I. Tổng quan:

_ Ngôn ngữ sử dụng : Python 3.5

_ Trang web sử dụng để lấy dữ liệu : http://tuoitre.vn/

_ Môi trường : Ubuntu 16.04

_ Các thư viện sử dụng :

- matplotlib : Sử dụng để vẽ đồ thị.
- tkinter : Tạo GUI.
- BeautifulSoup : Decode html byte thành html dạng lxml.
- selenium – webdriver - xvfb : Sử dụng để tạo trình duyệt ảo để lấy dữ liệu từ website.
- requests : Tạo kết nối http đến trang web và gọi phương thức GET để lấy dữ liệu.
- subprocess : Sử dụng terminal của Ubuntu.
- vnTagger : API tách từ và đánh dấu từ phát triển bởi thầy Lê Hồng Phương. URL : http://mim.hus.vnu.edu.vn/phuonglh/softwares.

_ Dữ liệu: Gồm 6750 bài viết chia thành 8 chuyên mục:

- Chính trị: 751 bài
- Giáo dục: 758 bài
- Công nghệ: 749 bài
- Kinh tế: 750 bài
- Pháp luật: 750 bài
- Thể thao: 750 bài
- Văn hóa: 750 bài
- Y tế: 750 bài
- Khác: 750 bài

II. Các bước thực hiện

1. Lấy url các bài viết từ web:

a) Cấu trúc trang báo điện tử Tuổi trẻ Online:

_ Các bài báo được phân thành 14 chuyên mục lớn:

Ảnh 1: Trang chủ Tuổi trẻ Online

_ Mỗi chuyên mục lớn được phân thành các chuyên mục nhỏ hơn:

Ảnh 2: Giao diện chuyên mục Chính trị - Xã hội

_ Các bài viết trong mỗi chuyên mục được tổ chức dưới dạng danh sách, nút ‘Xem thêm’ dùng để lấy thêm dữ liệu bài viết từ CSDL để hiển thị.

Ảnh 3: Cách tổ chức bài viết

b) Các bước lấy URL của bài viết:

_ Thu thập thủ công URL của từng chuyên mục con.

```
tech_links = ['http://nhipsongso.tuoitre.vn/dien-thoai.htm', 'http://nhipsongso.tuoitre.vn/thiet-bi-so.htm',
              'http://nhipsongso.tuoitre.vn/bao-mat.htm', 'http://nhipsongso.tuoitre.vn/tu-van-tieu-dung.htm'
             ,'http://nhipsongso.tuoitre.vn/blog.htm', 'http://nhipsongso.tuoitre.vn/thu-thuat-kien-thuc.htm',
              'http://nhipsongso.tuoitre.vn/thi-truong.htm']
```

Ảnh 4: URL của các chuyên mục con trong chuyên mục Khoa học công nghệ

_ Truy cập vào URL của từng chuyên mục con, lọc URL các bài viết trong các chuyên mục đó theo các bước:

- Sử dụng selenium - webdriver tạo trình duyệt ảo.

```
display = Xvfb()
display.start()
driver = webdriver.Firefox()
```

- Gửi HTTP GET chứa URL của từng chuyên mục trong website, lấy về file HTML tương ứng URL của từng chuyên mục.
- Tìm kiếm class chịu trách nhiệm lấy thêm bài viết từ CSDL và tạo request tới button của class đó (nút 'Xem thêm') để yêu cầu trang web gửi dữ liệu chứa URL của các bài viết trong CSDL đến khi thu thập đủ URL.

```
for url in topic_urls:
    driver.get(url)
    driver.implicitly_wait(5)
    # Request đến nút 'Xem thêm' để lấy link bài viết
    for j in range(page_count):
        driver.find_element_by_class_name(btn_name).click()
```

- Sử dụng BeautifulSoup để chuyển đổi file HTML sang dạng LXML và lọc toàn bộ các URL có trong file HTML.

```
html = driver.page_source
s = bs(html, 'lxml')
soup = str(s.find_all(attrs, {'class' : class_name})[0])
start_idx = find_all_ss('href', soup)
```

_ Lưu URL các bài viết ra file, mỗi chuyên mục có 1 list url các bài viết.

2. Lấy dữ liệu từng bài viết dưới dạng text:

_ Sử dụng thư viện requests để gửi HTTP GET đến với URL của các bài viết để lấy về định dạng HTML của URL đó và lưu vào file.

```
session = requests.session()
adapter = requests.adapters.HTTPAdapter(max_retries=10)
session.mount('http://', adapter)

for i, url in enumerate(urls):
    r = session.get(url.strip())
    html = bs(r.content, 'lxml')
```

_ Lọc tag và những ký tự không cần thiết để thu về dữ liệu bài viết ở dạng gốc.

```
for i in range(0, len(line)):
    # Đánh dấu 1 tag
    if line [i] == '<' and line [i+1] != ' ':
        open_char = open_char + 1
        continue
    # Kết thúc 1 tag
    elif line [i] == '>' and line [i-1] != ' ':
        open_char = open_char - 1
        continue
    # Đọc ký tự
    elif open_char == 0:
        processed_line += line[i]
if len(processed_line) > 0:
    text.append(processed_line)
```

_ Ghi dữ liệu bài viết ra file và lưu vào thư mục của chuyên mục của bài viết.

3. Tiền xử lý dữ liệu:

_ Dùng thư viện subprocess để sử dụng terminal gọi API vnTagger để tách từ và phân từ đối với từng file bài viết.

```
# Đánh dấu từ sử dụng command line
def tag_words(source, des=TEMP_FILE):
    cmd = 'cd ' + TAGGER + ' && ./vnTagger.sh -i {} -o {} -u -p'
    p = subprocess.Popen(cmd.format(source, des), stdout=subprocess.PIPE, shell=True)
    (output, err) = p.communicate()
    return output.decode('utf-8')
```

_ Loại bỏ các ký tự không có nghĩa : **:()<>|?*…"%^!“”…**

```
Lấy/V sự/N tàn_bạo/A của/E chọi/N trâu/N làm_trò/A vui/V lễ_hội/N

Vụ_việc/N trâu/N chọi/N húc/N chết/V chủ/N tại/E lễ_hội/N chọi/N trâu/N
Đồ_Sơn/Np Hải_Phòng/Np và/CC tranh_cãi/V giữ/V hay/V bỏ/V những/L lễ_hội/N
có/V tính/N bạo_lực/N như/C chọi/N trâu/N đã/R nhận/V được/R hàng/R trăm/M
phản_hồi/N của/E độc_giả/N cũng/R như/C ý_kiến/N các/N chuyên_gia/N
Bên_cạnh/N một_vài/N ý_kiến/N cho/V rằng/C không/R nên/V vì/E sự_cố/N đáng/
V tiếc/V trên/N mà/C bỏ/V lễ_hội/N chọi/N trâu/N thì/C đại_đa_số/N độc_giả/
N đều/R đồng_tình/V trong/E xã_hội/N văn_minh/N hiện_đại/N hiện_nay/N không/
R nên/V tiếp_tục/V duy_trì/V những/L lễ_hội/N mang/V tính/N bạo_lực/N như/C
vậy/P
Thật/T dã_man/A quá/R tàn_nhẫn/A
```

Ảnh 5: Bài viết sau khi được xử lý

_ Chia bộ dữ liệu thành 10 folds dùng cho cross validate test.

4. Học máy:

_ Sử dụng Naive Bayes với biến thế Multinomial để tính điểm của bài viết đối với từng chuyên mục và chọn chuyên mục có điểm cao nhất làm nhãn.

_ Bài viết d được gán nhãn theo công thức sau:

$$c_d = \underset{c_j \in C}{argmax}\, P(c_j) \prod_{i \in d} \left(\frac{n_{j,i} + \alpha}{n_j + \alpha N} \right)$$

với

- $P(c_j) = \frac{D_j}{D}$ là xác suất xuất hiện của chuyên mục c_j trong dữ liệu train

- $n_{j,i}$ là số lần xuất hiện của từ i trong các bài viết thuộc chuyên mục c_j
- n_j là tổng số từ có trong từ điển của chuyên mục c_j
- N là tổng số từ có trong từ điển chung của toàn bộ các chuyên mục
- α là chỉ số làm trơn

_Công thức trên được Logarit hóa để đảm bảo độ chính xác.

$$c_d = \underset{c_j \in C}{argmax} \log\left(P(c_j)\right) + \sum_{i \in d} \log\left(\frac{n_{j,i} + \alpha}{n_j + \alpha N}\right)$$

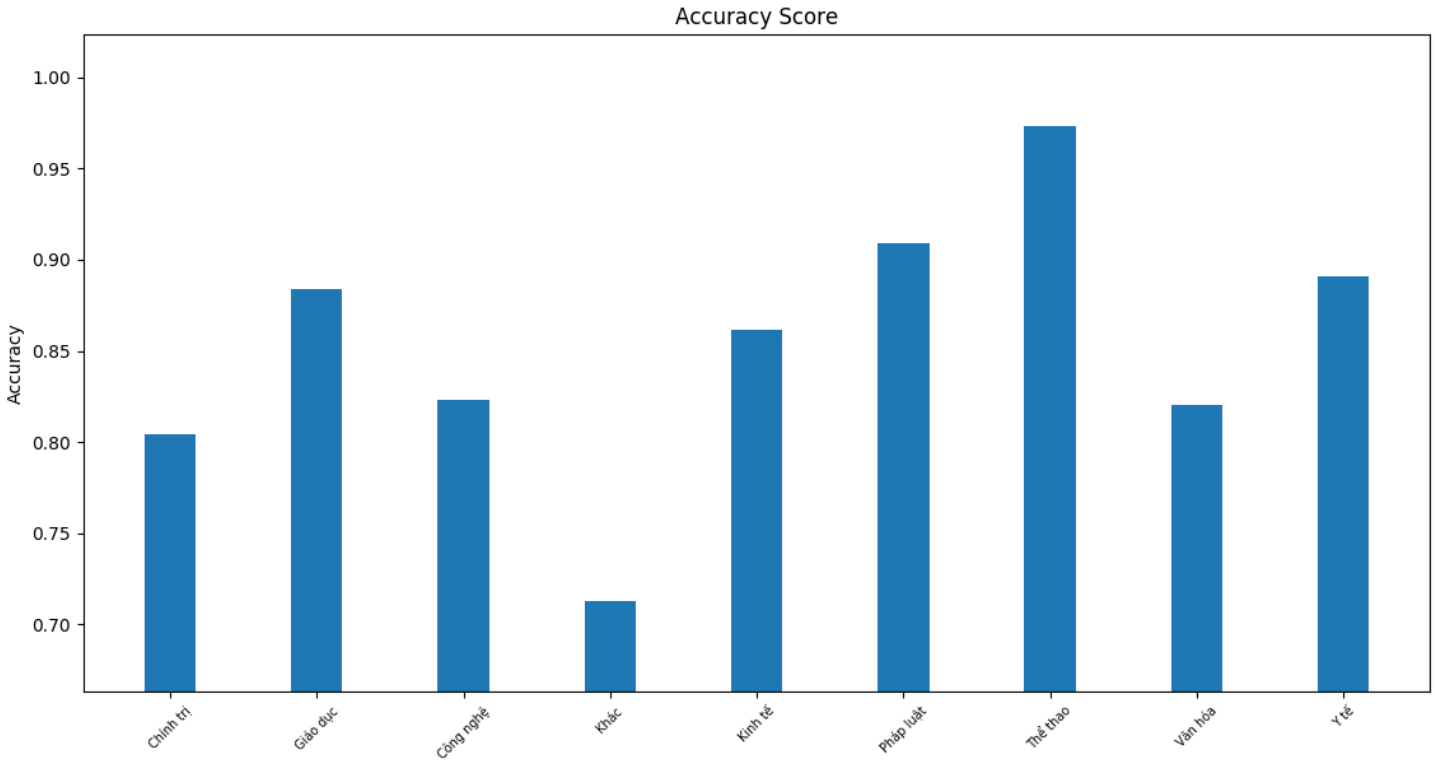
5. Kết quả thực nghiệm:

_ Kết quả trung bình sử dụng phương pháp cross validate với fold = 10:

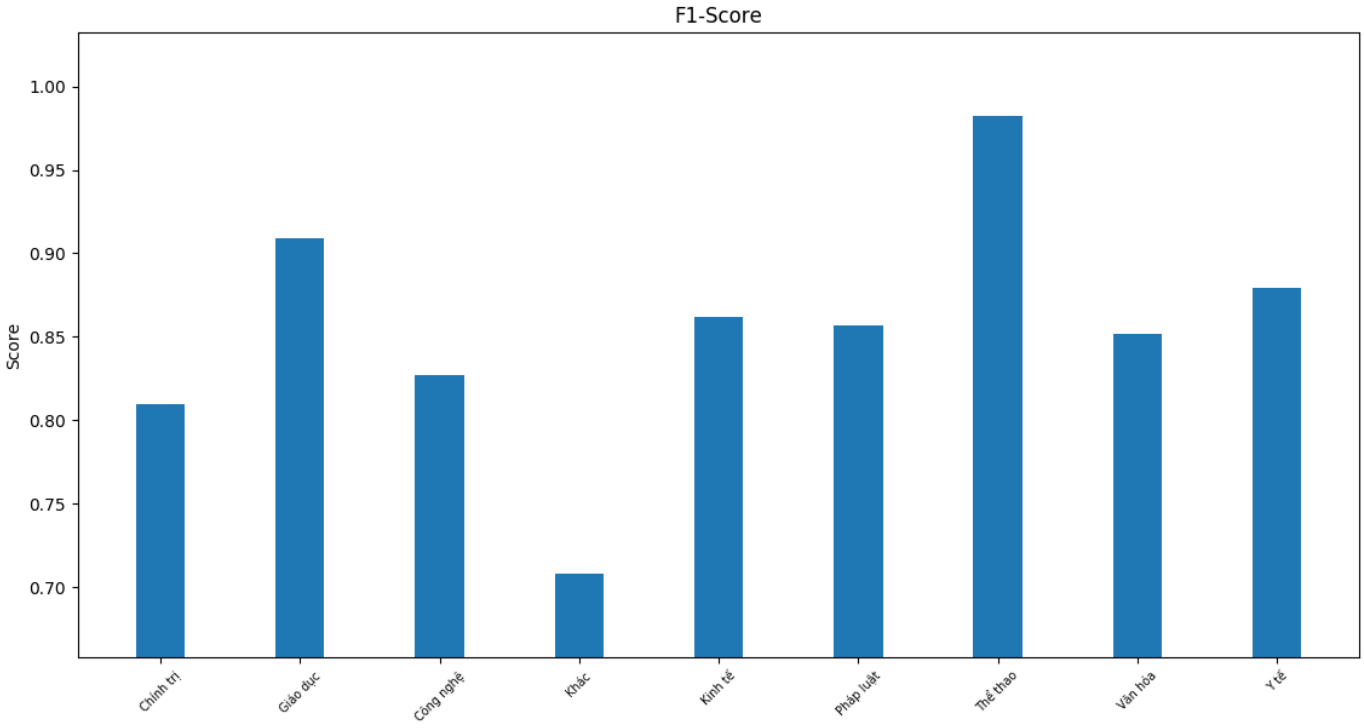
- Accuracy: 85.4 %
- Precision: 85.6 %
- Recall: 85.3 %
- F1-score: 85.4 %

_ Biểu đồ:

- Tỷ lệ phân loại chính xác ở mỗi chuyên mục:



- F1-score ở mỗi chuyên mục:

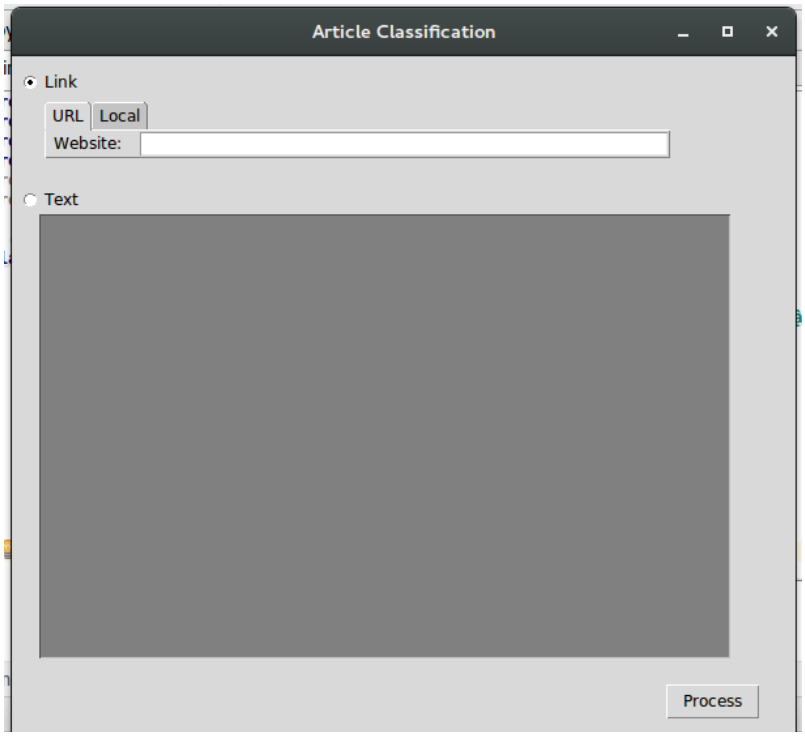


_ Đánh giá:

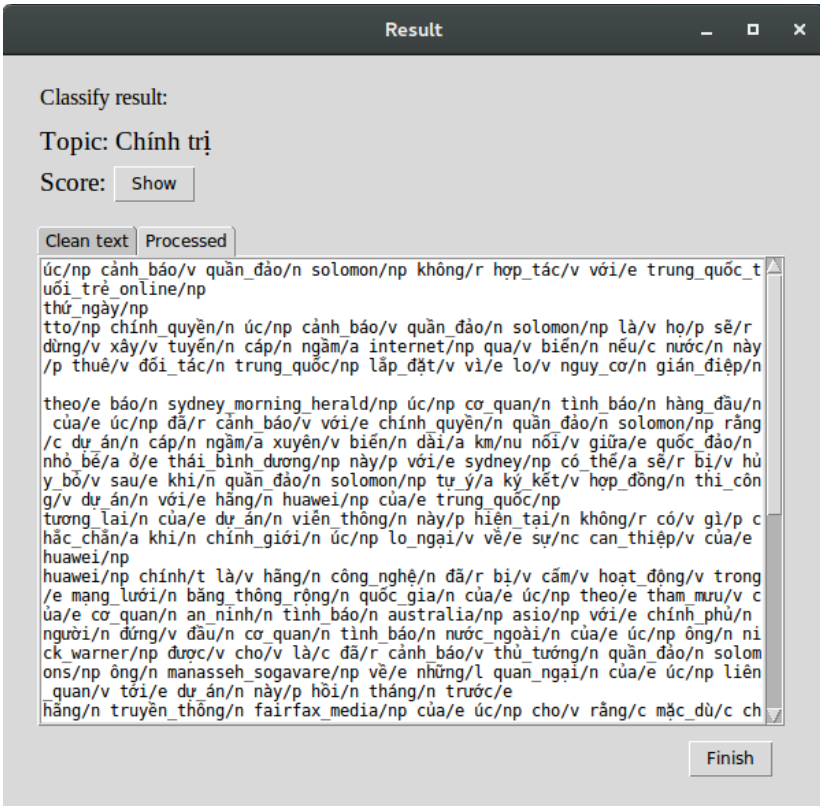
- Chuyên mục ‘Khác’ có điểm Accuracy và F1 thấp nhất vì các bài viết thuộc chuyên mục này có nội dung đa dạng và không có một chủ đề cụ thể.
- Chuyên mục ‘Thể thao’ có điểm số cao nhất vì nội dung khó có thể bị nhầm lẫn với các chuyên mục khác.

6. Giao diện hệ thống:

_ Dữ liệu đầu vào có thể thuộc 3 dạng : Text, Url hoặc đường dẫn đến file trong máy.



_ Giao diện kết quả với input URL : http://tuoitre.vn/tin/the-gioi/20170728/uc-canh-bao-quan-dao-solomon-khong-hop-tac-voi-trung-quoc/1359751.html



_ Biểu đồ điểm của mỗi chuyên mục: