BÁO CÁO BÀI TẬP COLAB 4
CRAWL DATA

-Lớp: Máy học - CS114.L22.KHCL
-Giảng viên hướng dẫn:
- PGS.TS. Lê Đình Duy
- Ths. Phạm Nguyễn Trường An
-Nhóm thực hiện:
- Thái Trần Khánh Nguyên - 19520188
- Nguyễn Khánh Như - 19520209
- Đoàn Nguyễn Nhật Quang - 19520235

1. Phân tích bài toán:

Đề bài:

- Mỗi nhóm chọn 03 trang báo điện tử châm biếng tiếng Anh từ danh sách sau: https://en.wikipedia.org/wiki/List_of_satirical_news_websites
- Và 03 trang báo điện tử uy tín tiếng anh từ 3 quốc gia nói tiếng Anh khác nhau.
- Thu thập tất cả tiêu đề của các bài báo mà 6 trang tin trên đăng trong vòng 03 năm trở lại đây.
- Tổ chức dataset theo cùng format với dataset tham khảo ở đây: https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection
- Các nhóm được hợp tác với nhau để làm bài tập này. Tuy nhiên mỗi trang tin không được có quá 4 nhóm cùng chọn. Các nhóm phải ghi rõ các danh sách 6 trang tin mình đã chọn trong comment.

- **Bài nộp bao gồm:**
  - Notebook chứa các code cần thiết để crawl dữ liệu.
  - Báo cáo kết quả quá trình thu thập dữ liệu.
  - File json theo đúng format.

Phân tích:

- Sau khi tham khảo link thầy gửi thì nhóm em nhận thấy dữ liệu cần crawl từ các trang web là:
  - is_sarcastic: 1 nếu là báo châm biếm còn lại là 0
  -headline: tiêu đề của bài báo
  -article_link: link dẫn đến bài báo

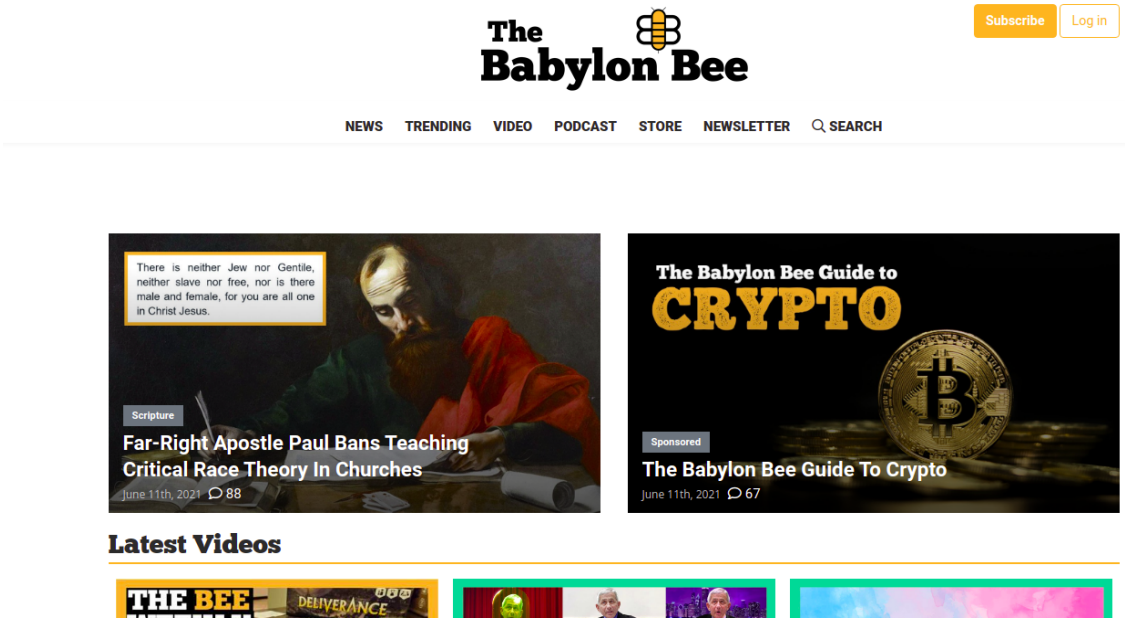
-Sau khi tham khảo các web châm biếm và các web báo chính thống thì nhóm em đã chọn ra 6 trang web để crawl dữ liệu:
  - Trang báo châm biếm:

+ The Babelon Bee
+ The Beaverton
+ The Chaser
  - Trang báo chính thống:

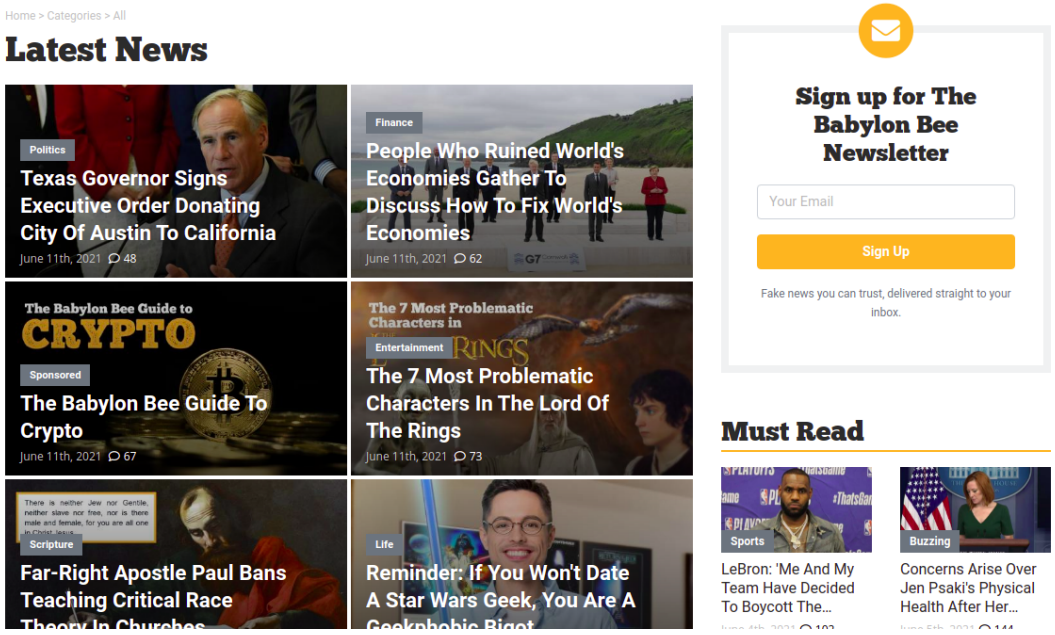
+ The Guardian
+ NBC News
+The Sun

2. Phân tích các trang web cần crawl:

Sau khi tìm hiểu trên website [The Babelon Bee]() nhóm em có một vài nhận xét như sau:

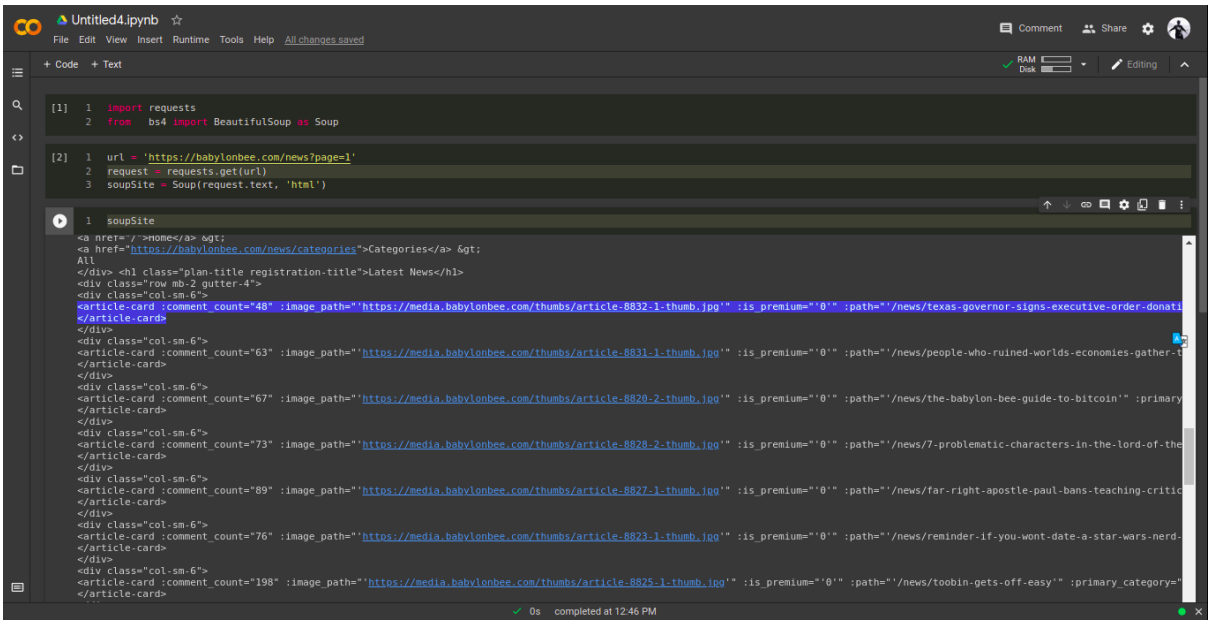


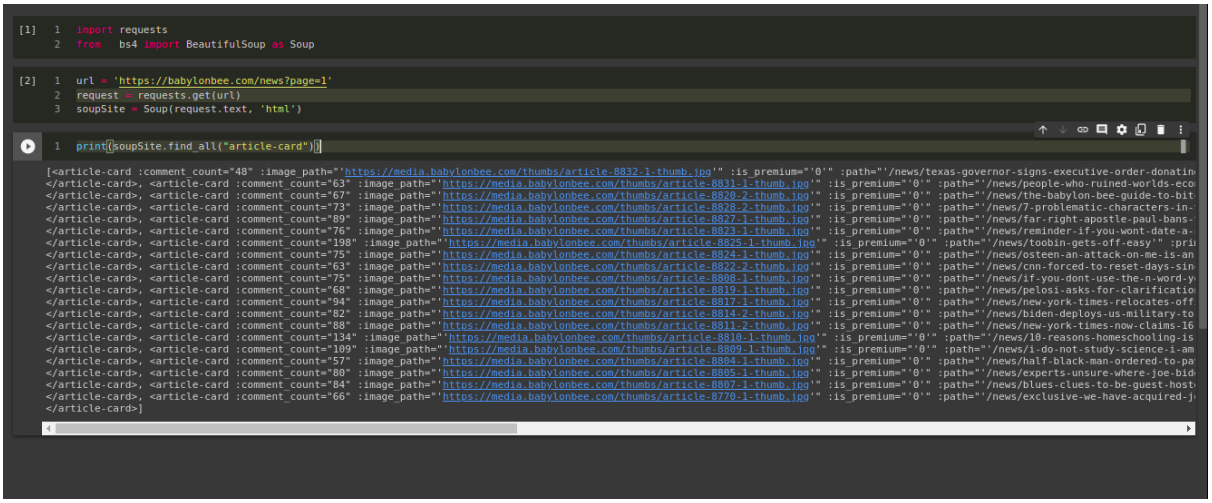
- Để có thể xem được danh sách bài báo đã được đăng thì website này có cung cấp đường dẫn có dạng :
  "babylonbee.com/news?page=" + số thứ tự của trang cần xem danh sách
  Ví dụ muốn xem danh sách các bài báo ở trang số 1 thì đường link sẽ là:
  "https://babylonbee.com/news?page=1"
- Và các bài báo ở đây được sắp xếp theo dạng newest:



- Điều nhóm em cần quan tâm đến là headline của bài báo, article link và thời gian đăng để khớp với yêu cầu các bài báo trong vòng 3 năm gần đây. Do đây là trang web báo châm biếm nên hiển nhiên `is_sarcastic = 1`

- Sau khi xem thử file html của web thì nhóm em cũng đã phát hiện ở thẻ article-card có lưu tất cả nhưng thông tin mà chúng em cần





-Chúng em xây dựng thêm các hàm để có thể lấy được dữ liệu mình cần:

```
# Dùng để gửi request đến trang web với tham số là  số thứ tự trang cần thu thập
def sendRequest(page):
 #Tạo đường liên kết đến trang cần thu thập dữ liệu
 url = "https://babylonbee.com/news?page=" + str(page)
 #Gửi request đến trang đó
 request = requests.get(url)
 #BeautifulSoup dùng để phân tích dữ liệu html thành dữ liệu cây để chúng ta dễ dàng thao tác sau này
 soupSite = Soup(request.text, 'html.parser')
 return soupSite
```

```
def getData(soupSite, data):
 count = 0
 for soup in soupSite:
   soup = str(soup)

   #check Time
   a = soup.find(':published_on=')
   b = soup.find('\'" ', a, len(soup))
   if (2019 <= int(soup[b-4:b])):
     # Lấy link
     posArticleLinkStart = soup.find(":path")
     posArticleLinkEnd = soup.find('\'" ', posArticleLinkStart, len(soup))
     ArticleLink ="https://babylonbee.com/" +
soup[posArticleLinkStart+8:posArticleLinkEnd]

     # Lấy Title
     posTitleStart = soup.find(':title=') + 8
     posTitleEnd = soup.find('>', posTitleStart, len(soup))
     Title = soup[posTitleStart:posTitleEnd].replace("&quot;",
"").replace("'","").replace('"', '')

     #Thêm phần vừa get được vào data chính
     data.append([Title, ArticleLink, 1])
     count += 1
 return count
```

```
# Hàm dùng để xuất file json theo yêu cầu của đề bài
def writeJson(data, fileName):
 with open(fileName+'.json', 'w') as f:
   for i in data:
     f.write('{"is_sarcastic": ' + str(i[2]) + ', "headline": "' + i[0] + '",
"article_link": "' + i[1] +'"}\n')
```

```
#Thêm thư viện tqdm để theo dõi quá trình crawl dữ liệu
def crawlDataBabylonBee():
 data = []
 count = 0

 print("___ CRAWL DATA FROM BABYLON BEE ___")
 for page in tqdm(range(1, 355)):
   soupSite = sendRequest(page).find_all("article-card")
   count += getData(soupSite, data)
 writeJson(data, 'BabylonBee')

 print('Completed...')
 print("Crawled:", count, "ArticleLinks")
```

- Các trang web khác cũng tương tự như vậy nhưng thông tin sẽ lưu tên khác nhau, code chi tiết em đã nộp trong file notebook kèm theo.
-Tuy nhiên trang web The Sun có tính năng chặn request ẩn danh nên nhóm em đã thêm headers vào để có thể crawl dữ liệu

3. Kết quả thu được:

Sau khi crawl hoàn tất 6 trang web trên thì nhóm em thu được kết quả:

Trang web	Label	Số tập dữ liệu crawl được
The Babelon Bee	1	4423
The Beaverton	1	2333
The Chaser	1	1434
The Guardian	0	1313
NBC News	0	49609
The Sun	0	35922
Tổng cộng:		95034