Sentence ordering problem

NLP領域中，句子排序問題在自動摘要系統、語言生成等領域是重要的一項任務，相關研究發現有邏輯性的排列句子使得閱讀者能夠有效閱讀，但句子如何排序，是依內文事件發生順序、邏輯通順順序，或是其他排序方法，都不是能夠簡單可以達成的，下列以多篇文檔自動摘要句排序，介紹幾項相關方法。

- Chronological ordering[1]
  該方法將多篇文章切割為多個子主題，切割方法有很多種，像是LSA、LDA等，並依照子主題內句子有出現的時間標籤資訊或是該文發布時間進行排序，但大部分文章不包含時間資訊，或是文章發布時間不等於事件發生時間，其適用性較低。

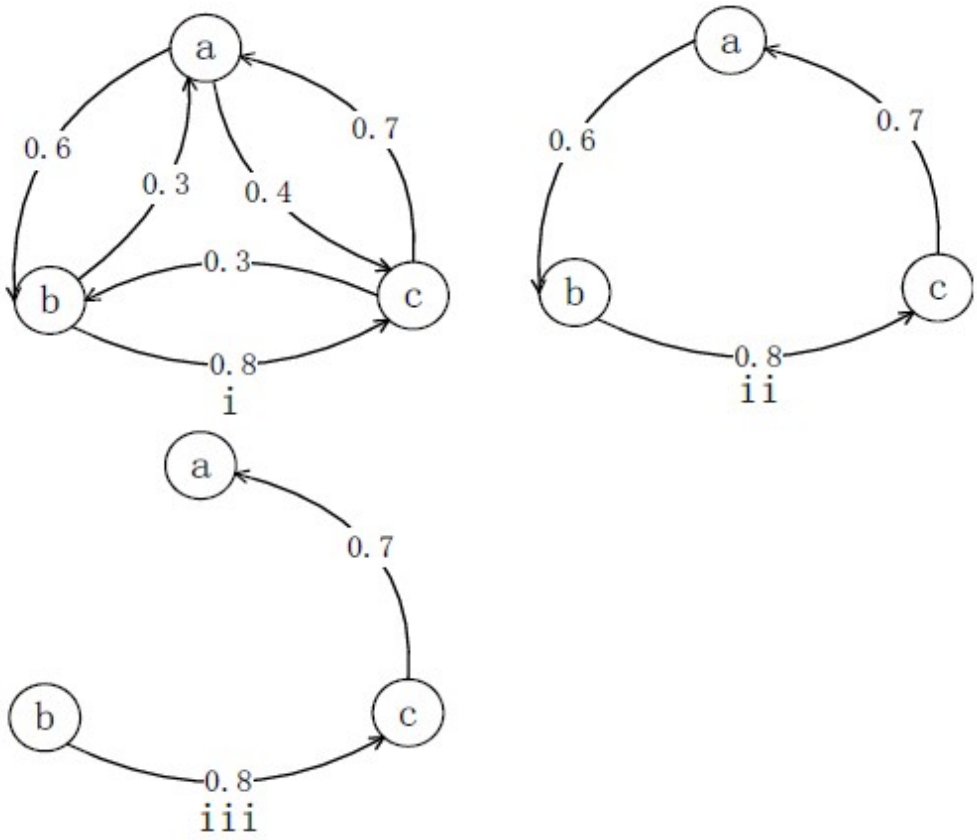
- Majority ordering[1]
  該方法將多篇文章切割為多個子主題，依照主題在不同篇文章中出現的順序進行主題排序，決定主題順序方法為多數決，若大部分文章主題A出現在主題B之前，則最終主題排序結果為主題A排序在主題B之前，決定好主題順序後，將摘要句對應到句子所屬的主題完成排序。該方法依賴於原始文檔中的主題出現的相對位置，若不同文檔中的主題位置變化大，則該方法無法有效排序。

- Machine learning ordering[2]
  該方法將排序問題視為二分類問題判斷兩sentence順序，以人工定義feature方式將句子輸入到分類器裡進行判斷。在選取feature上，以選取能夠判斷出句子位置的特徵，像是時間標間、兩句子相似度、句子上下文相似度、句子在原文中的前後順序等。此方法不是考量原文章的邏輯順序排序，其泛化效果較差，不同領域的資料須分別訓練分類器。

上述三種方法都是基於句子表面資訊，像是時間標籤，句子相對位置、相似度資訊進行排序，而不是基於原始文章邏輯順序，以data-driven方式讓資料自行表述其位置資訊，泛化效果較差。下列介紹以機率方式計算兩句子出現的條件機率進行排序，因為不是人工決定feature，所以不同領域的資料都能夠適用，泛化效果較佳。

- Conditional Entropy ordering[3]
  該方法先定義以機率計算句子順序方法，在句子A出現情況下，出現句子B的機率entropy值，其數學式定義為$H(B|A) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(a_i,b_j)log(p(b_j|a_i))$ ，
  a_i**代表**A**句子中的詞**，b_j**代表**B**句子中的詞** ，
  $p(a_i,b_j)$、$p(b_j|a_i)$**可由原始文章中的每個相鄰句子所統計出** 。當文章中句子A出現在句子B之前，則$H(B|A)$ 值趨近於0，反之句子B是出現在句子A之前，則$H(B|A)$ 會非常大，通過計算每對摘要句子的entropy值，可得出相對的排序。
  entropy值會與句子通順程度呈反比，這裡以反entropy值來代表句子對的通順程度，同時除以entropy值和來讓通順的句子分數能夠更凸顯出來，其數學式為
  $ent(B|A) = \frac{H(A|B)}{H(A|B)+H(B|A)}$ 。
  有了衡量句子排序方法，以下為排序句子的步驟，

1. 將每個摘要句當成node，並以反entropy值計算node間的有向線權重值，下方為示範圖例，圖i為步驟一完成後的狀態。



2. 挑選權重最大的有向線，範例中b->c的線0.8為最大值，此時確定句子b的下一句為c而不是a，並刪除b->a的有向線，而c的前一句為b，刪除a->c的有向線，為避免句子自體循環，刪除c->b的有向線，最後狀態為圖ii。
3. 重複步驟二，直到確定所有句子的排序狀態，圖iii為範例最終狀態，句子順序為b->c->a。

以上為一排序句子方式，句子排序問題較少相關研究，近年類神經網路再度發達，以類神經做為對文章結構塑模方法也一一發表出來，利用類神經排序句子方法也出現不少，有興趣讀者也可參考。

Reference

[1]潜在语义分析聚类算法在文摘句子排序中的应用
[2]A Bottom-up Approach to Sentence Ordering for Multi-document Summarization
[3]Sentence ordering based on conditional entropy and context proximity