Report Task Assignment
Name Entity Recognition

taindp98@gmail.com

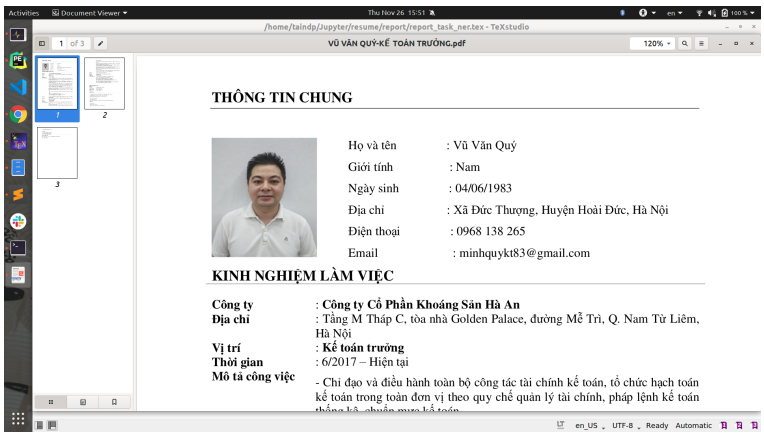
Ngày 28 tháng 11 năm 2020

Mục lục

1 Dataset

1.1 Dữ liệu PDF

Bộ data PDF sử dụng PyMuPDF để extract thông tin từ 30 resume định dạng .pdf



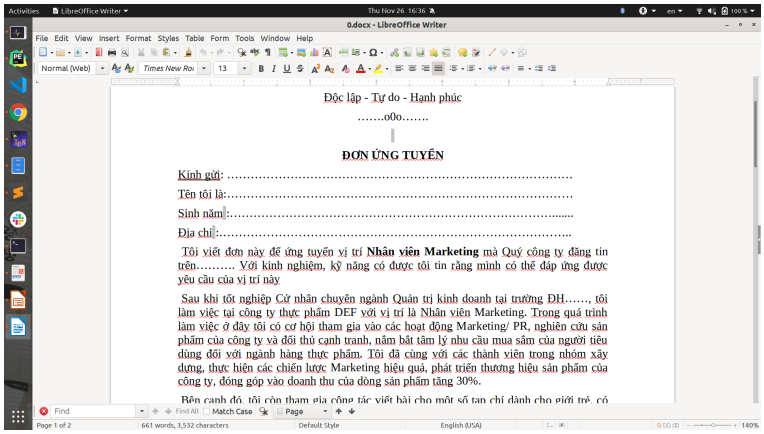
Hình 1: Một mẫu data định dạng pdf

Thông tin từ bộ dữ liệu PDF khá phong phú, đặc biệt là các feature như thông tin cá nhân, số điện thoại, địa chỉ, email,...
Tuy nhiên việc trích xuất thông tin pdf bằng các thư viện có sẵn gặp nhiều nhập nhằng do layout của mỗi sample khác nhau. Trong đó khó khăn nhất là các sample có layout dạng bảng.

Giải pháp tạm thời cho bài toán trích xuất thông tin resume là chỉ có thể sử dụng bộ data định dạng pdf cho việc trích xuất feature cho mô hình huấn luyện Name Entity Recognition.
Đối với model language cho việc phân loại category sẽ sử dụng bộ dữ liệu định dạng word được trình bày ở phần tiếp theo.

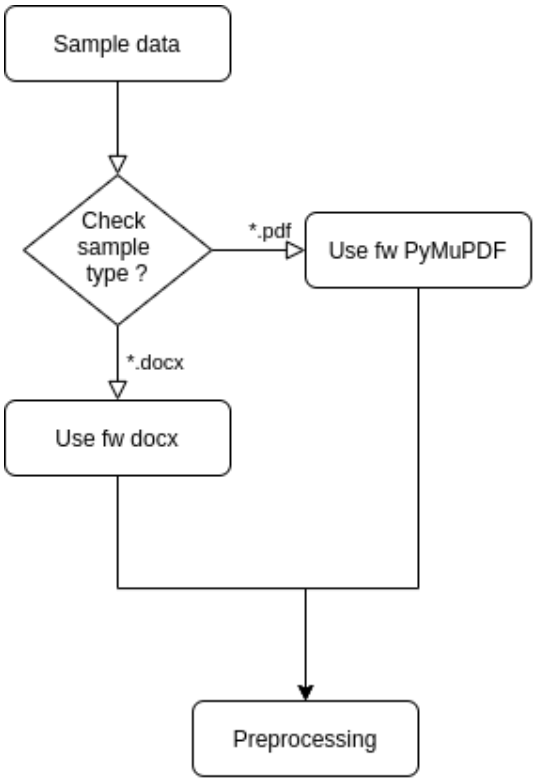
1.2 Dữ liệu Word

Bộ data Word sử dụng docx để extract thông tin khoảng 30 template resume được crawl từ các trang hướng dẫn tạo resume.



Hình 2: Một mẫu data định dạng word

Thông tin từ bộ dữ liệu Word khá tệ về mặt feature entity tuy nhiên lại rất phù hợp cho mô hình ngôn ngữ phân loại văn bản vì dễ dàng trích xuất từng câu ngôn ngữ tự nhiên.



Hình 3: Luồng xử lý một mẫu dữ liệu đầu vào

Đặc biệt với framework docx có thể trích xuất thông tin cả dạng table.

2 Tiền xử lý dữ liệu

2.1 Trích xuất thông tin

Đầu vào của module trích thông tin sẽ kiểm tra sample đang vào có phần extension là *.docx hay *.pdf.

Điểm đặc biệt là module trích xuất thông tin sample định dạng *.docx có thể kiểm tra trong sample có chứa table hay không.

2.2 Tiền xử lý dữ liệu

Các bước tiền xử lý bao gồm:

- convert unicode: chuyển ký tự về bảng mã chuẩn Unicode
- clear punctation: loại bỏ dấu câu
- clear special: loại bỏ ký tự đặc biệt
- clear multi space: loại bỏ nhiều ký tự trống

Out[139]:

	Sentence #	Word	Tag
0	Sentence 0	VŨ	personal_name
1	Sentence 0	THỊ	personal_name
2	Sentence 0	QUÝ	personal_name
3	Sentence 1	nutonuto	date_of_birth
4	Sentence 1	nuto	date_of_birth
5	Sentence 1	nutonutonutonuto	date_of_birth
6	Sentence 2	nutonutonuto	phone
7	Sentence 2	nutonutonuto	phone
8	Sentence 2	nutonutonuto	phone
9	Sentence 3	Số	O

Hình 4: Entity sau khi gán nhãn

3 Tiến hành gán nhãn cho entity

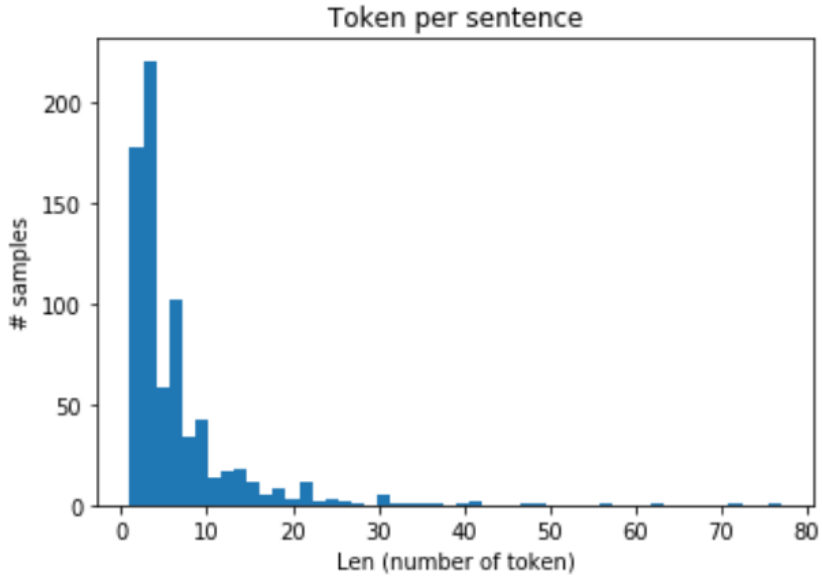
Category	Name Entity	ID	Example
Personal	name	0	Nguyễn Văn A
	dob	1	nuto nuto nutonutonutonuto
	sex	2	Nam
	phone	3	nutonutonutonuto nutonutonuto nutonutonuto
	address	4	Q1 TPHCM
	per_id	5	nutonutonuto nutonutonuto nutonutonuto
	email	6	anguyen gmail com
Education	time_edu	7	nuto nutonutonutonuto
	school/university	8	ĐHBK TPHCM
	major	9	xây dựng cầu đường
	other_edu	10	tin học văn phòng
Experience	job	11	giám sát công trình
	time_job	12	nuto nutonutonutonuto nuto nutonutonutonuto
	company	13	công ty tnhh xây dựng
Other entity	O	14	

Đối với những câu tự nhiên không chứa bất cứ entity nào sẽ loại bỏ để giảm noise cho mô hình huấn luyện.

4 Xây dựng Vocabulary

Chuyển từ word của câu ngôn ngữ tự nhiên thành index trong bộ Vocabulary đối với tập train. Đối với tập test sử dụng nhãn PAD đối với các word unseen trong Vocabulary.

Dựa vào visualize chiều dài các câu chứa entity trong tập train, chọn MAX_LEN cho mô hình huấn luyện là 15 words.
Thực hiện padding cho toàn bộ dữ liệu tập train cùng kích thước là MAX_LEN.
Đối với các entity có dạng chữ số, encode word thành "nuto".



Hình 5: Visualize chiều dài câu chứa entity

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 15)	0
embedding_1 (Embedding)	(None, 15, 200)	281600
bidirectional_1 (Bidirection	(None, 15, 400)	641600
lstm_2 (LSTM)	(None, 15, 400)	1281600
time_distributed_1 (TimeDist	(None, 15, 15)	6015
crf_1 (CRF)	(None, 15, 16)	544

Total params: 2,211,359
Trainable params: 2,211,359
Non-trainable params: 0

Hình 6: Kiến trúc BiLSTM + CRF

5 Kiến trúc mô hình

Xây dựng mạng neural học sâu với mục đích custom NER sử dụng thư viện Keras.
Sử dụng kiến trúc BiLSTM kết hợp layer CRF ở cuối. Lớp LSTM như một bộ lọc các thông tin không mong muốn và giữ lại các thông tin quan trọng như feature entity. Layer CRF được sử dụng để dự đoán nhãn đầu ra dựa trên các nhãn phân bố trong quá khứ.
Sử dụng hàm optimize là Adam với learning rate là 1e-2 và weight decay là 0.9.
Sử dụng hàm loss của layer CRF từ thư viện keras-contrib.
Huấn luyện mô hình với batch size là 21 mẫu trong 10 epochs.
Performace đạt được train_loss = 0.04, valid_loss = 0.47.
Sử dụng classify category thành 3 class trước khi feed vào mô hình NER để tăng độ tin cậy cho kết quả detect entity. Do đó có thể chia các entity đã gán nhãn thành 3 tập train ứng với 3 class của mô hình phân loại ngôn ngữ.
Tiến hành huấn luyện trên 3 tập train khác nhau được 3 models NER.

```
Word           ||True ||Pred
****************************************
Họ             : O     personal_name
và             : O     personal_name
tên            : O     personal_name
Trần           : personal_name personal_name
Nguyệt         : personal_name personal_name
Anh            : personal_name personal_name
F1-score: 71.7%
              precision    recall  f1-score   support

        phone       0.56      0.71      0.63         7
date_of_birth       0.90      1.00      0.95         9
          sex       1.00      0.57      0.73         7
      address       0.55      0.50      0.52        12
        email       0.78      1.00      0.88         7
personal_name       0.67      0.50      0.57         4

  avg / total       0.73      0.72      0.71        46
```

Hình 7: Đánh giá với tập train Personal

```
F1-score: 68.8%
             precision    recall  f1-score   support

  other_edu       0.80      0.33      0.47        12
major_name        0.67      0.67      0.67        12
  edu_time        1.00      0.92      0.96        12
 education        0.53      0.77      0.62        13

avg / total       0.74      0.67      0.68        49
```

```
In [137]: 1 test(model_class2,X_test_class2,y_test_class2,words_class2,idx2tag_class2)
```

```
Example #34
Word           ||True ||Pred
****************************************
Chứng          : other_edu other_edu
chỉ            : other_edu other_edu
tiếng          : other_edu other_edu
anh            : other_edu other_edu
TOEIC          : other_edu other_edu
Điểm           : other_edu other_edu
nutonutonuto   : other_edu other_edu
```

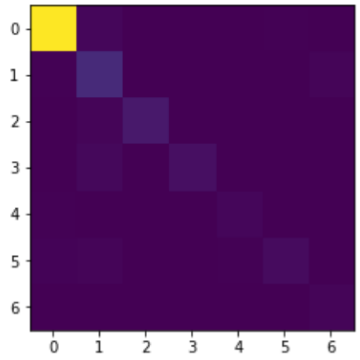
Hình 8: Đánh giá với tập train Education

```
Example #4
Word           ||True ||Pred
****************************************
Nhân           : job   job
viên           : job   job
IT             : job   O
Phần           : job   O
cứng           : job   O
Lập            : job   O
trình          : job   O
F1-score: 60.0%
              precision    recall  f1-score   support

   job_time       0.80      0.89      0.84        18
        job       0.73      0.58      0.65        19
    company       0.40      0.26      0.32        23

avg / total       0.63      0.55      0.58        60
```

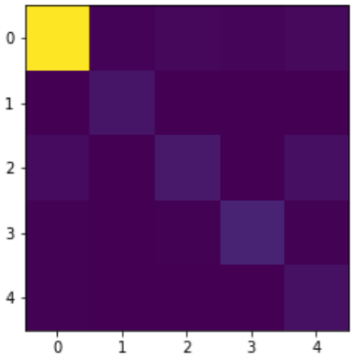
Hình 9: Đánh giá với tập train Experience

6 Đánh giá tập test và Classify report

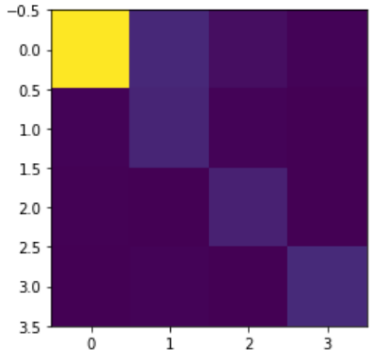
Test thử một vài sample trong tập test, kết quả ổn.



Hình 10: Confusion với tập train Personal



Hình 11: Confusion với tập train Education



Hình 12: Confusion với tập train Experience