Data Statistics and Visualization of Social Media

Introduction

Nowadays, there is an increasing demand for social media sites, whose hot lists often reflect current social resources. Our project is a way to count and visualise what is happening on social media to get a sense of what is currently happening and what topics are getting attention.

The whole project consists of five parts: sample selection rule, data crawling, keyword extraction algorithms and comparison, analysis of the results and future improvements.

For the sample selection rule, we have the following expectation: we hope to observe the current social attention through the visual word cloud map. Therefore, we chose some hot list data from mainstream media (Weibo, Zhihu). The word with more weight means that people tend to pay attention to news containing this word.

For data crawling, we use BeautifulSoup4 to crawl hot list data from Weibo and Zhihu.

For keyword extraction, the simplest method must be obtained by word frequency, and TF-IDF is a more advanced method, we compared the good and bad of these two algorithms and finally made a choice.

For the results analysis section, there are two rules:

we want to get some hot data over a long period of time, because it corresponds to a period of time that occupies more social resources. Therefore, we analyzed the change of hot list data on the time gradient, and the words that appeared within several days were corresponding to the hot events that the society paid high attention to.

We want to get the focus of the different sites. For example, Weibo is more in the field of social news, while Zhihu is more in the field of science and technology. Therefore, we compared the word cloud map of the hot list data of the two websites.

In the last section, we conclude by setting out where future improvements can be made.

Crawl hot list data

The crawler section contains the following four parts:

1. Crawler tool selection
2. Login status acquisition

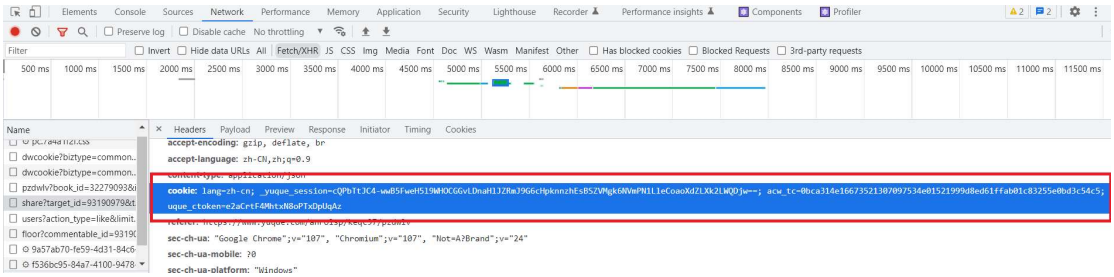
3. Craw the data
4. Write the data to the file in json format

To be able to crawl the hotlist, we need web scraping tools, which contains following functions:

The construction of an agent to download, parse, and organize data from the web in an automated manner; Extract relevant data from unstructured sources on the Internet; Also known as screen scraping, web harvesting, and web data extraction; Can extract text, contact information, images, videos, product items, etc.

As a result, we choose BeautifulSoup[1].Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree.

We use it as a crawler to crawl Weibo and Zhihu's hotlist data. The Weibo and Zhihu hotlist data requires user login verification, so we need to get the login status Cookie from the logged in user's page.

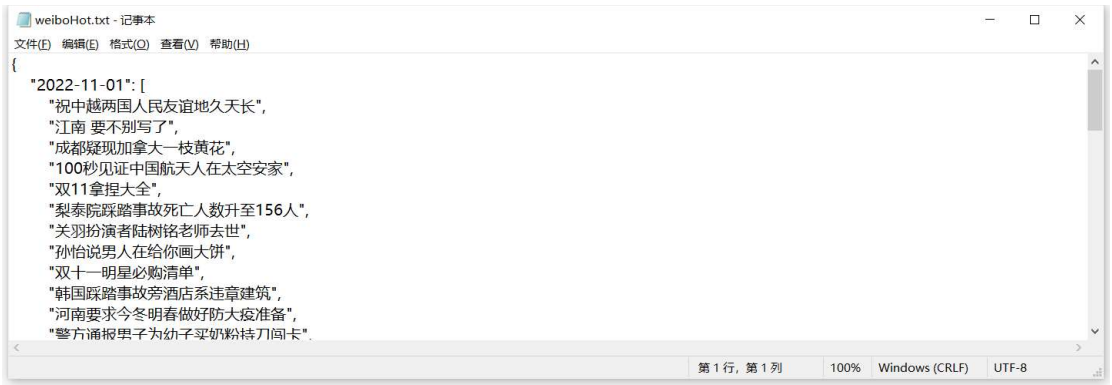


First create a crawler class. Define the login authentication cookie, the url of the site to be crawled, the request header, and the final data to be obtained as properties of the crawler.

Connect to the site, use requests to get the site text and use beautifulsoup to optimise the interface returned.

Get the data corresponding to the hotlist tag, which will be stored in the data attribute according to {date: list of data}.

Write the data attribute to the file in json format. This is shown below



Key word Extraction

The implementation of this part mainly includes the following steps:
1. Extract words from Chinese corpus
2. Statistics word frequency based on TF-IDF algorithm
3. Generate word cloud map based on word frequency

Extract words from Chinese corpus

For English text, we can directly use space judgment to segment words. But if it is continuous text in Chinese, there are no word boundaries in Chinese sentences, so we need to use some algorithms to segment words.

Common methods of Chinese word segmentation

For Chinese word segmentation, the commonly used algorithms are HMM (Hidden Markov Model), CRF (Conditional Random Field), SVM (Support Vector Machine), deep learning and other algorithms.

Jieba

Because of the difficulty of implementing Chinese word segmentation, we use a tool – Jieba, to help us to segment Chinese words.
"Jieba" (Chinese for "to stutter") Chinese text segmentation: built to be the best Python Chinese word segmentation module [2]. (https://github.com/fxsjy/jieba)

In this step, we import jieba, and prepare the user dictionary and stopwords, then input Chinese text, and ouput the segment words list.

```
Loading model cost 1.483 seconds.
Prefix dict has been built successfully.
老版,三国演义,关羽,扮演者,陆树铭,去世,老先生,一生,哪部,作品,印象,深刻,本土,科学家,Science,百万,大奖,热
祝中越,两国人民,友谊,地久天长,江南,不别,成都,疑现,加拿大,一枝黄花,见证,中国航天,太空,安家,拿捏,大全,梨泰
```

Statistics word frequency: compare tf and tf-idf

Word frequency based on TF

For keyword extraction, the easiest way is to get it directly through the word frequency. The input is words list from previous step, and output the MultiDictionary.

```
[('原神', 4), ('空间站', 3), ('富士康', 3), ('员工', 3), ('隔离', 3), ('角色', 3), ('重建', 3), ('公布', 3), ('生活', 3), ('热议', 2), ('事件', 2), ('相亲', 2),
[('踩踏', 5), ('事故', 5), ('双十一', 3), ('男子', 3), ('上线', 3), ('微博', 3), ('会员', 3), ('女子', 3), ('太空', 2), ('韩国', 2), ('男童', 2), ('续充', 2), (
```

However, the extraction of keywords based only on frequency counts is a high-frequency word, but high-frequency words are not necessarily keywords. For example, stop words such as "了" and "的" occur frequently in several documents, and obviously these stop words cannot be keywords. Therefore, an algorithm that can both count high frequency words and distinguish between categories of words (high independence) is our first choice.

Word frequency based on TF-IDF

In order to understand this algorithm, we need to understand what a TF is and what an IDF is.
TF is Term Frequency， a measure of how frequently a term t appears in a document d:。

$$tf_{t,d} = \frac{n_{t,d}}{Number\ of\ terms\ in\ the\ document}$$

IDF is Inverse Document Frequency， 代表的是 a measure of how important a term is.

$$idf_t = log\ \frac{number\ of\ documents(N)}{number\ of\ documents\ with\ term\ t\ (df_t)}$$

The main idea of TF-IDF is: if a word appears frequently TF in an article and rarely appears in other articles (high IDF), it is considered that the word or phrase has a good ability to distinguish between categories and is suitable for classification. We treat each line of hot search as a term, and then segment each line.

Wordcloud and Analysis

According to the above two methods, then we get two different word clouds. When using first basic method, we can see some stop words have high term frequency, but obviously they cannot appear as keywords. Such as “被”, “的” in this word cloud. About the word cloud, we import WordCloud library [3].

pic.1 wordcloud_tf

pic.2 wordcloud_tfidf

Vertical Comparison

Using multiple days of hotlist data as a document library, TF-IDF values are generated and then scored for each day of hotlist data, enabling the change in hotlist keywords under different days, i.e. the temporal trend of hotlist keywords.

We collected the hot search keywords on Weibo from October 31st to November 2nd, and displayed them in word clouds as follow.

pic.3 wordcloud_10.31

pic.4 wordcloud_11.1

pic.5 wordcloud_11.2

Horizontal Comparison

We compared the word cloud map of the hot list data of the two websites. Using TFIDF method to extract hot keywords for 11-02.

From the results, we can see that Zhihu's word cloud graph contains words in specialized fields, such as ARM, material, space, etc., which is also in line with the characteristics of technical forums. Compared with Zhihu, Weibo hot search is closer to the field of people's livelihood, Lanzhou, notification, police, etc. represent the hot social news.

pic.1 zhihu 11-02

pic.2 weibo 11-02

Improvements

In this project, the Word Cloud is a good visual expression of hot search, especially able to highlight the most popular keywords. At the same time, in this project, the color of Word Cloud is mainly black and white, which cannot give better play to its strong visual impact. Adding various colors to the Word Cloud may be an improvement.

We can use different keyword extraction methods, e.g. BM25, TextRank to finally come up with a comprehensive keyword extraction algorithm can be obtained by weighting the above algorithm.

Reference

[1] https://code.launchpad.net/~leonardr/beautifulsoup/bs4.
[2] https://github.com/fxsjy/jieba.
[3] https://amueller.github.io/word_cloud

Individual Contributions

Ao Ouyang

Responsible for introduction, data crawling and analysis of results, PPT and report writing. Preparation of the final PPT and report and presentation.

Jianhua Huang

Responsible for the keyword extraction section, the word cloud generation section, PPT and report writing. Preparation of the final PPT, report and presentatoin.

Boshen min

Improvements writing in the report.

Appendix

Data Statistics and Visualization of Social Media

Group 7
Members: Ao OUYANG、 Boshen MIN、 Jianhua HUANG

CONTENTS

1 Sample Selection Index

Hot Topic	Hope to observe the current social attention through the visual word cloud. Find out what people pay attention to.
vertical comparison	Look at how social concerns change over time
Horizontal Comparison	Observe the similarities and differences of different popular websites

1 Mainstream media Websites in China

Weibo

Hot search in Weibo reflects affairs news and entertainment news

Zhihu

A question-and-answer platform.
In addition to news discussions, Zhihu Hot List will also have some technical questions, and some interesting and popular questions.

1 Steps

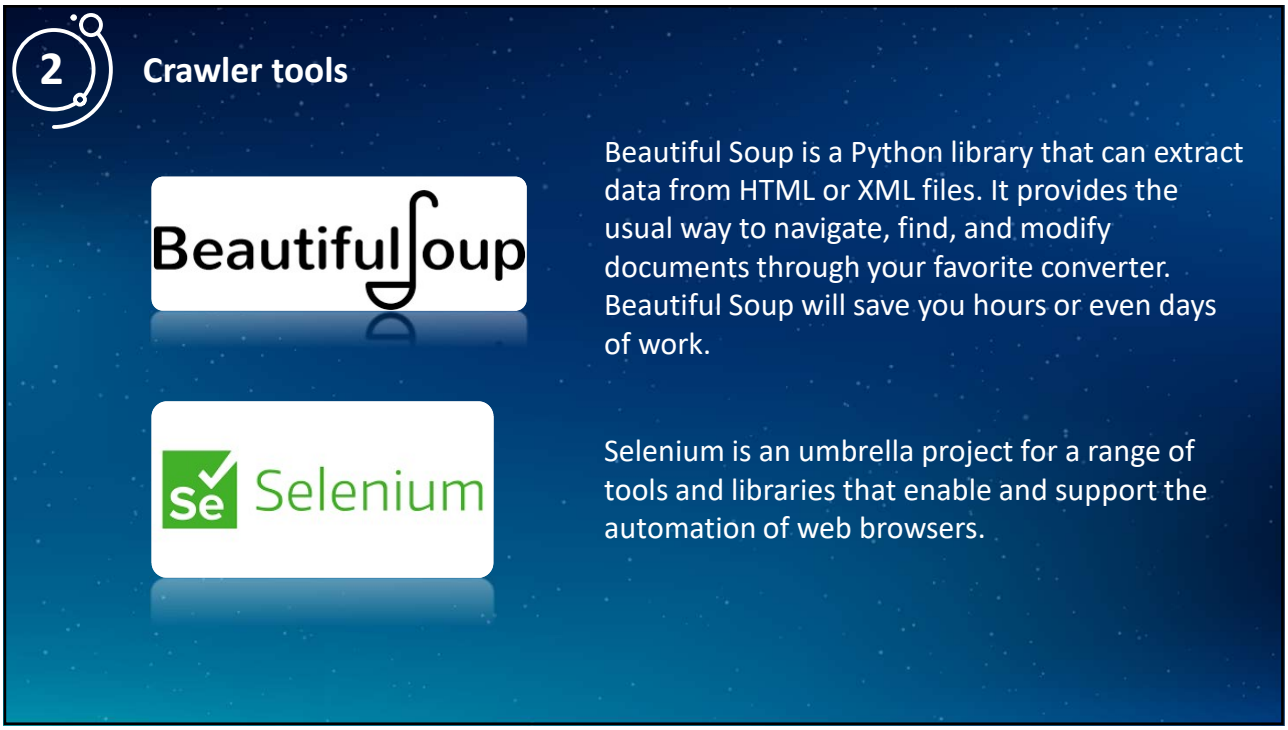
Scraping
crawl hotlist data from Weibo and Zhihu

Generating word clouds
Analyze word clouds horizontally and vertically

1 2 3

keyword extraction
Use two different keyword extraction methods, and compare them

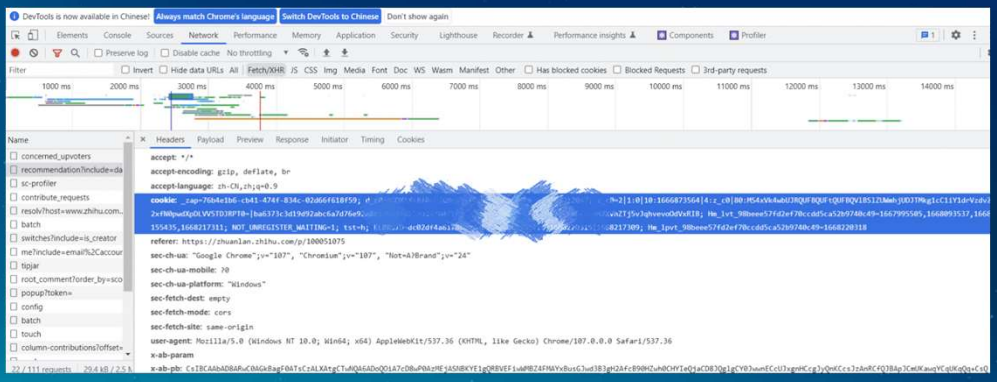

2
Crawl hot list data


2
Crawler tools
Beautiful Soup
Beautiful Soup is a Python library that can extract data from HTML or XML files. It provides the usual way to navigate, find, and modify documents through your favorite converter. Beautiful Soup will save you hours or even days of work.
Selenium
Selenium is an umbrella project for a range of tools and libraries that enable and support the automation of web browsers.

2 Take with Cookies

To bypass permission verification, we need Cookie in the request header

Use the developer tools, find the login status in the web request



2 Process

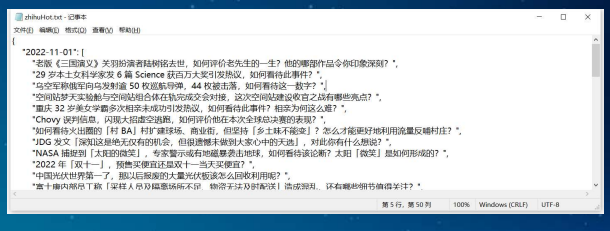
After connecting to the site, we request the site text and use beautifulsoup to clean returned data.

We get the data corresponding to the hotlist tag using API find and find_all.

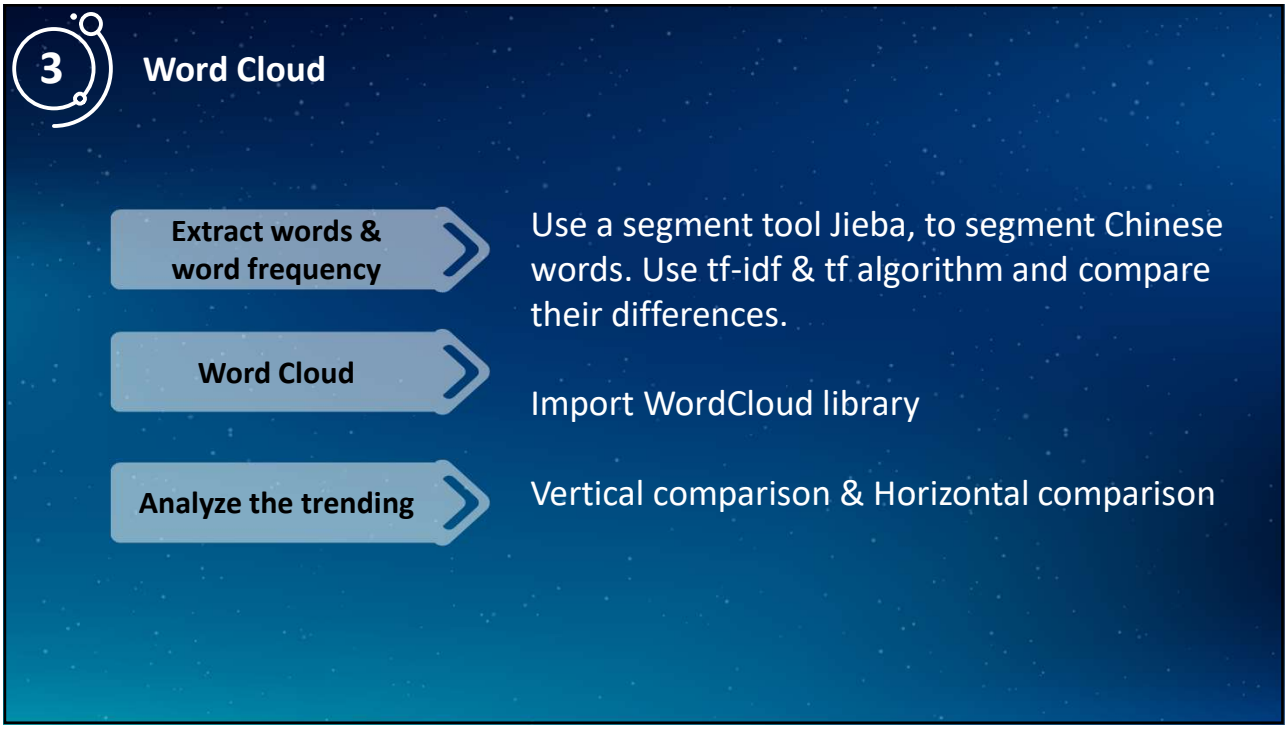
Store it in the dictionary and convert to json.

```
def getZhihu(self):
    self.headers['cookie'] = self.cookie['Zhihu']
    soup = self.getConn("Zhihu")
    hotList = soup.find(attrs={"class": "HotList-list"})
    sections = hotList.findAll("section", limit=50)
    titles = []
    date2title = {}
    for section in sections:
        title = section.find(attrs={"class": "HotItem-title"}).text
        titles.append(title)
    date2title[self.getDate()] = titles
    self.data["Zhihu"] = date2title

def getWeibo(self):
    self.headers["cookie"] = self.cookie["Weibo"]
    soup = self.getConn("Weibo")
    rows = soup.find(class_="data").findAll(class_="td-02", limit=50)
    titles = []
    date2title = {}
    for row in rows:
        title = row.find("a").text
        titles.append(title)
    date2title[self.getDate()] = titles
    self.data["Weibo"] = date2title
```




3
Word Cloud


3
Word Cloud
Extract words & word frequency
Use a segment tool Jieba, to segment Chinese words. Use tf-idf & tf algorithm and compare their differences.
Word Cloud
Import WordCloud library
Analyze the trending
Vertical comparison & Horizontal comparison

3 Compare TF and TFIDF

For keyword extraction, the simplest method must be obtained by word frequency, and TF-IDF is a more advanced method.

When counting word frequency, we know that some meaningless words have very high word frequency. Obviously, these words cannot appear as keywords, so it is necessary to pay attention to word filtering.

Analyze the result

When using Term Frequency algorithm directly, we can see some stop words such as “被”, “的” have high term frequency.

pic.1 wordcloud_tf

pic.2 wordcloud_tfidf

3 Vertical Comparation

To compare the time trend of the hot search keywords, we prepare hot search data of different days as the corpus, and show the keywords of a certain day.

pic.3 wordcloud_10.31

pic.4 wordcloud_11.1

pic.5 wordcloud_11.2

Analyze the result

From the word clouds, Itaewon Stampede was the main focus of social from October 31 to November 1.

3 Horizontal Comparation

Compare the hot search data of Zhihu and Weibo on a given day

pic.1 zhihu 11-02

pic.2 weibo 11-02

Analyze the result

Zhihu has some professional terms in its hot list, while Weibo is more of a daily news

4 Discussion

4 Discussion

Improvements

Beautify word cloud: Add various colors or enrich the pattern of the word clouds

Try more Keyword Extraction Algorithm.
Like TextRank

Create word frequency methods by ourselves.
A comprehensive keyword extraction algorithm can be obtained by weighting the above algorithm