Báo cáo thu thập dataset cho bài toán "Sarcasm detection in news headline"

Thành viên nhóm

19521731 Nguyễn Đại Kỳ

19521225 Văn Viết Hiếu Anh

19522054 Lê Văn Phước

Những trang nhóm lấy dữ liệu:

- Trang chính thống:
  - https://www.theguardian.com/
  - https://www.cbsnews.com
  - https://www.theaustralian.com.au/
- Trang châm biếm:
  - https://clickhole.com/
  - https://thehardtimes.net/
  - https://www.thepoke.co.uk/
  - https://babylonbee.com/

Tổng quan dữ liệu thu thập

Gồm **174.753** records từ (Dữ liệu được lưu ở file `data.json` hoặc `data.csv` ở cùng thư mục với file báo cáo)

Trang	Số lượng	Thể loại	Thành viên
https://www.theguardian.com/	34230	Chính thống	Văn Viết Hiếu Anh
https://www.cbsnews.com	60735	Chính thống	Văn Viết Hiếu Anh
https://www.thehardtimes.net	5917	Châm biếm	Lê Văn Phước
https://www.theaustralian.com.au	50151	Chính thống	Lê Văn Phước
https://www.clickhole.com	1771	Châm biếm	Nguyễn Đại Kỳ
https://www.thepoke.co.uk	1157	Châm biếm	Nguyễn Đại Kỳ
https://www.babylonbee.com	7043	Châm biếm	Nguyễn Đại Kỳ
https://www.newyorker.com	13704	Châm biếm	Nguyễn Đại Kỳ

Dataset Format

```
[
    {
        "article_link":
"https:\/\/www.theguardian.com\/law\/2021\/jun\/08\/powerful-new-watchdog-will-
target-unscrupulous-employers-says-no-10",
        "headline": " \u2018Powerful\u2019 new watchdog will target unscrupulous
employers, says No 10",
        "posted_at": "06-07-2021",
        "is_sarcastic": 0
    },
    ...
]
```


article_link Đường dẫn đến bài viết

headline Tiêu đề bài viết

posted_at Ngày bài viết được đăng

is_sarcastic Bài Viết có phải là châm biếm ?

Quá trình thu thập và xử lý

19521225 - Văn Viết Hiếu Anh

Thiết lập

Công cụ sử dụng **Scrapy**

`pipelines.py`

```
class csvWriterPipeline:
    def open_spider(self, spider):
        self.file = csv.writer(open('data.csv', 'a'), lineterminator='\n')

    def close_spider(self, spider):
        self.file.close()

    def process_item(self, item, spider):
        self.file.writerow(item.values())
        return item
```

Các bước thu thập

1. Sử dụng *JavaScript* trên trình duyệt để gom tất cả những đường dẫn đến các category của trang báo:

    ```
    # Các đường dẫn đã gom được
    start_urls = ['https://www.theguardian.com/world/coronavirus-outbreak/all',
                  'https://www.theguardian.com/world/all',
                  'https://www.theguardian.com/science/all',
                    ...
                  'https://www.theguardian.com/lifeandstyle/all',
                  'https://www.theguardian.com/commentisfree/all'
        ]
    ```

2. Với mỗi đường dẫn lấy ra tất cả thẻ trong wrapper* ta được 1 bài viết.

- Với mỗi bài viết lấy các thông tin cần thiết.
- Kiểm tra ngày đăng nếu là ngày 31/12/2017 thì dừng việc thu thập.
- Dùng css Selector tìm đường dẫn đến trang tiếp theo, nếu có trang tiếp theo thì tiếp thục thu thập, nếu không thì dừng lại

  **chỉ lấy trong wrapper vì nếu lấy ngoài có thể dính một số trang khác ví dụ trang tin nổi bật, tình trạng này xảy ra ở 1 số trang ví dụ CBS News, cấu trúc thẻ HTML của tin nổi bật giống hoàn toàn thẻ tin tức của category đó gây trùng lặp khi thu thập dữ liệu*

```
    def parse(self, response):
        for article in response.css('.fc-slice-wrapper ul >li.u-faux-block-
link'):

            article_link = article.css(
                'a.u-faux-block-link__overlay::attr(href)').get()
            headline = article.css('span.js-headline-text::text').get()
            posted_at = article.css('time::attr(datetime)').get()

            if '2017-12-31' == posted_at[:10]:
                return

            yield{
                'article_link': article_link,
                'headline': headline,
                'posted_at': posted_at,
                'is_sarcastic': 0
            }

        next_page = response.css('a[aria-label=" next page"]').attrib['href']
        if next_page is not None:
            yield response.follow(next_page, callback=self.parse)
```

Chuẩn hóa: `standardize.ipynb`

Vì những dữ liệu `03 Jan` (nghĩa là 03/01/2021) không thể chuyển về dạng datetime bằng hàm của `pandas` nên phải sử dụng hàm dưới đây để chuẩn hóa về dạng mà `pandas` có thể đọc được

```
def thisYear(date):
    # Find date of this year like "03 Jan" in CBS news
    pattern = re.compile(r'\w{3} \d{,2}$')

    # If match to pattern add 2021 to the end, else return itself
    if bool(pattern.match(date)):
        return date + ', 2021'

    return date

# Apply above function to posted_at column
df['posted_at'] = df['posted_at'].apply(thisYear)
```

Dạng dữ liệu còn lại là `<time> ago` vốn chính là ngày thu thập dữ liệu nên đổi thành string `today` để `pandas` có thể đọc được

```
# Turn posted_at to datetime. All error with be turn to NaT for the reason it the time like "16 hours ago" which mean today, so we will fill it with today
df['posted_at'] = pd.to_datetime(df['posted_at'], errors='coerce', utc=True)
df['posted_at'] = df['posted_at'].fillna(pd.to_datetime('today'))
df['posted_at'] = pd.to_datetime(df['posted_at'], errors='coerce', utc=True)
```

Chuyển toàn bộ datetime về 1 format cố định

```
# Turn all date to same format
df['posted_at'] = df['posted_at'].dt.strftime('%d/%m/%Y')
```

Lê Văn Phước

Công cụ sử dụng **Python Crawl** dữ liệu

Các bước thu thập

1. Lấy tất cả các đường link đến các category của trang báo cần lấy dữ liệu:

   ```
   # Các đường link đã gom được:
   list_urls = ['https://www.theaustralian.com.au/nation/politics',
                   'https://www.theaustralian.com.au/business/economics,
                   'https://www.theaustralian.com.au/sport/football,
                   ...
                   'https://thehardtimes.net/culture/',
                   'https://thehardtimes.net/music/'
      ]
   ```

2. Từ các đường link trên ta tiến hành lấy dữ liệu

   ```
   response = requests.get('https://www.theaustralian.com.au/nation/politics')
   ```

3. Tách dữ liệu vừa lấy

   ```
   soup = BeautifulSoup(response.content, "html.parser")
   ```

4. Phân tích dữ liệu vừa lấy

   Ta tiến hành lên trang web xác định data cần lấy trong trang web, từ đó lấy các thẻ và class chứa data hoặc link các bài báo cần lấy.

   Xác định các trường cần lấy trong mỗi bài báo:

   ```
   - article_link
   - headline
   - posted_at
   - is_sarcastic
   ```

   ```
   titles = soup.findAll('h3', class_='story-block__heading')
   links = [link.find('a').attrs["href"] for link in titles]
   headlines = [headline.find('a').text for headline in titles]
   publication_dates = soup.findAll('span', class_='show-for-xlarge')
   dates = [date.text for date in publication_dates]
   ```

5. Lưu data vừa lấy vào file

Nguyễn Đại Kỳ

Công cụ sử dụng **Selenium** và **Beautiful Soup**

Các bước thu thập

1. Lấy tất cả các đường link đến các category của trang báo cần lấy dữ liệu:

```
# Các đường link đã gom được:
list_urls = [
  'https://clickhole.com/category/news/',
  'https://www.thepoke.co.uk/category/news/',
  'https://babylonbee.com/news',
  'https://www.newyorker.com/latest/news',
   ]
```

2. Từ các đường link trên ta tiến hành lấy dữ liệu

- Truy cập web theo link.
- Duyệt qua danh sách các page chứa dữ liệu cần crawl của web.
- Dùng css selector để lấy ra nội dung quan tâm.
- Lưu dữ liệu vào file.

```
def load_page(self):
    browser = self.driver
    j = 1
    while j < 1395:

 browser.get("https://www.thepoke.co.uk/category/news/page/{page}/".format(pa
ge=j))
        time.sleep(self.delay)
        all_data =
browser.find_elements_by_css_selector('div.boxframe.archive > article.boxgrid
> a')
        for i in range(len(all_data)):
            line = {}
            line['article_link'] = all_data[i].get_attribute('href')
            line['headline'] = all_data[i].text.split('\n')[1]
            line['posted_at'] =
self.get_date(all_data[i].get_attribute('href'))
            line['is_sarcastic'] = 1
            with open('crawled_data/ThePoke.txt','a') as filedata:
                json.dump(line,filedata)
                filedata.write('\n')
        browser.execute_script('window.scrollTo(0,
document.body.scrollHeight);')
        time.sleep(self.delay)
        j += 1
```

3. Lấy thông tin ngày đăng

Nếu thông tin ngày đăng không có sẵn ở ngoài bài báo

Thông tin ngày đăng của file BabylonBee em chưa kịp chuẩn hóa

- Truy cập link bài viết.
- Dùng Beautiful Soup để tải html.
- Dùng findAll để tìm ra thông tin quan tâm.

```
def get_date(self,url):
    page = requests.get(url)
    soup = bs(page.text, 'html.parser')
    date_line = str(soup.body.findAll('p',attrs={'class':'byline'}))
    date = date_line[(date_line.find('Updated')+8):-5]
    return date
```