COMP 6721 Project 2 Report

Zhongxu Huang[1] and Yixuan Li[2]

[1] 40052560 realdonald9@gmail.com
[2] 40079830 liyixuan4030614@gmail.com

1 Experiment 1

1.1 Baseline experiment

In experiment #1, we will train our model for baseline experiment on the training set. First, we compute the prior for each class and conditional probability for each word in the vocabulary. While computing the conditional probability, the smoothing factor δ = 0.5. Then, we use the prior and the likelihood to compute *overrall_ham* (score(ham)), and *overrall_spam* (score(spam)) on all test set files and label the emails.

1.2 Results and analysis

By analyzing the result, we evaluate the baseline experiment with the test set by these values: Accuracy, Precision, Recall, F_1-measure.

Table 1. Analysis of baseline experiment

Class / Evaluation	Ham	Spam
Precision (P)	394 / 458 = 86.03%	336 / 342 = 98.25%
Recall (R)	394 / 400 = 98.50%	336 / 400 = 84.00%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	91.84%	90.57%
Accuracy of Model	(394+336) / 800 = 91.25%	

For the baseline experiment, the result shows that the model has lower precision for ham but higher recall, while for spam, the model has higher precision but lower recall. Overall, if the beta = 1, meaning that precision and recall have the same importance, then the model has slightly better performance for classifying ham than spam, and the accuracy of the entire model is 91.25%.

Table 2. Confusion matrix of baseline experiment

		Actual class		
		Ham	Spam	Total
Classes assigned by our model	Ham	394	64	458
	Spam	6	336	342
	Total	400	400	800

The confusion matrix shows that the baseline classifier makes more errors for spam and does well in ham emails classification.

2 Experiment 2

2.1 Stop-word filtering

In the experiment of stop-word filtering, we remove the given stop words from the vocabulary and ignore the stop words in emails, because stop words are the words that show up a lot in documents, such as prepositions, pronouns, etc.[1], and it does not help a lot for classification.

First, we compute the conditional probability for each word and use the same smoothing factor ($\delta = 0.5$) and the same prior as in experiment #1. And then, we use the prior and the updated likelihood to calculate the scores and label the emails on the test set.

2.2 Results and analysis

After removing the stop words in vocabulary and experimenting with the classifier, we show the test results in the same way as in experiment #1.

Table 3. Analysis of stop-word filtering experiment

Class / Evaluation	Ham	Spam
Precision (P)	394 / 457 = 86.21%	337 / 343 = 98.25%
Recall (R)	394 / 400 = 98.50%	337 / 400 = 84.25%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	91.95%	90.71%
Accuracy of Model	(394 + 337) / 800 = 91.375%	

Table 4. Confusion matrix of stop-word filtering experiment

		Actual class		
		Ham	Spam	Total
Classes assigned by our model	Ham	394	63	457
	Spam	6	337	343
	Total	400	400	800

The result tables show that the model has a higher recall for ham and higher precision for spam. It means that in stop-word filtering, the model labels well in all emails that are supposed to be ham, but it also has a high error rate (63/457) in ham identification. And the model has high accuracy for all the emails that are labeled by the model as spam, but it missed 63 emails that should be labeled as spam in the entire test set.

And the stop-word filtering has the accuracy of 91.375% overall which is slightly higher than the baseline experiment.

3 Experiment 3

3.1 Word-length filtering

In the word-length filtering, instead of removing the given stop words, we will remove all the words which length is equal or shorter than 2 and words which length is equal or longer than 9 from the vocabulary. And then compute the conditional probability for the rest of the words, and we use the same smoothing factor ($\delta = 0.5$), and prior is also the same as in the previous experiments.

3.2 Results and analysis

After we classify on the emails with the test set, here are the same 2 tables which show the results and performance:

Table 5. Analysis of word-length filtering experiment

Class / Evaluation	Ham	Spam
Precision (P)	393 / 450 = 87.33%	343 / 350 = 98.00%
Recall (R)	393 / 400 = 98.25%	343 / 400 = 85.75%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.47%	91.47%
Accuracy of Model	(393 + 343) / 800 = 92%	

Table 6. Confusion matrix of word-length filtering experiment

		Actual class		
		Ham	Spam	Total
Classes assigned by our model	Ham	393	57	450
	Spam	7	343	350
	Total	400	400	800

As we can see in the first table, the overall accuracy of word-length filtering is 92%, and same as in experiment #1 and #2, it does better in ham classification because the model has 1% higher F_1-measure performance for ham than spam considering both precision and recall.

In confusion matrix, it shows the model misses more spam emails (57) than ham emails (7), but the model has lower error rate for labeling spam emails (7/350) than error rate for ham emails (57/450).

4 Experiment #1, #2, #3 comparison

In this section, we will analyze the result (Accuracy, Precision, Recall, F_1-measure) given by the baseline model, stop-word filtering and word-length filtering. We will show the confusion matrix of 3 experiments in the same table and the performance change table, and then we will compare the performance of the three models.

Table 7. Analysis of experiment #1, #2, and #3

Experiment \ Evaluation		Accuracy	Recall		Precision		F_1-measure	
		model	ham	spam	ham	spam	ham	spam
#1	Baseline	91.25%	98.5%	84%	86%	98.3%	91.8%	90.6%
#2	Stop-word	91.375%	98.5%	84.3%	86.2%	98.3%	92%	90.7%
#3	Word-length	92%	98.3%	85.8%	87.3%	98%	92.3%	91.5%

Table 8. Changes in experiment #2 and #3 compared with experiment #1

Experiment \ Evaluation		Accuracy	Recall		Precision		F_1-measure	
		model	ham	spam	ham	spam	ham	spam
#2	Stop-word	↑ 0.125%	0%	↑ 0.3%	↑ 0.2%	0%	↑ 0.2%	↑ 0.1%
#3	Word-length	↑ 0.75%	↓ 0.2%	↑ 1.8%	↑ 1.3%	↓ 0.3%	↑ 0.5%	↑ 0.9%

Comparing the 3 experiments, stop-word model and word-length model both increase the accuracy of the model, however, the word-length model performs greater improvements than the stop-word model. That means that short-word and long-word removal achieves a better result than stop-word removal.

Furthermore, the word-length model has the highest F_1-measure value for both ham and spam classification out of 3 models, and it significantly increases the recall for spam by 1.8% and precision for ham by 1.3%, although the recall of ham and precision of spam drop slightly. While the changes in the stop-word model are not obvious.

Table 9. Confusion matrix of the first three experiments

		Actual class			
		Experiment	Ham	Spam	Total
Classes assigned by our model	Ham	Baseline	394	64	458
		Stop-word	394	63	457
		Word-length	393	57	450
	Spam	Baseline	6	336	342
		Stop-word	6	337	343
		Word-length	7	343	350
	Total		1200	1200	2400

We can also analyze on the confusion matrix table. All of the models in 3 experiments make more mistakes in spam email identification than ham, but word-length model does the least mistakes because it classifies the emails that were wrongly labeled in baseline experiment and stop-word experiment as spam emails correctly. And this might be the reason why the model in the word-length experiment increases a lot for the recall of spam and precision of ham as shown in table 8.

5 Encountered difficulties and interest on experiment #1, #2, #3

5.1 Difficulty and improvement

Difficulty. Independent assumption
Naïve Bayes classifier model assumes that the attributes are independent with each other. However, his assumption is not always applicable in practical applications. When the number of attributes is large or the correlation between attributes is tight, the efficiency of the classifier model might be inferior to the decision tree model.

To mitigate this disadvantage, we come out with 3 possible improvement ideas as following:

Improvement idea 1: cluster the associated keywords
The above experiments are based on the assumption where each word appears independently occurs. In fact, there is a certain correlation between the occurrence of keywords. If the words with higher correlations are clustered at the beginning, then the simplicity may be used for these related words, and Bayesian model results will be more reasonable.

Improvement idea 2: increase the feature vector
In reality, not only the keywords affect the outcome, but the factors such as the length of the message, the region where the message sender is located should also be taken into considerations. So we personally think a more dimensional feature vector may contribute to improving the effectiveness of the algorithm in the classifier model.

Improvement idea 3: increase the amount of data in the training set
Increasing the amount of data for training is a more intuitive idea. In general, the emails that are used for training is the closest to the real situation. So the larger the amount of data, the better the model should be.

5.2 Interests and inspirations

Interest. While we were searching for materials online, we found various methods for improving the basic model, such as fancy smoothing strategies, using n-grams, etc. But one that interested us was lemmatizing words [2], which is a approach to group together different inflections of the same word, for example “election”, “elections”, “elected” would be grouped together so that these words can be seen as a whole and analyzed as a single item [3]. Indeed, in real life, there may be many inflected forms of a word in an email, if we could develop an efficient lemmatization algorithm for the model, the classifier would be optimized and may achieve better performance.

Inspiration. When we use the confusion matrix to calculate the precision and recall of ham and spam, we can see that the precision and recall of ham do not use the values in spam column in the confusion matrix and vice versa. Hence, it means that it is unused and ignored part of our data for precision and recall calculation. In contrast, when we calculate the accuracy of the model, we take all the data into account.

Therefore, for the problem of email classification, measuring the performance of the model should consider the following point:

- If the cost of missing important emails (a.k.a labeling important emails as spam) is really high, we would rather introduce a parameter that adjusts the model to have higher precision for spam as much as possible and can tolerate lower recall for spam, thus we will be less likely to miss important emails.
- If the categories are balanced in real life, the classifier should identify instances in a balanced manner. If the category is unbalanced, the classifier should tune the model, and we think there are 2 possible ways for doing the tuning: one is changing the loss function and increase the penalty for the classification error with less category, another one is to increase the number of samples in the category with fewer categories.

At last, we personally think the reference value of the average of precision and recall is limited, if precision and recall are particularly low, then F_1-measure should also be low in order to emphasize that a model needs to take both precision and recall into consideration.

6 Experiment 4

6.1 Infrequent word filtering

Removing infrequent words may result in better performance because keywords which occur in lesser frequency in the corpus usually does not play an important role in classification [4]. So by removing those unimportant words, we can increase the frequency of important keywords in order to result in good accuracy.

While Removing the top frequent words might lead to lower accuracy because we guess that higher frequent words are most likely the keywords which are discriminating words, and these words play a significant role in computing scores.

In experiment 4, we will show infrequent word filtering. Firstly, we use a dictionary to store all the words in emails in the training set. Next, we compute frequency for all words, and we gradually remove the words with a certain number of frequency or the words with a certain proportion from the topmost frequent words from the vocabulary. Then, we compute the conditional probability of each word in vocabulary using the same smoothing factor ($\delta = 0.5$). Finally, we use the computed probabilities and the same prior to calculating the score(ham) and score(spam) to classify the emails in the test set.

6.2 Results and analysis

In this section, we will show the same 2 tables that represent the performance of the classifier as before. And we will compare the different word removal strategies and discuss by removing how much proportion based on the word frequency, the performance achieves the best. In the end, we will plot the performance of the classifier against the number of words left in the vocabulary (see Appendix for all line charts of test results).

Table 10. Analysis of infrequent word filtering experiment

Class / Evaluation	Ham	Spam
	Removing words with frequency = 1	
Precision (P)	394 / 456 = 86.40%	338 / 344 = 98.26%
Recall (R)	394 / 400 = 98.50%	338 / 400 = 84.50%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.06%	90.86%
Accuracy of Model	(394 + 338) / 800 = 91.5%	
	Removing words with frequency ≤ 5	
Precision (P)	394 / 454 = 86.78%	340 / 346 = 98.27%
Recall (R)	394 / 400 = 98.50%	340 / 400 = 85.00%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.27%	91.15%
Accuracy of Model	(394 + 340) / 800 = 91.75%	
	Removing words with frequency ≤ 10	
Precision (P)	394 / 454 = 86.78%	340 / 346 = 98.27%
Recall (R)	394 / 400 = 98.50%	340 / 400 = 85.00%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.27%	91.15%
Accuracy of Model	(394 + 340) / 800 = 91.75%	
	Removing words with frequency ≤ 15	
Precision (P)	394 / 453 = 86.98%	341 / 347 = 98.27%
Recall (R)	394 / 400 = 98.50%	341 / 400 = 85.25%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.38%	91.30%
Accuracy of Model	(394 + 341) / 800 = 91.875%	
	Removing words with frequency ≤ 20	
Precision (P)	394 / 453 = 86.98%	341 / 347 = 98.27%
Recall (R)	394 / 400 = 98.50%	341 / 400 = 85.25%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.38%	91.30%
Accuracy of Model	(394 + 341) / 800 = 91.875%	
	Removing words with the top 5%	
Precision (P)	388 / 434 = 89.40%	354 / 366 = 96.72%
Recall (R)	388 / 400 = 97.00%	354 / 400 = 88.50%

F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	93.05%	92.43%
Accuracy of Model	(388 + 354) / 800 = 92.75%	
	Removing words with the top 10%	
Precision (P)	388 / 438 = 88.58%	350 / 362 = 96.69%
Recall (R)	388 / 400 = 97.00%	350 / 400 = 87.50%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.60%	91.86%
Accuracy of Model	(388 + 350) / 800 = 92.25%	
	Removing words with the top 15%	
Precision (P)	388 / 439 = 88.38%	349 / 361 = 96.68%
Recall (R)	388 / 400 = 97.00%	349 / 400 = 87.25%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.49%	91.72%
Accuracy of Model	(388 + 349) / 800 = 92.125%	
	Removing words with the top 20%	
Precision (P)	388 / 439 = 88.38%	349 / 361 = 96.68%
Recall (R)	388 / 400 = 97.00%	349 / 400 = 87.25%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.49%	91.72%
Accuracy of Model	(388 + 349) / 800 = 92.125%	
	Removing words with the top 25%	
Precision (P)	388 / 439 = 88.38%	349 / 361 = 96.68%
Recall (R)	388 / 400 = 97.00%	349 / 400 = 87.25%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	92.49%	91.72%
Accuracy of Model	(388 + 349) / 800 = 92.125%	

Table 11. Confusion matrix of infrequent word filtering experiment

		Removing words with frequency		Actual class		
				Ham	Spam	Total
Classes assigned by our model	Removing infrequent words	= 1	Ham	394	62	456
			Spam	6	338	344
		≤ 5	Ham	394	60	454
			Spam	6	340	346
		≤ 10	Ham	394	60	454
			Spam	6	340	346
		≤ 15	Ham	394	59	453
			Spam	6	341	347
		≤ 20	Ham	394	59	453
			Spam	6	341	347
	R	the top 5%	Ham	388	46	434

			Spam	12	354	366
		the top 10%	Ham	388	50	438
			Spam	12	350	362
		the top 15%	Ham	388	51	439
			Spam	12	349	361
		the top 20%	Ham	388	51	439
			Spam	12	349	361
		the top 25%	Ham	388	51	439
			Spam	12	349	361
		Total		4000	4000	8000

We will analyze the results by two groups:

1) Removing the most infrequent words (the first 5 trials)
   Firstly, we will draw a graph showing the accuracy of the model as removing the infrequent words:

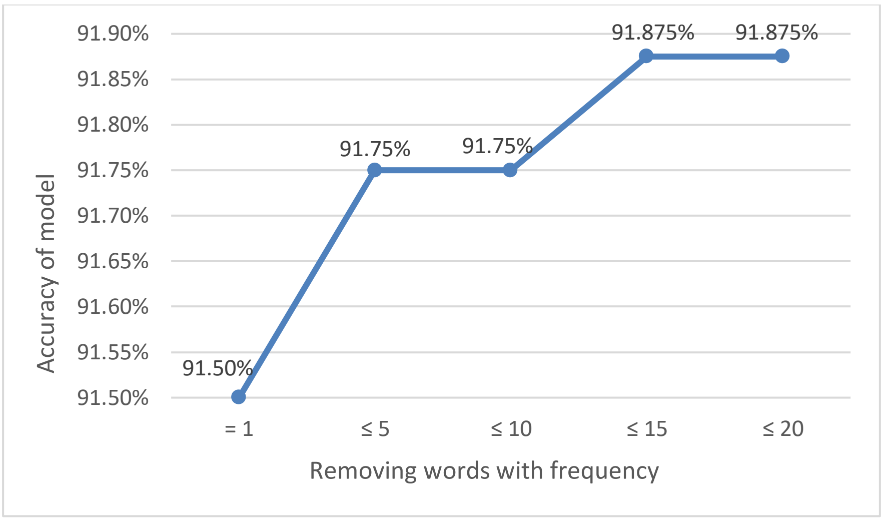


Figure 1. Accuracy with removing words with a certain number of frequency

As we see in Figure 1, the accuracy of the model keeps increasing as we removing the infrequent words from the vocabulary. And the results verify our previous guess "by removing those unimportant words, we can increase the frequency of important keywords in order to result in good accuracy" in section 6.1.

2) Removing the top frequent words (the last 5 trials)
   At first, we will draw a graph showing the accuracy of the model as removing the top most frequent words:

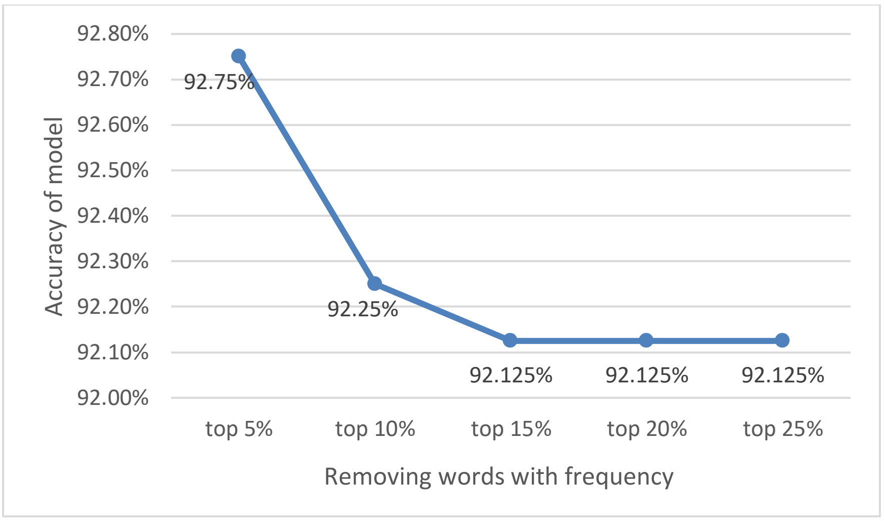


Figure 2. Accuracy with removing words with a certain proportion of the frequency

In Figure 2, we see that the accuracy of the model keeps decreasing as removing the most frequent words from the vocabulary, the reason is that keywords usually appear frequently, so as we removed the most frequent words, we are removing keywords from the vocabulary, thus the important words that count a lot in calculating scores are ignored.

However, one point in Figure 2 that we want to emphasize is the point when we are removing the top 5% frequent words, because it has high accuracy which is even higher than the highest accuracy in Figure 1. The reason to explain this phenomenon is that stop words also have a high frequency as keywords, so when we remove the most frequent words, we are still removing those stop words which does not play an important role in classification, which helps to increase the accuracy of the model.

7 Experiment 5

7.1 Experiment 5: change smoothing

As we have known, for the reason that a word, especially person names, proper noun, etc., which does not appear in any training text but appears in the test set leads to a score of 0. To balance this, smoothing plays an important role in Naïve Bayes classification. Commonly, additive smoothing is usual a component in the model. Most cases, we use add-one smoothing, but in practice, a smaller value is preferred [5].

The goal of smoothing is to increase the zero-probability words to a small probability in order to avoid a never-seen word will lead to the entire process failing [6]. And it not only improves the accuracy of the language model but also accommodates the generation of new words and non-informative words [7].

In this experiment, we will do different smoothing factor trials from no smoothing to $\delta = 1$ by increasing 0.1 at each step. For each step, we calculate the conditional probability for each word in the vocabulary and use them to compute the scores (see Appendix for all line charts of test results).

7.2 Results and analysis

In this section, we will analyze the results and evaluate which smoothing factor suits the best for the model.

Table 12. Analysis of smoothing changing experiment

Class / Evaluation	Ham	Spam
	Smoothing factor = 0	
Precision (P)	400 / 658 = 60.79%	142 / 142 = 100.00%
Recall (R)	400 / 400 = 100.00%	142 / 400 = 35.50%
F_1-measure (β = 1) $(\beta^2+1)PR/(\beta^2P+R)$	75.61%	52.40%
Accuracy of Model	(400 + 142) / 800 = 67.75%	
	Smoothing factor = 0.1	
Precision (P)	394 / 455 = 86.59%	339 / 345 = 98.26%
Recall (R)	394 / 400 = 98.50%	339 / 400 = 84.75%
F_1-measure (β = 1) $(\beta^2+1)PR/(\beta^2P+R)$	92.16%	91.01%
Accuracy of Model	(394 + 339) / 800 = 91.625%	
	Smoothing factor = 0.2	
Precision (P)	394 / 455 = 86.59%	339 / 345 = 98.26%
Recall (R)	394 / 400 = 98.50%	339 / 400 = 84.75%
F_1-measure (β = 1) $(\beta^2+1)PR/(\beta^2P+R)$	92.16%	91.01%
Accuracy of Model	(394 + 339) / 800 = 91.625%	
	Smoothing factor = 0.3	
Precision (P)	394 / 457 = 86.21%	337 / 343 = 98.25%
Recall (R)	394 / 400 = 98.50%	337 / 400 = 84.25%
F_1-measure (β = 1) $(\beta^2+1)PR/(\beta^2P+R)$	91.95%	90.71%
Accuracy of Model	(394 + 337) / 800 = 91.375%	
	Smoothing factor = 0.4	
Precision (P)	394 / 457 = 86.21%	337 / 343 = 98.25%
Recall (R)	394 / 400 = 98.50%	337 / 400 = 84.25%
F_1-measure (β = 1) $(\beta^2+1)PR/(\beta^2P+R)$	91.95%	90.71%
Accuracy of Model	(394 + 337) / 800 = 91.375%	
	Smoothing factor = 0.5	
Precision (P)	394 / 458 = 86.03%	336 / 342 = 98.25%
Recall (R)	394 / 400 = 98.50%	336 / 400 = 84.00%
F_1-measure (β = 1)	91.84%	90.57%

$(\beta^2+1)PR/(\beta^2P+R)$		
Accuracy of Model	(394 + 336) / 800 = 91.25%	
	Smoothing factor = 0.6	
Precision (P)	394 / 458 = 86.03%	336 / 342 = 98.25%
Recall (R)	394 / 400 = 98.50%	336 / 400 = 84.00%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	91.84%	90.57%
Accuracy of Model	(394 + 336) / 800 = 91.25%	
	Smoothing factor = 0.7	
Precision (P)	394 / 459 = 85.84%	335 / 341 = 98.24%
Recall (R)	394 / 400 = 98.50%	335 / 400 = 83.75%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	91.73%	90.42%
Accuracy of Model	(394 + 335) / 800 = 91.125%	
	Smoothing factor = 0.8	
Precision (P)	394 / 459 = 85.84%	335 / 341 = 98.24%
Recall (R)	394 / 400 = 98.50%	335 / 400 = 83.75%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	91.73%	90.42%
Accuracy of Model	(394 + 335) / 800 = 91.125%	
	Smoothing factor = 0.9	
Precision (P)	394 / 459 = 85.84%	335 / 341 = 98.24%
Recall (R)	394 / 400 = 98.50%	335 / 400 = 83.75%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	91.73%	90.42%
Accuracy of Model	(394 + 335) / 800 = 91.125%	
	Smoothing factor = 1.0	
Precision (P)	394 / 459 = 85.84%	335 / 341 = 98.24%
Recall (R)	394 / 400 = 98.50%	335 / 400 = 83.75%
F_1-measure ($\beta = 1$) $(\beta^2+1)PR/(\beta^2P+R)$	91.73%	90.42%
Accuracy of Model	(394 + 335) / 800 = 91.125%	

Table 13. Confusion matrix of smoothing changing experiment

	Smoothing factor		Actual class		
			Ham	Spam	Total
Classes assigned by our model	0	Ham	400	258	658
		Spam	0	142	142
	0.1	Ham	394	61	455
		Spam	6	339	345
	0.2	Ham	394	61	455
		Spam	6	339	345
	0.3	Ham	394	63	457

		Spam	6	337	343
	0.4	Ham	394	63	457
		Spam	6	337	343
	0.5	Ham	394	64	458
		Spam	6	336	342
	0.6	Ham	394	64	458
		Spam	6	336	342
	0.7	Ham	394	65	459
		Spam	6	335	341
	0.8	Ham	394	65	459
		Spam	6	335	341
	0.9	Ham	394	65	459
		Spam	6	335	341
	1.0	Ham	394	65	459
		Spam	6	335	341
	Total		4400	4400	8800

As analyzing the infrequent word filtering, we will show the relationship between the smoothing factor and the accuracy of the model.

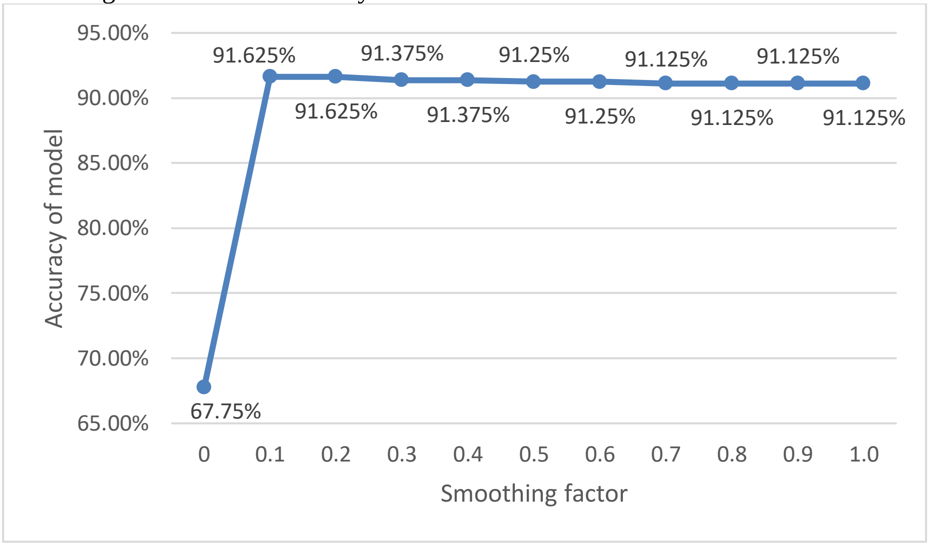


Figure 3. relationships between smoothing factor and the accuracy of the model

In Figure 3, we can clearly see that no smoothing is in the worst situation and smoothing factor δ = 0.1 and δ = 0.2 have the highest accuracy of the model. After δ = 0.2, as the smoothing factor is increasing, the accuracy of the model goes down little by little and stays stable at an accuracy of 91.125%.

8 Comparisons on all 5 experiments

In this section, we will find out the correlations between the 5 experiments. And we compare the corresponding experiments and the results.

First, we found that the experiment #2 stop-word filtering has some relationship with experiment #4 infrequent word filtering to some extent. Because stop words such as language stop words (e.g. “a”, “the”, “on” etc.), time stop words (e.g. “January”, “February” etc.) are the most commonly used words in a corpus [4]. And as we discussed in experiment #4, when we remove the top 5% most frequent words in the vocabulary, it did help the model to be more accurate. We will draw a graph that shows the accuracy of the model in stop-word filtering and infrequent word filtering:

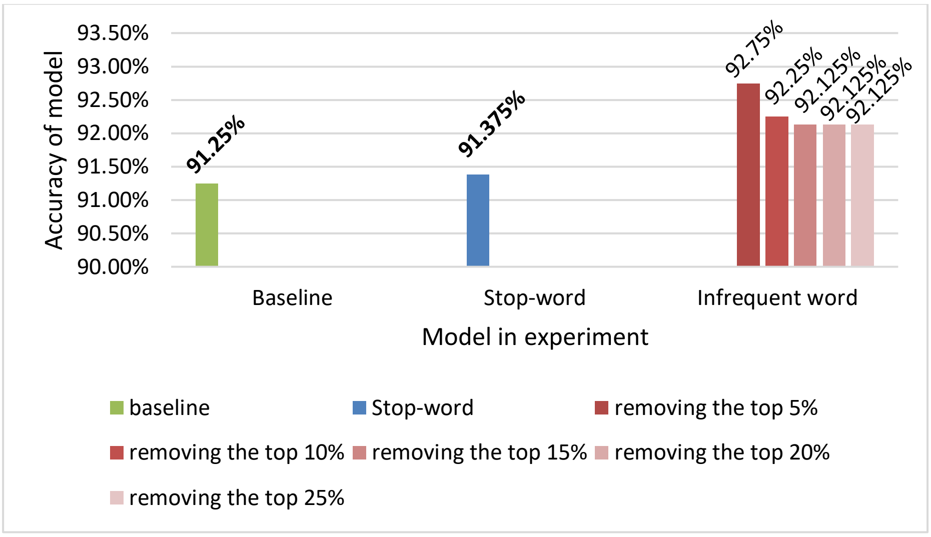


Figure 4. accuracy of models in baseline, stop-word and infrequent word experiments

As we can see in the graph, the accuracy of the model in infrequent word filtering are all higher than stop-word filtering, it means that a lot of the most frequent words are not discriminating for classification, and less discriminating words occupy a large part of all textual words. The reason for this phenomenon is that most of the language is redundant and structurally auxiliary. We personally consider that in each language, there should be a threshold which roughly indicates how much proportion of the top most frequent words in that language are undiscriminating words in all textual words.

Secondly, we will compare the experiment #5 with the baseline experiment. As usual, we will draw the graph to show the accuracy of models in the two experiments.

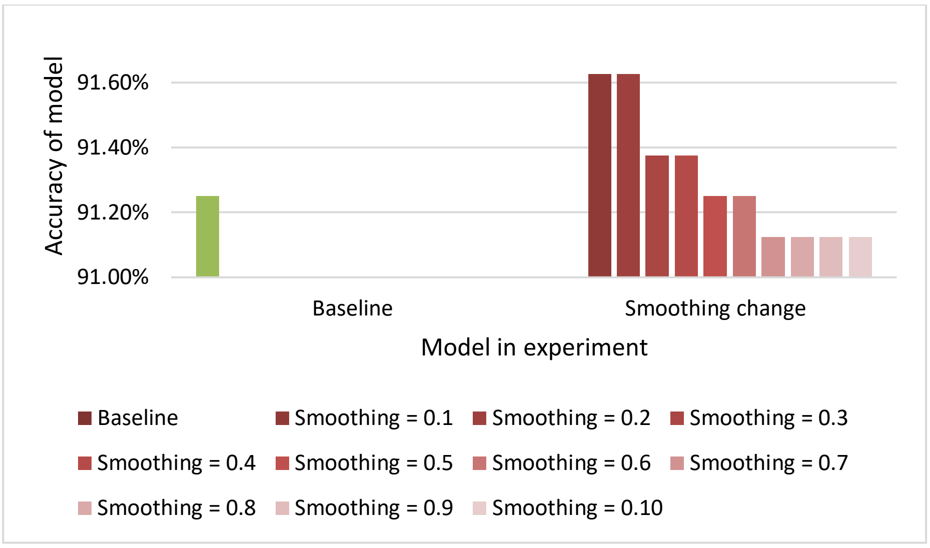


Figure 5. accuracy of models in baseline and change smoothing experiments.

Figure 5 shows that, in email classifier based on our training set, add-one smoothing performs the worst, and the smaller the smoothing factor, the better the performance.

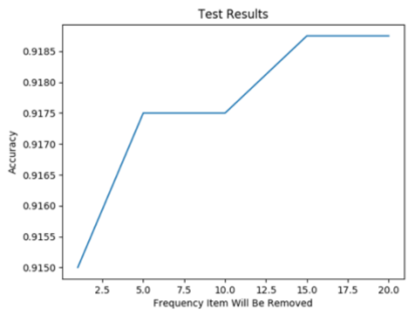
9 References

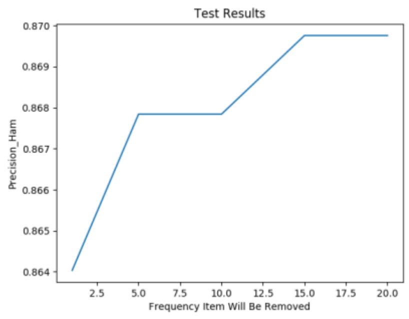
1. Multinomial Naïve Bayes Classifier for Text Analysis, https://towardsdatascience.com/multinomial-naive-bayes-classifier-for-text-analysis-python-8dd6825ece67
2. A practical explanation of a Naïve Bayes classifier, https://monkeylearn.com/blog/practical-explanation-naive-bayes-classifier/
3. Wikipedia, Lemmatisation, https://en.wikipedia.org/wiki/Lemmatisation
4. Blog, 6 practices to enhance the performance of a text classification model, https://www.analyticsvidhya.com/blog/2015/10/6-practices-enhance-performance-text-classification-model/
5. Wikipedia, Additive smoothing, https://en.wikipedia.org/wiki/Additive_smoothing
6. Quora, What is Laplacian smoothing and why do we need it in a Naïve Bayes classifier, https://www.quora.com/What-is-Laplacian-smoothing-and-why-do-we-need-it-in-a-Naive-Bayes-classifier
7. IJCSMC Journal, Vol. 3. Issue. 10, October 2014, page 869-878, https://www.academia.edu/9040601/NAIVE_BAYES_CLASSIFIER_WITH_MODIFIED_SMOOTHING_TECHNIQUES_FOR_BETTER_SPAM_CLASSIFICATION
8. Blog, Spam filter classifier, precision and recall, https://zhuanlan.zhihu.com/p/32300580 (Chinese website)
9. Blog, Classification model performance evaluation – Accuracy, Precision, Recall, F-Score, https://zhuanlan.zhihu.com/p/37246394 (Chinese website)

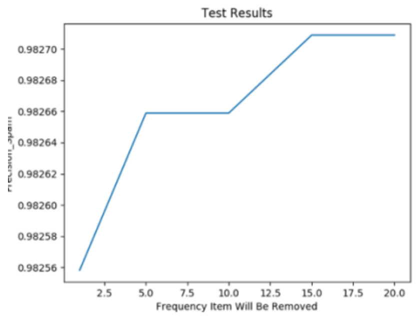
10 Appendix

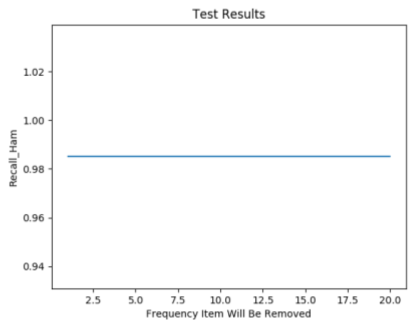
10.1 Output plots in experiment #4: infrequent word experiment

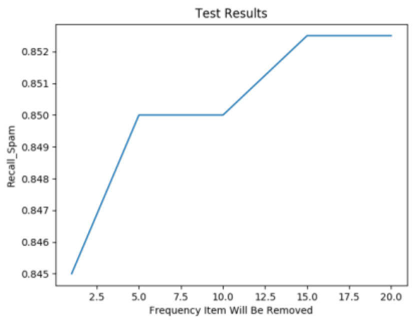
Plots of experiments removing words with frequency of = 1, ≤ 5, ≤ 10, ≤ 15, ≤ 20:

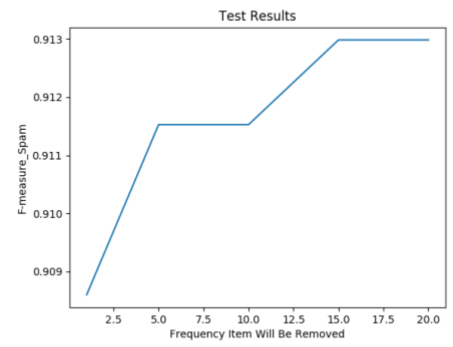


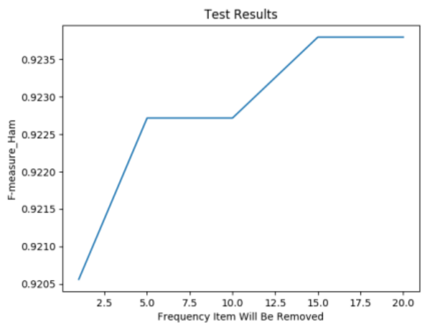




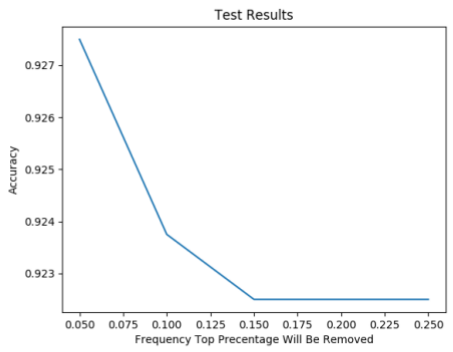


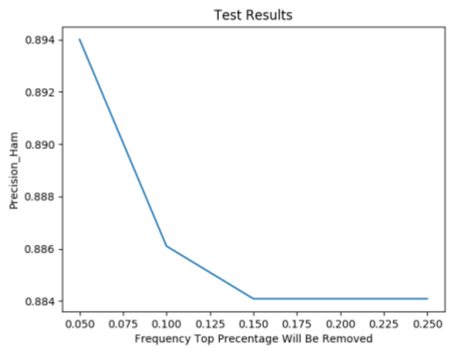


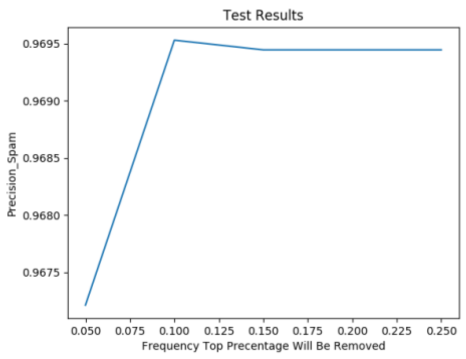


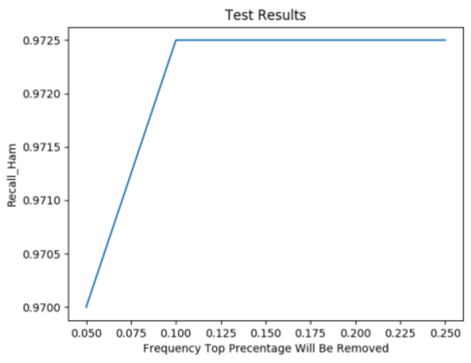


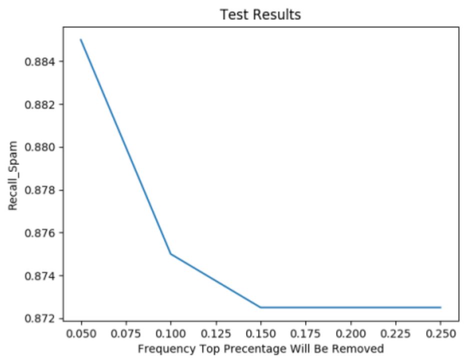
Plots of experiments removing words with frequency of the top 5%, 10%, 15%, 20%, 25%:

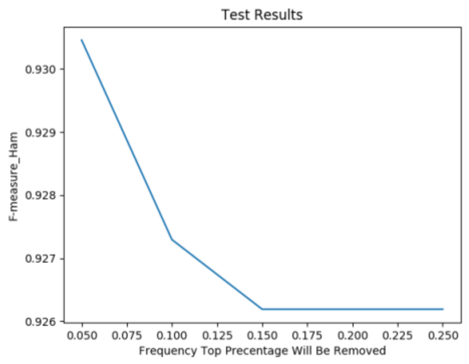


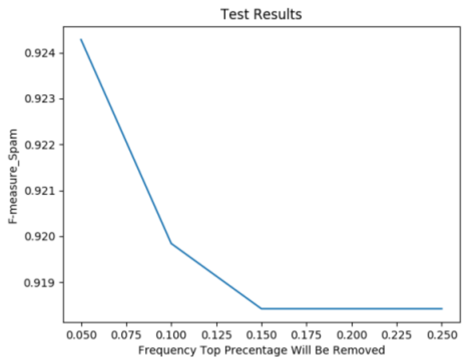








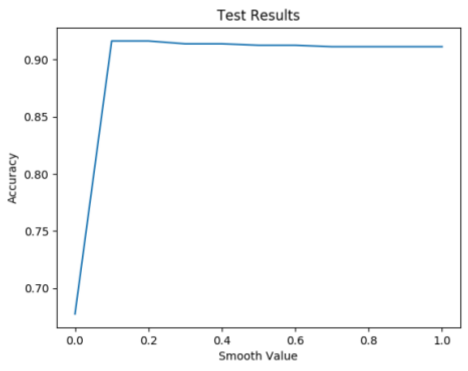




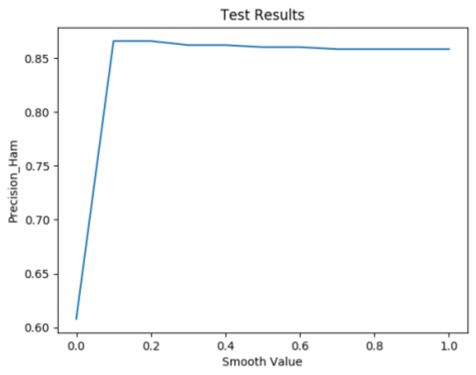
10.2 Output plots in experiment #5: smoothing change experiment

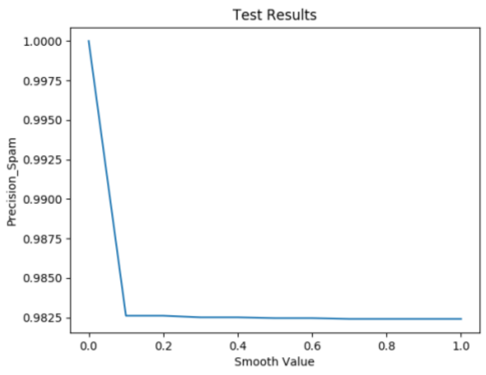
Plots of experiments smoothing factor = 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0

Accuracy:

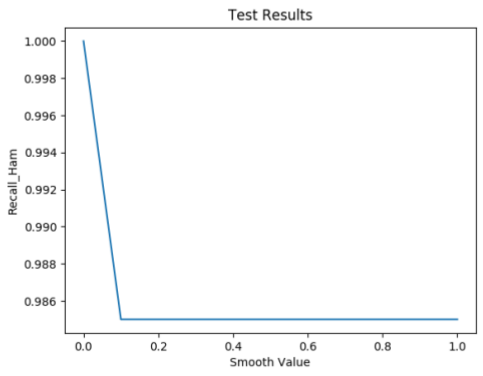


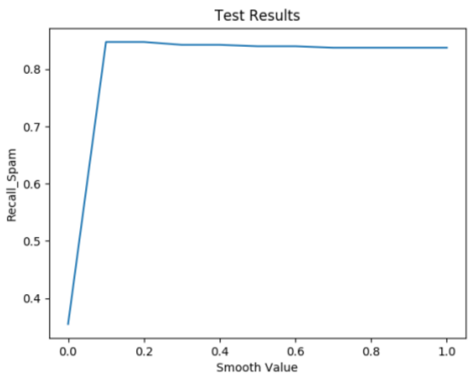
Precision of ham and spam:



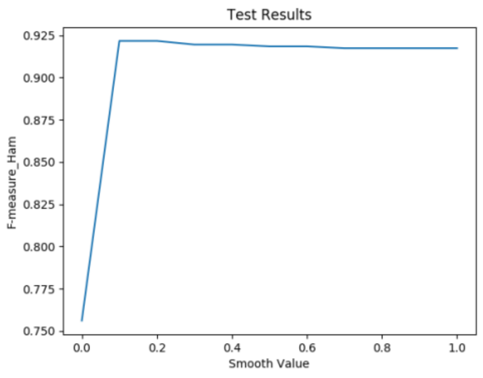


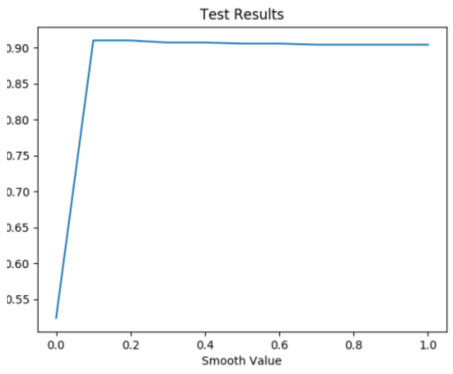
Recall of ham and spam:





F1-measure of ham and spam:


Test Results
F-measure_Ham
Smooth Value


Test Results
Smooth Value