Comp 550 : Assignment 1

Peter Park

September 28, 2019

Question 1

Every student took a course.

This sentence is not ambiguous because it has only one meaning associated with it. The phrase when interpreted literally has the meaning that each student took one or more class and there is no other way to interpret this sentence.

John was upset at Kevin but he didn't care.

This sentence is ambiguous as the word "he" in the sentence can lead to confusion whether Kevin or John is the one who "didn't care". One interpretation is that John did not care and the other is Kevin did not care whether he was upset. The domain of language that this cause involve in is "Paramatics - Deixis" as there is ambiguity to who the word "he" is referring to. The sort of knowledge that is needed for a natural language understanding system to disambiguate the passage is the clarification of the word "he" whether it is John or Kevin. Also proper use of punctuation could alleviate certain uncertainty in the meaning.

Sara owns the newspaper.

The sentence is ambiguous as the word "newspaper" can mean two different objects. One being the newspaper people read and another being the press associated with "newspaper". The domain of language this cause involves in is "semantics" and "Paramatics - Deixis" because there is ambiguity to which of the "newspaper" the word refers to. The sort of knowledge that is needed for natural language understanding system to disambiguate the passage is the clarification of the word "newspaper" is something to be read or a press.

He is my ex-father-in-law-to-be.

This sentence is ambiguous as the word "ex" and "to-be" are contextually incoherent. "ex" implies that the person referred to by "He" is a former "father-in-law" while "to-be" indicates that this person is also future "father-in-law". Again, the domain of language that this cause involve in is "semantics" and "Paramatics", given by the ambiguity of these two phrases aforementioned. Furthermore, it can be also viewed as "syntax" error as syntactically "ex" and "to-be" should not be written in same phrase describe the subject simultaneously. The knowledge required to disambiguate the passage is whether it is "ex" or "to-be" and two phrases are grammatically antonyms.

ttyl ;) [text message]

This sentence is not ambiguous as there is only one meaning associated with this message. It is only used as a way of wishing farewell or when the person saying this message is leaving somebody else. Given that there is only one meaning associated with this phrase, this is not ambiguous.

Question 2

The Naive-Bayes classifier has following conditional probability distribution.

$$\mathbb{P}(C = j|\mathbf{X} = \mathbf{x}) = \frac{\mathbb{P}(\mathbf{X} = \mathbf{x}|C = j)\mathbb{P}(C = j)}{\mathbb{P}(\mathbf{X} = \mathbf{x})}$$

with $C = \{1, \cdots, J\}$ Given that the prior is a categorical distribution, we have the following

$$\mathbb{P}(C = j) = p_j$$

with $\sum_{j=1}^{J} p_j = 1$. Now we make the following assumption

1. $\mathbb{P}(\mathbf{X} = \mathbf{x}|C = j) = \Pi_{i=1}^{p}\mathbb{P}(X_i = x_i|C = j)$
2. $P(\mathbf{X} = \mathbf{x}) = \sum_{j=1}^{J}\mathbb{P}(\mathbf{X} = \mathbf{x}|C = j)\mathbb{P}(C = j)$

Now we can write the conditional probability distribution function as follows:

$$\begin{aligned}
\mathbb{P}(C = k|\mathbf{X} = \mathbf{x}) &= \frac{\mathbb{P}(\mathbf{X} = \mathbf{x}|C = k)\mathbb{P}(C = k)}{\sum_{j=1}^{J}\mathbb{P}(\mathbf{X} = \mathbf{x}|C = j)\mathbb{P}(C = j)} \\
&= \frac{\Pi_{i=1}^{p}\mathbb{P}(X_i = x_i|C = k)\mathbb{P}(C = k)}{\sum_{j=1}^{J}\mathbb{P}(\mathbf{X} = \mathbf{x}|C = j)\mathbb{P}(C = j)} \\
&= \frac{\Pi_{i=1}^{p}\mathbb{P}(X_i = x_i|C = k)\mathbb{P}(C = k)}{\sum_{j=1}^{J}\Pi_{i=1}^{p}\mathbb{P}(X_i = x_i|C = j)\mathbb{P}(C = j)} \\
&= \frac{1}{1 + \sum_{j:j\neq k}\frac{\Pi_{i=1}^{p}\mathbb{P}(X_i=x_i|C=j)\mathbb{P}(C=j)}{\Pi_{i=1}^{p}\mathbb{P}(X_i=x_i|C=k)\mathbb{P}(C=k)}} \\
&= \frac{1}{1 + \sum_{j:j\neq k} e^{\log(\Pi_{i=1}^{p}\frac{\mathbb{P}(X_i=x_i|C=j)\mathbb{P}(C=j)}{\mathbb{P}(X_i=x_i|C=k)\mathbb{P}(C=k)})}} \\
&= \frac{1}{1 + \sum_{j:j\neq k} e^{\sum_{i=1}^{p}\log(\frac{\mathbb{P}(X_i=x_i|C=j)\mathbb{P}(C=j)}{\mathbb{P}(X_i=x_i|C=k)\mathbb{P}(C=k)})}}
\end{aligned}$$

Now if we let the distribution function of the features to be

$$\mathbb{P}(X_i = x_i|C = j) = \pi_{ij}^{x_i}$$

we get the following:

$$\begin{aligned}
\mathbb{P}(C = k|\mathbf{X} = \mathbf{x}) &= \frac{1}{1 + \sum_{j:j\neq k} e^{\sum_{i=1}^{p}\log\left((\frac{\pi_{ij}}{\pi_{ik}})^{x_i}\frac{p_j}{p_k}\right)}} \\
&= \frac{1}{1 + \sum_{j:j\neq k} e^{\sum_{i=1}^{p} x_i\log\left(\frac{\pi_{ij}}{\pi_{ik}}\right) + \frac{p_j}{p_k}}} \\
&= \frac{1}{1 + \sum_{j:j\neq k} e^{\sum_{i=1}^{p} w_{ik}x_i + b_k}} \\
&= \frac{1}{1 + \sum_{j:j\neq k} e^{\mathbf{w}^T\mathbf{x} + b_k}}
\end{aligned}$$

which shows that if $\frac{p_j}{p_k} = b$ and $w_{ik} = \log\left(\frac{x_{ij}}{x_{ik}}\right)$ then we have the probability function for logistic regression. Given that logistic regression is a linear classifier, Naive Bayes with categorical distribution on features and prior is linear classifier.

For simplicity we look at the example where $C = \{1, 2\}$, which means there are only 2 classes as in standard logistic regression. Then we have the following:

$$\mathbb{P}(C = k|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + e^{\sum_{i=1}^{n} x_i \log \Pi_{i=1}^{p} \left(\frac{\pi_{ij}}{\pi_{ik}}\right) + \frac{p_k}{1-p_k}}}$$

Now if we let $\frac{p_k}{1-p_k} = b$ and $w_i = \log\left(\Pi_{i=1}^{p}\left(\frac{\pi_{ij}}{\pi_{ik}}\right)\right)$ then we have

$$\mathbb{P}(C = k|\mathbf{X} = \mathbf{x}) = \frac{1}{1 + e^{\sum_{i=1}^{n} w_i x_i + b}}$$

Question 3

Please check the file *a1q3.py* for the code. The following is the result from this experiment:

Threshold for infrequent words	Lemmatization				Stemming			
	No Stopwords		Stopwords		No Stopwords		Stopwords	
	$C = 1$	$C = 0.01$	$C = 1$	$C = 0.01$	$C = 1$	$C = 0.01$	$C = 1$	$C = 0.01$
0.0001	0.7479	0.7113	0.7573	0.6826	0.7437	0.7130	0.7616	0.6976
0.001	0.7195	0.6993	0.7280	0.6763	0.7235	0.7042	0.7414	0.6948
0.005	0.6541	0.6499	0.6840	0.6539	0.6843	0.6741	0.7013	0.6698
0.01	0.6169	0.6167	0.6522	0.6368	0.6340	0.6351	0.6661	0.6456
0.05	0.5413	0.5234	0.5757	0.5772	0.5411	0.5746	0.5780	0.5746

Table 1: Table for accuracy for **Logistic Regression** with different parameters

Threshold for infrequent words	Lemmatization				Stemming			
	No Stopwords		Stopwords		No Stopwords		Stopwords	
	$C = 1$	$C = 0.01$	$C = 1$	$\lambda = 0.01$	$\lambda = 1$	$\lambda = 0.01$	$\lambda = 1$	$\lambda = 0.01$
0.0001	0.7280	0.7093	0.7360	0.6926	0.7309	0.7141	0.7320	0.7127
0.001	0.7045	0.6982	0.7138	0.6831	0.7235	0.7024	0.7175	0.7079
0.005	0.6493	0.6488	0.6382	0.6516	0.6775	0.6724	0.6721	0.6777
0.01	0.6161	0.6132	0.6391	0.6368	0.6257	0.6311	0.6417	0.6496
0.05	0.5408	0.5243	0.5698	0.5413	0.5718	0.5718	0.5794	0.5718

Table 2: Table for accuracy for **Linear SVM** with different parameters

Threshold for infrequent words	Lemmatization				Stemming			
	No Stopwords		Stopwords		No Stopwords		Stopwords	
	$C = 1$	$C = 0.01$	$C = 1$	$C = 0.01$	$C = 1$	$C = 0.01$	$C = 1$	$C = 0.01$
0.0001	0.7639	0.7303	0.7738	0.7352	0.7630	0.7272	0.7647	0.7323
0.001	0.7295	0.7249	0.7428	0.7403	0.7597	0.7371	0.7556	0.7553
0.005	0.6564	0.6567	0.6809	0.6820	0.6883	0.6874	0.6905	0.6897
0.01	0.6203	0.6203	0.6845	0.6940	0.6414	0.6413	0.6564	0.6579
0.05	0.5413	0.5246	0.5078	0.5772	0.5001	0.5769	0.5766	0.5769

Table 3: Table for accuracy for **Naive-Bayes** with different parameters

where C for the table for logistic regression / SVM indicates the inverse of penalization coefficient λ for logistic regression / SVM of L_2 norm and indicates the smoothing term for Naive-Bayes classifier. Above, no stopwords indicate removal of stopwords from corpus.

Problem setup

The problem setup was to identify set of 10000 movie reviews consisting of half positive and half negative into its respective categories. In such process, the reviews consisting of raw text was processed and the models such as logistic regression, Linear SVM, Naive-Bayes was used for classification.

Experimental Procedure

The raw text from the textfile was tokenized, meaning raw texts were separated by words. Then the text was processed further to remove stop words. The processed texts were either lemmatized or stemmed after the removal of stop words and the countv-ectorizer was used to convert the text of strings into matrix representing the counts of the words in the text. Finally the three models : logistic regression, linear SVM, and Naive-Bayes was used and their performance in accuracy was reported.

Parameters

ii) The list of parameters are written in the above table. For the regularization/smoothing parameter, 1 and 0.01 was tried to see the affect of high vs low regularization/smoothing. For the threshold for infrequent words, $0.01, 0.005, 0.01$ and 0.05 were tried for the frequency of words to appear in the text. lemmatizaion versus Stemming was also looked at as well as using the stop words. versus not using stop words.

Final Result

We can see from above that Naive-bayes model performed the best with high regularization helping for settings with low threshold for removing infrequent words. Usually removing the infrequent words decreases the performance of model often decreasing as more infrequent words are removed. This can be explained by the fact the models already have regularization/smoothing parameters that already considers infrequent term. For example, in logistic regression and SVM less frequent term are more likely to have less impact on the model's classification performance. Thus, further removing infrequent terms could exceed the optimal settings for the model and instead reducing information the models can use for better performance. Overall, removing stop words surprisingly led to worse performance for most cases and stemming has small improvement in performance over lemmatization for all three models. For example, when we look at exactly identical setting ($C = 1$, stop words, infrequent words threshold 0.001) which gives Naive-Bayes the best performance, we have: 75.73 % for logistic regression, 73.60 % for linear SVM and 77.38% for Naive-Bayes. Hence surprisingly we can see that the best performing model is usually the one with the simplest Naive-Bayes model (no lemmatization, threshold close to 0, and close to no penalty). This could be due to the fact that our features are simple unigrams and not the feature complexity is rather low. Below is the result for best performing model from the parameters listed above:

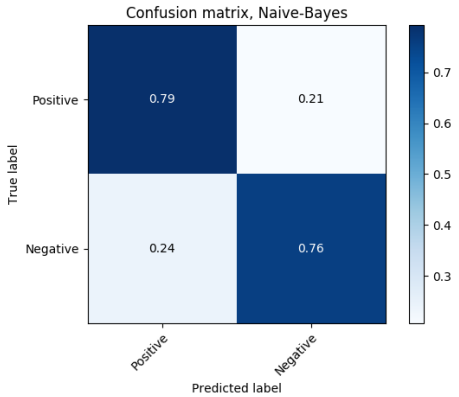

Figure 1: Best performing Naive-Bayes model