Natural Language Processing Summer 2021 #1

Chi-Jen Wu

Topics

- An introduction to NLP
- Language modeling
- Representation learning
- Web crawling and indexes
- Word Embeddings
- Text classification
- Sequence modeling
- Machine learning models
- Deep learning models

An introduction to NLP


Neuro-Linguistic
Programming (NLP)
NLP
NEUROLOGICAL PROCESSES
LANGUAGE
EXPERIENCE
PERSONAL DEVELOPMENT
COACHING
ACHIEVE GOALS

不是這個
這是心理學的

An introduction to NLP

- The process of computer analysis of input provided in a human language (natural language) and conversion of this input into a useful form of representation.
- to perform useful and interesting tasks with human languages.
- helping us come to a better understanding of human language.

簡單說 Natural Language Processing

- 使計算機能夠理解和接受人類用自然語言輸入的指令，完成從一種語言到另一種語言的翻譯功能。
- 自然語言處理技術的研究，可以豐富計算機知識處理的研究內容，推動AI技術的發展。

Representational Systems

Hearing - Auditory (A)

Internal (i)
Pictures
Sounds
Self talk
Feelings

External (e)
Seeing
Listening
Touching
Tasting
Smelling

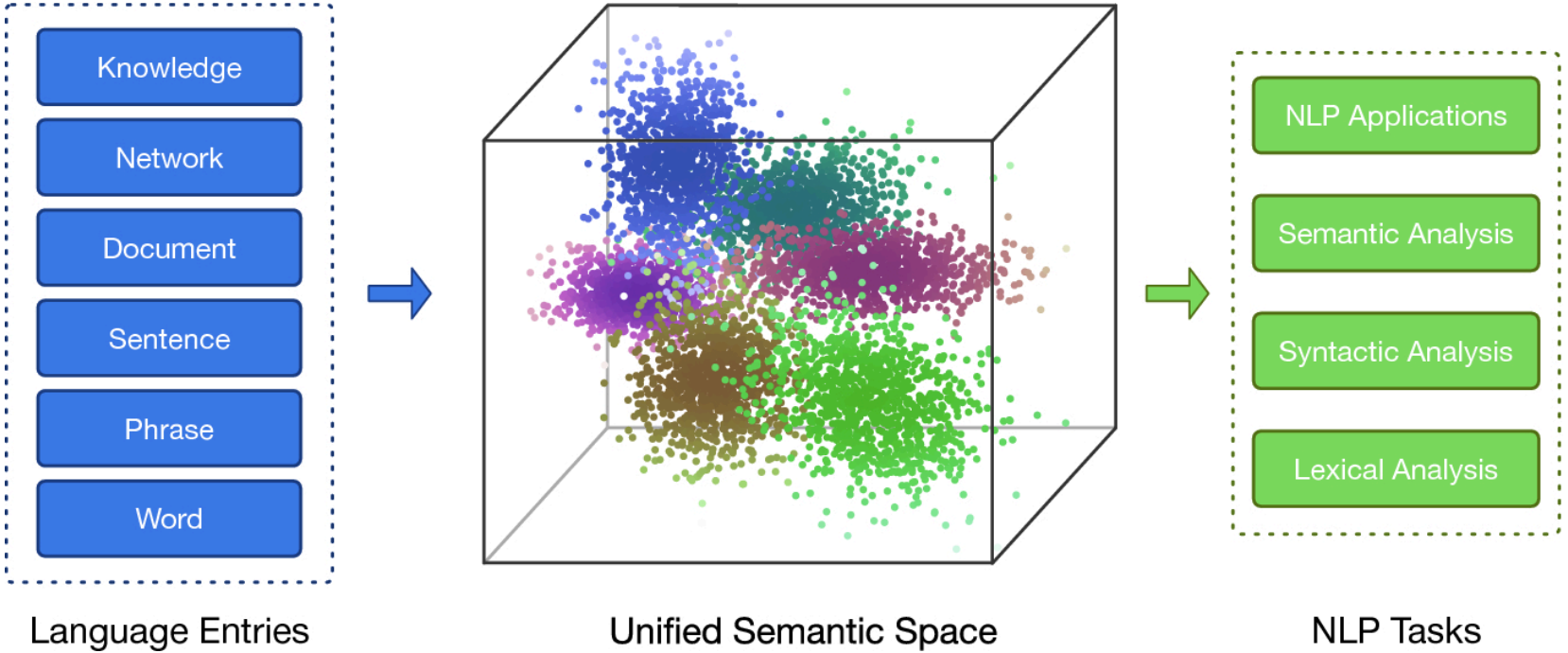
Seeing/Pictures - Visual (V)

Words/Language - Auditory Digital (Ad)
On/Off

Smells - Olfactory (K)
Tastes - Gustatory (K)

Touching/Feeling - Kinesthetic (K)

NLP Representational Systems



Forms of Natural Language

- The input/output of a NLP system can be:
  - **written text**
  - speech
- We will mostly concerned with **written text**
- To process written text, we need:
  - lexical, syntactic, semantic knowledge about the language
  - discourse information, real world knowledge
- To process spoken language (**HARDER**)
  - Need everything required to process written text, plus the challenges of speech recognition and speech synthesis.

Components of NLP

- Natural Language Understanding
  - Mapping the given input in the natural language into a useful representation
    - for computer
- Natural Language Generation
  - Producing output in the natural language from some internal representation
    - For human
- Understanding is much harder than Generation,
  - both of them are hard

為什麼難 應該是很難

- 基本上人類語言沒什麼規範 他是活的
- 錯字
  - 很多錯字用久了 變正確用法 XD
- 新詞
  - 是Open的 詞彙創造
    - 你很淡定耶
- 目前還是沒有一個很好的表現法 (for computer)
  - 好 很好 非常好 他是離散的
- 背景知識和上下文

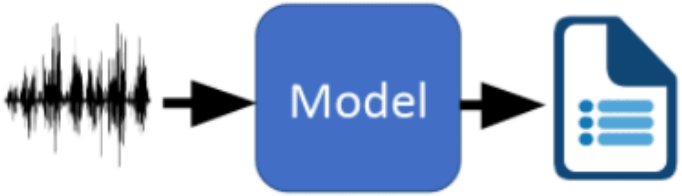
Natural Language Generation

- 利用結構化表示的語義，以輸出符合語法的、流暢的、與輸入語義一致的文本
- 內容選擇
  - 決定要表達哪些內容
- 句子規劃
  - 決定篇章及句子的結構，進行句子的融合、指代表述等
- 表層實現
  - 決定選擇什麼樣的詞彙來實現一個句子的表達

Natural Language Generation

- 目前主流的自然語言生成技術主要有
  - 基於數據驅動的自然語言生成技術
    - Machine Learning
  - 基於深度學習的自然語言生成技術
    - Deep Learning

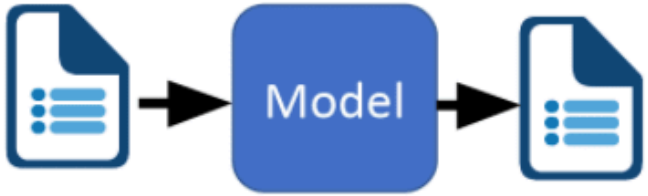
Natural Language Generation



輸入語音，產生文字

輸入文字，產生語音

輸入語音，產生語音



輸入文字，產生文字

Why NL Understanding is hard

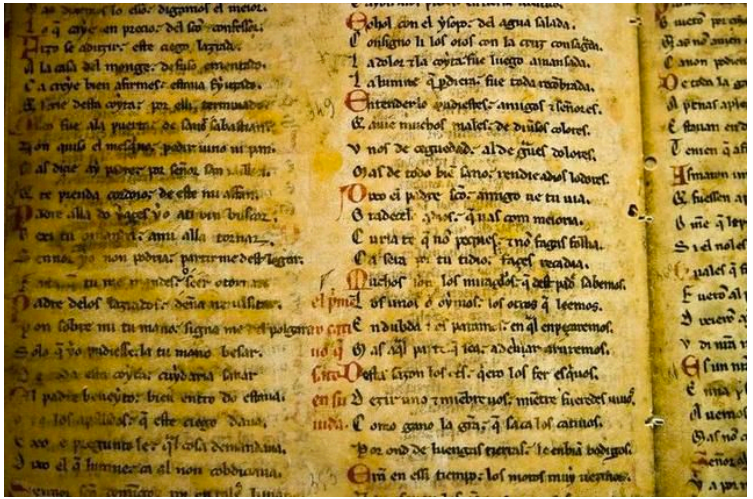
- Natural language is extremely rich in form and structure and very ambiguous
  - How to represent meaning
  - Which structures map to which meaning structures
- One input can mean many **different** things
  - “好棒”, “好棒棒”
- Many input can mean the **same** thing
  - 安安, 你好

Knowledge of Language

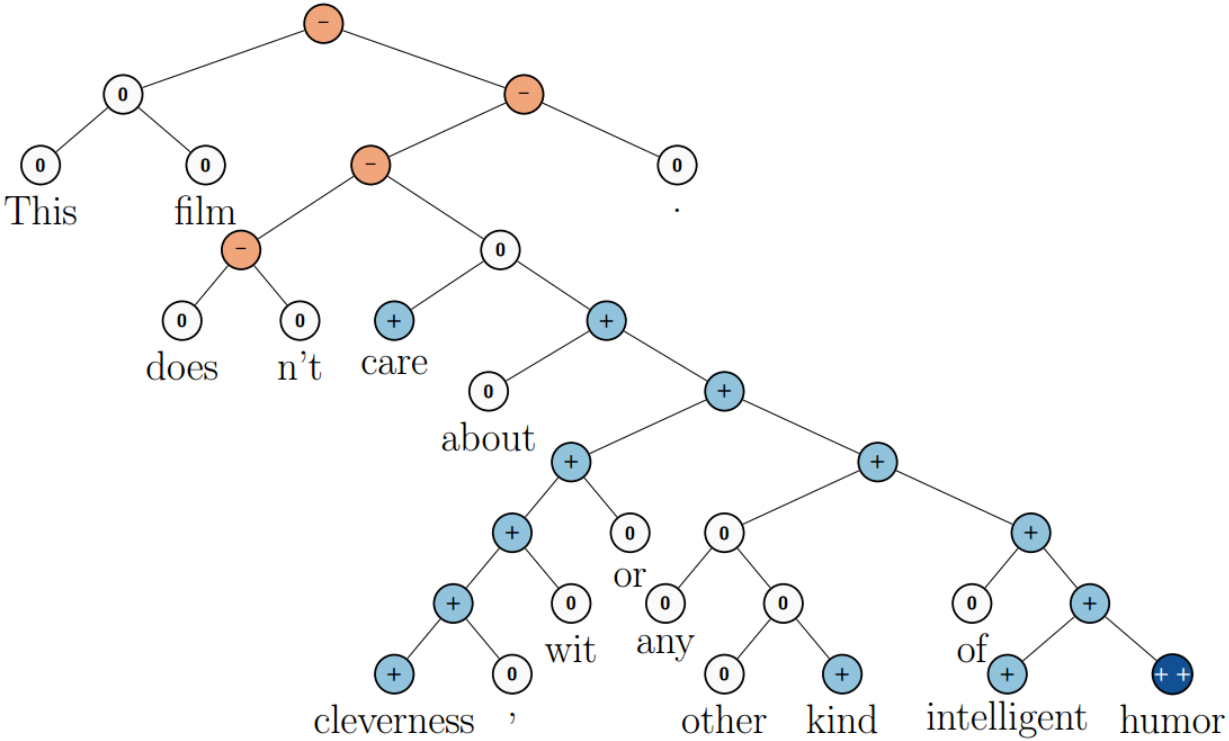
- Phonology (音韻學)
  - how words are related to the sounds
- Morphology (詞法學)
  - how words are constructed from more basic meaning units
- Syntax (文法)
  - how can be put together to form correct sentences and determines what structural role
- Semantics (語意)
  - what words mean and how these meaning combine in sentences

Models to Represent Linguistic Knowledge

- We will use certain models to represent the required linguistic knowledge.
- State Machines
  - HMMs
- Formal Rule Systems
  - Context Free Grammars
- Logic-based models
  - first order predicate logic
- Models of Uncertainty
  - Bayesian probability theory

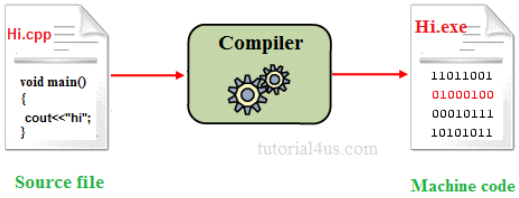


Models to Represent Linguistic Knowledge

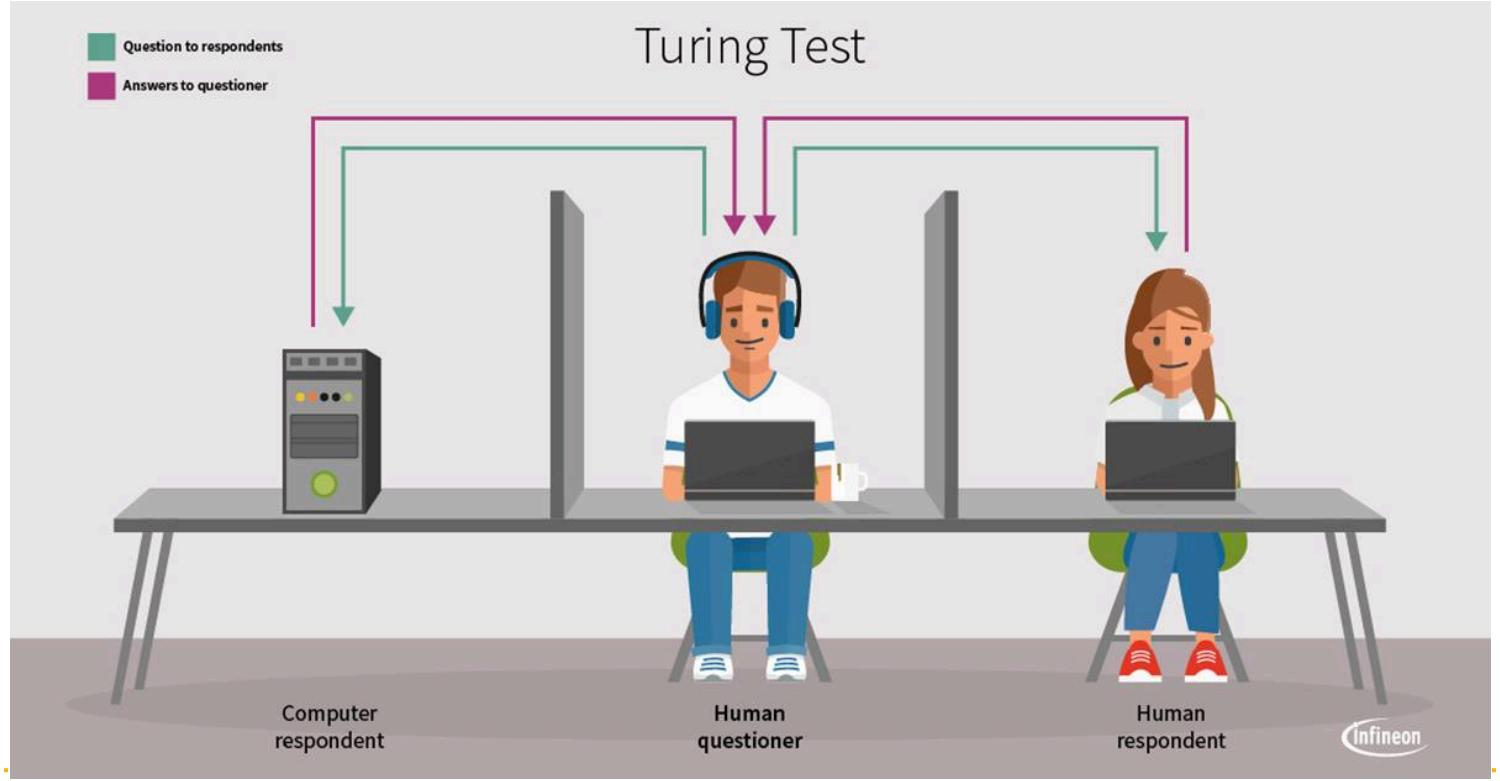


Algorithms to Manipulate Linguistic Knowledge

- manipulate the models of linguistic knowledge
  - to produce the desired behavior
- Most of the algorithms we will study are parsers.
- These algorithms construct some structure based on their input
- The language is ambiguous
  - It is never simple processes



Language and Intelligence



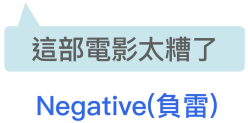
NLP Applications

- Machine Translation
  - 中翻英
- Information Retrieval
  - Google, search engine
- Query Answering/Dialogue
  - Siri, jarvis, chatbot
- Report Generation
  - SCIgen
- Recommendation
  - 文字情感分析

鋼鐵人 jarvis

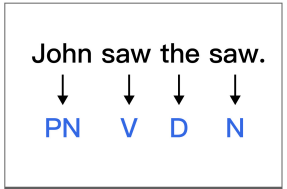
NLP Applications

偵測詐騙郵件

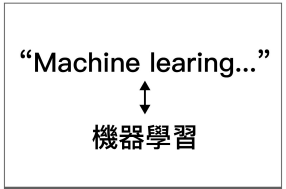


情緒分析

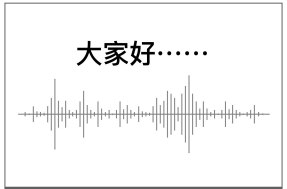
搜尋建議更正



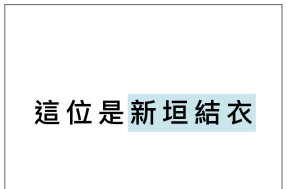
詞類標示



機器翻譯

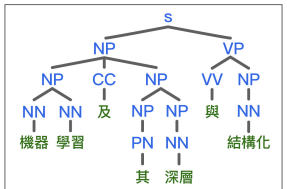


語音辨識



人名辨識擷取

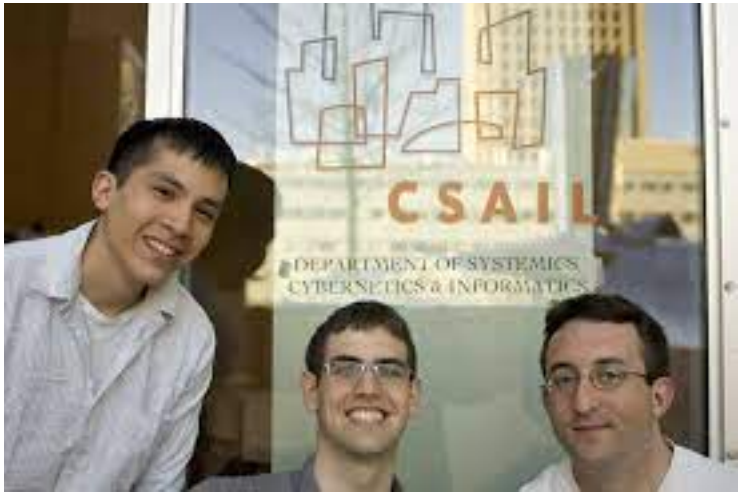
摘要文本大綱



句法分析

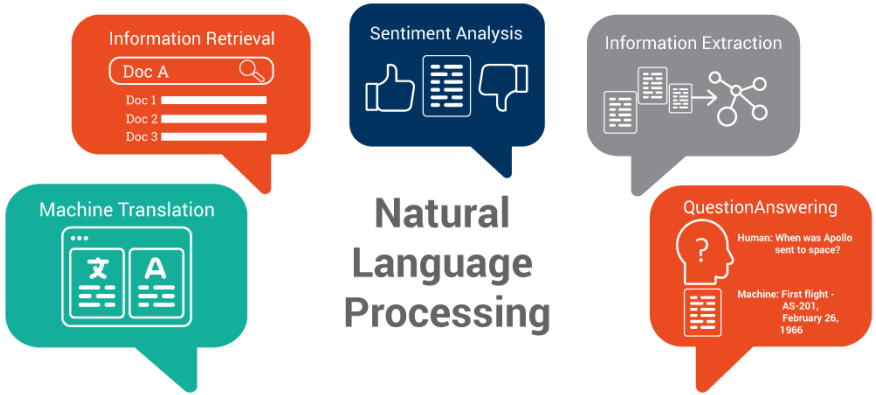
SCIgen-An Automatic CS Paper Generator

- SCIgen is a program that generates random Computer Science research papers
- 2005



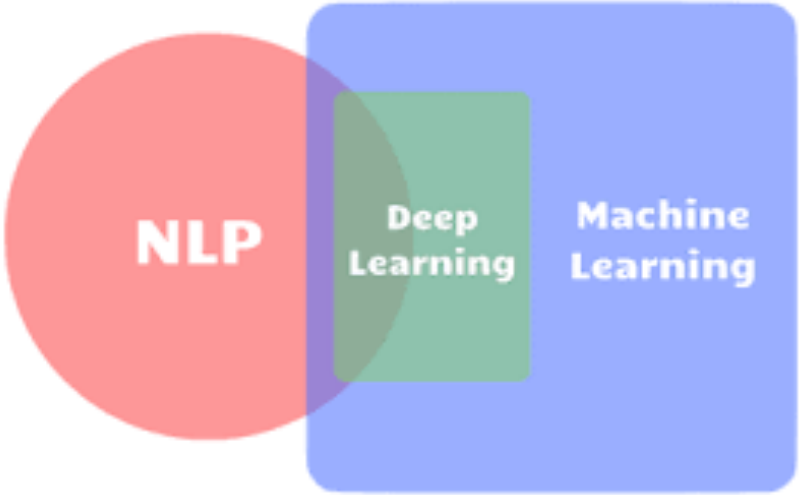
NLP - an inter-disciplinary Field

- 是一門水很深的領域 目前要做得好 滿難的
- 我們不會介紹太深入的東西
- 主要是入門 了解 然後強調快速實作



Two Approaches to NLP

- Machine Learning
  - 處理語料
  - 特徵工程
  - 分類器
- Deep Learning
  - 處理語料
  - 設計模型
  - 模型訓練



先得到要處理的語料

- 好心人已整理好的語料庫
- Wikipedia
- 網路爬蟲
  - 7 / 22
  - 7 / 29
  - 張維元老師會詳細介紹

處理語料

- 英文
  - 分詞 (Tokenization)
  - 詞性提取 (Stemming)
  - 詞性還原 (Lemmatization)
  - 命名實體識別 (Named Entity Recognition)
- 中文
  - 分詞 (Tokenization)
  - 詞性標註 (Parts of Speech)
  - 命名實體識別 (Named Entity Recognition)

分詞 (Tokenization)

- NLP中最基本步驟
- 將句子 段落 分解為 字詞
- 將複雜的句子轉化為數位方式可以表達
- 中文和英文的分詞難度
  - 英文有空格 中文很難 英文相對比較簡單
- 中文需考慮分詞顆粒度問題
- 中文沒有統一標準 XD
- 中文新詞識別問題
- 中文歧義詞切分

中文分詞都還在演進中

中文分詞顆粒度

- 長庚大學
- 長 / 庚 / 大 / 學 (1-gram)
- 長庚 / 庚大 / 大學 (2-gram)
- 長庚大 / 庚大學 (3-gram)
- 長庚大學 (4-gram)

N-gram 如何了解
n應該設定多少
目前也都沒有非常好的結果

中文分詞新詞識別問題

- 藍瘦香菇
- 耗子為汁

中文分詞歧義詞切分

- 兵乓球拍賣完了
  - 兵乓球拍 / 賣 / 完了
  - 兵乓球 / 拍賣 / 完了
- 切完會搞錯意思

中文分詞方法

- 詞典匹配
  - 建造詞典
- 統計方法
  - 考慮字詞的出現頻率
- 深度學習
  - 直接跑模型訓練
  - 目前最準 97%

我們的HW#1

分詞工具

- jieba
  - https://github.com/fxsjy/jieba
  - 最方便的開源工具之1
- pkuseg
  - https://github.com/lancopku/pkuseg-python
  - 最方便的開源工具之2
- CKIP
  - https://ckip.iis.sinica.edu.tw/
  - 繁體中文斷詞的霸主
- HanLp
  - https://github.com/hankcs/HanLP
  - 功能多

Python Jieba

CKIP Chinese Knowledge Information Processing Group

目前中文分詞工具

都是還好

可以用

就是 那樣你知道的

還是需要搭配專業詞典匹配

分詞 詞頻計算

- TF-IDF 演算法
  - 詞頻
    - 一個詞出現在一個文件的頻率
    - term frequency，TF
  - 逆向文件頻率
    - 所有文件中含有這個詞之數量的倒數
    - inverse document frequency，IDF

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{t' \in d} t', d}$$

$$idf_t = log \frac{D}{f_t}$$

TF-IDF 演算法 (經典算法)

- 詞頻
  - 一個詞出現在一個文件的頻率
- 逆向文件頻率
  - 所有文件中含有這個詞之數量的倒數

$$tf_{t,d} = \frac{f_{t,d}}{\sum_{t' \in d} t', d}$$

$$idf_t = log\frac{D}{f_t}$$

把兩個✖起來

$$tfidf(t, d, D) = tf(t, d) * idf(t, D)$$

TF-IDF 演算法例子

- 如果一篇文章中，“ 長庚資工” 出現的次數10次，
- 整篇文章的詞彙總數是200個，
  - 則 tf = 10 / 200 = 0.05
- 如果 “ 長庚資工 " ，在全部1000篇文章中出現40次
  - 則idf = log1000/40 = log25 = 1.398
- 則tf x idf = 0.07

TF-IDF 演算法例子

- 如果一個詞出現的文章數越少，則log值(IDF)會越大，
- 則乘上tf則越大，代表重要

- 反之則越小，若每篇文章都出現，則log1000/1000=0,
- 乘上tf還是會得0．代表不重要

- 透過TF-IDF計算，可以得到在**固定文本**中，單一詞彙的權重數值

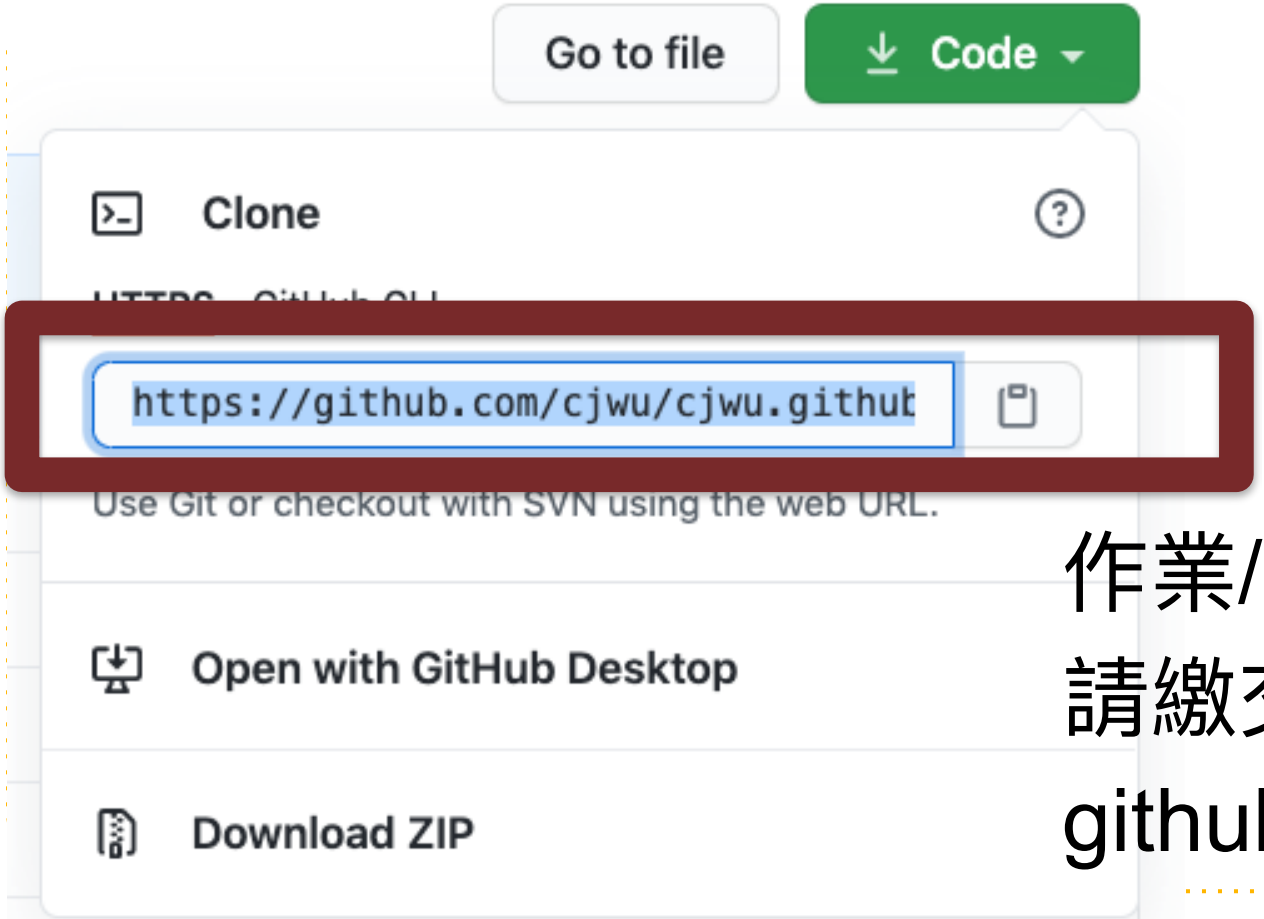
HW#1 詞頻分析

1. 請先使用分詞工具 斷詞
2. 統計前一百個高頻和TF-IDF權重高的字詞
3. 計算並畫出其統計圖型 (2個圖)
    - 3.1. 一行算一個文章
    - 3.2. x軸 字詞編號
      y軸 權重 fig#1和 出現頻率 fig#2
    - 3.3. y軸 要sort過 取前100個

HW#1 詞頻分析

4. 請 Commit ipynb 檔
5. 請記得要commit >= 5次
6. 不要改github branch name
7. 7/22 12:00 前 push github
8. 遲交或補交 扣20分
9. 一個function 不能超過50行 (超過一行扣一分)

HW#1 語料庫 https://github.com/cjwu/cjwu.github.io/raw/master/courses/nlp/hw1-dataset.txt



作業/隨堂練習
請繳交你自己
github上的這個Link

Conclusion

- An introduction to NLP
- Language modeling
- 處理語料
  - 中文分詞
- Slack 私訊我或是助教

Thanks!

Open for any questions

CJ Wu

cjwu@mail.cgu.edu.tw