Assignment 5:Relation Extraction & Classification

Shyam, Varsani
University of Alberta, varsani@ualberta.ca

Kavit, Gami
University of Alberta, gami@ualberta.ca

1. Introduction

The purpose of this report is to indicate the performance of our Naive Bayes classifier and to report a confusion matrix with precision and recall as in Fig 4.5. Also, report the aggregated (pooled) micro-averaged and macro-averaged precision as in Fig 4.6. This report briefly documents and justifies the decisions we made in our design.

2. Input Data Pre-processing and Decision Justification

Our program follows the bag of words NB classifier implementation that is detailed in chapter 4 of the course textbook. Before pre-processing the input data, we break it into 3 equal sets; train set, dev set and testing set to facilitate 3-fold cross-validation. Next, we clean up the input data by using the NLTK library to pre-process it. We achieve that by tokenizing all the text and converting it to lowercase first. Next, we strip it to get rid of any unnecessary white spaces and use the “re” library to remove any punctuation marks from the text. Lastly, we filter each tokenized document eliminating any stop words in the data. We decided not to use the head and tail indices provided in the input data to pre-process the data any further since we realized that not all documents in the data had accurate head and tail indices. The pre-processed data is now clean and can be classified based on the relation class it belongs to.

For the BOW implementation, we decided to use a dictionary to store the data. The keys of the dictionary are the 4 different relation classes and the values are dictionaries of the documents that belong to that class. These nested dictionaries contain all unique words in the document as keys and their counts in the same document as values.

3. 3-fold cross validation results

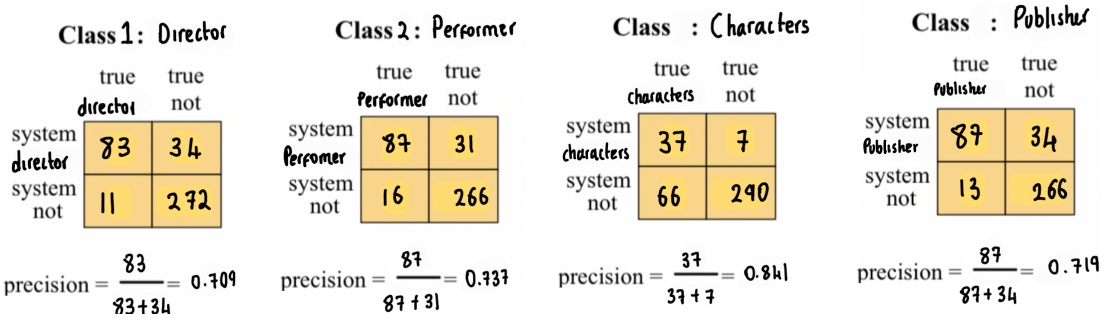
Model Type	**Accuracy**
Model-1 (train: set1, set2) (test: set3)	0.7589
Model-2 (train: set1, set3) (test: set2)	0.7699
Model-3 (train: set2, set3) (test: set1)	0.7661

4. Confusion Matrix

Gold labels

System output	director	performer	characters	publisher	
director	83	5	19	10	$Precision_d = \frac{83}{83+5+19+10}$
performer	6	87	25	0	$Precision_{per} = \frac{67}{6+87+25+0}$
characters	1	3	37	3	$Precision_c = \frac{37}{1+3+37+3}$
publisher	4	8	22	87	$Precision_{pub} = \frac{87}{4+8+22+87}$
	$Recall_d = \frac{83}{83+6+1+4}$	$Recall_{per} = \frac{87}{5+87+3+8}$	$Recall_c = \frac{37}{19+25+37+22}$	$Recall_{pub} = \frac{87}{10+0+3+87}$	

4.1 Separate Confusion matrix for the 4 classes



5. Micro- and Macro- Averaged Precision and Recall

Precision	**micro**	**0.735**
	macro	**0.752**

6. Error analysis

From our confusion matrix we can see that characters were misclassified as directors 19 times. However the opposite is not true since there is only 1 such instance. We also noticed that characters were misclassified as performers and publishers 25 and 22 times respectively. However, performers and publishers were not misclassified as characters as often. The confusion matrix we can see that other classes are also being misclassified but not as often as the characters class since each type of misclassification has less than 10 instances which is lower than those of the characters class. From this information, we can conclude that the class with the most tendency of being misclassified is the character class. This may be happening because the documents belonging to the characters class had many keywords that were also common in the other classes. This is the case because the documents belonging to the character class mostly contain names of characters and those names would be filtered as unknown words. Therefore, the remainder of the keywords would be very similar to those found in other classes. Moreover, the other classes had less misclassifications because most of the documents pertaining to those classes included the class name as a keyword thus making those classes less prone to errors.