TextRank 演算法介紹

2019/03/26

蔡岳霖

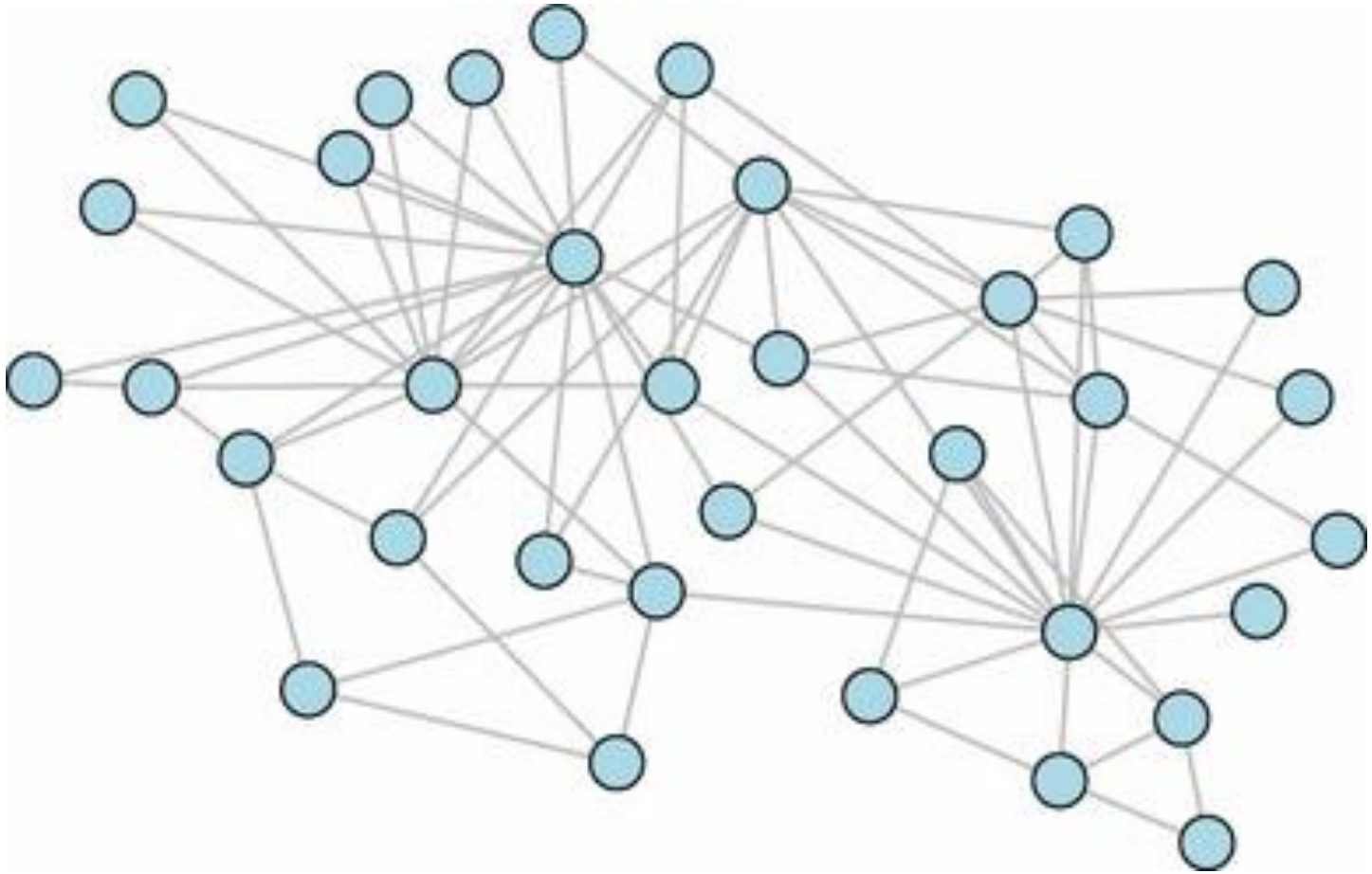
Hackmd連結

TextRank 簡介

TextRank 原始論文

- 受到PageRank啟發
- 以圖為基礎(Graph-based)的演算法
- 可用於”**文章摘要**”與”**關鍵詞萃取**”

TextRank 演算法



TextRank 演算法

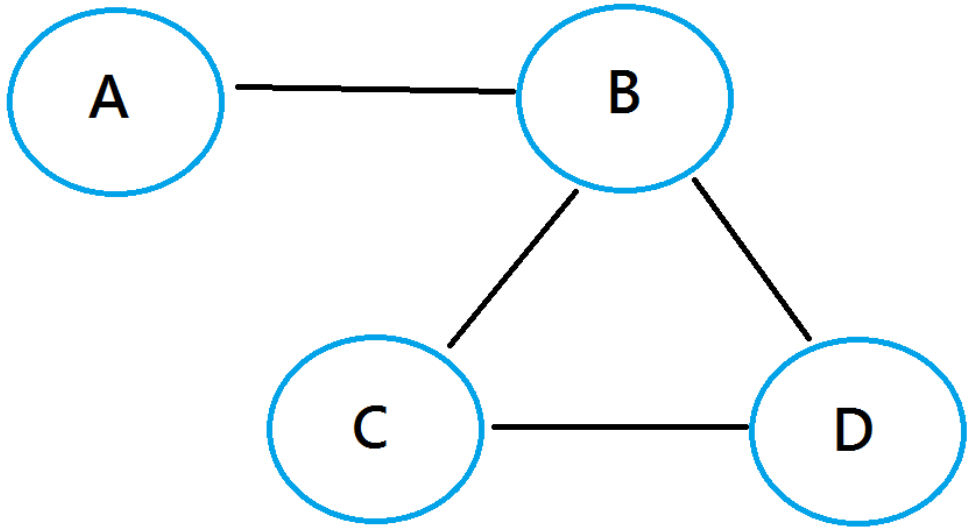
$$h(V_i)=(1-d)+d \bullet \sum_{V_j} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} h(V_j)$$

- V為節點
- $h(V_i)$為某個節點的TextRank分數(原論文使用$WS(V_i)$)
- d為阻尼係數，為定值且介於0~1之間，常被設定成0.85
- w_{ji}為節點之間相連的權重

TextRank 演算法

$$h(V_i)=(1-d)+d \bullet \sum_{V_j} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} h(V_j)$$

範例



$$H(C) = (1-d) + d * \left(\frac{1}{3} * H(B) + \frac{1}{2} * H(D) \right)$$

註：在此假設權重大小皆相同

迭代計算後，每個頂點的分數將會收斂

TextRank 用於文本摘要

- 每個句子作為頂點(vertex)
- 句子與句子之間有邊(edge)連接
- 句子之間的相似程度為邊的權重(weight)大小
- **計算TextRank分數，分數較高的前幾個句子作為摘要**

TextRank 用於關鍵詞萃取

- 每個詞作為頂點(vertex)
- 指定移動窗格大小，相鄰的詞在圖上有邊(edge)作為連接
- **計算TextRank分數，分數較高的前幾個詞為關鍵詞**

連結權重計算方式

- 文本摘要
  - 字詞交集程度(右圖)
  - 最小編輯距離
  - 句子向量的餘弦相似度

$$Similarity(S_i, S_j) = \frac{|\{w_k | w_k \in S_i \& w_k \in S_j\}|}{log(|S_i|) + log(|S_j|)}$$

- 關鍵詞萃取
  - 相同權重
  - 相鄰次數

TextRank implementation

- 以英文語系為主
  - gensim
  - summanlp

- 支援中文文本
  - jieba
  - TextRank4ZH
  - snownlp

相關連結

- 關鍵字提取-TextRank算法：不使用套件的TextRank實作
- 使用TextRank算法为文本生成关键字和摘要：TextRank4ZH原作者的文章
- Use TextRank to Extract Most Important Sentences in Article：以summa套件為基礎實作多語言版本(含中文)
- Keyword and Sentence Extraction with TextRank (pytextrank)：以pytextrank為基礎作些許優化