D07 使用CkipTagger進行中文斷詞

重要知識點

- 如何使用CkipTagger來進行實際的斷詞操作。

補充資料：簡單易用的中英文斷詞和詞性標註

繁體中文斷詞: CkipTagger

- CkipTagger為台灣中央研究院詞庫小組所開發的NLP(自然語言處理)套件，是個以深度學習模型為基礎而成的NLP(自然語言處理)應用。 其訓練的文本資料來源為中央社、wiki (用舊版套件先進行斷詞) 及 ASBC(Academia Sinica Balanced Corpus)(為期近10年人工標記)。

- CkipTagger模型主要的功能有：
1. WS: 斷詞
2. POS: 詞性標注
3. NER: 實體辨識

系統環境安裝

- 標準安裝 (同時安裝tensorflow, gdown)
- gdown 是與 Google Drive 之間的API，可以從 Google Drive上下載資料

pip install -U ckiptagger[tf,gdown]

```
pip install -U ckiptagger[tf,gdown]
```

- 最小安裝 (不安裝tensorflow, gdown)
- 若環境已有安裝tensorflow, gdown, 可使用此種安裝

pip install -U ckiptagger

```
pip install -U ckiptagger
```

- 完整安裝 (安裝gpu版tensorflow, gdown

pip install -U ckiptagger[tfgpu,gdown]

```
pip install -U ckiptagger[tfgpu,gdown]
```

CkipTagger優勢

在繁體中文上斷詞與詞性標記的表現進一步提升，並超越結巴系統

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%	--

資料來源：github

結合實體命名：目前CkipTagger能辨識11 類一般領域專有名詞及 7 類數量詞

Type	Description	中文
GPE	Countries, cities, states	行政區
PERSON	People, including fictional	人物
DATE	Absolute or relative dates or periods	日期
ORG	Companies, agencies, institutions, etc.	組織
CARDINAL	Numerals that do not fall under another type	數字
NORP	Nationalities or religious or political groups	民族、宗教、政治團體
LOC	Non-GPE locations, mountain ranges, bodies of water	地理區
TIME	Times smaller than a day	時間
FAC	Buildings, airports, highways, bridges, etc.	設施
MONEY	Monetary values, including unit	金錢
ORDINAL	"first", "second"	序數
EVENT	Named hurricanes, battles, wars, sports events, etc.	事件
WORK_OF_ART	Titles of books, songs, etc.	作品
QUANTITY	Measurements, as of weight or distance	數量
PERCENT	Percentage (including "%")	百分比率
LANGUAGE	Any named language	語言
PRODUCT	Vehicles, weapons, foods, etc. (Not services)	產品
LAW	Named documents made into laws	法律

資料來源：github

結合詞性標注

Type	Description
A	非謂形容詞
Caa	對等連接詞
Cab	連接詞，如：等等
Cba	連接詞，如：的話
Cbb	關聯連接詞
D	副詞
Da	數量副詞
Dfa	動詞前程度副詞
Dfb	動詞後程度副詞
Di	時態標記
Dk	句副詞
DM	定量式
I	感嘆詞
Na	普通名詞
Nb	專有名詞
Nc	地方詞
Ncd	位置詞
Nd	時間詞
Nep	指代定詞
Neqa	數量定詞

資料來源：github

CkipTagger的優勢
- 支援使用者自訂 參考/強制 詞典。
- 支援不限長度的句子。
- 不會自動 增/刪/改 輸入的文字。

CkipTagger斷詞技巧

- Word-level approach
  - Maximum length (長詞優先)
  - 動態規劃查找最大概率路徑(Jieba)

- Character-level approach
  - character Sequence Labeling

- CkipTagger 則是綜合上面兩種方法，針對 word 及 character 同時進行分析

以CkipTagger進行斷詞

- 下載預訓練權重
- 此模型需2GB的儲存空間

```
from ckiptagger import data_utils, WS
data_utils.downlaod_data_gdown("./")
```

```
from ckiptagger import data_utils, WS
data_utils.download_data_gdown("./")
```

- 建構斷詞器
- 使用Ckiptagger進行斷詞

```
#建構斷詞器
ws = WS("./data/")

#使用Ckiptagger進行斷詞
#返回為斷詞完成的list
input_string = '小明碩士畢業於國立臺灣大學，現在在日本東京大學進修深造'
word_sentence_list = ws(
    input_string,
    sentence_segmentation = True, # To consider delimiters
    segment_delimiter_set = {",", "。", ":", "?", "!", ";"})
print(word_sentence_list)
```

```
[['小'], ['明'], ['碩'], ['士'], ['畢'], ['業'], ['於'], ['國'], ['立'], ['臺'], ['灣'], ['大'], ['學'], ['，'], ['現'], ['在'], ['在'], ['日'], ['本'], ['東'], ['京'], ['大'], ['學'], ['進'], ['修'], ['深'], ['造']]
```

```
#建構斷詞
ws = WS("./data")

input_string = '小明碩士畢業於國立臺灣大學，現在在日本東京大學進修深造'
word_sentence_list = ws(
    input_string,
    sentence_segmentation = True, # To consider delimiters
    segment_delimiter_set = {",", "，", ":", "?", "!", ";"}) # This is the defualt set of delimiter
print(word_sentence_list)
```

以CkipTagger詞性標注

使用Ckiptagger詞性標注

```
from ckiptagger import POS

# 建構詞性標注器
pos = POS("./data")

pos_sentence_list = pos(word_sentence_list) #帶入的是經過斷詞後的 list，不是原始文本
print(pos_sentence_list)
```

```
[['VH'], ['Nd'], ['Na'], ['Na'], ['VH'], ['Na'], ['P'], ['Nc'], ['VC'], ['Nc'], ['Na'], ['VH'], ['VC'], ['COMMACATEGORY'], ['D'], ['P'], ['P'], ['Nc'], ['Nes'], ['Ncd'], ['Nc'], ['VH'], ['VC'], ['VCL'], ['VC'], ['VH'], ['VC']]
```

```
from ckiptagger import POS

pos = POS("./data")
pos_sentence_list = pos(word_sentence_list)
print(pos_sentence_list)
```

以CkipTagger命名實體辨識

使用Ckiptagger命名實體識別

帶入的是經過斷詞後的 list，不是原始文本
帶入的是經過詞性標注的 list，不是原始文本

```
from ckiptagger import NER

#建構命名實體器
ner = NER("./data")

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

```
[set(), set(), set(), set(), set(), set(), set(), set(), set(), {(0, 1, 'GPE', '臺')}, set(), set(), set(), set(), set(), set(), set(), {(0, 1, 'GPE', '日')}, set(), set(), {(0, 1, 'GPE', '京')}, set(), set(), set(), set(), set(), set()]
```

```
ner = NER("./data")

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

帶入自定義字典

將自定義字典加入斷詞器中

```
from ckiptagger import construct_dictionary

word_to_weight = {"日本東京大學": 1}
dictionary = construct_dictionary(word_to_weight)
```

```
from ckiptagger import construct_dictionary

word_to_weight = {"日本東京大學": 1}
dictionary = construct_dictionary(word_to_weight)
```

建構斷詞器

```
ws = WS("./data/")
input_traditional_str = ['小明碩士畢業於國立臺灣大學，現在在日本東京大學進修深造']
word_sentence_list = ws(input_traditional_str, recommend_dictionary=dictionary)
print(word_sentence_list)
```

```
[['小明', '碩士', '畢業', '於', '國立', '臺灣', '大學', '，', '現在', '在', '日本東京大學', '進修', '深造']]
```

```
ws = WS("./data")
input_traditional_str = ['小明碩士畢業於國立臺灣大學，現在在日本東京大學進修深造']
word_sentence_list = ws(input_traditional_str, recommend_dictionary=dictionary)
print(word_sentence_list)
```

詳細使用操作

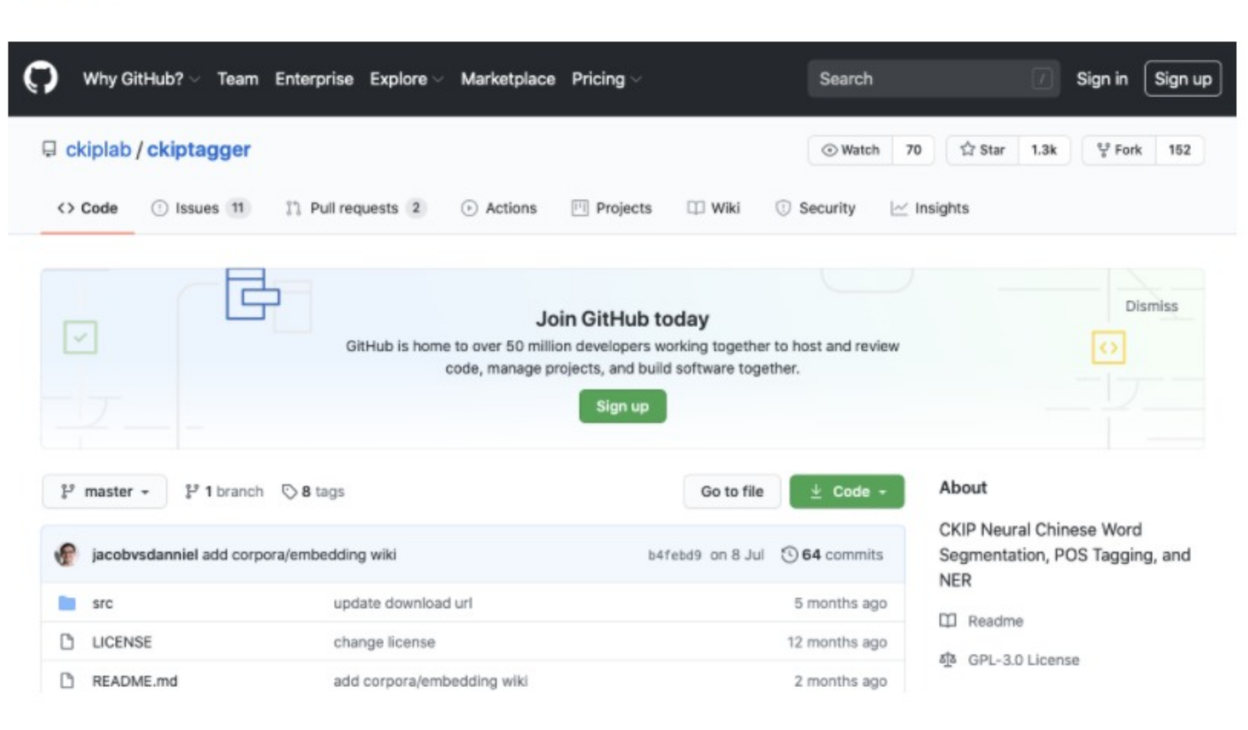
請參照使用使用CkipTagger進行繁體中文斷詞.ipynb檔進行更詳細的使用操作

知識點回顧

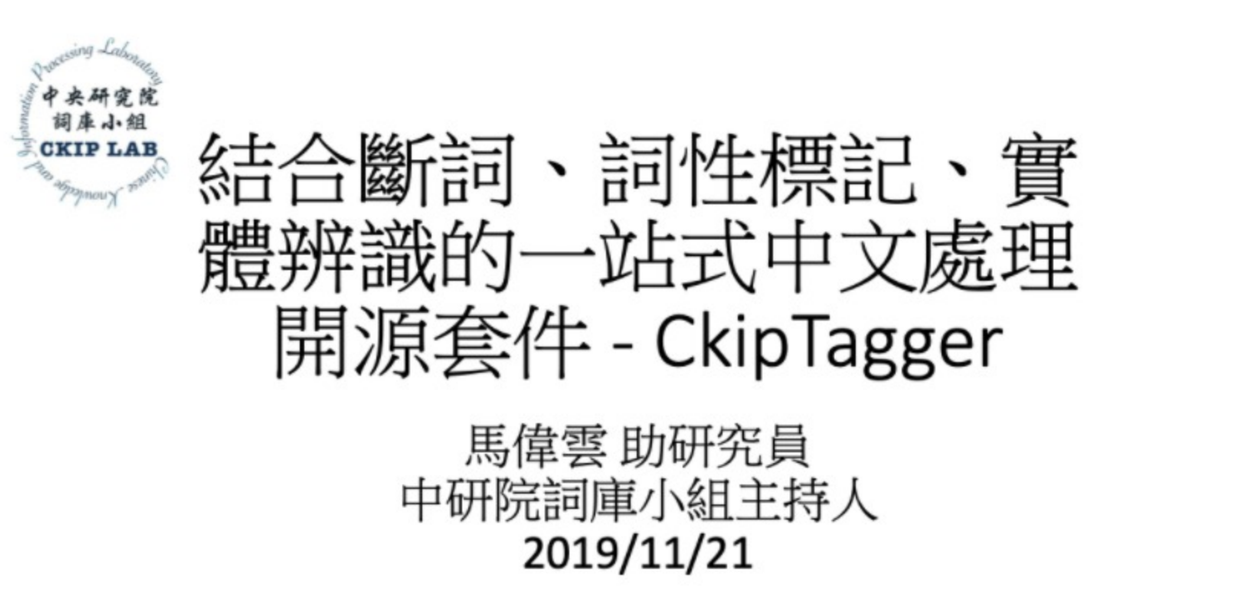
在這章節我們學習到
- 使用CkipTagger來進行各項任務操作(如斷詞)

延伸閱讀

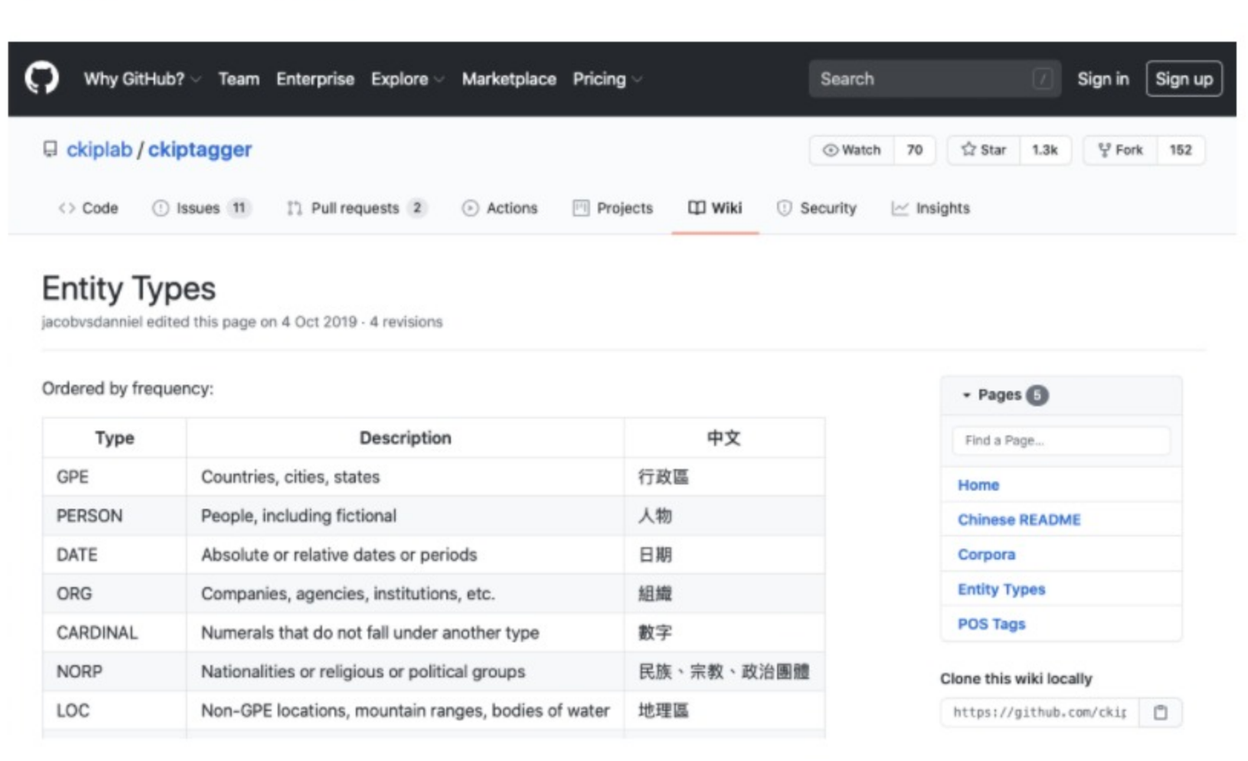
網站：CkipTagger官方Github
CkipTagger官方Github(提供操作說明)



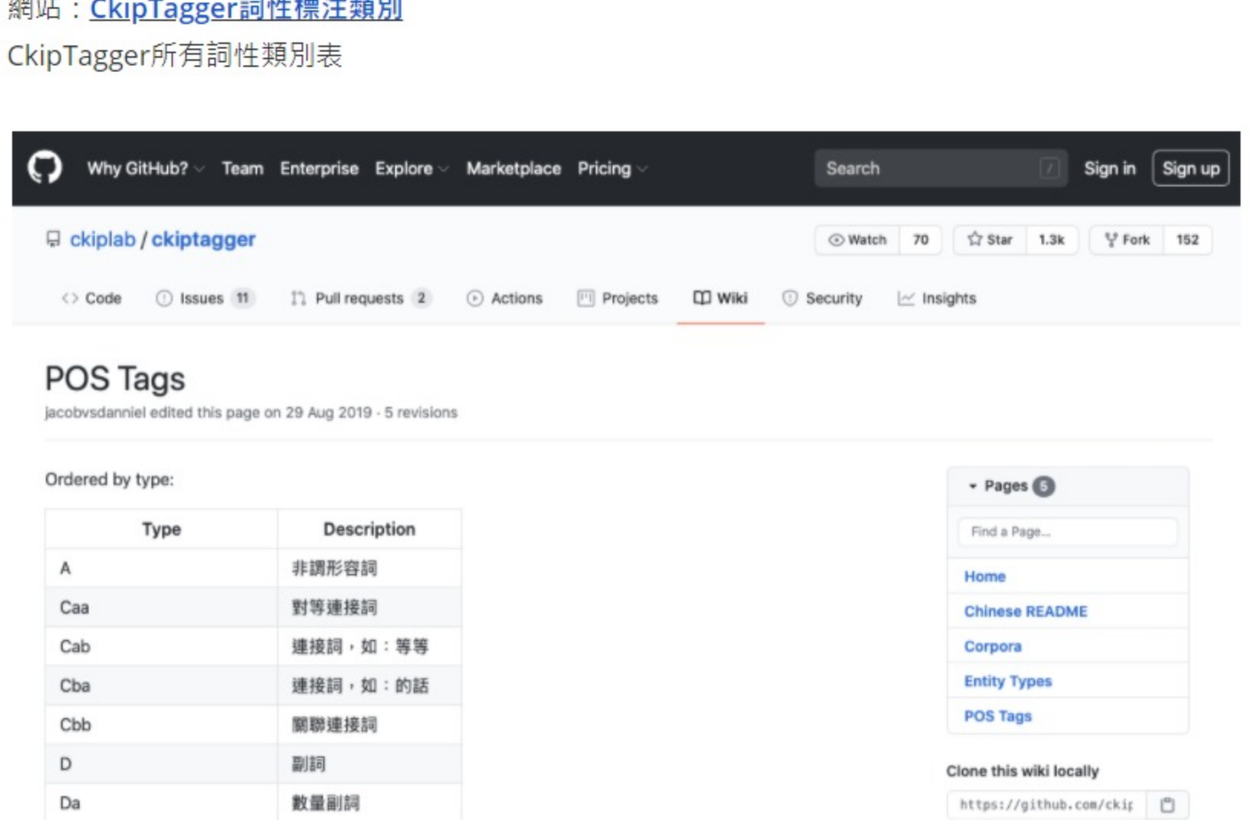
網站：中研院詞庫小組馬偉雲主持人演講
有關CkipTagger內部詳細介紹



網站：CkipTagger命名實體類別
CkipTagger所有命名實體類別表



網站：CkipTagger詞性標注類別
CkipTagger所有詞性類別表



網站：中研院詞庫小組馬偉雲主持人演講筆記
有關CkipTagger與NLP發展介紹筆記整理

下一步：閱讀範例與完成作業