Text Processing 考点（总结自past papers）

Notebook:	我的第一个笔记本		
Created:	1/26/2020 3:24 PM	Updated:	1/26/2020 6:28 PM
Author:	Alvin		

第一章：SA

1.subjectivity analysis和sentiment analysis的区别：

subjectivity analysis关注评价的是主观还是客观，sentiment analysis关注评价的好坏。

Bing Liu's Model的主要成分：

An **opinion** is a quintuple $(o_j, f_{jk}, so_{ijkl}, h_i, t_l)$, where:

- o_j is a target object.
- f_{jk} is a feature of the object o_j.
- so_{ijkl} is the sentiment value of the opinion of the
- opinion holder h_i (usually the author of the post)
- on feature f_{jk} of object o_j at time t_l.

so_{ijkl} is positive, negative, neutral, or a more granular rating, such as 1-5 stars as in movie reviews.

简单概括为：Oj,物品，fjk是物品的特征，SOijkl是对物品的评价，hi是评价人，ti是评价日期

三个**challenges for SA**：对应上面5个成分，Oj，是Name Entity Recognition，对应SOijkl是Sentiment Extraction，剩下fjk,hi, tl.都是Information Extraction

2.Lexicon-based approach to SA

Rule-based subjectivity classifier:

区别主观和客观，主要看有没有**emotion words**

Rule-based sentiment classifier:
区别评价的正负，主要看正负哪个评价多

Rule-based gradable sentiment classifier:
主要通过计算**emotion words**的分数来评价正负，
计算方法就是把**emotion words**的分数加起来。

- **Negation rule:** 遇到**not**，要**-1**，并且反转分数正负号。
-
- **Capitalization rule:** 遇到大写的，正评价的**+1**，负评价的**-1**
-
- **Intensifier rule:"definitely", "very", "extremely"**等加强副词不改变正负号，只加分，加多少取决于给定的**weight(Intensifier rule)**
-
- **Diminisher rule: "somewhat", "barely", "rarely"**等削弱副词不改变正负号，减分。减多少取决于给定的**weight(Diminisher rule)**
-
- **Exclamation rule:**叹号，同加强副词，不改变符号，加或者减，取决于评价是正还是负。加减多少取决于 **Weight(!!!).** （这个地方**lecture**只有正例子，没有负例子，比如 **too bad!!!**）
-
- **Emoticon rule:**表情包，有单独的分数。

举例：

- **"I am not good today". Emotion(good)= +3,** 因为有**not**，所以要减去**-1**，并且反转正负号，

最后得分为**-2.**

- **“I am GOOD today”**，因为**GOOD**大写，加一分，最后得分为**+4**
- 剩下的几种就是 **Emotion(x) ± weight(y)**。看情况决定。更多例子在**Lecture_SA2.pdf**的**12**页里面有

Lexicon-based approach优点：**effective**，**language independent**，**no require for training**，可以拓展新词。

缺点：需要**lexicon of emotion words**做数据支撑，对新词，缩写，误拼写的词理解不足。

3.Corpus-based/Supervised machine learning

分两步：

- 第一，**Subjectivity classifier**: 运行**binary classifer**，找到并消除客观内容
- 第二，**Sentiment classifier**，学习对不同**attribute**进行打分**weight**，然后**make prediction** 运用比如：**Naive Bayes**

运用朴素贝叶斯**Bayes Rule**：

Likelihood Prior

$$p(Y \mid X_1,\ldots,X_n) = \boxed{\frac{P(X_1,\ldots,X_n \mid Y)P(Y)}{P(X_1,\ldots,X_n)}}$$

Posterior Normalization Constant

- $P(Y)$: Prior belief (probability of hypothesis Y before seeing any data)
- $P(X_1,\ldots,X_n|Y)$: Likelihood (probability of the data if the hypothesis Y is true)
- $P(X_1,\ldots,X_n)$: Data evidence (marginal probability of data)
- $P(Y|X_1,\ldots,X_n)$: Posterior (probability of hypothesis Y after having seen the data)

贝叶斯的本质是用先验**(Prior)**概率去推算后验**(Posterior)**概率

corpus-based approach to SA的分数公式：

数据集：

A Naive Bayes classifier - a worked out example (ctd)

- **Features**: adjectives (bag-of-words)

Doc	Words	Class
1	Great movie, excellent plot, renowned actors	Positive
2	I had not seen a fantastic plot like this in good 5 years. amazing !!!	Positive
3	Lovely plot, amazing cast, somehow I am in love with the bad guy	Positive
4	Bad movie with great cast, but very poor plot and unimaginative ending	Negative
5	I hate this film, it has nothing original. Really bad	Negative
6	Great movie, but not...	Negative
7	Very bad movie, I have no words to express how I dislike it	Negative

先计算**Priors**：

Priors:

$$P(positive) = count(positive)/N = 3/7 = 0.43$$

$$P(negative) = count(negative)/N = 4/7 = 0.57$$

where N = total training examples

Prior是该类别的在所有样本中出现的数量。按上图，就是有**7**个**DOC**，其中**3**个被划分为**positive**，剩下**4**个被划分为**nagative.**

首先计算**Likelihoods**：

Likelihoods:

$$P(t_j|c_i) = \frac{count(t_j, c_i)}{count(c_i)}$$

这个公式计算的是每个词在每个类别中的出现的概率。

举个例子：**P(amazing|positive) = 2/10**，首先说明的是**postive**这个类中有**10**个**emotion words**（包括重复的），然后**amazing**出现了**2**次。

Final decision

$$\underset{c_i}{\operatorname{argmax}}\, P(c_i) \prod_{j=1}^{n} P(t_j|c_i)$$

Given a new segment to classify (**test time**):

Doc	Words	Class
8	This was a fantastic story, good, lovely	???

$$P(positive) * P(fantastic|positive) * P(good|positive) * P(lovely|positive)$$

$$3/7 * 1/10 * 1/10 * 1/10 = 0.00043$$

$$P(negative) * P(fantastic|negative) * P(good|negative) * P(lovely|negative)$$

$$4/7 * 0/8 * 0/8 * 0/8 = 0$$

So: $sentiment = positive$

要分别对正和负两个类分别计算，哪个分高，就是**sentiment**属于哪一类。

corpus-based approach to SA 额外几点：
1.只要**data**不稀疏（**sparse**），就很好用
2.Prior is very importanta especially on biased cases(原句)
3.如何**improve**，两方面，一方面从**features**特征下手

- **Using all words (in Naive Bayes) works well in some tasks**
- **Finding subsets of words may help in other tasks**
- **Using only adjectives can be limiting. Verbs like hate, dislike; nouns like love; words for inversion like not; intensifiers like very**
- **Pre-built polarity lexicons can be helpful**
- **Negation is important** （原句）

另一方面从**Algorithme**算法下手，使甬**MaxEnt**和**SVM**，比**Naive Bayes**更好。
4.非二分类 **non-binary classification**可以使用 **N-class ordinal classification (N**取决于**grades)**或者**regression algorithm**线性回归算法。

4.Comparative SA:

1.Comparative SA和**Direct SA**的区别：
举例：
Comparative: A比**B**好， **Direct SA**：**A**很好

所以**Comparative SA**跟**Direct SA**不同之处，是**comparative SA**有 **comparative opinions**，而**Direct SA**是**direct opinions**。
总之一句话就是**Comparative SA**有比较，**Direct SA**没有。

2.Bing Liu 4 types of comparative relation：
Gradable: A比**B**的某一特征大或者小
Equative: A和**B**的某一特征相同
Superlative: A比所有都好，或者比所有都差。
Non-gradable comparisons: 非打分的比较，比如：**A**和**B**不一样。

总结：**Naive Bays** 分类器，很好用，适合第一个尝试。朴素贝叶斯假设实际上不存在，尽管如此，还是很好用。

5.SA系统评价方法：

$$\text{Accuracy} = \frac{\#\text{ correctly classified texts}}{\#\text{ texts}}$$

$$\text{Precision Pos} = \frac{\#\text{ texts correctly classified as positive}}{\#\text{ texts classified as positive}}$$

$$\text{Recall Pos} = \frac{\#\text{ texts correctly classified as positive}}{\#\text{ positive texts}}$$

$$\text{F-measure Pos} = \frac{2 * \text{Precision Pos} * \text{Recall Pos}}{\text{Precision Pos} + \text{Recall Pos}}$$

Accuracy正确率：被正确分类（包括正负）个数/总样本
Precision精度：被正确划分为正类的个数/被划分为正类的个数
Recall召回率：被正确划分为正的个数/正类的个数