TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP GIỮA KỲ**

**MÔN NHẬP MÔN XỬ LÝ NGÔN NGỮ TỰ NHIÊN**

**Conditional Random Fields – CRFs**

*Người hướng dẫn*: **PGS.LÊ ANH CƯỜNG**

*Người thực hiện*: **NGUYỄN THÀNH KHANG – 518H0372**

Lớp **: 18H50303**

Khoá  **: 22**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

Nội dung

[Data Set 3](#_Toc65867948)

[Tạo các thuộc tính 4](#_Toc65867949)

[Xây dựng mô hình CRFs 5](#_Toc65867950)

[Đánh giá mô hình Conditional Random Fields 5](#_Toc65867951)

[Demo 9](#_Toc65867952)

[Tham khảo 9](#_Toc65867953)

**Trường xác suất có điều kiện**

**(Conditional Random Fields – CRFs)**

CRFs là thuật toán xác suất có điều kiện. Sự khác biệt ở mô hình này là sự phân phối xác suất theo điều kiện P(y/x), và xác suất này sẽ cố gắng đưa về xác suất bình thường: P(y, x).

Mô hình hồi quy Logistic, SVM, CRFs là những thuật toán phân loại có điều kiện. Naive Bayes, HMMs là những thuật toán phân loại sinh mẫu. CRFs được sử dung cho việc gắn nhãn NREs và POS.

Trong thuật toán CRFs, đầu vào là tập hợp các thuộc tính (dạng số thực) từ tập dữ liệu đầu vào theo một quy tắc. Trọng số của biểu thức với các thuộc tính đầu vào cùng các nhãn đã được găn thẻ trước đó và task sẽ được dùng để dự đoán cho việc nhãn gắn hiện tại. Ta sẽ ước lượng trọng số sao cho chỉ số **likelihood**(Trong thống kê dùng để đo đo mức độ phù hợp của một mô hình thống kê với một mẫu dữ liệu chứa các giá trị đã cho của các tham số chưa biết.) của nhãn trog bộ dữ liệu train là cực đại.

Hàm mục tiêu trong thuật toán sẽ xác định nhãn cho mỗi từ trong câu. Hãy tưởng tượng với một ví dụ sau: chữ cái đầu tiên của từ có phải là chữ cái viết hoa không?; các tiền tố và hậu tố của từ có dạng như thế nào?; …

Trong CRFs, chúng ta cũng xây dựng dự đoán nhãn từ hiện tại theo nhãn của các từ trước đó. Các trọng số của mô hình sẽ được hợp lý nhất.

## Data Set

Chúng ta sẽ làm ví dụ với bộ dữ liệu NLTK Treebank trên thư viện NLTK. Các nhãn trên NLTK được gắn trên thư viện gồm 12 nhãn: Động từ, Danh từ, Đại từ, Tính từ, Trạng từ, Adpositions, Conjunctions, Determiners, Cardinal Numbers, Particles, Từ khác / Từ mượn, Dấu câu. Bộ dữ liệu gồm có 3.914 câu được gắn thẻ và 12.408 từ vựng.

1. tagged\_sentence = nltk.corpus.treebank.tagged\_sents(tagset='universal')
2. **print**("Number of Tagged Sentences ",len(tagged\_sentence))
3. tagged\_words=[tup **for** sent **in** tagged\_sentence **for** tup **in** sent]
4. **print**("Total Number of Tagged words", len(tagged\_words))
5. vocab=set([word **for** word,tag **in** tagged\_words])
6. **print**("Vocabulary of the Corpus",len(vocab))
7. tags=set([tag **for** word,tag **in** tagged\_words])
8. **print**("Number of Tags in the Corpus ",len(tags))

Tiếp theo, chúng ta sẽ chia bộ dữ liệu hành bộ Train và bộ Test theo tỉ lệ 80:20. Như vậy, sẽ có 3,131 câu thuộc bộ dữ liệu train và 783 câu thuộc bộ dữ liệu test.

1. train\_set, test\_set = train\_test\_split(tagged\_sentence,test\_size=0.2,random\_state=1234)
2. **print**("Number of Sentences in Training Data ",len(train\_set))
3. **print**("Number of Sentences in Testing Data ",len(test\_set))

## Tạo các thuộc tính

Để xác định nhãn POS, chúng ta sẽ xây dựng một bộ từ điển với nhãn được gắn cho mỗi từ trong câu:

* Chữ cái đầu tiên có phải là chữ viết hoa không (Danh từ riêng có chữ cái đầu tiên viết hoa)?
* Đó có phải là từ đầu tiên trong câu không?
* Đó có phải là từ cuối cùng không?
* Từ này có chứa cả chữ và số không?
* Từ này có dấu gạch nối ở giữa không (phần này được áp dụng trong tiếng anh, những tính từ dài thường có dấu gạch nối: ví dụ: fast-growing, slow-moving).
* Các chữ cái đều được viết hoa toàn bộ?
* Đó là một số?
* Bốn hậu tố và tiền tố của từ là gì? (Trong tiếng anh, những từ có đuôi “ed” thường là động từ; những từ kết thúc bằng “ous” thường là tính từ.

Thuộc tính sẽ được xác định theo công thức như hình dưới và những features cho bộ dữ liệu train và test sẽ được trích xuất ra.

1. **def** features(sentence,index):
2. **return** {
3. 'is\_first\_capital':int(sentence[index][0].isupper()),
4. 'is\_first\_word': int(index==0),
5. 'is\_last\_word':int(index==len(sentence)-1),
6. 'is\_complete\_capital': int(sentence[index].upper()==sentence[index]),
7. 'prev\_word':'' **if** index==0 **else** sentence[index-1],
8. 'next\_word':'' **if** index==len(sentence)-1 **else** sentence[index+1],
9. 'is\_numeric':int(sentence[index].isdigit()),
10. 'is\_alphanumeric': int(bool((re.match('^(?=.\*[0-9]$)(?=.\*[a-zA-Z])',sentence[index])))),
11. 'prefix\_1':sentence[index][0],
12. 'prefix\_2': sentence[index][:2],
13. 'prefix\_3':sentence[index][:3],
14. 'prefix\_4':sentence[index][:4],
15. 'suffix\_1':sentence[index][-1],
16. 'suffix\_2':sentence[index][-2:],
17. 'suffix\_3':sentence[index][-3:],
18. 'suffix\_4':sentence[index][-4:],
19. 'word\_has\_hyphen': 1 **if** '-' **in** sentence[index] **else** 0   }

## Xây dựng mô hình CRFs

Bước tiếp theo, chúng ta sử dụng thư viện sklearn\_crfsuite để xây dựng mô hình. Mô hình được tối ưu hóa theo phương pháp Gradient(Là véc tơ biểu diễn tốc độ thay đổi của hàm) Descent và sử dụng thuật toán LBGS – L1, L2. Mô hình CRFs sẽ giúp gắn nhãn tất cả những các từ, kể cả những từ không có trong bộ tài liệu.

1. **from** sklearn\_crfsuite **import** CRF
2. crf = CRF(
3. algorithm='lbfgs',
4. c1=0.01,
5. c2=0.1,
6. max\_iterations=100,
7. all\_possible\_transitions=True
8. )
9. crf.fit(X\_train, y\_train)

## Đánh giá mô hình Conditional Random Fields

Chúng ta sử dụng chỉ số F-score để đánh giá mô hình. F-score sẽ cân bằng hai chỉ số Precision và Recall như sau:

**2\*((precision\*recall)/(precision+recall))**

**Precision** được định nghĩa là tỷ lệ những quan sát được dự đoán đúng trên tổng số quan sát được dự đoán đúng

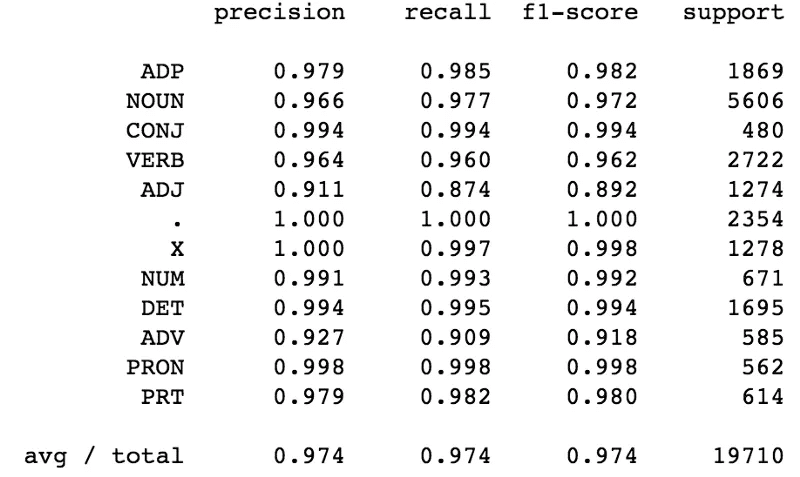
**Recall** được định nghĩa là số dự đoán chính xác trên tổng số số quan sát thực sự thuộc lớp đúng.

**Precision=TP/(TP+FP)**  
**Recall=TP/(TP+FN)**

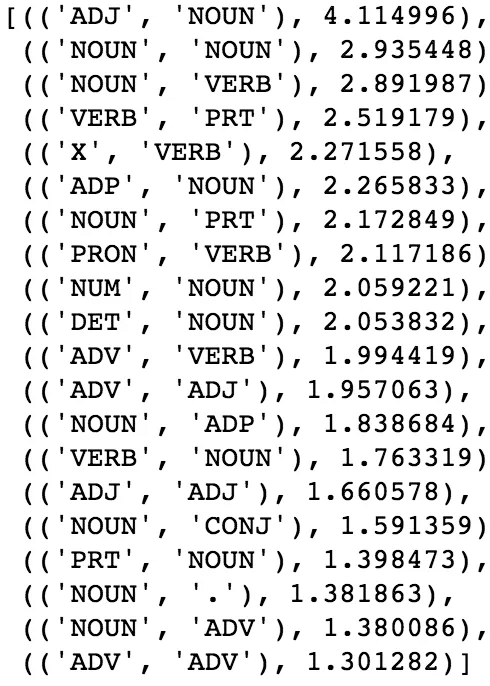
Mô hình Conditional Random Fields (CRFs) trên đã đưa ra F-score bằng 0.996 với bộ dữ liệu train và 0.97 với bộ dữ liệu test.

1. **from** sklearn\_crfsuite **import** metrics
2. **from** sklearn\_crfsuite **import** scorers
3. y\_pred=crf.predict(X\_test)
4. **print**("F1 score on Test Data ")
5. **print**(metrics.flat\_f1\_score(y\_test, y\_pred,average='weighted',labels=crf.classes\_))
6. **print**("F score on Training Data ")
7. y\_pred\_train=crf.predict(X\_train)
8. metrics.flat\_f1\_score(y\_train, y\_pred\_train,average='weighted',labels=crf.classes\_)
10. ### Look at class wise score
11. **print**(metrics.flat\_classification\_report(
12. y\_test, y\_pred, labels=crf.classes\_, digits=3
13. ))

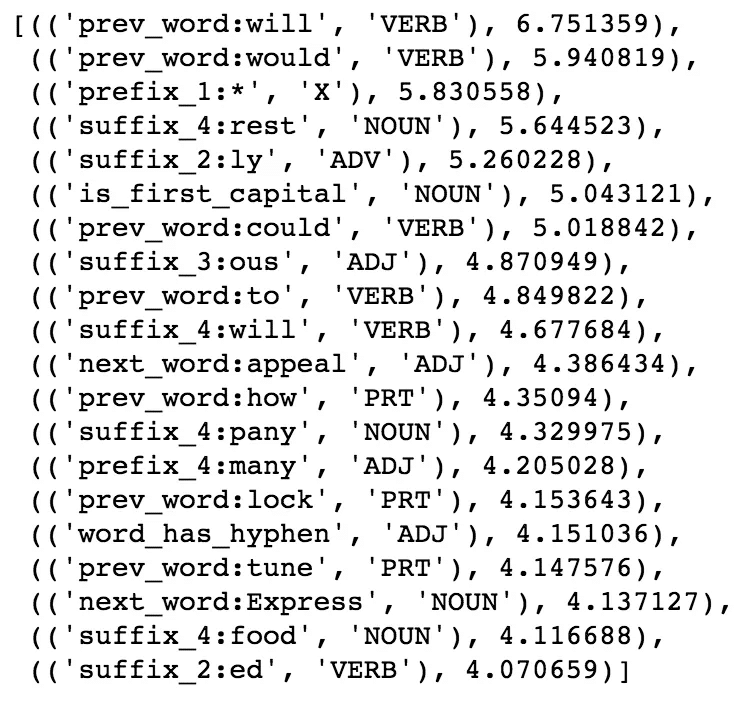
Từ bảng kết quả,, ta thấy rằng để dự đoán một từ có phải là tính từ không, các chỉ số precision, recall và F-score thấp. Như vậy ta cần thêm các features vào mô hình CRFs để dự đoán Tình từ tốt hơn.



Tiếp theo, chúng ta hãy xem cụm những nhãn hay đi cùng nhau nhất:



Như chúng ta thấy, tính từ thường được đi với danh từ. Còn động từ thì thường đi sau những Particle (ví dụ: “to”); những Determinant như “The” thường sẽ theo sau là một danh từ.



Các từ “Will”, “would” thường theo sau là động từ; những từ kết thúc bằng “ed” thường là động từ. Còn nếu trong từ có gạch nối, thì xác suất từ đó là một tính từ sẽ cao hơn. Tương tự như vậy, nếu từ đó bắt đầu bằng một chữ cái viết hoa, thì khả năng từ đó là danh từ cao hơn.

Thuật toán CRFs còn được sử dụng trong rất nhiều bài toán không chỉ POS, ví dụ như Tokenizer, NERs,…

## Demo

## https://colab.research.google.com/drive/1mcv\_Eu\_ppSvvnzBrq7RjFZKg2P53FO8y?usp=sharing

## Tham khảo

## [[NLP] Xử lý POS với thuật toán Conditional Random Fields - Trí tuệ nhân tạo (trituenhantao.io)](https://trituenhantao.io/kien-thuc/nlp-xu-ly-pos-voi-thuat-toan-conditional-random-fields/)

## [Conditional random field - Wikipedia](https://en.wikipedia.org/wiki/Conditional_random_field)

## [Conditional Random Fields - Stanford University (By Daphne Koller) - YouTube](https://www.youtube.com/watch?v=rc3YDj5GiVM)