

베이지안 스팸 필터 과제

제출 기한: 2009년 11월 14일 밤 11시 59분

제출 방법: xeraph@nchovy.com 메일 전송

(바이너리 포함 시 자동 필터링 되므로 압축 후 확장자에 .rename을 붙여서 보내기 바랍니다.)

제출 파일: 주석(영문)이 잘 달린 소스 코드, 실행 가능한 바이너리

(자바의 경우 JAR로 패키징하고 Main-Class를 지정해서 java -jar로 바로 실행할 수 있도록 패키징 하시기 바랍니다. Maven이나 Ant로 개발 환경 셋팅한 경우는 추가 점수 있음.)

프로그래밍 언어: Java, C, C++, Python 중 택일

(단, 윈도우 개발 환경이어야 하며, C/C++의 경우 Visual Studio 2008로 빌드 가능하도록 솔루션 파일과 프로젝트 파일을 같이 보내주시기 바랍니다.)

과제 내용:

1. 메일 데이터는 spam, ham, spam_test, ham_test 디렉터리에 분류되어 있습니다.
 - A. Ham 데이터는 ham 디렉터리 이하 모든 파일을, Spam 데이터는 spam 디렉터리 이하 eml_x 확장자를 가진 모든 파일을 읽어들이십시오.
2. 인코딩(7bit, 8bit, binary, base64, quoted-printable 총 5가지)에 맞게 파싱합니다.
3. 토큰라이징을 하고 어휘집(corpus) 통계를 생성합니다.
 - A. '&', ',', '!', '.', 'w', 't', 'r', 'n', '<', '>', '-', '|', '=', 'W', 'N', '(', ')', ':' 은 모두 구분자로 취급하여 제거하고 단어 단위로 통계를 생성합니다.
4. Paul Graham의 Naïve Bayesian 알고리즘을 이용하여 각 단어별로 스팸 확률을 구하고, Decision Matrix에 가장 극단적인 값을 가진 15개 단어를 올린 후, 이를 이용하여 전체 메시지가 스팸일 확률을 계산합니다.
 - A. 이 때 각 단어의 스팸 확률이 1인 경우 0.99로, 0인 경우는 0.01로 보정합니다.
 - B. 전체 확률 값이 0.9보다 큰 경우 스팸으로 판정합니다.
5. 아래와 같은 순서대로 입력과 출력을 수행합니다.
 - A. Spam과 Ham 어휘집에 들어있는 전체 단어 수를 출력합니다.
 - B. spam_test와 ham_test를 각각 분석하여 정확도, 맞춘 수, 실패한 수를 출력합니다.
 - C. 메일 파일이 위치한 임의의 경로를 입력 받아서 Decision Matrix를 화면에 출력하고 (각 단어별로 스팸 확률이 표시되어야 함) 해당 메일의 스팸 확률과 스팸 여부 판정을 출력합니다.

기타 궁금한 사항이 있으면 메일로 질문 바랍니다.