# Bilibili

Group Member: Fangfang Wan, Ling Dai, Yilun(Beca) Dai

# Introduction

- Bilibili (https://www.bilibili.com/): A Chinese video website with “Bullet Screen” (comments flying through the screen)

- **Goal:** Take advantage of explicit user feedback to build functional algorithms.

# Project Overview

Data Collection

Text Processing

Functional Algorithms

- Web scraping

- Based on jieba
- Able to preserve emoticons and other special patterns

- A video recommendation algorithm based on user feedback
- Keyword search algorithm
- Visualization (Word Cloud)

# Data Collection

- Collected video ranking data in JSON file from 11 different sections.
    - Too many videos on https://www.bilibili.com (over 20,000,000)

- Use web scraping to collect data for these videos on the ranking (~1200)

- Challenge: encountered A/B testing during scraping

# Overview of Collected Data

- Categories and number of videos
    - Anime 148
    - Daily Life 118
    - Dance 145
    - Domestic & Original 130
    - Entertainment 112
    - Fashion 169
    - Games 152
    - Kichiku 177
    - Movies 161
    - Music 125
    - Science 124

# Second Challenge: Text Processing

- Segmenting Chinese words is way harder than segmenting English words (no whitespace between Chinese words)

- Existing Chinese word segmentation algorithms tend to break up non-Chinese patterns, such as emoticons and Japanese words.

- Chinese Internet language contains a lot of repetitions (e.g. ‘2333’ and ‘2333333333’, ‘哈哈哈’ and ‘哈哈哈哈哈’)

# Solution: A Smart Word Segmentation Algorithm

- Based on jieba
- Able to preserve emoticons and other patterns (e.g. ‘(๑¯◡¯๑)’, ‘(ง •̀_•́)ง’)
- Shorten repetitive patterns:
  - ‘23333…..’ to ‘233’
  - ‘哈哈哈......’ to ‘哈哈哈’
- Remove Stopwords (e.g. ‘哈’, ‘哦’, ‘不’)
- Example: 高能预警演示这不是演习。。66666!?! This(ง •̀_•́)ง is not a test哈哈

# A Video Recommendation Algorithm Based on User Feedback

- Using the data scraped from Bilibili, we implemented a video recommendation algorithm that is based only on user feedback (content of bullet screens)

- Deep learning (gensim: Doc2Vec model)

- User interface (django)

# Test Case 1: 渣渣辉

【渣渣辉】我是贪玩小辉

[http://127.0.0.1:8000/](http://127.0.0.1:8000/)

# Test Case 2: 五五开

【五五开】目标是开挂大师

[http://127.0.0.1:8000/](http://127.0.0.1:8000/)

# Test Case 3: Ballet Beautiful

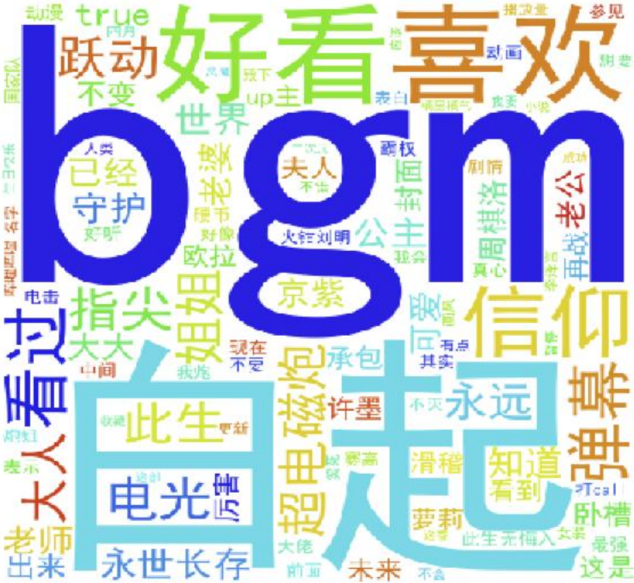
力荐！【中英双语】Ballet Beautiful 美丽芭蕾P4 大腿内侧燃脂塑形

http://127.0.0.1:8000/

# Search Algorithm Based on Keywords

- Not optimal at current stage (infer_vector() method in gensim produces different result for each run, causing our search algorithm to be unstable)
- Quality of search results increases as the number of keywords increases
- Possibly add more constraints on search criteria (e.g. title)
- http://127.0.0.1:8000/

# Visualization (Word Cloud) http://127.0.0.1:8000/



# Future Work

- Integrate visualization with other functions in a meaningful way (e.g. interactive visualization using Tableau)

- Improve the keyword search algorithm

- Build a neural network, instead of calculating similarity score every time