

Dynamic Markov Coding

for

Spam Filtering

Gordon V. Cormack

3 April, 2006

University of
Waterloo



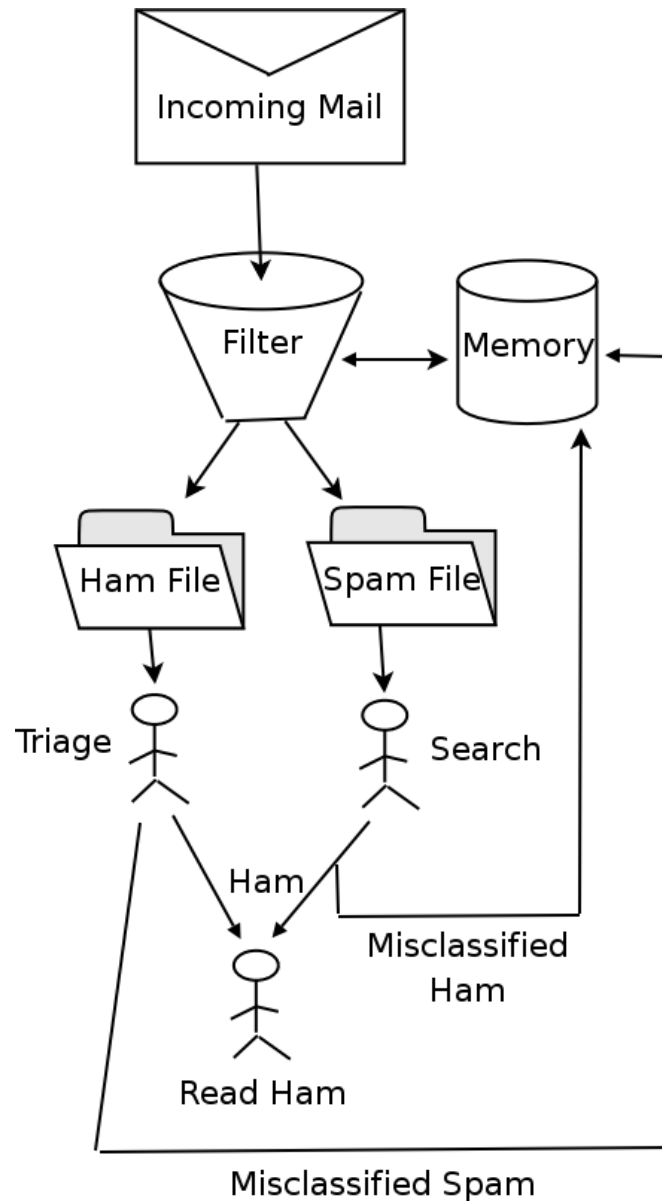
TREC definition

Unsolicited, unwanted email that was sent indiscriminately, directly or indirectly, by a sender having no current relationship with the recipient.

Depends on sender/receiver relationship

Not “whatever the user thinks is spam.”

Spam Filter Usage



Filter Classifies Email

Human addressee

Triage on ham File

Reads ham

Occasionally searches
for misclassified ham

Report misclassified
email to filter

TREC – Text Retrieval Conference (On-line)

chronological order

full email messages with *original* headers

idealized user feedback: *immediate, accurate*

Ling Spam (Batch, mailing list messages)

headers, punctuation, case removed

10-fold cross-validation

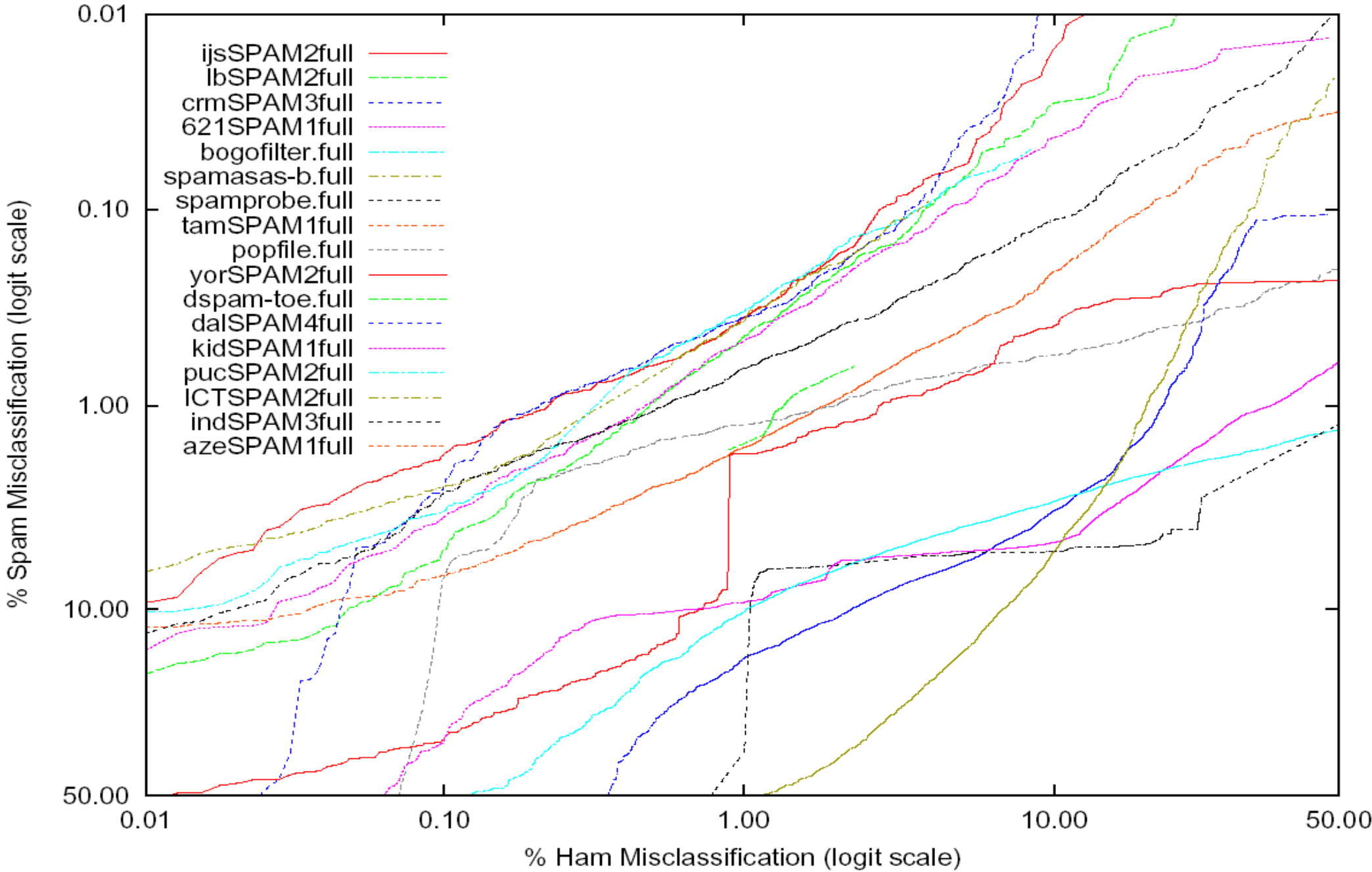
PU1 (Batch, obfuscated email messages)

tokens translated to decimal integers

10-fold cross-validation



ROC Curves – TREC



Hard Classification – TREC

Run	Hm%	Sm%	Lam%
bogofilter	0.01	10.47	0.30
ijsSPAM2	0.23	0.95	0.47
spamprobe	0.15	2.11	0.57
spamasas-b	0.25	1.29	0.57
crmSPAM3	2.56	0.15	0.63
621SPAM1	2.38	0.20	0.69
lbSPAM2	0.51	0.93	0.69
popfile	0.92	1.26	0.94
dspam-toe	1.04	0.99	1.01
tamSPAM1	0.26	4.10	1.05
yorSPAM2	0.92	1.74	1.27
indSPAM3	1.09	7.66	2.93
kidSPAM1	0.91	9.40	2.99
dalSPAM4	2.69	4.50	3.49
pucSPAM2	3.35	5.00	4.10
ICTSPAM2	8.33	8.03	8.18
azeSPAM1	64.84	4.57	22.92

Summary Measures – TREC

Run	(1-ROCA)%	Rank	Sm% @ Hm%=0.1	Rank	Lam%	Rank
ijsSPAM2	0.02	1	1.8	1	0.5	2
lbSPAM2	0.04	2	5.2	7	0.7	7
crmSPAM3	0.04	3	2.6	3	0.6	5
621SPAM1	0.04	4	3.6	6	0.7	6
bogofilter	0.05	5	3.4	5	0.3	1
spamasas-b	0.06	6	2.6	2	0.6	3
spamprobe	0.06	7	2.8	4	0.6	4
tamSPAM1	0.16	8	6.9	8	1.1	10
popfile	0.33	9	7.4	9	0.9	8
yorSPAM2	0.46	10	34.2	10	1.3	11
dspam-toe	0.77	11	88.8	15	1.0	9
dalSPAM4	1.37	12	76.6	13	3.5	14
kidSPAM1	1.46	13	34.9	11	3.0	13
pucSPAM2	1.97	14	51.3	12	4.1	15
ICTSPAM2	2.64	15	79.5	14	8.2	16
indSPAM3	2.82	16	97.4	16	2.9	12
azeSPAM1	28.89	17	99.5	17	22.9	17

Prediction by Partial Matching

For each class:

left context occurrences

left context+prediction

log-likelihood estimate

compressed length

Smoothing/backoff:

zero occurrence problem

Adaptation:

increment counts

assuming in-class

ai.stanford.?



Context (509 spam, 1 ham)

ai.stanford.e



Prediction (0 spam, 1 ham)

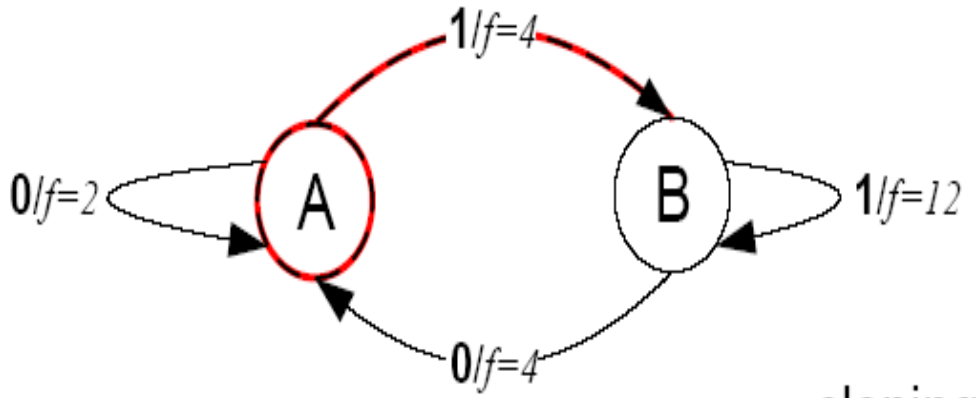
ai.stanford.E



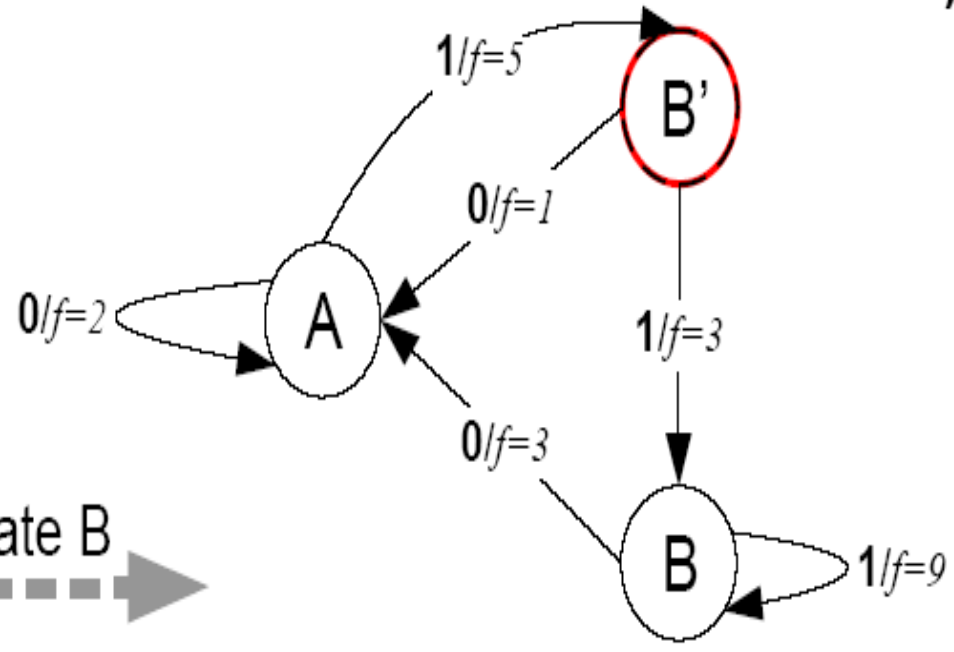
Prediction (509 spam, 0 ham)


DMC State Cloning

a)

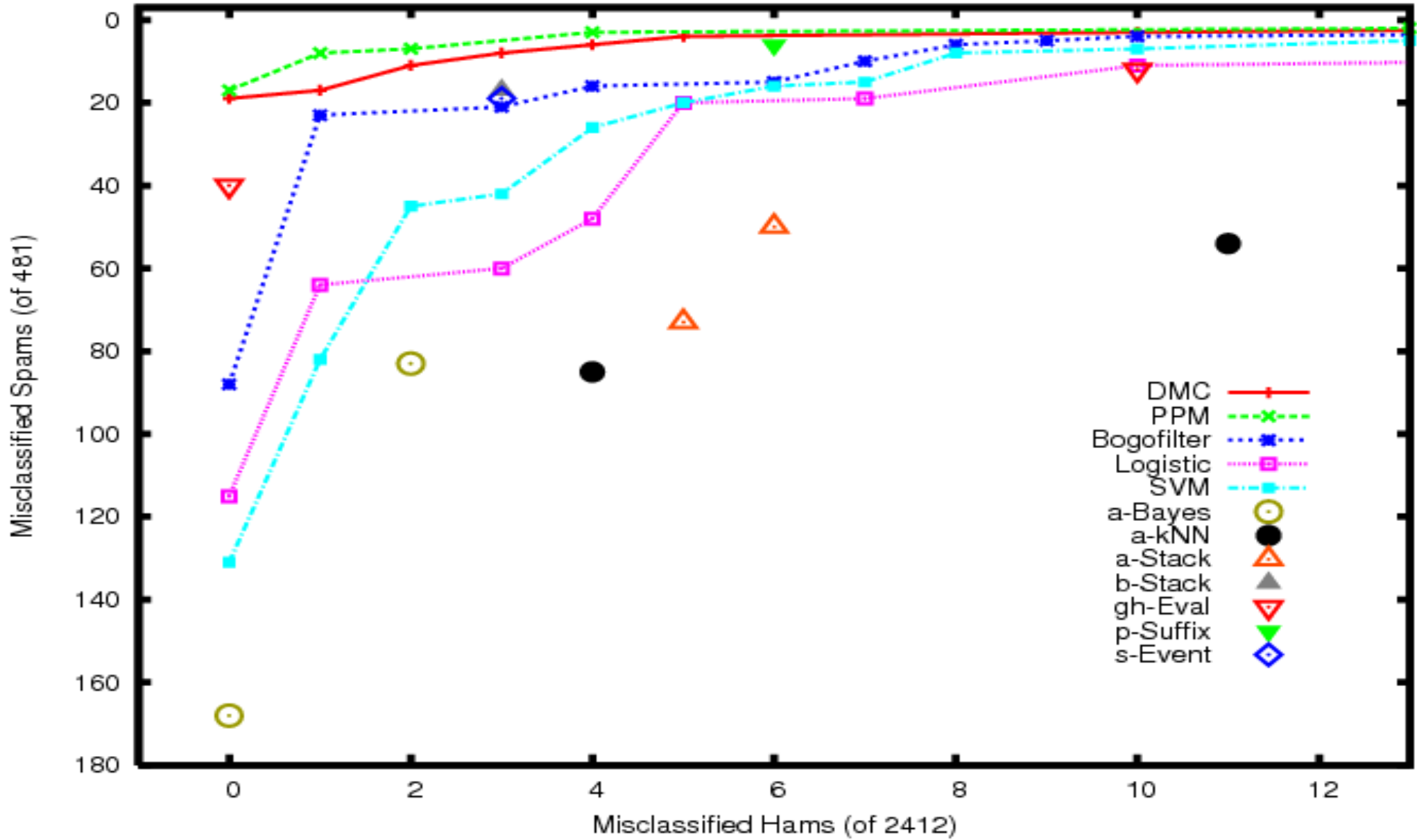


b)

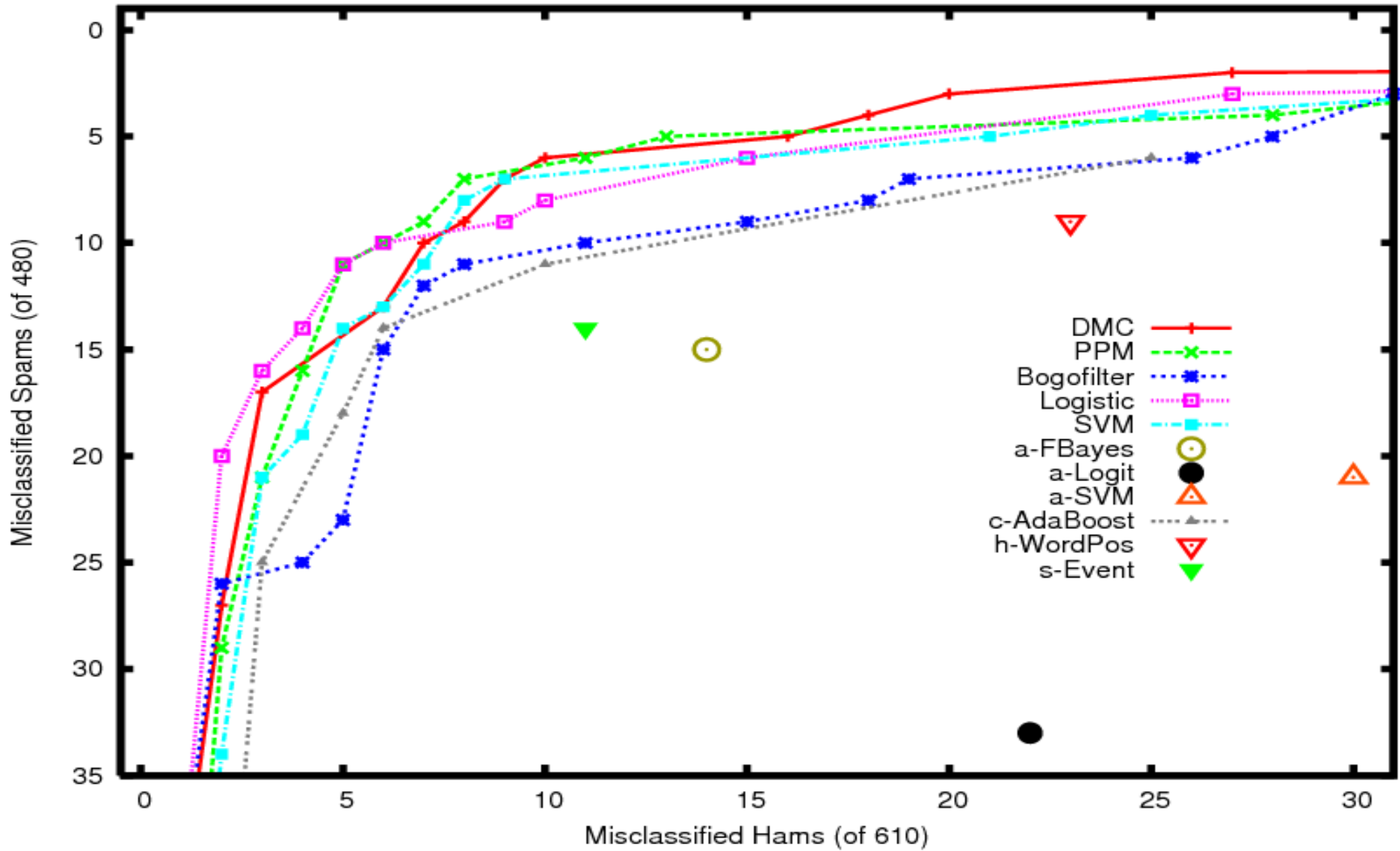


cloning of state B


Ling Spam Corpus



PU1 Corpus

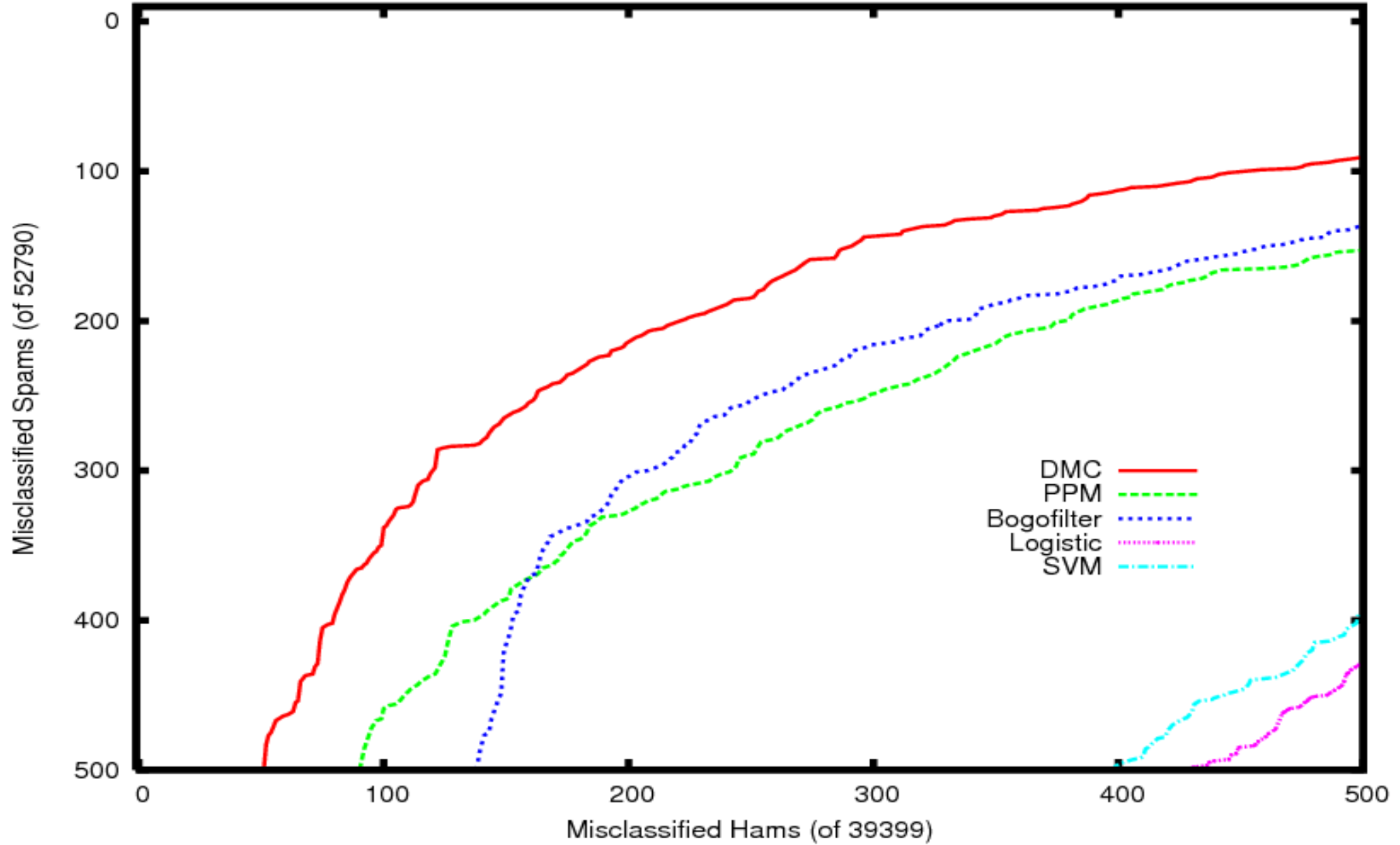


Ling Spam/PU1 (1-ROCA)%

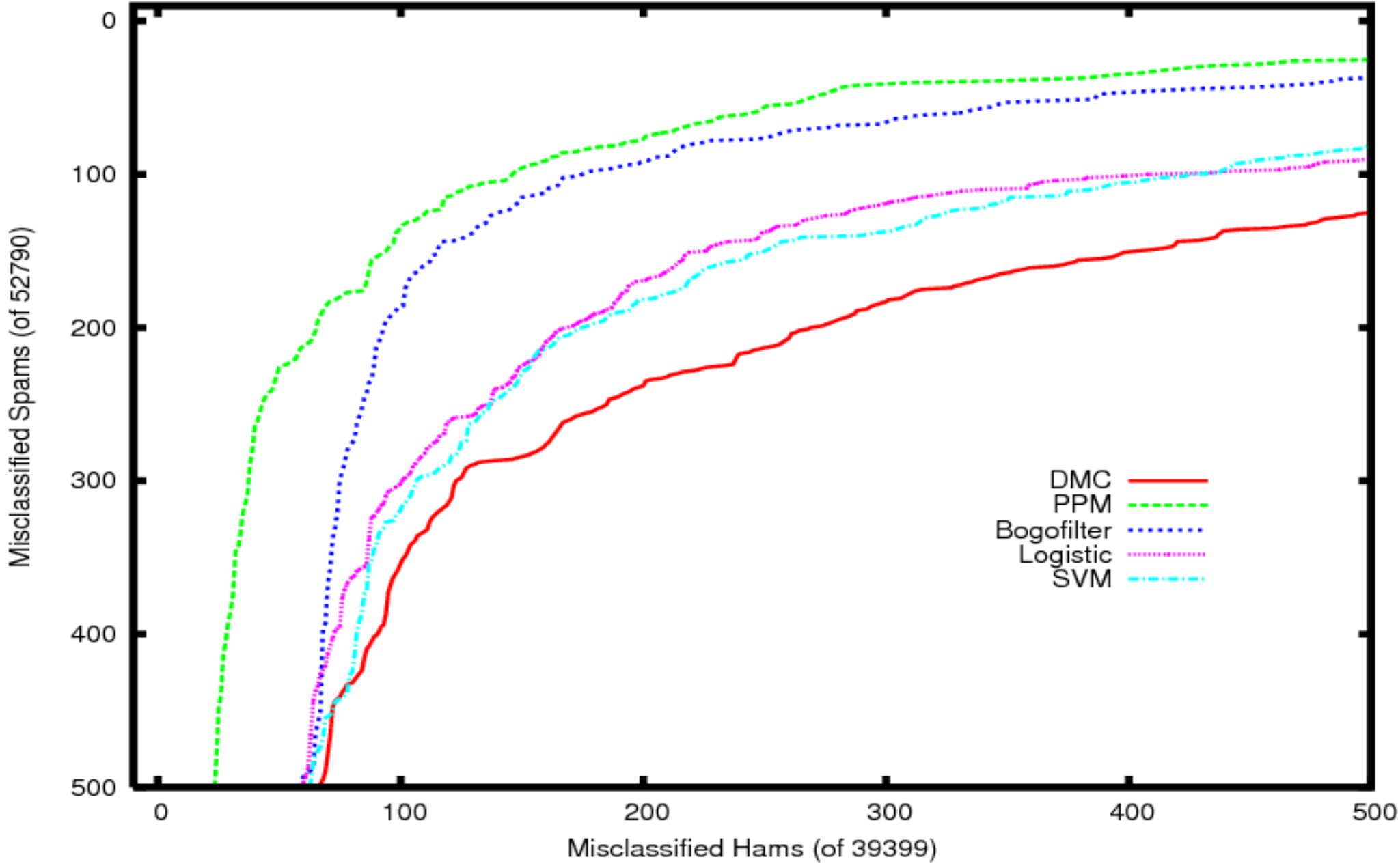


Method	Ling Spam	PU1
DMC	.07 (.01-.49)	.26 (.09-.80)
PPM	.04 (.004-.39)	.18 (.08-.43)
Bogofilter	.04 (.02-.11)	.21 (.10-.43)
LR	.087 (.04-.17)	.20 (.07-.53)
SVM	.14 (.08-.26)	.22 (.10-.50)

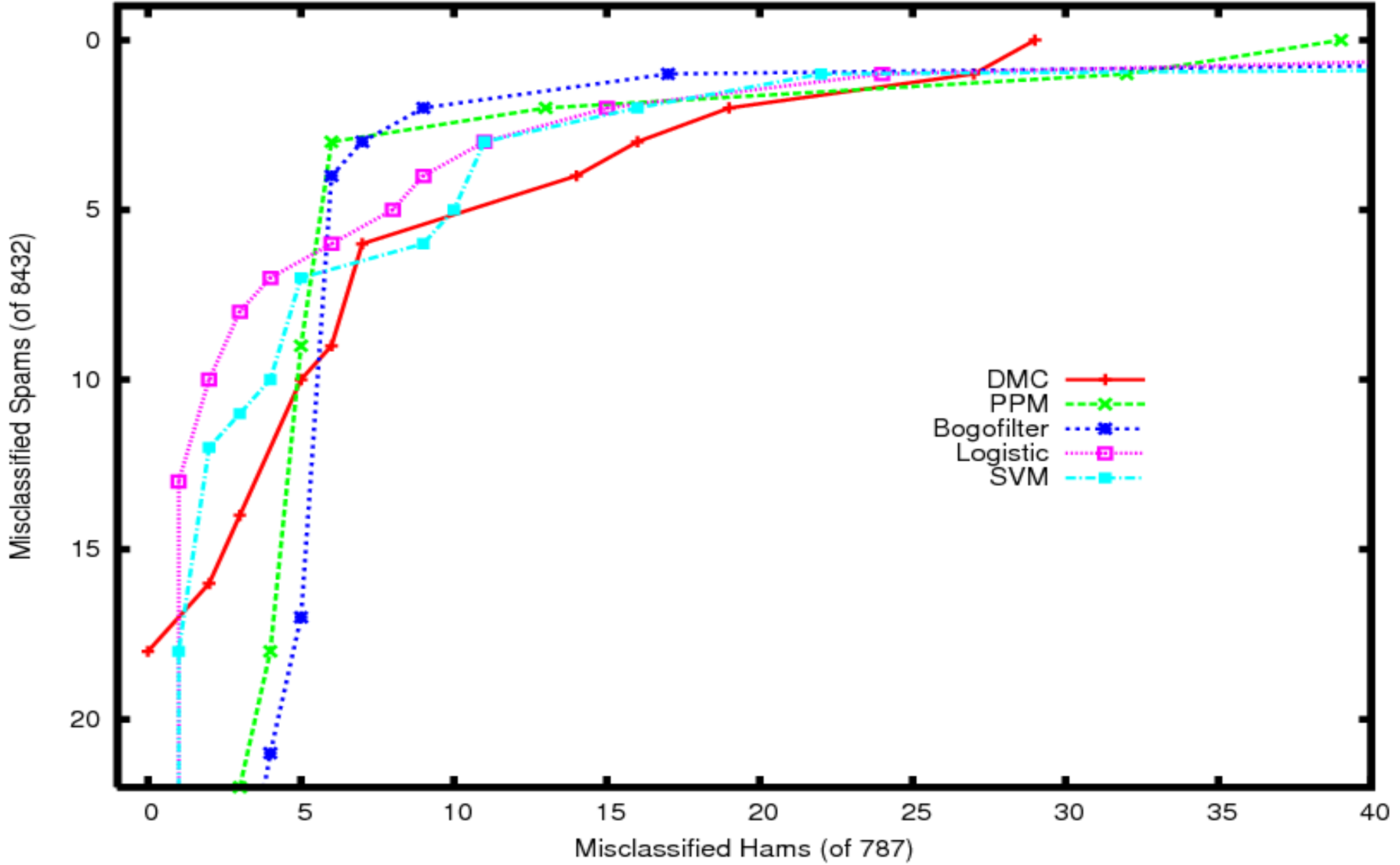
TREC Corpus, On-line



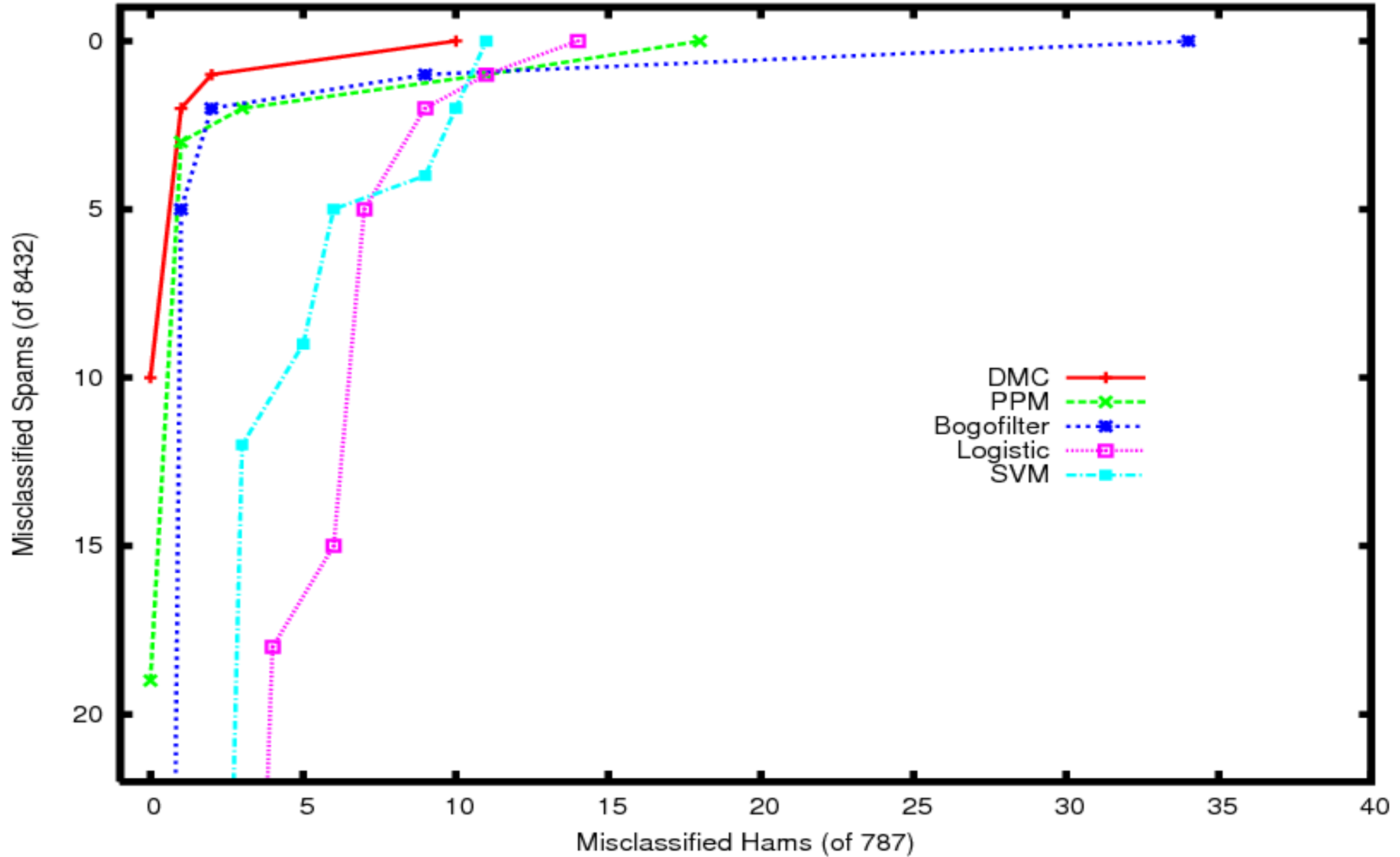
10-Fold Cross Validation



9:1 Chronological, Batch



9:1 Chronological, On-line



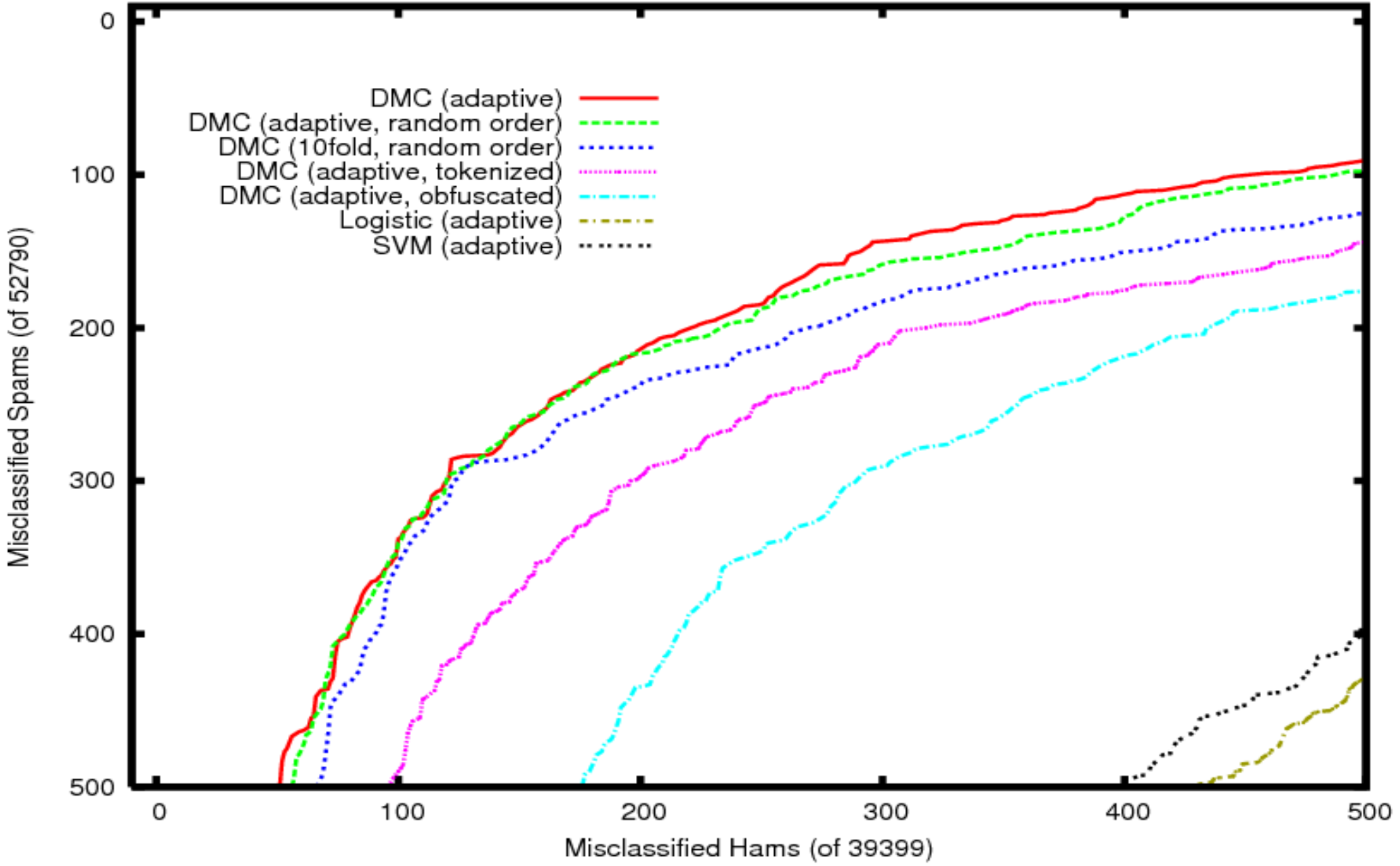
Batch, On-line (1-ROCA)%

Method	On-line		Batch	
	Full Corpus	9:1 Chronological	10-fold C.V.	9:1 Chronological
DMC	.013 (.010-.018)	.0003 (.0000-.003)	.015 (.012-.018)	.003 (.001-.006)
PPM	.017 (.014-.021)	.0007 (.0001-.005)	.006 (.004-.009)	.003 (.001-.008)
Bogofilter	.048 (.038-.062)	.002 (.0001-.041)	.020 (.012 - .033)	.009 (.003-.029)
LR	.068 (.058-.079)	.020 (.003-.135)	.016 (.012-.021)	.12 (.001-10.1)
SVM	.075 (.064-.088)	.007 (.0015-.033)	.021 (.015-.029)	.13 (.003-5.6)

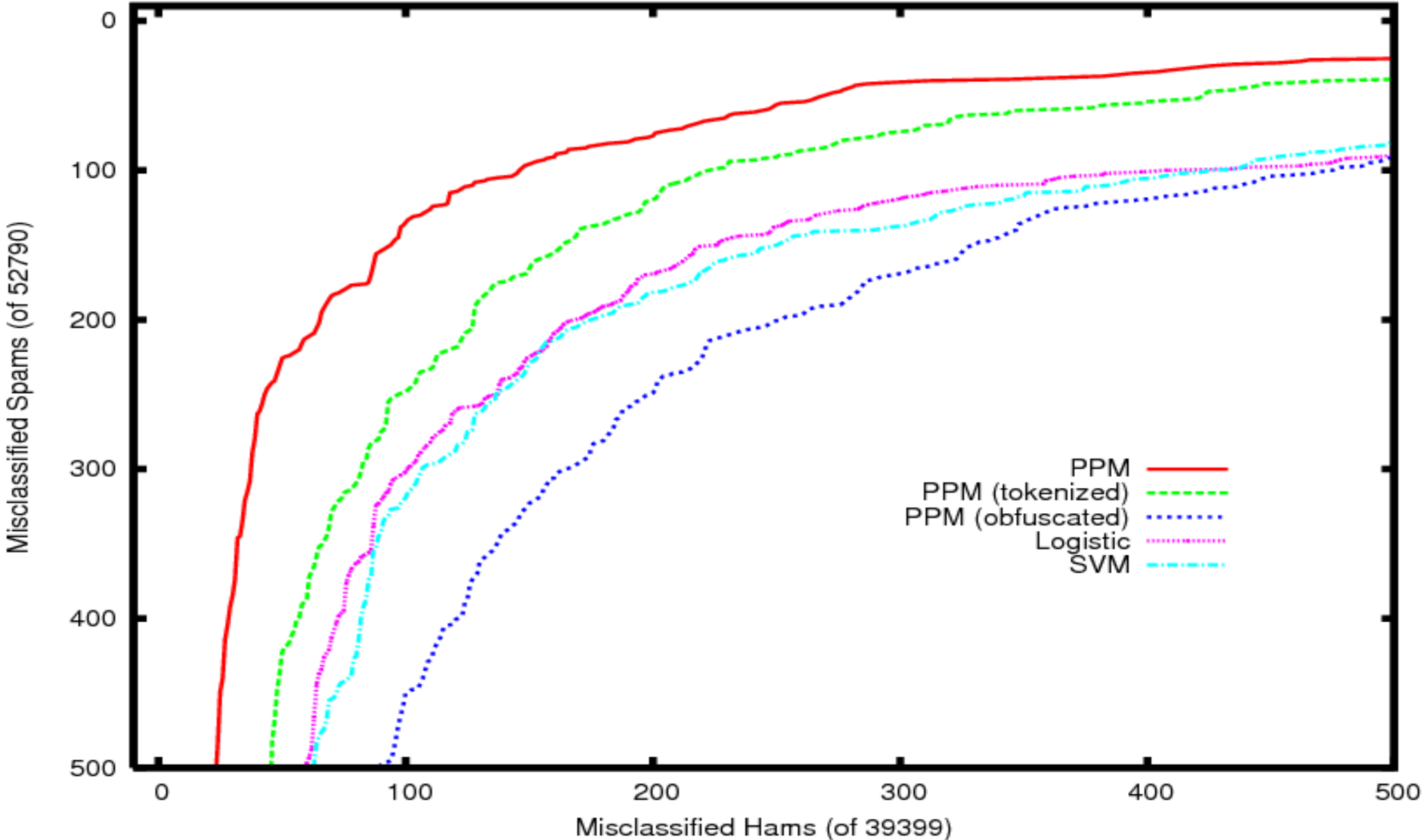
Effect of Order/Adaptation

Filter	Training Regimen	Testing Regimen		
		On-line Random Order	On-line Corpus Order	Batch
DMC	Random Order	.01 (.006-.017)	.007 (.004-.011)	.009 (.006-.015)
DMC	Corpus Order	.035 (.026-.047)	.037 (.024-.057)	.31 (.25-.37)
PPM	Batch	.0052 (.003-.01)	.0053 (.003-.009)	.0055 (.003-.01)

Tokenization, On-line



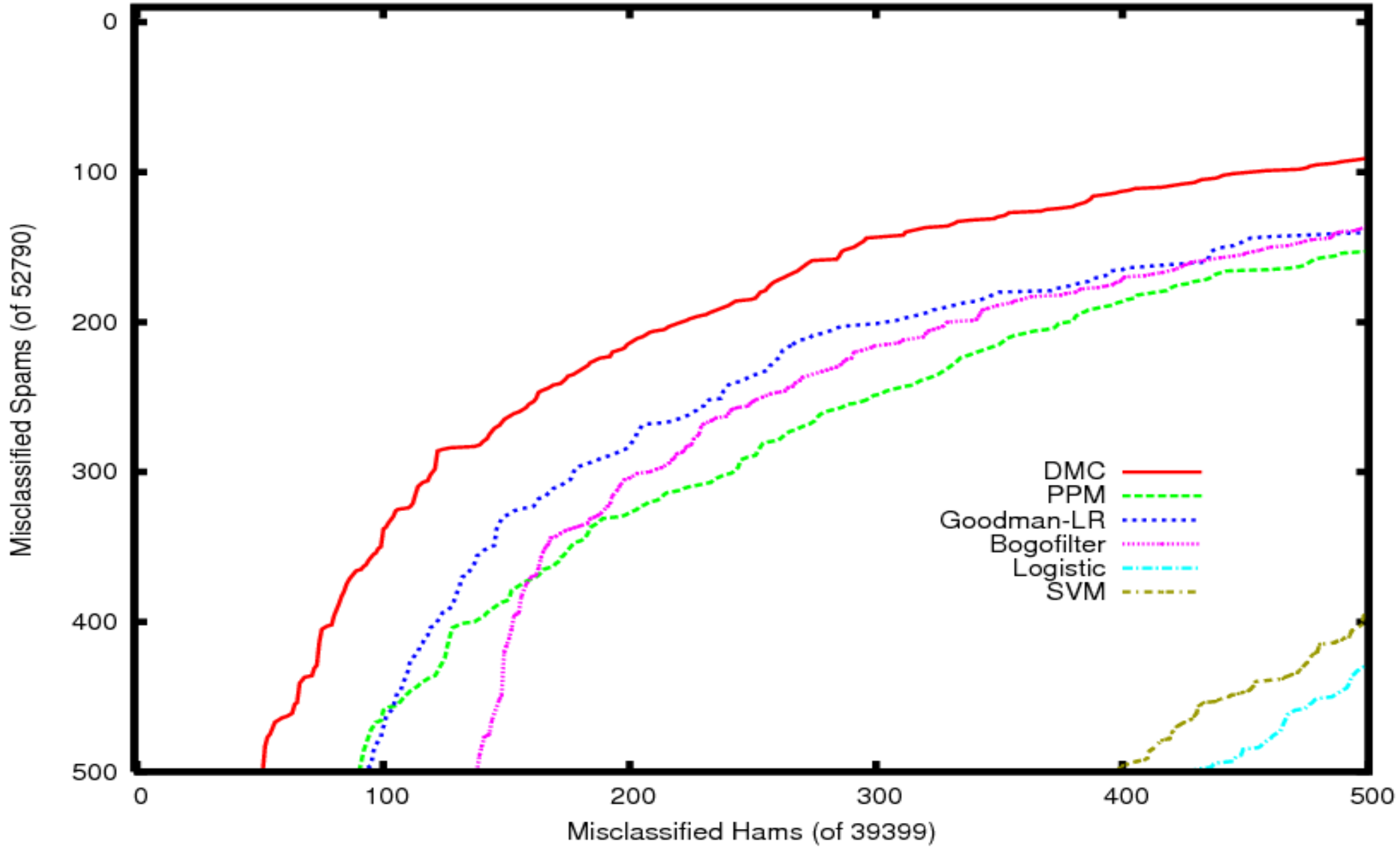
Tokenization, Batch



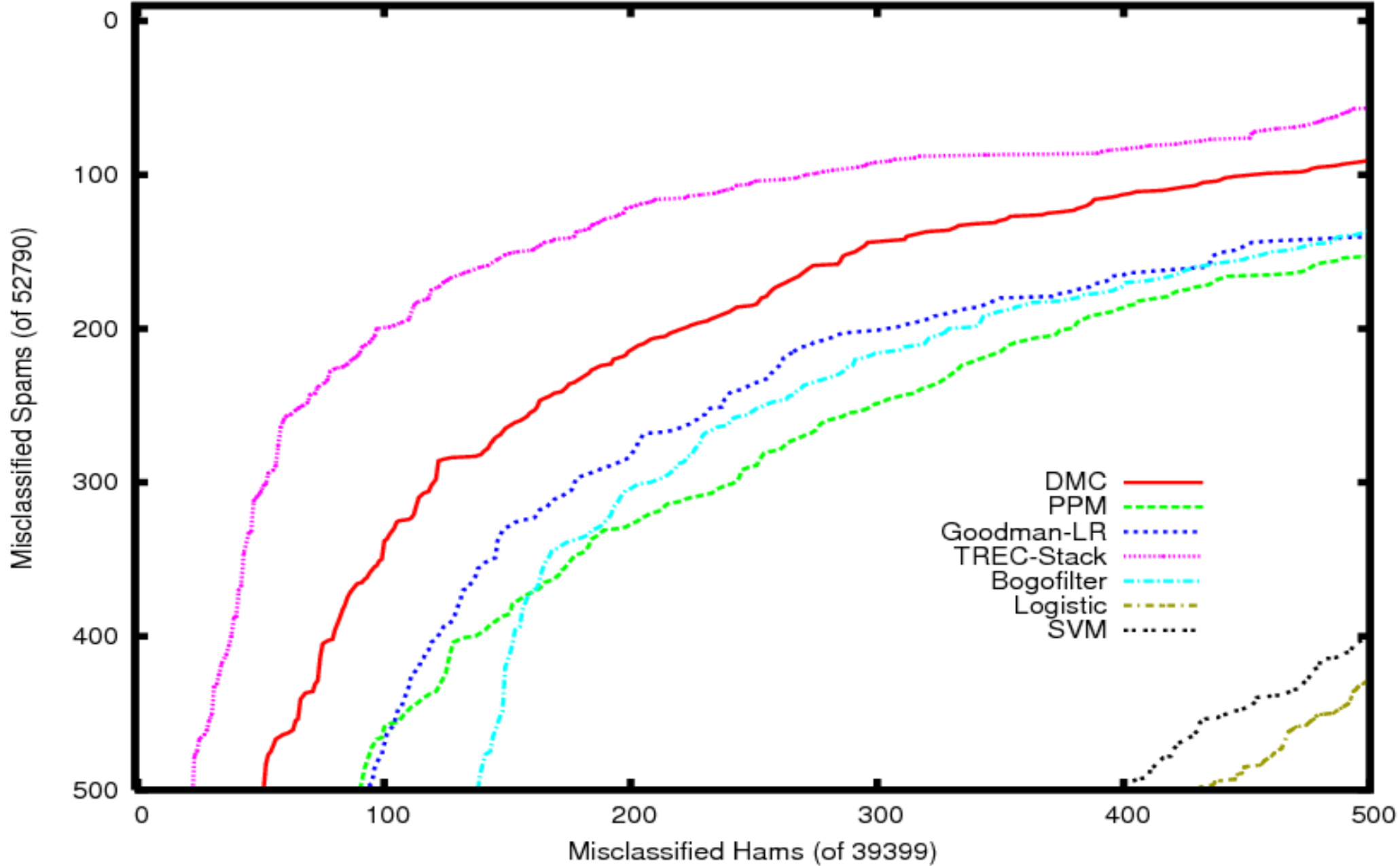
Tokenization/Obfuscation

Method	On-line		Batch	
	Full Corpus	9:1 Chronological	10-fold C.V.	9:1 Chronological
<i>DMC</i>	.013 (.010-.018)	.0003 (.0000-.003)	.015 (.012-.018)	.003 (.001-.006)
tokenized	.025 (.020-.032)	.0006 (.0001-.006)	.025 (.019-.033)	.001 (.000-.013)
obfuscated	.037 (.030-.045)	.0004 (.0000-.0042)	.029 (.023-.037)	.002 (.001-.006)
<i>PPM</i>	.017 (.014-.021)	.0007 (.0001-.005)	.006 (.004-.009)	.003 (.001-.008)
tokenized	.038 (.033-.045)	.0016 (.0003-.009)	.012 (.009-.016)	.005 (.002-.012)
obfuscated	.075 (.066-.084)	.0046 (.0016-.013)	.020 (.014-.027)	.015 (.006-.035)
<i>Bogofilter</i>	.048 (.038-.062)	.002 (.0001-.041)	.020 (.012 - .033)	.009 (.003-.029)
obfuscated	.13 (.11-.15)	.024 (.004-.14)	.055 (.045-.068)	.036 (.012-.11)

Goodman's Gradient Descent LR



Stacking – 53 TREC Filters



Further information

<http://plg.uwaterloo.ca/~gvcormac/dmcsbam.pdf>

<http://plg.uwaterloo.ca/~gvcormac/sigir.pdf>

ask Joshua

Different corpora

Practical deployment

personal implementation (RAM, state saving)

server implementation (inter-user effects)

ML methods

adaptation, feature engineering