

還是要

海嘯斷詞的吧！

Make AI Great Again

這裡有幾個字 (word) ?

這個星期日本想往後山藥師佛寺去世人罕至處想一想自己的人生

這個/星期日/本/想/往/後山/藥師佛寺/去/世人/罕至處/想一想/自己/的/人生

字數:14

字符數: 28

詞數:14

N-Gram

這個星期日本想往後山藥師佛寺去世人罕至處想一想自己的人生

uni-gram:

這/個/星/期/日/本/想/往/後/山/藥/師/佛/寺/去/世/人/罕/至/處/想/一/想/自/己/的/人/生

bi-gram:

這/個/個/星/星/期/期/日/日/本/本/想/想/往/往/後/後/山/山/藥/藥/師/師/佛/佛/寺/寺/去/去/世/世/人/人/罕/罕/至/至/處/處/想/想/一/一/想/想/自/自/己/己/的/的/人/人/生

tri-gram:

...

Jieba 結巴斷詞

<https://github.com/fxsjy/jieba>

❖ 背景：

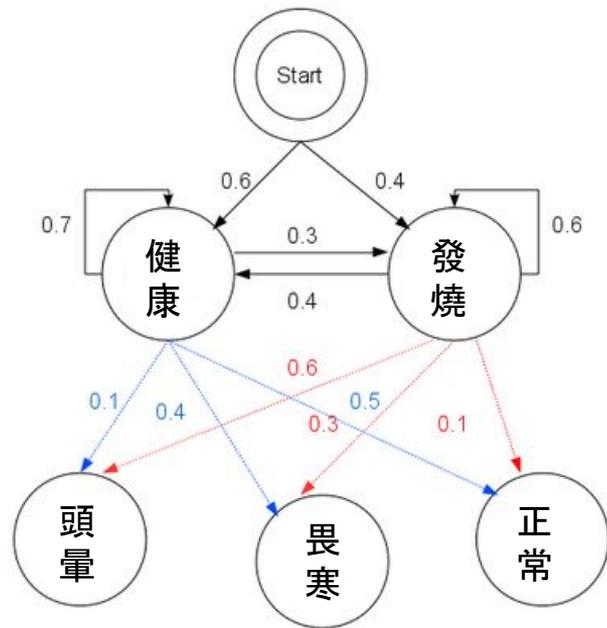
- 1998 人民日報
- 新詞基於 HMM (隱馬可夫模型 Hidden Markov Model)
 - [抗击新冠肺炎疫情斗争取得重大战略成果]
 - [抗擊新冠肺炎疫情鬥爭取得重大戰略成果]

❖ 重點：

- TF-IDF

❖ 使用方法：

- 安裝: pip3 install jieba
- 使用: 見文件



Jieba

```
Loading model cost 1.353 seconds.  
Prefix dict has been built succesfully.  
['這個',  
 '星期',  
 '日本',  
 '想',  
 '往後山',  
 '藥師',  
 '佛寺',  
 '去世',  
 '人罕至',  
 '處',  
 '想一想',  
 '自己',  
 '的',  
 '人生']
```

Machine Learning

請輸入欲處理的文字（限繁體中文）：

這個星期日本想往後山藥師佛寺去世人罕至處想一想自己的人生

Submit

全部顯示
 分開顯示

斷詞系統
 剖析系統

實體辨識

語義辨識 (β)

指代消解 (β)

關係抽取 (β)

斷詞系統

Show POS tagging

這_(Nep) 個_(Nf) 星期_(Na) 日本_(Nc) 想_(VE) 往_(P) 後山藥師_(Nb) 佛寺_(Nc) 去_(VCL) 世人_(Na) 罕_(D) 至處_(Nc) 想_(VE) 一_(D) 想_(VE) 自
己_(Nh) 的_(DE) 人生_(Na)

LLM

ChatGPT 4 ▾

↑ 分享



幫我把以下的句子做中文斷詞處理：
這個星期日本想往後山藥師佛寺去世人罕至處想一想自己的人生



這個星期/日本/想/往/後山/藥師佛寺/去/世人罕至處/想/一想/自己的/人生

这里的“日本”可能是切分错误，如果“日本”是表示国家，那么可能不应该将其与“这个星期”连在一起。如果是表达“这个星期，我想去...”，则应该是“这个星期/我/想往/后山/藥師佛寺/去/世人罕至处/想/一想/自己的/人生”。



CTWANT | 49.1k 人追蹤

☆ 追蹤

「AI教父」獲諾貝爾物理獎...家族全是超狂學霸 他童年壓力 超大畢業當過木匠



周刊王CTWANT | 林立浩

2024年10月9日 週三 下午12:05



加拿大科學家辛頓。(圖/達志/美聯社,下同)

Linguistics-Based Solutions



我來了

因為

已經沒事了

註冊並取得教學用 API 金鑰



The screenshot shows a web browser window with the URL <https://api.droidtown.co/member/>. The page title is "服務資訊" (Service Information). Below the title, there are six service tiles: Articut, KeyMoji, Loki, CopyToaster, Space, and 達文析. Below these tiles, the "API 金鑰" (API Key) is displayed as pEWjfx317g*****Weyirr&JR6. There are two buttons: "複製 API 金鑰" (Copy API Key) and "更新 API 金鑰" (Update API Key). The "剩餘額度" (Remaining Quota) is shown as 676,757 字 (characters).

點擊 [複製 API 金鑰] 鈕, 即可取得金鑰

基本操作範例：輸入

請將 username 更換為
您的 Articut 帳號Email

```
1  #!/usr/bin/env python3
2  #-*- coding:utf-8 -*-
3
4
5  from ArticutAPI import Articut
6  from pprint import pprint
7
8
9  if __name__ == "__main__":
10     username = "peter.w@droidtown.co"
11     apikey = "API_Key_Here"
12
13     articut = Articut(username, apikey)
14
15     inputSTR = "會被大家盯上，才證明你有實力"
16     resultDICT = articut.parse(inputSTR)
17     pprint(resultDICT)
```


進階用法之一:lv1 和 lv2 輸入

Articut Level斷詞的深度。數字愈小，切得愈細。

比如說「小紅帽」

在 lv1 的設定下，將會回傳
為 [小/紅/帽]。

但在 lv2 的設定下，則會回傳
[小紅帽]。

```
1  #!/usr/bin/env python3
2  #-*-coding:utf-8-*-
3
4
5  from ArticutAPI import Articut
6  from pprint import pprint
7
8
9  if __name__ == "__main__":
10     username = "peter.w@droidtown.co"
11     apikey = "API_Key_Here"
12
13     articut = Articut(username, apikey)
14
15     inputSTR = "小紅帽"
16     resultDICT = articut.parse(inputSTR, level="lv1") #注意 level 參數設定為 "lv1"
17     print("\n lv1 的設定下，處理結果：\n", resultDICT)
18
19     resultDICT = articut.parse(inputSTR, level="lv2") #注意 level 參數設定為 "lv2"
20     print("\n lv2 的設定下，處理結果：\n", resultDICT)
```

進階用法之一:lv1 和 lv2 輸出

比較 lv1 和 lv2 的結果, 可發現在 lv1 中切的極細的「小 / 紅 / 帽」, 在 lv2 中被結合成「小紅帽」一個詞彙。

lv1 的設定下, 處理結果:

```
{'exec_time': 0.020951271057128906, 'result_pos':  
['<MODIFIER>小</MODIFIER><MODIFIER_color>紅</MODIFIER_color><ENTITY_nounHead>帽</ENTITY_nounHead>'],  
'result_segmentation': '小/紅/帽', 'result_obj': [[{'text': '小', 'pos': 'MODIFIER'}, {'text': '紅',  
'pos': 'MODIFIER_color'}, {'text': '帽', 'pos': 'ENTITY_nounHead'}]], 'level': 'lv1', 'version':  
'v233', 'status': True, 'msg': 'Success!', 'word_count_balance': 717862, 'product':  
'https://api.droidtown.co/product/', 'document': 'https://api.droidtown.co/document/'}
```

lv2 的設定下, 處理結果:

```
{'exec_time': 0.02215099334716797, 'result_pos': ['<ENTITY_nouny>小紅帽</ENTITY_nouny>'],  
'result_segmentation': '小紅帽', 'result_obj': [[{'text': '小紅帽', 'pos': 'ENTITY_nouny'}]], 'level':  
'lv2', 'version': 'v233', 'status': True, 'msg': 'Success!', 'word_count_balance': 717859,  
'product': 'https://api.droidtown.co/product/', 'document': 'https://api.droidtown.co/document/'}
```

進階用法:lv1 和 lv2

同理, 在 lv1 下會把動詞和時態標記分開, 因此「創造了」會被切分成「創造了」;但在 lv2 的設定下, 則會把動詞和時態標記結合在一起, 因此「創造了」將在lv2 處理為「創造了」。

這是因為 Articut 將時態標記「了」視為像是英文裡的-ed。因此, 在 lv1 時, 採取將之處理為 "create/-ed" 分開的兩個元素, 但在lv2 的設定下, 則是以 "created" 這種「詞 + 時態標記」的形式輸出。

課間練習:

1. 仿照前例, 用 Python3 設計一段程式, 分別用 "lv1" 和 "lv2" 輸入「閱讀創造了奇蹟」, 觀察 lv1 和 lv2 的設定下, 回傳的結果有何差別。

```

{
  "result_segmentation": "這個/星期日/本/想/往/後/山/藥師佛寺/去/世人/罕至處/想一想/自己/的/人生",
  "result_pos": "<ENTITY_DetPhrase>這個</ENTITY_DetPhrase><TIME_week>星期日</TIME_week><MODIFIER>本</MODIFIER><ACTION_verb>想</ACTION_verb><FUNC_inner>往</FUNC_inner><RANGE_locality>後</RANGE_locality><ENTITY_oov>山</ENTITY_oov><ENTITY_nouny>藥師佛寺</ENTITY_nouny><ACTION_verb>去</ACTION_verb><ENTITY_nouny>世人</ENTITY_nouny><ENTITY_nouny>罕至處</ENTITY_nouny><ACTION_quantifiedVerb>想一想</ACTION_quantifiedVerb><ENTITY_pronoun>自己</ENTITY_pronoun><FUNC_inner>的</FUNC_inner><ENTITY_noun>人生</ENTITY_noun>",
  "result_obj": [
    [
      {
        "text": "這個",
        "pos": "ENTITY_DetPhrase"
      },
      {

```

進階功能

這個 **detphrase**

星期日 **time**

本 **modifier**

想 **verb**

往 **inner**

後 **locality**

山 **oov**

藥師佛寺 **nouny**

去 **verb**

世人 **nouny**

罕至處 **nouny**

想一想 **quantifiedverb**

自己 **pronoun**

的 **inner**

人生 **noun**

玉山是台灣最高的山，是清新的、是壯麗的，也是最美的。

以玉山為名，我們的決心就寫在名字上，決心要經營一家最好的銀行，成為顧客最愛的銀行。

**What you see
with your eyes.**

玉山是台灣最高的山，是清新的、是壯麗的，也是最美的。

以玉山為名，我們的決心就寫在名字上，決心要經營一家最好的銀行，成為顧客最愛的銀行。

**What you see
with your brain.**

aux_func(是) => str

conjunction_func(也) => str

degreeHead_func(最, right, 2) => str

inner_func(在) => str

aux_func(為) => str

玉山是台灣最高的山，是
清新的、是壯麗的，也是最
美的。

以玉山為名，我們的決心
就寫在名字上，決心要經
營一家最好的銀行，成為
顧客最愛的銀行。

pronoun_func(們, left, 2) => str

inner_func(的) => str

inner_func(以) => str

classifier_func(一, right, 1) => str

**Our brains have
many built-in
functions**

玉山是台灣最高的山

aux_func(是) => str

玉山是台灣最高的山

inner_func(的) => str

玉山是台灣最高的山

degreeHead_func(最, right, 2) => str

玉山是台灣最高的山

解開巢層 => Articut lv1

玉山是台灣最高的山

組成 X-bar 短語 => Articut lv2

玉山是台灣最高的山

Articut => Loki (POS => NER)

玉山是台灣最高的山

nouny AUX nouny DegreeP func nouny

locHead_func(last, 1) => str

玉山是台灣最高的山

Location AUX Location DegreeP func nouny

pai mi-gacalj ti yusiv, sa kacu-i
azua aljak katua kina, sa vaik a
pa-sa israil.

排灣語中譯：
約瑟起來，並帶著小孩子和母親
往去色列。



**What you see
with your eyes.**

注意有幾個 `_func()` 和中文是一樣的！

`focus_func(mi) => str`

`inner_func(ti) => str`

Pai mi-gacalj ti Yusiv sa kacu-i
azua aljak katua kina sa vaik a
pa-sa israil.

`focus_func(i) => str`

`conjunction_func(sa) => str`

排灣語中譯：

ti 約瑟 **mi-**起來，**並** 帶著**-i** 小孩子
和 母親 **pa-**去 以色列

這些 `_func()` 會告訴大腦「旁邊那個是什麼東西。是人名？是名詞？是動詞...」

What you see with your brain.

你不需要看過數十億次的「功能詞」，
大腦才學會什麼是「功能詞」；

你不需要看過數十億次的「實義詞」，
才能學會什麼是「實義詞」。

這些認知功能是「內建的」！只是語言的
變化比視覺更靈活，所以是以一種
metaclass 的形式內建在大腦裡。

Loki

Linguistics-**O**riented **K**eyword **I**nterface

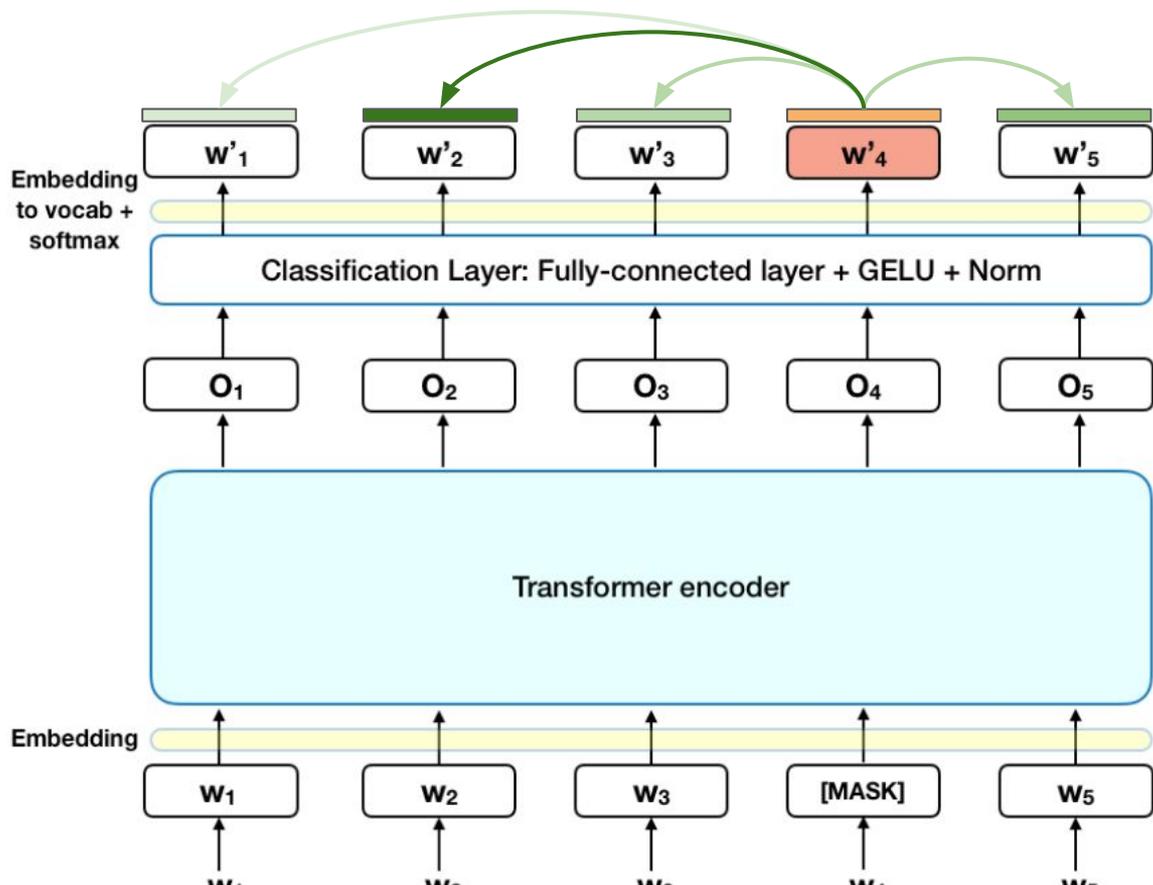
Loki 要做的事情和 BERT, GPT 的設計目標 並無二致

對於「語言」，我們要知道的是「哪一個元素」和「哪一些元素」之間彼此有相依關係。這個相依關係可以是直觀的 "She likes cats." 裡的 "-s" 和 "she" 有關，或是「他買了一條扁扁的魚」裡的「條」和「魚」有關。

也可以是較不直觀的「胖虎的妹妹說大雄暗戀自己」裡的「妹妹」指的是妹妹「自己」。

重要的問題是「這種相依關係怎麼從語言裡榨出來？」

BERT: 用上下文榨出來



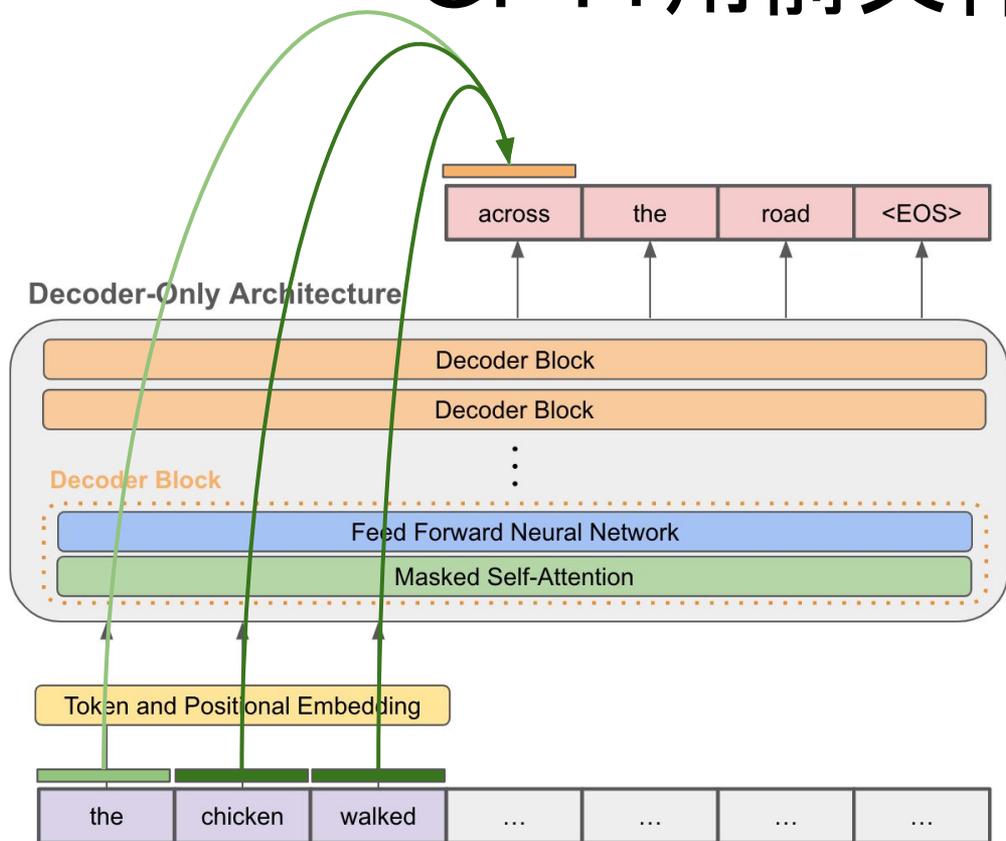
在 BERT 裡, 利用大量的上下文做 Bidirectional Encoder Representations from Transformers。

圖中的綠色 token 都是為了表示橘色 token 而被模型學習。

最重要的字是 "bidirectional"。也就是它假設「只要訓練資料夠多, 前後的 token 可以代表被 mask 的 token 的資訊」。

這個架構更適合做「搜尋」。

GPT: 用前文榨出來

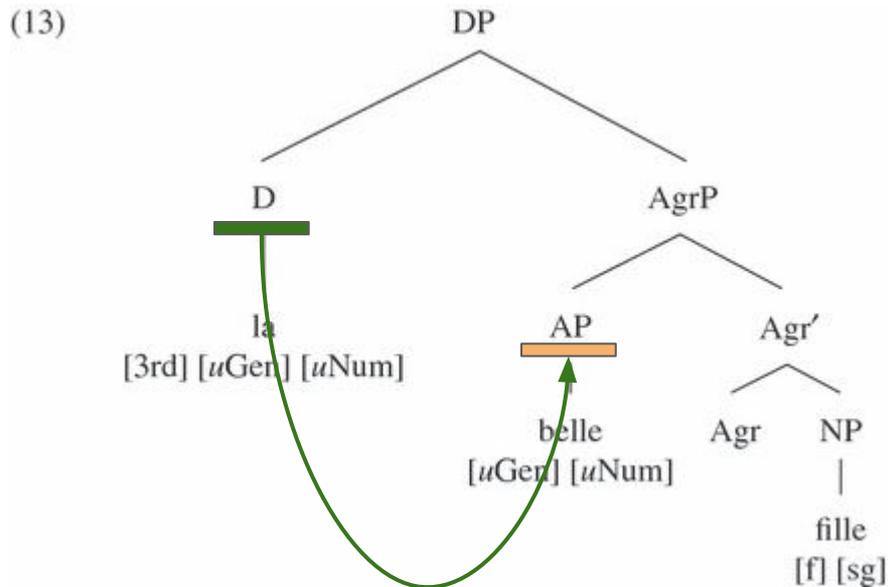


在 GPT 裡，利用大量的**前文** Generative Pre-trained Transformer。

圖中的綠色 token 都是為了表示橘色 token 而被模型學習。

如同它名字所說的，它是一個 "generative" 模型，因此更適合做「生成」。

Loki: 用語言結構榨出來



在 Loki 裡, 利用各語言通用的**句法樹** 解析各個詞彙元素間的相依關係。

依句法樹的結構, 圖中的綠色 token 節點 D 和橘色 token 節點 AP 有相依關係。(依「結構」就能知道了, 故不需要大量語料做訓練)

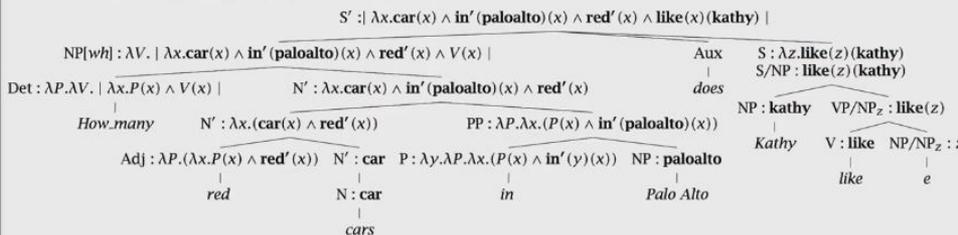
應用上, 語言學更適合做「理解」(NLU)。



DATA+AI
SUMMIT 2022
 ORGANIZED BY databricks

Most of Natural Language Understanding 1967–2017

How many red cars in Palo Alto does Kathy like?



```
select count(*) from Likes,Cars,Locations,Reds where Cars.obj =
Likes.liked AND Likes.liker = 'Kathy' AND Red.obj = Likes.liked AND
Locations.place = 'Palo Alto' AND Locations.obj = Likes.liked
```

Zenith: symbolic ML “semantic parsing”: Zettlemoyer & Collins 2005; Artzi & Zettlemoyer 2013; Liang, Jordan & Klein 2013

Loki alias masking 的功能，就是在「理解」文本意義的前提下，才能把機敏資訊挑出來，加以覆寫/還原。

這本來就不是生成模型或是 BERT 模型的強項。

Understanding 是 Syntax Tree 的強項。

利用句法樹並不是一個「新」的想法。從 1967 年就一直到現在了。差別只在 1967 年的那個是「舊的」句法樹。因為它有諸多缺點，Chomsky 教授的「現代語言學」已經改用解釋性更強的「新的」句法樹。

擷圖中，史丹佛大學的 Manning 教授正在說明 "Understanding in languages..." 的概念，配合的就是一個句法樹。

下方的 SQL 是用來類比說明「資料庫的搜尋 SQL 是要避免歧義的，否則搜尋來的資料將失去參考價值。」因此能讓 DB「正確理解」是基本要求。而「理解」正是語言學句法樹的長處。

分類任務

機器學習方法 vs. 語言學方法

可解釋性(1/2)

- 使用Saliency score技術標記對輸出分數影響力較高之Token，以下為模型判斷正確的臺灣新聞

日前新北幼兒園「餵藥案」引發關注，為強化「苯巴比妥」複方製劑及「氯二氮平」複方製劑管理，行政院9月12日公告修正「管制藥品分級及品項」，將前述製劑納為第四級管制藥品管理。食藥署公布，相關納管法規自112年12月1日生效，全面正式實施納管。食藥署副署長陳惠芳表示，苯巴比妥及氯二氮平複方製劑屬毒品危害防制條例的第四級毒品，並無排除列管；另各國及聯合國均無將苯巴比妥及氯二氮平成分複方製劑排除列管，為避免藥品遭不當使用，危害民眾健康，需將含苯巴比妥及氯二氮平等複方製劑列為管制藥品管理，才能依法追蹤流向。陳惠芳強調，自生

新北市衛生局今公布新增4例本土登革熱病例，為3女1男，年齡介於33至67歲間。分別居住於永和區、中和區、新莊區及汐止區，個案潛伏期間有台南市旅遊史。而為鼓勵市民和醫療院所主動通報，衛生局也提供市民500元禮券和臨床醫師2,500元的獎金。新北衛生局疾管科科长王美華指出，接獲疑似登革熱病例通報後，即進行疫情調查，執行戶外孳生源清除及環境噴藥，截至112年9月11日新北本土登革熱病例累計14例，12例與台南市有關，其他2例也都有南部縣市旅遊史或足跡。王美華提醒，目前南部疫情嚴峻，前往登革熱流行疫區，務必做好防蚊措施，若去流行疫

不少家長因「孩子長不高」、身高比同儕矮而焦慮不已。中醫師王大元指出，身高除了受父母遺傳基因調控外，還會受生長激素所影響，其中「糖分」是掌控其分泌量的最大因素之一，血糖上升會抑制生長激素分泌。他舉例，若孩子下課後陸續喝下2杯全糖飲料，生長激素會「徹底被壓抑」。擁有良好睡眠、多曬太陽和運動，就有機會長高。中醫師王大元在粉專《元氣中醫師王大元》指出，影響身高的生長激素，通常在白天分泌較少，但到了晚上22~2點和清晨5~7點會大量分泌，因此不要錯過這長高的關鍵時刻。王大元接著說，孩子吃甜食後血糖上升，會抑制生長激

國內近期登革熱疫情嚴峻，台南本土個案已破5000例！針對疫情發生，專家表示，南部以埃及斑蚊為主，北部以白線斑蚊為主，這些病媒蚊喜好出沒時段為上午8-10點、傍晚4-6點，一隻雌蚊每次吸血可產80-120粒蚊卵，約2-3天發育成蚊，[UNK] 9-11天就可變為成蚊，因此一定要特別留意清除環境孳生源。登革熱是經由帶登革熱病毒的「埃及斑蚊」和「白線斑蚊」叮咬而受到感染，而兩種斑蚊的習性不同，因此防治上可透過瞭解病媒蚊的特性避免蚊子叮咬。國立臺灣大學公共衛生學院蔡坤憲副院長表示，登革熱病媒蚊分布，南北不同，南部以埃及斑蚊為主，喜於室內

可解釋性(2/2)

- 使用Saliency score技術標記對輸出分數影響力較高之Token, 經觀察發現模型關注以下資訊
 - 年份(臺灣使用 **民國年** 而中國使用 **西元年**)
 - 地點與人名(出現 **臺灣的城市** 或 **人名** 通常為臺灣新聞)
 - [UNK](若模型有不認識的token, 則易判斷為中國新聞)
 - 少許用詞差異(例如: 出現「粉專」之新聞通常被判斷為臺灣新聞)

使用資源

- 模型於NVIDIA RTX A4000顯示卡上訓練，詳細資源使用量如下
 - VRAM: 約6,300MB
 - RAM: 約920MB
 - 花費時間: 完成整個訓練約 1分鐘(1個epoch約6秒)
- 使用模型進行預測時之資源使用量
 - VRAM: 約3,700MB
 - RAM: 約920MB
 - 花費時間: 完成測試資料集測試約 6秒(每筆資料約0.01秒)



LINGUISTICS

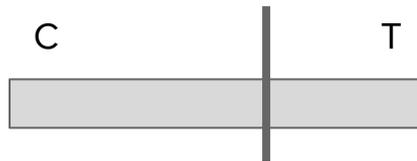
源中國文本偵測

Droidtown Linguistics Tech. Co.

卓騰語言科技

training data (unbalanced)

- 詞彙模型：
 - 台灣模型特徵詞： 7115 詞
 - 中國模型特徵詞：10044 詞
- 句型模型：
 - 台灣模型已知句型： 9315 句
 - 中國模型已知句型：15727 句



預期：

中國模型比台灣模型多 2929 個詞，也多 6412 個句型，可預期中國模型的預測力會較強。但我們將採用「因果網路模型」來強化小資料的預測能力與可解釋性。

ref. 科普 [因果網路] <https://youtu.be/PHTR4rON728?si=UjEYtga1-IVmhyS0>

結論：第一批 testing_data_1

中國文本預測：298 篇測資

Evaluation	testing_1_cn
總題數 A	298
錯誤數量 M	10
文本太短 E	4
不問解釋的正確率 Acc. = (A-M)/A	0.96644
忽略 E 的正確率 Acc. = (A-M)/(A-E)	0.97959
Precision	0.96644
Recall	0.96321
F1 Score	0.96482

台灣文本預測：190 篇測資

Evaluation	testing_1_tw
總題數 A	190
錯誤數量 M	11
應另立分類 E	5
不問解釋的正確率 Acc. = (A-M)/A	0.94211
忽略 E 的正確率 Acc. = (A-M)/(A-E)	0.96757
Precision	0.94211
Recall	0.94709
F1 Score	0.94459

結論：第二批 **testing_data_2** (測資誤含訓練資料)

中國文本預測：298 篇測資

Evaluation	testing_2_cn
總題數 A	298
錯誤數量 M	8
文本太短 E	2
不問解釋的正確率 Acc. = (A-M)/A	0.97315
忽略 E 的正確率 Acc. = (A-M)/(A-E)	0.97973
Precision	0.97315
Recall	0.92949
F1 Score	0.95082

台灣文本預測：300 篇測資

Evaluation	testing_2_tw
總題數 A	300
錯誤數量 M	22
應另立分類 E	5
不問解釋的正確率 Acc. = (A-M)/A	0.92667
忽略 E 的正確率 Acc. = (A-M)/(A-E)	0.94237
Precision	0.92667
Recall	0.97203
F1 Score	0.94881

因果網路模型不易受訓練資料影響。因其學習的是資料間的「因果關係」而不是資料分佈機率。

結論：第三批 testing_data_3

中國文本預測：298 篇測資

Evaluation	testing_3_cn
總題數 A	298
錯誤數量 M	8
文本太短 E	2
不問解釋的正確率 Acc. = (A-M)/A	0.97315
忽略 E 的正確率 Acc. = (A-M)/(A-E)	0.97973
Precision	0.97315
Recall	0.92652
F1 Score	0.94926

台灣文本預測：253 篇測資

Evaluation	testing_3_tw
總題數 A	253
錯誤數量 M	23
應另立分類 E	3
不問解釋的正確率 Acc. = (A-M)/A	0.90909
忽略 E 的正確率 Acc. = (A-M)/(A-E)	0.92000
Precision	0.90909
Recall	0.96639
F1 Score	0.93686

和第二批相比，錯誤的題目都被留下來，而移除了答對的題目，故分數下降。

原理：

中國的普通話和台灣的國語，是兩個「詞彙集合幾乎一樣，但句型表現不太一樣」的語言。兩者間的關係，可理解為 **{word, set, of, China}** 和 **{word, set, of, Taiwan}** 兩個集合之間，大部份的元素都一樣，只有少數不同。

除元素的稍稍不同以外，兩種語言另一個面向在於其「詞彙排列(即句型)」的不同。

可理解為：

[China, set, of, word] 和 **[word, set, of, Taiwan]** 的形式。兩個有序陣列之間，雖然元素相同，但習慣用的詞彙順序(即句型)卻不同。

需注意的是，詞彙易學，而句型難學。若為中國人刻意要模仿台灣的國語來書寫，即便是學到了某些詞彙，也會在句型上露出馬腳。

ps. 人腦具有強大的校正能力，因此視覺上的詞彙順序會被大腦中的語言區予以校正。

二分類任務：有 **C**, **T** 兩種標籤類別

二分類任務，依邏輯推估，結論矩陣中應有四種可能結果：



實際資料示意：詞彙

台灣：{

verb:[怒吃, 傳訊, 幫補, 請願, 稽核...],

noun:[議員, 原型食物, 優格...],

location:[雪梨, ...]

}

中國：{

verb:[糾正, 致脹, 焯水, 致龔, 撬動...],

noun:[精神需求, 心靈體魄, 貧困縣...],

location:[悉尼, ...]

}

詞彙模型需建兩套：CN 和 TW 各一套。兩個向量空間模型無法直接比較，而應以「輸入文本抽出的詞彙的向量」在兩個向量空間中的集中程度為依據。

愈集中，表示「這個語言的各詞性(各方面)詞彙，於文本的各個詞性/各個方面都是相近的」，則屬於該模型所代表的語言。

實際資料示意：一樣是「有」， CN/TW 的句型不同

- TW: 有 + Verb

有	338	有	/	<input checked="" type="checkbox"/>	小生命	/	來不及	/	<input checked="" type="checkbox"/>	張開	/	<input checked="" type="checkbox"/>	眼睛	/	<input checked="" type="checkbox"/>	認識	/	<input checked="" type="checkbox"/>	世界													
有	339	有	/	<input checked="" type="checkbox"/>	新事證	/	或	/	後續	/	有	/	<input checked="" type="checkbox"/>	需要	/	<input checked="" type="checkbox"/>	釐清	/	的	/	<input checked="" type="checkbox"/>	問題										
有	340	有	/	<input type="checkbox"/>	標示	/	不實	/	或	/	<input checked="" type="checkbox"/>	違反	/	<input type="checkbox"/>	食安	/	<input type="checkbox"/>	法	/	相關	/	<input type="checkbox"/>	規定									
有	341	有	/	<input checked="" type="checkbox"/>	超過	/	一半	/	的	/	<input checked="" type="checkbox"/>	颱風	/	<input checked="" type="checkbox"/>	強度	/	是	/	<input checked="" type="checkbox"/>	強烈	/	<input checked="" type="checkbox"/>	颱風									
有	342	有	/	<input type="checkbox"/>	那麼多	/	處	/	<input checked="" type="checkbox"/>	整傷	/	但	/	有	/	<input checked="" type="checkbox"/>	出現	/	<input checked="" type="checkbox"/>	過敏	/	<input type="checkbox"/>	症狀									
有	343	有	/	<input checked="" type="checkbox"/>	提供	/	<input checked="" type="checkbox"/>	年齡	/	<input checked="" type="checkbox"/>	45	/	<input checked="" type="checkbox"/>	歲	/	以上	/	至	/	未	/	<input type="checkbox"/>	滿	/	<input checked="" type="checkbox"/>	70	/	<input checked="" type="checkbox"/>	歲	/	<input type="checkbox"/>	婦女

- CN: 有 + N

有	333	有	/	<input checked="" type="checkbox"/>	視力	/	<input checked="" type="checkbox"/>	障礙	/	的	/	<input checked="" type="checkbox"/>	學生	/	的	/	<input checked="" type="checkbox"/>	學習	/	<input checked="" type="checkbox"/>	需求					
有	334	有	/	2小時	/	及	/	以上	/	的	/	<input checked="" type="checkbox"/>	戶	/	外	/	<input checked="" type="checkbox"/>	活動	/	<input checked="" type="checkbox"/>	時間					
有	335	有	/	<input checked="" type="checkbox"/>	氧	/	<input checked="" type="checkbox"/>	鍛鍊	/	後	/	的	/	<input checked="" type="checkbox"/>	血壓	/	<input checked="" type="checkbox"/>	降低	/	<input type="checkbox"/>	量	/	為	/	<input checked="" type="checkbox"/>	4.49
有	336	有	/	<input checked="" type="checkbox"/>	1.3萬	/	至	/	1.5萬	/	公頃	/	的	/	<input checked="" type="checkbox"/>	香蕉	/	<input checked="" type="checkbox"/>	種植地							
有	337	有	/	<input checked="" type="checkbox"/>	120萬	/	<input checked="" type="checkbox"/>	死於	/	<input checked="" type="checkbox"/>	阿	/	<input checked="" type="checkbox"/>	片類	/	<input checked="" type="checkbox"/>	藥物	/	<input checked="" type="checkbox"/>	使用	/	過量				
有	338	有	/	一點點	/	<input checked="" type="checkbox"/>	濃液	/	就	/	會	/	<input checked="" type="checkbox"/>	感覺	/	非常	/	的	/	<input checked="" type="checkbox"/>	脹					

中國模型比台灣模型多 6412 個句型，可預期中國模型的覆蓋率會較高。

在語料不平衡的實際考量下，不能「只憑句型」決定分類。

模型的可解釋性：模型究竟在看什麼？為什麼它看的東西有效果？

它看詞彙，因為...

1. 詞彙在 C/T 向量空間中愈集中，積分愈高，與假設一致時，表示在「動詞、名詞、地名...」等各方面，都使用了該語言的詞彙。故可推論『文本屬於該語言！』
2. 詞彙在 C/T 向量空間中愈離散，相似度愈低。與假設不一致時，可能表示該文本較短，沒有幾個詞可以計算，或是發生了「詞彙受另一個語言影響」的情況。此時再以句型做為補充推論。

它看句型，因為...

句型下，可以造出愈長的句子，表示對這個語言愈熟練，也表示有更多的詞彙論元。故句型模型可凸顯出對該語言的熟練程度，故可推論『文本屬於該語言！』

Team Assignment: (Due: 10/15 on Discord)

請觀賞：

<https://www.youtube.com/watch?v=PHTR4rON728>

並從生活中找出「兩個例子」，證明「人類思考的重點是因果關係，而非統計機率的共現關係」意即，人類天生就會在「前後兩個連續發生的事情中，不由自主地用因果關係來思考！」

你要找的兩個例子將會具有「前後兩件事情沒有因果關係，但只要擺在一起，就會產生『明明理智上你知道沒有因果關係，但是你就是有一股難以否認的『因為...所以...』的直覺想要套用上」的特點。

將你們組內討論過的例子貼在 Discord 裡，並說明「它為什麼沒有因果關係？為什麼會讓你覺得有因果關係？」

e.g.,

阿明喜歡流浪動物，從小就會撿狗回家。他對特殊食材的追求也絕不妥協！