Email Classification

Aim

The aim of this study is to implement text classification of emails using the Naïve Bayes classifier for spam detection.

Spam emails are usually sent to derive cash from the user either directly by tricking the user into purchasing something or indirectly by ticking them into parting with information. The classification of emails aims foremost to protect the unaware user from the malicious intent of spam emails. Classification of emails also saves the user the time and hassle of manually classifying and deleting spam.

Data Preprocessing and Feature Selection

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. 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 add it 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 (words which are appeared in the most number of documents) were selected as features for that corpus.
3. For each feature, its *tf-idf* score is calculated.
4. The *tf-idf* values are normalised using *cosine normalisation*.

The removal of stop words in text processing filters out common words that offer little value for classification. Before stop words were removed, there were 19886 and 1074 unique words in the body and subject corpuses respectively. After their removal, there were 19386 unique words remaining in the body corpus, and 915 words in the subject corpus.

Feature	DF	Feature	DF	Feature	DF	Feature	DF	Feature	DF
sum	30	qs	7	spanish	5	secrets	4	youthese	3
summary	26	syntax	7	list	5	address	4	offer	3
english	24	chinese	7	time	5	fwd	4	line	3
language	21	million	7	word	5	read	4	home	3
free	20	program	7	software	5	request	4	site	3
disc	19	business	6	languages	5	information	4	check	3
query	18	slip	6	linguist	5	systems	4	reference	3
linguistics	15	conference	6	jobs	5	american	4	background	3
comparative	13	lang	6	research	5	intuitions	4	cd	3
sex	13	part	6	native	5	great	4	teaching	3
words	12	german	6	resources	5	double	3	decimal	3
opposites	12	money	6	www	4	change	3	latin	3
book	10	workshop	6	unlimited	4	synthetic	3	names	3
email	10	armey	6	programs	4	credit	3	counting	3
method	9	speaker	6	web	4	requested	3	ipa	3
job	9	dick	6	banning	4	future	3	corpus	3
call	9	mail	6	phonetics	4	tonight	3	complete	3
japanese	8	internet	6	summer	4	make	3	world	3
correction	8	grammar	5	books	4	comparison	3	resolution	3
announcement	7	needed	5	pig	4	hey	3	dialect	3

Feature	DF	Feature	DF	Feature	DF	Feature	DF	Feature	DF
information	205	fax	116	interested	86	special	78	site	69
language	192	order	108	year	86	line	78	text	68
mail	183	call	103	day	86	days	77	read	68
university	179	form	101	working	85	internet	76	point	68
time	178	research	100	include	85	back	76	week	67
list	171	linguistic	99	case	85	american	75	ago	67
address	165	state	99	based	84	service	75	book	67
english	159	subject	98	ve	84	system	74	dear	66
linguistics	156	years	98	note	83	business	74	cost	66
http	156	world	98	home	83	full	74	making	66
people	146	contact	97	made	83	ac	73	question	65
send	146	de	96	part	83	today	73	simply	65
free	144	money	94	including	81	interest	72	offer	63
make	140	message	91	mailing	81	questions	72	received	63
email	133	word	91	type	80	remove	72	general	63
number	128	ll	89	web	79	john	71	data	62
work	128	check	88	give	79	related	70	important	62
www	122	phone	88	program	79	found	70	ca	61
languages	119	receive	88	place	79	linguist	69	summary	61
find	118	good	87	date	78	usa	69	long	61

Subject vs Body: Results and Discussion

Corpus: Subject	
	Accuracy [%]
ZeroR	
OneR	
1-NN	
3-NN	
NB	
DT	
MLP	
MyNB	

Corpus: Body	
	Accuracy [%]
ZeroR	
OneR	
1-NN	
3-NN	
NB	
DT	
MLP	
MyNB	

Challenge Results and Discussion

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.

Reflection

Instructions: How to run code