Email Classification with Naïve Bayes

Mitchell Della Marta, Shu Bor

1 Introduction

1.1 Aim

To implement text classification of emails using the Naïve Bayes classifier for spam detection.

1.2 Importance

As email is one of the main forms of communication, spam detection to remove spam is important. Email spam is an unsolved global issue that negatively affects productivity and uses up valuable resources. Spam emails are usually sent with malicious intent to users for some monetary gain. Email users spend a large amount of time regularly deleting these spam emails, which needlessly occupy storage space and consumes bandwidth. Hence the development of classifiers that are able to separate legitimate from spam emails is required.

Traditional non-machine learning based techniques are easily fooled by spammers and require periodic manual updates. Therefore, machine learning based methods are required, which are able to automatically analyse email contents and dynamically update themselves to cope with new spamming techniques.

2 Data Preprocessing

The files in the sample set *Lingpspam-mini600* are represented as a "bag-of-words", where words are extracted from the file and treated as a feature. The *document frequency* feature selection method was used, to select the 200 words which occured in the most documents as features, and used to build a classifier for the given documents. The features are then weighted using *td-idf* and normalised using *cosine normalisation*[1].

1. For each file in the sample set *Lingpspam-mini600*:
    1.1. Open the file and read it line by line.
    1.2. Replace all punctuation and special symbols from each line with a space (e.g. "don't" becomes "don", "t").
    1.3. The sub-corpus the line belongs to is determined by checking if the line begins with "Subject:". If it does the words within the line will be added to the subject corpus, otherwise they will be added to the body corpus.
    1.4. Words from each line are extracted using space as a delimiter.
    1.5. Stop words, numbers and words containing digits are removed.
    1.6. A counter is created for both corpora. For each word that appears in the email, it is added to its respective counter.

1.7. For each word that appeared in the counter for an email, it is added once to the counter for the corresponding subcorpus.

2. In each corpus the words with the top 200 document frequency score (words which appear in the most number of documents) are selected as features for that corpus.

3. For each feature, its *tf-idf* score is calculated.

$$tfidf(t_k, d_j) = \#(t_k, d_j) \times \log \frac{|Tr|}{\#Tr(t_k)}$$

4. The *tf-idf* values are normalised using *cosine normalisation*.

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|}(tfidf(t_s, d_j))^2}}$$

2.1 Data Characteristics

Table I: Characteristics of dataset

	Subject	**Body**
# Features before removing stop words	1074	19886
# Features after removing stop words	915	19386
# $Class_{nonspam}$	600	600
# $Class_{spam}$	200	200

The "bag-of-words" model produced 19886 and 1074 features in the body and subject corpora respectively. After the removal of stop words, there were 19386 unique words remaining in the body corpus, and 915 words in the subject corpus (see Table I).

3 Feature Selection

Feature selection is performed using document frequency. This method involves computing the number of documents a word occurs in, for every word, and selecting the top 200 words with the highest scores to build a classifier.

Shown in Table II and Table III are the top 100 words for the subject and body corpora respectively and their document frequency score. Removing stop words filters out extremely common words that have little value in classification. It is beneficial in text processing, as it removes low quality features, allowing more significant features to have precedence. Thus, given our task to process natural language text, the selection of words shown makes sense as it gives a better representation of the contents of the emails, and helps improve the accuracy of the classifier.

A comparison of Table II and Table III shows significant disparities in the document frequency of features and word distribution. The frequencies for the subject is significantly lower than that of the body. Whilst some features are shared between subject and body, most features selected are different.

Table II: Top 100 words in subject corpus and corresponding document frequency (DF) scores

Rank	Word	Score	Rank	Word	Score	Rank	Word	Score
1	sum	30	35	armey	6	69	address	4
2	summary	26	36	workshop	6	70	information	4
3	english	24	37	dick	6	71	future	3
4	language	21	38	internet	6	72	offer	3
5	free	20	39	languages	5	73	decimal	3
6	disc	19	40	grammar	5	74	chomsky	3
7	query	18	41	word	5	75	double	3
8	linguistics	15	42	research	5	76	change	3
9	sex	13	43	time	5	77	credit	3
10	comparative	13	44	linguist	5	78	opportunity	3
11	words	12	45	software	5	79	requested	3
12	opposites	12	46	needed	5	80	dental	3
13	email	10	47	native	5	81	reference	3
14	book	10	48	resources	5	82	school	3
15	call	9	49	spanish	5	83	counting	3
16	job	9	50	list	5	84	french	3
17	method	9	51	jobs	5	85	released	3
18	japanese	8	52	american	4	86	world	3
19	correction	8	53	request	4	87	linguists	3
20	chinese	7	54	intuitions	4	88	site	3
21	program	7	55	www	4	89	misc	3
22	syntax	7	56	read	4	90	addresses	3
23	announcement	7	57	pig	4	91	uniformitarianism	3
24	qs	7	58	programs	4	92	video	3
25	million	7	59	secrets	4	93	life	3
26	money	6	60	phonetics	4	94	debt	3
27	mail	6	61	banning	4	95	make	3
28	slip	6	62	books	4	96	youthese	3
29	business	6	63	fwd	4	97	names	3
30	part	6	64	great	4	98	corpus	3
31	speaker	6	65	unlimited	4	99	policy	3
32	lang	6	66	summer	4	100	dutch	3
33	conference	6	67	systems	4			
34	german	6	68	web	4			

Table III: Top 100 words in body corpus and corresponding document frequency (DF) scores

Rank	Word	Score	Rank	Word	Score	Rank	Word	Score
1	information	205	35	word	91	69	system	74
2	language	192	36	ll	89	70	full	74
3	mail	183	37	receive	88	71	ac	73
4	university	179	38	check	88	72	today	73
5	time	178	39	phone	88	73	remove	72
6	list	171	40	good	87	74	interest	72
7	address	165	41	year	86	75	questions	72
8	english	159	42	day	86	76	john	71
9	linguistics	156	43	interested	86	77	found	70
10	http	156	44	case	85	78	related	70
11	people	146	45	working	85	79	site	69
12	send	146	46	include	85	80	linguist	69
13	free	144	47	based	84	81	usa	69
14	make	140	48	ve	84	82	text	68
15	email	133	49	home	83	83	read	68
16	work	128	50	part	83	84	point	68
17	number	128	51	note	83	85	ago	67
18	www	122	52	made	83	86	week	67
19	languages	119	53	mailing	81	87	book	67
20	find	118	54	including	81	88	dear	66
21	fax	116	55	type	80	89	making	66
22	order	108	56	web	79	90	cost	66
23	call	103	57	give	79	91	question	65
24	form	101	58	program	79	92	simply	65
25	research	100	59	place	79	93	offer	63
26	linguistic	99	60	line	78	94	general	63
27	state	99	61	special	78	95	received	63
28	world	98	62	date	78	96	data	62
29	years	98	63	days	77	97	important	62
30	subject	98	64	back	76	98	ca	61
31	contact	97	65	internet	76	99	summary	61
32	de	96	66	service	75	100	long	61
33	money	94	67	american	75			
34	word	91	68	business	74			

4 Subject vs Body Analysis

4.1 Results

Table IV: Accuracy of various classifiers tested with 10 fold cross validation for the subject and body corpus

	Accuracy (%)	
Classifier	**Subject**	**Body**
ZeroR	66.67	66.67
OneR	70.00	82.00
1-NN	80.00	87.17
3-NN	69.83	84.33
NB	68.50	94.67
DT	66.67	92.50
MLP	78.17	96.67
MyNB		94.83

4.2 Discussion

5 Challenge Analysis

5.1 Results

5.2 Discussion

6 Conclusions

The tokens used in our classifier are formed from single words. Therefore, it will not analyse common consecutive words that are found in spam emails, leading to the failure of our classifier from detecting these emails.

By taking into account permutations of consecutive words, or words that appear within a specified distance of each other, the accuracy of our Bayesian classifier could be increased.

7 Reflection

8 Instructions: How to run code

References

[1] Fabrizio Sebastiani, *Machine learning in automated text categorization.* ACM Computing Surveys, 34(1):1-47, 2002.