

Dependency Parsing

對一個句子進行關係連結

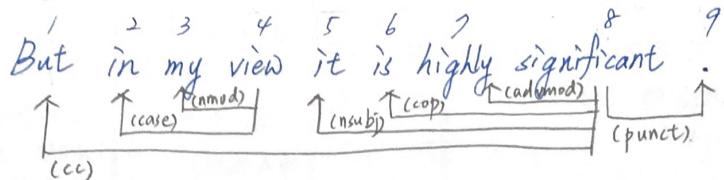
- 資料型態： idd: word index head: head index
- tok: word label: Word 和 head 之關係
- pos, pos2: 詞性

Ex. the cat the 是 cat 的 dependent. 開你的是 DET.

(dep) DET (head)

↑ (root)

△ 有效句子：



* Is projective:

所有 arrow 復此之間不能交叉

做法：每個 head 與其 dep 之間的 descendant，其 head 必須也要在 head ~ dep 之間

步驟：

1. get-oracles：先建立 training data：產生出每個句子每次執行步驟前的 configurations
每次要執行的步驟：gold-transition

◦ 根據每行如果 projective，執行 get-oracle(toks)

{ toks: 一個句子、每個字包含 (idd, tok, pos, head, label) : [(idd1, tok1, pos1, head1, label1), ...]

stack: 存放處理的字的 list，要處理的在尾巴，一開始先設 root → [0].

wbuffer: 存放待處理的字的 list，要處理的在尾巴 → [9 8 7 6 5 4 3 2 1].

arcs: 存放箭頭和關係、deps: 存放 head, dep 關係以便查找 { head1: { (h1, c1): dep1
(h1, c2): dep2 }
head2: { (h2, c1): dep3
(h2, c2): dep4 }

* tree-to-actions (wbuffer, stack, arcs, deps)

a. 判斷 stack 最後兩個字的關係 [root word1 word2]

b. 決定動作： I. shift： 只將 wbuffer 一個字拿過來

II. LEFT: [root word1 ← word2]

先判斷此關係存不存在

如果存在，動作成立 arc: (dep-label, head, child) (word2) (word1)

III. RIGHT: [root word1 → word2]

先判斷此關係存不存在

動作成立 ←

* 還要判斷 word2 有沒有 dependency 關係

如果有就執行

如果有自己的關係是不是都在 arcs 裡了。

arc: (dep-label, head, child)
(word1) (word2)

* tree-to-actions: 執行.

1. 執行 SHIFT:

a. 先把目前 config 存入 configurations

b. 再把動作 SHIFT 存入 gold-transitions

c. 把 first word 移到 stack

LOOP

判断 stack 最後兩個字的關係，確認後

a. 先把目前 config 存入 configurations

b. 再把動作 = SHIFT
| LEFT + label
| RIGHT + label 存入 gold-transitions

c. 如果有 LEFT 或 RIGHT 把 child 存入

把關係存入 arcs

ex.

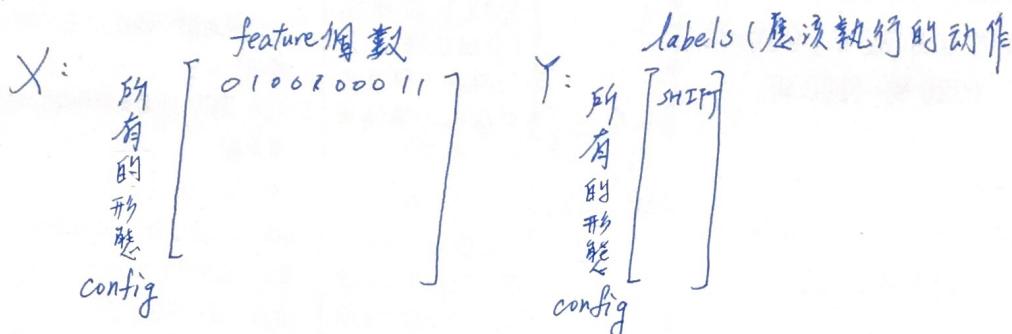
stack	wbuffer	gold-transitions	arcs
[0]	[9 8 7 6 5 4 3 2 1]	shift	
[01]	[9 8 7 6 5 4 3 2]	shift	
[012]	[9 8 7 6 5 4 3]	shift	
[0123]	[9 8 7 6 5 4]	shift	
[01234] \leftarrow	[9 8 7 6 5]	LEFTARC_label	(label, 4, 3)
[0124] \leftarrow	[9 8 7 6 5]	LEFTARC_label	(label, 4, 2)
[01]	[9 8 7 6 5]	shift	
[01.5]	[9 8 7 6]	shift	
[01.56]	[9 8 7]	shift	
[01567]	[9 8]	shift	
[015678] \leftarrow	[9]	LEFTARC_label	(label, 8, 7)
[01568] \leftarrow	[9]	"	(label, 8, 6)
[0158] \leftarrow	[9]	"	(label, 8, 5)
[018]	[9]	"	(label, 8, 1)
[08]	[9]	shift	
[089] \rightarrow	[]	RIGHTARC_label	(label, 8, 9)
[08] \rightarrow	[]	RIGHTARC_label	(label, 0, 8)
[0]	[]		

②

得到 gold-transitions 和 configurations 後。

將 configurations 改成 $\text{feat} : [\text{feat}_1 \text{ feat}_2 \dots]$ 存入 feats
 $\text{gold-transitions} = \text{labels}$

2) train



3) evaluate

將現在的 toks 丟入 classifier，讓其判斷在每個 config 下應該要什麼動作和 dep
 並且建一株 $\{\text{child: head, label}\}$ 的樹
 等於再模仿 training data 的樣子，去比較正確率。

* action-to-tree

會得到這個 config 下對於每個 label / 動作的機率 (predictions)

① 先進行排序，小到大

② 先取大的檢查是否不合理 (isValid)

如果合理 建立 tree

並進行動作

不合理 跳下一個 action