ITCS414: Information Storage and Retrieval

Project 2 step3: Prototype IR System

Submit to

Asst. Prof. Dr. Charnyote Pluempitiwiriyawej

Presented by

Kittitad	Jiraprasitchai	6288073	Section 1
Chancheep	Mahacharoensuk	6288092	Section 1
Kantapong	Matangkarat	6288160	Section 1

Faculty of Information and Communication Technology

Mahidol University

2021

Table of contents

AnimeFinder

Topic A

Introduction

Our AnimeFinder is built to meet the needs of watching Cartoon or Anime in their free time, and when searching the web page, we will show the details of that Anime for people who search to read roughly. About that particular Anime to know if it wants to watch it or not.

Data source we mainly reference information from MyAnimeList and AniList, this site lets you choose the type of Anime you want to watch. And have used search engines and concepts from Elasticsearch, Kibana, Inverted Index, TF/IDF, and web page structure and backend connections from HTML and Javascript.

Topic B

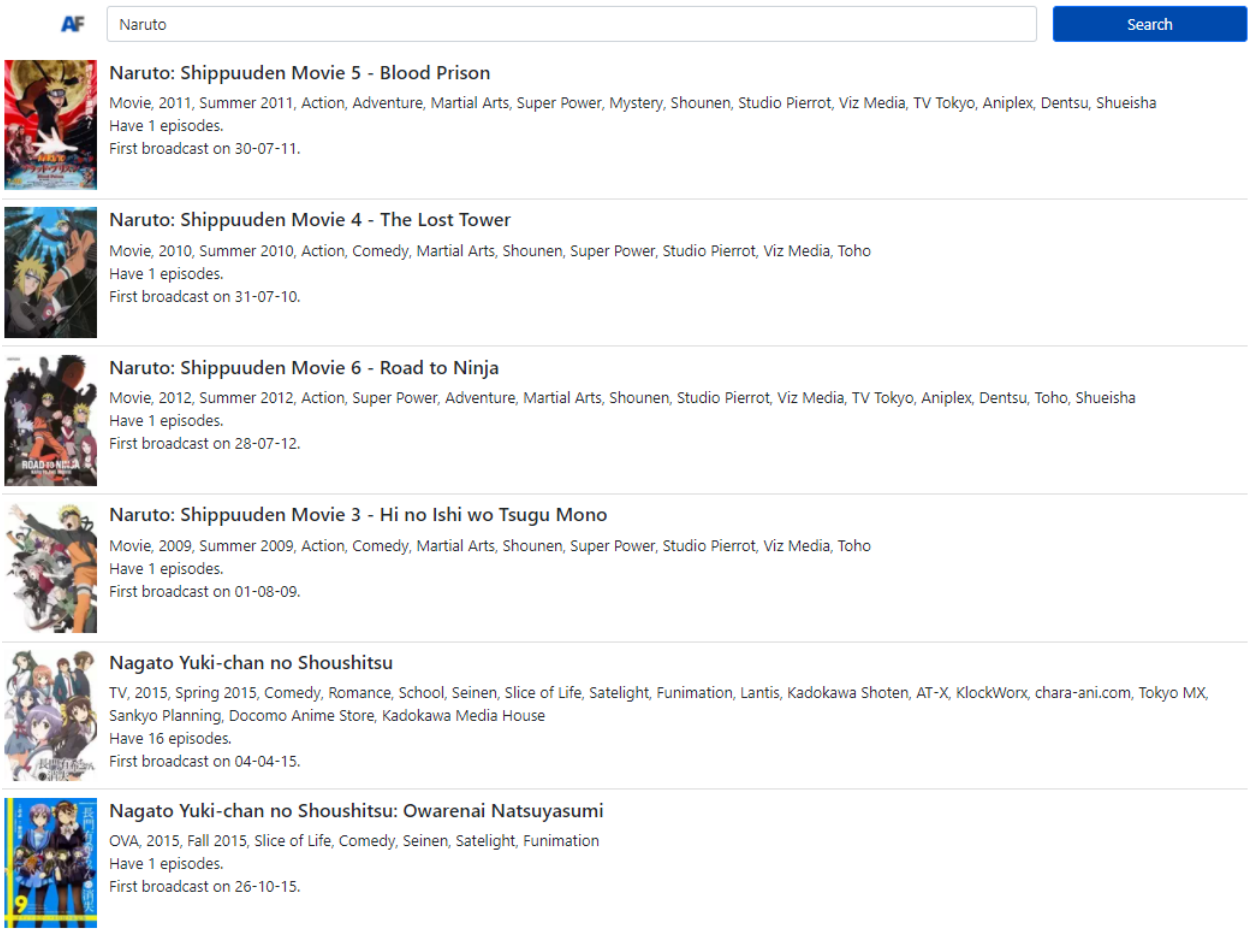
Problem(s) that you are trying to solve

Search system Some systems may be difficult to find websites. If you enter the data and it doesn't match 100 percent, it won't show the search results. Our system is made for easy and convenient search, just search for a short name and it will list all the match lists including Editing Distance.

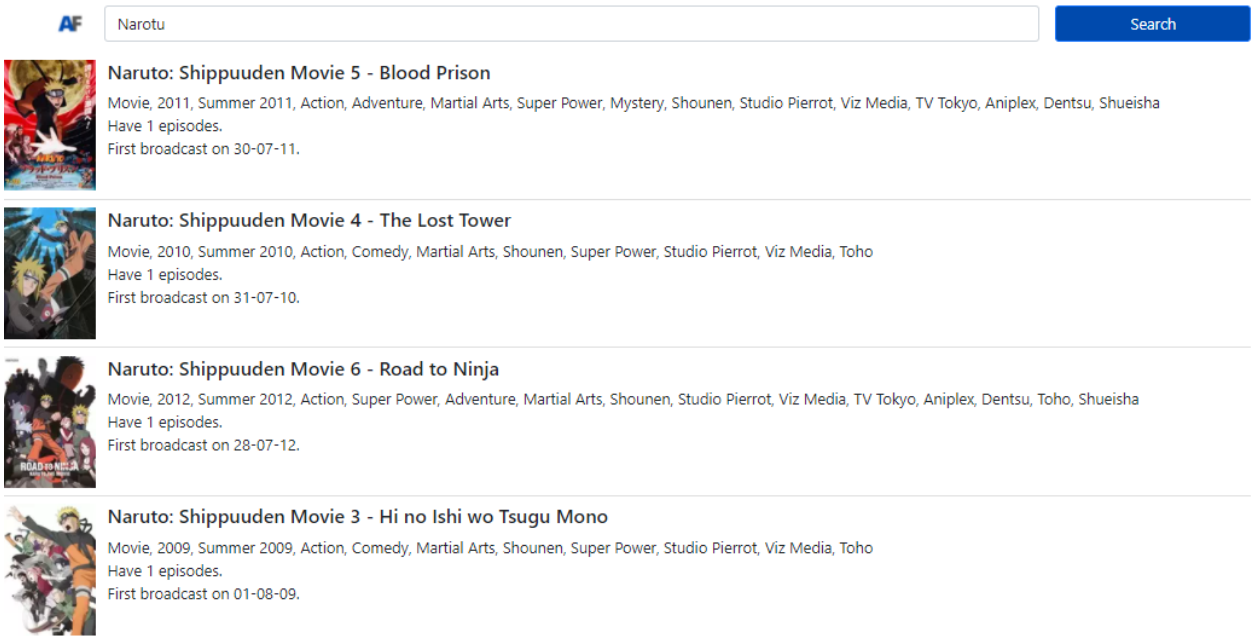
Topic C

(Track A) A literature review or existing relevant systems

Our search engine focuses on finding the anime you want. It's easy to use and comprehensive. This will give you information about the content of the anime you are looking for. Our web page can search by typing keywords, for example, if you want to find each Naruto chapter, you can just type "Naruto" and the system will list all Naruto chapters. It also displays titles that are close to what we're searching for. Nagato Yuki-chan no Shoshitsu



And another ability is when you type in Naruto but you typed it wrongly as "Narutu", the system can use Editing Distance to correct your own mistakes and search.



Topic D

Methodology

- **Elasticsearch**

We use it to send the form data of a JSON document to Elasticsearch. Make it possible to search and retrieve documents using ElasticSearch API can also use Kibana, a search engine to display the desired results.

- **Kibana**

In our search system, we use Kibana to import data into a JSON file to extract functionality from Elasticsearch. to search for information and display results based on what is input into the system.

- **Inverted Index**

An inverted index is an index data structure that holds mappings from content, words, or numbers to locations in a document or document set.

- **TF/IDF**

Term frequency This is what allows us to see word frequency. The purpose of TF is to look at how often each word appears in each document. The more search terms appear, the more relevant the document is.

Inverse document frequency We use it to value a word in all documents. If a word is found in multiple documents, it has less importance.

- **HTML**

We use it as the main page. and used to position the placement of objects Each part of the website looks beautiful. Including CSS and Bootstrap with grid support for responsive layout and ready-made components to use.

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Search Engine</title>
</head>
<body>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC" crossorigin="anonymous">
    <div class="container">
        <div class="row gy-5">
            <div class="col-12" style="margin-top:-10%">
                <img class="img-fluid col-md-6 offset-md-3" src="../static/img/logo3.png">
            </div>
            <div>
                <form class="col-md-6 offset-md-3 text-center" method="GET" action="{{ url_for('search') }}">
                    <div class="container">
                        <div class="row" >
                            <div class="col-sm-9"><input type="text" class="form-control" style="width: 100%;" name="keyword" placeholder="Type your anime name or genre......"></div>
                            <div class="col-sm-3"><button type="submit" class="btn" style="width: 100%; background-color: #004AAD; color: white;">Search</button></div>
                        </div>
                    </div>
                </form>
            </div>
        </div>
    </div>
</body>
</html>
```

```html
<!DOCTYPE html>
<html>
    <head>
        <title>Search engine</title>
        <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-F3w7mX95PdgyTmZZMECAngseQB83DfGTowi0iMjiWaeVhAn4FJkqJByhZMI3AhiU" crossorigin="anonym
    </head>

    <body>
        <header>
            <nav class="navbar navbar-expand-lg navbar navbar-light fixed-top" style="background-color: #004AAD;">
                <div class="container">
                <a class="navbar-brand" style="color: white;" href="{{ url_for('index') }}">Anime Finder</a>
                <div class="collapse navbar-collapse" id="navbarResponsive">

                </div>
                </div>
            </nav>
        </header>
        <div class="container">
            <form style="margin-top: 5%;margin-bottom: 10px;" class="row g-3" method="GET" action="{{ url_for('search') }}">
                <div class="col-1" >
                    <a href="{{ url_for('index') }}"><img class="img-fluid col-md-6 offset-md-3" style="width: 40%; margin-left: 60%;" src="./static/img/logo2.png"></a>
                </div>
                <div class="col-9">
                    <input type="text" class="form-control" name="keyword" placeholder="Type your anime name or genre......" value="{{ keyword }}">
                </div>
                <div class="col-2">
                    <button type="submit" style="width: 100%; background-color: #004AAD;" class="btn btn-primary">Search</button>
                </div>
            </form>
            {% if hits %}
            <div class="row">
                <ul class="list-group list-group-flush">
                    {% for doc in hits %}
                    <li class="list-group-item">
                        <div class="row">
                            <div class="col-sm-1"><img src="{{ doc['anime_image_path'] }}"></div>
                            <div class="col-sm-11">
                                <div class="row">
                                    <h5>{{ doc['anime_title'] }}</h5>
                                </div>
                                <div class="row">
                                    <h7>{{ doc['tags'] }}</h7>
                                </div>
                                <div class="row">
                                    <h7>Have {{ doc['anime_num_episodes'] }} episodes.</h7>
                                </div>
```

- **Python**

Python that we use to write web service, we have written the web page to show the data of each anime with anime name, release year, anime in which season, anime tag, and how many episodes total.

Tamala 2010: A Punk Cat in Space

Movie, 2002, Fall 2002, Psychological, Comedy, Dementia, Sci-Fi
Have 1 episodes.
First broadcast on 19-10-02.

```python
from flask import Flask, render_template, jsonify, request
from markupsafe import escape
from flask import render_template
from elasticsearch import Elasticsearch

import math

es = Elasticsearch()
app = Flask(__name__)

@app.route('/')
def index():
    return render_template('index.html')

@app.route('/search')
def search():
    page_size = 10
    keyword = request.args.get('keyword')
    if request.args.get('page'):
        page_no = int(request.args.get('page'))
    else:
        page_no = 1
    body ={
        'size': page_size,
        'from': page_size*(page_no-1),
        'query': {
            'multi_match': {
                'query': keyword,
                'fields': ['anime_title','tags'],
                'fuzziness' : 'auto',
                'fuzzy_transpositions' : 'true'
            }
        }
    }
    res = es.search(index='anime_index', doc_type='',body=body)
    hits = [{'anime_title': doc['_source']['anime_title'],'anime_num_episodes': doc['_source']['anime_num_episodes'], 'anime_start_da
    page_total = math.ceil(res['hits']['total']['value']/page_size)
    total = res['hits']['total']['value']

    return render_template('search.html', keyword=keyword, hits=hits, page_no=page_no, page_total=page_total, total = total)
```

Topic E

Implementation

We optimize the HTML web page and we link it. Elasticsearch with Kibana to connect and import the raw Json data files to Elasticsearch for Kibana to retrieve, and we use Python to connect to the web page and run the script to get the webpage ready to search.

```
page_size = 10
keyword = request.args.get('keyword')
if request.args.get('page'):
    page_no = int(request.args.get('page'))
else:
    page_no = 1
body ={
```

Determine the type of data we use to find

```
body ={
    'size': page_size,
    'from': page_size*(page_no-1),
    'query': {
        'multi_match': {
            'query': keyword,
            'fields': ['anime_title','tags'],
            'fuzziness' : 'auto',
            'fuzzy_transpositions' : 'true'
        }
    }
}
```

```
res = es.search(index='anime_index', doc_type='',body=body)
hits = [{'anime_title': doc['_source']['anime_title'],'anime_num_episodes': doc['_source']['anime_num_episodes'],
'anime_start_date_string': doc['_source']['anime_start_date_string'],
'anime_image_path': doc['_source']['anime_image_path'], 'tags': doc['_source']['tags']} for doc in res['hits']['hits']]
page_total = math.ceil(res['hits']['total']['value']/page_size)
total = res['hits']['total']['value']
```

- It is a command to check after typing input and then go to search which index.
- hits is a variable that holds the pending value and returns the specified value.
- page_total The formula calculates the page, how many datasets should a page have and how many total datasets are suitable for the data.

Topic G

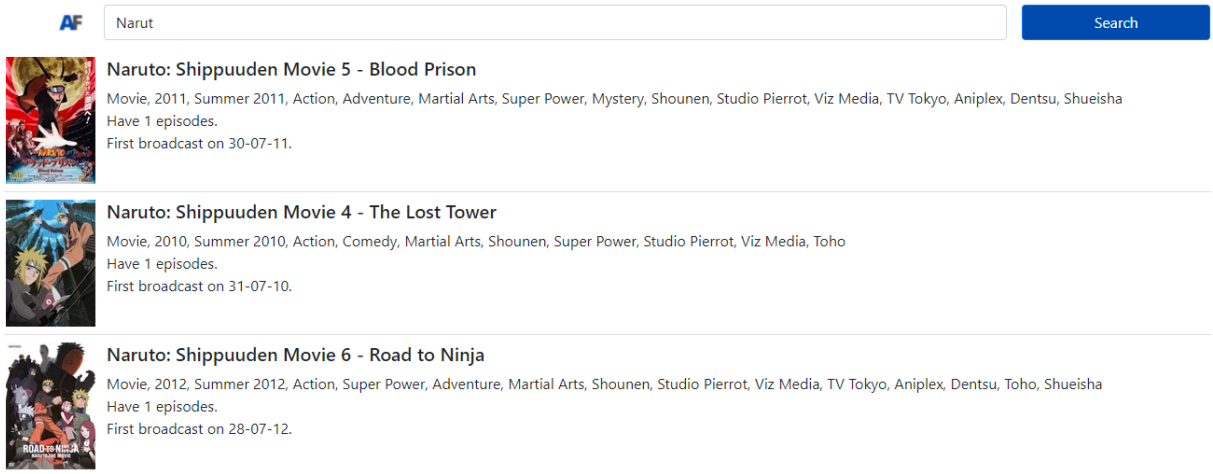
Results and Discussion

This section gives an example to understand what search cases in our AnimeFinder can support and how the search format can be flexible based on the amount of data. With the same name, the easier it is to search for Anime names.

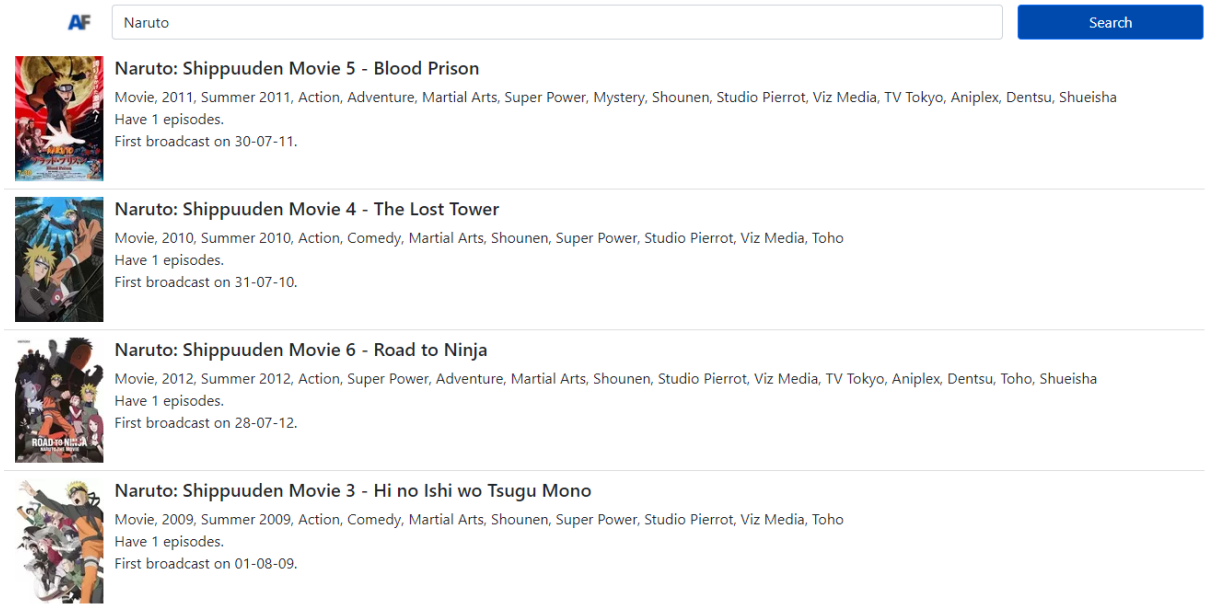
In this search system, we don't output the score result, we don't think it makes sense, but we output the score from Kibana for use if you want to see. The score you can see in the Kibana as in the picture. And our system is auto ranking.


```
        {
          "_index" : "anime_index",
          "_type" : "_doc",
          "_id" : "G9hdFH0B-OTwT80879iy",
          "_score" : 4.322431,
          "_source" : {
            "anime_title" : "Naruto: Shippuuden Movie 5 - Blood Prison",
            "tags" : "Movie, 2011, Summer 2011, Action, Adventure, Martial Arts, Super Power, Mystery, Shounen, Studio Pierrot, Viz Media, TV Tokyo, Aniplex, Dentsu, Shueisha"
          }
        },
        {
          "_index" : "anime_index",
          "_type" : "_doc",
          "_id" : "GdhdFH0B-OTwT80879iy",
          "_score" : 4.003389,
          "_source" : {
            "anime_title" : "Naruto: Shippuuden Movie 4 - The Lost Tower",
            "tags" : "Movie, 2010, Summer 2010, Action, Comedy, Martial Arts, Shounen, Super Power, Studio Pierrot, Viz Media, Toho"
          }
        },
        {
          "_index" : "anime_index",
          "_type" : "_doc",
          "_id" : "HdhdFH0B-OTwT80879iy",
          "_score" : 4.003389,
          "_source" : {
            "anime_title" : "Naruto: Shippuuden Movie 6 - Road to Ninja",
            "tags" : "Movie, 2012, Summer 2012, Action, Super Power, Adventure, Martial Arts, Shounen, Studio Pierrot, Viz Media, TV Tokyo, Aniplex, Dentsu, Toho, Shueisha"
          }
        },
        {
          "_index" : "anime_index",
          "_type" : "_doc",
          "_id" : "F9hdFH0B-OTwT80879iy",
          "_score" : 3.2776175,
          "_source" : {
            "anime_title" : "Naruto: Shippuuden Movie 3 - Hi no Ishi wo Tsugu Mono",
            "tags" : "Movie, 2009, Summer 2009, Action, Comedy, Martial Arts, Shounen, Super Power, Studio Pierrot, Viz Media, Toho"
          }
        }
      ]
    }
```

1.This is a case of inserting "Narut" and the system can detect that we want to find Naruto.



2.This is a case of inserting "Naruto" as a full text search and the system will list the output of every Naruto that Data has.



3. This is a case of typing the wrong input type in some text correctly. The system will use Editing Distance to detect correct words.

AF Naruto Ninja Movie10 Search

Naruto: Shippuuden Movie 6 - Road to Ninja

Movie, 2012, Summer 2012, Action, Super Power, Adventure, Martial Arts, Shounen, Studio Pierrot, Viz Media, TV Tokyo, Aniplex, Dentsu, Toho, Shueisha
Have 1 episodes.
First broadcast on 28-07-12.

Naruto: Shippuuden Movie 5 - Blood Prison

Movie, 2011, Summer 2011, Action, Adventure, Martial Arts, Super Power, Mystery, Shounen, Studio Pierrot, Viz Media, TV Tokyo, Aniplex, Dentsu, Shueisha
Have 1 episodes.
First broadcast on 30-07-11.

Naruto: Shippuuden Movie 4 - The Lost Tower

Movie, 2010, Summer 2010, Action, Comedy, Martial Arts, Shounen, Super Power, Studio Pierrot, Viz Media, Toho
Have 1 episodes.
First broadcast on 31-07-10.

Naruto: Shippuuden Movie 3 - Hi no Ishi wo Tsugu Mono

Movie, 2009, Summer 2009, Action, Comedy, Martial Arts, Shounen, Super Power, Studio Pierrot, Viz Media, Toho
Have 1 episodes.
First broadcast on 01-08-09.

Topic H

Conclusion

Our search engine uses Elasticsearch. which includes Inverted Index, a data structure from a JSON file and Kibana gets the data into the raw data format. After that, we write Python to work with. Elasticsearch and write HTML to display and arrange the position of each object to make our website beautiful, including Bootstrap and CSS to help organize the web page as needed. When running into the web page, we will be able to type the name of a Cartoon or Anime that we want to search into, and the system will display results.