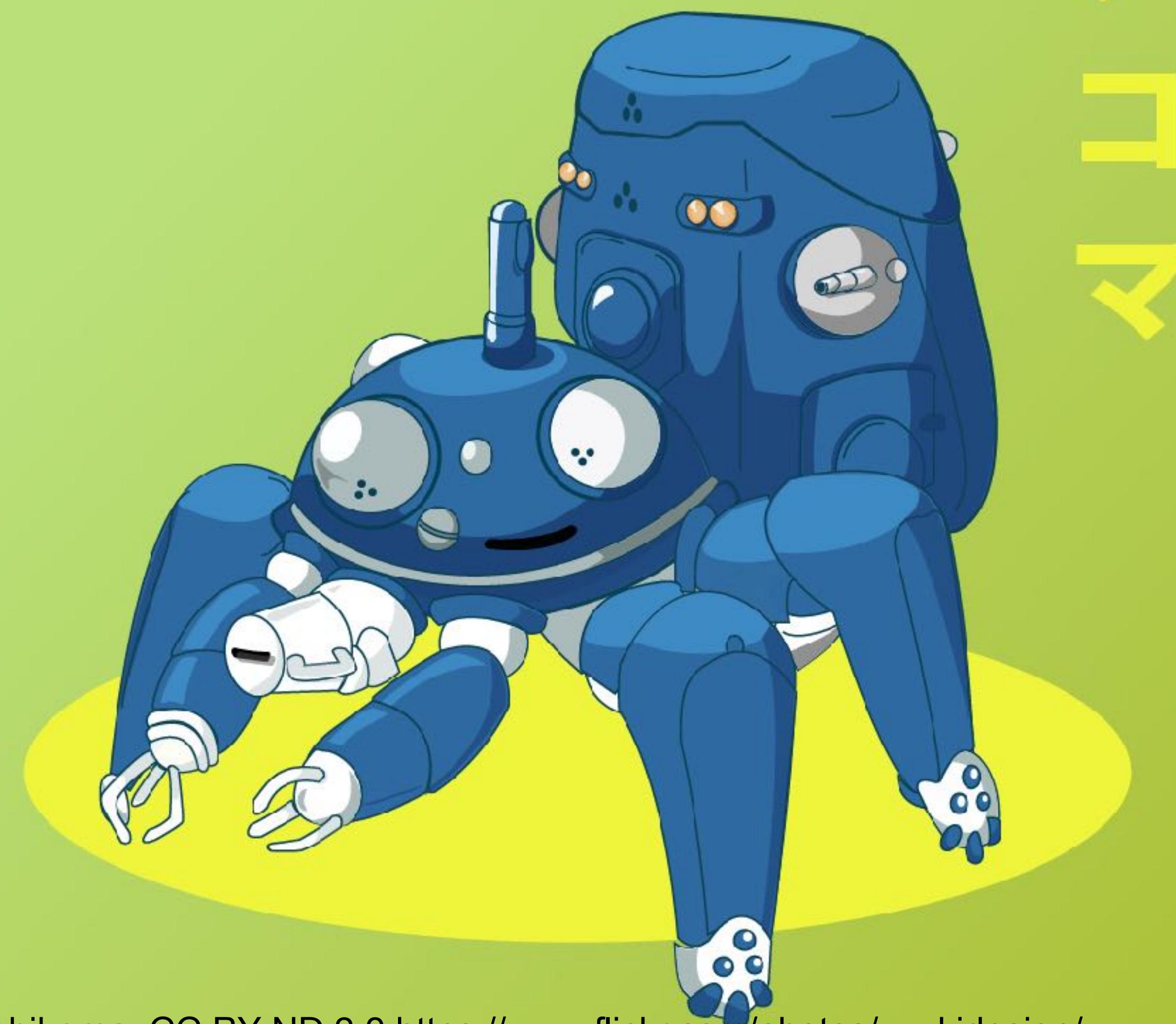


寫個漂亮又能幹  
的中文斷詞系統  
... 然後讓它養我

開箱 Articut 中文斷詞系統

@PyConTW 2019

PeterWolf  
(卓騰語言科技)



# 簡單地說...

本次分享內容將涉及一點點的中文語言學、我們在開發過程中遇到的有趣語料和斷詞測試，開發後續應用時，Python3.5 和 3.6 的 re 模組是如何地口是心非欺騙我們的感情，而明明說修好了 bug 的 Python3.7 的 re 模組又是如何地「再次玩弄」我們的感情。

還有看著 Articut 犯下和人類孩童一樣的語言錯誤時的感動、不同斷詞引擎處理特殊語料的修羅場以及最後用 SIGHAN 2005 的資料集，和其它文獻中的演算法進行良率競爭的結果。

# 中文沒有文法

宋慶元初趙子直當國加朱文公為侍講文公欣然而至積誠感悟且編次講義以進寧宗喜令點句以來日請問上曰宮中常讀之大要放心耳公因益推明其說曰坐下既知學問之要願勉強而力行之退謂其徒曰上可與為善若與得賢者輔道天下有望矣然



什麼沒文法？  
我切給你看！

宋慶元初  
趙子直當國  
召朱文公為侍講  
文公欣然而至  
積誠感悟且編次講義以進  
寧宗喜令點句以來  
他日請問上曰宮中常讀之大要主求放心耳  
公因益推明其說曰坐下既知學問之要  
願勉強而力行  
之謂其徒曰上可與為善  
若與得賢者輔道天下有望矣然

# 現代中文文字都黏在一起啊

常見三歧義：

組合型歧義：

小紅帽 => 小紅/帽 OR 小\紅帽 OR 小\紅/帽 OR 小紅帽

真歧義：

美國會派出特使來訪 => 「美國」 OR 「美國國會」

交集型歧義：

我幫爸媽買了一份超值保險套餐。

OOV:

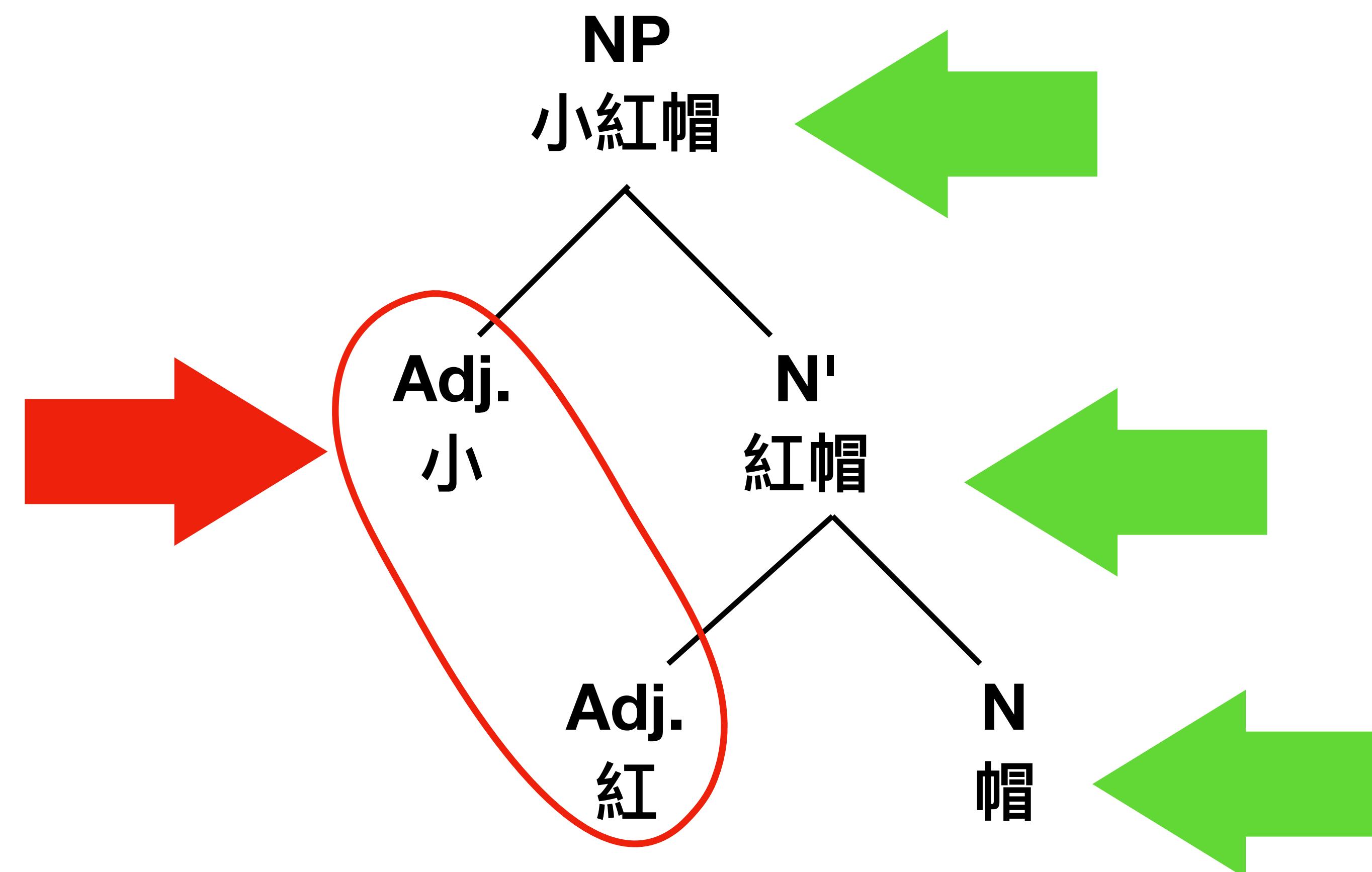
魚缸裡的谷精又浮起來了

# Chapter 1

## 語言學基礎

現代中文文字都 有結構地 黏在一起啊

組合型歧義：在句法結構上就能解釋哪些組合是正確的



現代中文文字都 **有層次地** 黏在一起啊

真歧義：「語言」層和「知識」層兩個維度各自解讀

美國會派出特使來訪

因為具有「美國是民主國家」 + 「民主國家有國會」 的知識，才會覺得這個句子有兩個意思。我們試試看

銀河第一帝國會派出特使來訪

有沒有發現歧義的感覺開始消失了？

現代中文文字都**有層次地**黏在一起啊

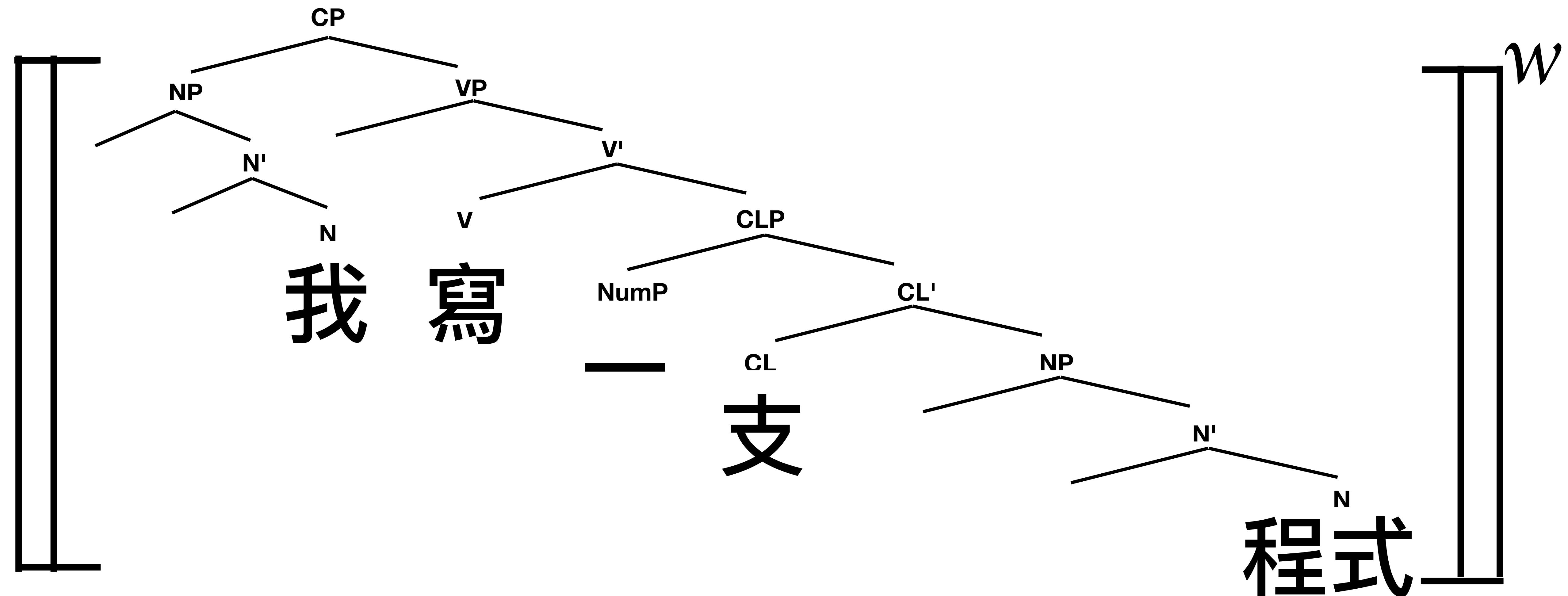
交集型歧義：「語言」層和「語境」層兩個維度各自解讀

我幫爸媽買了一份超值保險套餐

因為缺乏「語境」，所以「套」究竟該擺哪裡？你會不會又多個弟弟妹妹呢？其實兩個解讀都有可能。

你再做一次試看看！

# 現代理論語言學提出的句法樹結構



# 理論語言學提出的句法結構：只有一種句法樹

中央研究院-中文剖析樹檢索系統

[treebank.sinica.edu.tw](http://treebank.sinica.edu.tw) ▾

中文句結構樹資料庫簡介. Introduction. 「中文句結構樹資料庫」(Sinica Treebank Version 3.0) 包含了6個檔案：61,087個中文樹圖，361,834個詞，是中央研究院詞庫 ...

61087 : 1

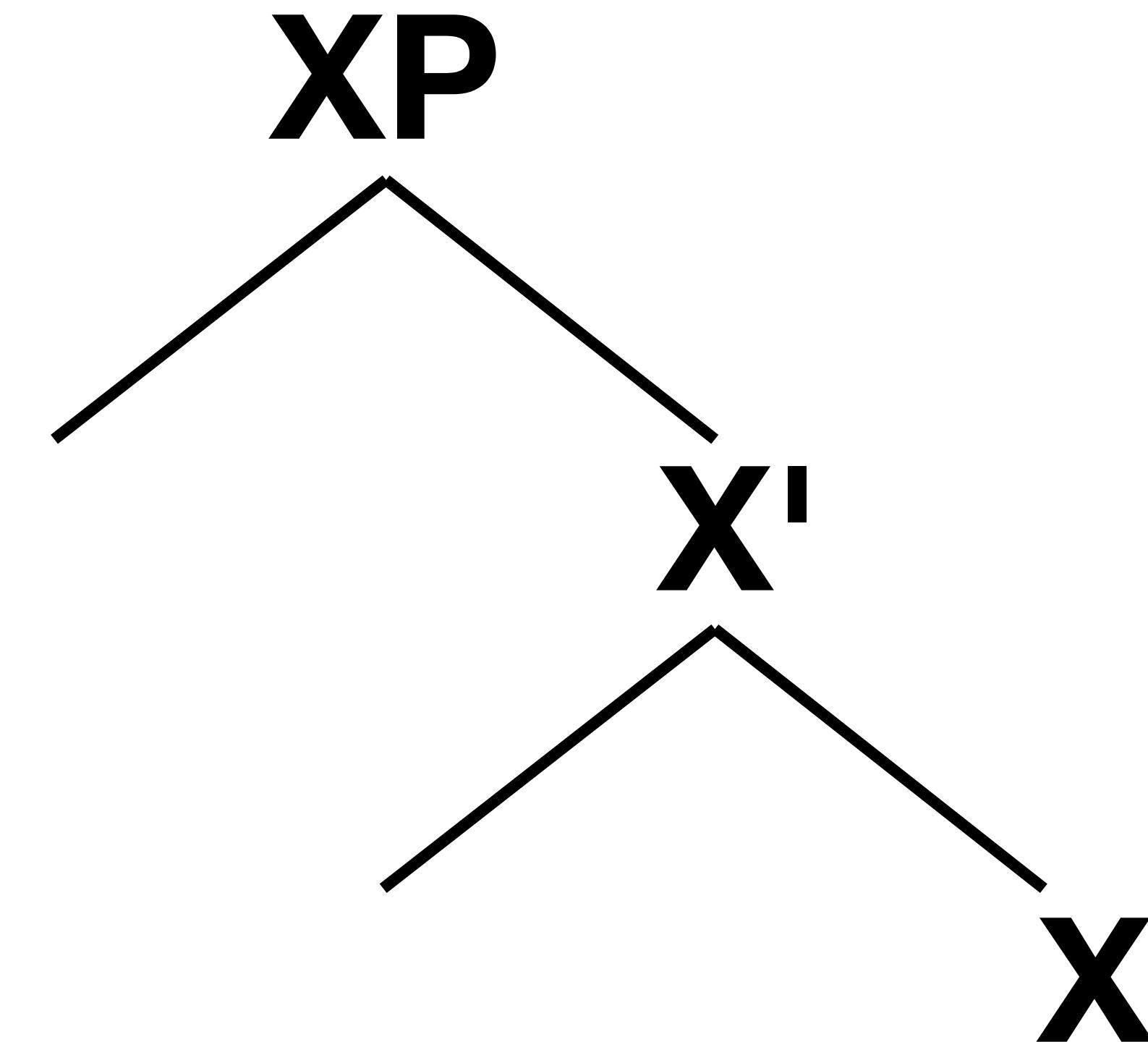
One Tree to Rule Them ALL



# 語言學概論的概論的概論的概論的懶人包

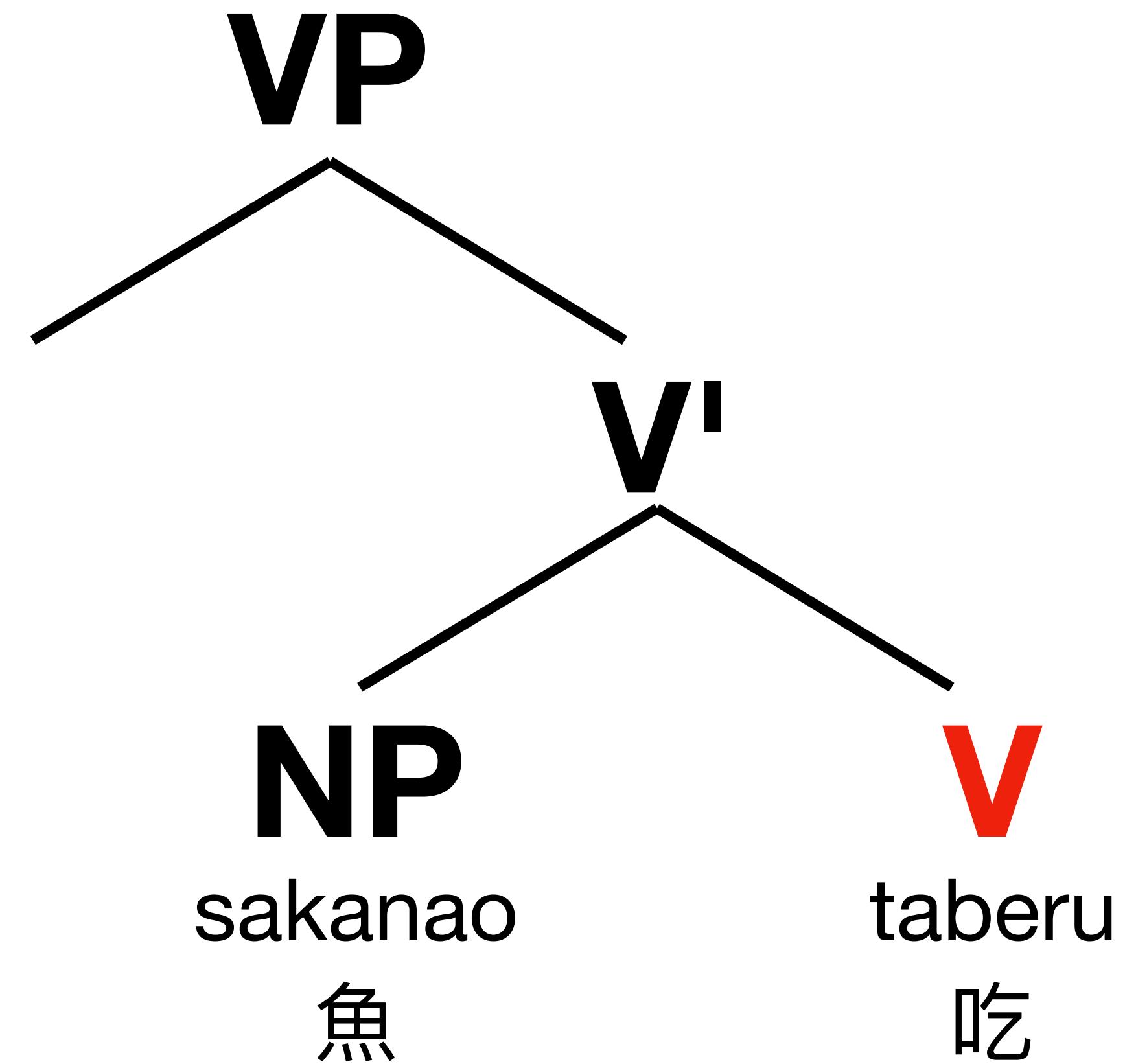
的一言以蔽之...

- 語言學的假設：
  - 所有人類大腦結構及功能是一致的
  - 所有人類都具有一樣的語言處理機制
  - 人類的母語語言習得機制允許參數改變
  - 所有語言的差異是來自參數的差異
  - 參數的變化是有限的
- 句法的 X-bar 基本結構如下：

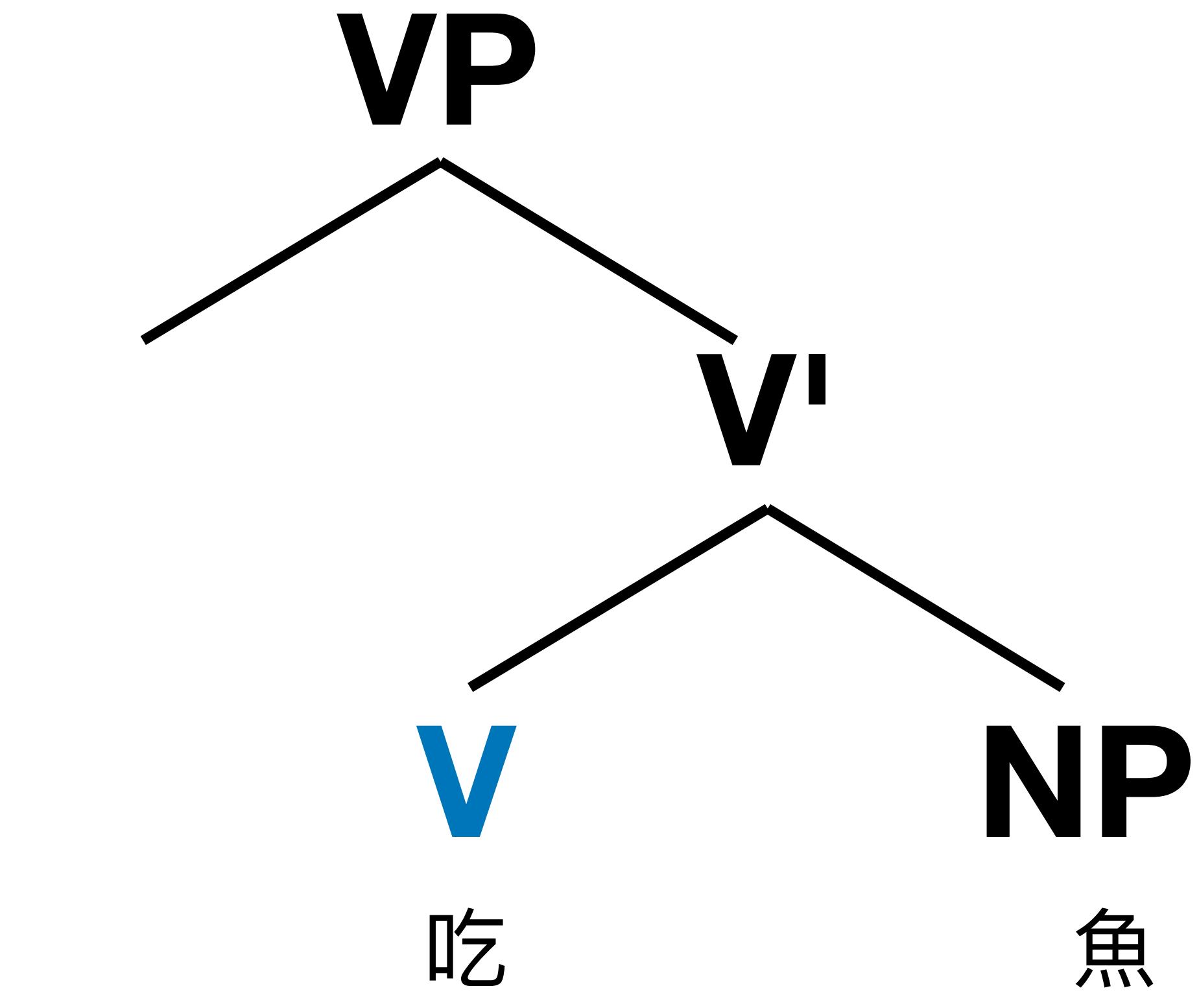


# 以動詞 Verb Phrase 參數為例：

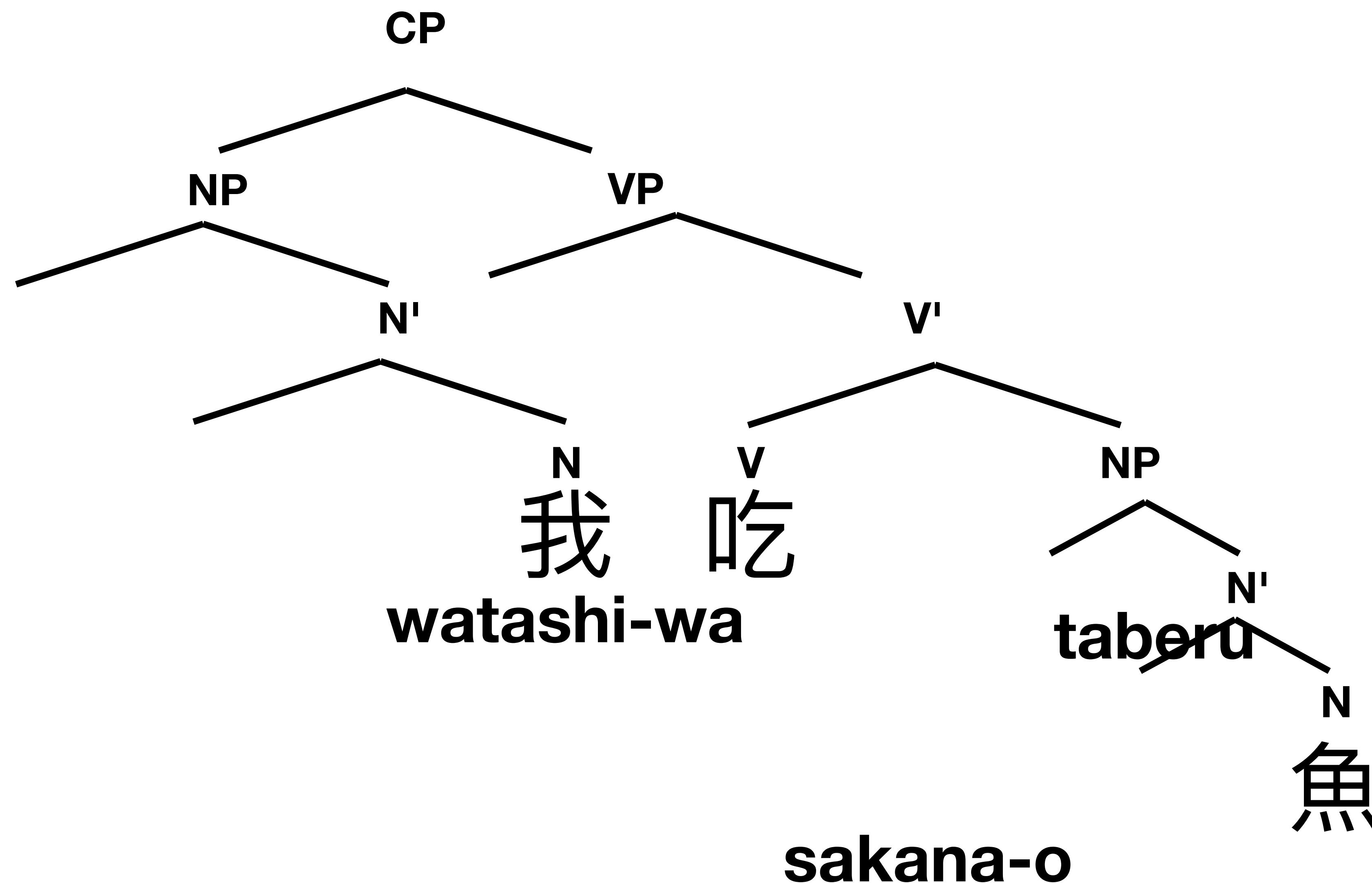
日文動詞：參數為 "head final"



中文動詞：參數為 "head first"



# 中文/日文使用「同一棵」句法樹 + 參數調整



## 小結語：

語言是有內部結構的

語言的內部結構可以用規則處理

語言規則的總數量是有限

既然它有結構，  
我看不如...



# Chapter 2

## 開發 Articut 索詞引擎

# 實作 Articut 時，架構設計考量的重點

既然有 X-bar 的結構，那表示詞性和斷詞是同時發生的

- 用結構解決，換言之我們**不需要大數據**做模型訓練
- 除了斷詞，還能**標上詞性標記**
- 有詞性標記，延伸的**應用可以用 RE 撰寫**
- 要能呈現歧義的變化



三個月後…

ps. 上圖為不相關的兩位路人

# 蔡英文總統談話逐字稿 (正式口語文字記錄)

四十年 / 前 / 的 / 這 / 一天 / 臺灣 / 關係法 / 通過 / 了 /  
為 / 臺美 / 關係 / 開啟 / 新頁 /

<TIME\_year>四十年</TIME\_year><RANGE\_period>前</RANGE\_period><FUNC\_inner>的</FUNC\_inner><FUNC\_determiner>這</FUNC\_determiner><TIME\_day>一天</TIME\_day><LOCATION>臺灣</LOCATION><ENTITY\_nouny>關係法</ENTITY\_nouny><ACTION\_verb>通過</ACTION\_verb><ASPECT>了</ASPECT>

<AUX>為</AUX><ENTITY\_nouny>臺美</ENTITY\_nouny><ENTITY\_noun>關係</ENTITY\_noun><ACTION\_verb>開啟</ACTION\_verb><ENTITY\_nouny>新頁</ENTITY\_nouny>

可分辨一個是 "法條"，  
而另一個是 "關係"

# 館長直播逐字稿（日常口語文字記錄）

齁 / 盡量 / 不要 / 買 / 武器 / 了 /

這個 / 我 / 沒有 / 辦法 / 接受 /

口癖也處理得不錯！

'<CLAUSE\_Particle>齁</CLAUSE\_Particle>', '，'，'<MODIFIER>盡量</MODIFIER><FUNC\_negation>不要</FUNC\_negation><ACTION\_verb>買</ACTION\_verb><ENTITY\_nouny>武器</ENTITY\_nouny><ASPECT>了</ASPECT>', '，'，'<ENTITY\_DetPhrase>這個</ENTITY\_DetPhrase><ENTITY\_pronoun>我</ENTITY\_pronoun><FUNC\_negation>沒有</FUNC\_negation><ENTITY\_nouny>辦法</ENTITY\_nouny><ACTION\_verb>接受</ACTION\_verb>'

# 新聞文本測試 (結構較完整的書面語)

今天 / 臺北 / 忠孝東路 / 很 / 熱鬧  
許多 / 民眾 / 搶看 / 懸日

<TIME\_day>今天</TIME\_day><LOCATION>臺北</LOCATION><ENTITY\_nouny>忠  
孝東路</ENTITY\_nouny><FUNC\_degreeHead>很</  
FUNC\_degreeHead><ACTION\_verb>熱鬧</ACTION\_verb>  
<QUANTIFIER>許多</QUANTIFIER><ENTITY\_noun>民眾</  
ENTITY\_noun><ACTION\_verb>搶看</ACTION\_verb><ENTITY\_nouny>懸日</  
ENTITY\_nouny>

# ptt 八卦板文本測試 (結構較自由的書面語)

這時 / 川董 / 拿出 / 鋼筆 / 開始 / 簽名 / 在 / 杯墊 / 上 /

這 / 是不是 / 暗示 / 小郭 / 參選 / 將會 / 是 / 個 / 杯具 /

<TIME\_justtime>這時</TIME\_justtime><ENTITY\_oov>川董</ENTITY\_oov><ACTION\_verb>拿出</ACTION\_verb><ENTITY\_nouny>鋼筆</ENTITY\_nouny><ACTION\_verb>開始</ACTION\_verb><ACTION\_verb>簽名</ACTION\_verb><FUNC\_inner>在</FUNC\_inner><ENTITY\_nouny><RANGE\_locality>上</RANGE\_locality>', '， '， '<FUNC\_determiner><CLAUSE\_AnotAQ><AUX>是</AUX><FUNC\_negation><AUX></CLAUSE\_AnotAQ><ACTION\_verb>暗示</ACTION\_verb><ENTITY\_pronoun><AUX><CLAUSE\_AnotAQ><ACTION\_verb>參選</ACTION\_verb><MODAL>將會</MODAL><AUX>是</AUX><ENTITY\_classifier>個</ENTITY\_classifier><ENTITY\_nouny>杯具</ENTITY\_nouny>

即便是 OOV，  
也能切得不錯哦！

沒有「內建字典」，只靠「句法結構」的錯誤是具有「可解釋性」的

正確：

...餃子 / 包 / 高麗菜...

錯誤：

...麵 / 包 / 牛奶...

歧義：

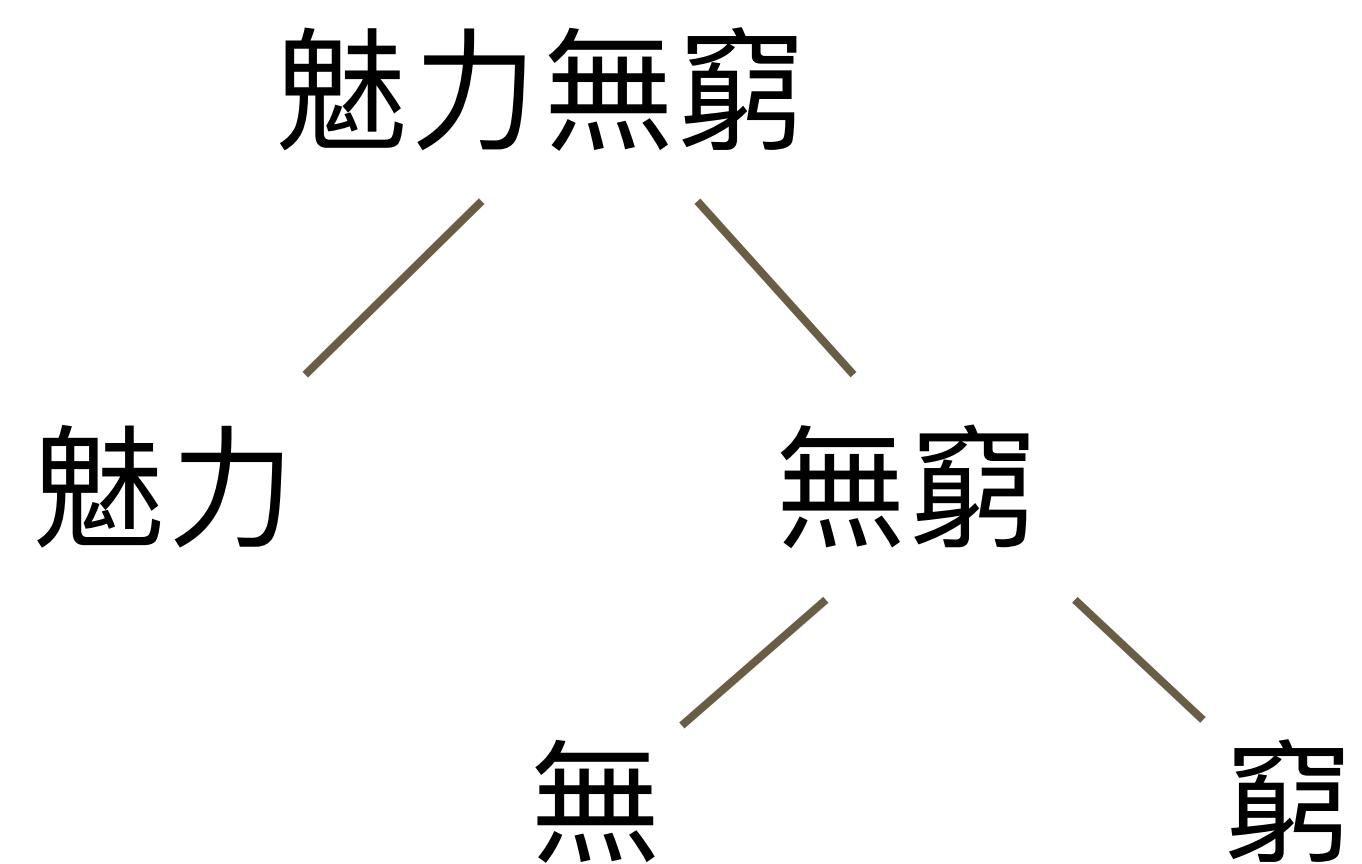
借 600 萬養老金

老金是哪位？  
養起來這麼貴啊？

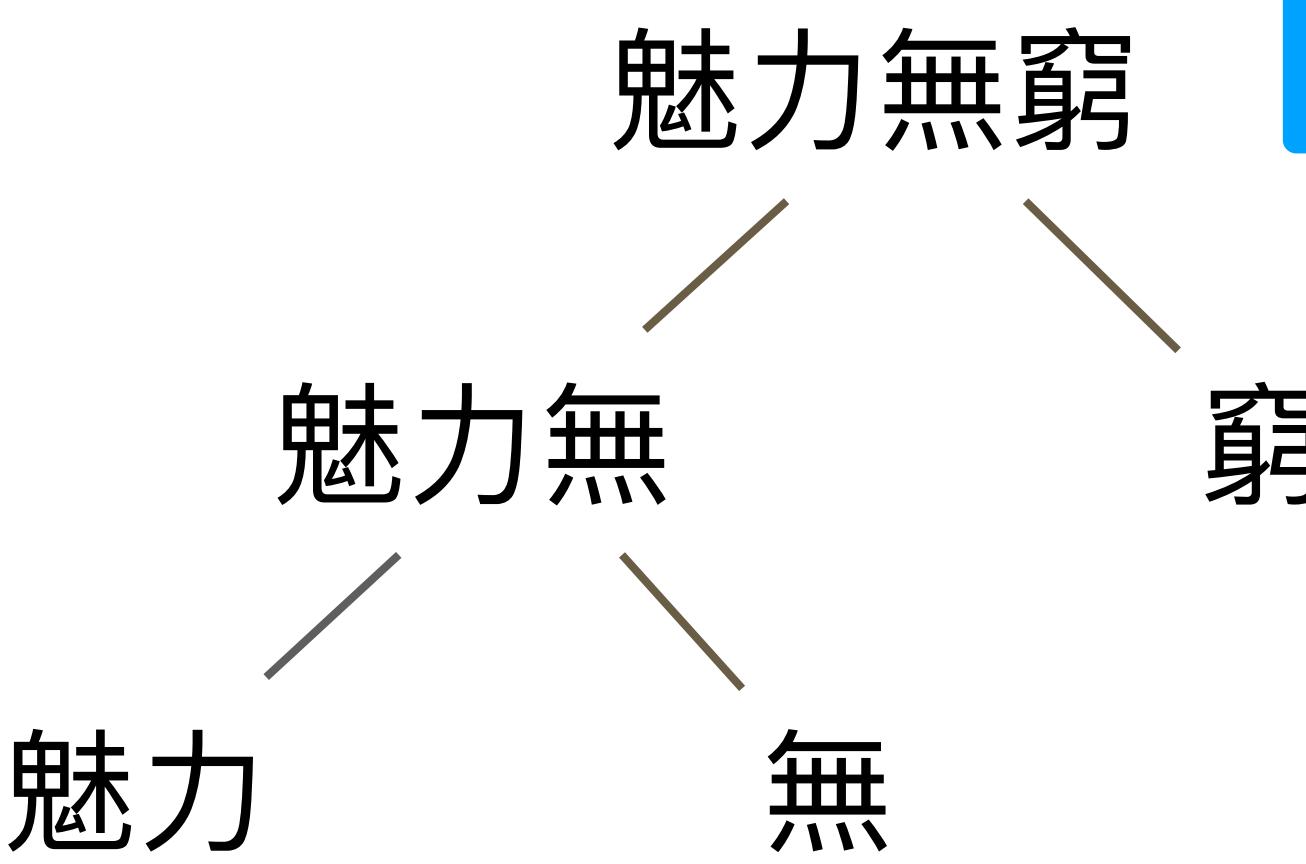
# 人類的幽默從何而來？兩種結構的歧義

斷詞：我個人的特質就是「魅力無窮」

=> 魅力/無/窮



=>擁有「無窮的魅力」



=> 魅力無，且窮

處理「斷詞」是不夠的！  
有結構，才能處理歧義。  
能處理歧義，人工智慧才  
會懂人類的幽默！



# **Chapter 3**

# **RE**

# 任性總裁壞RE：我想拿詞性標記來做更多事！

```
<TIME_day>今天</TIME_day><LOCATION>臺北</LOCATION><ENTITY_nouny>忠孝東路</ENTITY_nouny>  
<FUNC_degreeHead>很</FUNC_degreeHead><ACTION_verb>熱鬧</ACTION_verb>
```



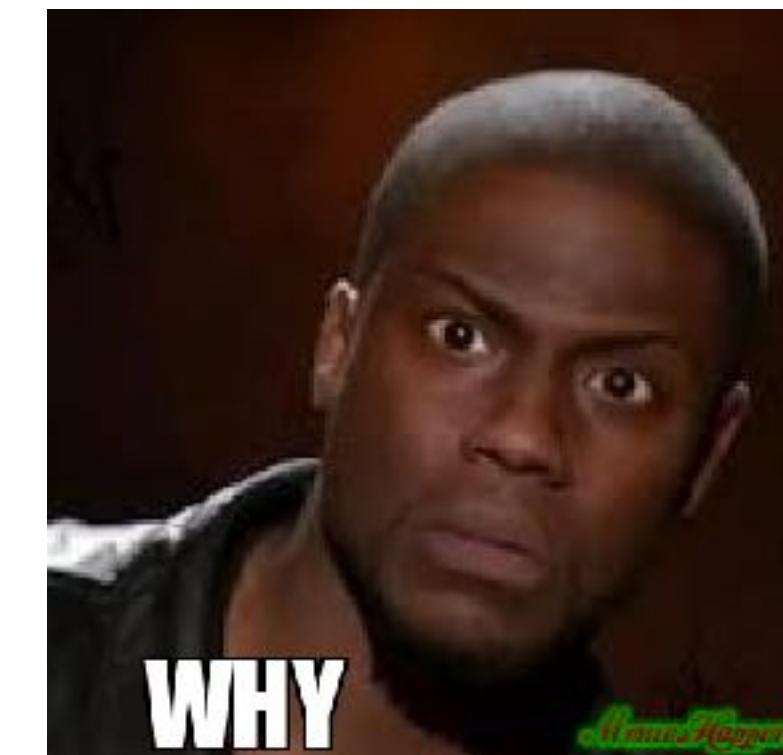
```
pat = re.compile("\d樓\d號(\d樓)?")
```

```
list(re.finditer(pat, "4號"))
```

```
#4號
```

```
re.findall(pat, "4號")
```

```
#[']
```



# 腦閻！尼這 RE 有貓餅啊！

## Python3.5

```
pat = re.compile("\d樓\d號(\d樓)?")  
  
list(re.finditer(pat, "4號"))  
  
#4號  
  
re.findall(pat, "4號")  
  
#[']
```

## Python3.6

```
pat = re.compile("\d樓\d號(\d樓)?")  
  
list(re.finditer(pat, "4號"))  
  
#4號  
  
re.findall(pat, "4號")  
  
#[']
```

<https://docs.python.org/3.6/library/re.html>

### Note

Due to the limitation of the current implementation the character following an empty match is not included in a next match, so `.findall(r'^\w+', 'two words')` returns `["wo", 'words']` (note missed "t")

This is changed in Python 3.7.

# 快快樂樂打開 Python3.7...

## Python3.7

```
#我們先試試看官方的修改...
```

```
pat = re.compile("^\\w+")
```

```
list(re.finditer(pat, "two words"))
```

```
re.findall(pat, "two words")
```

## Python3.7

```
pat = re.compile("\\d樓\\d號(\\d樓)?")
```

```
list(re.finditer(pat, "4號"))
```

```
#4號
```

```
re.findall(pat, "4號")
```

```
#[']
```

*"This is changed in  
Python 3.7." 言猶在耳...*



# Chapter 4

## 修羅場

# 重新審視定義：中文斷詞究竟是什麼？

『讓電腦把詞彙以「意義」為單位切割出來』(Fukuball)

但是基於統計方法做出來的斷詞結果，給的是「資料分佈」和「符號相連機率」，而不是「意義」啊！

『讓電腦把詞彙以「句子結構上的意義」為單位切割出來』  
(PeterWolf @ PyConTW 2019)

# 斷詞修羅場：長名詞表現 (組合型歧義)

**Jieba**

國 / 關 / 中心 / 的 / 研究 / 人員

**CKIP**

國關 / 中心 / 的 / 研究 / 人員

**Monpa**

國關中心 / 的 / 研究 / 人員

**CKIPtagger**

國關中心 / 的 / 研究 / 人員

能把「國關中心」黏  
在一起，才算過關哦！

**Articut**

國關中心 / 的 / 研究 / 人員



# 斷詞修羅場：罕見句表現 (交集型歧義)

Jieba

我 / 想 / 過 / 過 / 過兒 / 過 / 過 / 的 / 日子

CKIP

我 / 想 / 過 / 過過 / 兒 / 過過 / 的 / 日子

Monpa

我 / 想 / 過 / 過 / 過 / 兒 / 過 / 過 / 的 / 日子

CKIPtagger

我 / 想 / 過 / 過 / 過兒 / 過 / 過 / 的 / 日子

Articut

我 / 想過 / 過 / 過兒 / 過過 / 的 / 日子

我 / 想 / 過 / 過 / 過兒 / 過 / 過 / 的 / 日子



這個句子有兩種結構。一個表示「曾經思考過這件事」，另一個則是表示「想要試試看」。  
Articut 有能力呈現兩種可能。

# 斷詞修羅場：POS (詞性標記) 轉品表現

## Jieba

努力<sub>(ad)</sub> 才能(v) 成功<sub>(a)</sub>

他<sub>(r)</sub> 的<sub>(ui)</sub> 領導<sub>(n)</sub> 才能(v) 很<sub>(zg)</sub> 突出<sub>(v)</sub>

## CKIP

努力<sub>(VH)</sub> 才能(Na) 成功<sub>(VH)</sub>

他<sub>(Nh)</sub> 的<sub>(DE)</sub> 領導<sub>(VC)</sub> 才能(Na) 很<sub>(Dfa)</sub> 突出<sub>(VH)</sub>

## Monpa

努力<sub>(VH)</sub> 才(Da) 能(D) 成功<sub>(VH)</sub>

他<sub>(Nh)</sub> 的<sub>(DE)</sub> 領導<sub>(Nv)</sub> 才(Da) 能(D) 很<sub>(Dfa)</sub> 突出<sub>(VH)</sub>

## CKIPtagger

努力<sub>(VH)</sub> 才(Da) 能(D) 成功<sub>(VH)</sub>

他<sub>(Nh)</sub> 的<sub>(DE)</sub> 領導<sub>(Na)</sub> 才(Da) 能(D) 很<sub>(Dfa)</sub> 突出<sub>(VH)</sub>

## Articut

努力<sub>(verb)</sub> 才能(MODAL) 成功<sub>(verb)</sub>

他<sub>(pronoun)</sub> 的<sub>(inner)</sub> 領導<sub>(verb)</sub> 才能(nouny) 很<sub>(modifier)</sub> 突出<sub>(verb)</sub>

「才能」在兩個句子裡的 POS  
一樣的話，就一定是有一個錯了。



# 斷詞修羅場：NER (命名實體辨識)

## CKIPtagger

**Segmentation ==>**

```
[['復活島', '北方', '兩百', '公里', '處', '發現', '失事', '殘骸', '。']]
```

**NER ==>**

```
[(5, 9, 'QUANTITY', '兩百公里')]
```

**Segmentation ==>**

```
[['北方', '兩百', '公里', '處', '發現', '失事', '殘骸', '。']]
```

**NER ==>**

```
[(())]
```

**Segmentation ==>**

```
['99', '越南', '盾', '。']
```

**NER ==>**

```
[(0, 2, 'CARDINAL', '99'), (2, 5, 'PRODUCT', '越南盾')]
```

**Segmentation ==>**

```
['99', '印尼', '盾', '。']
```

**NER ==>**

```
[(0, 2, 'CARDINAL', '99'), (2, 5, 'GPE', '印尼盾')]
```

## IASL Multi-Objective NER POS Annotator

韓國瑜伽老師教韓國瑜伽藍尊者的故事。

Length limit: 50 characters. Longer text will be truncated.

Results:

韓國 LOC 瑜加 NA 老師 NA 教 C 韓 LOC 國瑜伽 NA 藍尊 NA 者 NA 的 DE 故事 NA  
S PERIODCATEGORY

中文的 NER 不是一個適用  
「機器學習」來解決的問題。  
不是不能解，而是偵測結果的  
「缺乏一致性」對後續應  
用的開發影響太大。



# 斷詞修羅場：利用句法樹同時完成斷詞、POS、NER

假設你的 NLP 應用需要「中文斷詞」、「詞性標記 POS」和「命名實體辨識 NER」以及後續的步驟...而你採用的方案有 95% 的中文斷詞 CWS 正確率、93% 的 POS 正確率以及 80% 的 NER 正確率。那麼你的應用服務的基本良率就是...

CWS	POS	NER
95%	x 93%	x 80% = 70.68%

但利用句法樹建構的 Articut，因為中文斷詞 CWS、POS 和 NER 三個步驟不是用分別跑資料來訓練模型，而是**同步完成**的。因此在全部條件相同的情況下，應用服務的基本良率就是...

CWS	POS	NER
92.55%	x 100%	x 100% = 92.55%

[https://miro.medium.com/max/1280/1\\*QpLbx800bbki-Vf5OJ2zA.png](https://miro.medium.com/max/1280/1*QpLbx800bbki-Vf5OJ2zA.png)

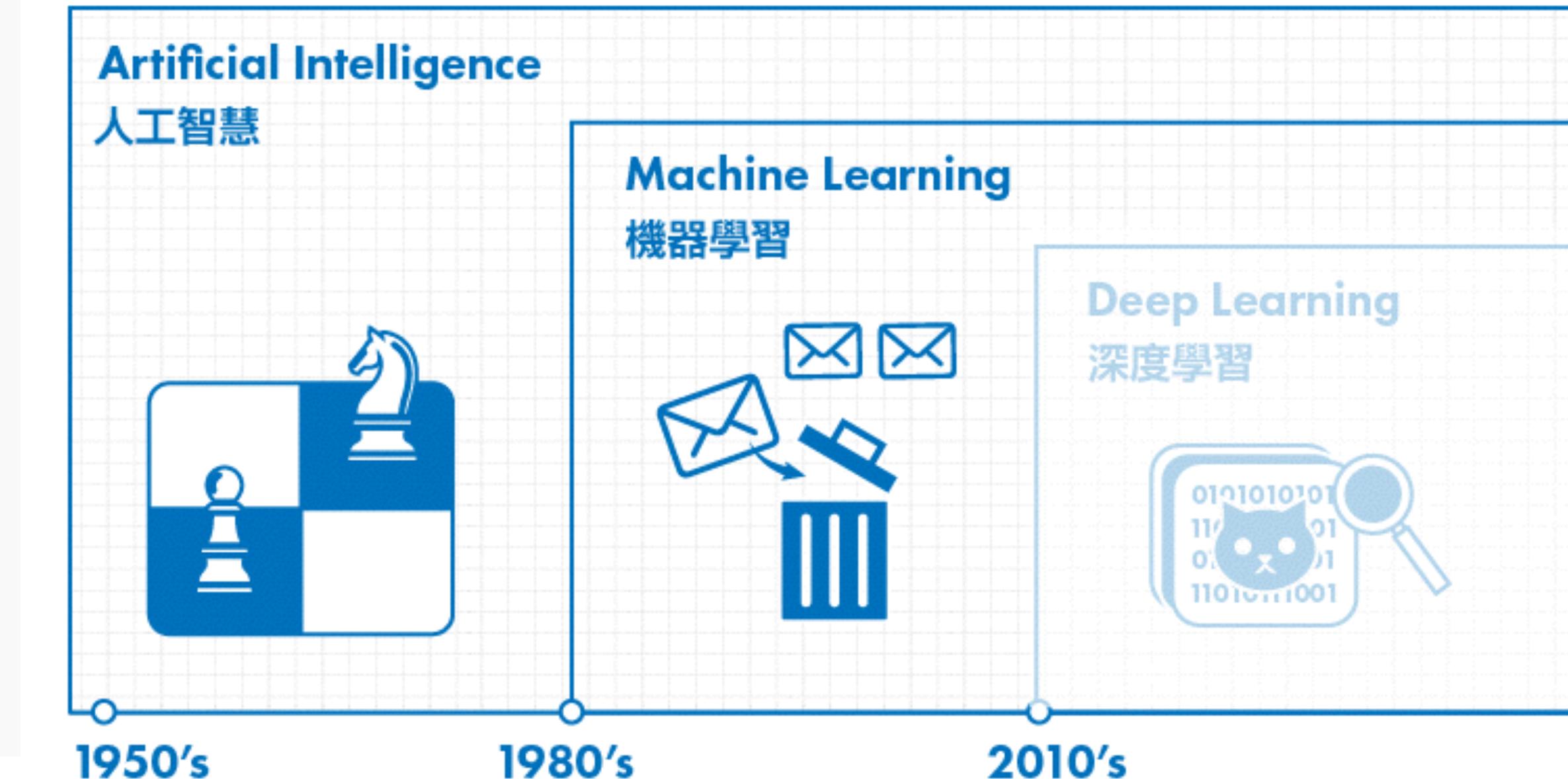
<https://medium.com/職人簡報與商業思維/我讀-人工智能在台灣-搞懂的四件事-經理人學習人工智能的第一本書-f69eb8b81358>

## 一、人工智能發展簡史



## 二、人工智能的歷史演進

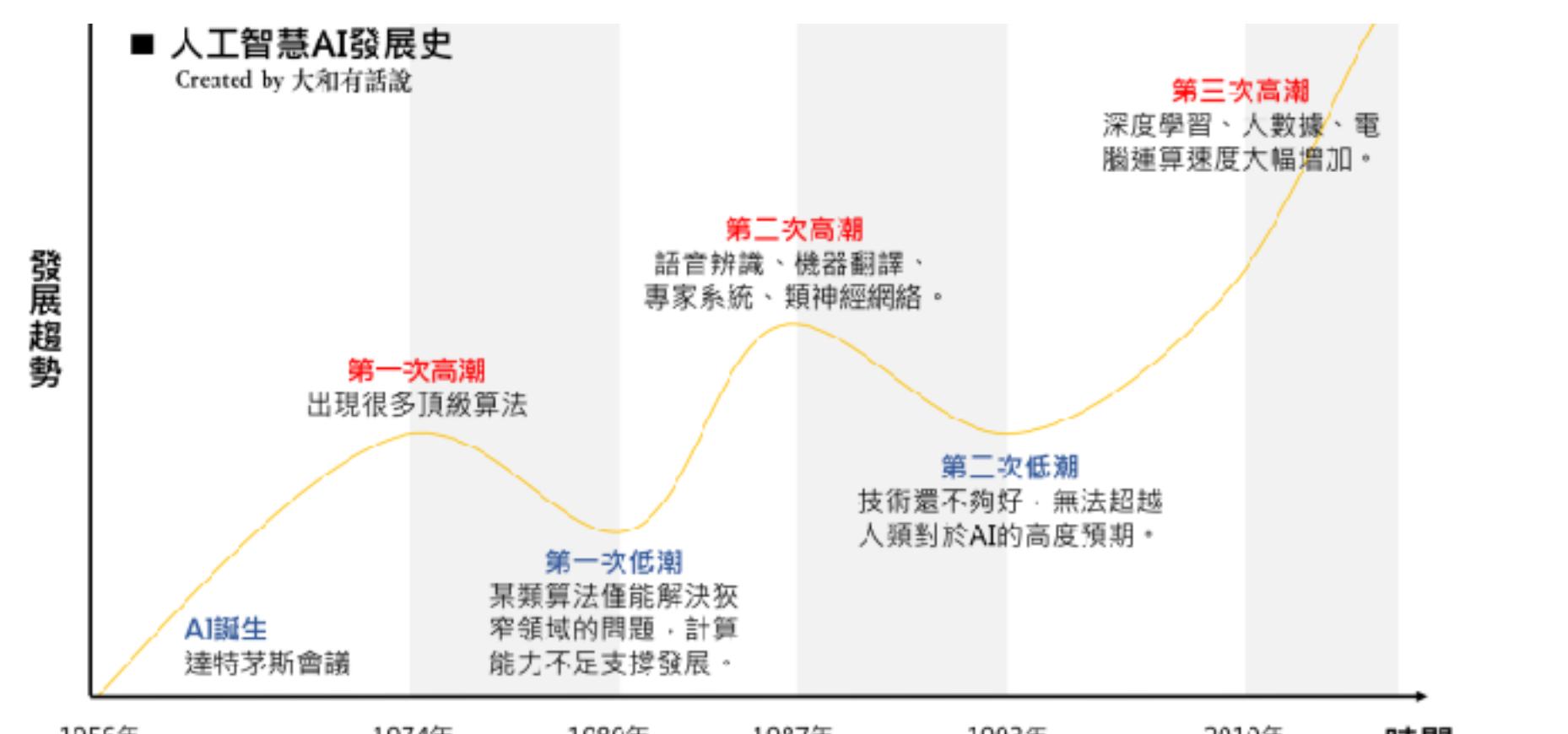
人工智能的一項分支是機器學習，機器學習的一項分支是深度學習（類神經網路）



第三次高潮  
深度學習、人數據、電腦運算速度大幅增加。  
電腦出現時便有人工智慧的呼聲出現，讓電腦能夠自行從歷史資料中學會一套技能

<https://images.stockfeel.com.tw/stockfeellimage/2016/12/圖01.png>

<https://www.stockfeel.com.tw/人工智能的黃金年代：機器學習/>



時間

<https://dahetalk.files.wordpress.com/2018/04/e4babae5b7a5e699bae685a7iae799bce5b195e58fb2.png?w=748>

<https://dahetalk.com/2018/04/08/完整解析ai人工智能：3大浪潮+3大技術+3大應用 | />

# 中文分词十年又回顾: 2007-2017\*

## Chinese Word Segmentation: Another Decade Review (2007-2017)



1. 用統計方法算組合機率的斷詞方法，良率沒有上升 (e.g., Jieba...等)
2. 用機器學習的方法也沒有明顯的優勢。
3. 既然語言確定有內部結構，那麼不如試試 看用 Rule-base 的方法吧！

### [内容简介]

本文回顾中文分词在2007-2017十年间的技术进展，尤其是自深度学习渗透到自然语言处理以来的主要工作。我们的基本结论是，中文分词的监督机器学习方法在从非神经网络方法到神经网络方法的迁移中尚未展示出明显的技术优势。中文分词的机器学习模型的构建，依然需要平衡考虑已知词和未登录词的识别问题。尽管迄今为止深度学习应用于中文分词尚未能全面超越传统的机器学习方法，我们审慎推测，由于人工智能联结主义基础下的神经网络模型有潜力契合自然语言的内在结构分解方式，从而有效建模，或能在不远将来展示新的技术进步成果。

赵海 Hai Zhao

蔡登 Deng Cai

上海交通大学 Shanghai Jiao Tong University

黄昌宁 Changning Huang

清华大学 Tsinghua University

揭春雨 Chunyu Kit

香港城市大学 City University of Hong Kong

「...而以我在自然言領域工作的經驗來看，越深入研究，越能感覺到語言學知識不足的掣肘。特別是深層次的語義了解，脫離了語言學知識，就會變成無源之水、無本之木。常見的自然語言處理書籍對於解決實際問題的方法說明已經足夠豐富，但對於語言學基礎理論的介紹和思考還略顯不足...」

-- 賈文杰

1998 人民日報語料庫高級研究員  
360 搜尋引擎分詞  
獵報移動自然語言處理部負責人

# 清華AI研究院基礎理論研究中心成立

訪文：「...在清華大學看來，現在人們眼中的人工智能幾乎全是機器學習，但機器學習只是 AI 領域中的很小一部分。人工智能還有知識表述、不確定性推理等很大一部分領域有待開發。或許在未來，人工智能的發展方向將出現很大轉變...」

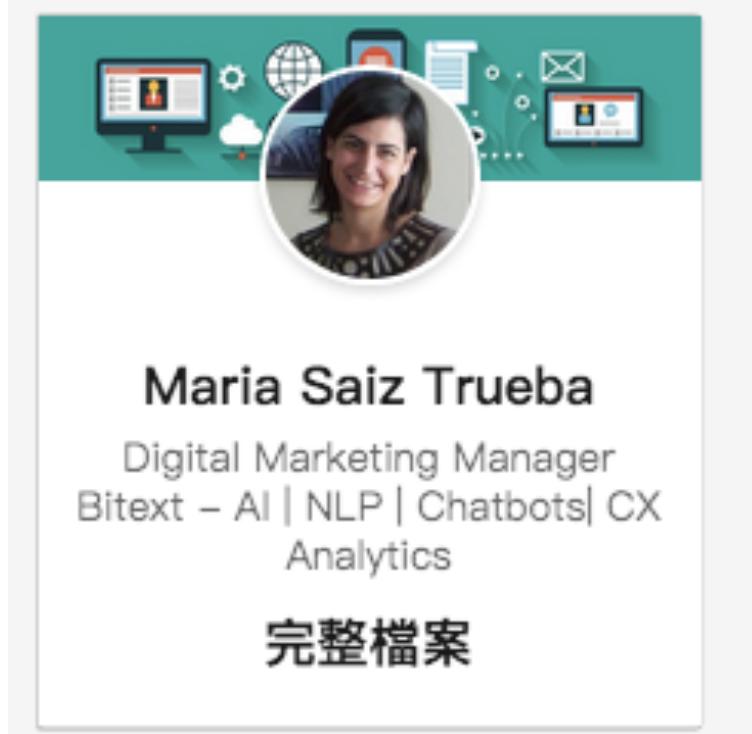
src. <https://www.jiqizhixin.com/articles/2019-05-06-8>

常常有人說「中國的 AI 發展領先世界，因為他們那裡的資料取得非常容易。」沒有資料當然有「無資料」的做法呀！

現在全世界「最大的中文 NLP 技術使用國」要轉向做基礎理論研究了。若我們能利用理論語言學做出來的 NLP 技術來往上衝刺，絕對有在未來的 AI 舞台上做主角的機會。



# Articut 是唯一一個用語言學規則的嗎？



新增回應.....

**Maria Saiz Trueba** 作者 2 週前 ...  
Digital Marketing Manager Bitext – AI | NLP | Chatbots| CX Ana...

Thank you for your comments David! Our API is based on a proprietary linguistic engine (linguistic rules). So, it's a mostly symbolic system, although we use statistical techniques for data collection and bootstrapping mostly. (已編輯)

翻譯年糕

Department of  
Language Science  
**UCI** School of Social Sciences

Search

加州大學爾灣分校成立了全新的「語言科學」學系  
探索「語言學」、「電腦科學」和「認知科學」...  
等基礎學科的結合領域

**bitext** see help of understand business

NLP FOR BOTS & VIRTUAL ASSISTANTS NLP FOR CX ANALYTICS NLP FOR AI ENGINES RESOURCES COMPANY BLOG BITEXT API

## Your Multilingual NLP Middleware

bitext 是一間利用「語言學規則」提供歐美各種語言的 NLP 解決方案的跨國企業

**Net Solutions for AI Engines**  
NLP Tool for your Machine Learning/ Deep Learning Engine [Download](#)

**Customer Experience (CX)**  
Automotive Industry Case Study  
Multilingual NLP engine from scratch [Download](#)

**Chatbots & Virtual Assistants**  
Techcrunch Case Study  
Multilingual Data for training bots to increase accuracy [Download](#)

A FRESH TAKE ON LANGUAGE  
The Department of Language Science is the first of its kind, synthesizing traditional areas of inquiry into innovative training programs.

# 斷詞修羅場：最後試煉之「你的模型超胖！」

CKIPtagger on  
i5-7500 3.40GHz CPU, 32GB Ram

Articut on Raspberry Pi Zero  
(ARMv6 1Ghz CPU, 512MB RAM)

```
dtstaff@cklab222-new:~/CKIP_Tagger$ lscpu | grep "Model name:" | sed -r 's/Model name:\s{1,}//g'
```

Done in  
0.5149 sec.

```
droidtown@rpi-zero:~/articut_standalone $
```

Done in  
0.0852 sec.

WHO  
WON?  
YOU DECIDE!

```
1 [ 0.0% Tasks: 62, 14 thr; 1 running
2 [ 0.0% Load average: 0.17 0.16 0.09
3 [|| 6.0% Uptime: 8 days, 07:15:13
4 [ 0.0%
Mem[||||| 1.10G/31.3G
Swp[ 3.22M/29.8G
```

```
CPU[|| 4.5% Tasks: 47, 13 thr; 1 running
Mem[||||| 79.0M/433M Load average: 0.13 0.16 0.14
Swp[ 0K/100.0M Uptime: 04:31:26
```

# Thanks

諸君，我們幫您搞定 NLP 的基礎問題了。要不要站在我們的肩膀上，來做真正的「強人工智慧」呀？



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

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

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

Talk Resources: <https://github.com/Droidtown/PyConTW2019>

