



Project 2 Prototype IR System

Manga Search

By

Miss. Pranungfun Prapaenee 6288034

Mr. Nopparat Pengsuk 6288103

Mr. Pongsakorn Piboonpongpun 6288107

A Report Submitted in Partial Fulfillment of
the Requirements for

ITCS414_Information Storage and Retrieval

Faculty of Information and Communication Technology

Mahidol University

2021

Table of contents

Introduction	3
Problem	4
Existing relevant systems	4
Methodology	5
Implementation	7
Using our Search-Engine	10
Results and Discussion	13
Conclusion	14
Reference	15

Introduction

This report will include a problem statement which is how we define the problem and how we will solve the problem, existing relevant systems which is a search system that is similar to our search system, methodology about how we will implement our search system, an implementation which is how we will develop our search system by using ElasticSearch and Flask framework, results of implementation, discussion about technical difficulties and challenges that we occur, and conclusion.

Problem

Our problem is focusing on

1. People who had a hard time finding the manga.
2. People who know a little information about manga details such as some of the character names, authors, and genres.
3. People who want to find manga by searching with keywords.

Therefore, we decided to create a search engine that will solve these problems by using Elastic search with our own data sources.

Existing relevant systems

From our research, we found that a lot of existing relevant systems such as [animenewsnetwork](#), [anisearch](#), and [myanimelist](#). They are similar search systems, but they are using relational databases. However, our system is using a non-relational database.

- **Animenewsnetwork**

Animenewsnetwork provides a variety of searches such as anime, manga, and people.

- **Anisearch**

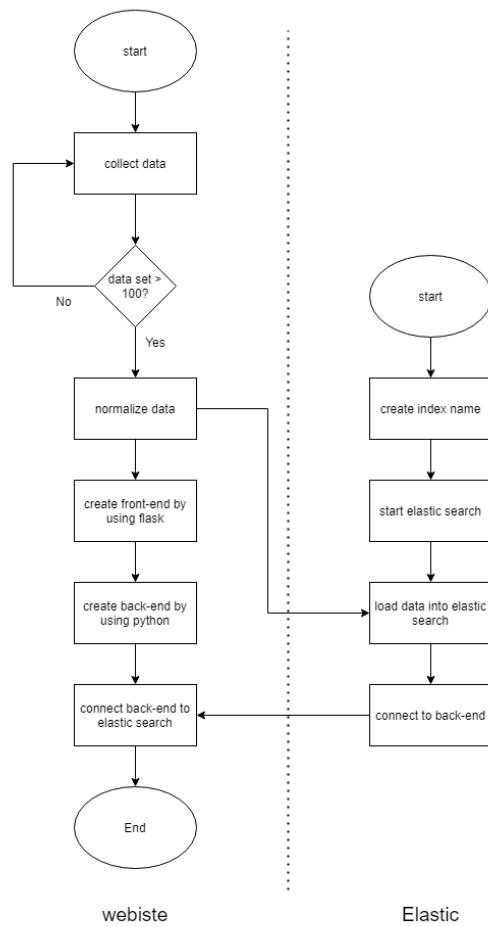
Anisearch is similar to anime news networks. They provide users to search anime, manga, and Asian TV shows.

- **Myanimelist**

Myanimelist is a little bit different from the previous two websites. It provides anime, manga, and community for users.

As we mentioned, these websites are using database lookup to search the data for users. Some of these websites also provide an API to search but use mangaid which is different from websites.

Methodology



Our methodology is simple. We will use ElasticSearch and python as core implementations. We will follow the step from the workshop and develop furthermore such as creating our own CSS style and having more pages. First, we start by collecting the data from sources such as [animenewsnetwork](#), [anisearch](#), and [myanimelist](#) until it reaches 100 data. Then, we will normalize the data first before loading it into the Elastic server. After loading data, we will develop the front end by using the Flask framework and Python as a back end. In the back end, we will use the Elastic search API to connect to the Elastic server.

The data that we collected will be a **JSON file** and will contain

1. Name
2. Relate name
3. Short story
4. Characters
5. Genres
6. Author
7. Publisher
8. Image list

Here is a snapshot of some datasets that we collected.

```
1  {
2      "name": "Attack on Titan",
3      "related name": [
4          "Shingeki no Kyojin"
5      ],
6      "short_story": "Humans are nearly exterminated by giant creatures called Titans. Titans are typically several
7      stories tall, seem to have no intelligence, devour human beings and, worst of all, seem to do it for the pleasure
8      rather than as a food source. A small percentage of humanity survived by walling themselves in a city protected by
9      extremely high walls, even taller than the biggest of titans. Flash forward to the present and the city has not
10     seen a titan in over 100 years. Teenage boy Eren and his foster sister Mikasa witness something horrific as the
11     city walls are destroyed by a colossal titan that appears out of thin air. As the smaller titans flood the city,
12     the two kids watch in horror as their mother is eaten alive. Eren vows that he will murder every single titan and
13     take revenge for all of mankind.",
14     "characters": [
15         "Eren Yeager",
16         "Mikasa Ackerman",
17         "Levi Ackerman",
18         "Reiner Braun",
19         "Connie Springer",
20         "Jean Kirschtein",
21         "Annie Leonhart",
22         "Historia Reiss",
23         "Erwin Smith",
24         "Sasha Braus",
25         "Pieck",
26         "Hange Zoe",
27         "Gabi Braun"
28     ],
29     "genres": [
30         "Action",
31         "Dark fantasy",
32         "Post-apocalyptic"
33     ],
34     "author": "Hajime Isayama",
35     "publisher": "Kodansha",
36     "img_list": [
37         "https://i.pinimg.com/564x/a6/aa/42/a6aa42b57ab861bce83e728bd5454e9d.jpg",
38         "https://i.pinimg.com/564x/a7/a4/2e/a7a42e99475a90dd34e1aa4302f995f3.jpg",
39         "https://i.pinimg.com/564x/4d/c0/b4/4dc0b42f152fd906db899ceca74f4861.jpg",
40         "https://i.pinimg.com/564x/2b/0f/80/2b0f805ce5c0e2c509dab1f4efab84e42.jpg",
41         "https://i.pinimg.com/564x/db/e3/bb/dbe3bb2334d3d8aa5bde3346834ceabf.jpg",
42         "https://i.pinimg.com/564x/73/61/02/736102e9321b84d15bba138cb6d40ed1.jpg",
43         "https://i.pinimg.com/564x/f4/87/55/f487551bfd9f2f4ffffa5a9618ae3c21.jpg",
44         "https://i.pinimg.com/564x/c2/31/b8/c231b8b49f40ea708912b3d0d52a9858.jpg",
45         "https://i.pinimg.com/564x/eb/fc/46/ebFc46e861369428131459988cf06d9.jpg",
46         "https://i.pinimg.com/564x/82/56/42/8256427c07d50e18d47e1d1cd5b50b09.jpg"
47     ]
48 }
```

Implementation

To implement, we manually collect data since our data sources don't have a proper API to use. In addition, we also create a file to help us normalize data because our data may contain some Japanese characters and we don't want to put it into the Elastic server. To collect the image list, we used the API from **serapi API[1]**. In the front end, we are using the **Flask framework[2]** and **Bootstrap[3]** as CSS. In the back end, we used API from **Elastic client[4]** to connect **Elastic server[5]**

Here is a snapshot of the API when we connect the back end to the Elastic server.

```
body = {  
    'size': page_size,  
    'from': page_size * (page_no-1),  
    'query': {  
        'multi_match': {  
            'query': keyword,  
            'fields': ['name', 'short_story', 'characters', 'genres', 'publisher', 'author'],  
            'fuzziness': 1  
        }  
    }  
  
res = es.search(index='manga_index', doc_type='', body=body)
```

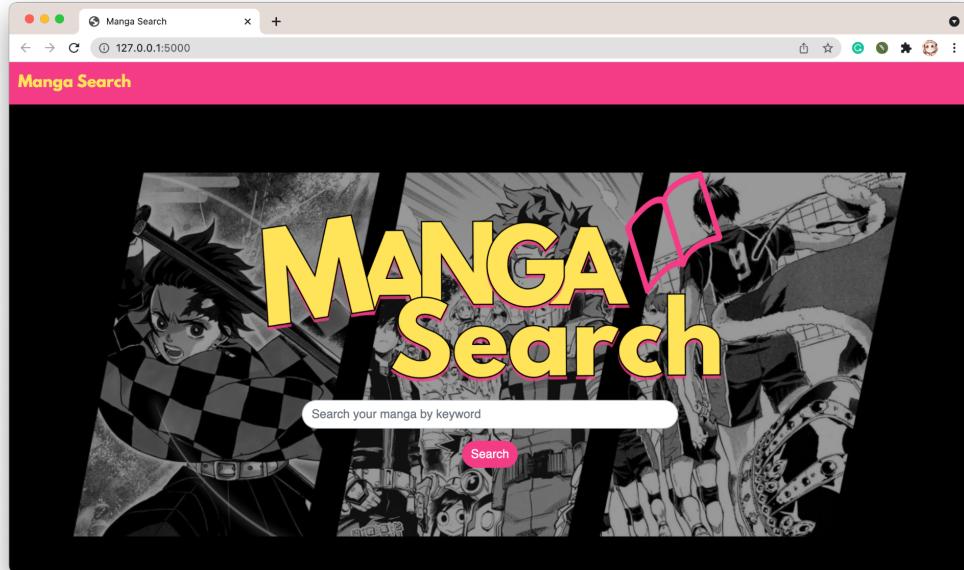
This code we extended from the workshop section. With a few lines of code, we can easily connect to the Elastic server. We used a multi-match type to search in the Elastic server. By using multi-match, we can find more relevant documents. We use fields: name, short story, characters, and genres to be searched because we think these fields are the most relevant to the query.

For the design, we use the dark theme to reduce blue light exposure and help with the eye strain that comes with prolonged screen time. That is why we chose this **palette[6]**.

Here is what our front end looks like.

1. Homepage

Manga search allows the user to find the Manga by typing the keyword in the search box.

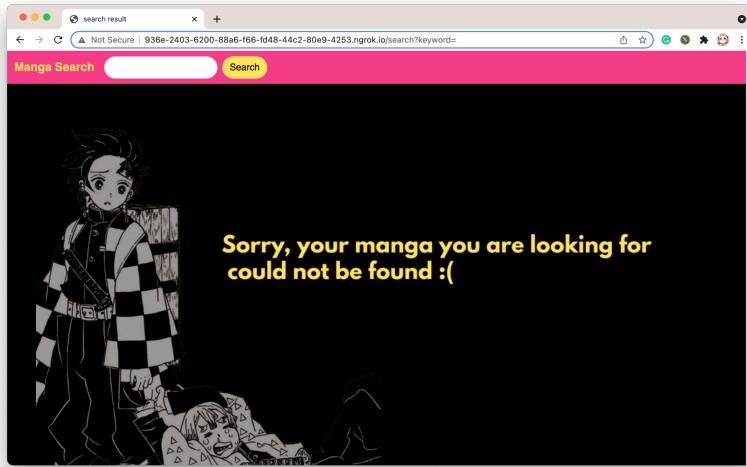


2. Search page

The search page displays a list of Manga titles that are relevant to the query.

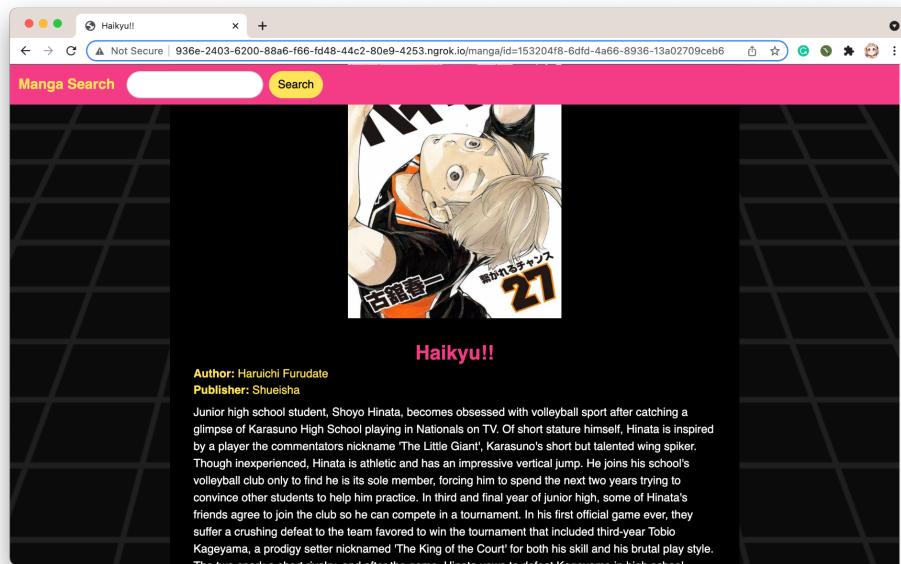
A screenshot of a web browser window titled "search result". The address bar shows the URL "Not Secure 936e-2403-6200-88a6-166-fd48-44c2-80e9-4253.ngrok.io/search?keyword=sport". The main content area has a pink header with the text "Manga Search" and a search bar containing the word "sport". Below the header, there is a list of manga titles with their descriptions and scores. To the right of the list, there is a grid of manga covers. The first few titles listed are Haikyuu!! (score: 6.035945), Slam Dunk (score: 3.6174746), Lament of the Lamb (score: 3.4487727), Ninja Girls (score: 3.3913257), and Bungo Stray dogs (score: 2.642642).

Otherwise, if it cannot find the Manga name it will inform the user that could not be found



3. Detail page

Contain the information about the Manga (Manga cover, Manga name, Author, Publisher, Genre, and related name)



In addition, you can see the full implementation of our search engine in the attached files. It will contain all of our necessary files including the dataset and the Python script.

Using our Search-Engine

Scenario 1: Search by character name

Data: Midoriya (character name)

Manga Search Midoriya Search

Boku no Hero Academia
In a world populated with superhumans, the superhero-loving Izuku Midoriya is without power. However...
score: 4.893585



Scenario 2: Search by genre

Data: sport

Manga Search sport Search

Haikyu!!
Junior high school student, Shoyo Hinata, becomes obsessed with volleyball sport after catching a gl...
score: 6.0355945

Slam Dunk
Haruko Akagi, ignorant of Hanamichi's history of misbehavior, notices his immense height and unwitting...
score: 3.6174746

Lament of the Lamb
The Takashiro family carries a genetic curse: They're vampires, well, sort of. Their condition causes...
score: 3.4487727

Ninja Girls
This story takes place during the civil war in Japan. There is a mysterious boy with a short horn on...
score: 3.3913257

Bungo Stray dogs
The story focuses on a young adult named Atsushi Nakajima. After being kicked out of his orphanage, A...



Haikyu!!
Author: Haruichi Furudate
Publisher: Shueisha

Scenario 3: Search by Character characteristics

Data: hero no hair

Manga Search hero no hair Search

One-Punch Man
One-Punch Man is a super hero who has trained so hard that his hair has fallen out, and who can over...
score: 5.703915

Horimiya
Although admired at school for her amiability and academic prowess, high school student Kyoko Hori ...
score: 4.7540464

Boku no Hero Academia
In a world populated with superhumans, the superhero-loving Izuku Midoriya is without power. However...
score: 4.5503006

Weak Hero
Once the bullies target you, its game over. The school is run by tyrants whose favorite hobby is tor...
score: 3.0861537

Solo Leveling
Ten years ago, the Gate appeared and connected the real world with the realm of magic and monsters.

Scenario 4: Search by Writer

Data: Atsushi Okubo

Soul Eater

Author: Atsushi Okubo
Publisher: Square Enix

Death City is home to the famous Death Weapon Meister Academy, a technical academy headed by the Shinigami Lord Death himself. Its mission: to raise "Death Scythes" for the Shinigami to wield against the many evils of their fantastical world. These Death Scythes, however, are not made from

Fire Force

Author: Atsushi Okubo
Publisher: Kodansha

Shinra Kusakabe is a third generation pyrokinetic youth who gained the nickname "Devil's Footprints" for his ability to ignite his feet at will, and was ostracized as a child for the fire that killed his mother and younger brother Sho 12 years ago. He joins Special Fire Force Company 8, which features other pyrokinetics who dedicated themselves to ending the Infernal attacks for good while investigating

Scenario 5: Search by Publisher

Data: Shueisha

The screenshot shows a web browser window with a pink header bar. The header bar contains the text "Manga Search" and "Shueisha" followed by a "Search" button. Below the header, the page content is displayed against a black background.

Anohana
A group of six childhood friends drifts apart after one of them, Meiko "Menma" Honma, dies in an acc...
score: 1.5815442

Boku no Hero Academia
In a world populated with superhumans, the superhero-loving Izuku Midoriya is without power. However...
score: 1.5815442

Black Clover
The series focuses on Asta, a young orphan who is left to be raised in an orphanage alongside his fe...
score: 1.5815442

Bleach
Ichigo Kurosaki is a teenager from Karakura Town who can see ghosts, a talent which lets him meet su...
score: 1.5815442

Blue Exorcist
Humans and demons are two sides of the same coin, as are Assiah and Gehenna, their respective worlds

On the right side of the screen, there is a large image of a group of people in school uniforms, likely students from one of the manga series. Below this main image are three smaller thumbnail images: a girl in a white dress, a boy in a red shirt, and a group of characters.

Results and Discussion

In this topic, we will discuss the challenge, difficulties, and the result.

For the challenge, we think the challenging part will be collecting the data. Since we manually collect it by hand and put it as a JSON file. Therefore, we had to deal with a lot of non-ASCII characters such as Japanese words and control characters. We solved this problem by creating our Python script.

Difficulties, we have a problem with missed typing when searching. As a result, ElasticSearch can't find the document for us. However, we find a way to fix it by using fuzziness. We set the value of fuzziness to be 1 because we think it can give the highest relevant documents rate.

From what we had developed, we think it's acceptable. However, if we have more data or have more fields, it will be better. The problems that we have faced are mostly about the dataset.

Furthermore, we think if we have to develop it in the future, we will create a script that can deal with non-ASCII and Japanese words better than this current script. Nevertheless, we will add more features and use more features that ElasticSearch provided.

Conclusion

In conclusion, from the problem that we defined which is search manga and we provide a search engine to solve the problem. We have developed our search engine by using ElasticSearch and Flask framework. Most of the difficulties and challenges are about the dataset that we manually collected. The result that we got is acceptable because we have a small dataset. If we have tools that can help us to collect all the relevant data, it will help improve the efficiency of search engines.

Reference

- [1] SerpApi "Google Images API" serpapi.com [Online]
Available: <https://serpapi.com/images-results> [Accessed: Nov. 30, 2021]
- [2] Flask "Flask framework" flask.palletsprojects.com [Online]
Available: <https://flask.palletsprojects.com/en/2.0.x/> [Accessed: Nov. 30, 2021]
- [3] Bootstrap "Bootstrap framework" getbootstrap.com [Online]
Available: <https://getbootstrap.com/> [Accessed: Nov. 30, 2021]
- [4] Python Elasticsearch Client "Doc | Python Elasticsearch Client" [Online]
<https://elasticsearch-py.readthedocs.io/>
Available: <https://elasticsearch-py.readthedocs.io/en/v7.15.2/> [Accessed: Nov. 30, 2021]
- [5] Elastic "Elastic"elastic.co [Online]
Available: <https://www.elastic.co/> [Accessed: Nov. 30, 2021]
- [6] Color Hunt
Available: <https://colorhunt.co/palette/11052c3d087bf43b86ffe459>