

基於語言學原理的語意運算 - Structural Pattern Matching

卓騰語言科技

PeterWolf

(peter.w@droidtown.co)

Been there, done that...

過去社群演講經驗：

PyConTW 2013 講者

PyConTW 2019 講者

COSCUP 2020 講者 (三場)

維基台灣社群年會 2020 講者

PyDay 2021 講者

PyHUG 講者 (三不五時)

Tainan.py 講者 (一次)

Taipei.py 講者 (一次)

社群經驗：

PyHUG、gov、WikiData

PyConTW 2012、2013 贊助組組長

創業：

卓騰語言科技 創辦人 (核心開發工程師)

校園演講：

2013: 臺灣大學語言學研究所演講

2014: 中正大學語言學研究所演講

2019: 東海大學哲學星期五社團演講

2020: 教育部推動大學程式設計教學計畫演講

2021: 臺灣大學外文系演講

2021: 宜蘭大學資工系演講

2021: 馬偕醫學院聽語障礙科學系演講

2021: 人工智慧科技基金會演講

2021: 暨南國際大學外文系演講

現在在這裡

授課:(Youtube 請查 "**Droidtown**")

2015: 臺灣科技大學: 應外系 語音學與應用 業師

2016: 臺灣科技大學: 應外系 語音學與應用 業師

2020: 臺灣師範大學: 文本分析與程式設計 講師

2021: 臺灣師範大學: 文本分析與程式設計 講師

AI (NLP) 發展迷思



1950 ~ 1960

第一波

Fail

符號邏輯
無上限的規則

1980 ~ 1990

第二波

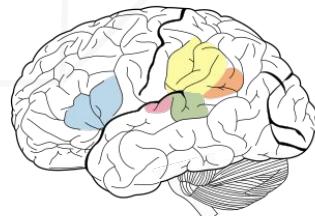
Fail

2010 ~ now

第三波

專家系統

<https://futurecity.cw.com.tw/article/743>



1957 ~ now

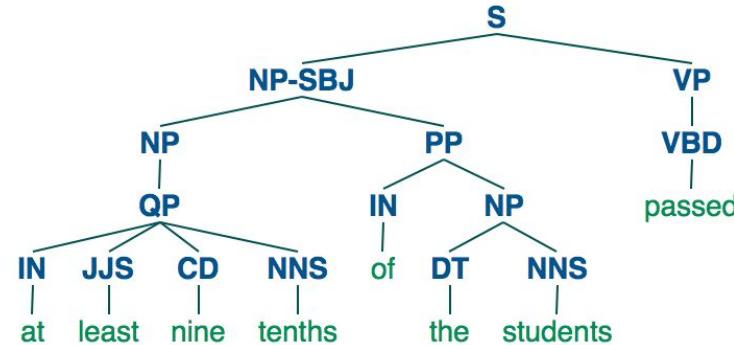
現代語言學

有限的規則



舊典範：句法樹 (syntax trees/phrase structure rules)

- 舊句法樹(複數)
 - $VP \rightarrow V + PP$
 - $VP \rightarrow V + NP + PP$
 - $PP \rightarrow P + PP$
 - $PP \rightarrow P + NP$
 - $PP \rightarrow P + V$
 - $NP \rightarrow Adj + N$
 - $NP \rightarrow Det + N$
 - $NP \rightarrow N + N$
 - $QP \rightarrow P + Adj + N + N$
 - $QP \rightarrow NP$
 - ...
- 用 NLTK 畫出舊句法樹

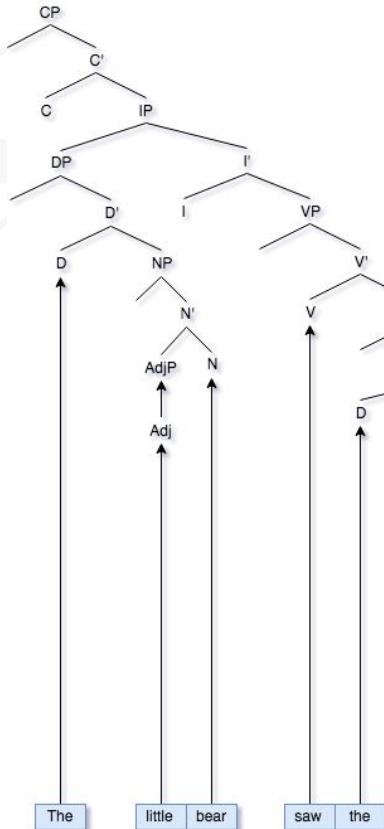
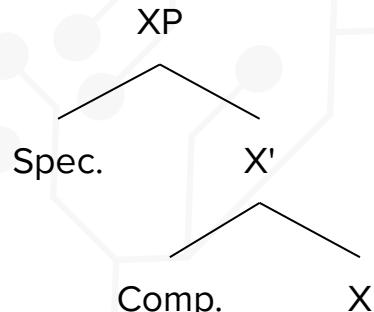


結構有這麼多變化，不同語言又不太一樣，幾乎不可能寫出可用的 parser

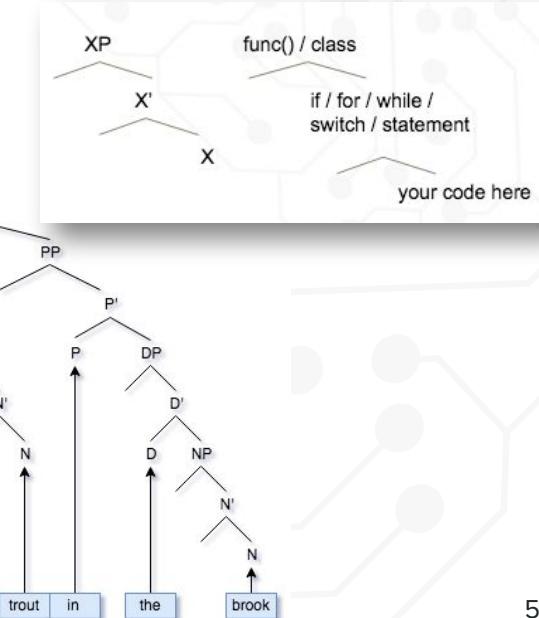
新典範：句法樹 (syntax tree/phrase structure rule)

- 新句法樹(單數)

- $XP \rightarrow Spec + X'$
- $X' \rightarrow Comp + X$



變化有限，層級一致，且全人類語言通用。據此寫出 parser 成為可能！



Home

PUBLIC

Stack Overflow

Tags

Users

FIND A JOB

Jobs

Companies

TEAMS

What's this?

 Create a Team

how can I get the binary parsed tree from coreNLP parser?

Asked 2 years, 1 month ago Active 1 year, 3 months ago Viewed 260 times

[Ask Question](#)

I need the binary parse tree of an sentence to do my experiment. But after I used Stanford Parser and CoreNLP parser, I got non-binary tree. I have tried to add property "parse.binaryTrees": "true", but it didn't work. I also have tried to startup a server in commandline like "-binarize", it also failed!! So how can I get a binary tree from parser??

```
java -Xmx4g -cp "*" edu.stanford.nlp.pipeline.StanfordCoreNLP$Server  
-serverProperties StanfordCoreNLP.properties -port 9000 -timeout 15000
```

```
nlp = StanfordCoreNLP(r'/home/lsl/stanford-corenlp-full-2018-10-05')  
output = nlp.annotate(sentence, properties={'annotators': 'parse',  
                                         'parse.binaryTrees': 'true',  
                                         'outputFormat': 'json'})
```

I want to use python to solve this problem. Thank you all!

[binary-tree](#) [stanford-nlp](#)[Share](#) [Improve this question](#) [Follow](#)

asked Jan 8 '19 at 12:27

 胜兰 廖胜兰

11 ● 2

[Add a comment](#)

1 Answer

[Active](#) [Oldest](#) [Votes](#)

def binarize(tree):
 """
 Recursively turn a tree into a binary tree.
 """
 if isinstance(tree, str):
 return tree
 elif len(tree) == 1:
 return binarize(tree[0])

0



The Overflow Blog

 What I wish I had known about single page applications

Featured on Meta

 Visual design changes to the review queues

 Survey questions for outdated answers

 Introducing Outdated Answers project

Related

0 Finding adverbs and what they modify using Stanford Parser

2 Activate makeCopulaHead in Stanford CoreNLP parser

0 Stanford CoreNLP - dashes

6 Why Stanford parser with nltk is not correctly parsing a sentence?

1 Dependencies are null with the German Parser from Stanford CoreNLP

1 StanfordCoreNLP differs from StanfordCoreNLP\$Server

0 Tree structure from Stanford CoreNLP parser

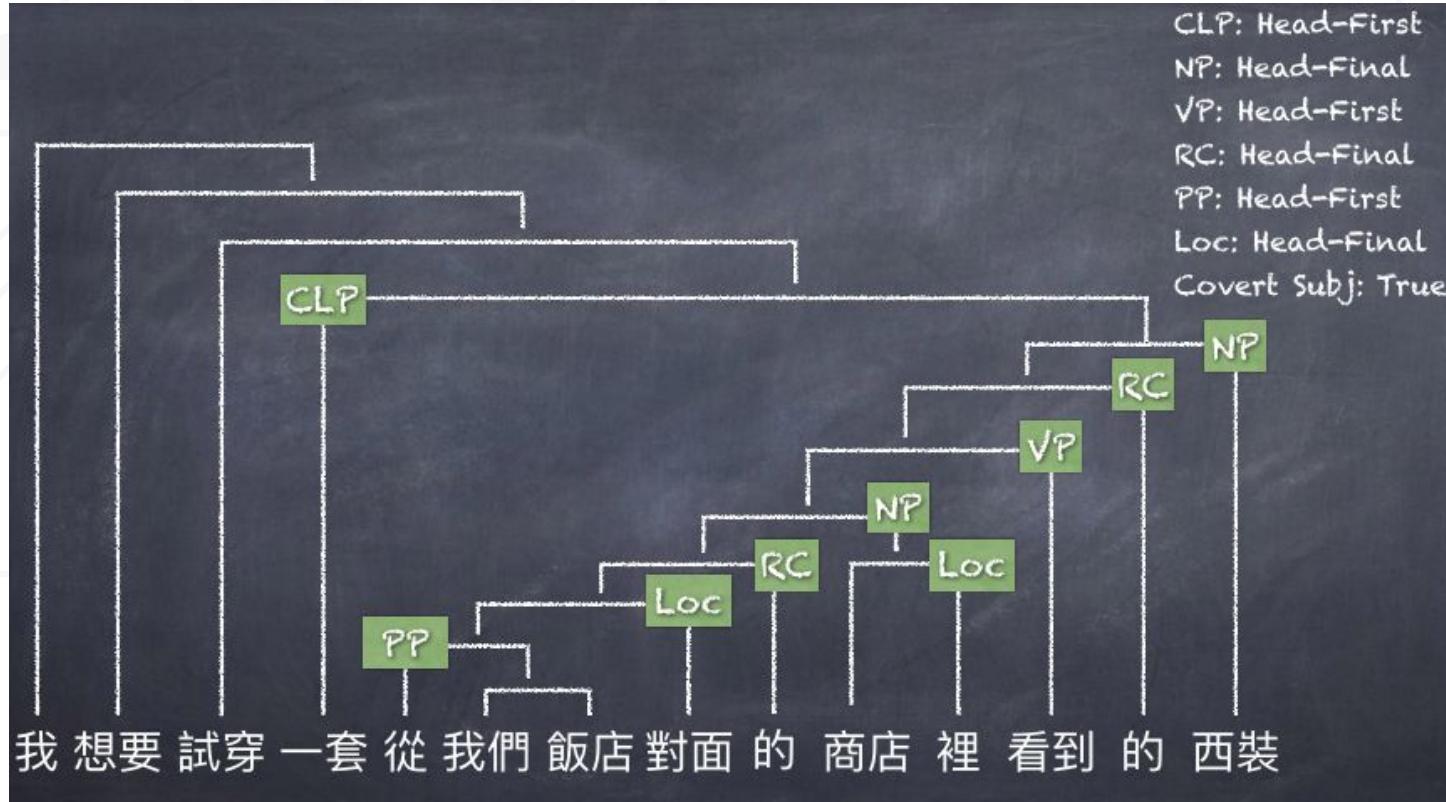
0 Dependency parse large text file with python

1 Customize NER in StanfordNLP Server

Hot Network Questions

新典範的跨語言通用性

<https://youtu.be/6Jq-4Pu3er8>



語言行為 (Linguistic Behavior)

Encyclopedia
(Common Sense +
Domain Knowledge)

每天人類稱之為「語言」的東西，包含了語言系統的運作和對所處世界百科知識以及領域知識裡的補充。

兩者結合的輸出即為「語言」！

語言

Linguistic System

Symbols are not Math, so Texts are not Languages

- text model != language model
- 透過文字系統產生的是文字模型，它呈現的是「文字符號的分佈模型」，缺乏語言視野。更不該被稱為語言模型。

Holy shit that sucks man.



那 NLU 的機器學習究竟在學什麼？

高維度的語音



低維度的符號

紅框裡有幾張人臉，我就問！

當「高維度」的意義出現時，人類的認知系統很難發現「低維度」的 ASCII 符號的存在。這是人類認知系統的運作方式。

機器學習需要從低維度的符號開始做資料擬合，呈現的是「符號分佈」的模型，更難以呈現人類認知中的「語意」的部份。

因此 NLU (自然語言理解) 很難透過 ML/DL 來做！

ML/DL 的方法容易看到「產出結果」的是 NLP 和 NLG，因為這兩者都可以透過操弄符號來完成。

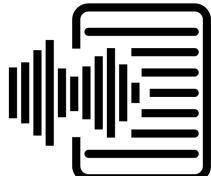
機器學習 (Machine Learning) 在 NLP 領域裡是一個「逐步失真」的過程

Language



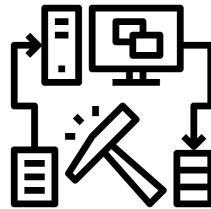
Created by Vicons Design
from Noun Project

Text



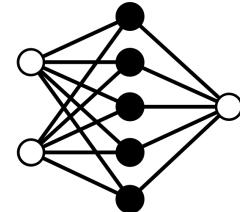
Created by Trevor Dsouza
from Noun Project

Preprocessing



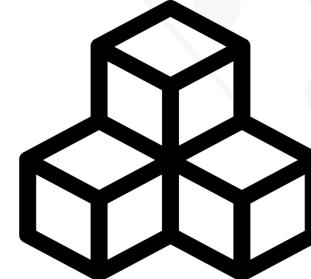
Created by Becriis
from Noun Project

Machine Learning



Created by Product Pencil
from Noun Project

Language Model



Created by Serhii Smirnov
from Noun Project

ASR 裂失：
語氣、語調、語速

斷句裂失：
前後文、語境

NN/Vec 裂失：
文法、句構

LM 裂失：
邏輯、因果、知識

語言學的典範轉移

典範轉移 (Paradigm shift)



典範轉移，又稱範式轉移或思角轉向，這個名詞最早出現於美國科學史及科學哲學家湯瑪斯·孔恩的代表作之一《科學革命的結構》。這個名詞用來描述在科學範疇裡，一種在基本理論上從根本假設的改變。這種改變，後來亦應用於各種其他學科方面的巨大轉變。

維基百科

<https://zh.wikipedia.org/wiki/認知革命>

https://en.wikipedia.org/wiki/Steven_Pinker

Before Paradigm Shift:

- 人類出生時大腦是一片空白
- 語言能力需藉由大量刺激獲得

After Paradigm Shift:

- 人類出生時大腦已有最適化演算結構
- 語言能力可藉由少量刺激啟動

Structure Matters

在人類的認知系統裡，有兩個系統在運作
表層形式：看得到的符號
深層形式：看不見的結構

<https://github.com/Droidtown/LiveDemo/tree/main/PyDay2021>

人類認知系統 00: 從 Structure 到 Pattern

這是一條線

在文本的 context 下

blah blah blah...

[畫底線] 的意圖

在數學的 context 下

α

β

[做除法] 的意圖

人類認知系統 01: 從 Structure 到 Pattern

$$\cos \square + \cos \square = 2 \cos \frac{\square + \square}{2} \cos \frac{\square - \square}{2}$$

$$\alpha \square \beta = 2 \square \frac{\alpha \square \beta}{2} \frac{\alpha \square \beta}{2}$$

人類認知系統 02：從 Structure 到 Pattern

我買了一隻會 _____ 的小狗

ML 都用這些做 training data

{趕羊、工作、睡覺、唱歌、算數學...}

但「語言」是靠 structure 運作

type({...}) = Verb

透過結構標記解析句法 (還有音韻規則) <https://api.croidtown.co>

設定 : lv2 詞組斷詞景點資料庫 : WikiData : 化學 : 辨識自定義詞典


文
獻

Articut\$ 免費字數 : 1699在一線工作時，穿著工作服又頭戴安全帽，偶爾會被叫李大哥，說話後才知道我是女生。」在華航修護組織租機管理部擔任工程師的李瑩珠，原本念商科，從零開始，錄取後歷經一年受訓，並考取維修執照後才真正上線，必須面對航機臨時甚至特殊情況，緊急執行檢修及故障排除。

機讀結果：「/在...(more)

進階功能

```
在 inner — num 線 oov 工作 noun 時 inner 穿著 verbp 工作服 nouny 又 inner 頭 nouny 戴 verb 安全帽 nouny  
偶爾 time 會 modal 被 lightverb 叫 verb 李大哥 pronoun 說話 verb 後 period 才 modal 知道 verb 我 pronoun 是 aux  
女生 noun 在 inner 華航 nouny 修護 verb 組織 verb 粗機 nouny 管理部 nouny 擔任 verb 工程師 noun 的 inner  
李瑩珠 person 原本 modifier 念 verb 商科 nouny 從 inner 零 num 開始 verb 錄取 verb 後 period 歷經 verb  
一年 time 受訓 verbp 並 inner 考取 verb 維修 verb 執照 nouny 後 locality 才 modal 真正 modifier 上線 verb  
必須 modal 面對 verb 航機 nouny 臨時 time 甚至 modifier 特殊 modifier 情況 nounhead 緊急 modifier 執行 verb  
檢修 verb 及 conjunction 故障 verb 排除 verb
```

在一線工作時，穿著工作服又頭戴安全帽，偶爾會被叫李大哥，說話後才知道我是女生。

若您有任何操作上的問題，或是需要討論斷詞結果、詞性標記的正確與否，歡迎到 [Articut 的 Facebook 粉絲專頁](#) 發文討論。

語意可由「結構」+「樣式比對」取得「論元」來計算嗎？

稍早的例子裡，我們知道「數學公式」的語意(這個式子想表達什麼)可以由

(記在腦子裡的)「結構」+ (眼睛看到的)「樣式比對」來取得「論元」(即 α 、 β)

稍早的例子裡，我們知道「自然語言」的語意(這個句子想表達什麼)可以由

(天生在腦子裡的)「結構」+ (眼睛看到的)「樣式比對」來取得「論元」

**但，「結構」+「樣式比對」並「取得論元」就是語意嗎？
怎麼證明這個想法？**

程式語言中的「語意」是什麼？

程式語言是「程式設計師」對「機器運算模組」溝通時使用的語言系統。它的目的在於把「程式設計師」的意圖傳達給「機器運算模組」，讓它明白你的程式裡「每一段落」的唯一意義，不能有歧義。

自然語言中的「語意」是什麼？

自然語言是「人類」對「另一個人類」溝通時使用的語言系統。它的目的在於把「人類」的意圖傳達給「另一個人類」，讓對方明白你說的話裡「每一段落」的唯一意義，不能有歧義。(如果有歧義，就要靠語境或是腦補來消歧義)

現代程式語言之父...剛好也是現代語言學大師



How has Noam Chomsky's work influenced the field of programming language theory?

<https://www.slideshare.net/YiShinChen1> p.19

跳回第 5 頁...

- Also see

[https://en.wikipedia.org/wiki/Semantics_\(computer_science\)](https://en.wikipedia.org/wiki/Semantics_(computer_science))

<https://www.cl.cam.ac.uk/teaching/0809/Semantics/notes-mono.pdf> p.60

<https://www.cl.cam.ac.uk/teaching/1011/L107/semantics.pdf> p.18

<https://www.python.org/dev/peps/pep-0636/#matching-multiple-values>

程式語言的語意可由「結構」+「樣式比對」取得「論元」來計算

The screenshot shows the Python.org homepage with a blue header bar containing the Python logo, a search bar, and navigation links for About, Downloads, Documentation, Community, Success Stories, News, and Events. Below the header, there's a section for tweets from @ThePSF, a link to the Python Developers Survey 2020, and a large feature box for PEP 636.

PEP 636 -- Structural Pattern Matching: Tutorial

PEP:	636
Title:	Structural Pattern Matching: Tutorial
Author:	Daniel F Moisset <dfmoisset at gmail.com>
Sponsor:	Guido van Rossum <guido at python.org>
BDFL-Delegate:	
Discussions-To:	Python-Dev <python-dev at python.org>
Status:	Final
Type:	Informational
Created:	12-Sep-2020
Python-Version:	3.10



Guido van Rossum @gvanrossum

Pattern Matching (PEP 634-636) was merged into cpython master today! Will be in 3.10 alpha 6, to be released Monday.

[翻譯推文](#)

上午7:37 · 2021年2月27日 · Twitter Web App

PEP 622: Structural Pattern Matching Basics

```
match TARGET:  
    case <pattern_1>:  
        do_this  
    case <pattern_2>:  
        do_thiat  
    case _:  
        do_something_else
```

先比對「型別」: instanceof()

型別對了，再比對屬性數量

屬性數量對了，再逐一比對每個屬性的型別 ... (列舉直到結束)

PEP 622: Structural Pattern Matching Example

```
match api_response:  
    case {"text": str(message)}:  
        ui.set_text(message)  
    case 500:  
        ui.alert("Oops, something's wrong.")  
    case ("url", "ogg"):  
        ui.play(api_response[0], api_response[1])  
    case None:  
        ui.alert("Unsupported audio format")
```

先比對「型別」: instanceof()

型別對了，再比對屬性數量

屬性數量對了，再逐一比對每個屬性的型別 ... (列舉直到結束)

Structural Pattern Matching 取得了什麼資訊？

取得了型別資訊 : `isinstance()`

知道型別，就能知道這個**型別的 [attribute]** 和 [method]

取得了 [attribute] 的**總數量**

取得了每個 [attribute] 的型別

知道每個 [attribute] 的型別，又能知道這個型別的 [attribute] 和 [method]

那...自然語言的**型別**怎麼做？

自然語言的「型別」是由動詞決定的

1. 基於歐美語系的 NLU : (LUIS/DialogFlow)

- a. 撷取「實體」！
- b. 遇到中文時沒有「斷詞」處理
- c. 其實是 BOW + CharString 機率

I bought this skirt at your store yesterday,

my wife wants me to return it.

1. 專為中文設計的 NLU: (Loki)

- a. 定锚在「動詞」！
- b. 建立「實體」和「動詞」間的樣式
- c. 能處理中文「詞彙」

我昨天買了這件裙子，我太太要我來退貨。
這件裙子我昨天買的，我太太要我來退貨。
昨天我買了這件裙子，我太太要我來退貨。
昨天這件裙子我買的，我太太要我來退貨。
昨天我買的這件裙子，我太太要我來退貨。

...

1. 英文很容易找到「動詞」

2. 英文句型裡的詞彙順序是穩定的

3. 因為結構穩定，所以只要抓實體詞彙就好。結構本身就提供了良好的模型訓練基礎。

1. 中文不容易找到「動詞」，故需要 POS 資訊 (而 LUIS 沒有, DialogFlow 也沒有。)

2. 中文的詞彙順序是相對靈活的，但動詞是穩定的一個錨點 (anchoring point)。

3. 因為結構靈活，因此要抓「動詞」和「實體」

功能語言學提出的語意解析架構

<https://blog.droidtown.co/post/627953496713560064/languageengineering02>

場景

校長室

嘉勉

斥責

功能

語句

語句

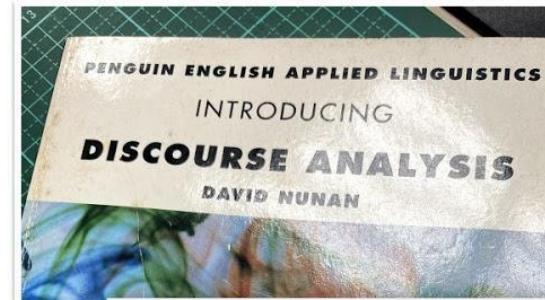
功能

語句

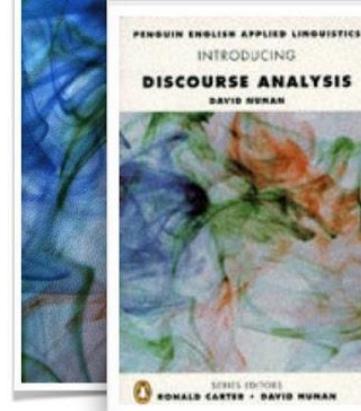
語句

語句

語句



Categories: Semantics | Grammar, Syntax



Introducing Discourse Analysis

★★★★★ 4.16 (77 ratings by Goodreads)

Paperback

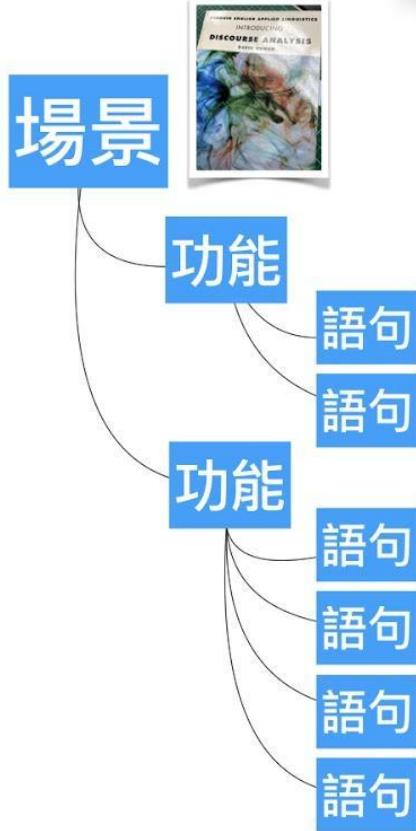
By (author) David Nunan

Share



Examines and explains discourse, visual examples from a wide range of spoken and written sources. The book also includes a number of exercises and projects to help the reader study discourse and discourse analysis in relation to their own teaching.

Loki 的操作架構



專案名稱

(或App主要功能)



意圖

語句

語句

語句

語句

語句

語句

intent 對應 function

所以一個 intent 的背後大概也就只對應「一、兩組」函式 (function) 是最適當的

Loki 以「句型」做分類，因此每一種「句型」只要出現一次就夠了。

使用 Loki LIVE Demo 與解說

The screenshot shows the Loki LIVE Demo interface. At the top, there's a navigation bar with the logo '卓騰語言科技 AI > NLP', links for 'GraphQL', '應用範例', '購買服務', '說明文件', 'FAQ', 'English', and a user profile icon. Below the navigation bar is a banner with the text '自然對話意圖分析模型產生器' (Natural Dialogue Intent Analysis Model Generator) and the 'Loki' logo.

The main area displays a table with two rows of project data:

專案名稱	專案金鑰	意圖	專案範本
FinChatbot	GcNwqO=+pP#xGQaWY3+BznLC17*agff	1	✓ --選擇-- Python Java --選擇--
PSMath01	^=z9gCBl4ZVGx=wqSGspaXi&gqg6Ma+	3	✓ 複製 --選擇--

A context menu is open over the third project row ('PSMath01'), specifically over the '專案範本' column. The menu items are: '✓ --選擇--', 'Python' (which is highlighted in blue), 'Java', and '--選擇--'. There are also small red trash can icons next to each menu item.

更多開源的 [意圖模型]: <https://github.com/Droidtown/LokiHub>

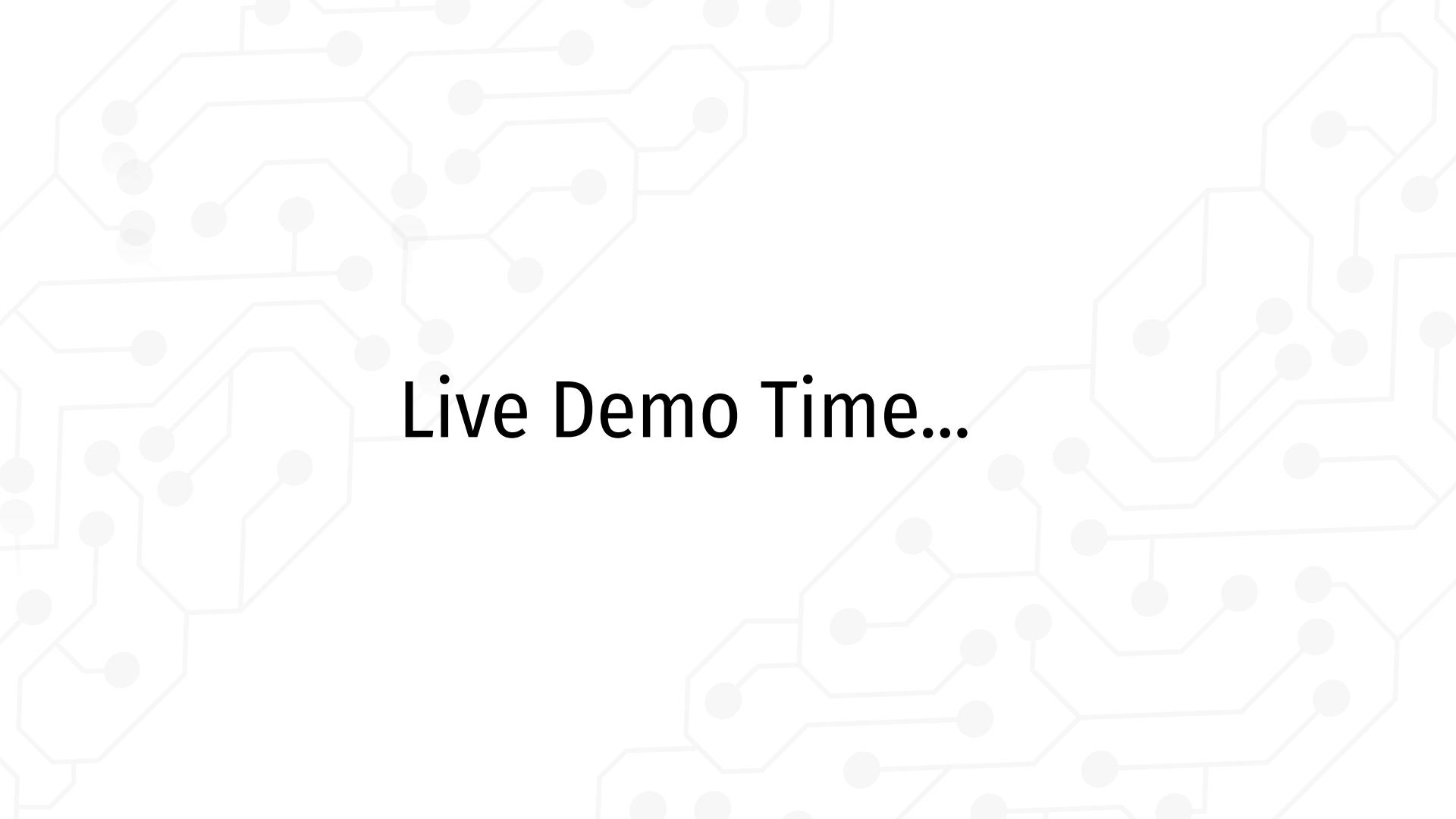
實例：

- Loki 算數學

在跳出 ML 的坑以前...先來看一下不是用 Structural Pattern Matching 的 NLU 是什麼樣的？

The screenshot shows a user interface for a machine learning-based Natural Language Understanding (NLU) system. On the left, there's a sidebar with buttons for "all entities" and "Move to". Below that is a section titled "Example user inputs" with a text input field containing "Type an example of what". At the bottom of the sidebar, there are two buttons: "我要付款" (I want to pay) and "我要付費" (I want to pay). The main area displays five examples of user utterances:

- 我要賴帳**: Predicted entity is "None (0.753)" with an "Inspect" button.
- 我要結帳**: Predicted entity is "None (0.753)" with an "Inspect" button.
- 我要搶錢**: Predicted entity is "None (0.753)" with an "Inspect" button.
- 我要那款**: Predicted entity is "Billing (0.722)" with an "Inspect" button.
- 我要這款**: Predicted entity is "None (0.415)" with an "Inspect" button.



Live Demo Time...

模糊的 NLU：以 LUIS 改善 (給更多句子)

我要付款
我會付款
我家人會付款
我爸爸會付款
我媽媽會付款
他媽媽要付款
他爸爸要付款
這位先生會付款
這位小姐會付款
這位女士會付款
這位太太會付款
那位先生會付款
那位女生會付款
那個男生會付款
那個女生會付款



我會來付款
我家人會來付款
我爸爸會來付款
我媽媽會來付款
媽要來付款
爸要來付款
生會來付款
日會來付款
會來付款
會來付款
來付款
位女生會來付款
那個男生會來付款
那個女生會來付款
他媽媽會來付款
他爸爸會來付款

我要買單
我會來買單
我家人會來買單
我爸爸會來買單
我媽媽會來買單
他媽媽要來買單
他爸爸要來買單
這位先生會來買單
這位小姐會來買單
這位女士會來買單
這位太太會來買單
那位先生會來買單
那位女生會來買單
那個男生會來買單
那個女生會來買單
他媽媽會來買單
他爸爸會來買單

語意計算：收斂發散的歧義

指令式的語意分類：

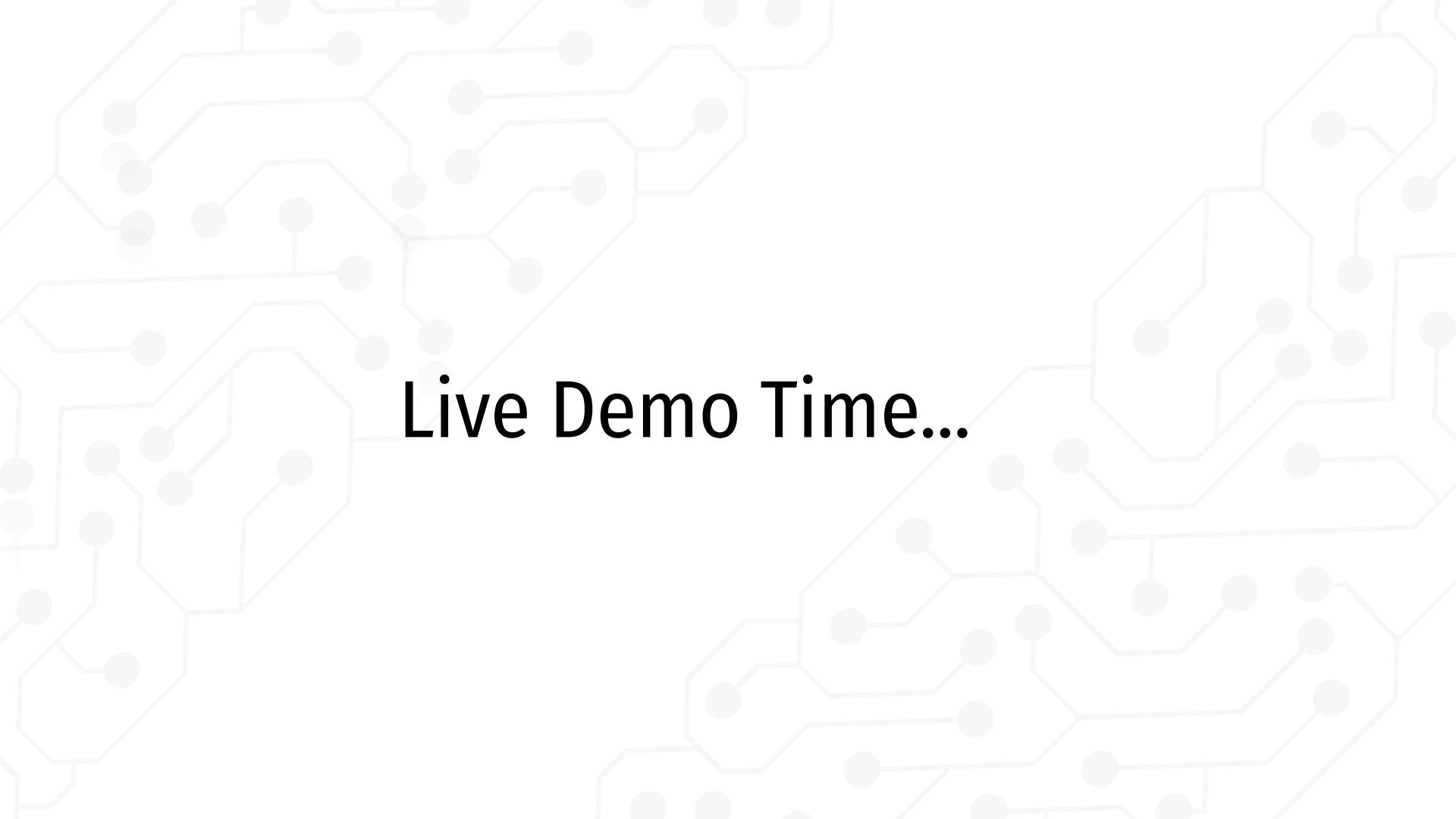
- 用**關鍵字**分類
- KeyWord: [美, 幣, 金, 元, ...]
- 所有詞彙的都要「正向 / 負向列表」

無法收斂！

Loki:

- 用**結構**分類
- 正向結構: _entity_ 夠 _modifier_
- 負向結構: _entity_ 不 / 夠 _modifier_

快速收斂！



Live Demo Time...

開始使用 Loki

利用 Loki 實做換匯功能

<https://github.com/Droidtown/ArticutAPI>

<https://api.droidtown.co/document/#lokiUserGuide>

美金/100元/可以/兌換/台幣/多少

泰銖/十萬元/可以/換/加幣/多少

開始使用 Loki

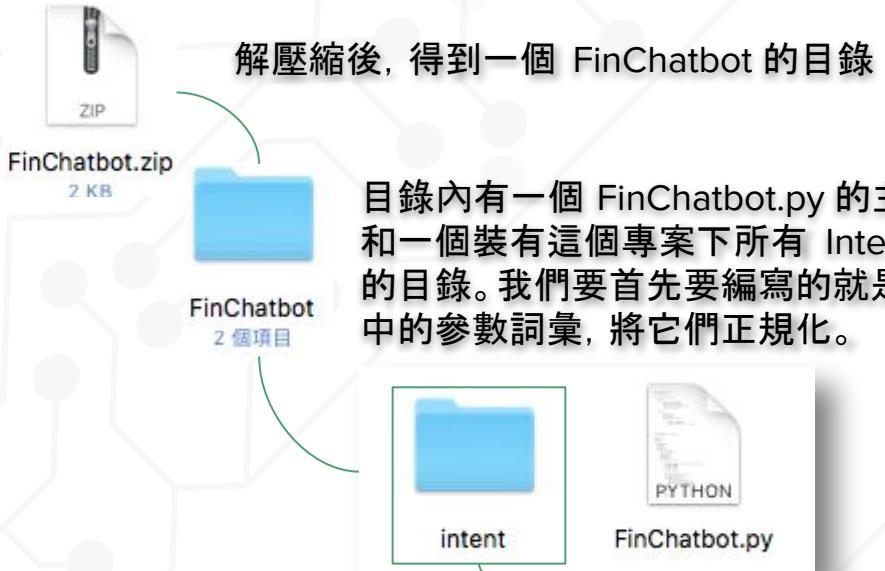
下載可配合模型 (在雲端) 操作的程式原始碼到本機

The screenshot shows the Loki NLP Model Generator interface. At the top left is the logo '卓騰語言科技 AI > NLP'. The top right features links for GraphQL, Application Examples, Purchase Services, Documentation, FAQ, English version, and user profile. Below the header is a banner with the text '自然對話意圖分析模型產生器' (Natural Dialogue Intention Analysis Model Generator). On the left, there's a 'Home' button with a red box around it. To the right of the banner are 'Documentation' and a search bar. The main area displays two projects: 'FinChatbot' and 'PSMath01'. Each project has a 'Code Sample' section with a copy button, an 'Intention' column with a value of 1, and a 'Template' column with a dropdown menu. The dropdown for 'FinChatbot' is open, showing options: 'Python' (selected), 'Java', and '--選擇--'. The dropdown for 'PSMath01' is also open, showing 'Python' (selected), 'Java', and '--選擇--'.

專案名稱	專案金鑰	意圖	專案範本
FinChatbot	GcNwqO=+pP#xGQaWY3+BznLC17*agff	1	✓ --選擇-- Python Java --選擇--
PSMath01	^=z9gCBl4ZVGx=wqSGspaXi&gqg6Ma+	3	✓ --選擇-- Python Java --選擇--

開始使用 Loki

利用 Loki 實做換匯功能



解壓縮後，得到一個 FinChatbot 的目錄

目錄內有一個 `FinChatbot.py` 的主程式，
和一個裝有這個專案下所有 Intent (意圖)
的目錄。我們要首先要編寫的就是 intent
中的參數詞彙，將它們正規化。

```
File: Loki_Exchange_luis.py
Users > peter > Downloads > FinChatbot > intent > Loki_Exchange_luis.py > getResult
15 def debugInfo(intent, args):
16     if DEBUG_Exchange_luis:
17         print(intent, "====>", args)
18
19 def getResult(pattern, args, resultDICT):
20     # [我]想要[美金][100元]
21     if pattern == "<ENTITY_UserDefined>[^<]*?</ENTITY_UserDefined>((<ACTION_verb>[^<]*?
22     # write your code here
23     resultDICT["source"] = "台幣"
24     resultDICT["target"] = args[2]
25     resultDICT["amount"] = args[1]
26     # [我]想要[100元][美金]
27
28     if pattern == "<ENTITY_UserDefined>[^<]*?</ENTITY_UserDefined>((<ACTION_verb>[^<]*?
29     # write your code here
30     resultDICT["source"] = "台幣" ...
31     resultDICT["target"] = args[2]
32     resultDICT["amount"] = args[1]
33     # [我]想買[日幣][10000元]
34
35     if pattern == "<ENTITY_UserDefined>[^<]*?</ENTITY_UserDefined>((<ACTION_verb>[^<]*?
36     # write your code here
37     resultDICT["source"] = "台幣" ...
38     resultDICT["target"] = args[2]
39     resultDICT["amount"] = args[1]
40     # [我]想買[100元][美金]
41
42     if pattern == "<ENTITY_UserDefined>[^<]*?</ENTITY_UserDefined>((<ACTION_verb>[^<]*?
43     # write your code here
44     resultDICT["source"] = "台幣"
45     resultDICT["target"] = args[1]
46     resultDICT["amount"] = args[2]
47     # [我]想買[100元][美金]
48
49     if pattern == "<ENTITY_UserDefined>[^<]*?</ENTITY_UserDefined>((<ACTION_verb>[^<]*?
50     # write your code here
51     resultDICT["source"] = "台幣"
52     resultDICT["target"] = args[2]
53     resultDICT["amount"] = args[1]
```

其中一個 intent 打開後的模樣！

What We Have Learned Today?

1. Paradigm Shift of Modern Linguistics.
2. Structure Rocks!
3. Language is about Structures, not Words
4. Linguistics studies structures, not grammars.
5. Cognition system works on structure, not bag-of-words nor
bag-of-entities
6. Loki (Linguistic Oriented Keyword Interface) helps processing
semantics (NLU).

DROIDTOWN

API Github: <https://github.com/Droidtown/ArticutAPI>

API Doc.: <https://api.droidtown.co>

Twitter: <https://twitter.com/DroidtownLing>

FB FansPage: <https://www.facebook.com/Droidtown>

FB FansPage: <https://www.facebook.com/Articut>

諸君，要不要來做真正的
「強人工智慧」呀？

