Jason Ballantyne
13432788

Text Analytics COMP47600
Practical Number 9

11/11/2021

Q1:

a) I have asked three classmates for opinions about their phone. Please see the results of opinions from classmates (opinion holders) about their phone below:
Opinion 1: "Samsung has great screen quality and battery"
Opinion 2: "Battery life on a OnePlus is better than any smart phone"
Opinion 3: "The iPhone camera quality is better than any Android phone"

b) I have asked three different people (raters) to rate the above comments as positive, negative, neutral or can't say. I have assigned a numerical representation to illustrate the raters comments. Please see the representation below.

Positive rating ➔ 1
Negative rating ➔ -1
Neutral rating ➔ 0
Can't say ➔ NA

The results of the rates can be seen below:
Rater 1 ➔[1, 0, 1]
Rater 2 ➔[1, -1, 1]
Rater 3 ➔[1, 0, -1]

The following is a 3x3 matrix representation of the results:

Opinions vs Raters			
	Opinion 1	**Opinion 2**	**Opinion 3**
Rater 1	1	0	1
Rater 2	1	-1	1
Rater 3	1	0	-1

c) Taking the above 3x3 matrix, we will now find the inter-rater reliability between our three raters using Kappa.
Cohen's kappa coefficient (κ) is a statistic that is used to measure inter-rater reliability and also intra-rater reliability for qualitative categorical items. Cohen's Kappa measures the agreement between two raters who each classify N items into C mutually exclusive categories. A simple way to think this is that Cohen's Kappa is a quantitative measure of reliability for two raters that are rating the same thing, corrected for how often that the raters may agree by chance. The formula to calculate Cohen's kappa can be seen here:

$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$

Where p_o is the relative observed agreement among raters,
and p_e is the hypothetical probability of chance agreement. The value for kappa can be less than 0 (negative). A score of 0 means that there is random agreement among raters, whereas a score of 1 means that there is a complete agreement between the raters. Therefore, a score that is less than 0 means that there is less agreement than random chance. The score will always be less than or equal to 1.

Based on Mary L McHugh's article, "Interrater reliability: the kappa statistic", Mary defines interprets Cohen's kappa level of agreement as follows:

Value of Kappa	Level of Agreement	% of Data that are Reliable
0 – 0.20	None	0 – 4%
0.21 – 0.39	Minimal	4 – 15%
0.40 – 0.59	Weak	15 – 35%
0.60 – 0.79	Moderate	35 – 63%
0.80 – 0.90	Strong	64 – 81%
Above 0.90	Almost Perfect	82 – 100%

(McHugh, 2012)

We will now calculate Kappa score. In order to do this, we will take two raters at a time and then carry out this computation for all of the raters. We will be using cohen_kappa_score() from sklearn to help us calculate these scores. Please see the results below:

From the results above, we can see that Rater 1 & Rater 2 are at **Minimal agreement** for all the opinions based off of Mary L McHugh's measure of level of agreement. Rater 2 & Rater 3 are at a level of agreement of **None** for all the opinions. Finally, Rater 1 & Rater 3 are at a **Weak agreement** for all the opinions. Based on this, we can conclude that Rater 1 & Rater 3 have the strongest agreement for all the opinions.

Cohen Kappa Scores	
Inter-rater	**Scores**
Rater 1 & Rater 2	0.39
Rater 2 & Rater 3	0
Rater 1 & Rater 3	0.5

d) To get correlation between raters, I would implement Pearson's Correlation. Pearson's Correlation Coefficient is a measure of linear correlation between two sets of data which represents how strongly two raters are associated. It returns a value of between -1 and +1. It attempts to draw a line of best fit through the data of two variables, and the Pearson correlation coefficient, r, indicates how far away all these data points are to this line of best fit. The value of r can take a range of values from +1 to -1. A -1 means there is a **strong negative correlation** and +1 means that there is a **strong positive correlation**. A 0 means that there is **no correlation**. The formula for Pearson's correlation can be seen here:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}}$$

We have described what we would to get the correlation between raters. We will now implement Pearson's Correlation to obtain the correlation between raters. To compute Pearson's Correlation, we will use the numpy function corrcoef(). Please see the results here:

Pearson correlation coefficient	
Inter-rater	**Correlation results**
Rater 1 & Rater 2	0.99
Rater 2 & Rater 3	0
Rater 1 & Rater 3	0

Based on the results, after computing Pearson's Correlation coefficient. We can see that Rater 1 & Rater 2 have a very **strong positive correlation**. Rater 2 & Rater 3 have **no correlation** and Rater 1 & Rater also have **no correlation**. After performing Pearson's Correlation Coefficient, we can deduce that Rater 1 and Rater 2 have the strongest correlation between raters.

Q2: Please see the three sentiment lists I have found that are commonly used in previous research:

1. SentiWordNet is an opinion lexicon derived from the WordNet database where each term is associated with numerical scores indicating positive and negative sentiment information (sentiwordnet.isti.cnr.it. (2019), Text Learning Group)
2. Opinion Lexicon by Hu and Liu contains around 6800 positive and negative words (Hu and Liu, KDD-2004) https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon
3. Multi-Domain Sentiment Dataset (Blitzer et al. ,ACL 2007). This dataset contains positive and negative files for thousands of Amazon products and has been used in several papers. https://www.cs.jhu.edu/~mdredze/datasets/sentiment/

I will now select 10 positive and 10 negative words randomly from the following two lists: **List 1**: Opinion Lexicon by Hu and Liu and **List 2**: Multi-Domain Sentiment Dataset. I will evaluate each word, discussing whether it is actually positive/negative. For each one, I will find a sentential context in which it will be interpreted with the opposite valence.

Jason Ballantyne
13432788

Text Analytics COMP47600
Practical Number 9

11/11/2021

Opinion Lexicon by Hu and Liu		
Positive Words		
Word	**Evaluation & whether it is positive/negative**	**Oppositve valence**
applauding	The term applauding is inherently **positive** in my opinion as it is mostly often linked to a moment of happiness or joy	The crowd laughed at the poor woman while jeering and applauding
compassion	The term compassion is **positive** I believe, as it expresses a moment of gratitude for the individual	There is a serious lack of empathy and compassion nowadays
flutter	I think flutter is a bit more ambiguous and can be described as both **positive** and **negative**. In moments of fear and excitement the verb flutter can be used to describe the situation.	In absolute fear, my heart began to flutter.
geeky	Again, I would evaluate geeky to be both **positive** and **negative.** It can be seen as a positive asset but it also subject to scrutiny eg. Typical	He can't play sports, he's too geeky
keen	Keen can be evaluated to be **positive** and **negative**. Being keen can be positive but being too keen can be a negative.	That person seems overly keen to get involved.
low-risk	Low-risk is mostly **positive.** I believe there are much more occurances where low-risk is a good thing rather than bad.	The venture capitalist disapproved of the idea as he felt it was too low-risk.
recover	Recover is a **positive** word. It is mostly used to describe the opposite of a negative occurance. Eg. He is recovering from a foot injury.	After the attack, the woman said it would take some time to recover.
rich	Rich is a **positive** word, it is described as having a plentiful supply of something. Eg. Rich with happiness.	The thieves were now rich after their horrendous escapade
swift	Swift is **ambiguous** and is more used as an adjective, an attribute of a noun and it depends whether than noun is positive or negative.	Robbers were described as notorious and swift in their actions
wholeheartedly	Wholeheartedly is mostly **positive**. It describes how someone puts their whole heart into doing something, which is **positive**	I wholeheartedly disagree with your opinion.
Negative Words		
broke	I believe broke is **negative**. It is mostly used in a context where there is pain or a negative connotation.	Thank goodness for the sun, it has broke up the day nicely.
bulkier	Bulkier can be **positive** or **negative**. It is just describing the size of something. That thing is open to interpretation whether it is good or bad.	This bag of sweets feels bulkier than normal
cheap	Cheap is **ambiguous**. Something looking cheap can be **negative**, however something being cheap can be a **positive**	I cannot believe this deal, it's so cheap!
demolished	Demolished is mostly negative. Is refers to something being destroyed which usually relates to a bad occurance.	You must have really enjoyed that dinner, you demolished it
deny	Deny is a **negative** word, it refers to a negative action.	I can't deny it, I really do enjoy the sun
lengthy	Lengthy is **ambiguous**, it can be either **positive** or **negative**. It depends on what the word being describes as lengthy is	That movie was nice and lengthy
pollution	Pollution is a **negative** word. The word relates to harmful materials in the atmosphere causing further negative results.	Ireland has much less polution in rural areas.
rigid	Rigid can be **positive** or **negative**. Rigid can be used to describe a person which is negative but to describe an object, it might be positive.	This chair is sturdy and rigid
unorthodox	Unorthodox can be evaluated to **positive** or **negative**. It refers to just out of the ordinary which can be good or bad.	That man's talent is very unorthodox
limited	Limited is mostly **negative**. If you're talking in terms of supply, there is a small amount available which is negative.	I'm so happy I got these shoes, they were limited edition

Jason Ballantyne Text Analytics COMP47600 11/11/2021
13432788 Practical Number 9

Multi-Domain Sentiment Dataset		
Positive Words		
Word	**Evaluation & whether it is positive/negative**	**Oppositve valence**
care	I would evaluate the word care to be **positive.** It is mostly used to show empathy to another individual	I don't care, it's none of my business
essentials	I think the word essentials is **positive**. It correlates to objects or things that are important and bring happinness	These are not part of my essentials, I don't want them
helpful	I think the word helpful is **positive**. It is a kind word used to help out others	The lady behind the counter was not very helpful
favorite	Favourite is a **positive** word. It can be used to associate with things that bring joy to the individual	These are my least favourite shoes
effort	Effort can be **positive** or **negative**. The word can mean that something is too difficult to do but also worth doing	The man did not put in a lot of effort
anticipated	I would evaluate anticapted to be **positive**. It is a word that reflects joy that a person may feel leading up to an event	The show was highly anticipated, but failed to meet expectations
society	Society is **ambiguous.** This can be interpretted as a **positive** or **negative** society.	Our society is corrupt
advice	Advice is mostly **postitive.**It related to a person attempting to lend a hand to another person	I don't want to receive negative advice
events	events can be **positive** or **negative**. We have historicially had both **positive** and **negative** major events in history for example.	That really was a series of unfortunate events
completion	I believe completion is mostly **postitive**. It related to a task being completed which is a positive aspect.	This task is nowhere near completion
Negative Words		
bias	Bias is inherently a **negative** word. A bias favours one thing or another which is a bad thing	Positive bias refers to the human tendency to overestimate the possibility of positive (good) things happening in life or in research
gaps	gaps can be **ambiguous.** You may see positive or negative gaps in everyday life.	We were able to capitalise on the situation due to sufficient gaps in the market
consequences	I would evaluate consequences to be **negative.** Consequences generally happen as a result of a negative event.	A positive consequence will increase the frequency of positive behavior.
banned	Banned is a **negative** word. Banned refuses or rejects individuals doing certain things which is inherently bad	Thank goodness the robbers were banned from the store
little	Little is a **negative** word. It relates to something that isn't enough, which is inherently bad	Sometimes a little of something is a good thing
twindled	I believe twindled is a **negative** word. It refers to the reduce of something which is negative.	Her fears of the night twindled as she got older
slow	Slow is a **negative** word also. It has more associations with the negative rather than positive.	Slow and steady wins the race
disturbing	Disturbing is a **negative** word. It generally correlates with fear and other negative connotations.	I'm going for a great sleep, don't disturb it
unsupported	Unsupported is a **negative** word. It is the opposite of supported, which is positive	She acomplished her goals on her own, unsupported by anyone else.
decomposing	Decomposing relates to death and is therefore a **negative** word in my opinion.	Decomposing food is a great feritilizer for soil

Q3:

a) Bromberg's Sentiment Program is a function that takes a feature selection mechanism and returns its performance in a variety of metrics. We can see that there are print statements added to print the results of metrics including **accuracy**, positive and negative **precision**, positive and negative **recall**.
We have now run the program, please see the results to the right:

```
using all words as features
train on 7998 instances, test on 2666 instances
accuracy: 0.77344336084021
pos precision: 0.7881422924901186
pos recall: 0.7479369842460615
neg precision: 0.7601713062098501
neg recall: 0.7989497374343586
Most Informative Features
             engrossing = True              pos : neg    =     17.0 : 1.0
                  quiet = True              pos : neg    =     15.7 : 1.0
               mediocre = True              neg : pos    =     13.7 : 1.0
              absorbing = True              pos : neg    =     13.0 : 1.0
               portrait = True              pos : neg    =     12.4 : 1.0
                  flaws = True              pos : neg    =     12.3 : 1.0
              inventive = True              pos : neg    =     12.3 : 1.0
             refreshing = True              pos : neg    =     12.3 : 1.0
           refreshingly = True              pos : neg    =     11.7 : 1.0
                triumph = True              pos : neg    =     11.7 : 1.0
```

Based on the results, we receive a **77%** accuracy score as well as a **79%** pos precision score, **75%** pos recall score, **76%** neg precision score and **80%** neg recall score. All of these percentages have been rounded to two decimal places.
I think what might happen when I remove the stop words is that it should improve the performance of a model. Removing these stop words becomes a lot more useful when we start using longer word sequences as model features.

b) We have implemented the **removal of the stop words**, please see the new results of Bromberg's Sentiment Program below:

```
using all words as features
<class 'list'>
<class 'list'>
train on 7998 instances, test on 2666 instances
accuracy: 0.7625656414103525
pos precision: 0.7619760479041916
pos recall: 0.7636909227306826
neg precision: 0.7631578947368421
neg recall: 0.7614403600900225
Most Informative Features
              engrossing = False          pos : neg    =     17.0 : 1.0
                   quiet = False          pos : neg    =     15.7 : 1.0
                mediocre = False          neg : pos    =     13.7 : 1.0
               absorbing = False          pos : neg    =     13.0 : 1.0
                portrait = False          pos : neg    =     12.4 : 1.0
               inventive = False          pos : neg    =     12.3 : 1.0
                   flaws = False          pos : neg    =     12.3 : 1.0
              refreshing = False          pos : neg    =     12.3 : 1.0
                 triumph = False          pos : neg    =     11.7 : 1.0
            refreshingly = False          pos : neg    =     11.7 : 1.0
```

We have also implemented another solution which was to increase the size of the training set to 80% and remove the stop words, please see the results below:

```
using all words as features
train on 9596 instances, test on 1068 instances
accuracy: 0.7865168539325843
pos precision: 0.7771739130434783
pos recall: 0.8033707865168539
neg precision: 0.7965116279069767
neg recall: 0.7696629213483146
Most Informative Features
                    flat = True           neg : pos    =     21.7 : 1.0
              engrossing = True           pos : neg    =     20.3 : 1.0
                mediocre = True           neg : pos    =     15.7 : 1.0
                 generic = True           neg : pos    =     15.0 : 1.0
                    loud = True           neg : pos    =     14.3 : 1.0
                 routine = True           neg : pos    =     13.7 : 1.0
              refreshing = True           pos : neg    =     13.7 : 1.0
                  boring = True           neg : pos    =     13.3 : 1.0
               inventive = True           pos : neg    =     13.0 : 1.0
              disturbing = True           pos : neg    =     13.0 : 1.0
```

To compare the precision and recall results, please see the table below.

	Positive Precision	Positive Recall	Negative Precision	Negative Recall
Raw Data	0.79	0.75	0.76	0.8
Removing Stop Words	0.76	0.76	0.76	0.76
Removing Stop Words & Increasing training set	0.78	0.8	0.8	0.77

Based on the results for precision and recall above, we can see that our initial theory was right in that our model would improve since the data is cleaner after removing stop words. As we can see the positive recall increased from 0.75 to 0.76 then eventually 0.80. However, we can see that positive precision decreased its accuracy overall when it went from 0.79 down to 0.78. We could say that for this feature, our stop words were possibly too expansive and caused this decrease. Overall, we can say that the removal of stop words does not show any negative consequences on the model we train for our task. Removal of stop words definitely reduces the dataset size and thus reduces the training time due to the fewer number of tokens involved in the training.