

文字探勘實作 - NER

David Chiu

命名實體識別

命名實體識別(Named Entity Recognition)

- 從一段自然語言文本中找出相關實體，並標註出其位置以及類型

- **Named entity recognition (NER)** is the problem of finding references to entities (mentions) in natural language text, and labeling them with their location and type.

General domain(news)

- Person name
- Organization name
- Location name
- Numeric expression

Biomedical domain

- Gene
- Protein
- Disease
- Chemical
- Cell

Protein

DNA

RNA

Chemical

Disease

Glucocorticoid receptors in peripheral blood lymphocytes of patients with bronchial asthma. Quantitation of glucocorticoid receptors (GCR) and the study of their affinity for glucocorticosteroids (GCS) were made in peripheral blood lymphocytes of bronchial asthma (BA) patients in consideration of GCR treatment and serum levels of endogenous cortisol.

為什麼要做 NER?

■ 字典比對方法

- ▣ 建立一字典，並比對斷詞過的文章中是否有該關鍵字詞

■ 範例

```
org = ['台積電']
```

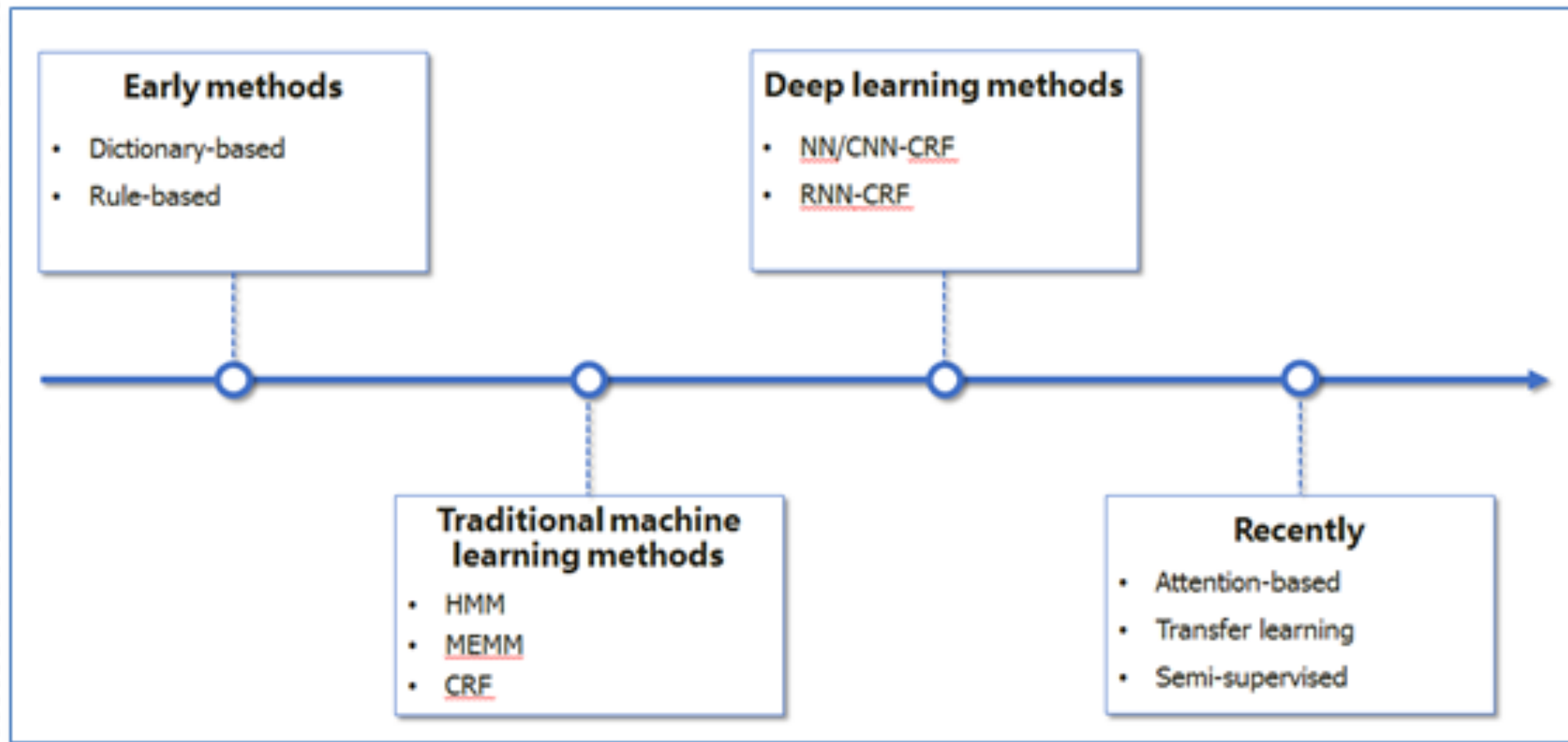
```
import jieba
```

```
set(org) & set(list(jieba.cut('今、明年資本支出近兆台積電大擴產商機來了')))
```

但如果有同義詞出現？

- 台積電也可以被稱作 TSMC, 台積
 - e.g. 台積鉅額交易爆萬張大量
- 解決方案
 - 建立同義辭典(耗工費時)
 - 做命名實體識別(Named Entity Recognition)

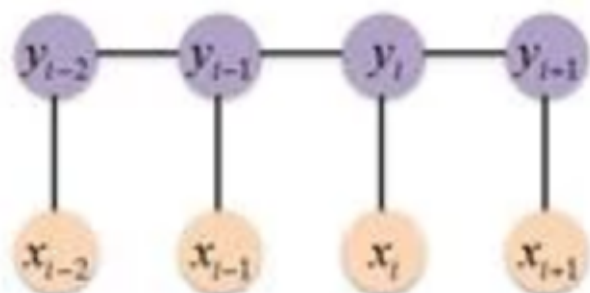
NER研究進展



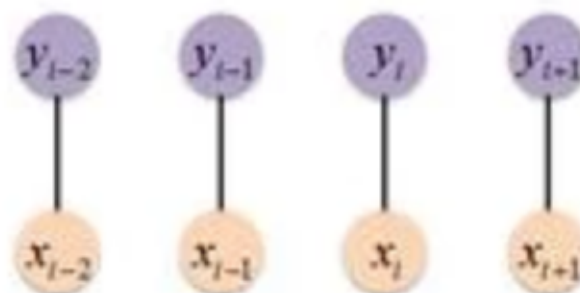
序列標注方法

- 在基於機器學習的方法中，NER被當作是序列標註問題
- 序列標註問題中當前的預測標籤不僅與當前的輸入特徵相關，還與之前的預測標籤相關，即預測標籤序列之間是有強相互依賴關係

序列標注方法



分類方法



序列標注問題

■ I come from New York

▣ 對應的標註是：O O O B-loc I-loc O

▣ New York 標示成 B-loc I-loc，B表示開頭，I表示之後的字，loc代表自己定義entity的類別

■ I come from Taiwan

■ 對應的標註是：O O O B-loc O

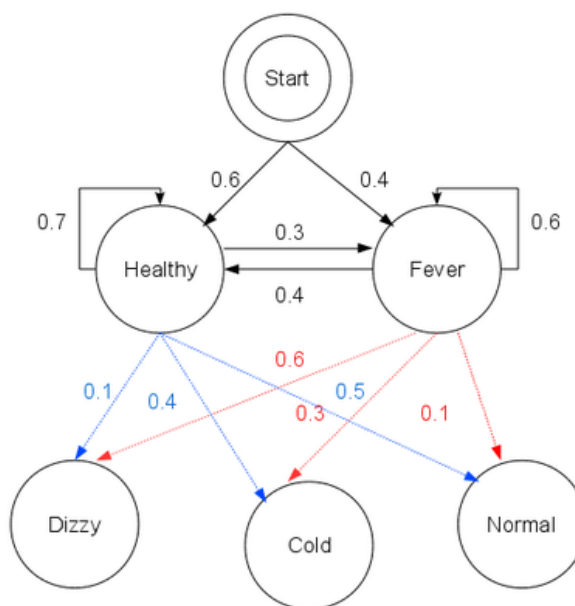
使用者可以自訂所需要的entity

NER 方式

- 任務簡單或訓練資料量很少，用正則表達式或直接比對資料庫
- 如果訓練資料量夠多的話就可以用：HMM或CRF
- HMM 或 CRF 都可以寫成Evaluation 與 Inference 兩個步驟
 - Evaluation: 定義 $F(x,y)$ ， x 代表輸入序列， y 代表輸出序列， $F(x,y)$ 代表好壞程度，值越大代表 y 越符合我們的需要
 - Inference: 在所有可能的 y 集合裡找到一組 y 能最大化 $F(x,y)$ 的值

隱馬爾可夫模型

- 用來描述一個含有隱含未知參數的馬爾可夫過程
- 目的是從可觀察的參數中確定該過程的隱含參數。然後利用這些參數來作斷詞



誰是馬可夫？

- Andrey Markov

(14 June 1856 N.S. – 20 July 1922)

- Calculated letter sequences of the Russian language



問題描述

■ States -> "F", "L"

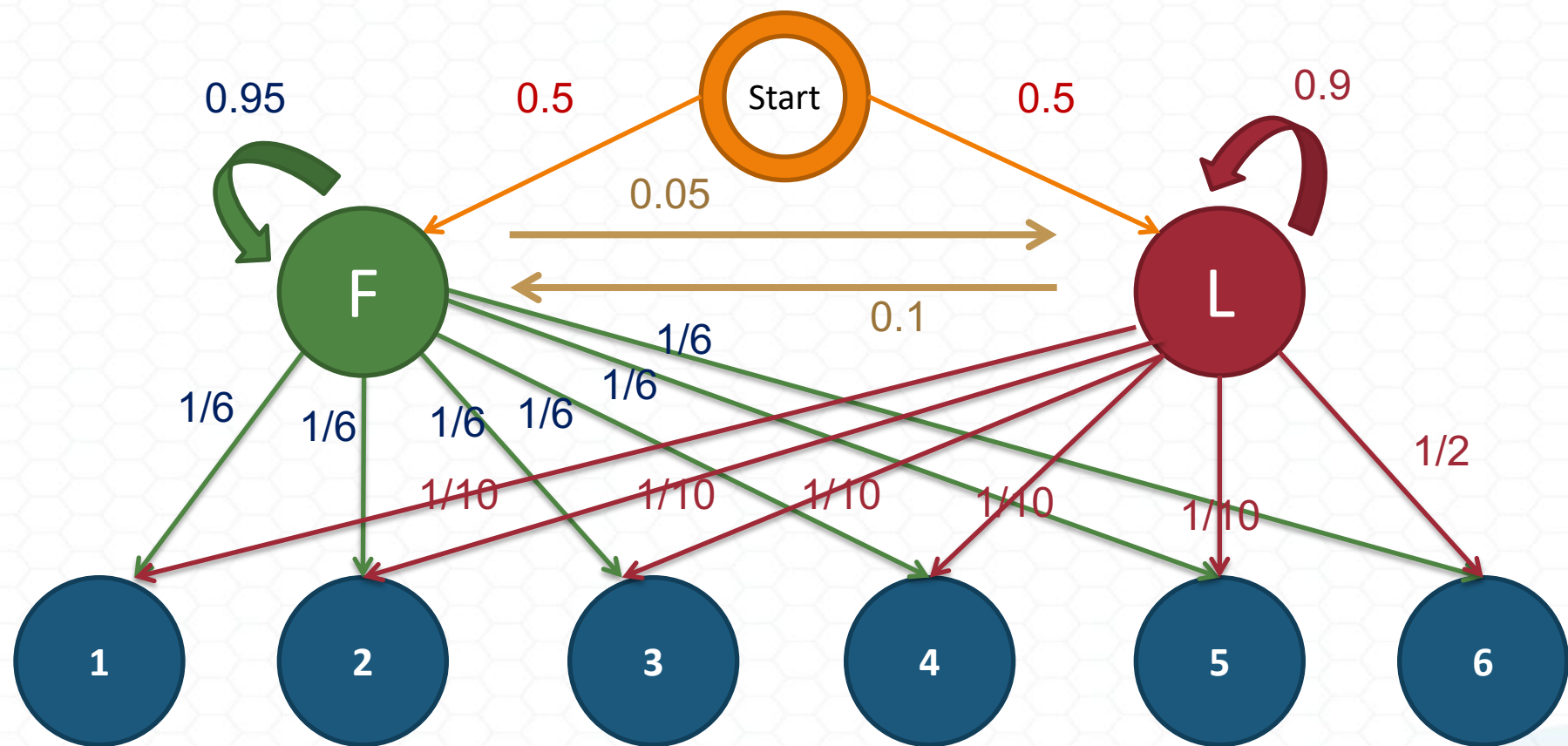
■ Transition Matrix

	Fair	Loaded
Fair	0.95	0.05
Loaded	0.1	0.9

■ Emission Matrix

	Fair	Loaded
1	1/6	1/10
2	1/6	1/10
3	1/6	1/10
4	1/6	1/10
5	1/6	1/10
6	1/6	1/2

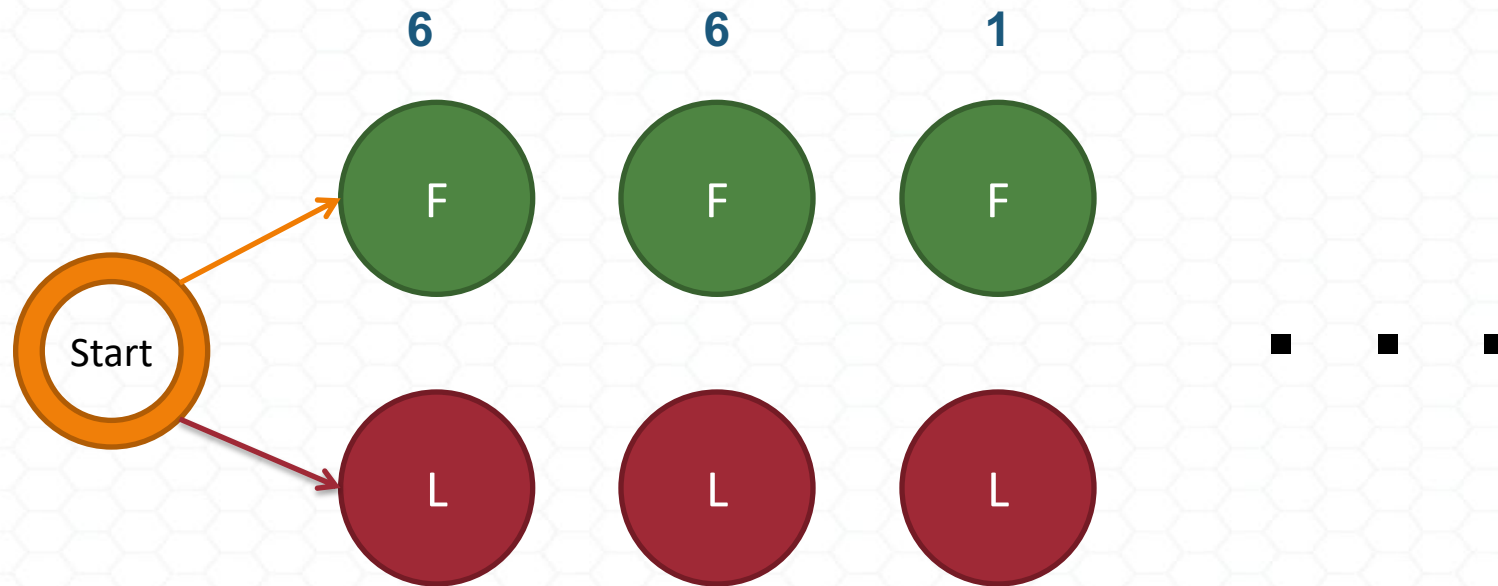
Problem Description



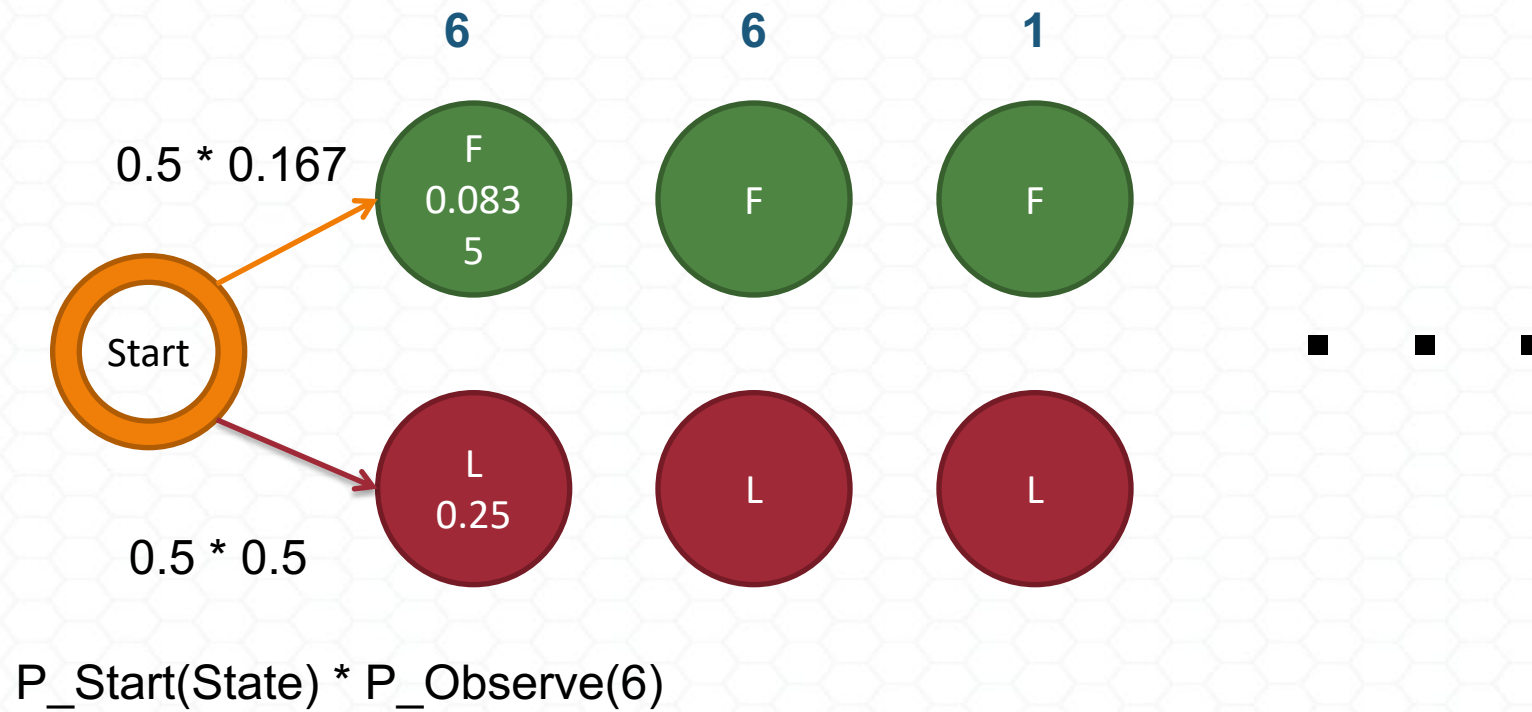
Algorithm

■ Given Observable Sequence:

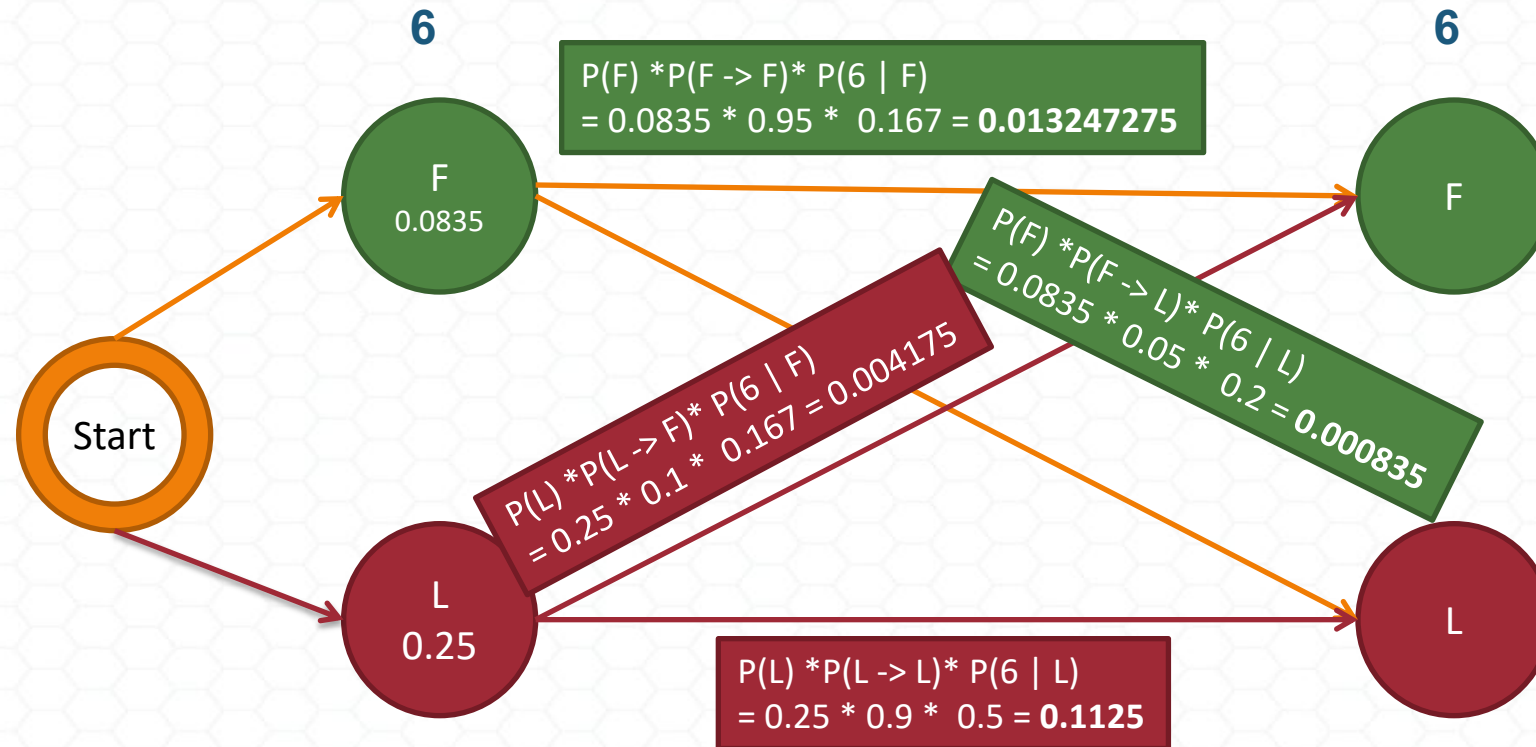
6,6,1,5,3,2...



Start to Step 1

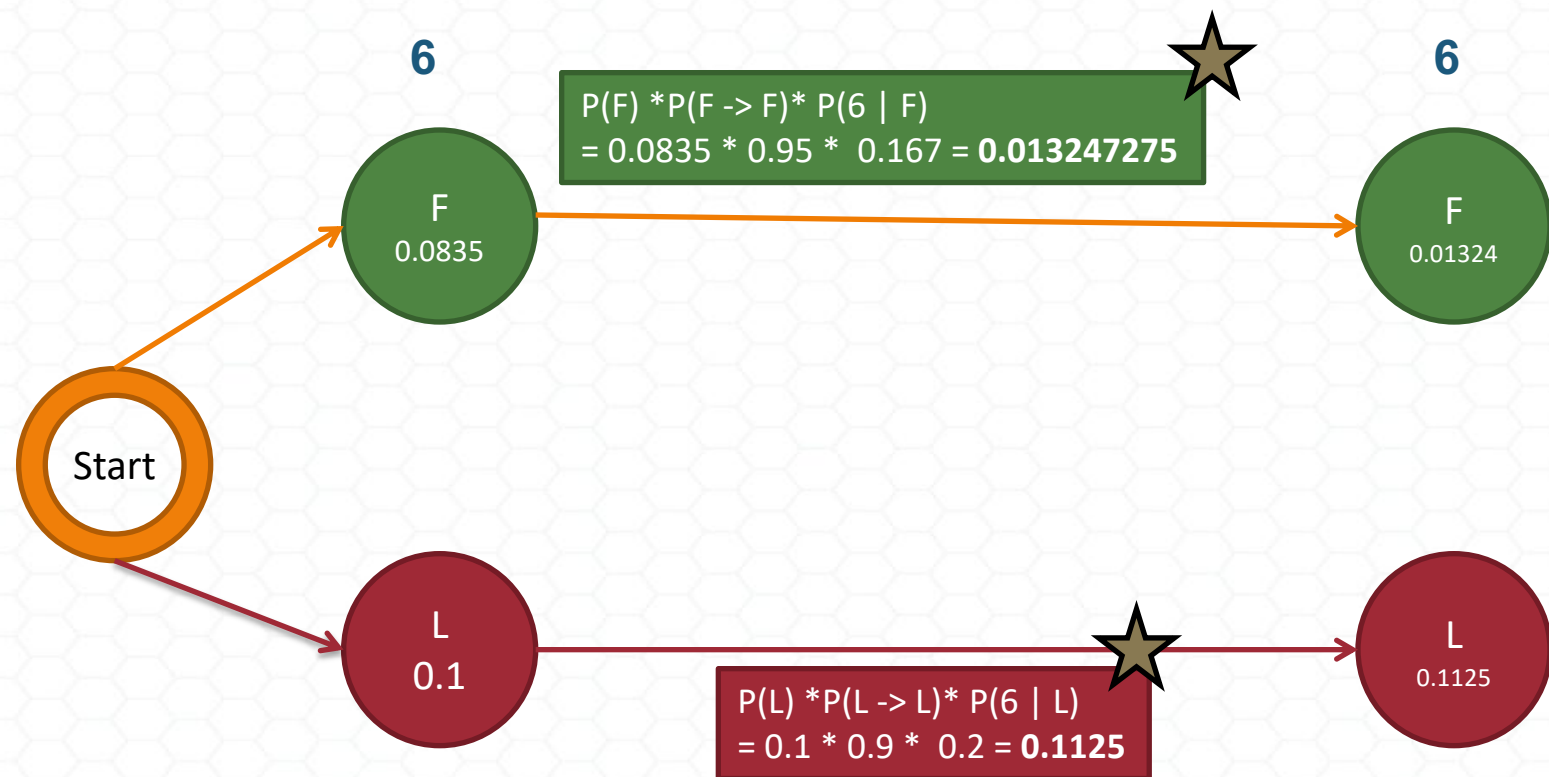


Step 1 to Step 2

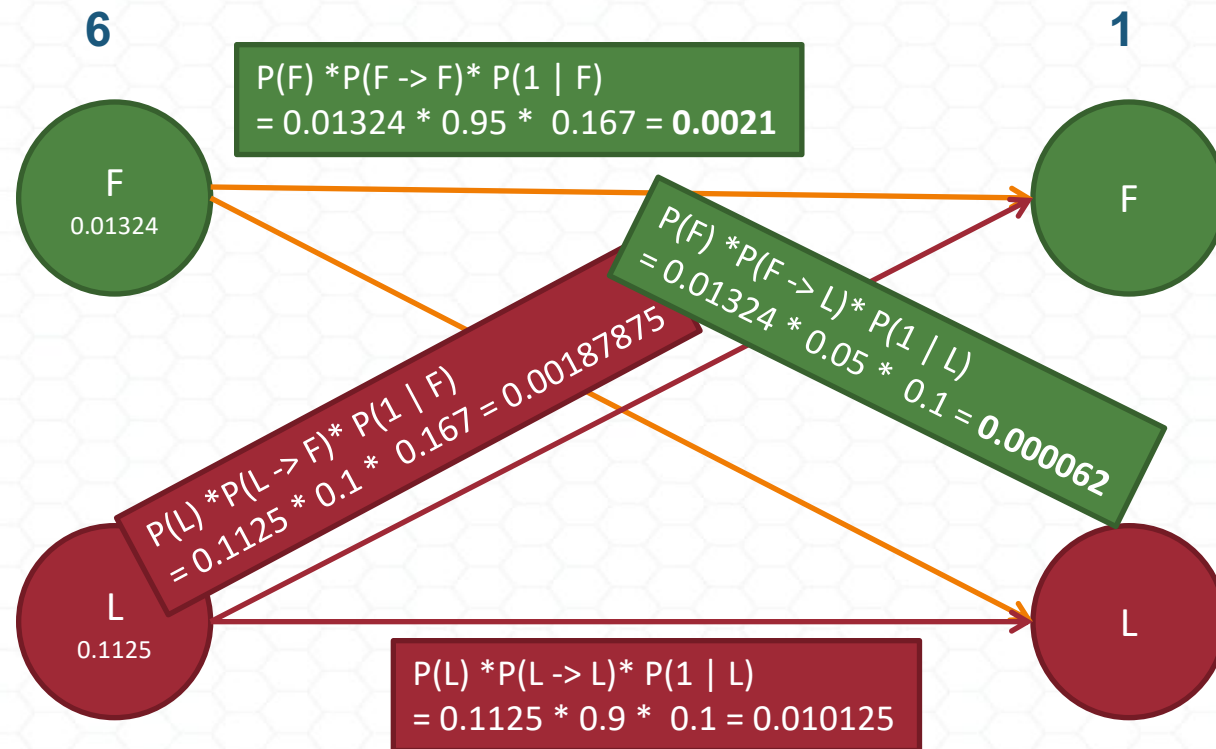


$$P_OldState(State) * P_Trans(Old_State \rightarrow New_State) * P_Observe(6 | New_State)$$

Most Likely Path

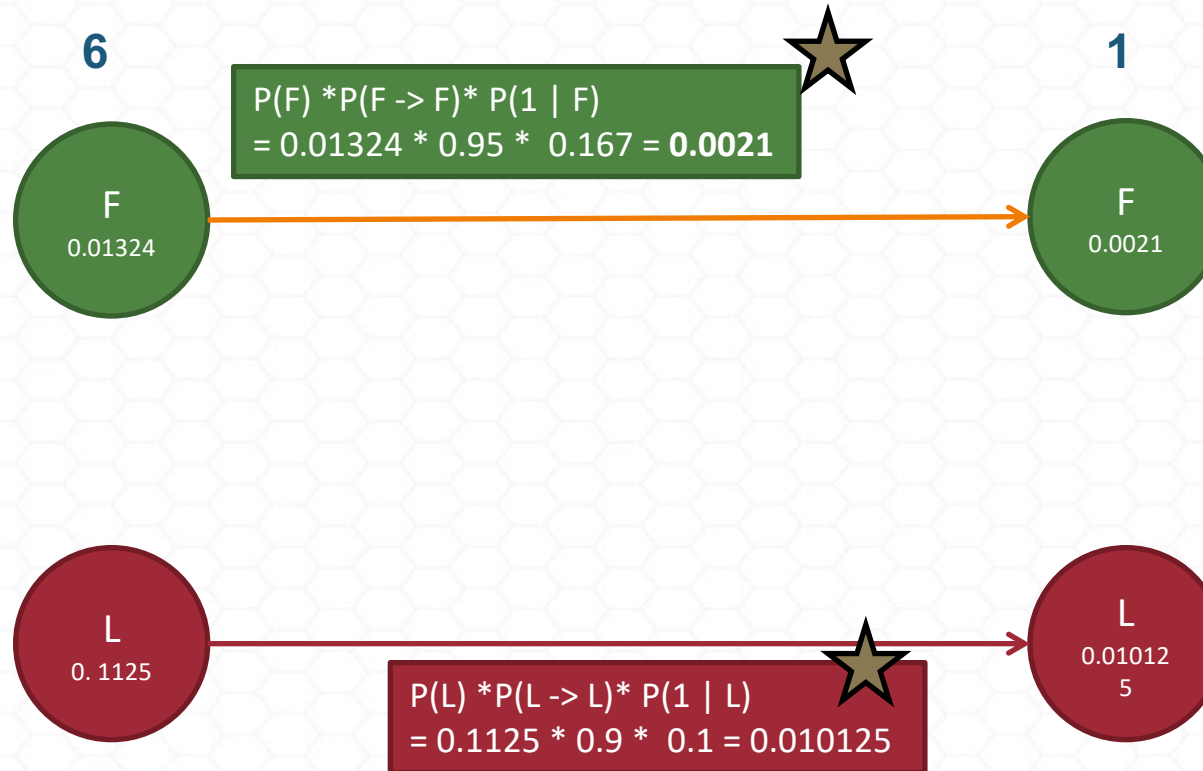


Step 2 to Step 3



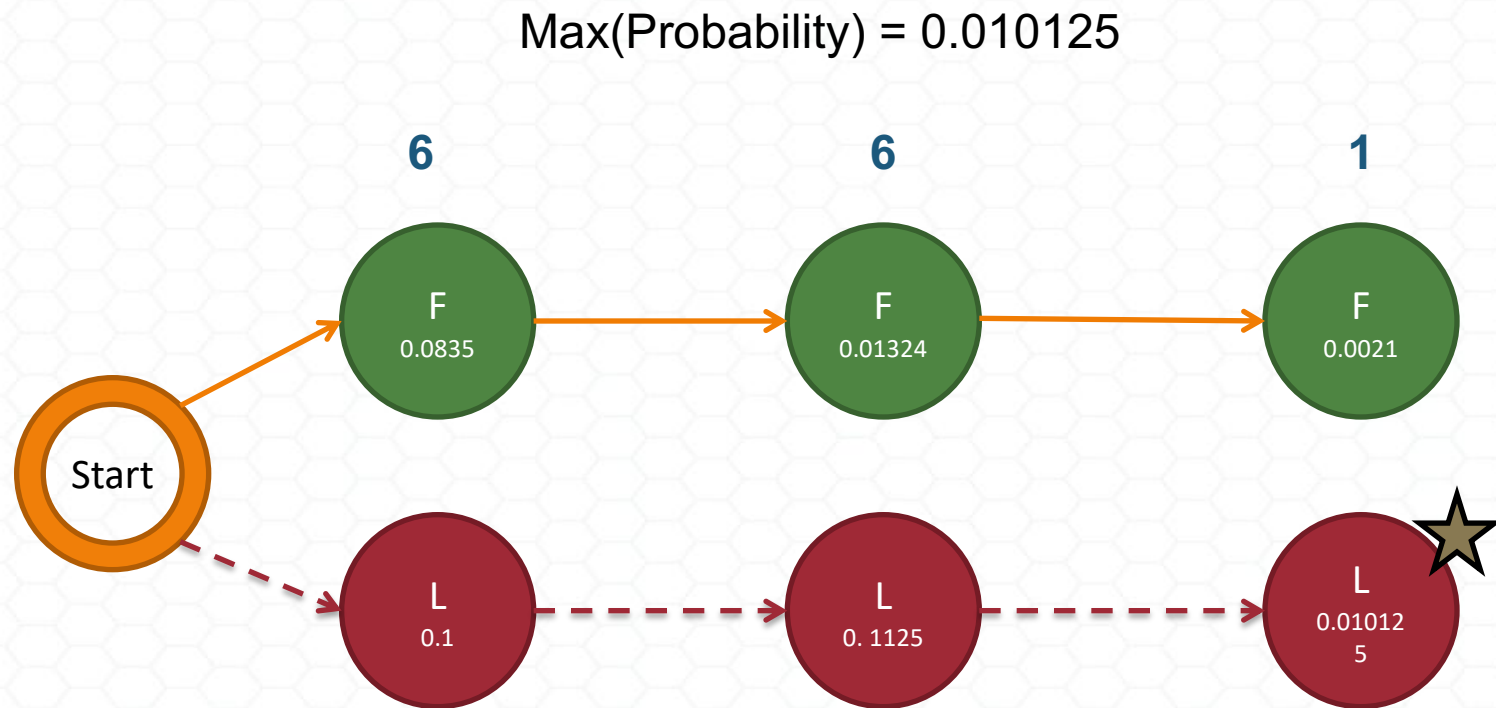
$$P_OldState(State) * P_Trans(Old_State \rightarrow New_State) * P_Observe(1 | New_State)$$

Most Likely Path

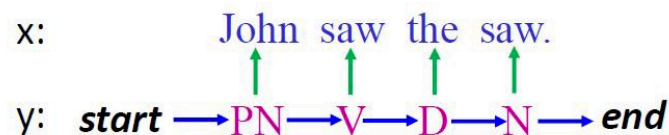


$$P_OldState(State) * P_Trans(Old_State \rightarrow New_State) * P_Observe(1 | New_State)$$

Path Construction



HMM



$$P(x,y) = P(y)P(x|y)$$

$$\begin{aligned} P(y) &= P(PN|start) \\ &\times P(V|PN) \\ &\times P(D|V) \\ &\times P(N|D) \end{aligned} \qquad \begin{aligned} P(x|y) &= P(John|PN) \\ &\times P(saw|V) \\ &\times P(the|D) \\ &\times P(saw|N) \end{aligned}$$

- $P(y)$ 裡的每一項可以由訓練數據統計得到，這裡我們會得到一個N維向量代表Start Probability
- N*N的矩陣代表Transition Probability
- $(x|y)$ 代表有了這個標註後，產生這個詞的機率，也可以由統計後得到，寫成N*M的矩陣代表Emission Probability代表有多少種類別標籤

HMM Inference

- 用 Viterbi 演算法窮舉所有的 y

$$\tilde{y} = \arg \max_{y \in \mathbb{Y}} P(x, y)$$

- 選出最高的 y 當成標注結果

條件隨機域(CRF)

- CRF中的特徵向量，接受四個參數：
 - 句子 s （就是我們要標註詞性的句子）
 - i ，用來表示句子 s 中第 i 個單詞
 - l_i ，表示要評分的標註序列給第 i 個單詞標註的詞性
 - l_{i-1} ，表示要評分的標註序列給第 $i-1$ 個單詞標註的詞性
- 它的輸出值是0或者1, 0表示要評分的標註序列不符合這個特徵，1表示要評分的標註序列符合這個特徵。

條件隨機域(CRF)

- 定義好一組特徵函數後，我們要給每個特徵函數 f_j 賦予一個權重 λ_j 。
。現在，只要有一個句子 s ，有一個標註序列 l ，我們就可以利用前面定義的特徵函數集來對 l 評分

$$score(l|s) = \sum_{j=1}^m \sum_{i=1}^n \lambda_j f_j(s, i, l_i, l_{i-1})$$

求每一個特徵函數 f_j 評分值的和

求句子中每個位置的單詞的特徵值的和

CRF 範例(1)

$$f_1(s, i, l_i, l_{i-1}) = 1$$

- 當 l_i 是”副詞”並且第 i 個單詞以”ly”結尾時，我們就讓 $f_1 = 1$ ，其他情況 f_1 為0
- f_1 特徵函數的權重 λ_1 應當是正的。而且 λ_1 越大，表示我們越傾向於採用那些把以”ly”結尾的單詞標註為”副詞”的標註序列

CRF 範例(2)

$$f_2(s, i, l_i, l_{i-1}) = 1$$

- 如果 $i=1$ ， l_i =動詞，並且句子 s 是以“？”結尾時， $f_2=1$ ，其他情況 $f_2=0$
- λ_2 應當是正的，並且 λ_2 越大，表示我們越傾向於採用那些把問句的第一個單詞標註為“動詞”的標註序列。

CRF 範例(3)

$$f_3(s, i, l_i, l_{i-1}) = 1$$

- 當 l_{i-1} 是介詞， l_i 是名詞時， $f_3 = 1$ ，其他情況 $f_3=0$ 。 λ_3 也應當是正的，並且 λ_3 越大，說明我們越認為介詞後面應當跟一個名詞

CRF 範例(4)

$$f_4(s, i, l_i, l_{i-1}) = 1$$

- l_i 和 l_{i-1} 都是介詞，那麼 f_4 等於1，其他情況 $f_4=0$ 。這裡，我們應當可以想到 λ_4 是負的，並且 λ_4 的絕對值越大，表示我們越不認可介詞後面還是介詞的標註序列。

條件隨機域(CRF)

$$score(l|s) = \sum_{j=1}^m \sum_{i=1}^n \lambda_j f_j(s, i, l_i, l_{i-1})$$

求每一個特徵函數 f_j 評分值的和

求句子中每個位置的單詞的特徵值的和

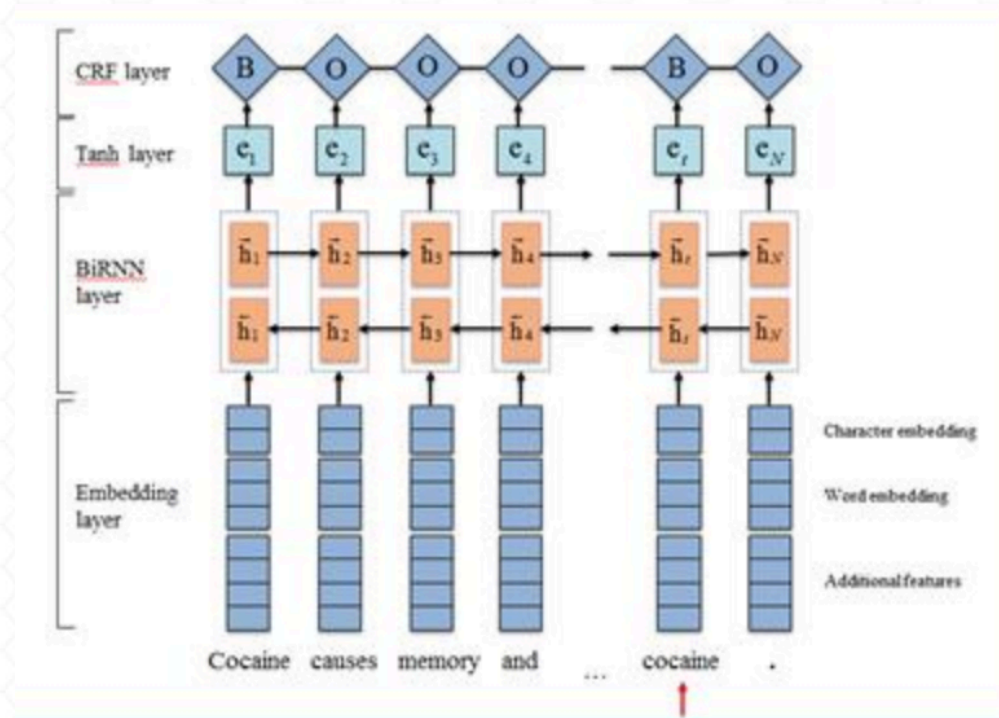
- 建條件隨機場，我們首先要定義一個特徵函數集，每個特徵函數都以整個句子 s ，當前位置 i ，位置 i 和 $i-1$ 的標籤為輸入。然後為每一個特徵函數賦予一個權重，然後針對每一個標註序列 l ，對所有的特徵函數加權求和，也可以把求和的值轉化為一個概率值

CRF與HMM的比較

- HMM 可以視為 CRF的特殊類型
- HMM模型中，當前的單詞只依賴於當前的標籤，當前的標籤只依賴於前一個標籤。只能定義局部性的特徵函數
- CRF卻可以著眼於整個句子 s 定義更具有全局性的特徵函數

神經網路方法

- CRF, HMM 等方法，還是需要定義出特徵，使用深度學習方法可以直接實現End-to-end Learning



深度學習方法與 CRF 方法的比較

■ 深度學習方法並沒有一定的優勢

模型/实体类型	地名	组织	人名
<u>BILSTM+softmax</u>	85%	70%	81%
BILSTM+CRF	84%	85%	91%
作者(40轮)	91%	85%	87%
CRF++	91%	85%	86%

知乎 @陈海斌

NER 實作

NLTK

```
import nltk  
from nltk.tokenize import word_tokenize  
from nltk.tag import pos_tag
```

```
sent = nltk.word_tokenize(sent)  
sent = nltk.pos_tag(sent)  
sent = preprocess(ex)  
sent
```

再利用詞性萃取出專有名詞

Spacy

■ 工業級文字處理工具

The screenshot shows the spaCy website's 'Install spaCy' page. The sidebar on the left contains links for 'GET STARTED' (Installation, Quickstart, Instructions, Troubleshooting, Changelog), 'Models & Languages', 'Facts & Figures', and 'GUIDES' (Linguistic Features, Rule-based Matching, Processing Pipelines, Vectors & Similarity, Training Models, Saving & Loading). The main content area has a header 'Install spaCy' and a paragraph stating: 'spaCy is compatible with **64-bit CPython 2.7 / 3.5+** and runs on **Unix/Linux, macOS/OS X and Windows**. The latest spaCy releases are available over [pip](#) and [conda](#).' Below this is a 'Quickstart' section with tabs for 'macOS/OS X', 'Windows', and 'Linux'. A red box at the bottom contains the following commands:

```
pip install -U spacy
python -m spacy download xx_ent_wiki_sm
```

On the right side of the page, there is a blue patterned background with a dark overlay box containing the text: 'LOOKING FOR THE OLD DOCS? To help you make the transition from v1.x to v2.0, we've uploaded the old website to [legacy.spacy.io](#). Wherever possible, the new docs also include notes on features that have changed in v2.0, and features that were introduced in the new version.'

Spacy

- SpaCy的命名實體識別已經在OntoNotes 5語料庫上進行了訓練，它支持以下實體類型

TYPE	DESCRIPTION
PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.
LANGUAGE	Any named language.
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day.
PERCENT	Percentage, including "%".
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	"first", "second", etc.
CARDINAL	Numerals that do not fall under another type.

Entity(命名)

```
import spacy
from spacy import displacy
from collections import Counter
import en_core_web_sm
nlp = en_core_web_sm.load()
```

```
doc = nlp('European authorities fined Google a record $5.1 billion on  
Wednesday for abusing its power in the mobile phone market and ordered the  
company to alter its practices')
pprint([(X.text, X.label_) for X in doc.ents])
```

列出標記

```
pprint([(X, X.ent_iob_, X.ent_type_) for X in doc])
```

TAG	DESCRIPTION
BEGIN	The first token of a multi-token entity.
IN	An inner token of a multi-token entity.
LAST	The final token of a multi-token entity.
UNIT	A single-token entity.
OUT	A non-entity token.

視覺化呈現標記

```
displacy.render(nlp(str(sentences[20])), jupyter=True, style='ent')
```

Firing Mr. Strzok PERSON, however, removes a favorite target of Mr. Trump PERSON from the ranks of the F.B.I. GPE and gives Mr. Bowdich PERSON and the F.B.I. GPE director, Christopher A. Wray PERSON, a chance to move beyond the president's ire.

ckiptagger

■ 中研院開源出來的切詞套件

□ <https://github.com/ckiplab/ckiptagger>

ckiplab / ckiptagger

Watch 62 Unstar 1.1k Fork 114

<> Code Issues 4 Pull requests 3 Projects 0 Wiki Security Insights

CKIP Neural Chinese Word Segmentation, POS Tagging, and NER

54 commits 1 branch 5 releases 3 contributors GPL-3.0

Branch: master New pull request Create new file Upload files Find file Clone or download

ckipma Update README.md	Latest commit b2a3237 on Sep 10
src	support typo downlaod last month
LICENSE	change license last month
README.md	Update README.md last month
demo.py	rename to ckiptagger 2 months ago
setup.py	update last month

`pip install -U ckiptagger`

ckiptagger

■ 切詞準確度最高

Tool	(WS) prec	(WS) rec	(WS) f1	(POS) acc
CkipTagger	97.49%	97.17%	97.33%	94.59%
CKIPWS (classic)	95.85%	95.96%	95.91%	90.62%
Jieba-zh_TW	90.51%	89.10%	89.80%	--

■ 支援 POS, 切詞與NER 功能

■ GNU General Public License v3.0

Ckistagger 範例

```
from ckistagger import data_utils, construct_dictionary, WS, POS, NER
```

```
ws = WS("./data")  
pos = POS("./data")  
ner = NER("./data")
```

需要先下載資料

```
sentence_list = [
```

"全聯福利中心強力推出「PX Pay」行動支付後，更進一步開放8家銀行的實體信用卡、33家金融機構金融卡、3大電子票證、3大國際行動Pay與台灣Pay等交易。同時釋出8大銀行刷卡優惠，其中聯邦卡首刷500元送1,000點福利點最高，平日則以國泰世華天天消費滿500元送300點最強。",

```
]
```

```
word_sentence_list = ws(  
    sentence_list,  
)
```

```
pos_sentence_list = pos(word_sentence_list)
```

```
entity_sentence_list = ner(word_sentence_list, pos_sentence_list)
```

The background features a light blue hexagonal grid pattern. Overlaid on this is a series of concentric, semi-transparent circles in shades of blue and white, creating a ripple effect. A solid dark blue horizontal line runs across the top of the image, and a dark teal horizontal band is at the bottom.

THANK YOU