Information Retrieval and Extraction – Term Project 2

Team 8

R06922086 林凡煒、r06944026 方珮雯

(agree to share report: YES)

### Division Of Works

林凡煒 (Learning-based Model & Merged Model)

方珮雯 (Rule-based Model)

### Methodology

1. **Learning-based Model**

我們主要目的是將Training Data跟Testing Data中的每筆測試資料，建立一個Feature Array List。針對這些Feature去Learning，採用Soft-max作Multiple Classification，

並用此來強化Rule-base Model的準確度。

Feature:

a. 首先我們會對文本作兩個預先處理：

1. 利用已經斷過詞的文本建立字典
2. 將篩選過後的詞分為九類(Group) ─

**婚配、直系、尊卑、旁系、手足、遠親、師徒、主僕、命令、地點**

b. 我們會利用上述的資訊與Training Data的資料屬性來建立 Feature：

1. Relation Type (Label Class)

2. Term-Frequency Feature

3. Group-Frequency Feature

4. Character Number & Character Last Name

Training & Test Data Transform:

a. 依Input Data 針對每個Row 對文本進行擷取，依擷取範圍可以分為四類：

1. 出現在一句話中 – Weight: 16

2. 出現在三句話中 – Weight: 4

3. 各自出現在文本的第一句話 – Weight: 2

4. 皆沒出現在文本中 – Weight: 1

並依上述擷取出來的詞彙，依擷取範圍乘上該類的權重，並累加至Term - frequency。

同時也累加至Group – frequency。

Result:

最後將產生出的Feature中作Training，最後採用Soft-max 的Multiple Classification，得到每筆Row 產生12個Label: Probability 的Output。

1. **Rule-based Model**

ruleBase是運用紅樓夢文本的特性，建出來的規則模型。步驟如下：

1. 依據文本terms和tf-idf計算結果，建立term weight字典

要判斷人物關係，名詞和動詞扮演關鍵角色。因此，我們從Dream\_of\_the\_Red\_Chamber\_seg 檔案取出Na（普通名詞）, Nc（地方詞）, Nd（時間詞）, V（動詞）四種詞性種類，成為term dictionary的組成元素。

根據學期初學到的資訊檢索原則，term出現的頻率，決定其是否為關鍵字，能不能用他找出目標文件。因此，以tf-idf weighting scheme為原則，找出term dictionary內所有terms的weight，得到term weight字典。

由於本次作業只有單一文本，我們把各段落當成分開的documents，變數說明如下：

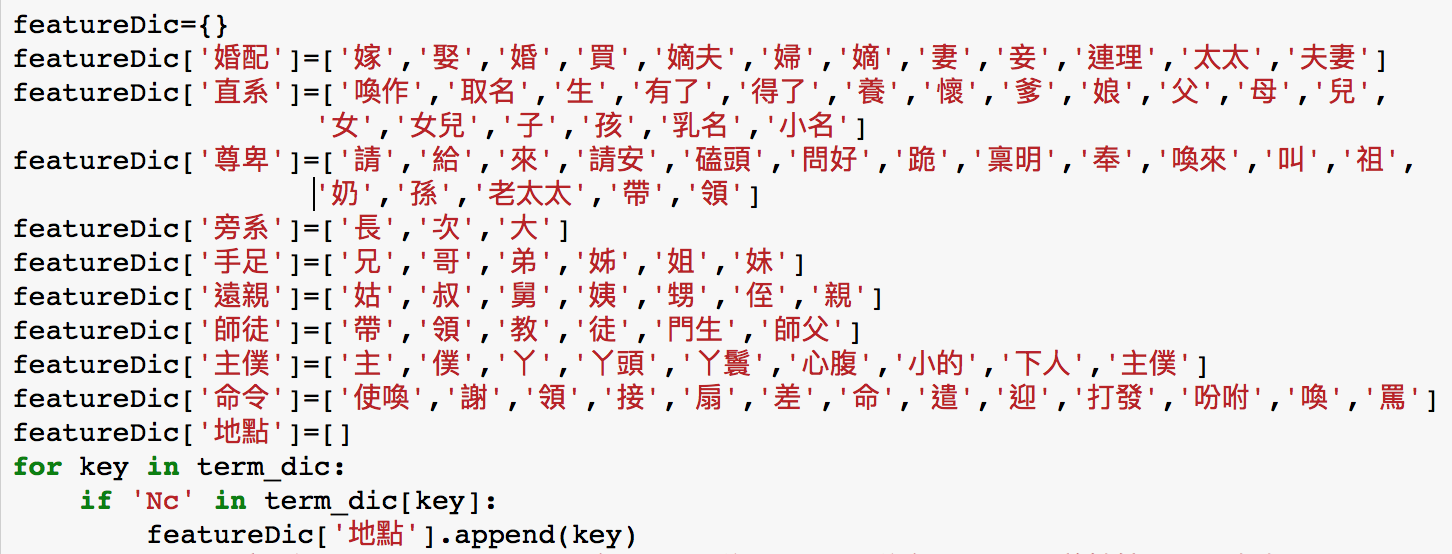
：紅樓夢全文文本總段落數

：term i 出現在幾個段落

：term i在紅樓夢全文文本出現總次數

1. 尋找人物間出現字詞，建立feature字典

將兩個人物出現的距離，分成出現在同一行（中間沒有標點符號，；。？！）、同一句（中間沒有標點符號。？！）、三句內（中間最多只能出現兩次。？！）、同一段四大類，後者並不包含前者。將結果print出，再經過人工觀察兩人姓名附近可能會出現的terms，尋找12種關係分別可能會出現的關鍵字，建立rule-based 特性字典，列舉如下：



feature 字典並非一對一，一個特性可能暗示了一種以上的人物關係：

「婚配」：可能為夫妻關係

「直系」：可能為祖孫、父子、父女、母子、母女之直系血親關係

「尊卑」：可能為祖孫、主僕、夫妻關係

「旁系」：可能為兄弟姐妹關係

「手足」：可能為兄弟姊妹、主僕關係

「遠親」：可能為姑叔舅姨甥侄、遠親關係

「師徒」：可能為師徒關係

「主僕」：可能為主僕關係

「命令」：可能為夫妻、主僕、父子、父女、母子、母女關係

「地點」：只要其中一個字詞為地方詞，必為居處關係，因此無需經過接下來的步驟，從輸入的兩個人物就可以判斷出來。

1. 尋找人物間的term，依人物出現距離設定比重，建立term-weight vector

字詞出現在一行、一句、三句、一段，重要性理應漸減。因此，單一字詞在四種集合出現的次數，需要再根據出現距離近至遠，乘上由大至小的比重，求出加權總和。這個總和乘上前面計算出的weight，就是一個term的特徵值，也是這個rule-based model真正的weight。所有term weights組合成term-weight vector後，就可以進入下一階段。

1. 依據特性規則和出現字詞，建立feature list，判斷人物可能關係

feature list有12元素，元素值代表12種不同關係的可能性。結合前兩步驟，將符合feature字典裡的terms擷取出來，分別乘以term weights再加總，放入可能暗示關係的元素中。元素累加的值最大，代表最有可能是這個關係，這就是rule-based model的最終判斷結果。

1. **Merged Model**

先用Learning-based Model 對每筆Row 產生12個 Label: Probability 的對應。

並設定一個Threshold，若是Max Probability 並沒有超過Threshold，則改用Rule-base Model 產生的結果，來提高整體準確度。

### Experiments

1. **Learning-based Model**

採用Xgboost Library 來協助Training 的過程。

* 1. 針對各種Feature 與 Learning Depth 組合來Training:

|  |  |  |  |
| --- | --- | --- | --- |
| Feature  Max Depth | Term  Frequency | Group  Frequency | Term + Group  Frequency |
| 0 | 0.339 | 0.340 | 0.339 |
| 1 | 0.304 | 0.402 | 0.420 |
| 2 | 0.286 | 0.384 | 0.429 |
| 3 | 0.295 | 0.393 | 0.401 |
| 4 | 0.295 | 0.384 | 0.384 |
| 5 | 0.295 | 0.384 | 0.384 |

1. **Rule-based Model**
   1. 發想feature字典儲存元素

feature字典的字詞是人工找的，本來想嘗試一種關係對應一種feature，但很快就發現紅樓夢裡有許多特例，像是男尊女卑關係，讓夫妻間也有上對下的成份在；大家庭的背景下，感情好的主僕們也會互稱哥哥姊姊，所以才決定讓一種feature對應不只一種關係，希望能讓單一關係有更多考量因子。

feature也不是分得越細越好。比如說，我想區分祖孫和遠親關係，這在文本中常常有兩個人完全不曾出現於同一段的狀況。我嘗試把表現出祖孫關係的名詞從尊卑關係獨立出，卻導致其他直系血親關係被誤認。

* 1. 設定term weight

term weight是最難調控的係數。在我們rule-based model中有三道關卡：tf-idf比重、出現頻率和遠近比重、feature字典比重，三者加權方才呈現最後成果，任何一處改動係數都會造成影響。

tf-idf是我唯一沒動的，因為在後面有feature字典的前提下，就算算出來，也只會挑出重要關鍵字，他們的weight沒有明顯差別，影響也不大。

出現距離遠近比重關乎語句，是重要的一步，所以幾經更改。我們曾經在一行、一句、三句、一段嘗試過等差遞減、指數遞減（8, 4, 2, 1倍）係數，發現後者效果較佳，且一行重要性加倍（16）效果更好。一句或更遠距離的係數更動影響不大，可能遠不如feature terms一開始設定好的比重。

各種feature所佔比重完全沒有標準，就是隨便設一個基準數字，因為最後比的是誰最多，相對關係應該是一樣的。困難的是，在同樣的基準下，字典裡不同feature到底要給多少比重，這完全是在碰運氣。依照明顯程度調配一下，發現差別不大，可能是嘗試的組合不夠多，但我們認為重點應該還是rules不夠完備，中間模稜兩可字詞加太多次，卻沒有把某些關鍵字詞挑出，才會被多次判斷成錯誤的關係。

* 1. 加入字詞後處理規則

後處理規則是輸出結果前最後一道關卡，目的在從兩位人物姓名做最終檢查，把不可能的關係在feature list的值設為0。一個明顯的剔除法是同姓，因為不會是主僕（僕人通常是暱稱），且更有可能是父子、父女、祖孫、兄弟姊妹、姑叔舅姨甥侄、遠親。然而觀察後另外發現，女性若冠夫姓或父姓，同姓仍有可能是母子或母女，這時會出現「姐」, 「母」, 「娘」, 「媽」, 「奶」, 「嬤」等關鍵字，所以在女性偵測上多建立了一個查詢字典，這個字典也方便我們剔除父子、師徒等不會有女性的關係，在測試後增加一點準確率。

1. **Merged Model:**
   1. Merged 後針對 Learning Feature(已採用最佳化的Learning Depth) 與 Threshold 作討論

|  |  |  |  |
| --- | --- | --- | --- |
| Feature  Threshold | Term  Frequency | Group  Frequency | Term + Group  Frequency |
| 0.00 | 0.340 | 0.402 | 0.423 |
| 0.01 | 0.340 | 0.402 | 0.423 |
| 0.02 | 0.340 | 0.402 | 0.423 |
| 0.03 | 0.340 | 0.402 | 0.423 |
| 0.04 | 0.340 | 0.402 | 0.423 |
| 0.05 | 0.340 | 0.402 | 0.423 |
| 0.06 | 0.340 | 0.402 | 0.423 |
| 0.07 | 0.340 | 0.402 | 0.423 |
| 0.08 | 0.340 | 0.402 | 0.423 |
| 0.09 | 0.340 | 0.402 | 0.423 |
| 0.10 | 0.340 | 0.420 | 0.455 |
| 0.11 | 0.340 | 0.456 | 0.455 |
| 0.12 | 0.527 | 0.456 | 0.455 |
| 0.13 | 0.527 | 0.482 | 0.491 |
| 0.14 | 0.527 | 0.482 | 0.527 |
| > 0.15 | 0.527 | 0.527 | 0.527 |

### Discussions

1. **Learning-based Model**

利用Learning方式去計算，我們發現有它的極限。

首先在實作時，我們一開始只有測試將詞頻加進Feature當中，

效果並不是很好。之後將Rule-based的分類方法加進來後，Performance有了將近10%的提升，但整體來說尚未達到Base-Line的標準。

之後無論新增任何Feature，對於Performance的提高幾乎沒有幫助，有時候甚至

頻頻拉低整體的Performance。

根據我們的觀察，在資料量不足的情況下，採用這種Learning的方式效果並不好。

非常容易造成Overfitting的情況。在Learning Depth的設定，也不敢採用過高的

Learning Depth。

1. **Rule-based Model**

目前字典仍然不完備，很多問題不知怎麼克服。比如說，祖孫關係可能不見得有「祖」、「孫」關鍵字，而是用兒子的兒子去呈現，這種邏輯推理關係目前還做不到。此外，姑叔舅姨甥侄也是需要層層推斷出的關係，缺乏明顯特徵，很難在統計結果突出。就連我們人腦馬上能分辨出的父子、父女、母子、母女關係，也因為電腦無從判斷中國古代男性、女性向命名偏好，而難以細分成功。

當親屬關係越遠，要尋找特徵就更難了。不管在同一段與否，中間挾雜太多不相干、甚至干擾判斷的人物和關鍵字，就容易導致祖孫、遠親等被判斷成別的關係。目前在這個rule-based model嘗試在這兩種關係的判斷中，增加「兩人出現第一句」集合、「沒有任何關係最有可能是遠親」等法則，跑的結果反而都變差，除了測資特例外，可能還是因為掌握不到完整的rules、設不出合適的係數。當然，不排除一開始分行、句、段就不太恰當的可能性，這在兩種model都有改進空間。

1. **Merged Model**

整體上我們是構思是分兩個Model一起進行，最後再進行合併。  
在實驗初期，Learning-based Model的確對整體Performance 提供了些許幫助，

可是越作越下去，發現了Learning-based Model有它的極限在。

最後漸漸完全被的Rule-based Model 的 Performance給超越過去。

所以我們將人力全面轉移到Rule-based Model上。

最後得到準確度為 52.67%，略微超過Base Line 的 45%