Nhập môn xử lý ngôn ngữ tự nhiên

Báo cáo đồ án

Danh sách thành viên:

1. 19120660_Trương Công Thành
2. 19120667_Nguyễn Văn Thịnh
3. 19120659_ Phạm Văn Thành

Bộ môn Xử Lý Ngôn Ngữ Tự Nhiên

Khoa Công nghệ thông tin

Đại học Khoa học tự nhiên TP HCM

Báo Cáo Đồ Án Môn Học Web hỗ trợ phân tích và vẽ cây cú pháp cho tiếng Anh và Tiếng Việt

Các nội dung chính

Mục tiêu tài liệu tập trung vào các chủ đề:

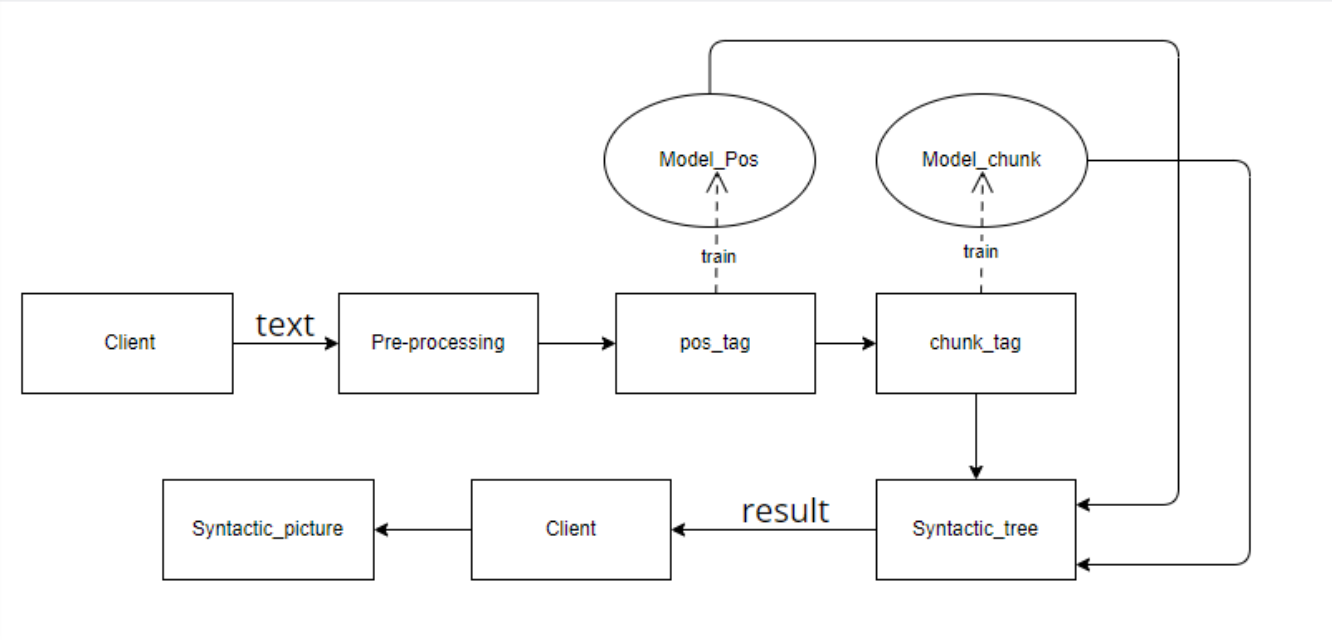
- ✔ Ý tưởng về web
  - • Yêu cầu
  - • Các bước thực hiện
- ✔ Website
  - ▪ Framework
  - ▪ Giao diện
- ✔ Server.
  - ● Word_tokenize
  - ● Model (POS,Chunk)
  - ● Main
- ✔ Hướng dẫn sử dụng
- ✔ Tài liệu tham khảo

1 Ý tưởng xây dựng web

1.1 Phát biểu bài toán.

Xây dựng web hỗ trợ phân tích và vẽ cây cú pháp cho 2 ngôn ngữ Tiếng Anh và Tiếng Việt

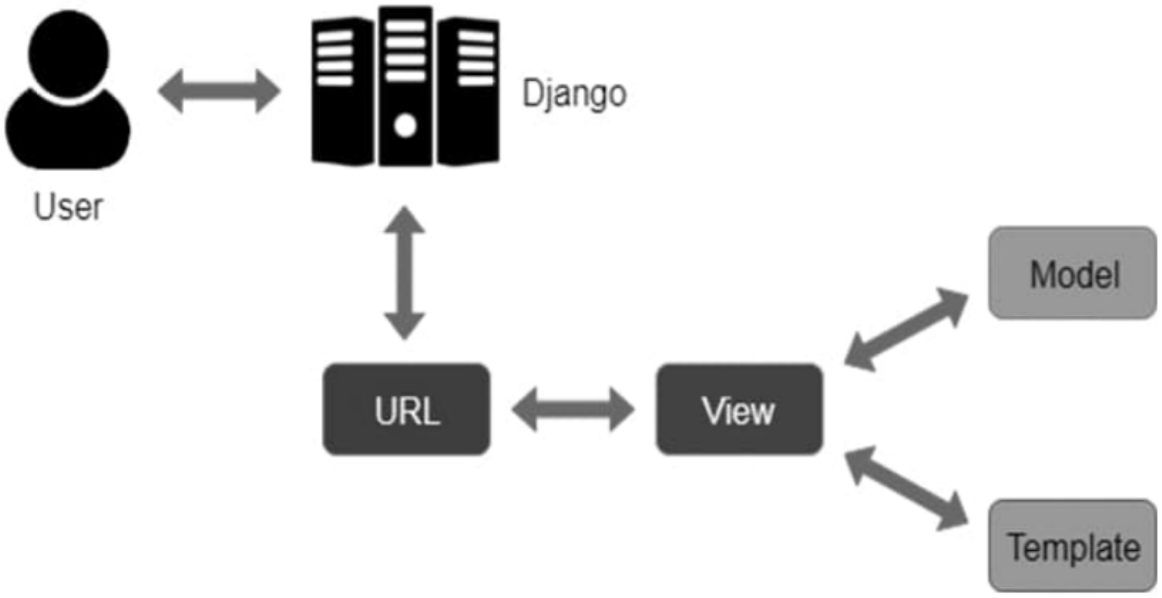
1.2 Các bước thực hiện.



2 Website

2.1 Web framework(Django)

Django là một Framework lập trình web bậc cao, mã nguồn được viết bằng ngôn ngữ lập trình Python dựa tên mô hình MTV (Model-Template-Views). Django giúp cho việc xây dựng web trở nên nhanh chóng hơn cùng với nhiều ưu điểm như tính bảo mật tốt, mở rộng được trong tương lai và đặc biệt rất dễ dàng sử dụng.



Mô hình MVT của Django

2.2 Giao diện (HTML5, CSS, Javasript) :

Giao diện người dùng được thiết kế dựa trên bộ ba ngôn ngữ html5, css, javascript. Trong đó:

+ HTML5: xây dựng nội dung và cấu trúc cơ bản cho trang web.

+ CSS: được sử dụng để kiểm soát trình bày, định dạng và bố cục.

+ Javascript: được sử dụng để kiểm soát hành vi cùng các yếu tố khác nhau chủ yếu là các thao tác của người dùng.

Ngoài ra trang web còn sử dụng bootstrap5 trong việc nhằm tối ưu và đơn giản hóa website

3 Server

3.1 Word tokenizes

- *Tiếng Anh*:

+ Nhóm sử dụng function word_tokenzie() trong thư viện nltk để tách từ trong tiếng Anh
+ Example:

text=" i need to get my story straight "
Result: ['i', 'need', 'to', 'get', 'my', 'story', 'straight']

- *Tiếng Việt*:

+ Nhóm sử dụng hàm word_tokenzie() trong thư viện Underthesea để tách từ trong tiếng Việt
+Example:

Text= ""Anh ta chơi đá bóng khá hay và là một học sinh giỏi""
Result: ['Anh', 'ta', 'chơi', 'đá', 'bóng', 'khá', 'hay', 'và', 'là', 'một', 'học sinh', 'giỏi']

3.2 POS-Part of speech

3.2.1 Gán nhãn từ loại

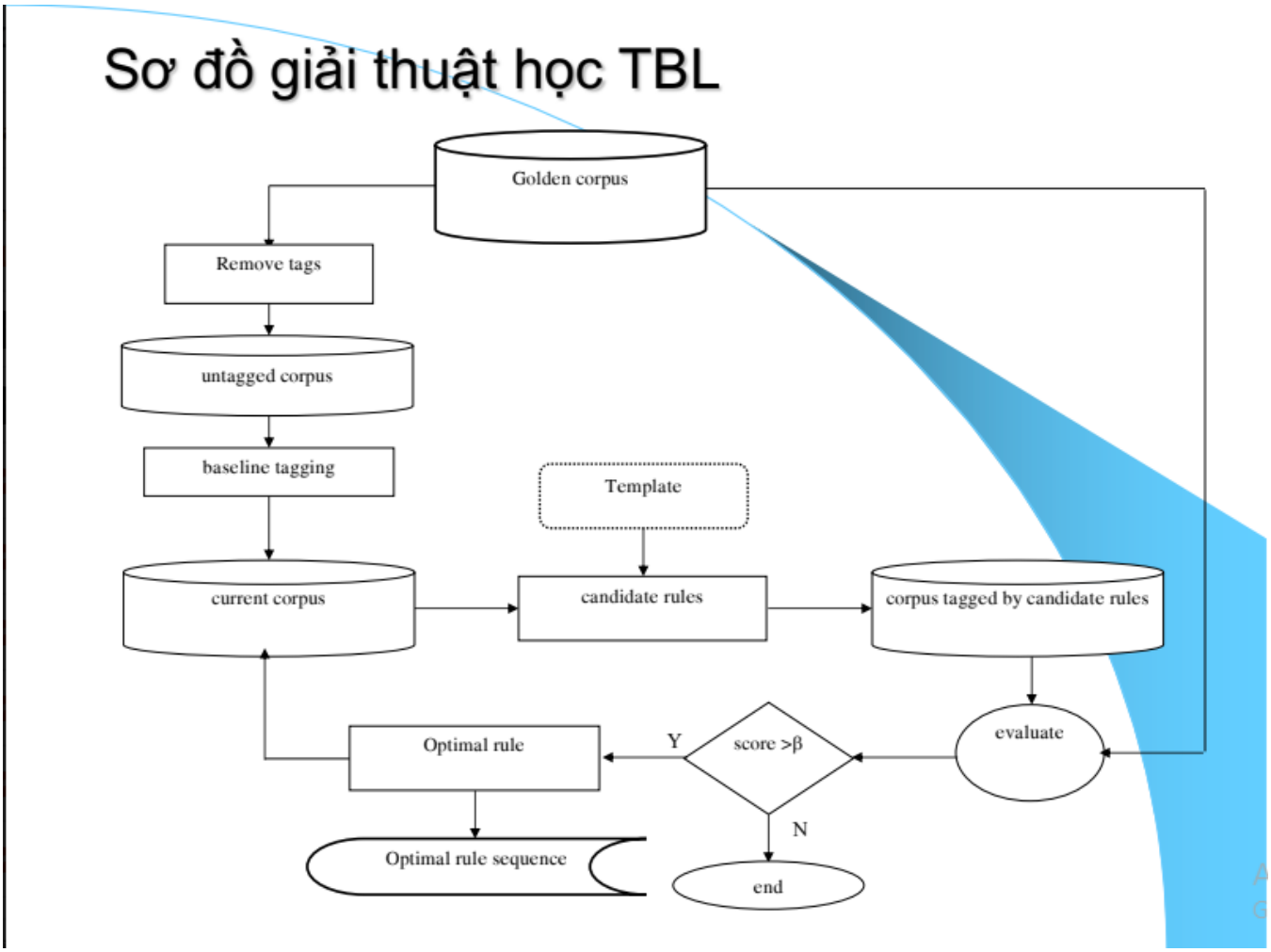
Gán nhãn từ loại là một giai đoạn khá quan trọng trong dịch máy. Nó có ảnh hưởng to lớn đến kết quả của các giai đoạn sau nó cũng như kết quả dịch máy. Việc gán nhãn từ loại chính xác không những ảnh hưởng đến kết quả của dịch máy, nó còn ảnh hưởng rất lớn đến kết quả của các bài toán khác trong xử lý ngôn ngữ tự nhiên, khai khoán dữ liệu như bài toán tìm từ đồng nghĩa, gần nghĩa, bài toán trích chọn thông tin, bài toán phân loại, làm chỉ mục…

3.3.2 Mô hình sử dụng

- Nhóm sử dụng Giải thuật học chuyển đổi dựa trên luật cải biến (TBL): (Transformation-Based Learning)

- **Đặc điểm của TBL**: Đây là phương pháp học giám sát (supervised learning), học dựa trên ký hiệu (symbol). TBL còn có tên là học hướng lỗi (error-driven learning) bằng cách tìm cách giảm được nhiều lỗi nhất so với ngữ liệu huấn luyện (ngữ liệu chuẩn: golden corpus).

- ***Sơ đồ giải thuật:***



- *Mô tả hoạt động của giải thuật*:

Quá trình học của giải thuật được bắt đầu với một ngữ liệu thô (ngữ liệu chưa được gán nhãn).Sau đó, ngữ liệu này được tiến hành gán nhãn cơ sở,hay còn gọi là gán nhãn ban đầu(initialstate). Việc gán nhãn cơ sở chỉ là gán cho ngữ liệu một giá trị ban đầu. Việc gán nhãn có sở có thể không chính xác, chẳng hạn gán nhãn từ loại cho các từ trong câu là danh từ, hoặc cũng có rất chính xác, chúng ta có thể chọn kết quả giải thuật nào đó làm nhãn cơ sở. Nhãn này có thể chính xác hoặc không chính xác. Sau khi dữ liệu đã nhận trạng thái khởi tạo, dữ liệu này được so sánh với các trạng thái đúng của chúng (ngữ liệu vàng). Qua việc so sánh này, các lỗi của dữ liệu hiện hành được xác định. Thông qua các lỗi này chúng xác định được các luật chuyển đổi nhằm biến đổi ngữ liệu từ trạng thái ngây thơ (trong quá quá trình khởi tạo) hay trạng thái hiện hành (đã có áp dụng qua luật chuyển đổi) thành dạng giống hơn so với các trạng thái đúng. Một tập hợp các khung luật lúc này được sử dụng để tạo ra các luật ứng viên. Các khung luật được xác định trước như quy tắc xác định trạng thái "ngây thơ" ở giai đoạn khởi tạo. Mỗi khung luật chứa các biến điều kiện chưa xác định giá trị. Ví dụ mẫu luật sau:

"Nếu nhãn đứng trước X là Z thì đổi nhãn X thành Y". X, Y, và Z là các biến. Với mỗi bộ giá trị của X, Y, Z ta được một luật phát sinh từ mẫu luật này. Trong khung luật trên X và Y là các biến, nó có thể nhận bất kì một giá trị nào trong bộ nhãn mà chúng ta đề ra.

Thuật toán sinh ra các luật ứng viên bằng cách thay các giá trị có thể vào cho các biến trong khung luật. Luật ứng viên sau khi được tạo ra nó sẽ được áp dụng vào trong ngữ liệu đang được gán nhãn hiện hành để tạo ra ngữ liệu được gán nhãn khi áp dụng luật ứng viên này. Ngữ liệu được gán nhãn theo luật ứng viên vừa tạo ra sẽ được so sánh đối chiếu với ngữ liệu đúng (hay ngữ liệu vàng). Khi so sánh với ngữ liệu chính xác chúng ta sẽ biết được luật ứng viên vừa tạo ra chỉnh ngữ liệu từ đúng thành sai bao nhiêu trường hợp và từ sai thành đúng bao nhiêu trường hợp. Từ đó ta tính ra được điểm cho luật ứng viên này. Điểm của luật ứng viên này chính là hiệu số giữa số trường hợp luật chỉnh ngữ liệu từ sai thành đúng và số trường hợp luật chỉnh ngữ liệu từ đúng thành sai. Sau khi tất cả các luật ứng viên được tạo ra chúng ta sẽ biết được luật ứng viên nào có điểm cao nhất, luật ứng viên có điểm cao nhất sẽ được giữ lại cho các lần gán nhãn sau nếu như luật này thoả mãn điều kiện nó có điểm lớn hơn mức ngưỡng mà chúng ta cho trước. Luật này sẽ được áp dụng để chuyển ngữ liệu ở trạng thái thứ k sang trạng thái mới trạng thái thứ k+1. Ngữ liệu ở trạng thái mới này lại lần lượt thử trên các luật ứng viên để chọn ra luật tối ưu mới. Quá trình này sẽ được lặp đi lặp lại cho đến khi không còn có luật tối ưu nào có điểm lớn hơn mức ngưỡng.

Kết thúc giai đoạn huấn luyện chúng ta sẽ thu được một danh sách các luật tối ưu. Các luật tối ưu này sẽ được sử dụng vào quá trình thực thi của giải thuật theo thư tự các luật có điểm cao được áp dụng trước các luật thấp được áp dụng sau

3.2.3 Tagset

- Tập nhãn sử dung trong tiếng Anh : Pen Tree Bank

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two, three*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb, base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb, past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb, gerund	*eating*
JJ	adjective	*yellow*	VBN	verb, past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb, non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb, 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, singular	*IBM*	$	dollar sign	$
NNPS	proper noun, plural	*Carolinas*	#	pound sign	#
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

Penn treebank tags

- Tập Nhãn sử dụng cho từ loại Tiếng Việt:

Phụ lục B: Bộ nhãn từ loại tiếng Việt.

STT	Nhãn từ loại	Ý nghĩa
1	CC	Liên từ
2	CD	Số từ
3	DT	Định từ
4	FW	Từ nước ngoài
5	IN	Giới từ
6	A	Tính từ)
7	LS	Dấu liệt kê
8	MD	Từ tình thái
9	N	Danh từ
10	POS	Sở hữu cách
11	P	Đại từ nhân xưng
12	P$	Đại từ sở hữu
13	R	Trạng từ
14	RP	Tiểu từ
15	SYM	Ký hiệu
16	UH	Thán từ
17	V	Động từ

3.2.4 Corpus

- *Tiếng Anh* **:** Nhóm sử dụng ngữ liệu Treebank với 3914 câu đã được đánh nhãn theo tập nhãn của Penn Tree Bank

=> The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

- *Tiếng Việt* : Nhóm sử dụng ngữ liệu từ file corpus_pos.txt

=> Hươu/N là/V loài/N vật/N được/V con_người/N thuần_dưỡng/V đã/R hàng/R trăm/M năm/N ./CH

- Khó khăn trong gán nhãn từ loại tiếng Việt :

- Đặc trưng riêng về ngôn ngữ
- Thiếu các kho dữ liệu chuẩn như Brown hay Penn Treebank
- Khó khăn trong đánh giá kết quả

3.3.3 Train model

-Đầu vào: Ngữ liệu
-Đầu ra: model
Cài đặt:
- B1: Khởi tạo ngữ liệu cho tập training và testing
+ Đối với Tiếng Anh :
- Tải ngữ liệu từ thư viện nltk

+Đối với Tiếng Việt:
- Load ngữ liệu từ file corpus_pos.txt

+Sử dụng 90% của ngữ liệu cho train_data
+Sử dụng 10% còn lại cho Test_data

-B2 :Training với Brill tagger
+ Gán nhãn cơ sở
- Khởi tạo defaultTagger
- Tạo danh sách backoff tagger
  . unigram Tagger : 1 token / word ,backoff = defaultTagger
  . Bigram Tagger :Sử dụng tag trước đó để đoán tag cho word hiện tại,backoff = uni_Tagger
  . Trigram Tagger:Sử dụng 2 tags trước đó để đoán tag hiện tại, ,backoff = bi_Tagger
  //the BigramTagger subclass looks at two items (the previous tagged word and the current word), and the TrigramTagger subclass looks at three items.
  . The backoff_tagger function creates an instance of each tagger class in the list, giving it train_sents and the previous tagger as a backoff.
  //Nếu tagger hiện tại không biết tag một từ như thế nào thì nó sẽ gọi backoff tagger để tag từ đó

+Sử dụng giải thuật TBL (brillTagger) để train model

- Cung cấp template rules
- Tạo các luật ứng viên từ template
- Giới hạn các luật ứng viên phát sinh về 100 và số điểm tối thiểu để luật ứng viên được chọn = 3 (ta có thể tăng các chỉ số này để cải thiện độ chính xác nhưng sẽ tốn nhiều thời gian hơn để train)

-B3: Lưu model với pickle

⇨ Độ chính xác sau model sau khi train = `0.916539440203562`

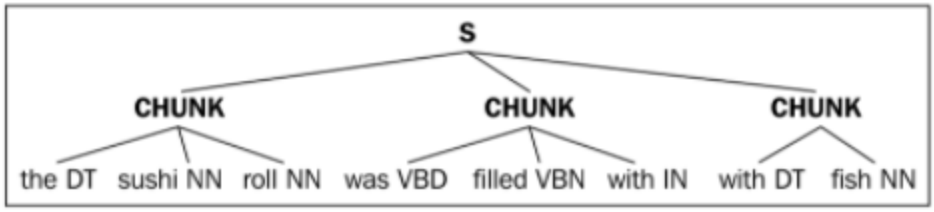
3.3 Chunking

3.3.1 Khái niệm

Chunking là một quá trình trích xuất các cụm từ từ văn bản không có cấu trúc. Thay vì chỉ các mã thông báo đơn giản có thể không đại diện cho ý nghĩa thực sự của văn bản, bạn nên sử dụng các cụm từ như " Nam Phi " như một từ duy nhất thay vì các từ riêng biệt " Nam " và " Châu Phi ".

Chunking hoạt động dựa trên việc gắn thẻ POS, nó sử dụng tag Pos làm đầu vào và cung cấp các khối làm đầu ra. Tương tự như thẻ POS, có một tập hợp tiêu chuẩn của các thẻ Chunk như Cụm danh từ (NP), Cụm động từ (VP), v.v. Việc tách cụm là rất quan trọng khi bạn muốn trích xuất thông tin từ văn bản như Vị trí, Tên người, v.v. Trong NLP được gọi là Khai thác thực thể được đặt tên (NER)

Chunk biểu diễn dưới dạng cây:



Một dạng biểu diễn khác của chunk là IOB tags. IOB tags khá giống với part-of-speech tags, nhưng nó biệu thị các vị trí inside, outside, và beginning của một từ trong một chunk. Không chỉ có mỗi cụm danh từ (NP),chúng ta có thể tạo nhiều cụm chunk khác nhau,(VP,PP,..).

3.3.2 Ngữ liệu

- Tiếng Anh: Sử dụng ngữ liệu coll2000

+10948 câu ngữ liệu chuẩn

+ Được viết viết dưới dạng tree và đánh nhãn theo Penn tree và IOB

+Dưới đây là câu ví dụ trong conll2000 corpus sau khi thay đổi format (mỗi từ được biểu diễn với một part-of-speech tag và theo sau bởi IOB tag):

Mr. NNP B-NP

Meador NNP I-NP

had VBD B-VP

been VBN I-VP

executive JJ B-NP

vice NN I-NP

president NN I-NP

of IN B-PP

Balcor NNP B-NP

. . O

-Trong đó: B-NP denotes the beginning of a noun phrase, while I-NP denotes that the word is inside of the current noun phrase. B-VP and I-VP denote the beginning and inside of a verb phrase. O ends the sentence.

- Tiếng Việt: Lấy ngữ liệu từ file corpus_chunk.txt

+ Một câu trong ngữ liệu:

```
Thấp_thoáng V B-VP
phía N B-NP
sau A I-NP
những L B-NP
ngôi Nc I-NP
nhà N I-NP
ngói N I-NP
đỏ A I-NP
mới A I-NP
. CH O
```

3.3.3 Training model

- B1: Khởi tạo ngữ liệu cho tập training và testing

+ Đối với Tiếng Anh :

- Tải ngữ liệu "coll2000"từ thư viện nltk

+Đối với Tiếng Việt:

- Load ngữ liệu từ file corpus_chunk.txt

+Sử dụng 90% của ngữ liệu cho train_data

+Sử dụng 10% còn lại cho Test_data

-B2 :Trích xuất cặp (`POS-TAG, IOB-CHUNK-TAG`) cho data
+ Chuyển câu từ ngữ liệu dạng tree về format dạng (word, pos_tag, IOB_tag) đối với Tiếng Anh (ngữ liệu tiếng Việt đã ở sẵn dạng format)
+ Trích xuất , giữ lại cặp (POS, IOB)
-B3: Training Ngram tagger tương tự như đã làm bên POS_tag thông qua unigram,bigram,Trigram -B4: Saving model thông qua pickle

⇨ Độ chính xác : `0.8925600739371534`

3.4 Gọi và sử dụng các model đã train

- Tạo hàm loadmodel (sử dụng cho cả pos và chunk model)
- Tạo Class Parser
  + Hàm khởi tạo là 2 model đã được load
  + Class function parser() với input là đoạn text cần phân tích cú pháp, output là cây cú pháp dưới dạng tree
    - Gọi hàm word_tokennize để phân tách từ
    - Loại bỏ punctuation
    - posTag cho các từ sau khi đã tiền xử lý
    - từ danh sách pos_Tag tìm chunk_Tag
    - lấy danh sách (word, pos_tag, chunk_tag) từ các cặp (word, pos_tag) và (chunk_tag)
    - Chuyển format (word, pos_tag, chunk_tag) về dạng tree

4 Hướng dẫn sử dụng

- Đầu tiên, ta tiến hành tải dữ liệu về từ github: http://dlib.net/files/data/ibug_300W_large_face_landmark_dataset.tar.gz
- Link github có chứa source code: https://github.com/congthanhtr/django-nlp
- Xem video demo chương trình được đi kèm với báo cáo này

5 Tài liệu tham khảo

1. https://www.amazon.com/Python-Text-Processing-NLTK-Cookbook/dp/1782167854
2. Vietnamese corpus training