BÁO CÁO BÀI TẬP

MÔN HỌC: KHAI THÁC VĂN BẢN VÀ ỨNG DỤNG

Giảng viên hướng dẫn: Nguyễn Trường Sơn

Học viên: Nguyễn Huỳnh Trường Khang

20C12006

Mục lục

1 MÔI TRƯỜNG CHẠY CHƯƠNG TRÌNH

Các chương trình trong báo cáo được chạy với môi trường thiết lập như sau:
- Python 3.6
- Nltk 3.6.7
- Scikit-learn 0.23.2
- Sklearn-crfsuite 0.3.6

2 BÀI 1

2.1 YÊU CẦU

Xây dựng chương trình python cho phép rút trích thông tin email và số điện thoại từ văn bản.

2.2 HƯỚNG DẪN

Khi chạy chương trình chia làm hai trường hợp:
- Trường hợp 1: không truyền tham số vào chương trình

Chương trình yêu cầu nhập đoạn văn bản cần rút trích (có thể nhập nhiều đoạn) và xuất ra màn hình danh sách các email và số điện thoại có trong văn bản.
- Trường hợp 2: truyền tham số vào chương trình, bao gồm input là file văn bản và output là tên file chứa danh sách các email và số điện thoại có trong văn bản

Chương trình đọc văn bản cần rút trích từ input và ghi kết quả ra file output (Nếu file ouput chưa tồn tại thì sẽ tạo mới).

2.3 KẾT QUẢ

Trường hợp 1

Input

```
Enter/Paste your content. Ctrl-Z ( windows ) to save it.
Địa chỉ: Số 28 Trần Hưng Đạo - Quận Hoàn Kiếm - Hà Nội. Điện thoại Cổng thông tin điện tử
: (+84)24 2220 2828. Email: support@mof.gov.vn
Công ty CP truyền thông Luật Việt Nam. Điện thoại: 093 836 1919. Email: cskh@luatvietnam.
vn
^Z
```

Output

```
Phone number list= ['(+84)24 2220 2828', '093 836 1919']
Email list= ['support@mof.gov.vn', 'cskh@luatvietnam.vn']
```

Trường hợp 2

Tập tin input_text.txt

```
<p><strong>Kuala Lumpur</strong><strong>:</strong> +60 (0)3 2723 7900</p>
        <p><strong>Mutiara Damansara:</strong> +60 (0)3 2723 7900</p>
        <p><strong>Penang:</strong> + 60 (0)4 255 9000</p>
        <h2>Where we are </h2>
        <strong> Call us on:</strong> +6 (03) 8924 8686
        </p></div><div class="sys_two">
    <h3 class="parentSchool">General enquiries</h3><p style="FONT-SIZE: 11px">
     <strong> Call us on:</strong> +6 (03) 8924 8000
+ 60 (7) 268-6200 <br />
 Fax:<br />
 +60 (7) 228-6202<br />
 The standard American telephone number is ten digits, such as (555) 555-1234 .
 0974007496 0911905428
Phone:</strong><strong style="color: #f00">+601-4228-8055</strong>
email: </strong><strong style="color: #f00">support123@gmail.com</strong>
```

Input

```
bai1.py input_text.txt result.txt
```

Output

```
The result of phone number and email extraction is saved in  result.txt
```

Tập tin result.txt

```
Phone number list:
(7) 268-6200
(7) 228-6202
(555) 555-1234
723 7900
723 7900
255 9000
924 8686
924 8000
601-4228
Email list:
support123@gmail.com
```

3 BÀI 2

3.1 YÊU CẦU

Xây dựng chương trình python cho phép phân loại câu hỏi tiếng Việt sử dụng thuật toán SVM

3.2 HƯỚNG DẪN

Dữ liệu cung cấp sẵn được xử lý theo dạng: *<class>:<câu hỏi>* và lưu ở file *dataset.txt*
Sử dụng file *bai2_config.py* để thiết lập các cấu hình.
Sử dụng file *bai2_train.py* để huấn luyện mô hình theo thuật toán SVM và lưu thành tập tin tại thư muc *models*.

Sử dụng file *bai2_main.py* để load mô hình lên và phân loại câu hỏi do người dùng nhập vào.

3.3 KẾT QUẢ

File dataset.txt

```
DESC.def:Con gà là con gì
DESC.def:Định nghĩa định lí pytago là gì
DESC.def:Thế nào gọi là bò sát
DESC.def:Thế nào gọi là gia cầm
DESC.def:Thế nào gọi là động vật có vú
DESC.def:Thế nào gọi là hiệu ứng nhà kính
DESC.def:Định nghĩa của môi trường sống
DESC.desc:Quả cam có vị chua như thế nào
DESC.desc:Trời hôm nay nóng như thế nào
DESC.desc:Cơm mẹ nấu ngon như thế nào
DESC.desc:Tỉ phú Phạm Nhật Vượng giàu như thế nào
DESC.desc:Giá đất đắt như thế nào
DESC.desc:Xe Mercedes mắc như thế nào
DESC.reason:Tại sao đèn xanh lại phải đi
DESC.reason:Tại sao mật ong lại ngọt
DESC.reason:Tại sao tránh uống cafe vào buổi tối
DESC.reason:Tại sao không nên ăn nhiều vào ban đêm
```

Chạy file *bai2_train.py* để huấn luyện mô hình. Quá trình huấn luyện (bao gồm xây dựng và tối ưu tham số cho mô hình) sẽ được ghi nhận lại tại *bai2_train.log* (được cấu hình mặc định trong *bai2_config.py*)

Do tập dữ liệu ít nên cần xử lý lại, chỉ lấy theo các phân lớp lớn:

```
 Class: {'SELECT', 'HUM', 'ABBR', 'LOC', 'ENTY', 'NUM', 'DESC'}
Kết quả đạt được với mô hình ban đầu:
 Accuracy: 0.8640350877192983
Tiếp theo, ta tiến hành tối ưu các tham số để đạt được kết quả tốt hơn:
 Best score= 0.8738521752220383
 with parameter:
 clf__C: 2.0
 tfidf__use_idf: False
 vect__ngram_range: (1, 2)
Mô hình được lưu lại tại đường dẫn
The SVM model is saved in C:\models\svm_text_cls
```

Cuối cùng, ta chạy *bai2_main.py* để load mô hình vừa lưu và nhập vào câu hỏi để kiểm tra khả năng phân loại của mô hình:

```
D:\Demo>python bai2_main.py
Enter/Paste your content. Ctrl-Z ( windows ) to save it.
Tại sao trái đất hình tròn?
Hồ Chí Minh là ai?
^Z
```

Kết quả thu được

```
Tại sao trái đất hình tròn? ==> DESC
Hồ Chí Minh là ai? ==> HUM
```

4 BÀI 3

4.1 YÊU CẦU

Xây dựng chương trình python cho nhận diện thực thể định danh (Named Entity Recogtion) sử dụng thuật toán CRF.

4.2 HƯỚNG DẪN

Dữ liệu cung cấp sẵn trong thư mục CoNLL-2003 với các file *train.txt, valid.txt* và *test.txt*
Sử dụng file *bai3_config.py* để thiết lập các cấu hình và cung cấp các hàm rút trích các feature của các từ.
Sử dụng file *bai3_train.py* để huấn luyện mô hình theo thuật toán CRF và lưu thành tập tin tại thư mục *models*.
Sử dụng file *bai3_main.py* để load mô hình lên và chạy chương trình chính.

4.3 KẾT QUẢ

File train.txt

```
-DOCSTART- -X- -X- O

EU NNP B-NP B-ORG
rejects VBZ B-VP O
German JJ B-NP B-MISC
call NN I-NP O
to TO B-VP O
boycott VB I-VP O
British JJ B-NP B-MISC
lamb NN I-NP O
. . O O

Peter NNP B-NP B-PER
Blackburn NNP I-NP I-PER

BRUSSELS NNP B-NP B-LOC
1996-08-22 CD I-NP O
```

File valid.txt

```
-DOCSTART- -X- -X- O

CRICKET NNP B-NP O
- : O O
LEICESTERSHIRE NNP B-NP B-ORG
TAKE NNP I-NP O
OVER IN B-PP O
AT NNP B-NP O
TOP NNP I-NP O
AFTER NNP I-NP O
INNINGS NNP I-NP O
VICTORY NN I-NP O
. . O O

LONDON NNP B-NP B-LOC
1996-08-30 CD I-NP O

```

File test.txt

```
-DOCSTART- -X- -X- O

SOCCER NN B-NP O
- : O O
JAPAN NNP B-NP B-LOC
GET VB B-VP O
LUCKY NNP B-NP O
WIN NNP I-NP O
, , O O
CHINA NNP B-NP B-PER
IN IN B-PP O
SURPRISE DT B-NP O
DEFEAT NN I-NP O
. . O O

Nadim NNP B-NP B-PER
Ladki NNP I-NP I-PER
```

Chạy file *bai3_train.py* để huấn luyện mô hình. Quá trình huấn luyện (bao gồm xây dựng và tối ưu tham số cho mô hình) sẽ được ghi nhận lại tại *bai3_train.log* (được cấu hình mặc định trong *bai3_config.py*)

Ban đầu ta được huấn luyện mô hình với thông số sau:

```
L-BFGS optimization
c1: 0.100000
c2: 0.100000
num_memories: 6
max_iterations: 100
```

Ta thu được F1 score như sau:

```
F1 score: 0.8811508100893499
```

Ta sử dụng RandomizedSearchCV để tối ưu các tham số nhằm thu được mô hình tốt hơn:

```
Best score in train dataset= 0.8495688951646253
Best score in test dataset= 0.8856710914573505
with best_params= {'c2': 0.01, 'c1': 0.0001}
```

Tiếp tục, ta tạo mô hình với tham số vừa tối ưu được

```
Create model with best params:
```

Và lưu tại đường dẫn

```
The CRF model is saved in  D:\models\ner_conll_crf
```

Cuối cùng, ta chạy *bai3_main.py* để load mô hình vừa lưu và nhập vào các câu input để chương trình xử lý

```
Enter/Paste your content. Ctrl-Z ( windows ) to save it.
He is a German who works at Facebook Inc.
The French President's remarks drew swift and angry condemnation as tensions rise
over new vaccine pass
^Z
```

Kết quả đạt được

```
He ==> O
is ==> O
a ==> O
German ==> B-MISC
who ==> O
works ==> O
at ==> O
Facebook ==> B-ORG
Inc ==> I-ORG
. ==> O
The ==> O
French ==> B-MISC
President ==> O
's ==> O
remarks ==> O
drew ==> O
swift ==> O
and ==> O
angry ==> O
condemnation ==> O
as ==> O
tensions ==> O
rise ==> O
over ==> O
new ==> O
vaccine ==> O
pass ==> O
```