E-Mail 시스템의 첨부파일 형식별 자동분류 및 스팸 제거 에이전트 설계

현영순*, 정옥란, 조동섭
이화여자대학교 과학기술대학원 컴퓨터학과
e-mail : {toyloveys, orchung, dscho}@ewha.ac.kr

Agent for File Format based Classification of the Attached File in E-Mail System

Young-Soon Hyun*, Ok-Ran Jeong, Dong-Sub Cho
Dept. of Computer Science & Engineering, Ewha Womans University

요 약

인터넷과 E-mail 의 사용자가 증가하게 되면서 대량의 메일을 송수신하는 경우, 메일에 대한 효율적 관리의 문제와 불필요한 메일에 대한 관리의 중요성이 부각되고 있다. 본 논문에서는 -mail 시스템의 첨부파일 형식별 자동분류 에이전트는 메일의 내용을 읽어 Keyword 를 검색, 추출한 뒤 불필요한 메일로 판단되는 경우 자동삭제 시키고 그렇지 않은 경우 카테고리별로 폴더를 생성하여 첨부파일 들을 형식별로 분류 시켜주는 E-mail 시스템의 첨부파일 형식별 자동분류 에이전트를 제안하였다. 수신된 메일을 일일이 확인하고 분류해야만 했던 기존의 시스템과는 달리 본 논문에서 제안하고자 하는 시스템을 이용했을 경우 노력과 시간을 절감하고 불필요한 메일에 의한 저장공간의 낭비감소와 첨부파일을 효과적으로 관리할 수 있다는 장점이 있다.

1. 서론

인터넷과 E-mail 의 사용자가 늘어남에 따라 대량의 E-mail 에 대한 관리 문제와 불필요한 메일에 대한 관리의 필요성이 점점 중요시되어지고 있다.

일반적인 메일 시스템들은 수신된 메일을 서버에 무분별하게 적재하는 방식이기 때문에 관리자가 메일을 확인하고 첨부되어 온 파일들을 분류, 저장하는데 번거로움이 많고 관리부담이 컸다.

또한 불필요하게 수신된 메일들은 저장공간의 낭비를 가져온다.

수신된 메일을 내용을 읽어 목적에 맞는 폴더로 첨부파일을 자동으로 분류해주는 에이전트를 사용함으로써 관리자의 업무부담을 줄이고, 첨부파일을 효과적으로 관리 할 수 있을 것이다.

본 논문에서는 E-mail 에 첨부되어 오는 파일들의 효과적인 관리를 위해 E-mail 시스템의 첨부파일 형식별 자동분류 에이전트를 제안하고자 한다.

E-mail 시스템의 첨부파일 형식별 자동분류 에이전트는 메일의 내용을 읽어 Keyword 를 검색, 추출한 뒤 불필요한 메일로 판단되는 경우 자동삭제 시키고 그렇지 않은 경우 카테고리별로 폴더를 생성하여 첨부파일들을 형식별로 분류 시켜준다.

본 논문의 구성은 다음과 같다. 2 절에서는 메일 시스템에 대한 기존의 연구에 대해 언급한다. 3 절에서는 제안하는 E-mail 시스템의 첨부파일 형식별 자동분류 에이전트의 전체 구성에 대해 설명하고, 4 절에서는 본 논문의 결론과 추후 연구 계획에 대해 언급한다.

2. 관련 연구

2.1 웹 기반 전자우편 시스템의 동작원리

현재 많은 회사에서 일반 사용자들에게 무료로 서비스하고 있는 웹 기반 전자우편 시스템의 동작원리를 나타내면 다음과 같다. 사용자는 일반 웹 브라우저를 통해 서버에 접속한 후 보내고자 할 내용을 입력하고 POST 방식으로 서버에 전송하면 서버는 전송받은 메시지를 CGI 프로그램에서 파싱한다. 파싱된 메시지는 연속적인 8-bit 를 4 개의 ASCII 문자로 변화시키는 Base64 인코딩 작업을 거친 후, sendmail

프로그램을 구동해서 목적지 서버로 전송하게 된다. 목적지 서버에 도착한 메시지는 서버에 저장되어 있다가 수신자가 웹 브라우저를 통해 서버에 접속해서 요청을 하게 되면 반대과정을 거쳐 수신자의 브라우저에 보여지게 된다[1][2][3].

웹 기반 전자우편 시스템은 사용자가 특정회사의 전자우편 클라이언트 프로그램을 구입하지 않아도 브라우저만 있는 환경이면 전자우편을 송수신할 수 있도록 함으로써 사용자에게 매우 편리한 환경을 제공한다. 이러한 환경은 기업이나 학교 등의 인트라넷 환경에 적용될 수 있으며 공동의 사용자 인터페이스를 제공하므로 업무 수행능력이나 소속감 등을 높이는 데에도 기여할 수 있다.

웹 기반 전자우편 시스템을 통하여 수신된 메일들은 다양한 내용을 포함한다. 기업에서 인트라넷 환경에 웹 기반 전자우편 시스템을 적용하여 사용할 경우 기업은 매우 많은 양의 메일을 수신할 것이고 그 내용 또한 방대할 것이다. 사용자가 보내는 방대한 양의 메일을 데이터베이스에 저장하기만 한다면 관리자의 입장에서는 메일들을 일일이 읽어보아야만 하는 번거로운 작업이 된다. 따라서 수신된 메일을 적절하게 분류하여 목적에 맞는 부서로 메일을 자동으로 보내주는 에이전트가 필요하다.

2.2 MIME(Multipurpose Internet Message Extensions)

이진파일 추가와 멀티미디어 지원을 추가하는 프로토콜로서 RFC 1521, 1522 에 기술되어 있다. 물론, 기존의 RFC 822 메시지 포맷과 호환되고, 이진파일 전송, 메시지 유형의 결정, 새로운 문자집합, 미래를 위한 성장지원 등을 포함하고 있다.

2.3 메시지 구조(MIME)

RFC822 메시지 구조에 추가한 헤더 필드로 수신자는 메시지가 MIME 구조인지 확인할 수 있으며, MIME 으로 해석하게 된다.

메일 헤더의 각 구성요소는 다음과 같다.

"Return-Path"는 수신자에게 메시지를 배달하는 마지막 메일 서버에서 덧붙이는 필드로 송신자로 메일을 다시 반환할 주소와 경로 등의 명확한 정보를 포함하고 있다.

"Received"는 편지가 배달되는 경로를 나타내며 "Received"가 한 줄 이상 나타나는 경우에는 메일 보내고 받는 서버 이외의 다른 서버들을 통과해온 것을 의미한다.

"From"은 보내는 사람의 주소를 의미한다.

"Reply-To"필드는 메시지를 처음 받은 메일 서버에서 추가하는 정보로 수신자가 메시지에 대한 답변을 회신 할 주소가 된다. 만약에 송신자가 이 필드를 비워둔다면 "From"필드의 주소로 회신하게 된다.

"To"는 수신자의 주소를 나타내며 "Subject"는 이메일의 제목, "X-mailer"는 송신자가 사용한 메일 클라이언트 프로그램, "Data"는 이메일이 보내진 날짜를 의미한다.

"Message-ID"는 해당 이메일에 지정된 식별 번호로 메일 서버가 메시지를 외부로 보내며 붙이는 일련번호로 해당 메시지가 어떤 컴퓨터에서 보내졌는지 알 수 있다.

"Content-Type"은 메일 본문이 어떤 형태인지 알려주는데 text/plain 은 일반 문자열을 사용한 본문이고, 일반 문자열과 여러 인코딩 방식이 섞여 있을 경우에, multipart/mixed 는 일반 문자열과 파일을 첨부하였을 때, multipart/alternative 는 같은 내용이 일반 문자열과 HTML 로 반복하여 선택하여 읽을 수 있는 경우, multipart/related 는 HTML 형식의 메시지를 보내며 배경그림을 첨부했을 때 사용된다.

"Content-Transfer-Encoding"은 본문이 인코딩된 방식을 표시하는데, 한글 메시지에 8 비트라고 표시되어 있으면 인코딩 없이 본문을 그대로 보낸 것이고 BASE64 라든가 Quoted printabledlfkrh 적혀 있으면 그런 방식으로 인코딩 했다는 의미이다[4].

2.4 Base64

Base64 인코딩은 RFC 2045 에 서술되어 있다. 이것은 7 비트와 quoted-printable 인코딩 유형이 적합하지 않은 파일을 인터넷을 통해 보내야 할 때 사용된다.

Base64 인코딩은 3 개의 옥텟(24 비트)을 갖게 되고, 그것을 4 개의 6 비트 블록으로 매핑한 다음 각각의 6 비트 블록을 64 문자 알파벳 중 한 문자로 표현한다. 이러한 매핑 때문에, base64 로 인코딩된 정보는 본래 데이터보다 약 3 분의 1 정도 길어진다.

(그림 1)은 인코딩과 패딩을 도식화한 것이다.

본래의 데이터 "GIF89"

1. 옥텟 스트림으로 변경한다.

010001110100100101000110001110000011100 1

2. 처음부터 6 비트 블록으로 나누고, 나머지가 6 비트가 아니면 그대로 둔다.

010001 110100 100101 000110 001110 000011 1001

3. 마지막 블록이 6 개 비트를 가질 때까지 0 을 패딩한다.

010001 110100 100101 000110 001110 000011 100100

4. 6 비트 블록을 base64 문자로 변경한다.

십진 표현

17	52	37	6	14	3	36

문자표현

R	0	1	G	O	D	K

5. 문자의 총 개수가 4 로 나누어 떨어질 때까지 스트링 마지막에 등호 표시를 패딩한다.

6. 최종 결과: **ROIGODK=**

(그림 1) 하나의 옥텟 스트림에 대한 base64 알고리즘 적용

2.5 Base64 디코딩

Base64 데이터를 디코딩할 때는 base64 알파벳에 있지 않은 행분리와 문자들은 무시한다. 만일 데이터에서 여백이나 다른 합법적이지 않은 문자를 만나게 되면 데이터가 손상되기 때문에 예외적인 것은 사용자에게 맡기는 방향으로 고려해야 할 것이다.

등호 표시는 데이터의 마지막인 경우를 제외하고 base64 로 인코딩된 데이터에서는 어떤 곳에서도 발생하지 않는다. 다른 것이 발생하는 경우 에러가 된다.

Base64 디코딩의 나머지는 인코딩 단계를 역으로 하는 것처럼 단순하다. 패딩 문자(=)는 디코딩 시에 제거되도록 확실히 점검한다[5].

3. E-Mail 시스템의 첨부파일 형식별 자동분류 에이전트의 설계

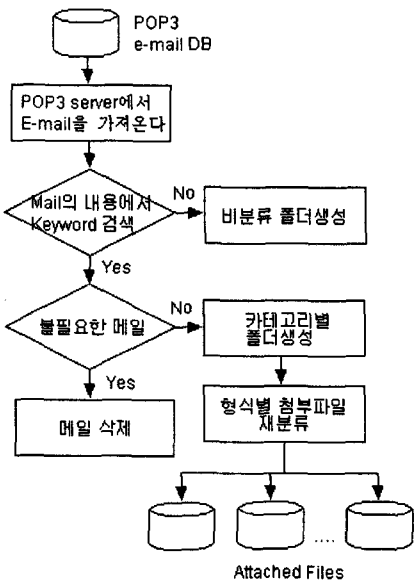
3.1 전체구조

본 논문에서 제시한 E-Mail 시스템의 첨부파일 형식별 자동분류 에이전트의 구조는 (그림 2)와 같다.

관리자는 POP3 를 이용하여 수신된 메일을 자신의 PC 로 가져오고, 메일의 내용을 읽어 미리 정해놓은 keyword 가 있는지 검색한다.

만약 불필요한 메일로 판단되는 keyword 가 검색되었을 경우 자동삭제 되도록 하고, 카테고리 별로 미리 정해놓은 keyword 가 검색되면 관련 폴더 내에 다시 한번 형식별로 첨부파일을 재분류 시킨다.

E-Mail 시스템의 첨부파일 형식별 자동분류 에이전트를 이용함으로써 대량의 첨부파일에 대한 관리를 효율적으로 할 수 있을 뿐만 아니라, 관리자의 업무부담을 줄이고 불필요한 메일로 인해 낭비되는 저장공간도 없앨 수 있다.



(그림 2) 에이전트 전체구성도

3.2 동작원리

본 논문에서 제시한 E-Mail 시스템의 첨부파일 형식별 자동분류 에이전트의 동작원리는 다음과 같다.

전송자가 보낸 메일과 첨부파일은 한 곳의 메일서버에 저장된다.

서버에 저장되어 있는 메일을 관리자가 POP3 서비스를 이용하여 관리자의 PC 로 가져온다. 메일의 내용을 텍스트 형식으로 읽어 들이고 미리 정해놓은 Keyword 를 검색한다.

메일의 Content-Transfer-Encoding 이 8bit 인 경우에는 인코딩 없이 본문을 그대로 전송한 것이므로 특별한 디코딩 작업 없이 keyword 를 검색한다.

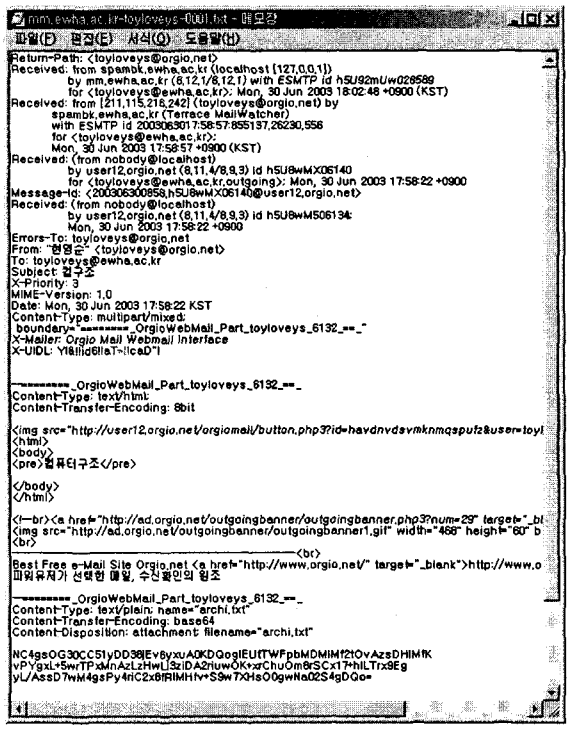
Content-Transfer-Endcoding 이 Base64 라든가 Quoted printable 이라고 적혀 있으면 그런 방식으로 인코딩 했다는 의미이므로 디코딩 작업이 필요하게 된다.

미리 정해놓은 keyword 가 발견되면 관리자의 PC 에 관련 폴더를 생성한다.

이때 메일의 제목이나 본문에 '광고' 등 불필요한 메일로 판단되는 keyword 가 발견될 경우 자동 삭제되어지도록 하고, 그렇지 않은 경우 첨부파일의 형식을 검색하여 관련 폴더 내에 형식별 폴더를 재생성하고 첨부파일을 분류, 저장하도록 한다.

Keyword 를 찾을 수 없으면 비분류 폴더를 생성하고, 첨부파일의 형식 검색 과정부터는 keyword 발견시와 동일하게 수행한다.

(그림 3)는 MIME 메시지 구조로 전송된 메일이다.



(그림 3) MIME 메시지 구조로 전송된 메일

3.3 분류과정

1. 수신된 메일을 POP3 서비스를 이용하여 관리자의 PC로 가져온다.
2. 메일의 내용을 읽어 미리 정해놓은 keyword가 있는지 검색한다.
3. 불필요한 메일로 판단되는 keyword 가 검색되면 자동삭제 되도록 하고, 그렇지 않은 경우 관련 폴더를 생성한다.
4. 첨부되어온 파일의 형식을 식별하여 관련 폴더 내에 첨부파일을 형식별로 재 분류한다.
5. keyword 를 찾을 수 없으면 비분류 폴더를 생성하고 첨부파일을 저장한다.

4. 결론 및 추후연구

본 논문에서 제안한 E-mail 시스템의 첨부파일 형식별 자동분류 에이전트는 keyword 검색을 통해 메일에 첨부되어 오는 파일들을 자동 분류함으로써 관리자의 업무부담을 줄이고, 불필요한 메일에 의한 저장공간의 낭비를 감소시켰다.

이는 대량의 메일을 송수신 해야 하는 경우 첨부파일 관리의 필요성이 많은 곳에서 유용하게 사용될 것으로 생각된다.

이 논문의 한계는 첨부파일을 분류하기 위한 방법으로 키워드를 미리 지정해 주는 데 있다. 이는 미리 지정해놓은 키워드를 포함하지 않은 문서에 대해서는 분류가 어렵다는 단점을 가지고 있다. 따라서 성능 향상을 위하여 문서들의 특징을 전혀 모르는 상황에서도 문서 내용에서 공통된 패턴을 발견하고 문서를 분류할 수 있는 자동 문서 분류에 대한 연구가 필요할 것이다[6][7][8].

참고문헌

[1] Stallings, W, Network and Internetwork Security : Principles and Practice. Prentice Hall, 1995.

[2] 박동욱, 박재희, 김진상, 김일민. "PGP 방식을 이용한 웹 기반 전자우편 보안 시스템," 한국정보처리학회 논문지 C, 2001.2.

[3] Sol, S. and Berznieks, G., CGI/PERL : Web Scripts. M&T Books, 1997.

[4] C. Buckley, G. Salton and J. Allan "The Effect of Adding Relevance Information in a Relevance Feedback Environment, "Proc. 17th ACM SIGIR International Conference on Research and Development in Information Retrieval, pp.292-298, 1994

[5] David Wood. "Internet Email Programming," O' REILLY.

[6] C. Buckley, G. Salton and J. Allan "The Effect of Adding Relevance Information in a Relevance Feedback Environment," Proc. 17th ACM SIGIR International Conference on Research and Development in Information Retrieval, pp.292-298, 1994.

[7] C.J. van Rijsbergen, Information Retrieval, Butterworths, London. 2nd Edition, 1979.

[8] G. Salton, and M.J. McGill, Introduction to Modern Informaion Retrieval, McGraw-Hill, New York. 1983.