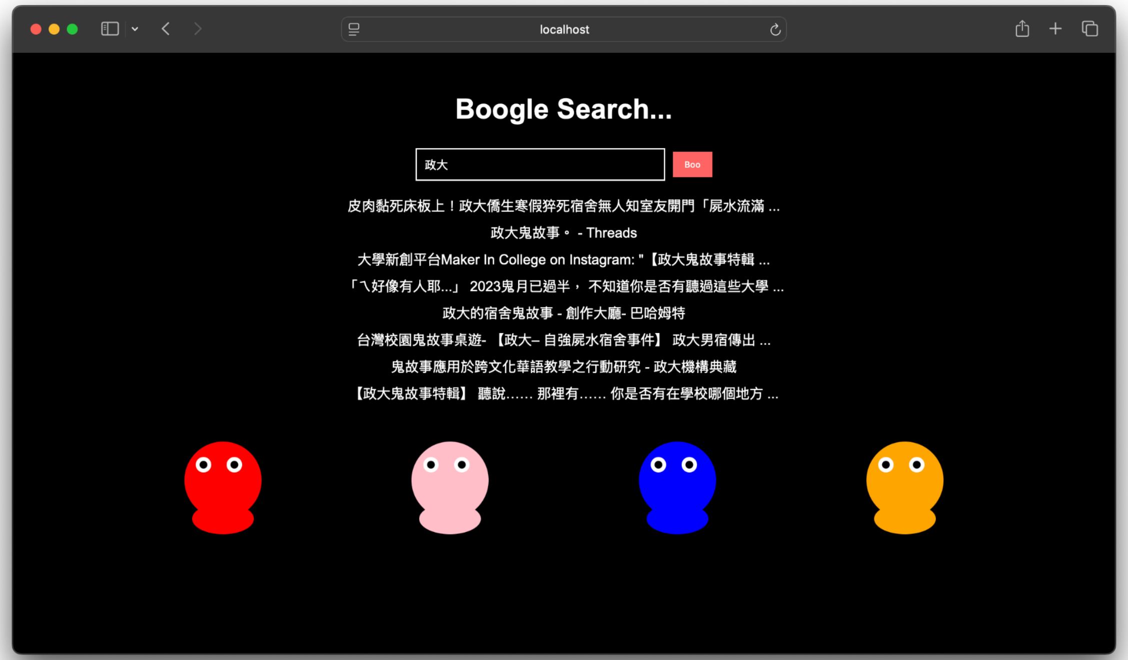
   Our system uses both Min-Heap and Max-Heap structures to manage high- and low-scoring search results, enabling an efficient sorting process. High-scoring results (≥ 2.8) are prioritized and sorted based on different levels of severity while ensuring they are highly relevant to the keywords. Low-scoring results are further filtered, with less relevant results placed at the end, improving search accuracy and enhancing the user experience.

3. **Subpage Search Support**

   Our system provides search functionality that extends beyond the homepage, allowing for deep exploration of various subpages within the website. The system assigns scores based on the content and relevance of each subpage, calculating an accurate score for each search result. This approach significantly enhances the depth and accuracy of search results, ensuring users receive highly relevant information and optimizing the overall search experience.

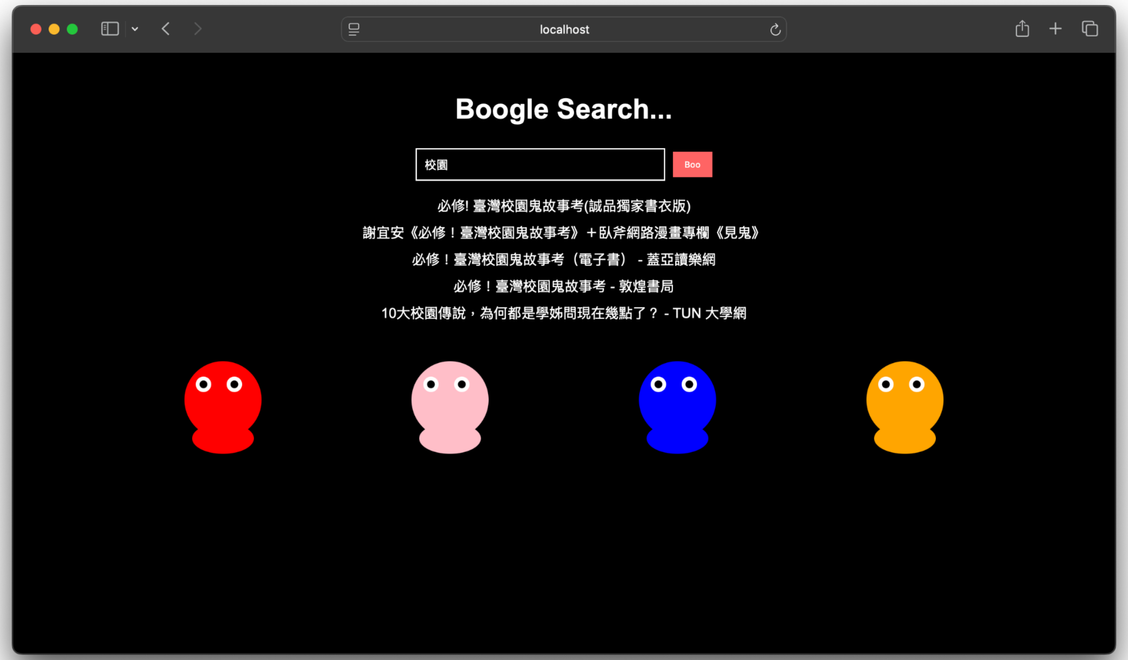
Results

1. NCCU



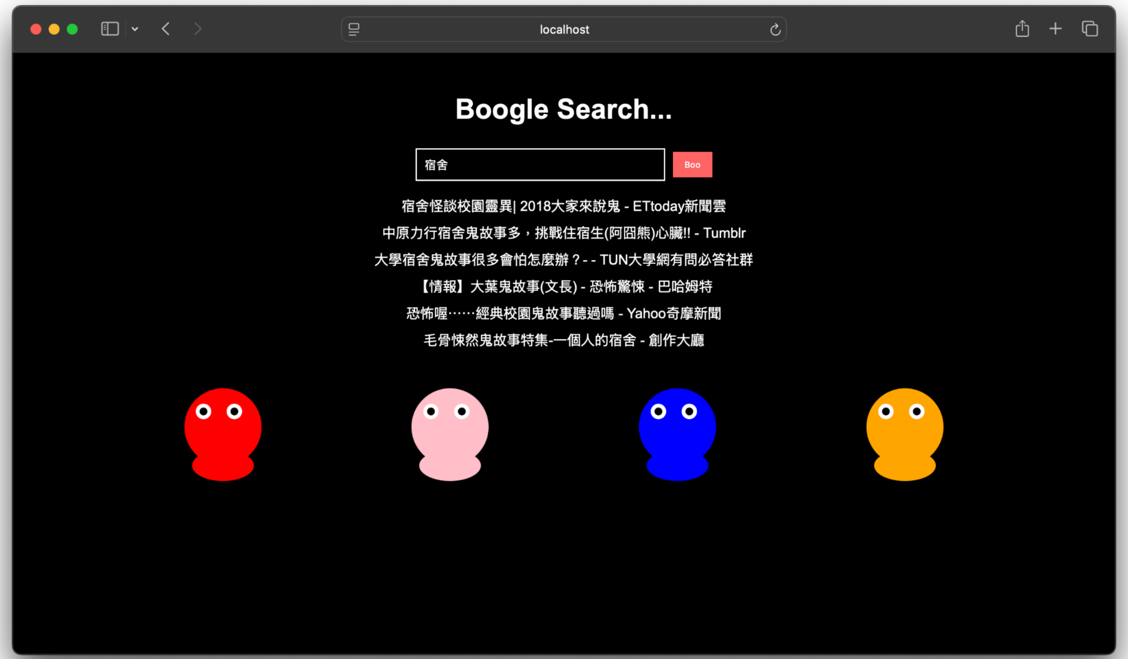
皮肉黏死床板上！政大僑生寒假猝死宿舍無人知室友開門「屍水流滿 ...
政大鬼故事。 - Threads
大學新創平台Maker In College on Instagram: "【政大鬼故事特輯 ...
「ㄟ好像有人耶...」 2023鬼月已過半， 不知道你是否有聽過這些大學 ...
政大的宿舍鬼故事 - 創作大廳- 巴哈姆特
台灣校園鬼故事桌遊- 【政大– 自強屍水宿舍事件】 政大男宿傳出 ...
鬼故事應用於跨文化華語教學之行動研究 - 政大機構典藏
【政大鬼故事特輯】 聽說…… 那裡有…… 你是否有在學校哪個地方 ...

2. Campus



必修! 臺灣校園鬼故事考(誠品獨家書衣版)
謝宜安《必修！臺灣校園鬼故事考》＋臥斧網路漫畫專欄《見鬼》
必修！臺灣校園鬼故事考（電子書） - 蓋亞讀樂網
必修！臺灣校園鬼故事考 - 敦煌書局
10大校園傳說，為何都是學姊問現在幾點了？ - TUN 大學網

3. Dormitory

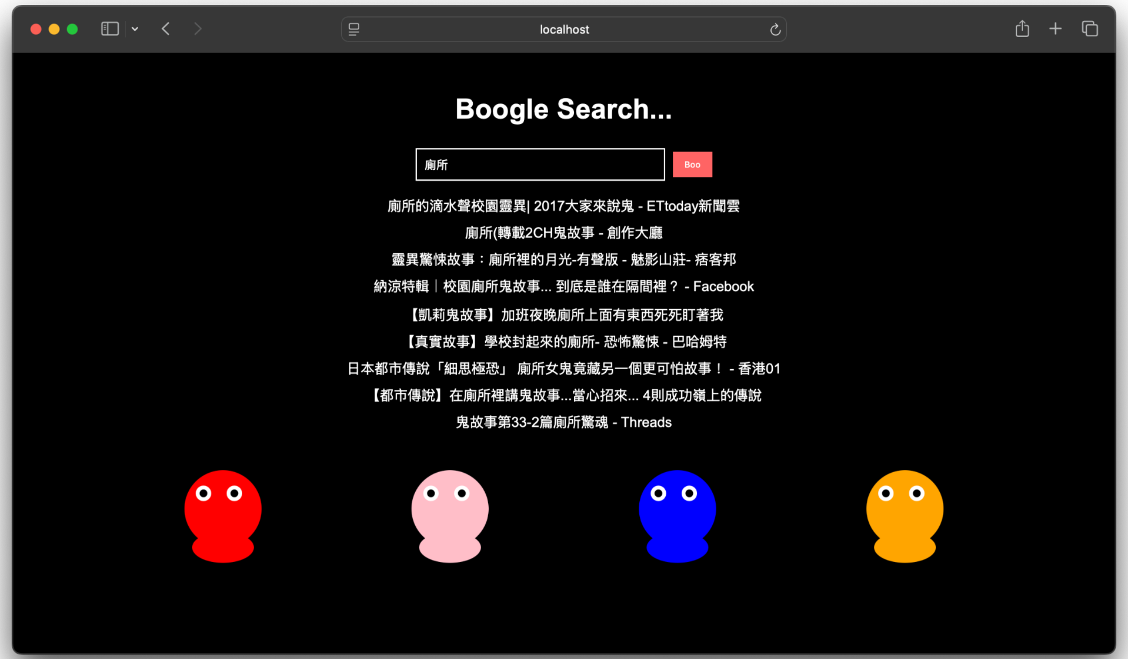


Boogle Search...

宿舍

宿舍怪談校園靈異| 2018大家來說鬼 - ETtoday新聞雲
中原力行宿舍鬼故事多，挑戰住宿生(阿囧熊)心臟!! - Tumblr
大學宿舍鬼故事很多會怕怎麼辦？- - TUN大學網有問必答社群
【情報】大葉鬼故事(文長) - 恐怖驚悚 - 巴哈姆特
恐怖喔……經典校園鬼故事聽過嗎 - Yahoo奇摩新聞
毛骨悚然鬼故事特集-一個人的宿舍 - 創作大廳

4. Restroom



Boogle Search...

廁所

廁所的滴水聲校園靈異| 2017大家來說鬼 - ETtoday新聞雲
廁所(轉載2CH鬼故事 - 創作大廳
靈異驚悚故事：廁所裡的月光-有聲版 - 魅影山莊- 痞客邦
納涼特輯｜校園廁所鬼故事... 到底是誰在隔間裡？ - Facebook
【凱莉鬼故事】加班夜晚廁所上面有東西死死盯著我
【真實故事】學校封起來的廁所- 恐怖驚悚 - 巴哈姆特
日本都市傳說「細思極恐」 廁所女鬼竟藏另一個更可怕故事！ - 香港01
【都市傳說】在廁所裡講鬼故事...當心招來... 4則成功嶺上的傳說
鬼故事第33-2篇廁所驚魂 - Threads

Improvement

The following adjustments and optimizations were made in the updated code:

1. **Integrated keywordWeights into encodeKeyword**

   Purpose: Ensures that all searches yield results relevant to the desired topic.

2. **Implemented the sublinks mechanism**

   Purpose: Enhances internal navigation within the webpage, improving user experience.

These updates aim to improve the overall functionality and usability of the system.

Prospect

1. **Dynamic Keyword Adjustment**
   - Incorporate learning algorithms to dynamically adjust keyword weights based on user behavior.
2. **Multi-language Support**
   - Expand support for searching in multiple languages to enhance user experience.