Sentimental Analysis 正反情緒分析

* Environment

Eclipse Java EE

JDK/JRE v1.6

UTF-8 File Encoding

* Process

pos\_by\_training.txt

training.txt

neg\_by\_training.txt

result.txt

answer.txt

adv.txt

positive.txt

negative.txt

opinion.txt

除了原有的外部情緒字典（正、反、程度詞），利用Training找出其他特別的正反詞彙

之後，根據句中「正反詞彙」出現的數量，搭配加重語氣的程度詞，給定一分數，作為評斷標準

* Frame Work

SentimentAnalyzer.java

9. Start Analyzing

SentimentalDictionary.java

KeyWordFinder.java

TextReader.java

7. Read and Arrange Words into Hash Map

5. Find out Frequent Words

3. Segment Word into Pieces

2. Store

8. Read

FrequencyRecorder.java

6. Append to Dictionary

4. Count Frequency

SegChinese.java

TextReader.java

1. Read

* Details about Training

Step1　斷詞

先由標點符號斷句，並使用Open Source的Library（MMSeg）

實現最大匹配、最大單詞長度的分詞

Reference: <http://function1122.blogspot.tw/2010/10/mmseg4j-java-55.html>

Step2　計算各單詞出現次數（頻率）

得到以「詞」為單位的資料後，計算整份Training Data中，各單詞出現的字數（頻率）

Step3　選擇一些在該類文章中，具代表性的正負面詞彙，加入字典

選擇的標準：SO值 > 3.0，加入Positive字典；SO值 < -3.0，加入Negative字典

* Details about Analyzing

Step1　斷句

以標點、各式符號斷句（不以分行斷句，因為一行視為一則評論或回覆）

Step2　找程度詞

將一個句子切分成小部分，判斷截斷後的詞彙是否屬於Dictionary中的程度詞

如果是，則將該句子的分數倍率乘以2

* Example

「這家旅館的爛服務非常差勁」會切成

「家旅館的爛服務非常差勁」、「這家旅館的爛服務非常差」…「常差」、「非常」…「這」

由長到短、後往前的截字方式（避免長詞關鍵字沒先抓到，反而抓到短詞）

抓到程度詞關鍵字後，會將倍率乘2，並把關鍵詞從句子中刪除

Step3　找正反面情緒用詞

截字、刪字方式同上一步，只是把截斷後的詞彙拿去Positive、Negative Dictionary中比對

比對後，如果是正面詞彙，分數+1，負面則-1（搭配程度詞的倍率，可能變為±2）

Step4　找出Shifter（不、沒）

比對句子中剩餘的字彙，是否包含「不」或「沒」

如果有，則將該句子的分數乘上-1

Step5　判斷整則評論的正反傾向

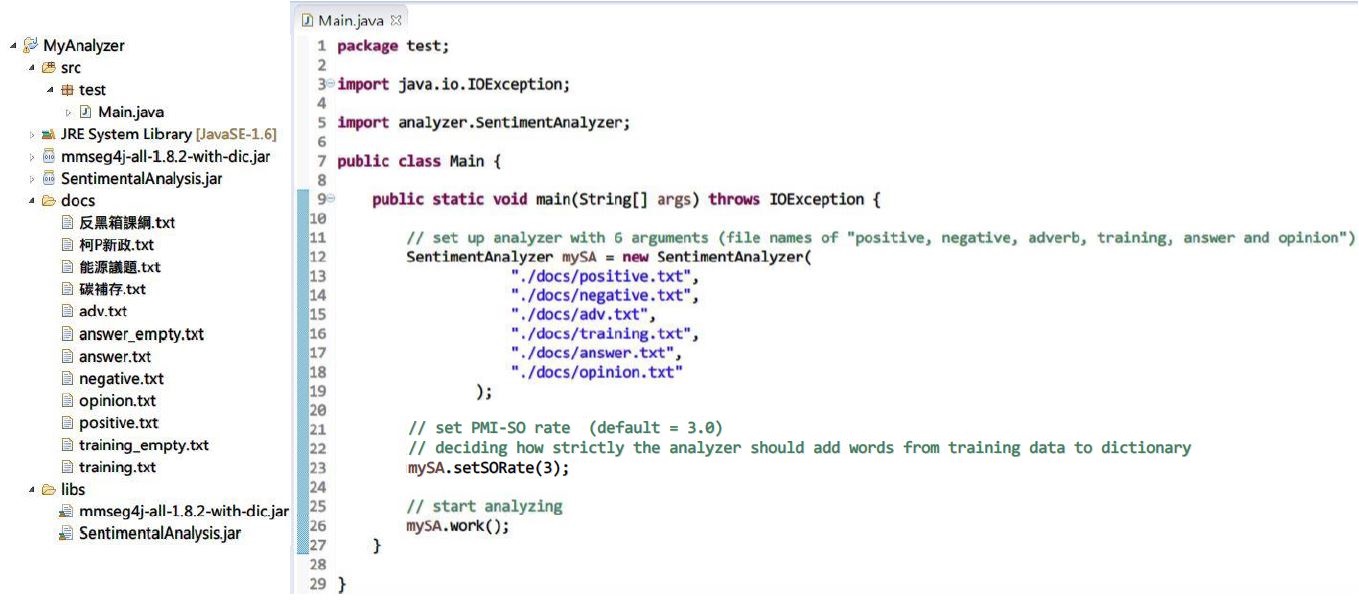
整則評論的分數=各句子的分數加總，若評論分數 ≧ 0，判為正面傾向，反之負面

* API

Source Code已打包成SentimentalAnalysis.jar，外加mmseg4j-all-1.8.2-with-dic.jar

Library Setting好之後，使用SentimentAnalyzer()建構子和method－work()來run

另外，可使用method－setSORate(double)來調整Training時取字的嚴謹程度

多緒方面，method－setNTHREADS(int)可設定Training和Analyzing時的Threads數量

* Input File Format
  + Training用的文字檔 預設為docs/training.txt

一則評論占一行，不加編號（若無，須建立空檔案）

* + Training的答案 預設為docs/answer.txt

行數與training.txt相同，一行一字，以半形大寫P/N來表示（若無，須建立空檔案）

* + 正、反、程度字典 預設為docs/positive.txt, docs/negative.txt, docs/adv.txt

一個單詞（單字）占一行，不加編號

* + 欲分析的評論 預設為docs/opinion.txt

格式與Training的檔案相同，一則評論占一行，不加編號

* Output File Format

分析後會於當前目錄產生result.txt

result.txt中每則評論的分析占4行：

Line1 「NO.%d rate = %d (Positive)」 或　「NO.%d rate = %d (Negative)」

Line2 原評論的斷詞結果

Line3 「Keywords Found: 」+數個「%s(+1、-1或adv) 」，為找到的關鍵字和其意義

Line4 空行

檔尾則會另列此次分析中，正反評論中的前10名關鍵字

* Design of Experiments

1. Training Data與Testing Opinions的搭配
   * + Problem

在現有詞彙固定的情況下，要如何選擇Training Data來搭配，效果才比較好？

不使用Training功能、使用其他領域的資料、使用自己領域的資料，還是綜合各領域？

以下使用「旅館」和「課綱」的資料

組出四種Training Data（包含無），分別對「旅館」和「課綱」進行準確率測試

* + - Variables and Output

|  |  |
| --- | --- |
| Fixed Items | Values |
| Rate of SO-PMI | 3.0 (default) |
| Number of Positive Words | 3648 |
| Number of Negative Words | 11386 |
| Number of Degree-Terms | 202 |
| Positive/Negative in Training Data (Hotel) | 750/750 |
| Positive/Negative in Training Data (Course Guideline) | 369/693 |
| Positive/Negative in Training Data (Hotel + Course Guideline) | 1119/1443 |
| Positive/Negative in Testing Opinions (Hotel) | 750/750 |
| Positive/Negative in Testing Opinions (Course Guideline) | 369/693 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| **Accruacy Table** | | | | |
| **Training Data**  **Opinions** | None | Hotel | Course Guideline | Hotel + Course Guideline |
| Hotel | 82.6%  (1) | 78.8%  (3) | 74.5%  (4) | 80.3%  (2) |
| Course Guideline | 64.6%  (3) | 60.6%  (4) | 69.6%  (2) | 74.0%  (1) |

* + - Conclusion
      * + 使用與Testing Opinions完全無關的資料來Training的話，效果不好

（不論是旅館還是課綱，這種搭法的準確率都是最後一名）

* + - * + 使用「綜合版」來Training，準確率分別拿下第一和第二名

（綜合版的資料較為全面，不僅包含自己的領域，還可擴充萬用正反詞）

* + - * + 若該Training Data中完全沒有該領域的資料，不如不要Training

（旅館就算只使用現有的詞彙，準確率也有82%）

1. SO-PMI的Rate設定
   * + Problem

Training時，會根據SO-PMI的值來決定單詞要不要納入情緒字典中

而SO-PMI的Rate設定得越高，選字的門檻就會越嚴格

選字門檻若變得嚴格，代表情緒字典中字詞的代表性增加，但字詞的總數卻會降低

「字少但高品質」的字典與「字多但品質中庸」的字典，哪一個的效果會比較好？

以下使用五個SO-PMI的Rate值，分別對「旅館」和「課綱」進行準確率測試

（旅館的評論，使用旅館領域的Training Data；課綱的評論，使用課綱領域的）

* + - Variables and Output

|  |  |
| --- | --- |
| Fixed Items | Values |
| Number of Positive Words | 3648 |
| Number of Negative Words | 11386 |
| Number of Degree-Terms | 202 |
| Positive/Negative in Training Data (Hotel) | 750/750 |
| Positive/Negative in Training Data (Course Guideline) | 369/693 |
| Positive/Negative in Testing Opinions (Hotel) | 750/750 |
| Positive/Negative in Testing Opinions (Course Guideline) | 369/693 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | | | |  |
| **Accuracy and Number of New Positive/Negative Words** | | | | | |
| **SO-PMI Rates**  **Data Sets** | NaN | 4.5 | 3.5 | 2.5 | 1.5 |
| Hotel | 82.6%  +0/+0  (2) | 82.9%  +26/+83  (1) | 81.5%  +173/+596  (3) | 78.7%  +1094/+2172  (4) | 66.9%  +2193/+5049  (5) |
| Course Guideline | 64.6%  +0/+0  (5) | 68.7%  +32/+18  (4) | 68.9%  +219/+100  (3) | 71.2%  +1120/+539  (2) | 76.0%  +1221/+3463  (1) |

* + - Conclusion
      * + Rate超過3.5，Training抓不太到什麼字，對分析的影響不大

而Rate低於2.5時，抓到的詞彙會大量增加，顯著影響分析結果

* + - * + 推測現有詞彙和「旅館」的關聯性已經很高了，因此Training的效果不好

反之，「課綱」使用的詞彙與現有詞彙可能存在不少差異

所以Rate調低時，可以抓出更多相關詞彙，使準確率上升

* + - * + 若現有詞彙與分析主題的相關性高，則Rate可設定高一點，以避免雜訊

反之，可將Rate略微調低，多抓一些新詞彙，稀釋原有詞彙的影響力