

Ending Spam

- 스팸의 역사
 - 최초 스팸의 등장과 진화
- 스팸 대응의 역사
 - 통계 기반 필터링 이전의 시대 (~2002년)
- 통계적 필터링의 기초
 - Bayesian Content Filtering
- 스팸머의 역습
 - 필터를 통과하기 위한 몸부림
- 필터의 정확도 향상 기법
 - HMM, 가중치 자동 조정, 쓰레기 토큰 제거, 집단 지성
- 참고자료

- 단순한 정의
 - 수신을 원하지 않는 대량의 이메일
- 최초의 출현
 - 1978년
 - Digital Equipment Corporation 광고
 - 모든 수신자를 수작업으로 입력함
 - 메일 헤더의 수신자 목록만 9쪽
 - 첫 브로드캐스트에 320개 발송됨
 - SNDMSG 프로그램 자체의 버퍼 한계



최초의 스팸 전문 - DEC



Mail-from: DEC-MARLBORO rcvd at 3-May-78 0955-PDT
Date: 1 May 1978 1233-EDT
From: THUERK at DEC-MARLBORO
Subject: ADRIAN@SRI-KL

DIGITAL WILL BE GIVING A PRODUCT PRESENTATION OF THE NEWEST MEMBERS OF THE DECSYSTEM-20 FAMILY; THE DECSYSTEM-2020, 2020T, 2060, AND 2060T. THE DECSYSTEM-20 FAMILY OF COMPUTERS HAS EVOLVED FROM THE TENEX OPERATING SYSTEM AND THE DECSYSTEM-10 (PDP-10) COMPUTER ARCHITECTURE. BOTH THE DECSYSTEM-2060T AND 2020T OFFER FULL ARPANET SUPPORT UNDER THE TOPS-20 OPERATING SYSTEM. THE DECSYSTEM-2060 IS AN UPWARD EXTENSION OF THE CURRENT DECSYSTEM 2040 AND 2050 FAMILY. THE DECSYSTEM-2020 IS A NEW LOW AND MEMBER OF THE DEC-SYSTEM-20 FAMILY AND FULLY SOFTWARE COMPATIBLE WITH ALL OF THE OTHER DECSYSTEM-20 MODELS.

WE INVITE YOU TO COME SEE THE 2020 AND HEAR ABOUT THE DECSYSTEM-20 FAMILY AT THE TWO PRODUCT PRESENTATIONS WE WILL BE GIVING IN CALIFORNIA THIS MONTH. THE LOCATIONS WILL BE:

TUESDAY, MAY 9, 1978 - 2 PM
HYATT HOUSE (NEAR THE L.A. AIRPORT)
LOS ANGELES, CA

THURSDAY, MAY 11, 1978 - 2 PM
DUNFEY'S ROYAL COACH
SAN MATEO, CA
(4 MILES SOUTH OF S.F. AIRPORT AT BAYSHORE, RT 101 AND RT 92)

A 2020 WILL BE THERE FOR YOU TO VIEW. ALSO TERMINALS ON-LINE TO OTHER DECSYSTEM-20 SYSTEMS THROUGH THE ARPANET. IF YOU ARE UNABLE TO ATTEND, PLEASE FEEL TO CONTACT THE NEAREST DEC OFFICE FOR MORE INFORMATION ABOUT THE EXCITING DECSYSTEM-20 FAMILY

○ 1988년

- Jay-Jay's College Fund
 - 여러 뉴스 그룹에 스팸 글 게시

○ 1990년대 초

- The Jesus Spam
 - 수신자 목록을 입력받아 자동으로 스팸 발송
- Canter & Siegal
 - 프로그래머를 고용하여 전용 스팸 발송기를 제작
 - 광고, 마케팅 업계에 대량 메일 발송 기법을 전파
- Jeff Slaton
 - 초창기에 스팸으로 돈을 번 사람 중 하나
 - 스팸 명단에서 제외하는 대가로 \$5씩 받는 뻔뻔함을 보임
 - 연간 \$300,000 ~ \$600,000 정도 벌었을 것으로 추산

○ 1995년

- 스팸 발송이 비즈니스로 진화
- Stanford Wallace, Cyber Promotions
 - AOL에서 계정을 차단하자 고소했으나 기각됨
 - CompuServe와 AOL 등 네트워크 제공자에게 패소
- Floodgate - 최초의 스팸웨어
 - 28.8K 모뎀으로 1시간에 1000개의 스팸 발송
 - 백만 개의 이메일 주소를 \$100에 판매
 - 스팸 발송의 대중화

○ 1996년

- 경쟁적으로 발표된 스팸 웨어
 - Lightning Bolt, Ready-Aim-Fire, E-Mail Blaster

○ Spamhaus

- 1996년 7월, 블랙리스트의 원형
 - SBL (Spamhaus Blackhole List)
 - XBL (Exploits Blackhole List)
- 스팸머들에게 고소 협박을 많이 받음
 - 상표 등록된 자사의 도메인 불법 사용
 - 명예 훼손

○ "UCE" 용어 등장

- 1996년 4월, Usenet abuse FAQ
- Unsolicited Commercial Email

- 1997년 스팸의 급격한 팽창
 - 1분기 "오픈 릴레이" 용어 등장
 - 2분기 동안 스팸 10배 이상 증가
 - 새로운 스팸 툴의 등장
 - Extractor Pro, Stealth, Goldrush
 - 당시의 초보적인 스팸 필터를 우회
 - 2억개가 넘는 메일 주소 목록을 CD로 판매
 - 실시간 블랙홀 리스트 Paul Vixie's RBL 등장
 - ISP 라우터에서 통신 중에도 차단시키는 강경 조치

스팸 대응의 역사

- 시기
 - 1994년 ~ 1997년
- 원리
 - "Call Now"나 "Free Trial" 같은 스팸 문구를 필터링
- 장점
 - 그 당시 사용 가능했던 유일한 해결책
 - 커스터마이징이 매우 쉬움
- 단점
 - 엄청난 유지보수 노력이 필요
 - 낮은 정확도
 - 높은 에러율
- 사용 여부
 - 폐기됨

- 시기
 - 1997년 ~ 1999년
- 원리
 - 구독 기반의 실시간 블랙홀 리스트
- 장점
 - 네트워크 리소스 보존
- 단점
 - ISP 주소 할당이 동적인 경우 전파 속도가 느림
 - 블랙리스트 관리가 어려움
 - 정상적인 메일도 차단됨
- 사용 여부
 - 오픈 릴레이 서버나 봇넷을 이용하면서 거의 쓸모 없어짐

○ 시기

- 1990년대 후반 BrightMail

○ 원리

- 스팸 마스터가 데이터베이스 관리
- 오탐 샘플을 보내서 데이터베이스에 추가

○ 단점

- 데이터베이스 유지보수 어려움

○ 시기

- 2001년 SpamAssassin 등장

○ 원리

- 다양한 휴리스틱 룰셋 이용
 - 헤더 특징 분석
 - 특정 단어 분석
 - 블랙리스트 대조
- 각 점수의 합이 임계치를 넘으면 스팸 취급

○ 단점

- 스팸 점수 부여가 자의적이고 일관성이 없음
- 필터를 통과하는 스팸을 만들어 발송하면서 정확도가 매우 떨어짐

○ 원리

- 메일 전송자 기준으로 필터링

○ 장점

- 위조가 없다고 가정하면 100% 정확도

○ 단점

- SMTP는 전송자 주소를 쉽게 위조 가능함
- 기존에 연락하던 사람 외에는 모두 거부됨

○ 원리

– 절차

- 미등록 사용자의 메일은 보류
- 미등록 사용자에게 특정 링크를 눌러 활성화 요구
- 해당 메일 주소가 활성화되면 메일 교환 가능

○ 장점

- 화이트리스트 유지의 부담을 사용자에게 넘김

○ 단점

- 사용자들이 매우 짜증내는 방식
- 많은 경우 메일 교환을 아예 포기함

○ 원리

- 대량 발송의 특징을 역이용하여 트래픽 제한

○ 장점

- 스팸에 들어가는 자원을 억제할 수 있음
- 적극적인 대응으로 스팸머를 곤란하게 함
 - TarProxy
 - reject : 554 I don't need any Viagra. Go Away
 - tempfail : 451 I'm tired of this. Spam me later.

○ 단점

- 정상적인 대량 메일 발송도 제한됨
- 정상적인 사용자를 화나게 만들 수 있음

○ 사용 여부

- 대형 ISP에서 쓸만한 방법

- 원리
 - 메일 주소를 사람만 식별 가능하게 작성
- 장점
 - 메일 주소를 기계적으로 수집하기 어려움
- 단점
 - 근본적인 해결책이 아님

- 원리
 - SMTP 프로토콜에 인증 기능을 추가
- 장점
 - 스팸 메일 중계를 막을 수 있음

○ 원리

- DNS TXT 레코드에 SPF 관련 정보를 추가
- ISP에게 reverse MX 조회를 허용
 - 보낸 사람 메일 주소의 위조 여부 확인
- 예시 (자세한 것은 RFC 4408 참조)
 - example.org. IN TXT "v=spf1 a mx -all"
 - v : SPF 버전 정의
 - a : A 레코드에 전송자 IP가 있으면 매치
 - mx : MX 레코드에 전송자 IP가 있으면 매치
 - -all : 앞에서 매치되지 않은 것들을 모두 거부

○ 장점

- 위조된 스팸 메일 주소를 몰아낼 수 있음
- 스팸 도메인을 일괄적으로 거부할 수 있게 됨

○ 단점

- 메일을 보낼 때 항상 해당 호스트를 통해야 함
- 모든 서버에서 사용하는 기법은 아니므로 스팸 방어에는 불충분

○ 원리

- 웹 페이지에 아주 까다로운 라이선스 작성
- HTML 주석으로 메일 주소 작성
- 봇이 주소를 수집하고 스팸 보내면 고소

○ 장점

- 확실하게 스팸 비즈니스 불능화

○ 단점

- 일일이 법률적 대응하기가 곤란함

○ 원리

- 사용 단어나 문장 구조 등의 패턴 인식
- 내용이 아닌 전송자 기준으로 메시지 그룹화
- SpammerPrinting 참조

○ 원리

- 특정 문구를 지적 재산으로 등록
- 문구를 본문에 포함시켜 보내는 경우 통과
- 스팸머가 해당 문구를 사용하면 고소

○ 단점

- 범용적으로 사용하기 어려움
- 일일이 법률적으로 대응하기 어려움

○ Probing

- 전송자가 오픈 릴레이로 동작하는지 확인
- 약간의 전송 지연 시간이 발생함

통계적 필터링의 기초

○ 1990년대 ~ 2002년

- 휴리스틱 필터링이 유일한 스팸 해결 방안
- 끝없는 무장 경쟁
 - 스팸머는 스팸을 변형
 - 필터 작성자는 룰을 변경
 - 스팸머에게 밀릴 수 밖에 없는 상황 전개

○ 차세대 기술의 등장 : 언어 분류법

- 텍스트를 특정 범주로 분류
- 여러가지 응용 가능
 - 웹사이트 자동 분류
 - 스팸 자동 분류
- 훈련을 통한 기계 학습이 핵심
 - 유지보수가 필요없어져 상황이 역전됨

○ Bayesian content filtering

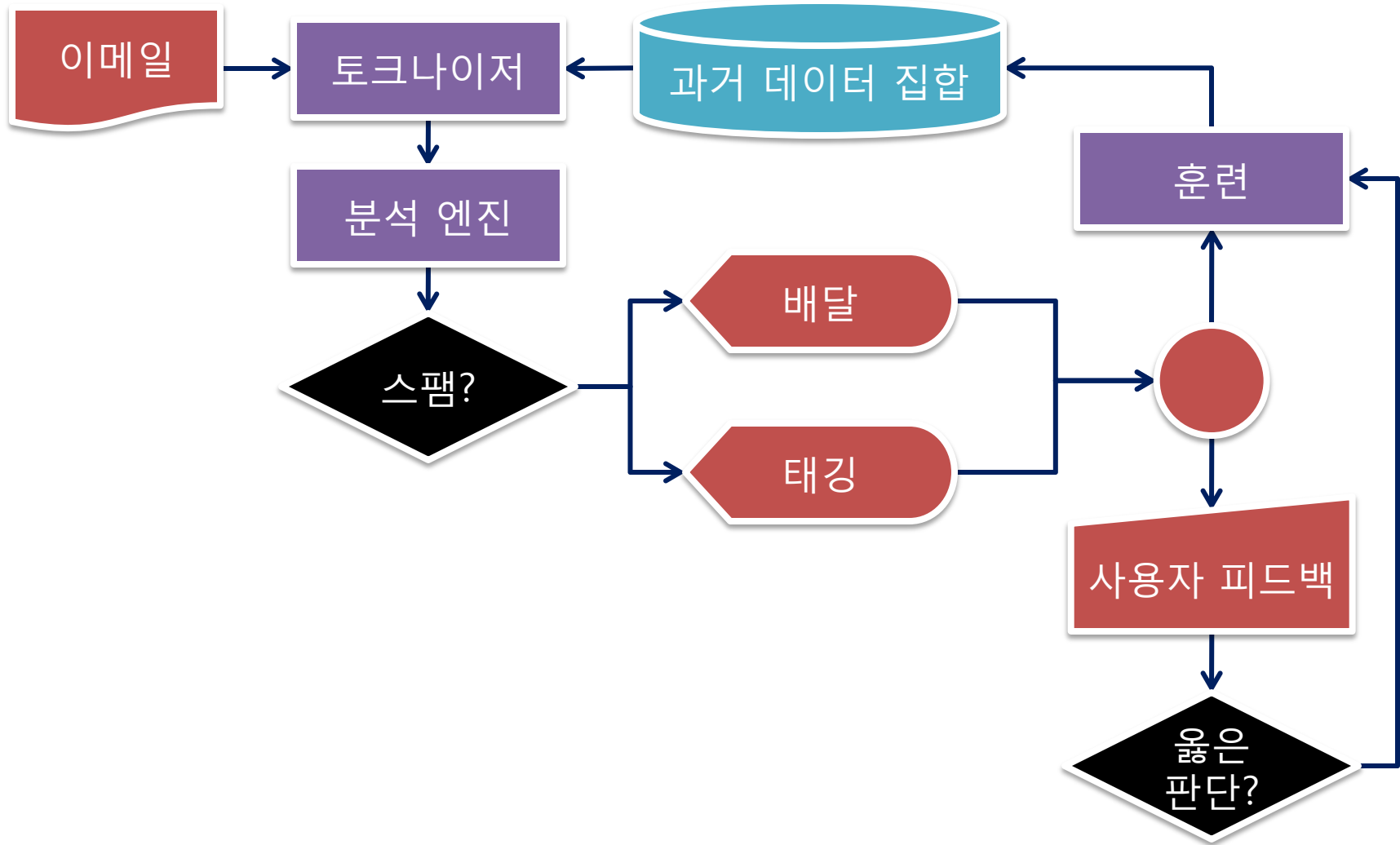
– Bayes' Theorem

- 불확실성을 정량화 하는 방법
- 다양한 변수를 넣고 주어진 확률이 참일 확률 계산

– Paul Graham

- 2002년, A Plan for Spam 논문으로 첫 발표
- 메일의 독립적 특성들을 조합하여 스팸 판정하는데 Bayes' Theorem 이용

언어 분류기의 처리 흐름



○ 4가지 훈련 방식

– Train-Everything (TEFT)

- 모든 메일을 학습
- 사용자의 이메일 사용 패턴 변화에 빠르게 적응
- 수신량이 많은 경우 변동폭이 커서 에러율 증가
- 데이터 집합 크기가 매우 커질 수 있음

– Train-on-Error (TOE)

- 에러가 발생한 경우에만 학습
- 아주 많은 양의 메일을 받는 경우 적합한 방식
- 이메일 사용 패턴이 자주 바뀌는 경우 정확도가 떨어짐
- 새로운 유형의 스팸 등장에 반응하는 속도도 느림

– Train-Until-Mature (TUM)

- TEFT와 TOE의 절충안
- 새로운 토큰의 정확도가 확보되었을 때 TEFT에서 TOE로 전환

– Train-Until-No-Errors (TUNE)

- 에러 없을 때까지 훈련하고 더 이상 학습하지 않음
- 에러가 나타나면 처음부터 다시 학습시켜야 하므로 귀찮음
- 잘 쓰이는 방식은 아니지만 TOE보다 정확도가 높은 것으로 나타남

Corpus 입력

| 토큰 | 스팸 출현 | 햄 출현 |
|------------|-------|------|
| fun | 19 | 9 |
| girlfriend | 4 | 0 |
| mariners | 0 | 7 |
| tell | 8 | 30 |
| the | 96 | 48 |
| vehicle | 11 | 3 |
| viagra | 20 | 1 |

오류 정정하기

○ 토큰 재분류

- 단순한 카운트 조작
 - 1 빼고 1 더한다
- 실제 스팸이 아닌 경우,
 - SpamHit -1, HamHit+1
- 실제 스팸인 경우,
 - SpamHit +1, HamHit-1

개별 토큰의 스팸 확률 계산

$$\frac{\binom{SH}{TS}}{\binom{SH}{TS} + \binom{IH}{TI}} = P$$

- SH
– 토큰이 스팸에 나타난 횟수
- IH
– 토큰이 햄에 나타난 횟수
- TS
– 전체 스팸 수
- TI
– 전체 햄 수

예시

$$\frac{\binom{96}{224}}{\binom{96}{224} + \binom{48}{112}} = 0.5$$

- SH
– 96
- IH
– 48
- TS
– 224
- TI
– 112

○ Single-Corpus 토큰

- 스팸이나 햄 중 한 쪽으로만 나타난 토큰
- 확률 하나가 공식 전체 값을 좌우하는 문제
 - 게다가 스팸 100%와 햄 100% 나오면 모순됨
- 보정
 - 스팸에만 나타난 경우 0.9900 대입
 - 햄에만 나타난 경우 0.0100 대입

○ 정렬

- $\text{abs}(0.5 - P)$ 기준으로 토큰을 내림차순 정렬

○ 입력값 선택

- 가장 스팸이나 햄으로 치우친 토큰 선택
- 보통 상위 15개 토큰을 선택함
- Decision Matrix 입력 수가 제한되므로 쓰레기 단어를 많이 집어넣어도 스팸 판정 가능

$$\frac{ABC \dots N}{ABC \dots N + (1 - A)(1 - B)(1 - C) \dots (1 - N)}$$

○ 수식

- Bayes' Theorem 이용
- 여러 토큰의 스팸 확률을 조합
- 일반적으로 매우 극단적인 결과가 나옴
 - 0.0에 아주 가깝거나 1.0에 아주 가까운 결과
 - 0.9 이상이면 스팸으로 간주

○ Graham 방식의 변형

- Decision Matrix 입력을 27개로 늘림
- 두 번 이상 반복된 단어는 슬롯 2개 할당
 - 여러 번 나타난 단어에 가중치 부여
 - 토큰 수가 너무 적은 경우 중요하지 않은 토큰이 힘을 발휘하게 되는 문제 해결

○ 상호보완성

- 이메일 크기에 따라 적용하면 정확도 향상
 - 일반적인 크기 : Graham 방식
 - 너무 작거나 너무 큰 크기 : Burton 방식

$$P = 1 - ((1 - P1)(1 - P2) \dots (1 - PN))^{\frac{1}{N}}$$

$$Q = 1 - ((P1)(P2) \dots (PN))^{\frac{1}{N}}$$

$$S = \frac{1 + \left(\frac{P - Q}{P + Q}\right)}{2}$$

○ 기하 평균

- P : 메시지의 스팸 수준
- Q : 메시지의 햄 수준
- S : 조합된 결과값
 - Bayesian 결과값은 너무 극단적
 - 55% 이상인 경우 스팸으로 간주
- 현재는 개량된 Fisher-Robinson 공식을 사용함

$$H = C^{-1}(-2 \ln(F1F2 \dots FN), 2N)$$

$$S = C^{-1}(-2 \ln((1.0 - F1)(1.0 - F2) \dots (1.0 - FN))), 2N)$$

$$I = (1 + H - S)/2$$

○ Inverse Chi-Square

- 변수

- H : 햄일 확률
- S : 스팸일 확률
- I : 최종 결과
- N : Decision Matrix에서 사용된 토큰의 수
- F1F2...FN : 각 토큰 스팸 확률의 곱

- 특징

- 정렬이 필요없음
- 확률이 0.0~0.1, 0.9~1.0 으로 분포

스팸머의 역습

- 6가지 Content-Transfer-Encoding
 - 7bit
 - 내용이 짧고 US-ASCII 문자집합만 나오는 경우
 - 8bit
 - 내용이 짧고 ASCII가 아닌 문자들이 나타나는 경우
 - 유니코드나 제어 문자 등
 - binary
 - 내용이 길고 ASCII 아닌 문자들을 포함하는 경우
 - 8bit와 다르게 줄 길이 제한을 받지 않음
 - base64
 - 256가지를 출력 가능한 ASCII 표현으로 변환
 - 원본 대비 33% 정도 큰 결과물이 나오게 됨
 - quoted-printable
 - ASCII 이외의 문자만 인코딩
 - 5가지 규칙 적용. RFC 2045 참조
 - custom-encoding
 - 확장의 여지를 남겨놓은 것으로 잘 쓰이지 않음
 - x-myencoding

- 메시지 헤더 인코딩
 - 두 가지 인코딩 방식
 - base64
 - quoted-printable
 - RFC 2047 참조
 - 헤더의 특정 부분만 인코딩 될 수 있음
 - 헤더 하나에 여러가지 인코딩 혼재 가능

○ HTML 악용

– 문자 인코딩

- `CALL NOW, IT'S FREE!`
- CALL NOW, IT'S FREE!
- URL 인코딩에도 이런 기법을 이용함

– 주석이나 태그 쑺셔넣기

- With awe<!tapestry>some results for hun<!wield>dreds of thous<!locale>ands of men all over the planet!

– 쓰레기 단어를 사람 눈에만 안 보이게 가리기

- ``
- ``

○ 문자열을 쪼개기

– 예시

- Get Your F/R/E/E 10 Day Supply N/O/W!

- F-R-E-E V/I/A/G/R/A

– 단일 문자에 대한 스팸성 증가

http://geocities.com/dava183230duz/?a=wjRY1w1gnmLfsGmp&q=qnMj05MjQtYWFtTE1MDMwNTA

| | | | | | | | | | | | | |
|------------------|------|------------|------------------|------------|--------------|--------------|----------|-----------|------------|------|----------------|------------|
| ap9v | 3565 | dx00r7g1z2 | 2001w7pn7pb24z6p | 818fs4z0rg | xhhqudd4 | sk5qq9q7fan2 | c3kp | 1m520e8t1 | | | | |
| kn6u | 742f | 96n8 | 065n | d71e | 57q3 | nf | 4938 | gg2g | d67i | 436v | m0hu | 9m |
| c8n5 | 0usz | b004 | aya2 | 193o | tggy | 6a5h | 8nk2 | gugg | h2w2 | i5p3 | r5d1nw | |
| cx63 | 06h1 | 3sdh | 7q57 | cswy | 37x3 | 1g19 | 6wsa | ofsu | 9k90 | | | |
| hm633s8yca1832ga | 488b | fg41 | 428k | 3iw2 | v5xq | av828kqq6126 | 8909 | | 22d5ke6gan | | | |
| 09e4 | gko7 | z174 | f3z4 | v635 | 865q | 00581375 | 9w45 | 0vdg | y4hh | 0289 | | o637ub |
| 9thv | 4ten | 2570 | 8192 | 2h48 | 11zn | os39 | 67v9 | 2co3 | 1832 | 7f7v | | 86k3 |
| k407 | 2yle | 2x04 | 0p9s | 3x63 | r1cc | ad13 | 5h12 | 199o | 4p2o | n1r1 | 38 | y3p8 |
| o45s | o8y3 | g7vw6xap61 | 4g1x | | slfnf82qd528 | | t850n11o | ur7v | 1961 | | 33828588g184h1 | g8o8871tcb |

http://geocities.com/dava183230duz/?a=wjRY1w1gnmLfsGmp&q=qnMj05MjQtYWFtTE1MDMwNTA

| | | | | | | | | | | |
|-------------|-------------|------------|------------|----------|------|---------|------------|------|------|------|
| q5pwz6oc11 | | | | 9auh | 9ctc | w3pi | | l1yf | | |
| jnx8evrs8w | | | | rm3j | | k10x | | rpvo | | |
| fe0z p5jp | p5cv40 | 72buegc | 1bph byn8 | ukxfnfm9 | yxlo | p5dclrg | 0muk | 6c02 | 96ys | |
| 11as ms5x | eq96sk02 | qa1qathj | asfx sune | 3fiho1h5 | pk42 | b8oqh5h | 8smf | 1eqv | objg | |
| gfe8w5zzu | ho1 y15q | 0 zw5t | uros | iflo | 8eye | fpey | nkhj | 0pf9 | qxhw | |
| hxczmn5qem | xpww powt | upqm | coa7 | wtde | lkuz | wecf | n8g6 | 9xr6 | 19r8 | noxy |
| y93zyoa4yds | s4aegkti4zm | k60c5feq | 4k0a | ea7u | pnok | nrfy | ck8d | qgdt | 0h5z | 1bz3 |
| pn6s 991z | d8bv1s112y2 | qs2gf0g1g | d8jb | vo2w | r4ko | ytzm | r6nj | m8gt | nuy8 | k6wn |
| vyf9 ccdu | cypc | t09r 2rvo | vdvm | pmfq | vd80 | 9td6 | igsx | us57 | f7at | gps4 |
| 9np8 q77c | 8h6m 3 | gakf 4zhi | oaag | fqsn | q9tw | fgsx | bba4 | ow2u | 2car | lo0x |
| kdingpeazb0 | jclik um | mm5s 2kt9 | 01uib | m03e | b1yq | wvy2 | bck8 | kuft | p60f | vub4 |
| 1tm0j2g0wy | dfiz20wp1 | 3w01gxnltz | be6fzoi7ws | 8126e75 | xbwe | k741 | f3fmyxnu4d | e7by | | |
| e99uo4n89 | dsxiyix | r3na 4f7f | zjf6 h0cl | wyfl5 | 4a6d | 4q93 | cuyd 5afn | 8opy | | |

링크를 포함하면 필터링 가능
링크가 없는 경우 클릭 불가능
OCR 기법으로 검출할 수 있음

| | |
|-------------|-----------|
| 7cvjya5u | t1fi |
| kr6cs34ig8 | wp7s |
| ulzqixg14 | yf2s |
| 5p1a fr07 | 765b |
| 5yjh vf05 | uifrus8 |
| 9t949jnfv | or4m sq1 |
| hkwg00mwgo | q4mkb0fh |
| 1c5dpgoc5s | bwgozcurh |
| 1cyq ctry | p0re icwm |
| qqdz 6gu6 | nqqs d79f |
| 0lek xtax | umt7 zje4 |
| 78as1foz2mg | 6c1apobxk |
| 2ndvntne21 | ibo7 5q10 |
| | s5yx eh9e |
| | ptx8 a479 |
| | uvzm h |
| | c 512m |
| | o6 ml1m |
| | nbohzjiiw |

Date: 17-Nov-2006 11:00:58 -0500
 From: "MP" <ojwmkqpkjg@owenstewart.com>
 To: me
 Subject: Viewer Manage view

| | | | | | |
|-------------------|----------------|------------------|------------------|-----------------|------------------|
| 27631308145417360 | 307287681786 | 71548135321170 | 343054 | 10756823513 | 72707635758730 |
| 63865732427520885 | 30780073083450 | 8101665817720702 | 68004385 | 48701832577472 | 470364002347335 |
| 11782 | 22657 | 64276 | 0768 | 0007 | 4630431642 |
| 71161 | 14005 | 64626 | 8654 | 7537 | 1803 |
| 40448 | 64835 | 20883 | 7245855412226466 | 5570 | 3126 |
| 56868 | 32527 | 44273 | 37710684244385 | 566572 | 64171 |
| 62216 | 37781 | 45280 | 366776180148 | 540767160432825 | 25446 |
| 48121 | 03438 | 14467 | 2654 | 7380 | 2626143882585433 |
| 42378 | 86763 | 55151 | 6740 | 42024 | 05724 |
| 81283 | 37463407431821 | 3033 | 6826 | 22780 | 3454 |
| 66248 | 153307651208 | 7650 | 4853 | 50860 | 5133 |
| | | | | 0282 | 18078 |
| | | | | 08674725827005 | 208568201314677 |
| | | | | 431647200725 | 37622271248486 |

○ URL 쪼개기

- type `http://www` then the following URL into your browser: `.somewebsite.com/page.html`
- 사용자가 바로 클릭할 수 없을 뿐 아니라, 필터를 통과할 가능성도 낮음

○ 자바스크립트 내장

- 메일 클라이언트 보안성 향상으로 실행 불가

○ 공백 제거

- `CockLargeMaryellen PlumpingDickJuliette`
- 통과시키긴 쉽지만 사람도 알아보기 어려움

○ 데이터 집합 자체에 대한 공격

– 정상 메일에 스팸 토큰을 일부 포함시켜 전송

- 헤더에 임의의 조합 문자열을 포함

- X-q0djq0dw9j: lkej lwk23 01 ofwj0 w0j 9 09jr320 j09jr32lnfdlkn lkf wef

– 이런 공격은 어렵고 노력이 매우 많이 필요함

- 사용자가 이런 메일을 스팸 처리하는 경우 무효화

- 극히 일부 사례로 관찰됨

○ 원리

- 필터가 메일링리스트를 신뢰한다는 점을 악용
- 순서
 - 메일링리스트를 일정 기간동안 관찰
 - 유효한 메일 주소 목록 수집
 - 베이지안 필터로 정상 메일 특징 분석
 - 필터를 통과하는 스팸을 만들어 발송
 - 한 두 번 정도는 써먹을 수 있음

○ 통짜 이미지

- 스팸머들이 매우 좋아하는 방식
 - 정상적인 통짜 이미지도 많으므로 통과가 쉬움
 - 이미지 조회 기록까지 서버에서 확인 가능
- 최근에는 메일 클라이언트에서 이미지 블럭
 - 허용한 주소에 한해서 이미지 출력
- 적은 양의 HTML에서도 특징을 찾을 수 있음
 - Untitled Document 같은 자동 생성된 문구

○ 임의의 쓰레기 텍스트 삽입

– 현상

- 아주 소설을 쓰는 경우도 있음
- 임의로 조합된 쓰레기 텍스트를 덧붙임
- URL에 임의의 문자열 추가

– 원리

- Decision Matrix에 스팸 토큰이 못 들어가도록 유도

– 해결책

- 처음 발견된 토큰은 0.4~0.5를 부여

필터의 정확도 향상 기법

○ HMM = Hidden Markov Model

– Markov Model

- 상태와 전이 확률이 알려진 모델

– Hidden Markov Model

- 관찰 결과를 보고 상태를 역으로 유추

– 음성 인식기 응용

- 신호 판독 후 정확하게 알 수 없는 단어 발생
- 앞 뒤 단어 나열된 것을 보고 확률로 단어 유추
- 정확도가 80%에서 95% 이상으로 향상

– 스팸 필터 응용

- 단어의 나열을 보고 스팸이냐 아니냐 역으로 유추
- 무수히 많은 단어 나열의 조합을 계산하기 어려움
- 5단어씩 끊어 복잡도를 낮추는 단순화된 기법 사용
 - n-gram 비슷한 결과. 저장 공간을 매우 많이 사용하는 문제 있음.

○ Calibration

- 신뢰도에 기반하여 가중치를 부여하는 방법
 - 이전에 많은 오류가 있었던 토큰은 제외
- 조정 방안
 - 최종 Decision Matrix 계산 결과를 조정
 - 신뢰도를 이용하여 개별 토큰의 확률을 조정
 - 아예 믿을 수 없는 토큰은 입력 값에서 제외
- 4개의 상태 변수 추가 활용
 - 토큰을 스팸으로 잘못 분류한 횟수
 - 토큰을 햄으로 잘못 분류한 횟수
 - 토큰을 스팸으로 정확히 분류한 횟수
 - 토큰을 햄으로 정확히 분류한 횟수
- $(1-K)$ 를 확률 몰타기나 Decision Matrix 입력으로 이용

$$K^S = 0.5 + \frac{M^S}{S(C^S + M^S)}$$

$$K^I = 0.5 + \frac{M^I}{S(C^I + M^I)}$$

$$K = \frac{K^S K^I}{(K^S K^I) + (1 - K^S)(1 - K^I)}$$

K^S, K^I : 치우친 정도 S : 스케일러. 0.5 ~ 2.0 사이

K : 0.5 이상 스팸으로 치우침, 0.5 이하 정상으로 치우침

C^S : 스팸으로 정확히 분류된 횟수 M^S : 스팸으로 잘못 분류된 횟수

C^I : 정상으로 정확히 분류된 횟수 M^I : 정상으로 잘못 분류된 횟수

○ Bayesian Noise Reduction

- 베이저안 필터를 잡음 제거에 이용
 - Mom Would Be Proud Try Viagra Now!
 - 0.60 0.34 0.71 0.20 0.91 0.99 0.99
 - 경우의 수가 너무 많으므로 적당히 단위를 맞춤
- 3개 토큰씩 확률을 체인으로 엮어 패턴화
 - 0.60_0.35_0.70 0.35_0.70_0.20 0.70_0.20_0.90
0.20_0.90_1.00 0.90_1.00_1.00
 - $P = (SH / TS) / (SH / TS + IH / TI)$
- 스팸이면 햄 토큰 무시, 햄이면 스팸 토큰 무시
- 응용
 - 네트워크 이벤트 필터링
 - 로그 마이닝

- 예방 접종 (Message Inoculation)
 - TOE 발생할 때 다른 사용자에게 전파
 - 허니팟에서 수집한 스팸을 미리 학습
 - 가짜 메일 주소를 뿌려놓고 스팸 수신
 - 스팸이 수신되는 순간 전송자 IP 전파, 차단
 - 이메일 수집기를 쓸모없게 만들어버림
- 분류 그룹 (Classification Groups)
 - 불확실성이 높을 때 그룹 내 다른 노드에게 질문
 - 다수가 스팸이라고 판정하는 경우 스팸으로 판정
 - 새로운 사용자의 필터가 학습이 덜 된 경우 유용함
- 실시간 블랙리스트 (Streamlined Blackhole List)
 - 공격 당한 네트워크 주소 갯수 기준으로 차단
 - 24시간이 지나면 차단 목록에서 제거

○ Fight Back!

- 스팸의 대량 발송 특징을 이용
- 메일에 포함된 URL을 HTTP GET 요청
 - 스팸이라면 DoS, DDoS 공격을 받게 될 것
 - 정상 메일은 소량의 트래픽만 발생
- DoS 공격 이용 가능성의 억제
 - 정상 사이트를 스팸으로 뿌려서 공격 시도 가능
 - URL 화이트리스트 작성
 - 자동으로 URL의 내용을 통계적으로 분석 (스팸?)

- Ending Spam, Jonathan A. Zdziarski
 - Bayesian Content Filtering And The Art Of Statistical Language Classification
- Bayesian Noise Reduction
 - <http://bnr.nuclearelephant.com/>