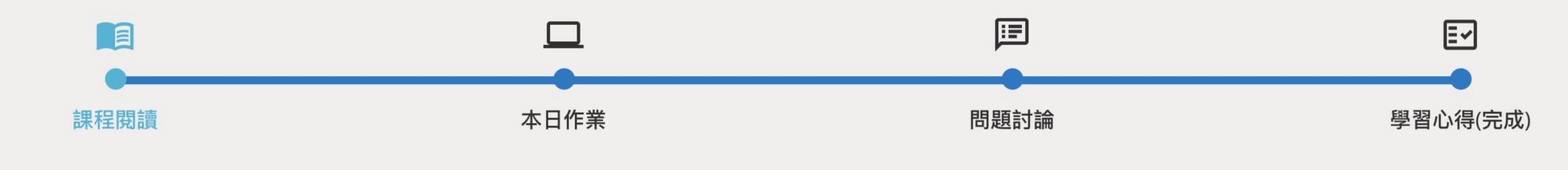
AI共學社群 > Part1 - NLP 經典機器學習馬拉松 > 自製中文選字系統: 進階篇

我的

AI共學社群



自製中文選字系統:進階篇





重要知識點



- 運用以下所學來完成此專題:Python String 基礎操作、斷詞、基礎語言模型
- 學習如何用結巴斷詞並用來改善 Ngram
- 學習 Smoothing of Language Models 來改善中文選字系統

實作提示

請搭配 Jupyter Notebook 使用本教材

Ngram 的選字推薦的問題

範例:木柵動物園











→ 使用斷詞系統改善

使用 Ngram 的問題

有一些 Ngram 組合而成的字並不具有意義

假設使用 Trigram 來做中文字詞的預測會有一些問題

$$P(w_1, w_2, \cdots, w_m) = \prod_{i=1}^m P(w_i | w_{i-2} w_{i-1})$$

• 當 Trigram 預測每個字的機率都是零,也就是說它並沒有看過前面兩個字再搭配一個字的組

(給兩個字預測下一個字)

合,這個時候該怎麼辦?

• 其實,我們可以用 Bigram 和 Unigram 來輔助解決這個問題,我們稱之為 Smoothing of

Language Models,它有兩大類方法,我們接下來逐一介紹。

Smoothing of Language Models

Back-off Smoothing

• $\bar{P}_n = P_n$ if $P_n > 0$ else $a\bar{P}_{n-1}$

Back-off Smoothing

- ullet $ar{P}_n$ 是指平滑化後的語言模型, P_n 是指原來的unigram、bigram ... • 這是一個遞迴的公式
- 舉例:
- $P_1 = unigram(w_i)$; $P_2 = P(w_i | w_{i-1})$; $P_3 = P(w_i | w_{i-1} | w_{i-2})$
- 當 $P_3 = 0$ 但 $P_2 > 0$ 時, $\bar{P}_3 = aP_2$

Interpolation Smoothing

Interpolation Smoothing

• $\bar{P}_n = \lambda P_n + (1 - \lambda)\bar{P}_{n-1}$

• 這是一個遞迴的公式

- ullet $ar{P_n}$ 是指平滑化後的語言模型, P_n 是指原來的unigram、bigram ...
- 舉例:
- $P_1 = unigram(w_i)$; $P_2 = P(w_i | w_{i-1})$; $P_3 = P(w_i | w_{i-1}w_{i-2})$ • 用平滑後的語言模型評估: $\bar{P}_3 = \lambda P_3 + (1 - \lambda)[\lambda P_2 + (1 - \lambda)P_1]$

詳見:<u>完整介紹</u>

實作

- 使用斷詞結果當作 Ngram 的語料庫 • 請讀者們選擇任意一種 Smoothing 方式實作平滑後的語言模型 Back-off Smoothing
- Interpolation Smoothing • 並且調整參數找到你覺得表現最好的語言模型