# Financial News Search Engine

[Group 5]

[Frank 谢欣言 1930026136]

[Tony 焦海旭 1930026057]

A report submitted to

Dr. Canhao Xu

for

COMP4123: Information Retrieval and Search Engine (1001)

Data Science (DS)

Division of Science and Technology (DST)

[2022.5.24]

## Contents

1. **Introduction** .......... 3

2. **Data Collection** .......... 3

3. **Search Engine Frontend and Backend:** .......... 10

4. **Conclusion and Reflection** .......... 39

5. **Reference** .......... 41

## 1. Introduction

1) Goal of the project:

The goal of this project is to implement a system that can retrieve search results (text file or structured data) related to the query using Elasticsearch and Python on Ubuntu.

To achieve that, we worked on **data collection**, **search engine frontend and backend** which implemented **all of the 16 required functions**.

2) Background and our motivation:

With the strength of China, our economy has ranked second in the world. In this situation, understanding financial literacy is an urgent need for all of us, including computer students, at the moment. Therefore, we have developed this financial news search engine, which contains the big events of the global financial world in the previous years, covering both Chinese and English, and you can get the latest news in Chinese in real time.

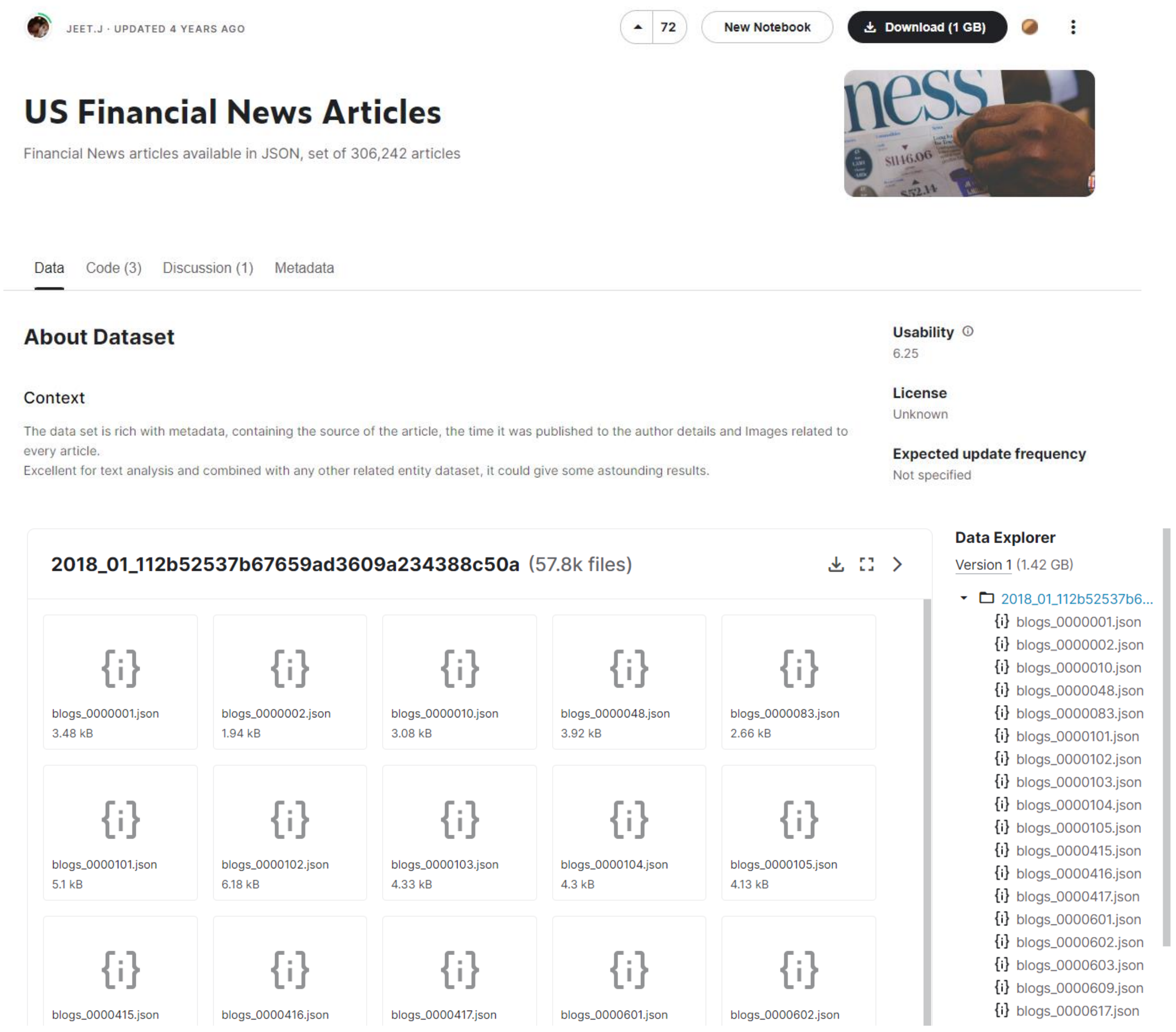
We hope to use this search engine to provide a convenient platform for everyone to learn about finance.

## 2. Data Collection

1) News Dataset:

Our news data resources for normal search (BM25) are collected from **Kaggle**, which are lots of US financial news articles (about 1.9GB unzip).

The URL is: https://www.kaggle.com/datasets/jeet2016/us-financial-news-articles.


JEET.J · UPDATED 4 YEARS AGO

72 New Notebook Download (1 GB)

US Financial News Articles

Financial News articles available in JSON, set of 306,242 articles

Data Code (3) Discussion (1) Metadata

About Dataset

Context

The data set is rich with metadata, containing the source of the article, the time it was published to the author details and Images related to every article.
Excellent for text analysis and combined with any other related entity dataset, it could give some astounding results.

Usability
6.25

License
Unknown

Expected update frequency
Not specified

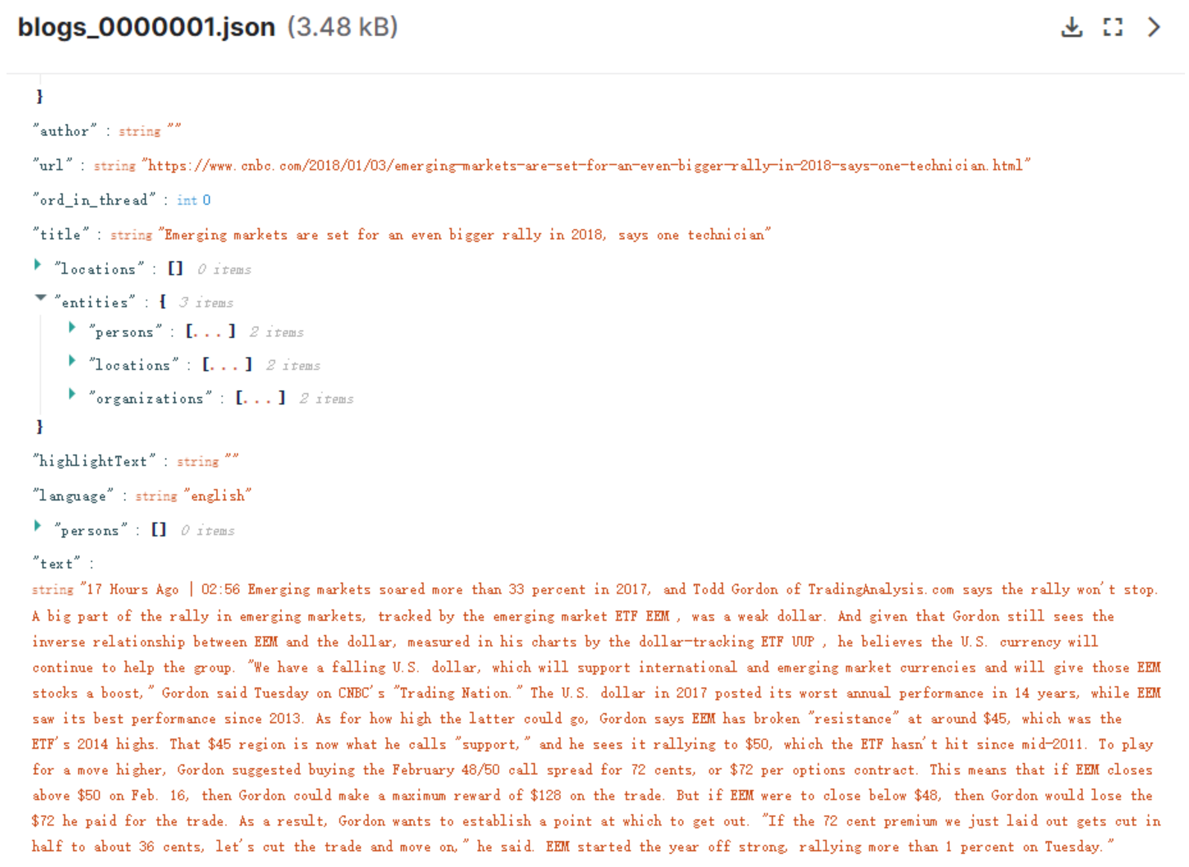
2018_01_112b52537b67659ad3609a234388c50a (57.8k files)

| | | | | |
|---|---|---|---|---|
| blogs_0000001.json 3.48 kB | blogs_0000002.json 1.94 kB | blogs_0000010.json 3.08 kB | blogs_0000048.json 3.92 kB | blogs_0000083.json 2.66 kB |
| blogs_0000101.json 5.1 kB | blogs_0000102.json 6.18 kB | blogs_0000103.json 4.33 kB | blogs_0000104.json 4.3 kB | blogs_0000105.json 4.13 kB |
| blogs_0000415.json | blogs_0000416.json | blogs_0000417.json | blogs_0000601.json | blogs_0000602.json |

Data Explorer
Version 1 (1.42 GB)

2018_01_112b52537b6...
- blogs_0000001.json
- blogs_0000002.json
- blogs_0000010.json
- blogs_0000048.json
- blogs_0000083.json
- blogs_0000101.json
- blogs_0000102.json
- blogs_0000103.json
- blogs_0000104.json
- blogs_0000105.json
- blogs_0000415.json
- blogs_0000416.json
- blogs_0000417.json
- blogs_0000601.json
- blogs_0000602.json
- blogs_0000603.json
- blogs_0000609.json
- blogs_0000617.json

The files in the dataset are **json** files, and the data structure is like following:

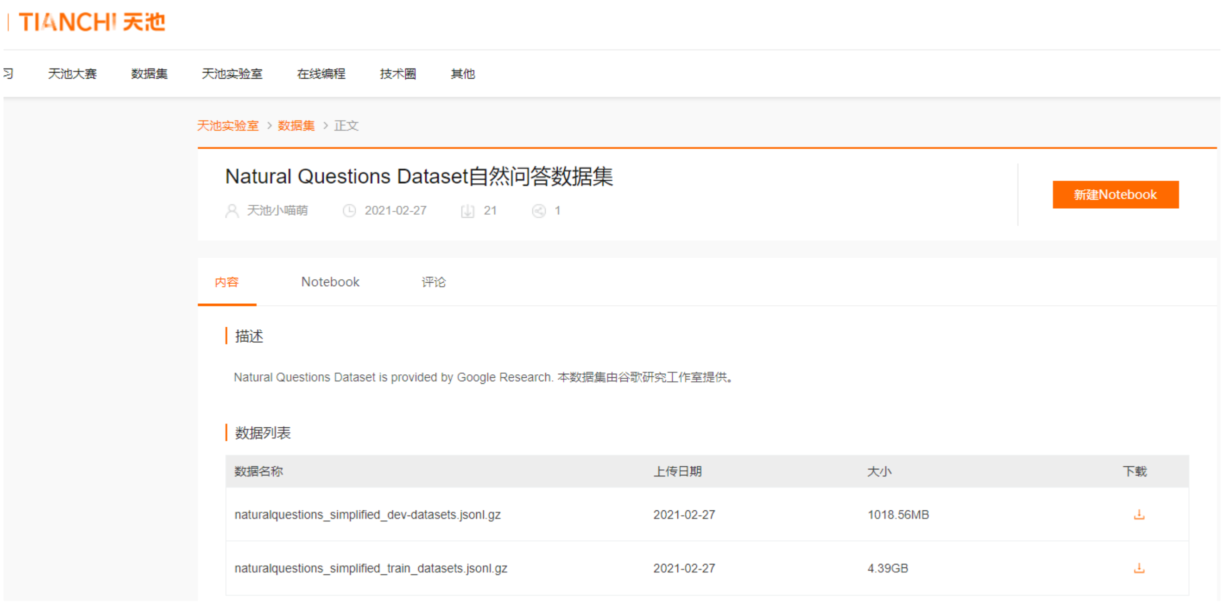

blogs_0000001.json (3.48 kB)

```
}
"author" : string ""
"url" : string "https://www.cnbc.com/2018/01/03/emerging-markets-are-set-for-an-even-bigger-rally-in-2018-says-one-technician.html"
"ord_in_thread" : int 0
"title" : string "Emerging markets are set for an even bigger rally in 2018, says one technician"
▸ "locations" : [] 0 items
▾ "entities" : { 3 items
    ▸ "persons" : [...] 2 items
    ▸ "locations" : [...] 2 items
    ▸ "organizations" : [...] 2 items
}
"highlightText" : string ""
"language" : string "english"
▸ "persons" : [] 0 items
"text" :
string "17 Hours Ago | 02:56 Emerging markets soared more than 33 percent in 2017, and Todd Gordon of TradingAnalysis.com says the rally won't stop. A big part of the rally in emerging markets, tracked by the emerging market ETF EEM , was a weak dollar. And given that Gordon still sees the inverse relationship between EEM and the dollar, measured in his charts by the dollar-tracking ETF UUP , he believes the U.S. currency will continue to help the group. "We have a falling U.S. dollar, which will support international and emerging market currencies and will give those EEM stocks a boost," Gordon said Tuesday on CNBC's "Trading Nation." The U.S. dollar in 2017 posted its worst annual performance in 14 years, while EEM saw its best performance since 2013. As for how high the latter could go, Gordon says EEM has broken "resistance" at around $45, which was the ETF's 2014 highs. That $45 region is now what he calls "support," and he sees it rallying to $50, which the ETF hasn't hit since mid-2011. To play for a move higher, Gordon suggested buying the February 48/50 call spread for 72 cents, or $72 per options contract. This means that if EEM closes above $50 on Feb. 16, then Gordon could make a maximum reward of $128 on the trade. But if EEM were to close below $48, then Gordon would lose the $72 he paid for the trade. As a result, Gordon wants to establish a point at which to get out. "If the 72 cent premium we just laid out gets cut in half to about 36 cents, let's cut the trade and move on," he said. EEM started the year off strong, rallying more than 1 percent on Tuesday."
```

2) QA Dataset:

We also collected a dataset for question answering from **Ali Tianchi** (4.39GB), which contains lots of general questions (including many financial questions that we can search for).

The URL is: https://tianchi.aliyun.com/dataset/dataDetail?dataId=92187.


TIANCHI 天池

习 天池大赛 数据集 天池实验室 在线编程 技术圈 其他

天池实验室 > 数据集 > 正文

Natural Questions Dataset自然问答数据集

天池小喵萌 2021-02-27 21 1

新建Notebook

内容 Notebook 评论

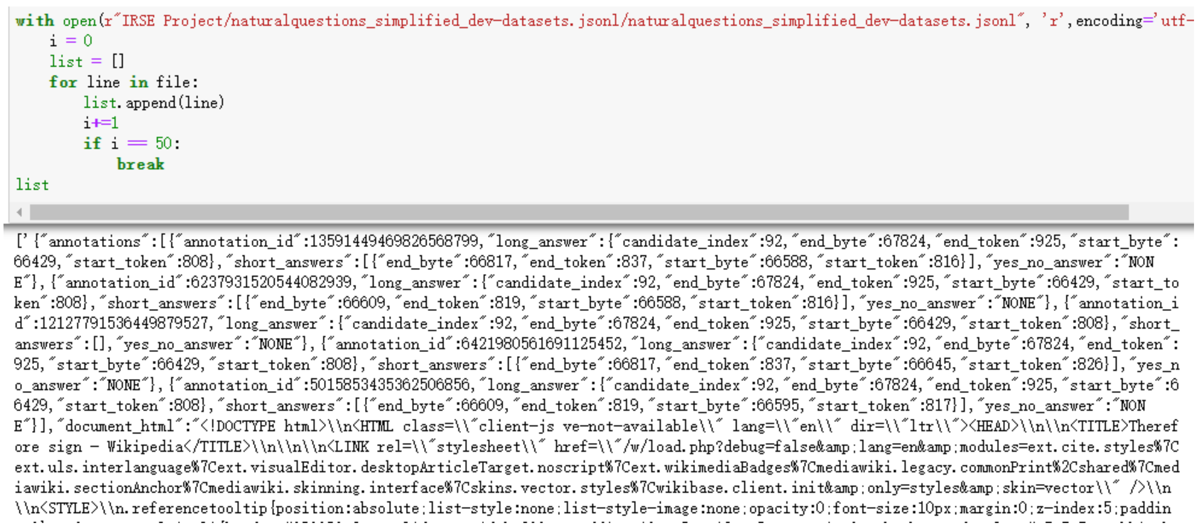
描述

Natural Questions Dataset is provided by Google Research. 本数据集由谷歌研究工作室提供。

数据列表

| 数据名称 | 上传日期 | 大小 | 下载 |
|---|---|---|---|
| naturalquestions_simplified_dev-datasets.jsonl.gz | 2021-02-27 | 1018.56MB | ⤓ |
| naturalquestions_simplified_train_datasets.jsonl.gz | 2021-02-27 | 4.39GB | ⤓ |

The questions and answers are stored in **jsonl** files, the data structure is like following:

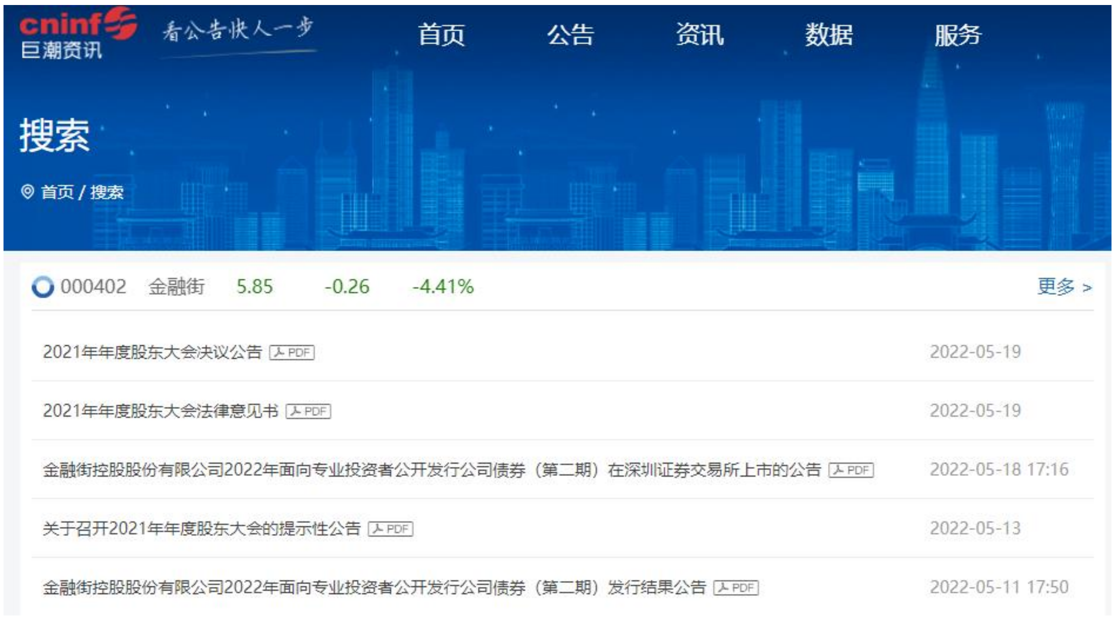

```
with open(r"IRSE Project/naturalquestions_simplified_dev-datasets.jsonl/naturalquestions_simplified_dev-datasets.jsonl", 'r',encoding='utf-
    i = 0
    list = []
    for line in file:
        list.append(line)
        i+=1
        if i == 50:
            break
list
```

```
['{"annotations":[{"annotation_id":13591449469826568799,"long_answer":{"candidate_index":92,"end_byte":67824,"end_token":925,"start_byte":
66429,"start_token":808},"short_answers":[{"end_byte":66817,"end_token":837,"start_byte":66588,"start_token":816}],"yes_no_answer":"NON
E"},{"annotation_id":6237931520544082939,"long_answer":{"candidate_index":92,"end_byte":67824,"end_token":925,"start_byte":66429,"start_to
ken":808},"short_answers":[{"end_byte":66609,"end_token":819,"start_byte":66588,"start_token":816}],"yes_no_answer":"NONE"},{"annotation_i
d":12127791536449879527,"long_answer":{"candidate_index":92,"end_byte":67824,"end_token":925,"start_byte":66429,"start_token":808},"short_
answers":[],"yes_no_answer":"NONE"},{"annotation_id":6421980561691125452,"long_answer":{"candidate_index":92,"end_byte":67824,"end_token":
925,"start_byte":66429,"start_token":808},"short_answers":[{"end_byte":66817,"end_token":837,"start_byte":66645,"start_token":826}],"yes_n
o_answer":"NONE"},{"annotation_id":5015853435362506856,"long_answer":{"candidate_index":92,"end_byte":67824,"end_token":925,"start_byte":6
6429,"start_token":808},"short_answers":[{"end_byte":66609,"end_token":819,"start_byte":66595,"start_token":817}],"yes_no_answer":"NON
E"}],"document_html":"<!DOCTYPE html>\\n<HTML class=\\"client-js ve-not-available\\" lang=\\"en\\" dir=\\"ltr\\"><HEAD>\\n\\n<TITLE>Theref
ore sign - Wikipedia</TITLE>\\n\\n\\n<LINK rel=\\"stylesheet\\" href=\\"/w/load.php?debug=false&amp;lang=en&amp;modules=ext.cite.styles%7C
ext.uls.interlanguage%7Cext.visualEditor.desktopArticleTarget.noscript%7Cext.wikimediaBadges%7Cmediawiki.legacy.commonPrint%2Cshared%7Cmed
iawiki.sectionAnchor%7Cmediawiki.skinning.interface%7Cskins.vector.styles%7Cwikibase.client.init&amp;only=styles&amp;skin=vector\\" />\\n
\\n<STYLE>\\n.referencetooltip{position:absolute;list-style:none;list-style-image:none;opacity:0;font-size:10px;margin:0;z-index:5;paddin
```

Because the data structure is too complex and there is too much unrelated information, we split the large **jsonl** file, only selected the questions and answers and stored them in **csv** files.

3) Crawled Dataset:

We also have crawled some data for real-time news in Chinese from **CNINF**: http://finance.sina.com.cn/realstock/company/sh000001/nc.shtml.



000402 金融街 5.85 -0.26 -4.41% 更多 >

| | |
|---|---|
| 2021年年度股东大会决议公告 PDF | 2022-05-19 |
| 2021年年度股东大会法律意见书 PDF | 2022-05-19 |
| 金融街控股股份有限公司2022年面向专业投资者公开发行公司债券（第二期）在深圳证券交易所上市的公告 PDF | 2022-05-18 17:16 |
| 关于召开2021年年度股东大会的提示性公告 PDF | 2022-05-13 |
| 金融街控股股份有限公司2022年面向专业投资者公开发行公司债券（第二期）发行结果公告 PDF | 2022-05-11 17:50 |

4) Data in Other Types:

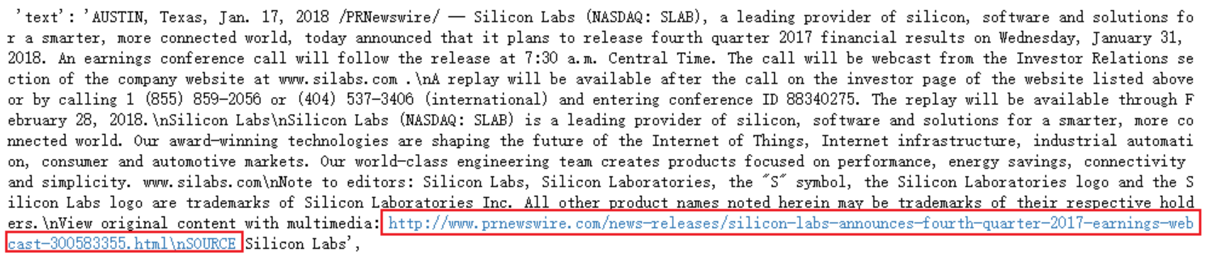
To support searching for multiple types of data, we also have a small amount of data in other types:

- **txt, doc** and **pdf**: from our news dataset from **Kaggle**, we copy some of them and stored them into files.
- **Image**: our images are from **Baidu** pictures.
- **Video**: our videos are from **Bilibili**.

5) Data preprocessing:

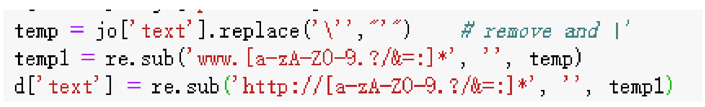
For the news dataset, because there are many unexpected characters such as “\” and URLs which may affect the result of BM25 search, we use “re” package to remove and clean them.

Before cleaning:

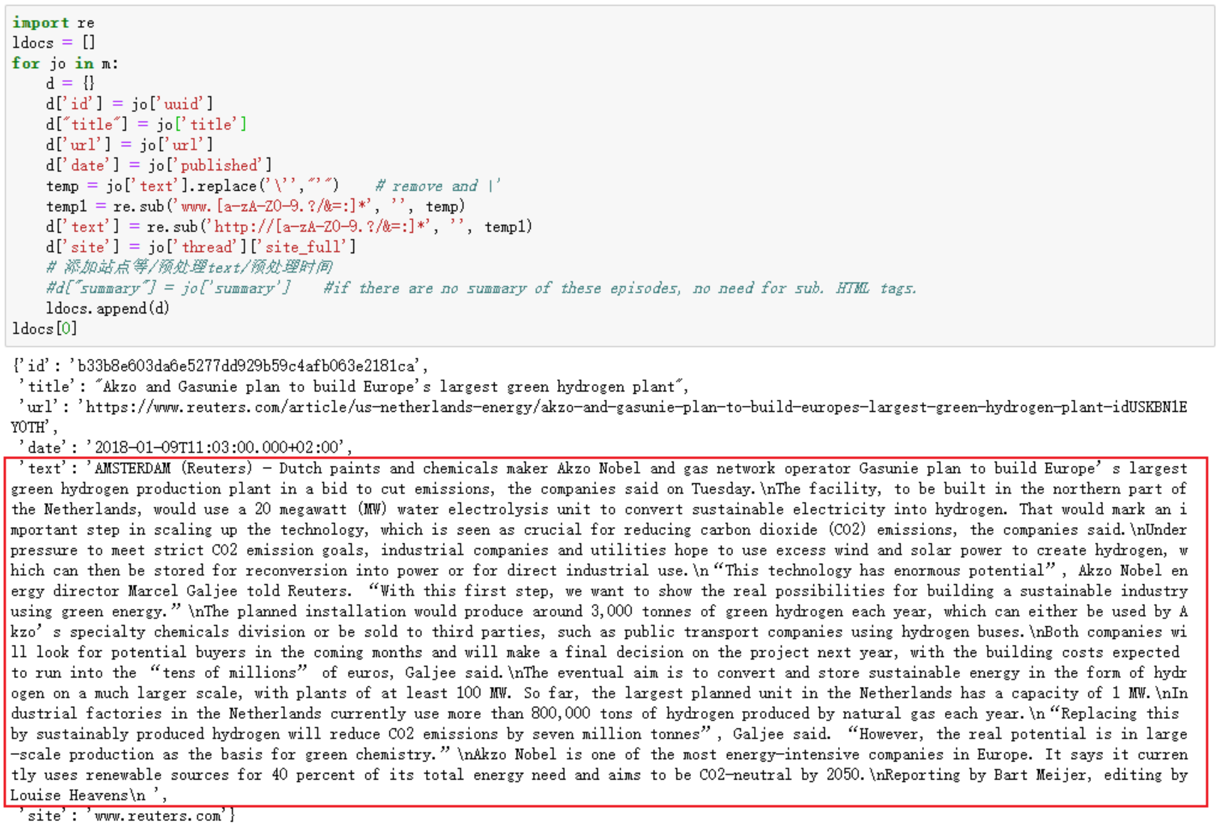


```
 'text': 'AUSTIN, Texas, Jan. 17, 2018 /PRNewswire/ — Silicon Labs (NASDAQ: SLAB), a leading provider of silicon, software and solutions fo
r a smarter, more connected world, today announced that it plans to release fourth quarter 2017 financial results on Wednesday, January 31,
2018. An earnings conference call will follow the release at 7:30 a.m. Central Time. The call will be webcast from the Investor Relations se
ction of the company website at www.silabs.com .\nA replay will be available after the call on the investor page of the website listed above
or by calling 1 (855) 859-2056 or (404) 537-3406 (international) and entering conference ID 88340275. The replay will be available through F
ebruary 28, 2018.\nSilicon Labs\nSilicon Labs (NASDAQ: SLAB) is a leading provider of silicon, software and solutions for a smarter, more co
nnected world. Our award-winning technologies are shaping the future of the Internet of Things, Internet infrastructure, industrial automati
on, consumer and automotive markets. Our world-class engineering team creates products focused on performance, energy savings, connectivity
and simplicity. www.silabs.com\nNote to editors: Silicon Labs, Silicon Laboratories, the "S" symbol, the Silicon Laboratories logo and the S
ilicon Labs logo are trademarks of Silicon Laboratories Inc. All other product names noted herein may be trademarks of their respective hold
ers.\nView original content with multimedia: http://www.prnewswire.com/news-releases/silicon-labs-announces-fourth-quarter-2017-earnings-web
cast-300583355.html\nSOURCE Silicon Labs',
```

After cleaning:



```
temp = jo['text'].replace('\'',"'")     # remove and |'
temp1 = re.sub('www.[a-zA-Z0-9.?/&=:]*', '', temp)
d['text'] = re.sub('http://[a-zA-Z0-9.?/&=:]*', '', temp1)
```

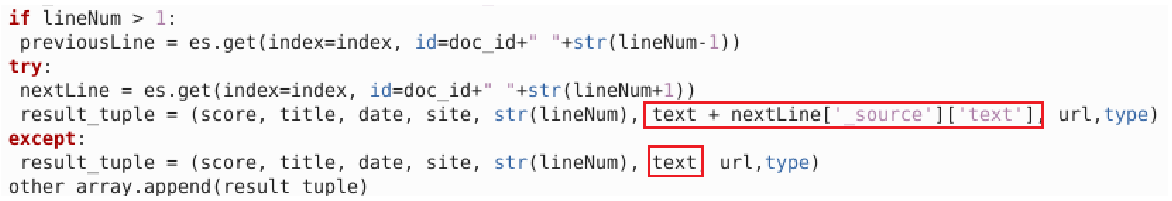


```
import re
ldocs = []
for jo in m:
    d = {}
    d['id'] = jo['uuid']
    d["title"] = jo['title']
    d['url'] = jo['url']
    d['date'] = jo['published']
    temp = jo['text'].replace('\'',"'")     # remove and |'
    temp1 = re.sub('www.[a-zA-Z0-9.?/&=:]*', '', temp)
    d['text'] = re.sub('http://[a-zA-Z0-9.?/&=:]*', '', temp1)
    d['site'] = jo['thread']['site_full']
    # 添加站点等/预处理text/预处理时间
    #d["summary"] = jo['summary']     #if there are no summary of these episodes, no need for sub. HTML tags.
    ldocs.append(d)
ldocs[0]
```

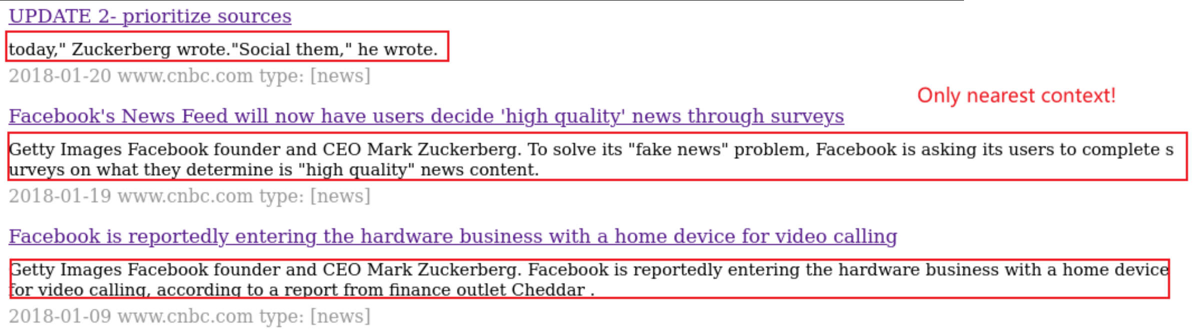
```
{'id': 'b33b8e603da6e5277dd929b59c4afb063e2181ca',
 'title': "Akzo and Gasunie plan to build Europe's largest green hydrogen plant",
 'url': 'https://www.reuters.com/article/us-netherlands-energy/akzo-and-gasunie-plan-to-build-europes-largest-green-hydrogen-plant-idUSKBN1E
Y0TH',
 'date': '2018-01-09T11:03:00.000+02:00',
 'text': 'AMSTERDAM (Reuters) - Dutch paints and chemicals maker Akzo Nobel and gas network operator Gasunie plan to build Europe' s largest
green hydrogen production plant in a bid to cut emissions, the companies said on Tuesday.\nThe facility, to be built in the northern part of
the Netherlands, would use a 20 megawatt (MW) water electrolysis unit to convert sustainable electricity into hydrogen. That would mark an i
mportant step in scaling up the technology, which is seen as crucial for reducing carbon dioxide (CO2) emissions, the companies said.\nUnder
pressure to meet strict CO2 emission goals, industrial companies and utilities hope to use excess wind and solar power to create hydrogen, w
hich can then be stored for reconversion into power or for direct industrial use.\n"This technology has enormous potential", Akzo Nobel en
ergy director Marcel Galjee told Reuters. "With this first step, we want to show the real possibilities for building a sustainable industry
using green energy."\nThe planned installation would produce around 3,000 tonnes of green hydrogen each year, which can either be used by A
kzo' s specialty chemicals division or be sold to third parties, such as public transport companies using hydrogen buses.\nBoth companies wi
ll look for potential buyers in the coming months and will make a final decision on the project next year, with the building costs expected
to run into the "tens of millions" of euros, Galjee said.\nThe eventual aim is to convert and store sustainable energy in the form of hydr
ogen on a much larger scale, with plants of at least 100 MW. So far, the largest planned unit in the Netherlands has a capacity of 1 MW.\nIn
dustrial factories in the Netherlands currently use more than 800,000 tons of hydrogen produced by natural gas each year.\n"Replacing this
by sustainably produced hydrogen will reduce CO2 emissions by seven million tonnes", Galjee said. "However, the real potential is in large
-scale production as the basis for green chemistry."\nAkzo Nobel is one of the most energy-intensive companies in Europe. It says it curren
tly uses renewable sources for 40 percent of its total energy need and aims to be CO2-neutral by 2050.\nReporting by Bart Meijer, editing by
Louise Heavens\n ',
 'site': 'www.reuters.com'}
```

6) Index the Data:

- For news dataset, we only selected **"title", "url", "date", "lineNum", "text", "site"** and **"type"** to put into our index. And we only index one line each time, because by this method we can get the "nearest context" of the search term, which means the last and next sentences of the sentences we wanted:



```
if lineNum > 1:
 previousLine = es.get(index=index, id=doc_id+" "+str(lineNum-1))
try:
 nextLine = es.get(index=index, id=doc_id+" "+str(lineNum+1))
 result_tuple = (score, title, date, site, str(lineNum), text + nextLine['_source']['text'], url,type)
except:
 result_tuple = (score, title, date, site, str(lineNum), text  url,type)
other_array.append(result_tuple)
```



UPDATE 2- prioritize sources

today," Zuckerberg wrote."Social them," he wrote.

2018-01-20 www.cnbc.com type: [news]

Only nearest context!

Facebook's News Feed will now have users decide 'high quality' news through surveys

Getty Images Facebook founder and CEO Mark Zuckerberg. To solve its "fake news" problem, Facebook is asking its users to complete surveys on what they determine is "high quality" news content.

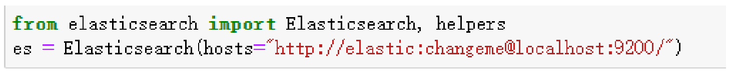
2018-01-19 www.cnbc.com type: [news]

Facebook is reportedly entering the hardware business with a home device for video calling

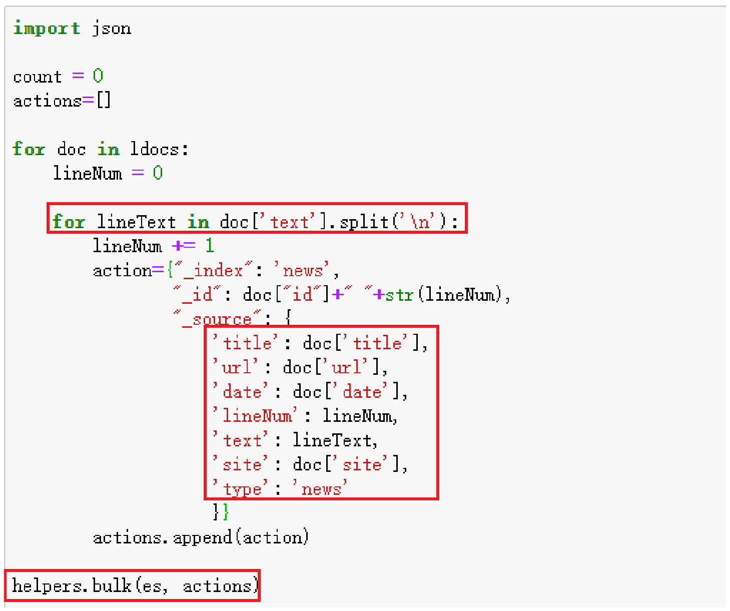
Getty Images Facebook founder and CEO Mark Zuckerberg. Facebook is reportedly entering the hardware business with a home device for video calling, according to a report from finance outlet Cheddar .

2018-01-09 www.cnbc.com type: [news]

And here we use **bulk** for efficient indexing:

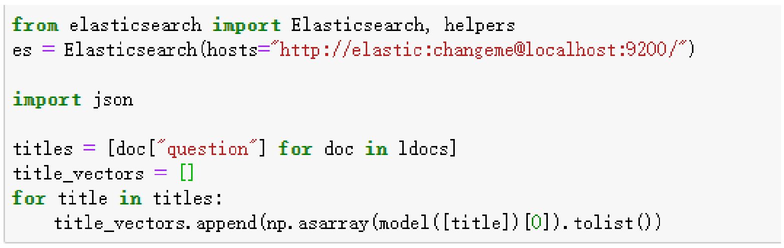


```
from elasticsearch import Elasticsearch, helpers
es = Elasticsearch(hosts="http://elastic:changeme@localhost:9200/")
```

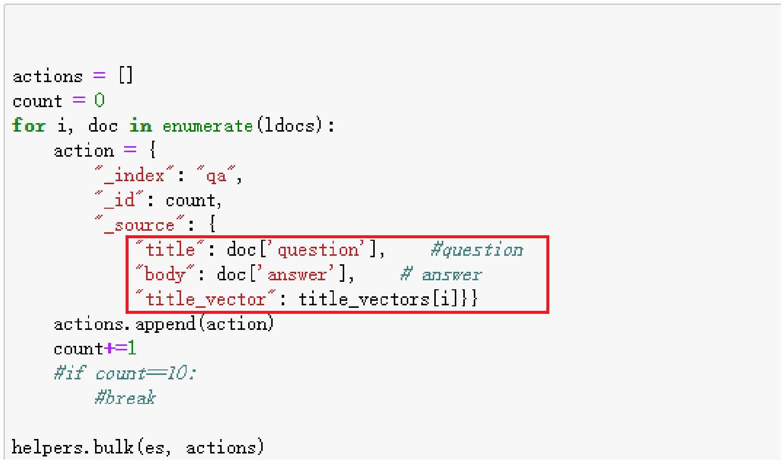


```
import json

count = 0
actions=[]

for doc in ldocs:
    lineNum = 0

    for lineText in doc['text'].split('\n'):
        lineNum += 1
        action={"_index": 'news',
                "_id": doc["id"]+" "+str(lineNum),
                "_source": {
                    'title': doc['title'],
                    'url': doc['url'],
                    'date': doc['date'],
                    'lineNum': lineNum,
                    'text': lineText,
                    'site': doc['site'],
                    'type': 'news'
                    }}
        actions.append(action)

helpers.bulk(es, actions)
```

- Then, for QA dataset, we only selected **"id", "question", "answer"** and the **"title_vector"**, which are embeddings transformed by **USE** model:



```python
from elasticsearch import Elasticsearch, helpers
es = Elasticsearch(hosts="http://elastic:changeme@localhost:9200/")

import json

titles = [doc["question"] for doc in ldocs]
title_vectors = []
for title in titles:
    title_vectors.append(np.asarray(model([title])[0]).tolist())
```



```python
actions = []
count = 0
for i, doc in enumerate(ldocs):
    action = {
        "_index": "qa",
        "_id": count,
        "_source": {
            "title": doc['question'],     #question
            "body": doc['answer'],      # answer
            "title_vector": title_vectors[i]}}
    actions.append(action)
    count+=1
    #if count==10:
        #break

helpers.bulk(es, actions)
```

## 3. Search Engine Frontend and Backend:

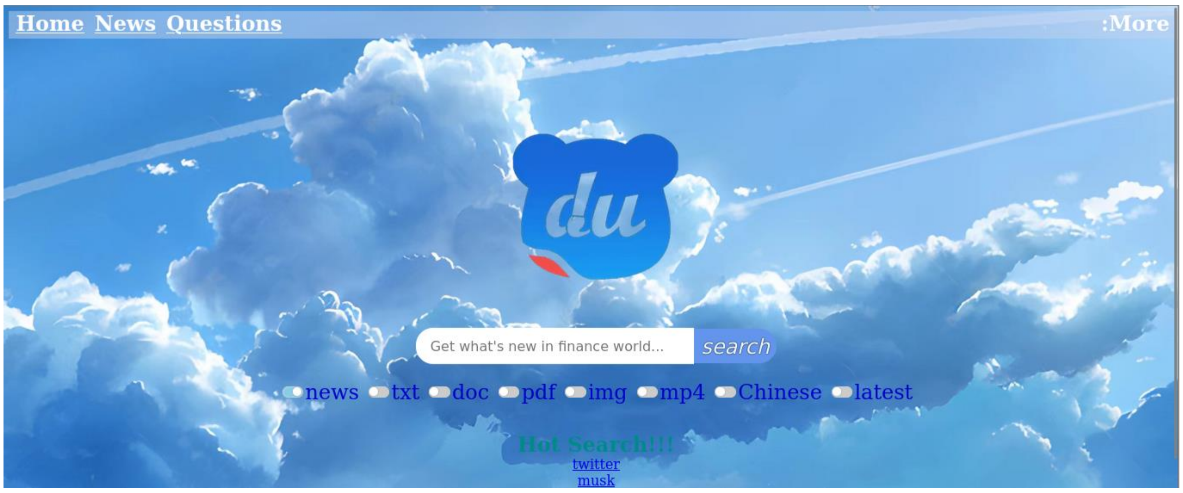
1) Frontend: Web Interface:

We have 4 web pages in total: **index.html** for home page, **search.html** for normal news search, **QA.html** for questions answering and **backend.html** for monitoring users' search terms in real time:

■ Index.html:

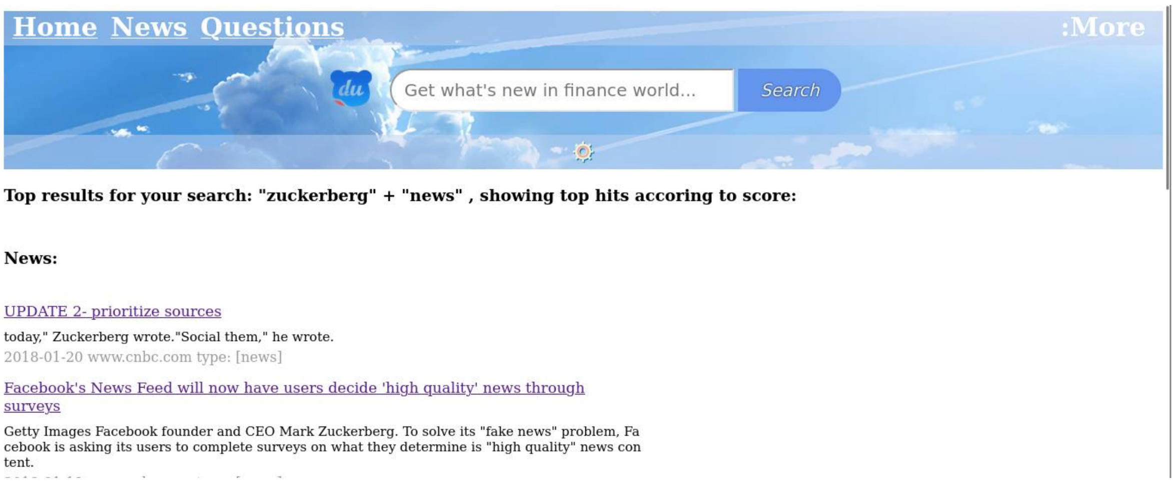
A big logo and basic search settings for users to search for different types of files.

Besides, there are hot search terms at the bottom (links) and users can search for them by clicking them directly:


Home News Questions
:More
du
Get what's new in finance world...
search
news txt doc pdf img mp4 Chinese latest
Hot Search!!!
twitter
musk

- Seach.html

  For normal news search. Users can customize their search (different types of files and sort by scores/date/category) by clicking the "settings" button under the search input bar:



**Top results for your search: "zuckerberg" + "news" , showing top hits accoring to score:**

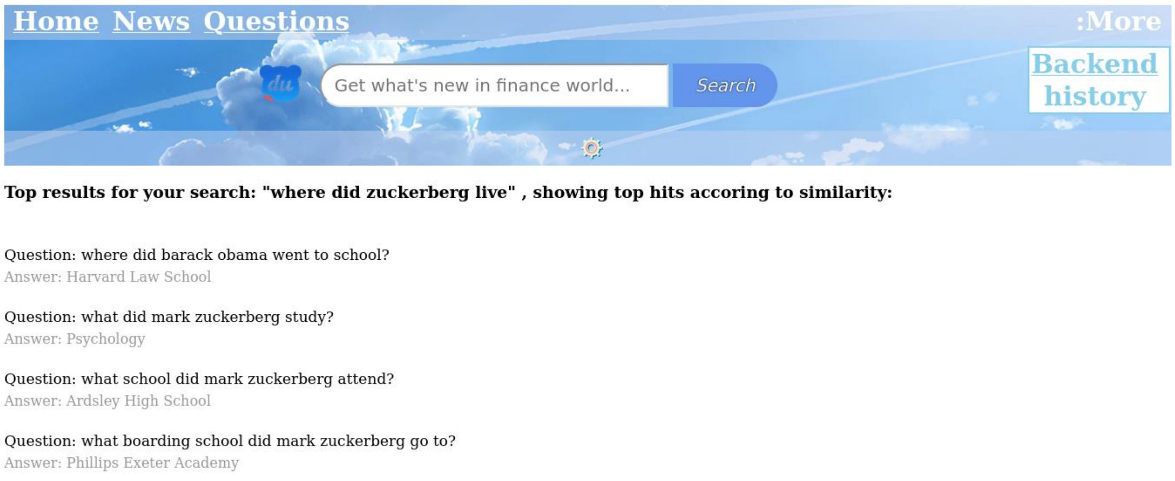
**News:**

UPDATE 2- prioritize sources
today," Zuckerberg wrote."Social them," he wrote.
2018-01-20 www.cnbc.com type: [news]

Facebook's News Feed will now have users decide 'high quality' news through surveys
Getty Images Facebook founder and CEO Mark Zuckerberg. To solve its "fake news" problem, Fa cebook is asking its users to complete surveys on what they determine is "high quality" news con tent.

- QA.html

  For searching for answers for questions related in finance field:



**Top results for your search: "where did zuckerberg live" , showing top hits accoring to similarity:**

Question: where did barack obama went to school?
Answer: Harvard Law School

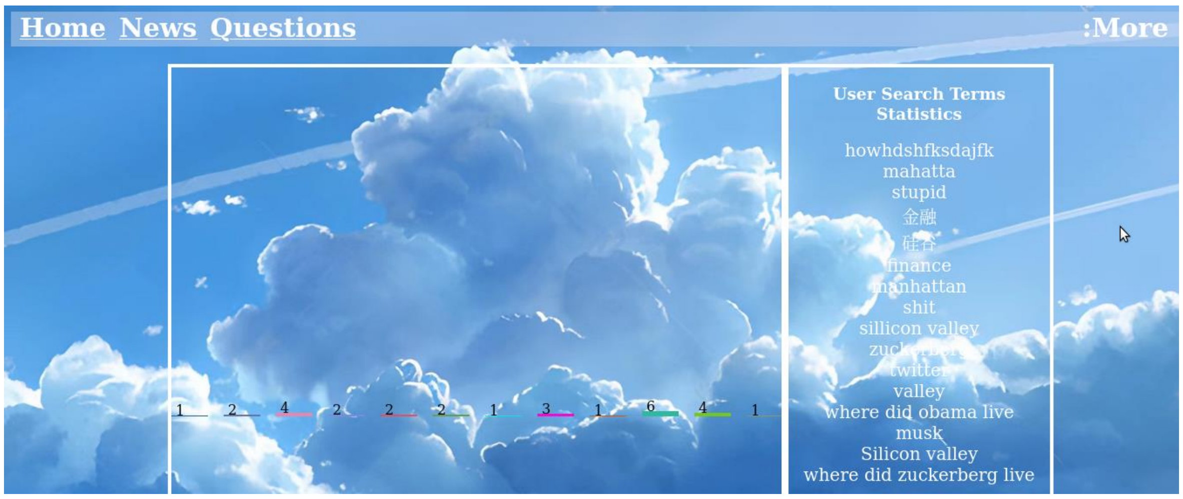
Question: what did mark zuckerberg study?
Answer: Psychology

Question: what school did mark zuckerberg attend?
Answer: Ardsley High School

Question: what boarding school did mark zuckerberg go to?
Answer: Phillips Exeter Academy

- Backend.html:

  A backend interface for website administrators to visualize what users are searching and get the statistics of search terms in histograms:

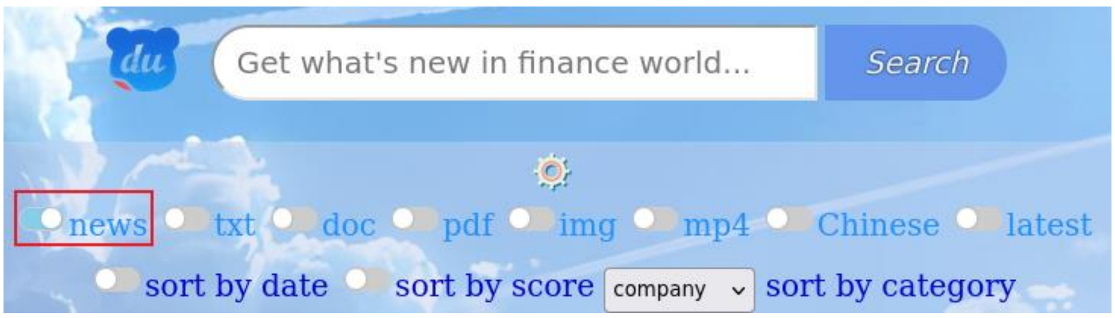


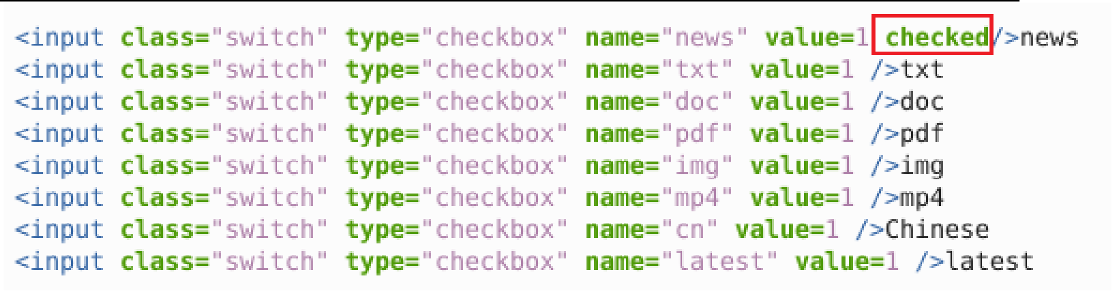
**User Search Terms Statistics**

howhdshfksdajfk
mahatta
stupid
金融
硅谷
finance
manhattan
shit
sillicon valley
zuckerberg
twitter
valley
where did obama live
musk
Silicon valley
where did zuckerberg live

2) Backend: Basic Functions:

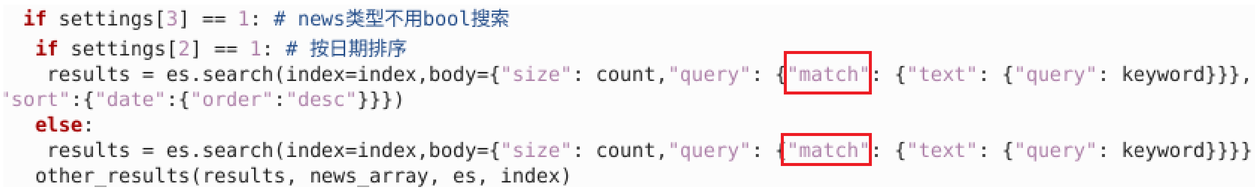
In our search engine, the default search filed is **"news"** ("news" option in settings is checked by default), which are the financial news in our **news** index:





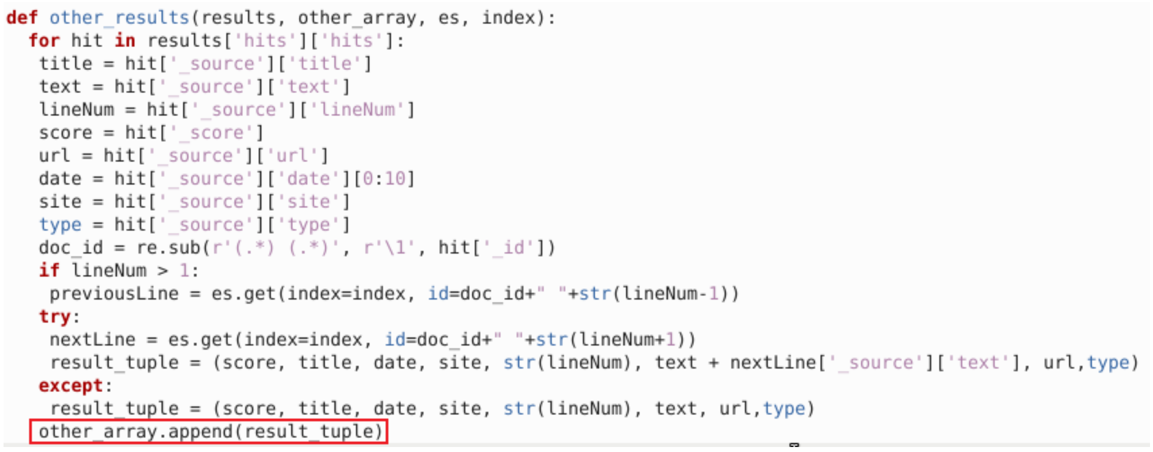
```
<input class="switch" type="checkbox" name="news" value=1 checked/>news
<input class="switch" type="checkbox" name="txt" value=1 />txt
<input class="switch" type="checkbox" name="doc" value=1 />doc
<input class="switch" type="checkbox" name="pdf" value=1 />pdf
<input class="switch" type="checkbox" name="img" value=1 />img
<input class="switch" type="checkbox" name="mp4" value=1 />mp4
<input class="switch" type="checkbox" name="cn" value=1 />Chinese
<input class="switch" type="checkbox" name="latest" value=1 />latest
```

For the basic search function, we use the default **BM25** scoring method and the **“match”** method to search for news:



```
 if settings[3] == 1: # news类型不用bool搜索
  if settings[2] == 1: # 按日期排序
   results = es.search(index=index,body={"size": count,"query": {"match": {"text": {"query": keyword}}},
'sort":{"date":{"order":"desc"}}})
  else:
   results = es.search(index=index,body={"size": count,"query": {"match": {"text": {"query": keyword}}}}
  other_results(results, news_array, es, index)
```

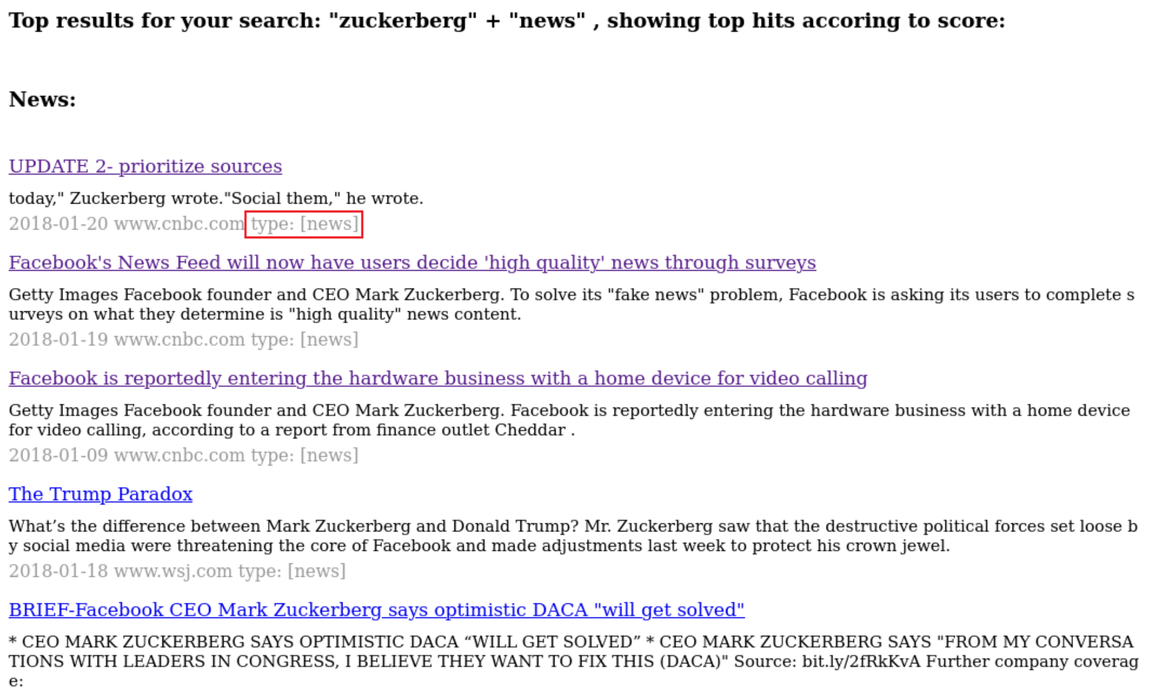
Then call the **“other_results()”** function to read and store the results:



```
def other_results(results, other_array, es, index):
  for hit in results['hits']['hits']:
   title = hit['_source']['title']
   text = hit['_source']['text']
   lineNum = hit['_source']['lineNum']
   score = hit['_score']
   url = hit['_source']['url']
   date = hit['_source']['date'][0:10]
   site = hit['_source']['site']
   type = hit['_source']['type']
   doc_id = re.sub(r'(.*) (.*)', r'\1', hit['_id'])
   if lineNum > 1:
    previousLine = es.get(index=index, id=doc_id+" "+str(lineNum-1))
   try:
    nextLine = es.get(index=index, id=doc_id+" "+str(lineNum+1))
    result_tuple = (score, title, date, site, str(lineNum), text + nextLine['_source']['text'], url,type)
   except:
    result_tuple = (score, title, date, site, str(lineNum), text, url,type)
   other_array.append(result_tuple)
```

And send the results in array to **“views.py”**, then to the **html**.

The news results in **html** will be shown like following, we arrange the results just like in Baidu or Google, users can get news **title** with **URL**, the “**nearest context**”, **date**, **site** and the **type**:



**Top results for your search: "zuckerberg" + "news" , showing top hits accoring to score:**

**News:**

UPDATE 2- prioritize sources

today," Zuckerberg wrote."Social them," he wrote.
2018-01-20 www.cnbc.com type: [news]

Facebook's News Feed will now have users decide 'high quality' news through surveys

Getty Images Facebook founder and CEO Mark Zuckerberg. To solve its "fake news" problem, Facebook is asking its users to complete s urveys on what they determine is "high quality" news content.
2018-01-19 www.cnbc.com type: [news]

Facebook is reportedly entering the hardware business with a home device for video calling

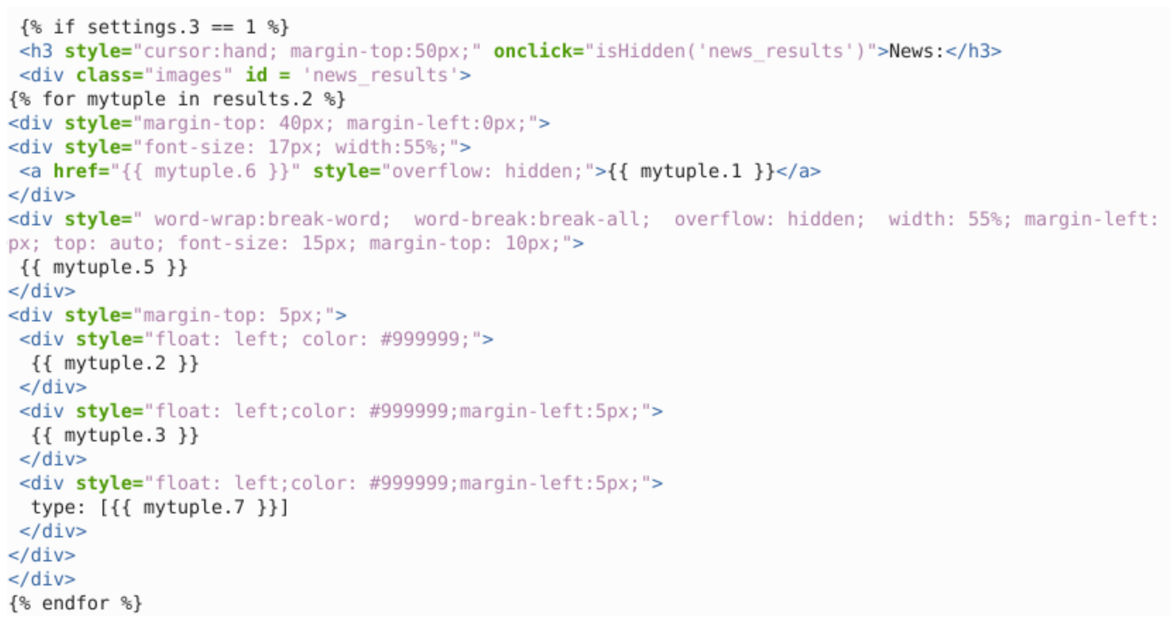
Getty Images Facebook founder and CEO Mark Zuckerberg. Facebook is reportedly entering the hardware business with a home device for video calling, according to a report from finance outlet Cheddar .
2018-01-09 www.cnbc.com type: [news]

The Trump Paradox

What’s the difference between Mark Zuckerberg and Donald Trump? Mr. Zuckerberg saw that the destructive political forces set loose b y social media were threatening the core of Facebook and made adjustments last week to protect his crown jewel.
2018-01-18 www.wsj.com type: [news]

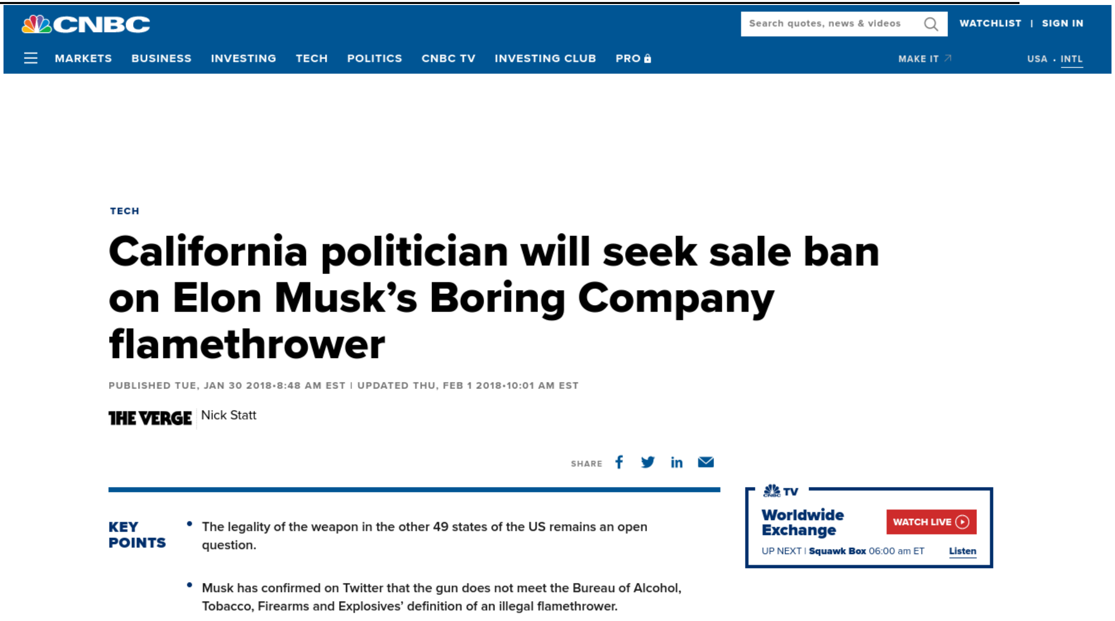
BRIEF-Facebook CEO Mark Zuckerberg says optimistic DACA "will get solved"

* CEO MARK ZUCKERBERG SAYS OPTIMISTIC DACA “WILL GET SOLVED” * CEO MARK ZUCKERBERG SAYS "FROM MY CONVERSA TIONS WITH LEADERS IN CONGRESS, I BELIEVE THEY WANT TO FIX THIS (DACA)" Source: bit.ly/2fRkKvA Further company coverag e:



```
 {% if settings.3 == 1 %}
 <h3 style="cursor:hand; margin-top:50px;" onclick="isHidden('news_results')">News:</h3>
 <div class="images" id = 'news_results'>
{% for mytuple in results.2 %}
<div style="margin-top: 40px; margin-left:0px;">
<div style="font-size: 17px; width:55%;">
 <a href="{{ mytuple.6 }}" style="overflow: hidden;">{{ mytuple.1 }}</a>
</div>
<div style=" word-wrap:break-word;  word-break:break-all;  overflow: hidden;  width: 55%; margin-left:
px; top: auto; font-size: 15px; margin-top: 10px;">
 {{ mytuple.5 }}
</div>
<div style="margin-top: 5px;">
 <div style="float: left; color: #999999;">
  {{ mytuple.2 }}
 </div>
 <div style="float: left;color: #999999;margin-left:5px;">
  {{ mytuple.3 }}
 </div>
 <div style="float: left;color: #999999;margin-left:5px;">
  type: [{{ mytuple.7 }}]
 </div>
</div>
</div>
{% endfor %}
```

By clicking the title with URL, we can get to the website containing the whole context of that news:


CNBC
Search quotes, news & videos
WATCHLIST | SIGN IN
MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV INVESTING CLUB PRO
MAKE IT
USA · INTL
SHARE
TV
Worldwide Exchange
WATCH LIVE
UP NEXT | Squawk Box 06:00 am ET
Listen

TECH

California politician will seek sale ban on Elon Musk's Boring Company flamethrower

PUBLISHED TUE, JAN 30 2018·8:48 AM EST | UPDATED THU, FEB 1 2018·10:01 AM EST

THE VERGE Nick Statt

KEY POINTS

- The legality of the weapon in the other 49 states of the US remains an open question.
- Musk has confirmed on Twitter that the gun does not meet the Bureau of Alcohol, Tobacco, Firearms and Explosives' definition of an illegal flamethrower.

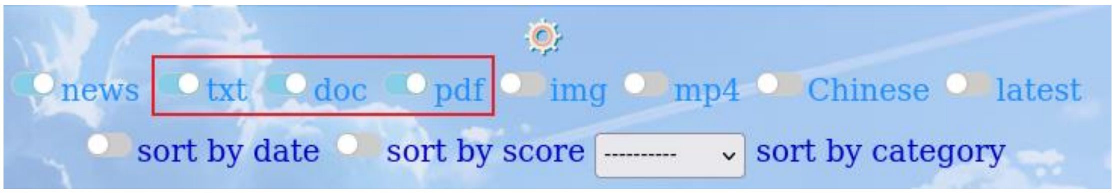
3) Backend: 16 Extra Functions:

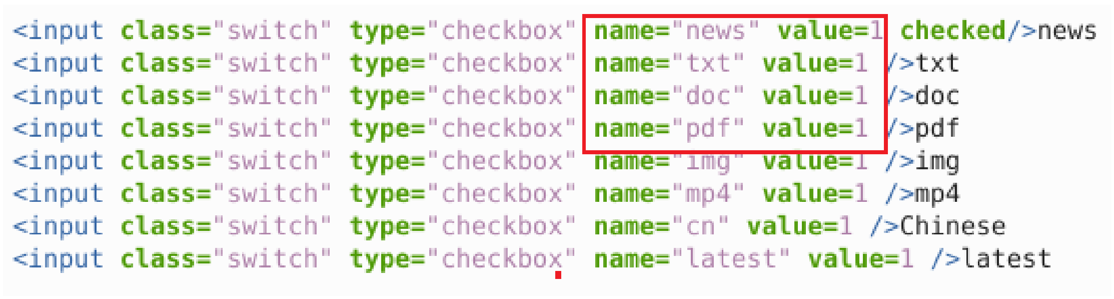
① Search/index Different Types of Files, e.g. pdf/doc:

Our search engine support searching for **doc**, **txt** and **pdf** files. To implement that:

■ In frontend:

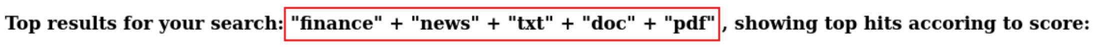
We add an area for "**settings**" below the search bar, and each selection will send the variable indicating to the type of files with value **"1"** to the backend:



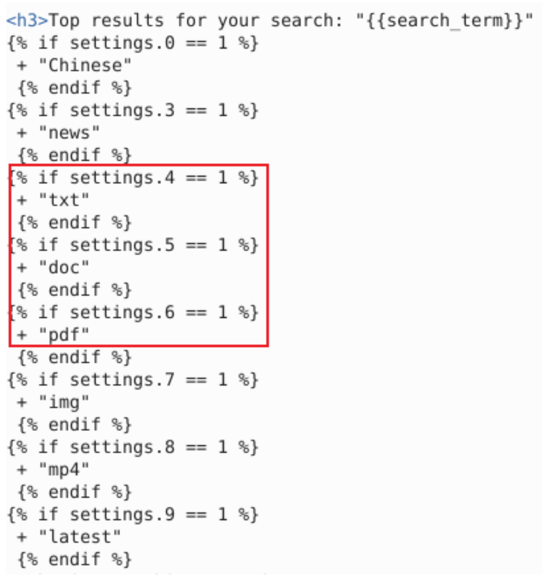


```
<input class="switch" type="checkbox" name="news" value=1 checked/>news
<input class="switch" type="checkbox" name="txt" value=1 />txt
<input class="switch" type="checkbox" name="doc" value=1 />doc
<input class="switch" type="checkbox" name="pdf" value=1 />pdf
<input class="switch" type="checkbox" name="img" value=1 />img
<input class="switch" type="checkbox" name="mp4" value=1 />mp4
<input class="switch" type="checkbox" name="cn" value=1 />Chinese
<input class="switch" type="checkbox" name="latest" value=1 />latest
```

While getting results from the backend, the hint message will show the combination of the settings of users:

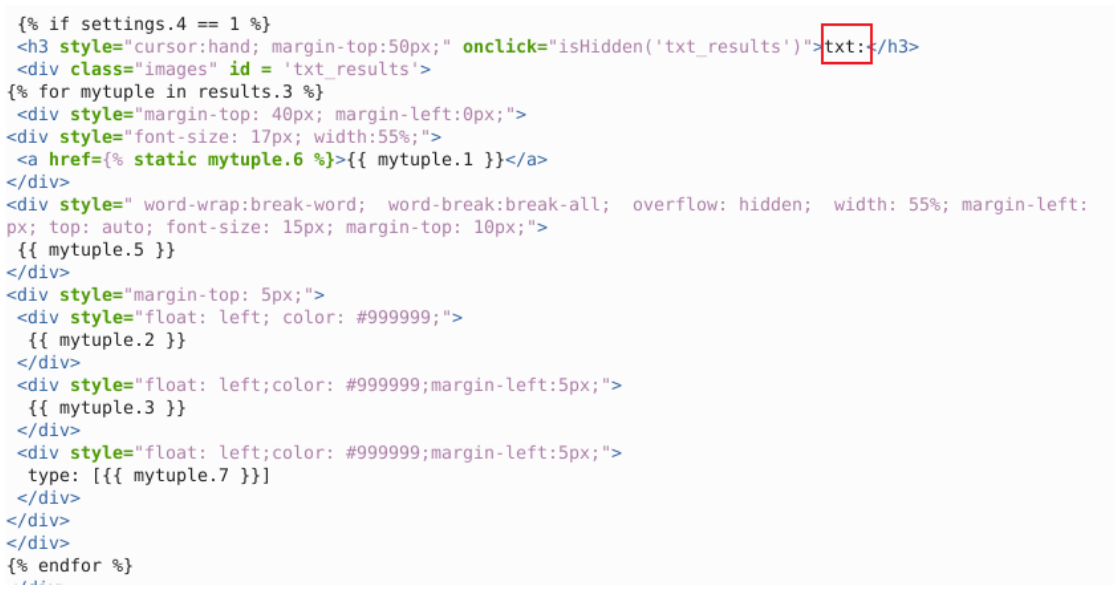


**Top results for your search: "finance" + "news" + "txt" + "doc" + "pdf" , showing top hits accoring to score:**



```
<h3>Top results for your search: "{{search_term}}"
{% if settings.0 == 1 %}
 + "Chinese"
 {% endif %}
{% if settings.3 == 1 %}
 + "news"
 {% endif %}
{% if settings.4 == 1 %}
 + "txt"
 {% endif %}
{% if settings.5 == 1 %}
 + "doc"
 {% endif %}
{% if settings.6 == 1 %}
 + "pdf"
 {% endif %}
{% if settings.7 == 1 %}
 + "img"
 {% endif %}
{% if settings.8 == 1 %}
 + "mp4"
 {% endif %}
{% if settings.9 == 1 %}
 + "latest"
 {% endif %}
```

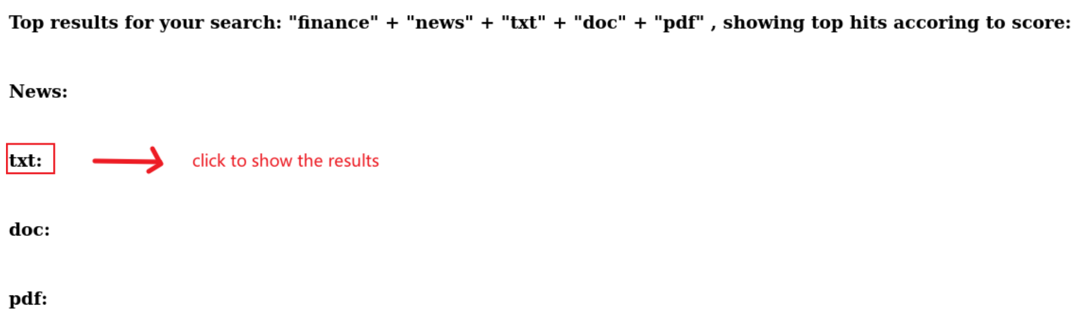
And for each type of files, we have a special arrangement for them:



```
 {% if settings.4 == 1 %}
 <h3 style="cursor:hand; margin-top:50px;" onclick="isHidden('txt_results')">txt:</h3>
 <div class="images" id = 'txt_results'>
{% for mytuple in results.3 %}
 <div style="margin-top: 40px; margin-left:0px;">
<div style="font-size: 17px; width:55%;">
 <a href={% static mytuple.6 %}>{{ mytuple.1 }}</a>
</div>
<div style=" word-wrap:break-word;  word-break:break-all;  overflow: hidden;  width: 55%; margin-left:
px; top: auto; font-size: 15px; margin-top: 10px;">
 {{ mytuple.5 }}
</div>
<div style="margin-top: 5px;">
 <div style="float: left; color: #999999;">
  {{ mytuple.2 }}
 </div>
 <div style="float: left;color: #999999;margin-left:5px;">
  {{ mytuple.3 }}
 </div>
 <div style="float: left;color: #999999;margin-left:5px;">
  type: [{{ mytuple.7 }}]
 </div>
</div>
</div>
{% endfor %}
```

For a nice view in web page, we can choose to hide or show a type of results:

Hide the results:



**Top results for your search: "finance" + "news" + "txt" + "doc" + "pdf" , showing top hits accoring to score:**

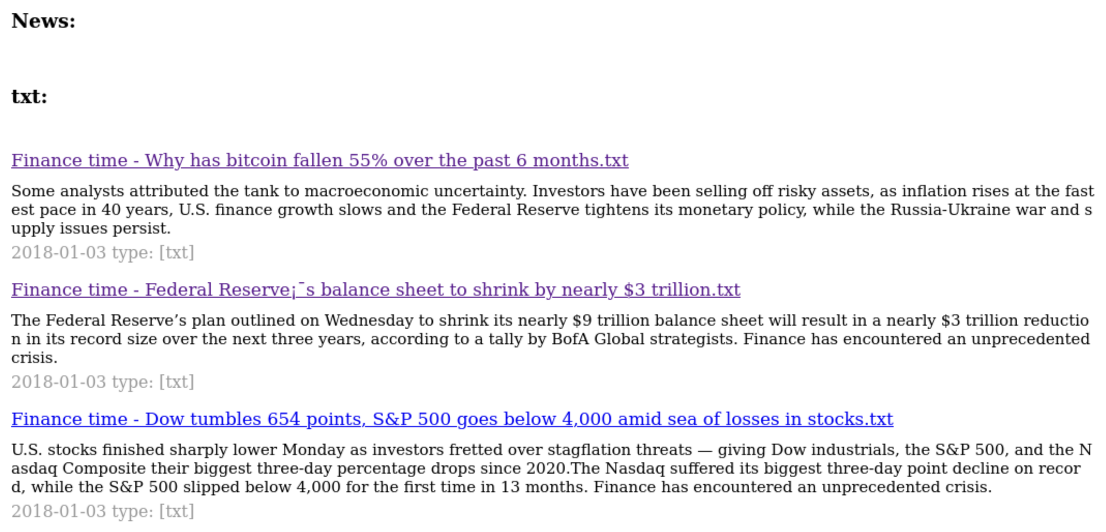
**News:**

**txt:**

**doc:**

**pdf:**

Show the results:



**News:**

**txt:**

Finance time - Why has bitcoin fallen 55% over the past 6 months.txt

Some analysts attributed the tank to macroeconomic uncertainty. Investors have been selling off risky assets, as inflation rises at the fastest pace in 40 years, U.S. finance growth slows and the Federal Reserve tightens its monetary policy, while the Russia-Ukraine war and supply issues persist.
2018-01-03 type: [txt]

Finance time - Federal Reserve¡¯s balance sheet to shrink by nearly $3 trillion.txt

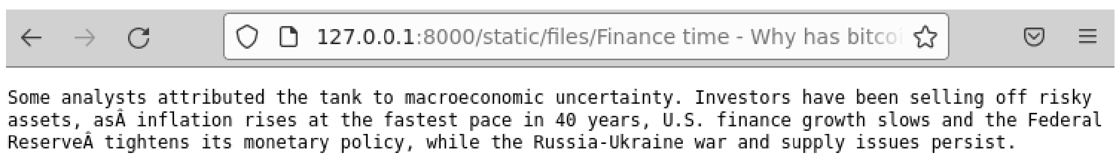
The Federal Reserve's plan outlined on Wednesday to shrink its nearly $9 trillion balance sheet will result in a nearly $3 trillion reduction in its record size over the next three years, according to a tally by BofA Global strategists. Finance has encountered an unprecedented crisis.
2018-01-03 type: [txt]

Finance time - Dow tumbles 654 points, S&P 500 goes below 4,000 amid sea of losses in stocks.txt

U.S. stocks finished sharply lower Monday as investors fretted over stagflation threats — giving Dow industrials, the S&P 500, and the Nasdaq Composite their biggest three-day percentage drops since 2020.The Nasdaq suffered its biggest three-day point decline on record, while the S&P 500 slipped below 4,000 for the first time in 13 months. Finance has encountered an unprecedented crisis.
2018-01-03 type: [txt]

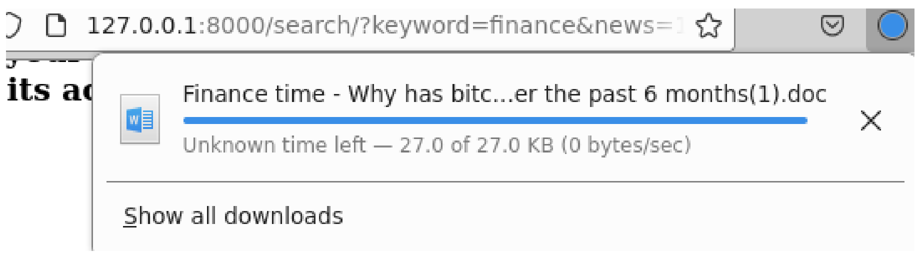
The URL of these types of files can allow users to download or read the files directly:

Txt:



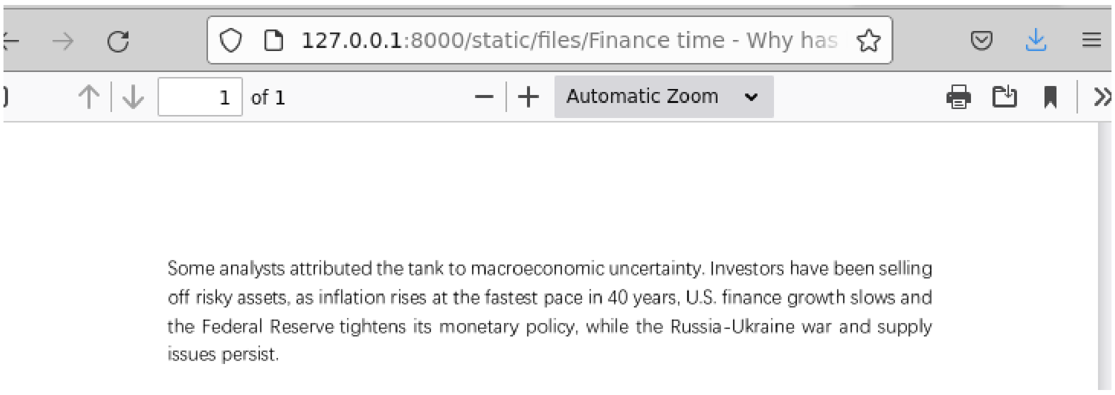
```
Some analysts attributed the tank to macroeconomic uncertainty. Investors have been selling off risky
assets, asÂ inflation rises at the fastest pace in 40 years, U.S. finance growth slows and the Federal
ReserveÂ tightens its monetary policy, while the Russia-Ukraine war and supply issues persist.
```

Doc:



127.0.0.1:8000/search/?keyword=finance&news=
its ac
Finance time - Why has bitc...er the past 6 months(1).doc
Unknown time left — 27.0 of 27.0 KB (0 bytes/sec)
Show all downloads

Pdf:



127.0.0.1:8000/static/files/Finance time - Why has

1 of 1

Automatic Zoom

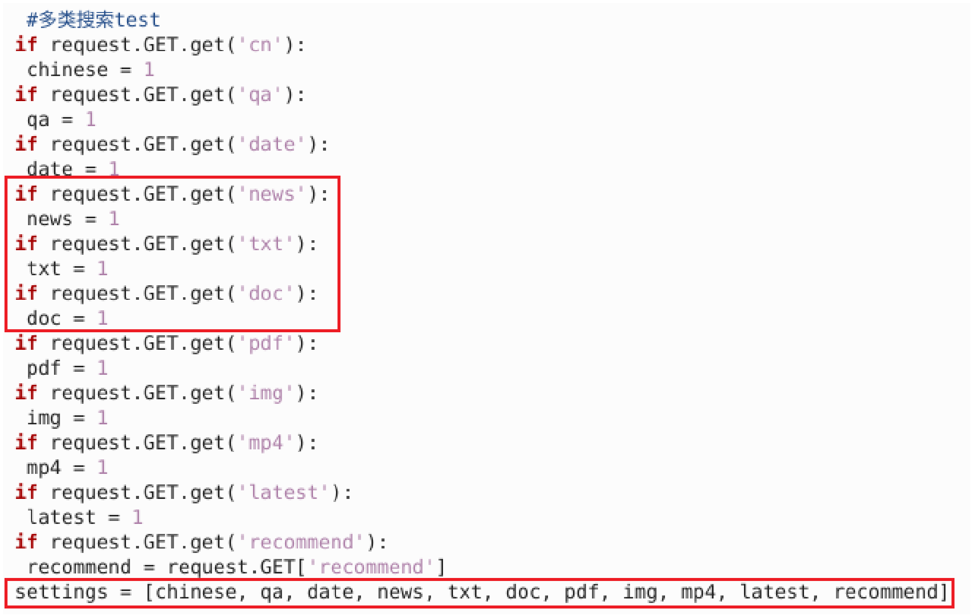
Some analysts attributed the tank to macroeconomic uncertainty. Investors have been selling off risky assets, as inflation rises at the fastest pace in 40 years, U.S. finance growth slows and the Federal Reserve tightens its monetary policy, while the Russia-Ukraine war and supply issues persist.

- Backend:

In **views.py**, we have a list of all "**settings**".

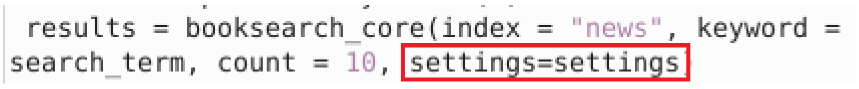
if we get the variable indicating to the types of files with value **"1"**,

we will set the bit of number in **"settings"** to **1**:



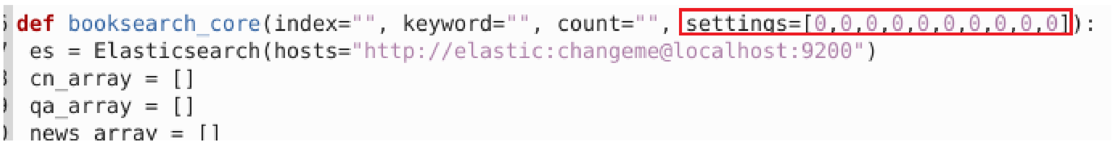
```
 #多类搜索test
if request.GET.get('cn'):
 chinese = 1
if request.GET.get('qa'):
 qa = 1
if request.GET.get('date'):
 date = 1
if request.GET.get('news'):
 news = 1
if request.GET.get('txt'):
 txt = 1
if request.GET.get('doc'):
 doc = 1
if request.GET.get('pdf'):
 pdf = 1
if request.GET.get('img'):
 img = 1
if request.GET.get('mp4'):
 mp4 = 1
if request.GET.get('latest'):
 latest = 1
if request.GET.get('recommend'):
 recommend = request.GET['recommend']
settings = [chinese, qa, date, news, txt, doc, pdf, img, mp4, latest, recommend]
```

Then the **"settings"** will be sent to the **search core function**:



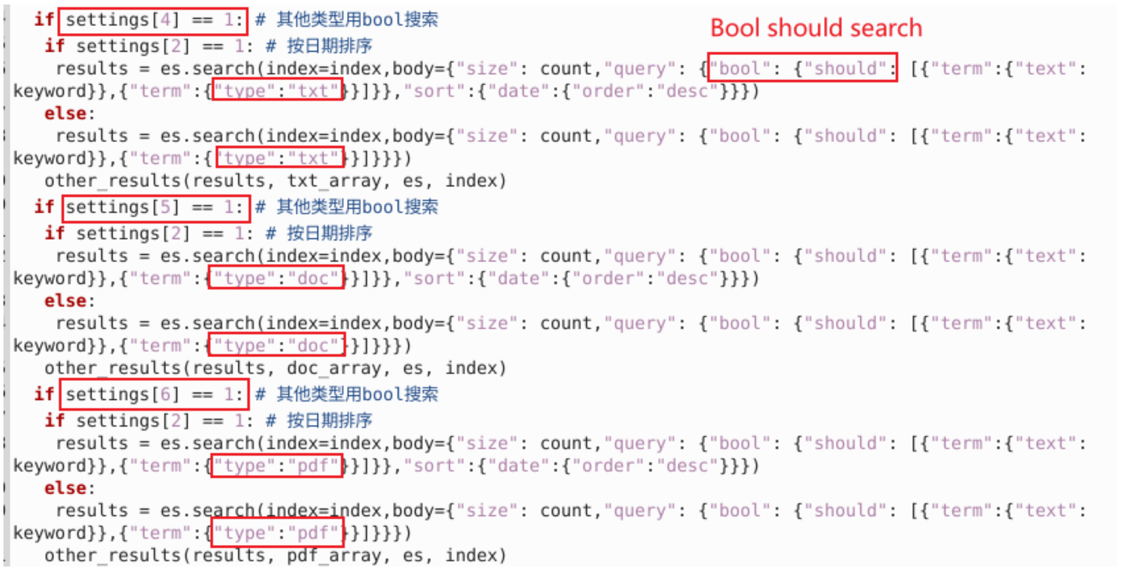
```
 results = booksearch_core(index = "news", keyword =
search_term, count = 10, settings=settings
```

The **search core function** will take the **"settings"** as parameter (the default values are all **0**):



```
def booksearch_core(index="", keyword="", count="", settings=[0,0,0,0,0,0,0,0,0,0,0]):
 es = Elasticsearch(hosts="http://elastic:changeme@localhost:9200")
 cn_array = []
 qa_array = []
 news_array = []
```

Then, according to different types (settings), we will do the search differently using the **Boolean "should"** search method:



```
 if settings[4] == 1: # 其他类型用bool搜索
  if settings[2] == 1: # 按日期排序
   results = es.search(index=index,body={"size": count,"query": {"bool": {"should": [{"term":{"text":
keyword}},{"term":{"type":"txt"}}]}},"sort":{"date":{"order":"desc"}}})
  else:
   results = es.search(index=index,body={"size": count,"query": {"bool": {"should": [{"term":{"text":
keyword}},{"term":{"type":"txt"}}]}}})
  other_results(results, txt_array, es, index)
 if settings[5] == 1: # 其他类型用bool搜索
  if settings[2] == 1: # 按日期排序
   results = es.search(index=index,body={"size": count,"query": {"bool": {"should": [{"term":{"text":
keyword}},{"term":{"type":"doc"}}]}},"sort":{"date":{"order":"desc"}}})
  else:
   results = es.search(index=index,body={"size": count,"query": {"bool": {"should": [{"term":{"text":
keyword}},{"term":{"type":"doc"}}]}}})
  other_results(results, doc_array, es, index)
 if settings[6] == 1: # 其他类型用bool搜索
  if settings[2] == 1: # 按日期排序
   results = es.search(index=index,body={"size": count,"query": {"bool": {"should": [{"term":{"text":
keyword}},{"term":{"type":"pdf"}}]}},"sort":{"date":{"order":"desc"}}})
  else:
   results = es.search(index=index,body={"size": count,"query": {"bool": {"should": [{"term":{"text":
keyword}},{"term":{"type":"pdf"}}]}}})
  other_results(results, pdf_array, es, index)
```

The results of them will be append together into **"results_array"** and be sent back to the frontend:

```
results_array = []
results_array.append(cn_array)
results_array.append(qa_array)
results_array.append(news_array)
results_array.append(txt_array)
results_array.append(doc_array)
results_array.append(pdf_array)
results_array.append(img_array)
results_array.append(mp4_array)
```

② Auto Suggestion (term/phrase):

■ Frontend:

While deciding whether to show results, we have a variable indicating whether there are results (because our **"results"** is a muti-level array, even when it's empty, **"if results"** will return **true**):

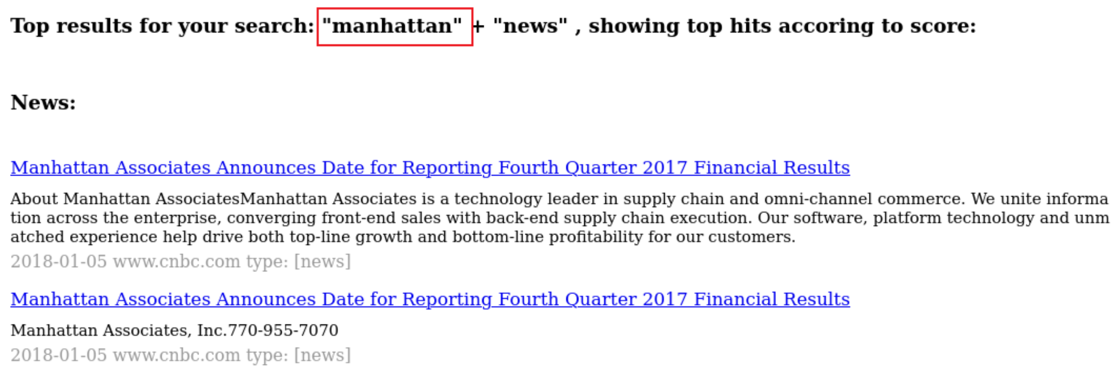
```
{% elif no_results != 1 %}
```

When there are no results, the first probability is that the user input a wrong word, and we will give them **suggestions** (if there are):

**No results found, do you mean: manhattan manhattans mahatma manatt maatta ?**

```
{%elif suggestion %}
<h3>No results found, do you mean:
{%for suggest in suggestion %}
  <a href="http://127.0.0.1:8000/search/?keyword={{suggest}}&news=1">{{suggest}}</a>
  {% endfor %}?
  </h3>
```

By clicking the links, users can search for the suggestions directly:



**Top results for your search: "manhattan" + "news" , showing top hits accoring to score:**

**News:**

Manhattan Associates Announces Date for Reporting Fourth Quarter 2017 Financial Results

About Manhattan AssociatesManhattan Associates is a technology leader in supply chain and omni-channel commerce. We unite information across the enterprise, converging front-end sales with back-end supply chain execution. Our software, platform technology and unmatched experience help drive both top-line growth and bottom-line profitability for our customers.
2018-01-05 www.cnbc.com type: [news]

Manhattan Associates Announces Date for Reporting Fourth Quarter 2017 Financial Results

Manhattan Associates, Inc.770-955-7070
2018-01-05 www.cnbc.com type: [news]

- Backend:

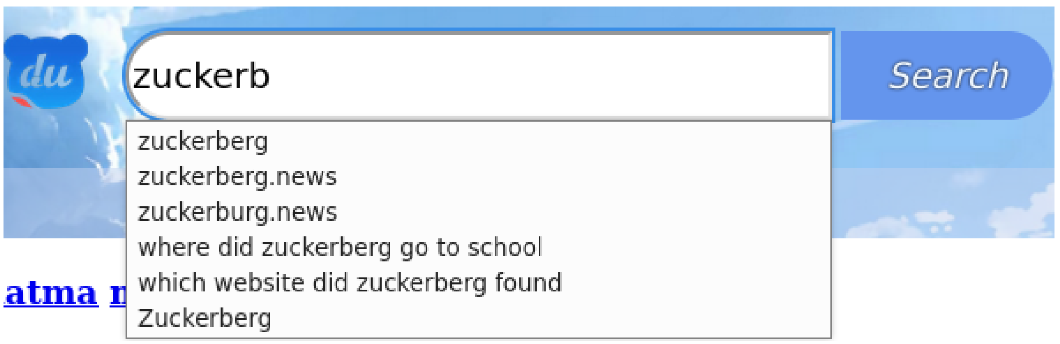
  **Each time** doing the searching, the search core also searches for suggestions and send them to the frontend (whether to show them is decided by the **frontend**):

```
suggestion = es.search(index = index, body={"suggest":{"my-suggestion":{"text":keyword,"term":
{"field":"text"}}}})
```

③ Auto Completion (real-time):

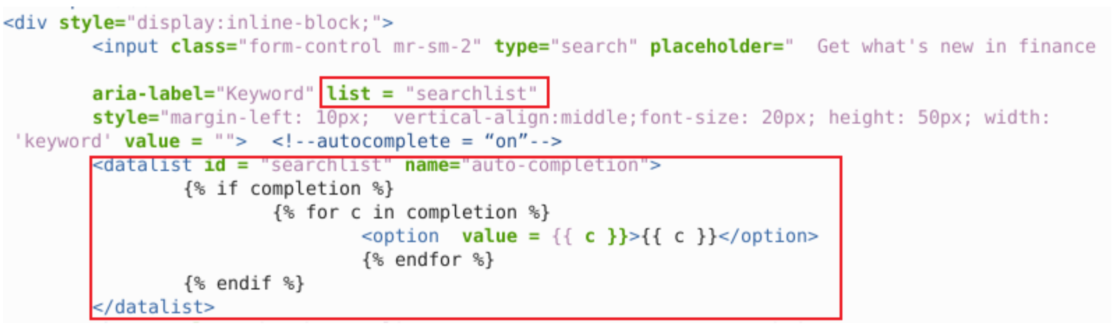
- Frontend:

  While typing the words in our index, the search bar will give users the related words, and users can choose to complete their search terms automatically:



```
zuckerb
zuckerberg
zuckerberg.news
zuckerburg.news
where did zuckerberg go to school
which website did zuckerberg found
Zuckerberg
```

That's because our input bar has a **datalist** to show all results in **"completion"** (got from backend) in real-time:



```
<div style="display:inline-block;">
        <input class="form-control mr-sm-2" type="search" placeholder="  Get what's new in finance

        aria-label="Keyword" list = "searchlist"
        style="margin-left: 10px;  vertical-align:middle;font-size: 20px; height: 50px; width:
 'keyword' value = "">  <!--autocomplete = "on"-->
        <datalist id = "searchlist" name="auto-completion">
                {% if completion %}
                        {% for c in completion %}
                                <option  value = {{ c }}>{{ c }}</option>
                                {% endfor %}
                {% endif %}
        </datalist>
```

■ Backend:

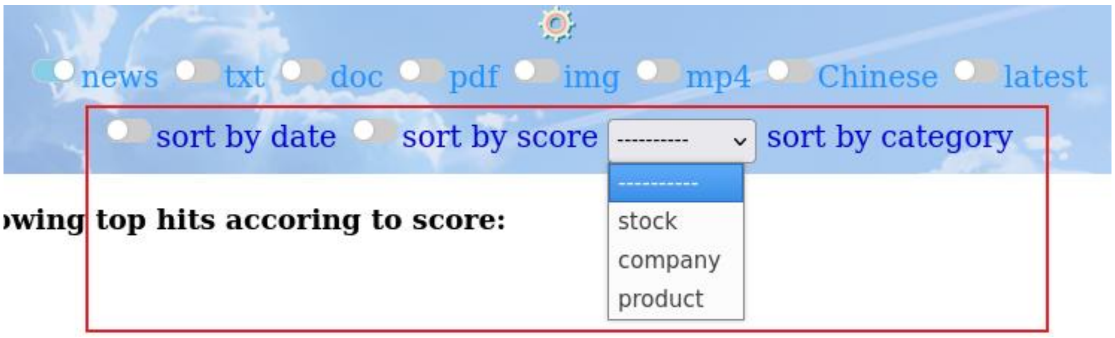
In **search core function**, es will always search for the completion of users' search terms in **"completion"** index (contains the same contents in **"news"** index but in **completion-suggester** arrangement), and send it to the frontend:

```
 completion = es.search(index = "completion", body={"suggest":{"completion-suggestion": {"prefix":
keyword, "completion": {"field": "suggest","size": 5}}}})
```

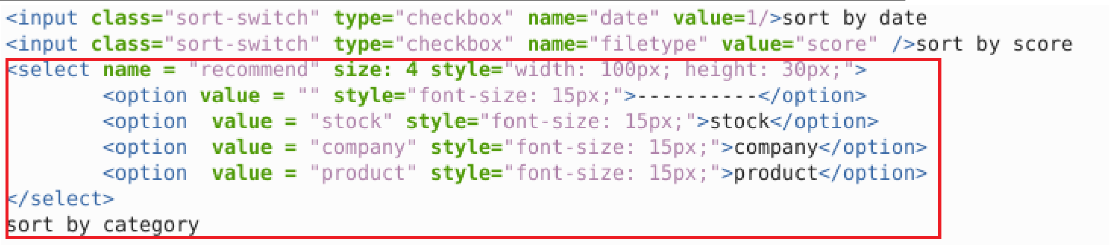
④ Auto Recommendation:

■ Frontend:

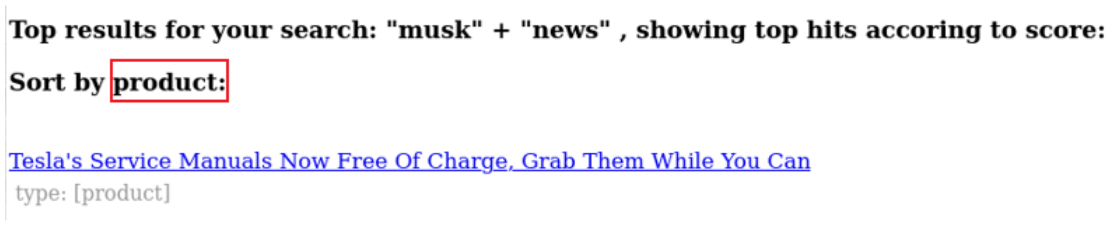
In the **settings** of search, the second line contains settings for sorting method:



We can choose to **"sort by category"** if we want to get recommendation and our search terms have **different meaning** in **different fields/categories** (here we have "**stock**", "**company**" and "**product**"):

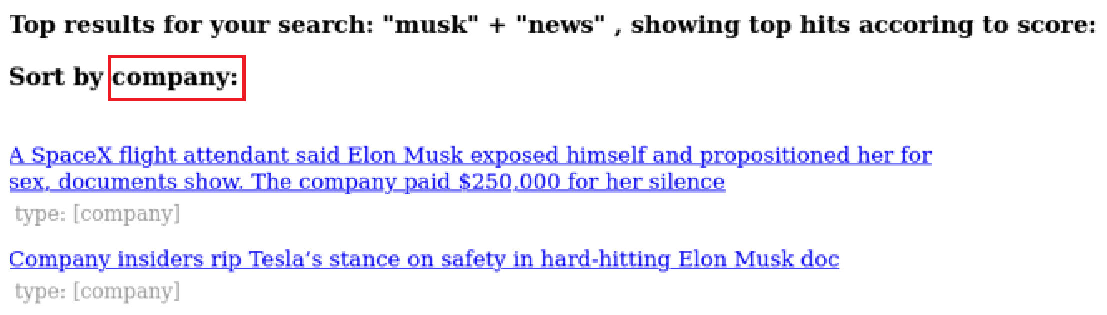

```
<input class="sort-switch" type="checkbox" name="date" value=1/>sort by date
<input class="sort-switch" type="checkbox" name="filetype" value="score" />sort by score
<select name = "recommend" size: 4 style="width: 100px; height: 30px;">
        <option value = "" style="font-size: 15px;">----------</option>
        <option  value = "stock" style="font-size: 15px;">stock</option>
        <option  value = "company" style="font-size: 15px;">company</option>
        <option  value = "product" style="font-size: 15px;">product</option>
</select>
sort by category
```

And it will only show the results in that category:


**Top results for your search: "musk" + "news" , showing top hits accoring to score:**

**Sort by product:**

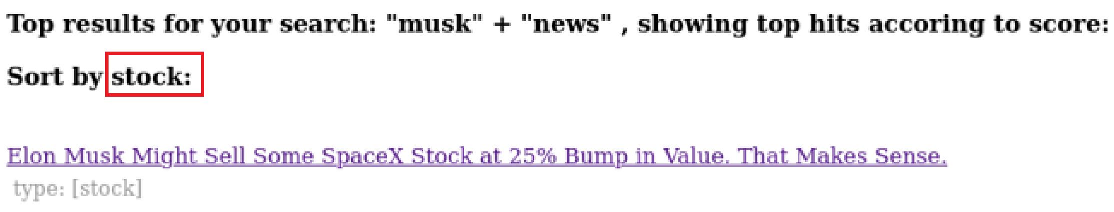
Tesla's Service Manuals Now Free Of Charge, Grab Them While You Can
type: [product]


**Top results for your search: "musk" + "news" , showing top hits accoring to score:**

**Sort by company:**

A SpaceX flight attendant said Elon Musk exposed himself and propositioned her for sex, documents show. The company paid $250,000 for her silence
type: [company]

Company insiders rip Tesla's stance on safety in hard-hitting Elon Musk doc
type: [company]

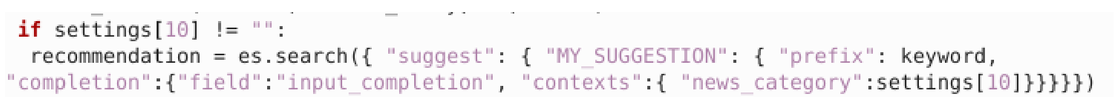

**Top results for your search: "musk" + "news" , showing top hits accoring to score:**

**Sort by stock:**

Elon Musk Might Sell Some SpaceX Stock at 25% Bump in Value. That Makes Sense.
type: [stock]

■ Backend:

While get the **"settings"** that enable recommendation, the **search core function** will search the results in using **context-suggester**:

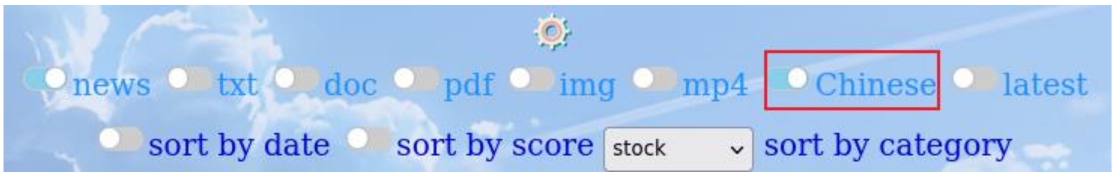

```
 if settings[10] != "":
  recommendation = es.search({ "suggest": { "MY_SUGGESTION": { "prefix": keyword,
"completion":{"field":"input_completion", "contexts":{ "news_category":settings[10]}}}}})
```

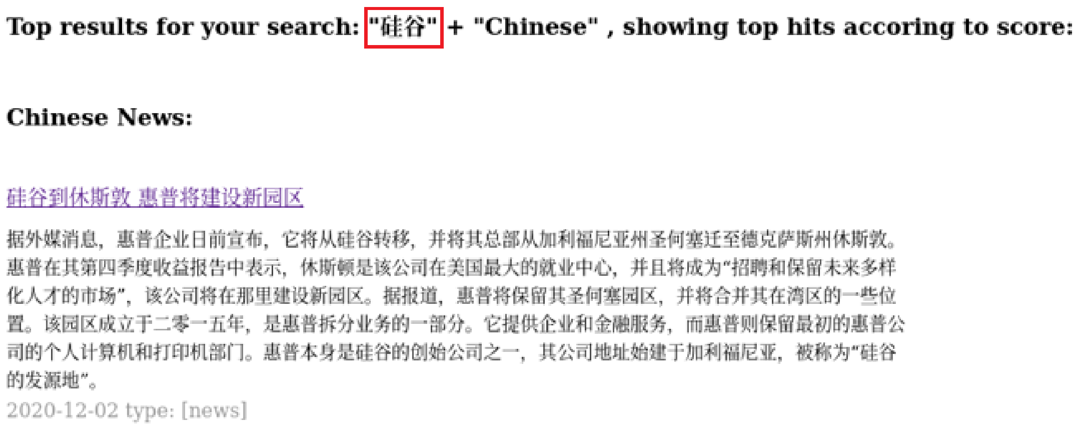
And send them to the frontend.

⑤ Chinese Pinyin Search (e.g. type "guangdongzhuhai" or "gdzh" will search for "广东珠海"):

■ Frontend:

If we enabled the **“Chinese”** search in settings, we can do the search using **Chinese/Pinyin**:



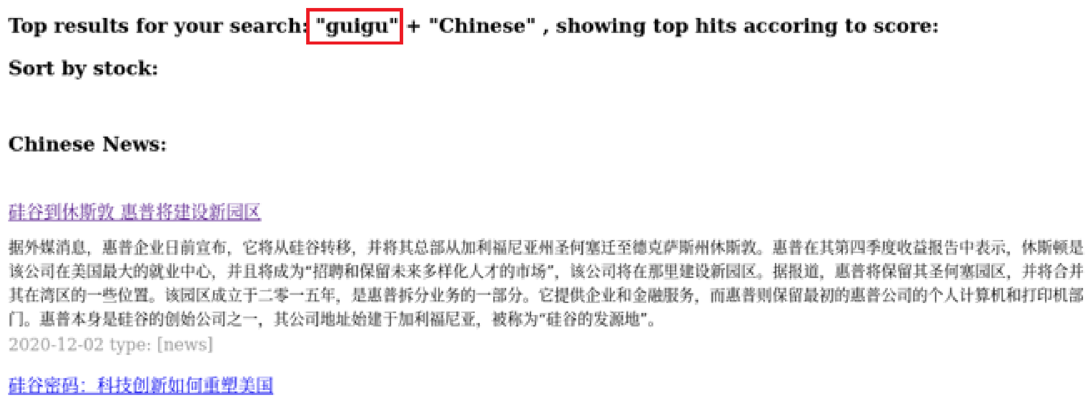


**Top results for your search: "硅谷" + "Chinese" , showing top hits accoring to score:**

**Chinese News:**

硅谷到休斯敦 惠普将建设新园区

据外媒消息，惠普企业日前宣布，它将从硅谷转移，并将其总部从加利福尼亚州圣何塞迁至德克萨斯州休斯敦。惠普在其第四季度收益报告中表示，休斯顿是该公司在美国最大的就业中心，并且将成为"招聘和保留未来多样化人才的市场"，该公司将在那里建设新园区。据报道，惠普将保留其圣何塞园区，并将合并其在湾区的一些位置。该园区成立于二零一五年，是惠普拆分业务的一部分。它提供企业和金融服务，而惠普则保留最初的惠普公司的个人计算机和打印机部门。惠普本身是硅谷的创始公司之一，其公司地址始建于加利福尼亚，被称为"硅谷的发源地"。
2020-12-02 type: [news]



**Top results for your search: "guigu" + "Chinese" , showing top hits accoring to score:**
**Sort by stock:**

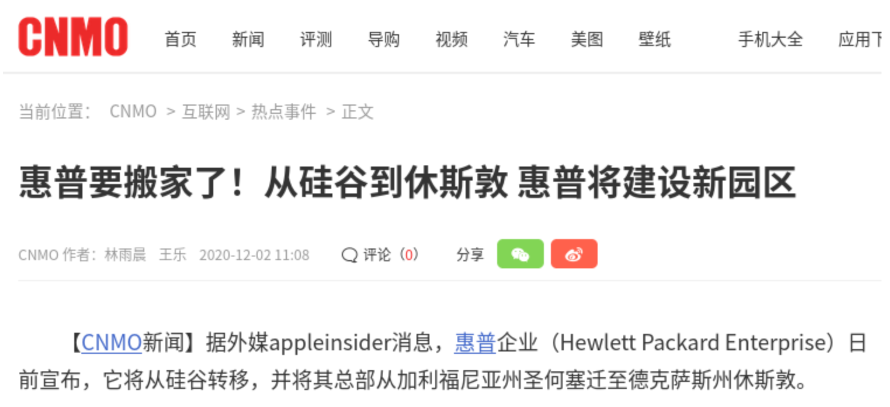
**Chinese News:**

硅谷到休斯敦 惠普将建设新园区

据外媒消息，惠普企业日前宣布，它将从硅谷转移，并将其总部从加利福尼亚州圣何塞迁至德克萨斯州休斯敦。惠普在其第四季度收益报告中表示，休斯顿是该公司在美国最大的就业中心，并且将成为"招聘和保留未来多样化人才的市场"，该公司将在那里建设新园区。据报道，惠普将保留其圣何塞园区，并将合并其在湾区的一些位置。该园区成立于二零一五年，是惠普拆分业务的一部分。它提供企业和金融服务，而惠普则保留最初的惠普公司的个人计算机和打印机部门。惠普本身是硅谷的创始公司之一，其公司地址始建于加利福尼亚，被称为"硅谷的发源地"。
2020-12-02 type: [news]

硅谷密码：科技创新如何重塑美国

By clicking the URLs, we can get to the Chinese news site:



当前位置： CNMO > 互联网 > 热点事件 > 正文

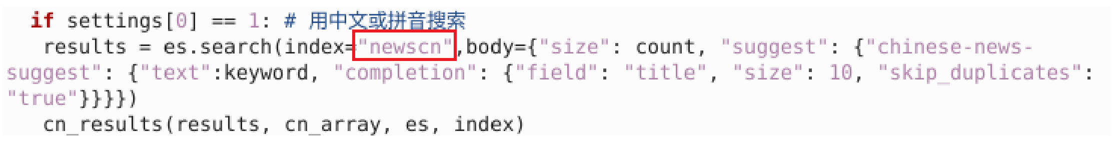
惠普要搬家了！从硅谷到休斯敦 惠普将建设新园区

CNMO 作者：林雨晨 王乐 2020-12-02 11:08 评论（0） 分享

【CNMO新闻】据外媒appleinsider消息，惠普企业（Hewlett Packard Enterprise）日前宣布，它将从硅谷转移，并将其总部从加利福尼亚州圣何塞迁至德克萨斯州休斯敦。

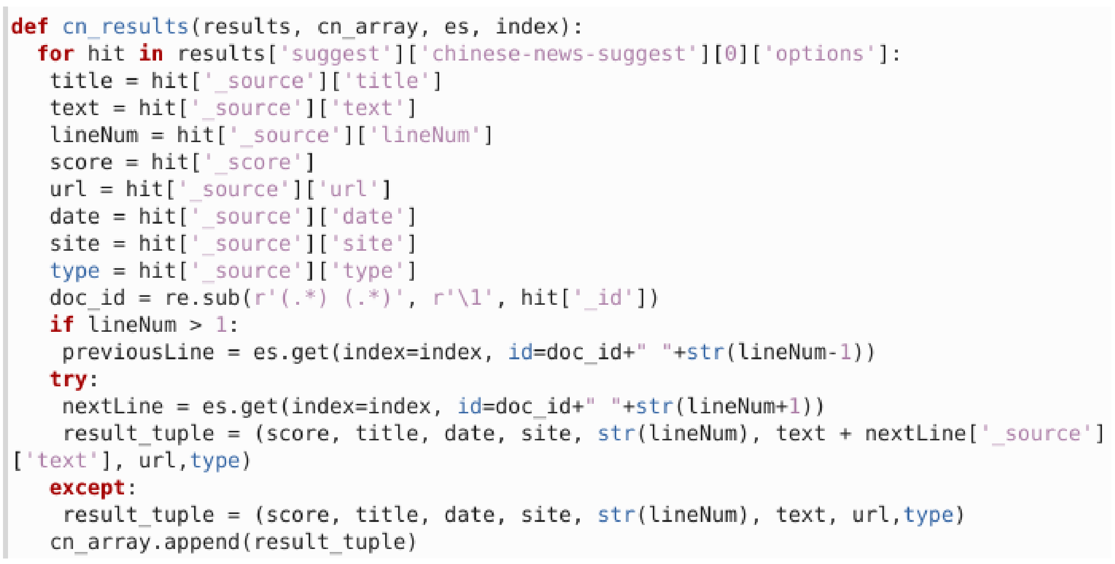
■ Backend:

While get the **"settings"** enabled the Chinese search, the search core function will search for results in **"newscn"** (contains some news in Chinese in the same structure as **"news"**) index using Chinese-suggester:



```
  if settings[0] == 1: # 用中文或拼音搜索
    results = es.search(index="newscn",body={"size": count, "suggest": {"chinese-news-
suggest": {"text":keyword, "completion": {"field": "title", "size": 10, "skip_duplicates":
"true"}}}})
    cn_results(results, cn_array, es, index)
```

And call a function to process the Chinese results:

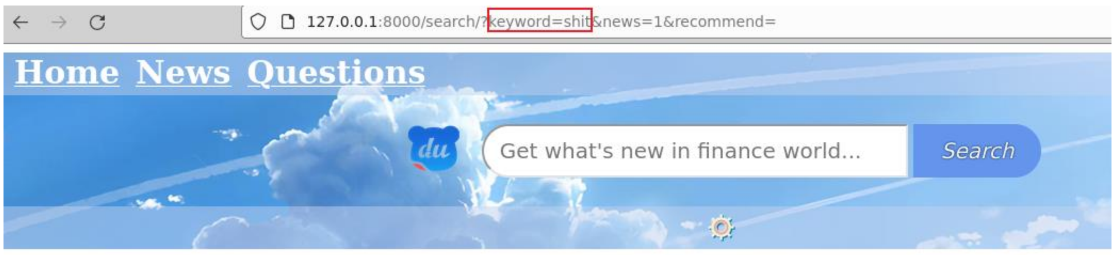


```
def cn_results(results, cn_array, es, index):
  for hit in results['suggest']['chinese-news-suggest'][0]['options']:
    title = hit['_source']['title']
    text = hit['_source']['text']
    lineNum = hit['_source']['lineNum']
    score = hit['_score']
    url = hit['_source']['url']
    date = hit['_source']['date']
    site = hit['_source']['site']
    type = hit['_source']['type']
    doc_id = re.sub(r'(.*) (.*)', r'\1', hit['_id'])
    if lineNum > 1:
     previousLine = es.get(index=index, id=doc_id+" "+str(lineNum-1))
    try:
     nextLine = es.get(index=index, id=doc_id+" "+str(lineNum+1))
     result_tuple = (score, title, date, site, str(lineNum), text + nextLine['_source']
['text'], url,type)
    except:
     result_tuple = (score, title, date, site, str(lineNum), text, url,type)
    cn_array.append(result_tuple)
```

⑥ Blacklist Word (searching for some political/offensive words will not return results):

■ Frontend:

If we search for a word in the blacklist, the search will return no results and give a warning:


127.0.0.1:8000/search/?keyword=shit&news=1&recommend=
Home News Questions
Get what's new in finance world...
Search

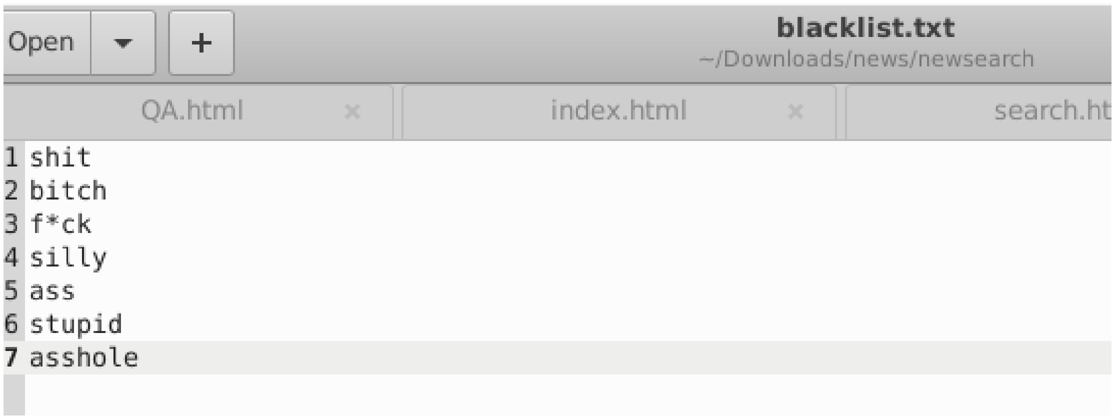
**Contains forbidden words! No results shown.**

It happened because there's a check to determine whether there are forbidden words in the search term:

```
{% if black_word %}

 <h3>Contains forbidden words! No results shown.</h3>
```

- Backend:

We have a **blacklist file** which contains all of the forbidden words, and we can add or remove some of them at any time:

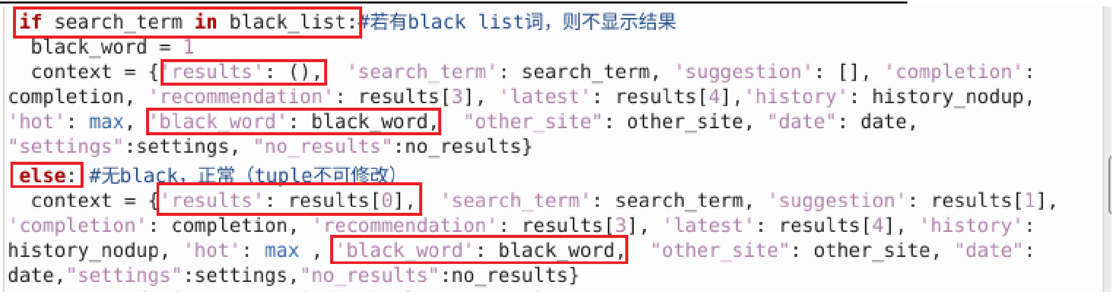


```
shit
bitch
f*ck
silly
ass
stupid
asshole
```

**"view.py"** would open and check whether there are words of search term inside the blacklist, and use a variable **"black_word"** to indicate whether the search term is **"clean"**:

```
black_word = 0 # indicate whether there are words in black list
black_list = []
with open('/root/Downloads/news/newsearch/blacklist.txt', 'r') as file:
 for line in file:
  black_list.append(line.strip())
```

The search results sent to the frontend will be **empty** if there are black words, and the variable **"black_word"** will also be sent to the frontend:

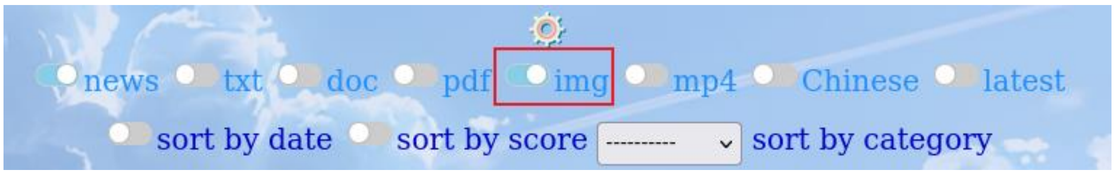


```
 if search_term in black_list:#若有black list词, 则不显示结果
   black_word = 1
   context = {'results': (),  'search_term': search_term, 'suggestion': [], 'completion':
completion, 'recommendation': results[3], 'latest': results[4],'history': history_nodup,
'hot': max, 'black_word': black_word,  "other_site": other_site, "date": date,
"settings":settings, "no_results":no_results}
 else: #无black, 正常 (tuple不可修改)
   context = {'results': results[0],  'search_term': search_term, 'suggestion': results[1],
'completion': completion, 'recommendation': results[3], 'latest': results[4], 'history':
history_nodup, 'hot': max , 'black_word': black_word,  "other_site": other_site, "date":
date,"settings":settings,"no_results":no_results}
```

⑦ Search for Picture:

■ Frontend:

If we enabled the **"img"** search in settings, we can search for images:



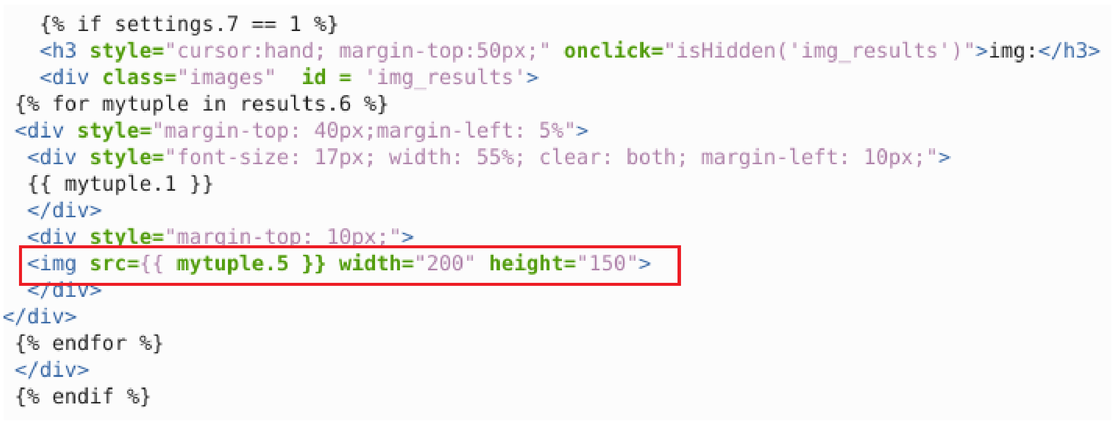
**Top results for your search: "sillicon valley" + "img" , showing top hits accoring to score:**

**img:**

Silicon Valley.jpg

Silicon Valley.jpg

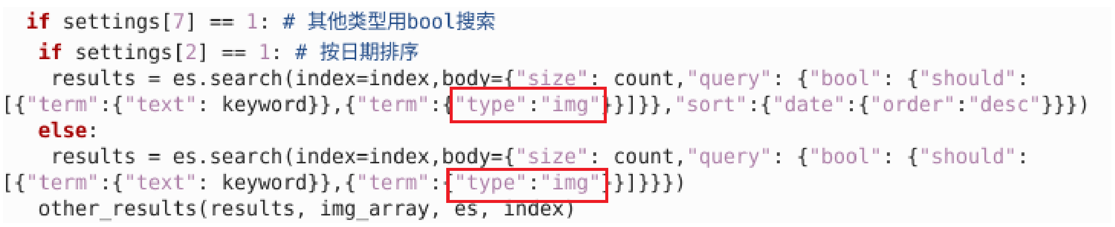
The arrangement is different from texts, which will show the picture of the URLs stored in the "text" field of images:



```
  {% if settings.7 == 1 %}
  <h3 style="cursor:hand; margin-top:50px;" onclick="isHidden('img_results')">img:</h3>
  <div class="images"  id = 'img_results'>
 {% for mytuple in results.6 %}
 <div style="margin-top: 40px;margin-left: 5%">
  <div style="font-size: 17px; width: 55%; clear: both; margin-left: 10px;">
  {{ mytuple.1 }}
  </div>
  <div style="margin-top: 10px;">
  <img src={{ mytuple.5 }} width="200" height="150">
  </div>
</div>
 {% endfor %}
 </div>
 {% endif %}
```

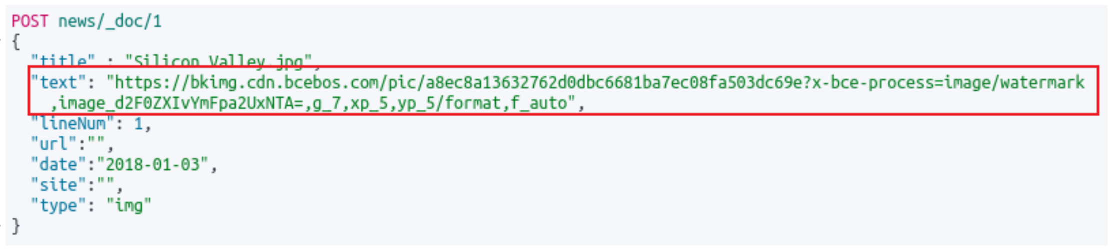
■ Backend:

Also, when get the "**settings**" of images, we use the Boolean search for **"img"** type:



```
  if settings[7] == 1: # 其他类型用bool搜索
   if settings[2] == 1: # 按日期排序
    results = es.search(index=index,body={"size": count,"query": {"bool": {"should":
[{"term":{"text": keyword}},{"term":{"type":"img"}}]}},"sort":{"date":{"order":"desc"}}})
   else:
    results = es.search(index=index,body={"size": count,"query": {"bool": {"should":
[{"term":{"text": keyword}},{"term":{"type":"img"}}]}}})
   other_results(results, img_array, es, index)
```

In the index, the images are actually in the form of **URLs** stored in the **"text"** field:

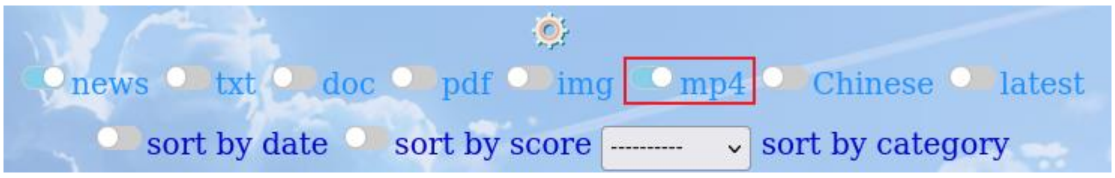


```
POST news/_doc/1
{
  "title" : "Silicon Valley.jpg",
  "text": "https://bkimg.cdn.bcebos.com/pic/a8ec8a13632762d0dbc6681ba7ec08fa503dc69e?x-bce-process=image/watermark
   ,image_d2F0ZXIvYmFpa2UxNTA=,g_7,xp_5,yp_5/format,f_auto",
  "lineNum": 1,
  "url":"",
  "date":"2018-01-03",
  "site":"",
  "type": "img"
}
```

⑧ Search for Video:

■ Frontend:

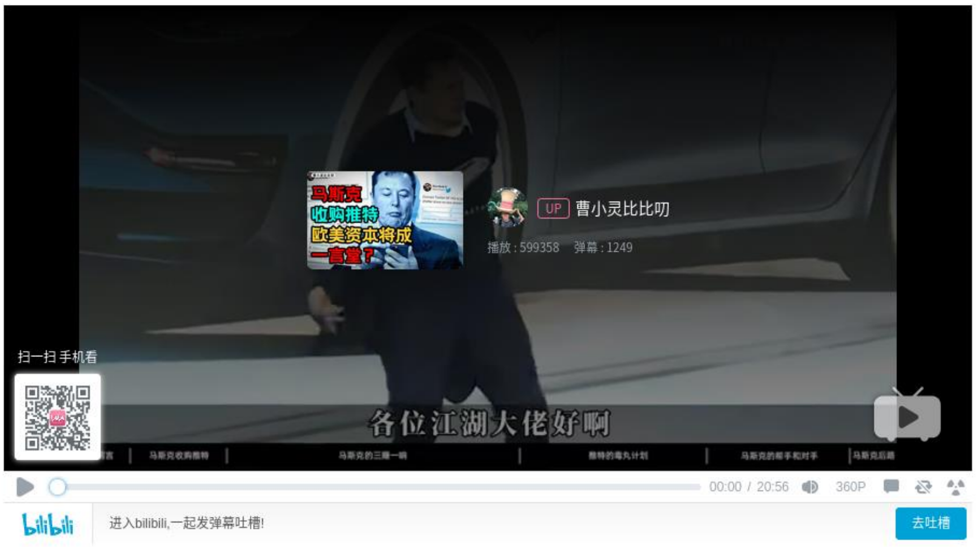
If we enabled the **"mp4"** search in settings, we can search for videos:



**Top results for your search: "twitter" + "mp4" , showing top hits accoring to score:**

**mp4:**

Twitter agrees to sell itself to Elon Musk.mp4


马斯克收购推特 欧美资本将成一言堂?
UP 曹小灵比比叨
播放 : 599358 弹幕 : 1249
扫一扫 手机看
各位江湖大佬好啊
00:00 / 20:56
360P
bilibili
进入bilibili,一起发弹幕吐槽!
去吐槽

To show and play videos in our web page, we call the **web embedded links** of **Bilibili** and put them in an **"iframe"**:

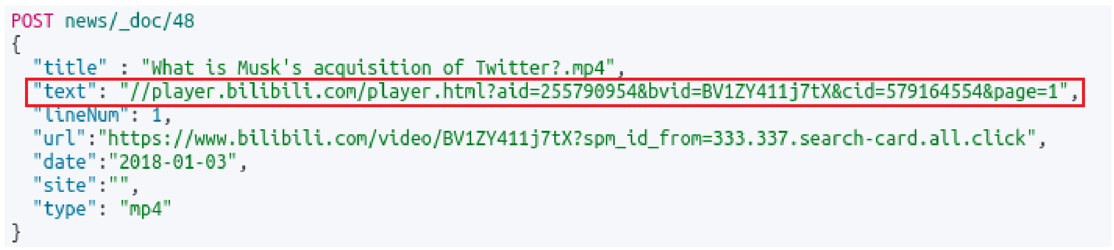
```
   {% if settings.8 == 1 %}
   <h3 style="cursor:hand; margin-top:50px;" onclick="isHidden('mp4_results')">mp4:</h3>
   <div class="images" id = 'mp4_results'>
 {% for mytuple in results.7 %}
 <div class="images" style="margin-top: 20px; margin-left: 5%;">
 <div>
  <a href= {{ mytuple.6 }} >{{ mytuple.1 }}</ a>
 </div>
 <div style="margin-top: 10px;">
  <iframe src={{ mytuple.5 }} scrolling="no" border="0" frameborder="no" framespacing="0"
allowfullscreen="true" width = "900" height = "500"> </iframe>
 </div>
</div>
 {% endfor %}
 </div>
 {% endif %}
```

- Backend:

Also, when get the "**settings**" of videos, we use the Boolean search for **"mp4"** type:

```
  if settings[8] == 1: # 其他类型用bool搜索
   if settings[2] == 1: # 按日期排序
    results = es.search(index=index,body={"size": count,"query": {"bool": {"should":
[{"term":{"text": keyword}},{"term":{"type":"mp4"}}]}},"sort":{"date":{"order":"desc"}}})
   else:
    results = es.search(index=index,body={"size": 4,"query": {"bool": {"should": [{"term":
{"text": keyword}},{"term":{"type":"mp4"}}]}}})
   other_results(results, mp4_array, es, index)
```

In the index, the videos are actually in the form of **embedded URLs** stored in the **"text"** field:

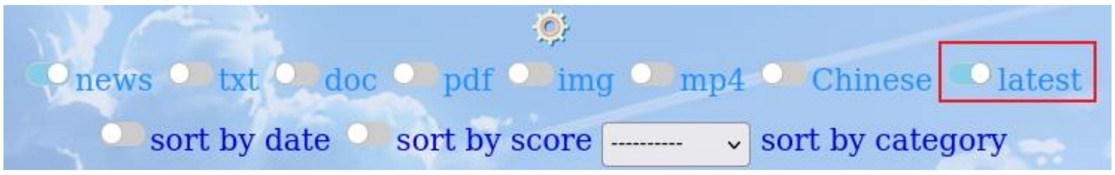


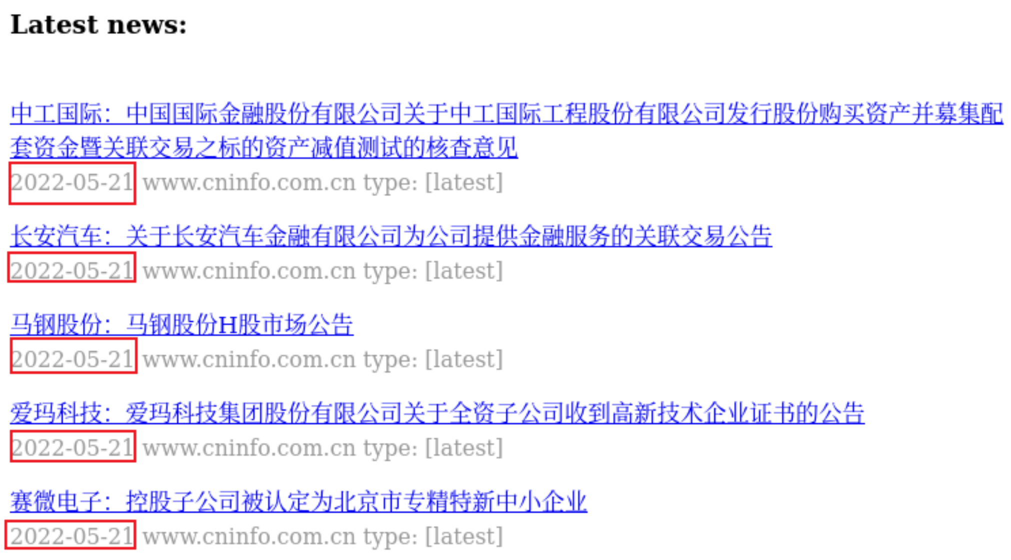
```
POST news/_doc/48
{
  "title" : "What is Musk's acquisition of Twitter?.mp4",
  "text": "//player.bilibili.com/player.html?aid=255790954&bvid=BV1ZY411j7tX&cid=579164554&page=1",
  "lineNum": 1,
  "url":"https://www.bilibili.com/video/BV1ZY411j7tX?spm_id_from=333.337.search-card.all.click",
  "date":"2018-01-03",
  "site":"",
  "type": "mp4"
}
```

⑨ Search for News, e.g., real-time crawling and indexing of latest website documents (news article from QQ/163, crawl data by ourselves):

■ Frontend:

If we enabled the "**latest**" search in settings, we can search for real-time crawled news (in Chinese):





**Latest news:**

[中工国际：中国国际金融股份有限公司关于中工国际工程股份有限公司发行股份购买资产并募集配套资金暨关联交易之标的资产减值测试的核查意见]
2022-05-21 www.cninfo.com.cn type: [latest]

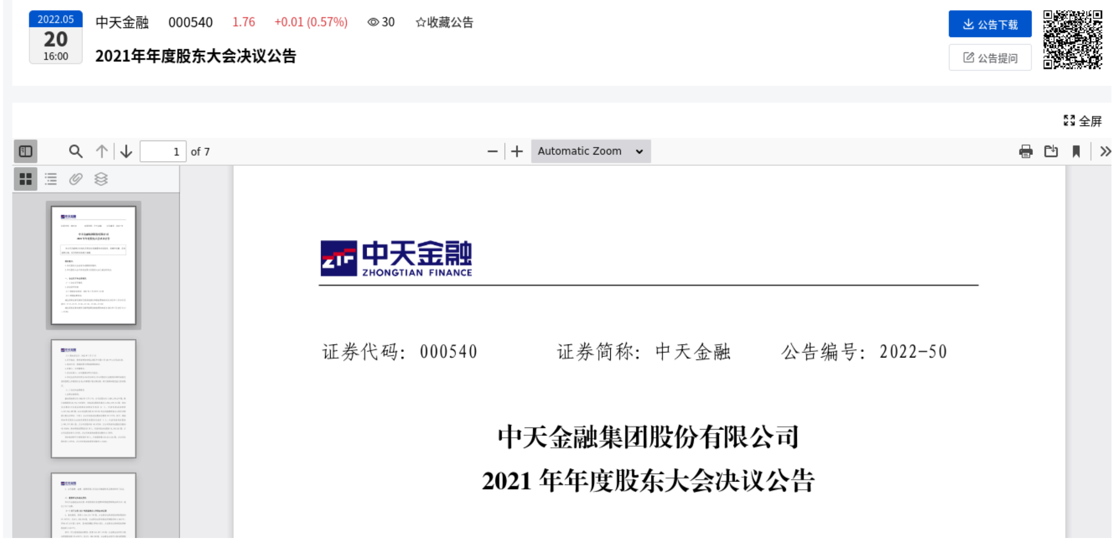
[长安汽车：关于长安汽车金融有限公司为公司提供金融服务的关联交易公告]
2022-05-21 www.cninfo.com.cn type: [latest]

[马钢股份：马钢股份H股市场公告]
2022-05-21 www.cninfo.com.cn type: [latest]

[爱玛科技：爱玛科技集团股份有限公司关于全资子公司收到高新技术企业证书的公告]
2022-05-21 www.cninfo.com.cn type: [latest]

[赛微电子：控股子公司被认定为北京市专精特新中小企业]
2022-05-21 www.cninfo.com.cn type: [latest]

By clicking the URLs, we can check the detailed contents:


2022.05 20 16:00 中天金融 000540 1.76 +0.01 (0.57%) 30 ☆收藏公告
2021年年度股东大会决议公告

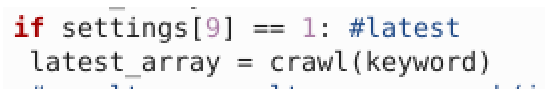
中天金融 ZHONGTIAN FINANCE

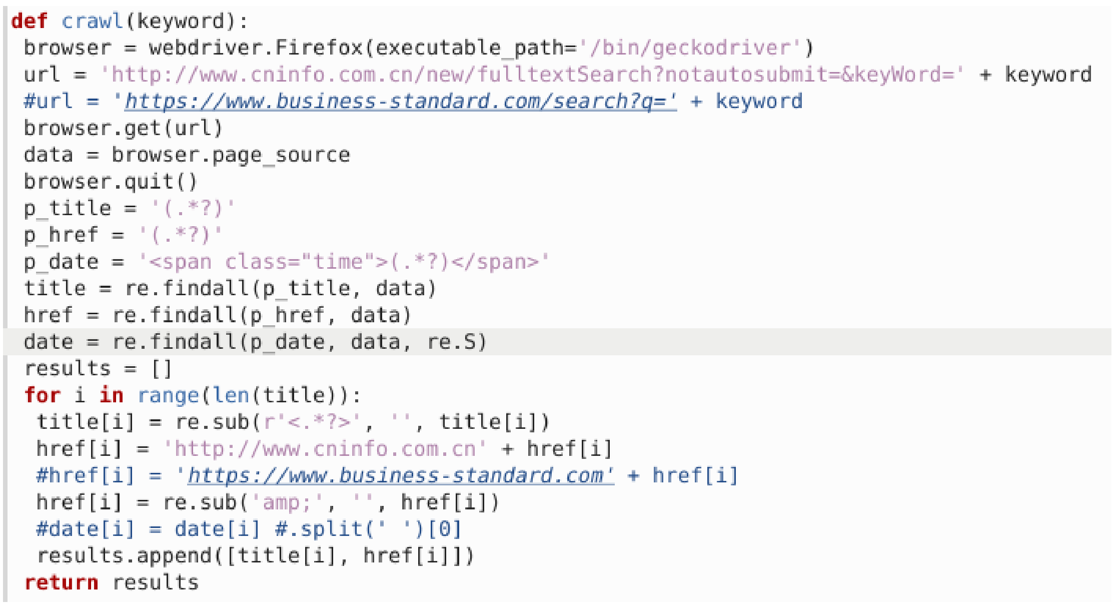
证券代码：000540　　证券简称：中天金融　　公告编号：2022-50

中天金融集团股份有限公司
2021 年年度股东大会决议公告

■ Backend:

When get the “**settings**” of latest news, we call the function to crawl data from http://finance.sina.com.cn/realstock/company/sh000001/nc.shtml:


```
if settings[9] == 1: #latest
    latest_array = crawl(keyword)
```

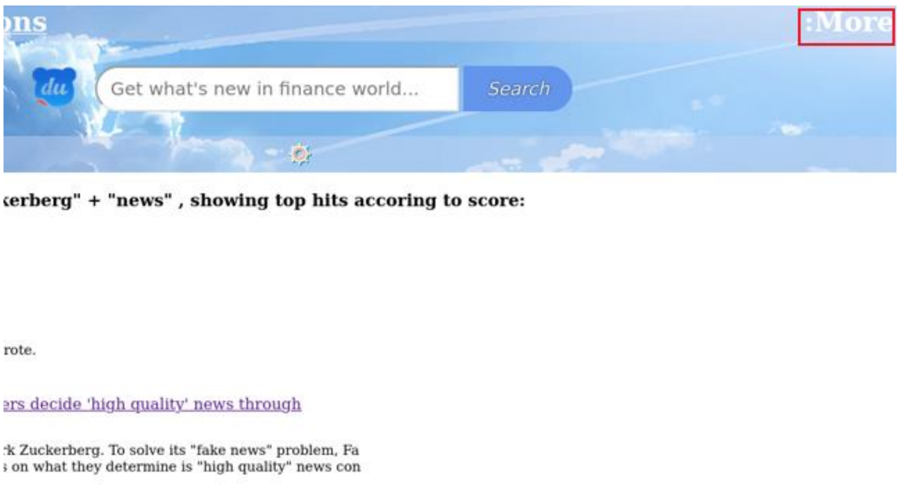

```
def crawl(keyword):
 browser = webdriver.Firefox(executable_path='/bin/geckodriver')
 url = 'http://www.cninfo.com.cn/new/fulltextSearch?notautosubmit=&keyWord=' + keyword
 #url = 'https://www.business-standard.com/search?q=' + keyword
 browser.get(url)
 data = browser.page_source
 browser.quit()
 p_title = '(.*?)'
 p_href = '(.*?)'
 p_date = '<span class="time">(.*?)</span>'
 title = re.findall(p_title, data)
 href = re.findall(p_href, data)
 date = re.findall(p_date, data, re.S)
 results = []
 for i in range(len(title)):
  title[i] = re.sub(r'<.*?>', '', title[i])
  href[i] = 'http://www.cninfo.com.cn' + href[i]
  #href[i] = 'https://www.business-standard.com' + href[i]
  href[i] = re.sub('amp;', '', href[i])
  #date[i] = date[i] #.split(' ')[0]
  results.append([title[i], href[i]])
 return results
```

⑩ Record Search History:

■ Frontend:

In every page of our search engine, we have a “**More**” button, which is a **drop-down menu** containing the link to “**backend**” page and

**history** bar:

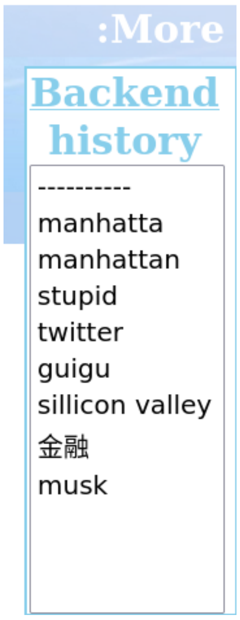


erberg" + "news" , showing top hits accoring to score:

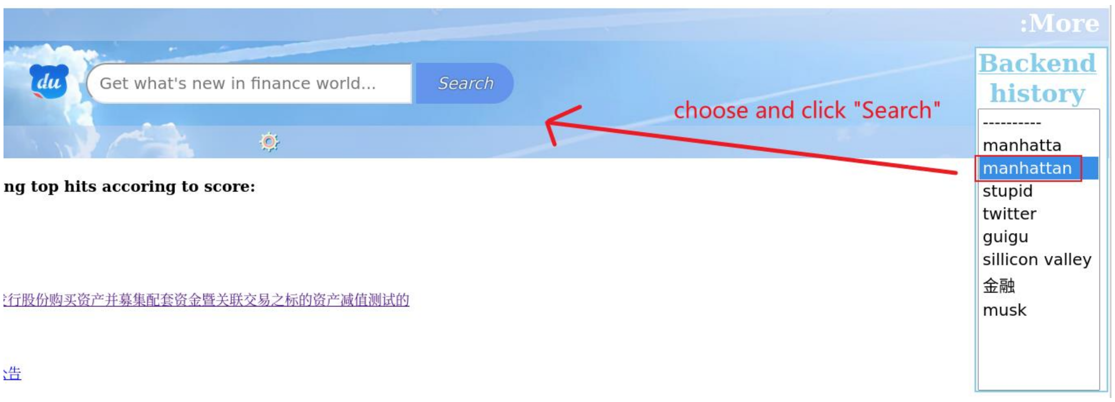
rote.

ers decide 'high quality' news through

k Zuckerberg. To solve its "fake news" problem, Fa
on what they determine is "high quality" news con



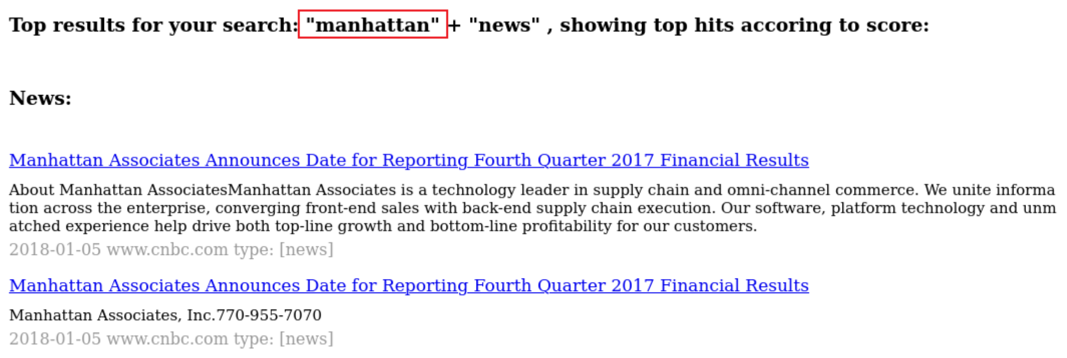
We can choose one history to search directly:



ng top hits accoring to score:

行股份购买资产并募集配套资金暨关联交易之标的资产减值测试的

造



**Top results for your search: "manhattan" + "news" , showing top hits accoring to score:**

**News:**

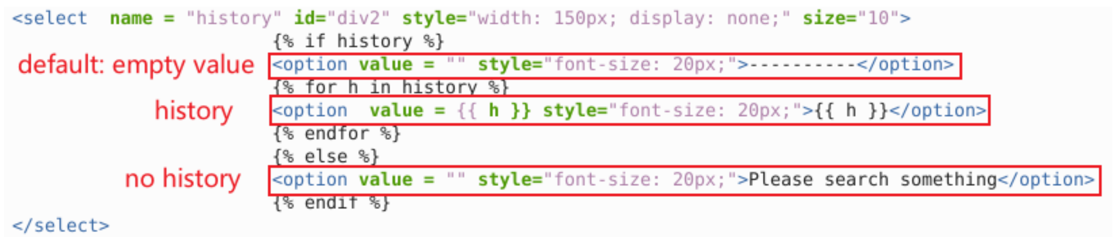
Manhattan Associates Announces Date for Reporting Fourth Quarter 2017 Financial Results

About Manhattan AssociatesManhattan Associates is a technology leader in supply chain and omni-channel commerce. We unite informa tion across the enterprise, converging front-end sales with back-end supply chain execution. Our software, platform technology and unm atched experience help drive both top-line growth and bottom-line profitability for our customers.
2018-01-05 www.cnbc.com type: [news]

Manhattan Associates Announces Date for Reporting Fourth Quarter 2017 Financial Results

Manhattan Associates, Inc.770-955-7070
2018-01-05 www.cnbc.com type: [news]

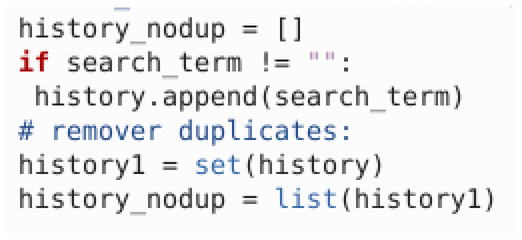
That's because the "**history**" bar is a selection bar, which can show the history got from backend, and send the value (history) we choose as a search term to backend to search:



```
<select  name = "history" id="div2" style="width: 150px; display: none;" size="10">
                          {% if history %}
                          <option value = "" style="font-size: 20px;">----------</option>
                          {% for h in history %}
                          <option  value = {{ h }} style="font-size: 20px;">{{ h }}</option>
                          {% endfor %}
                          {% else %}
                          <option value = "" style="font-size: 20px;">Please search something</option>
                          {% endif %}
</select>
```

■ Backend:

In **"views.py"**, we stored all search terms in a list of history, and then **remove duplicates** in it because each history needs to be unique (if a term is searched for more than one times, that number can be checked in the "**backend**" page of our search engine):

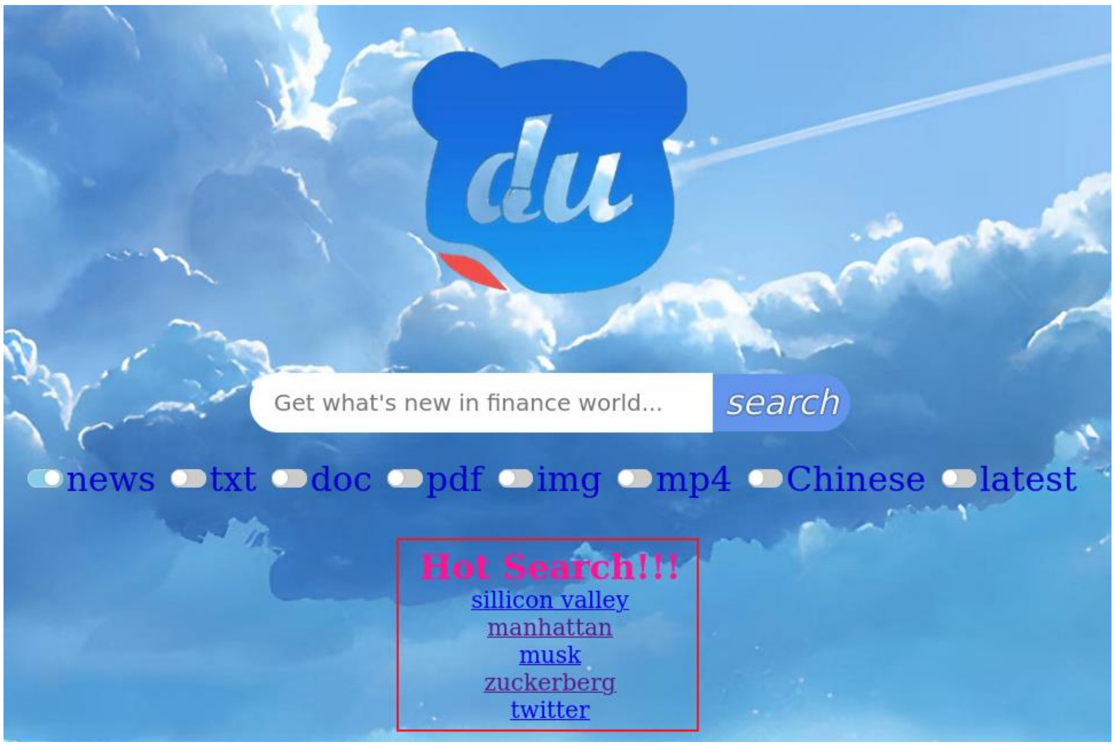


```
history_nodup = []
if search_term != "":
 history.append(search_term)
# remover duplicates:
history1 = set(history)
history_nodup = list(history1)
```

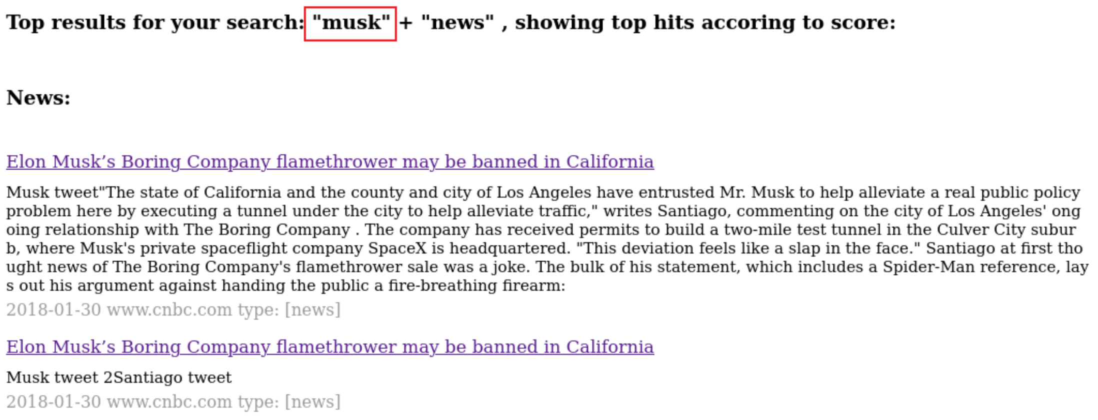
⑪ Show the "Hot Search":

■ Frontend:

In our home page (**index.html**), there's an area to show hot search, which are terms that users searched for the most times (**top 5**):



We can click them to search directly:



**Top results for your search: "musk" + "news" , showing top hits accoring to score:**

**News:**

Elon Musk’s Boring Company flamethrower may be banned in California

Musk tweet"The state of California and the county and city of Los Angeles have entrusted Mr. Musk to help alleviate a real public policy problem here by executing a tunnel under the city to help alleviate traffic," writes Santiago, commenting on the city of Los Angeles' ong oing relationship with The Boring Company . The company has received permits to build a two-mile test tunnel in the Culver City subur b, where Musk's private spaceflight company SpaceX is headquartered. "This deviation feels like a slap in the face." Santiago at first tho ught news of The Boring Company's flamethrower sale was a joke. The bulk of his statement, which includes a Spider-Man reference, lay s out his argument against handing the public a fire-breathing firearm:

2018-01-30 www.cnbc.com type: [news]

Elon Musk’s Boring Company flamethrower may be banned in California

Musk tweet 2Santiago tweet

2018-01-30 www.cnbc.com type: [news]

The frontend code is like following:

```
<div style="height: 20px; font-size: 30px;">
        <h4 id="char">Hot Search!!!</h4>
</div>
<div style="margin-top: 15px;">
{% for m in hot %}
<div>
        <a href="http://127.0.0.1:8000/search/?keyword={{m}}&news=1">{{ m }}</a>
        </div>
{% endfor %}
</div>
```

- Backend:

In “views.py”, we calculated the top 5 search terms (“**max**”) in history (with duplicates) by search times:

```
count_set = {}  # 存放元素和出现次数的字典，key为元素,value为出现次数
for item in history_nodup:
 count_set[item] = history.count(item)
sorted_history = sorted(count_set.items(), key=operator.itemgetter(1))

max = []  # 存放最后的结果
for item in sorted_history[::-1]: # 按value值从大到小排序
 max.append(item[0])
max = max[0:5]
```

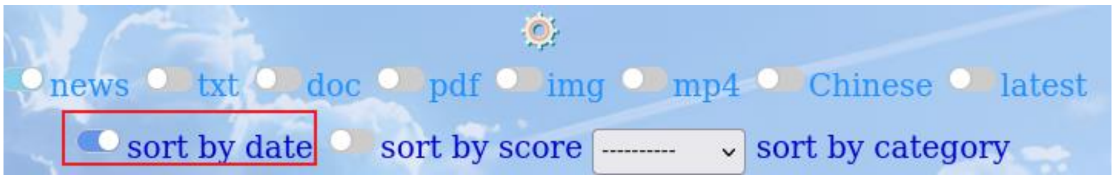
And send it to the frontend:

```
 else: #无black，正常 (tuple不可修改)
  context = {'results': results[0],  'search_term': search_term, 'suggestion': results[1],
'completion': completion, 'recommendation': results[3], 'latest': results[4], 'history':
history_nodup, 'hot': max , 'black_word': black_word,  "other_site": other_site, "date":
date,"settings":settings,"no_results":no_results}
```

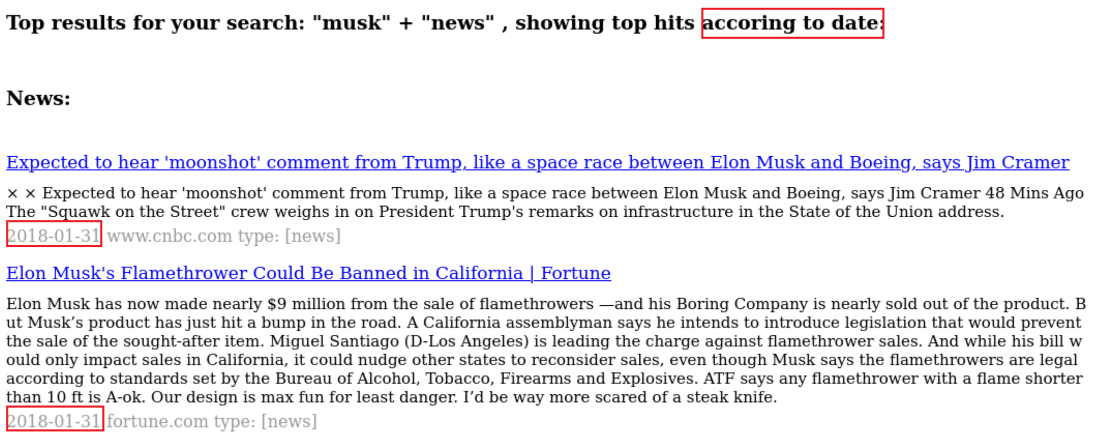
⑫ Ranking Results with Item Date (more recent items get higher weight):

■ Frontend:

We can choose to sort the results by date in settings (default setting is "**sort by score**")



Then we can find our results are sorted by the date of the source news (the newest date of our source data is **2018-01-31**):



**Top results for your search: "musk" + "news" , showing top hits accoring to date:**

**News:**

Expected to hear 'moonshot' comment from Trump, like a space race between Elon Musk and Boeing, says Jim Cramer

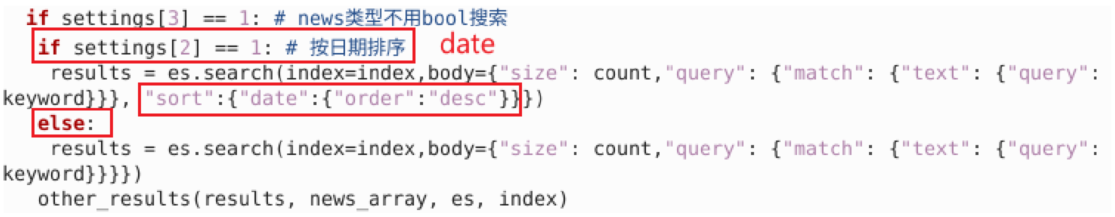
× × Expected to hear 'moonshot' comment from Trump, like a space race between Elon Musk and Boeing, says Jim Cramer 48 Mins Ago The "Squawk on the Street" crew weighs in on President Trump's remarks on infrastructure in the State of the Union address.
2018-01-31 www.cnbc.com type: [news]

Elon Musk's Flamethrower Could Be Banned in California | Fortune

Elon Musk has now made nearly $9 million from the sale of flamethrowers —and his Boring Company is nearly sold out of the product. But Musk's product has just hit a bump in the road. A California assemblyman says he intends to introduce legislation that would prevent the sale of the sought-after item. Miguel Santiago (D-Los Angeles) is leading the charge against flamethrower sales. And while his bill would only impact sales in California, it could nudge other states to reconsider sales, even though Musk says the flamethrowers are legal according to standards set by the Bureau of Alcohol, Tobacco, Firearms and Explosives. ATF says any flamethrower with a flame shorter than 10 ft is A-ok. Our design is max fun for least danger. I'd be way more scared of a steak knife.
2018-01-31 fortune.com type: [news]

■ Backend:

In the search core function, for each type of search (each type of files), we determine whether to use "sort by date" method instead of use BM25 scores to sort results:

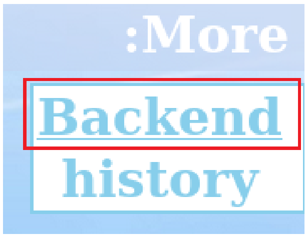


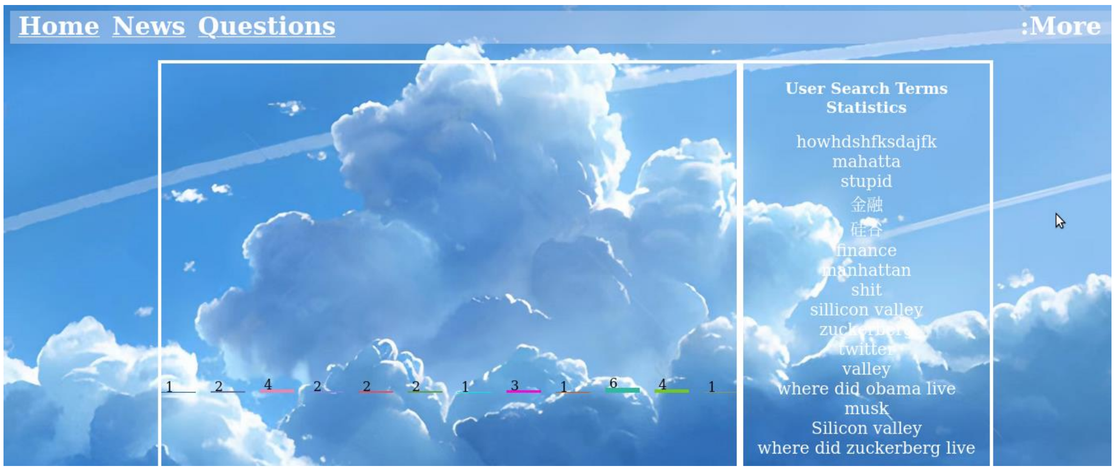
```
 if settings[3] == 1: # news类型不用bool搜索
  if settings[2] == 1: # 按日期排序   date
    results = es.search(index=index,body={"size": count,"query": {"match": {"text": {"query":
keyword}}}, "sort":{"date":{"order":"desc"}}})
  else:
    results = es.search(index=index,body={"size": count,"query": {"match": {"text": {"query":
keyword}}}})
   other_results(results, news_array, es, index)
```

⑬ Real-time Monitoring of the Search Terms (what people searching for now):

■ Frontend:

In each page of our search engine, in the "**More**" menu, we can click "**backend**" to move to the **search term statistics** page, which can visualize the search time of each search term by **histogram**:

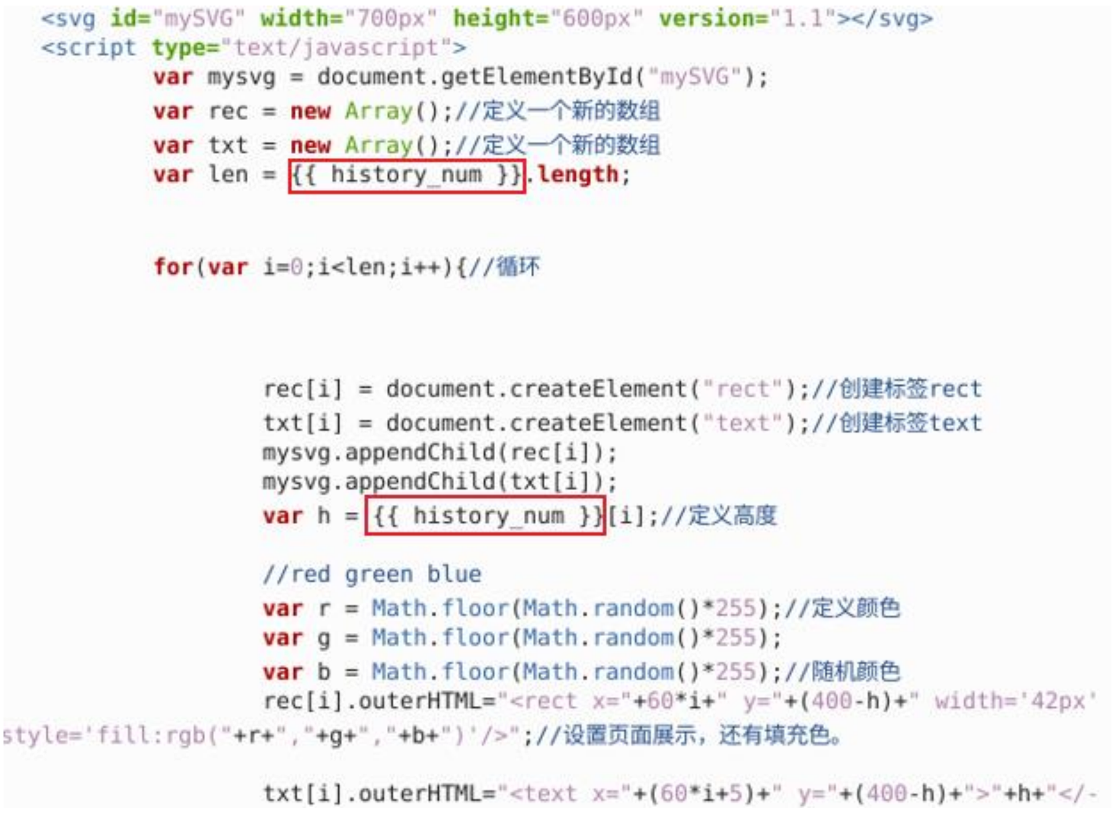




User Search Terms Statistics

howhdshfksdajfk
mahatta
stupid
金融
硅谷
finance
manhattan
shit
sillicon valley
zuckerberg
twitter
valley
where did obama live
musk
Silicon valley
where did zuckerberg live

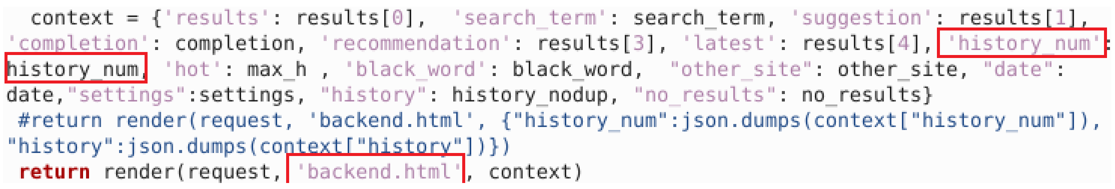
In "**backend.html**", we use **JavaScript** to draw histograms for each search terms in **history**:



```
 <svg id="mySVG" width="700px" height="600px" version="1.1"></svg>
 <script type="text/javascript">
         var mysvg = document.getElementById("mySVG");
         var rec = new Array();//定义一个新的数组
         var txt = new Array();//定义一个新的数组
         var len = {{ history_num }}.length;


         for(var i=0;i<len;i++){//循环


                 rec[i] = document.createElement("rect");//创建标签rect
                 txt[i] = document.createElement("text");//创建标签text
                 mysvg.appendChild(rec[i]);
                 mysvg.appendChild(txt[i]);
                 var h = {{ history_num }}[i];//定义高度

                 //red green blue
                 var r = Math.floor(Math.random()*255);//定义颜色
                 var g = Math.floor(Math.random()*255);
                 var b = Math.floor(Math.random()*255);//随机颜色
                 rec[i].outerHTML="<rect x="+60*i+" y="+(400-h)+" width='42px'
style='fill:rgb("+r+","+g+","+b+")'/>";//设置页面展示，还有填充色。

                 txt[i].outerHTML="<text x="+(60*i+5)+" y="+(400-h)+">"+h+"</-
```

■ Backend:

In "**views.py**", we write a new function for "**backend.html**" to send the number of each search term in "**history**" to the frontend, which is helpful for generating histograms:

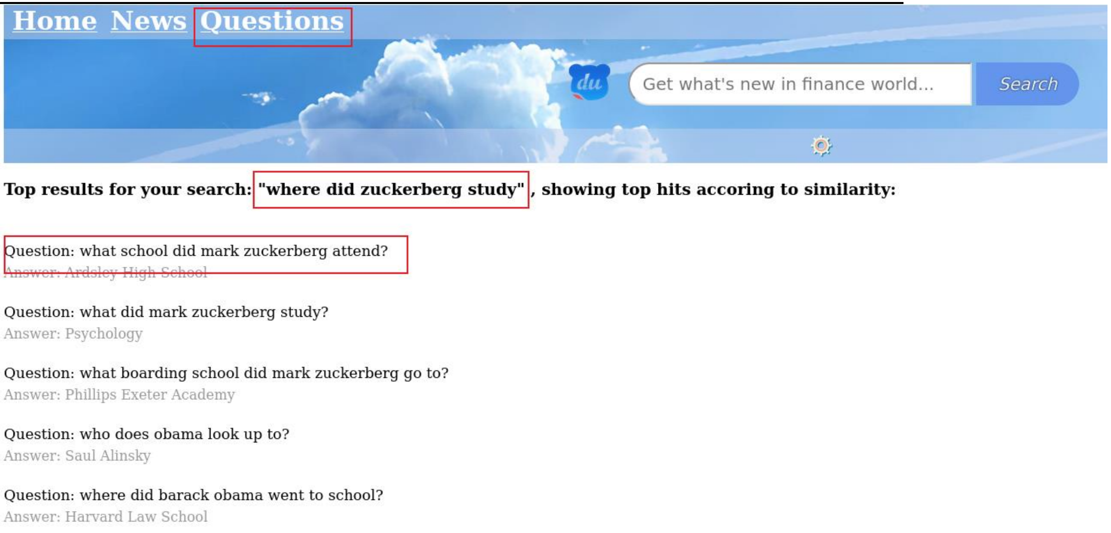


```
  context = {'results': results[0],  'search_term': search_term, 'suggestion': results[1],
'completion': completion, 'recommendation': results[3], 'latest': results[4], 'history_num':
history_num, 'hot': max_h , 'black_word': black_word,  "other_site": other_site, "date":
date,"settings":settings, "history": history_nodup, "no_results": no_results}
 #return render(request, 'backend.html', {"history_num":json.dumps(context["history_num"]),
"history":json.dumps(context["history"])})
 return render(request, 'backend.html', context)
```

⑭ Semantic Search: Use an Embedding Model for Word/Sentence Semantic Search:

■ Frontend:

In "**qa.html**" (by clicking "**Questions**" on the top of each page to move to), we can search for questions related to finance, and get the most similar questions and answers in our index:



Top results for your search: "where did zuckerberg study", showing top hits accoring to similarity:

Question: what school did mark zuckerberg attend?
Answer: Ardsley High School

Question: what did mark zuckerberg study?
Answer: Psychology

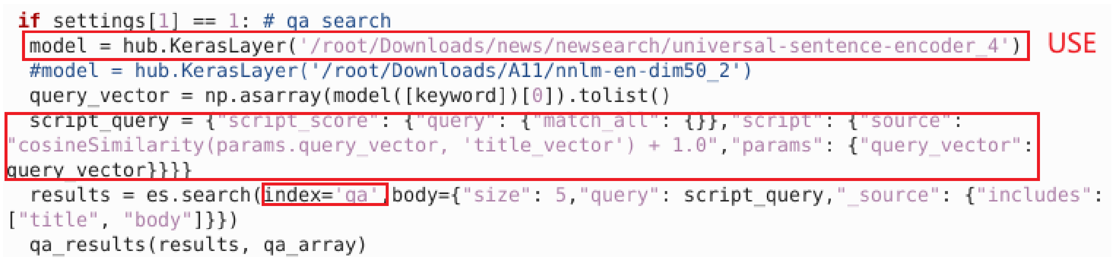
Question: what boarding school did mark zuckerberg go to?
Answer: Phillips Exeter Academy

Question: who does obama look up to?
Answer: Saul Alinsky

Question: where did barack obama went to school?
Answer: Harvard Law School

- Backend:

In the **search core function**, if we get the **"settings"** of **"qa"** search, we will transform the search term into **embeddings** by model *(HINT: here we use the **"USE"** model which has **512 dimensions**, but in the presentation, we chose to use **"nnlm"** model which has only **50 dimensions** due to the memory reason, so the results were not accurate enough. **However, we now have changed it back to USE and it's much more accurate now!**)*, and use a **customized script** to compare the search term with **title vectors** in the index by **cosine similarity**:



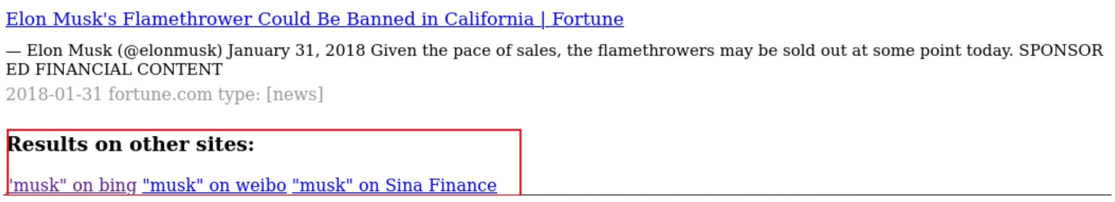
```
if settings[1] == 1: # qa search
 model = hub.KerasLayer('/root/Downloads/news/newsearch/universal-sentence-encoder_4')
 #model = hub.KerasLayer('/root/Downloads/A11/nnlm-en-dim50_2')
 query_vector = np.asarray(model([keyword])[0]).tolist()
 script_query = {"script_score": {"query": {"match_all": {}},"script": {"source":
"cosineSimilarity(params.query_vector, 'title_vector') + 1.0","params": {"query_vector":
query_vector}}}}
 results = es.search(index='qa',body={"size": 5,"query": script_query,"_source": {"includes":
["title", "body"]}})
 qa_results(results, qa_array)
```

⑮ Aggregation Search: Aggregate Search Results from Other Websites, e.g. weibo:

- Frontend:

In each page which contains search results, we have an area called

**"Results on other sites"**, which contains **external links** to results of the same search term on the other sites such as **bing**, **weibo**, and **Sina Finance**:

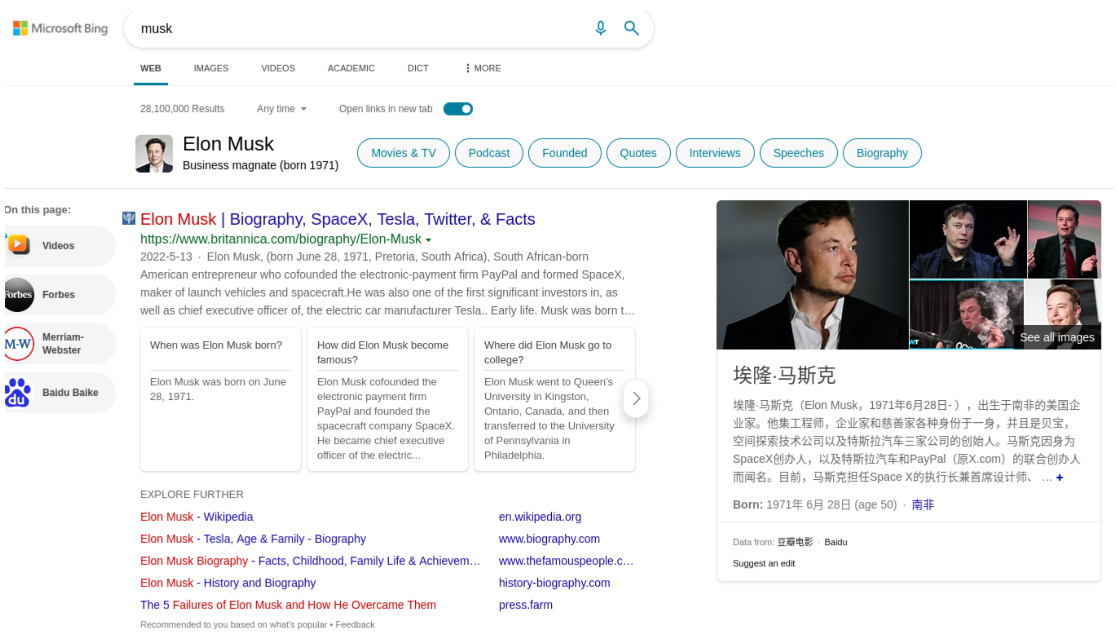


Elon Musk's Flamethrower Could Be Banned in California | Fortune

— Elon Musk (@elonmusk) January 31, 2018 Given the pace of sales, the flamethrowers may be sold out at some point today. SPONSORED FINANCIAL CONTENT
2018-01-31 fortune.com type: [news]

**Results on other sites:**

"musk" on bing "musk" on weibo "musk" on Sina Finance



Microsoft Bing | musk

WEB IMAGES VIDEOS ACADEMIC DICT ⋮ MORE

28,100,000 Results Any time Open links in new tab

Elon Musk
Business magnate (born 1971)

Movies & TV | Podcast | Founded | Quotes | Interviews | Speeches | Biography

On this page: Videos, Forbes, Merriam-Webster, Baidu Baike

Elon Musk | Biography, SpaceX, Tesla, Twitter, & Facts
https://www.britannica.com/biography/Elon-Musk
2022-5-13 · Elon Musk, (born June 28, 1971, Pretoria, South Africa), South African-born American entrepreneur who cofounded the electronic-payment firm PayPal and formed SpaceX, maker of launch vehicles and spacecraft.He was also one of the first significant investors in, as well as chief executive officer of, the electric car manufacturer Tesla.. Early life. Musk was born t...

When was Elon Musk born?
Elon Musk was born on June 28, 1971.

How did Elon Musk become famous?
Elon Musk cofounded the electronic payment firm PayPal and founded the spacecraft company SpaceX. He became chief executive officer of the electric...

Where did Elon Musk go to college?
Elon Musk went to Queen's University in Kingston, Ontario, Canada, and then transferred to the University of Pennsylvania in Philadelphia.

EXPLORE FURTHER

| | |
|---|---|
| Elon Musk - Wikipedia | en.wikipedia.org |
| Elon Musk - Tesla, Age & Family - Biography | www.biography.com |
| Elon Musk Biography - Facts, Childhood, Family Life & Achievem... | www.thefamouspeople.c... |
| Elon Musk - History and Biography | history-biography.com |
| The 5 Failures of Elon Musk and How He Overcame Them | press.farm |

Recommended to you based on what's popular • Feedback

See all images

埃隆·马斯克

埃隆·马斯克（Elon Musk，1971年6月28日-），出生于南非的美国企业家。他集工程师，企业家和慈善家各种身份于一身，并且是贝宝，空间探索技术公司以及特斯拉汽车三家公司的创始人。马斯克因身为SpaceX创办人，以及特斯拉汽车和PayPal（原X.com）的联合创办人而闻名。目前，马斯克担任Space X的执行长兼首席设计师、 ... +

Born: 1971年 6月 28日 (age 50) · 南非

Data from: 豆瓣电影 · Baidu

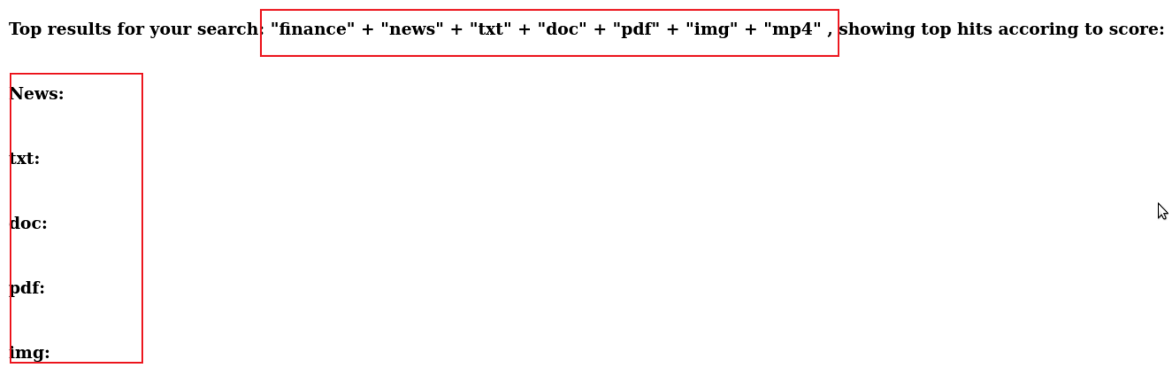
Suggest an edit

While there are results, we will show that using codes:

```
<div>
<h3 style=" margin-top:50px;">Results on other sites:</h3>
</div>
<div>
<a href="https://cn.bing.com/search?q={{search_term}}">"{{ search_term }}" on bing</a>
<a href="https://s.weibo.com/weibo?q={{search_term}}">"{{ search_term }}" on weibo</a>
<a href="http://biz.finance.sina.com.cn/suggest/lookup_n.php?-
q={{search_term}}">"{{ search_term }}" on Sina Finance</a>
</div>
```

⑯ Mixed search: Our Search Results is a Combination of BM25, Semantic, Aggregation, Suggestion, Recommendation, and so on. And we can define the final combination:

In summary, as you can see above, our search engine supports a combination of **BM25**, **semantic**, **aggregation** of other sites, **suggestion** and **recommendation** search, and we can search for different types of data such as **news**, **txt/doc/pdf files**, **images**, and **videos** together in the same time, which means we implemented **“mixed search”** for it.



Top results for your search: "finance" + "news" + "txt" + "doc" + "pdf" + "img" + "mp4" , showing top hits accoring to score:

News:

txt:

doc:

pdf:

img:

## 4. Conclusion and Reflection

To achieve our goal that is to implement a system that can retrieve multiple types results related to the query and provide users a good place to learn financial news, we used Elasticsearch with Python, calculate the BM25 scores and sorted the results. For a better user experience, we also built a Django web server for users with 16 advanced functions.

In the development process we have come up with some “good” ideas to implement this system:

1) All the 16 functions:

   The required functions such as **history**, **auto completion** can extremely improve the experience of users to search.

2) Customized search settings:

   Users can **customize their search** using the detailed settings, and get the combination of results in what they want, such as **“news + txt + img + mp4”** to get them in the same time, and hide/show some of them.

3) Highly integrated and scalable backend settings:

   We use a **“setting”** in the backend to control all of the search settings, such as

**types of files, the sorting method (score/date/category), QA search and Chinese search**. It's very easy to control them (just need to change or read a bit in the list) or add new settings in them.

4) Well-designed web interface:
   - Our web interface has **4 different pages** including a home page, two search pages and a backend page, which is easy to use.
   - The **history** and **backend** are hide as a drop-down menu at the top-right side.
   - And the **settings** can be hide and show, which is very use-friendly.
   - Besides, the whole website is designed into **blue and white** style, all buttons, icons and texts are harmonized and well-designed.

However, we also realized some problems that are not "good" enough:

1) The Accuracy of Semantic Search and Memory Problem:
   - Because of the memory problem, USE is not useful in most time, which leads to a low accuracy of our results in presentation.
   - However, if we change the model back to USE, the speed of the website will become much slower and the Elasticsearch will shut down frequently.
   - We think the best solution is to change to a **high-performance server or cluster** to run the program because in real case the data are even larger and personal computer definitely could not hold it.
   - But actually, we do think that 16GB RAM is enough for this program with only 5GB data, so maybe we can **improve our algorithm in some parts** especially for **crawling data** in real time to improve the performance.

In the future researches and development, we will work hard on those problems to improve our system and try to provide a better search engine.

## 5. Reference

Gheorghe, R., Hinman, M. L., & Russo, R. (2015). *Elasticsearch in action*. Manning.

Dixit, B. (2016). *Elasticsearch Essentials*. Packt Publishing Ltd.

Divya, M. S., & Goyal, S. K. (2013). *ElasticSearch: An advanced and quick search technique to handle voluminous data*. Compusoft, 2(6), 171.