문서 자동요약 기술을 적용한 클라우드 스토리지 기반 지능적 아카이빙 시스템†

(Cloud storage-based intelligent archiving system applying automatic document summarization)

유 기 동*
(Keedong Yoo)

요 약 제로 클라이언트 체제는 기업의 문서 중앙화를 위해 제시된 최신의 방법이며, 이를 기업의 현실에 보다 적합하도록 토착화시키는 작업은 필수적이다. 본 연구는 제로 클라이언트 체제의 잘 알려진 보안 및 프라이버시 상의 문제점을 제외한, 사용 상의 문제점을 보완하고자 고안되었다. 즉, 작업자가 작업한 문서를 기업 클라우드 스토리지에 저장하고자 저장 카테고리를 검색하는 과정의 부담과 정확성을 향상시키기 위하여, 작업 문서의 주제어를 자동으로 파악하고, 이를 바탕으로 해당 문서가 저장되어야 하는 카테고리를 자동으로 검색하여 작업자의 확인을 통해 자동 저장되도록 하는, 지능적 아카이빙 방식을 제시한다. 본 연구에서 제시하는 주제어 자동 파악 및 자동 아카이빙을 위한 방법론과 프로토타입 시스템은 기업 환경에 적용이 가능할 정도로 정확성과 확장성을 갖추고 있다.

핵심주제어 : 지능적 아카이빙, 주제어 파악, 클라우드 스토리지, 제로 클라이언트

Abstract Zero client-based cloud storage technology is gaining much interest as a tool to centralized management of organizational documents nowadays. Besides the well-known cloud storage's defects such as security and privacy protection, users of the zero client-based cloud storage point out the difficulty in browsing and selecting the storage category because of its diversity and complexity. To resolve this problem, this study proposes a method of intelligent document archiving by applying an algorithm-based automatic topic identification technology. Without user's direct definition of category to store the working document, the proposed methodology and prototype enable the working documents to be automatically archived into the predefined categories according to the extracted topic. Based on the proposed ideas, more effective and efficient centralized management of electronic documents can be achieved.

Key Words : Intelligent Archiving, Topic identification, Cloud storage, Zero client

1. 서 론

최근에 들어 기업의 문서 보안 및 중앙화를 위한 노력의 산물 중 하나로 관심을 얻고 있는 이른바 '깡통 PC', 즉 '제로 클라이언트' 기술은, '데스크탑 가상화(Virtual Desktop Infrastructure, VDI)' 기술을 클라우드 스토리지 기술과 접목시켜 기업의 작업 환경을

† 이 논문은 2011-2012년도 정보(교육과학기술부)의 재원으로 한국연구재단 기초연구사업의 지원을 받아 수행된 연구임(H00021).
* 단국대학교 경영학부, 제1저자, 교신저자

보다 효율적이고 효과적으로 개선하고자 고안된 방법이다. 데이터 센터 서버 내의 가상 데스크탑에 접속하여 데이터 및 운영체제, 그리고 응용프로그램 등을 활용하는 서버 기반 컴퓨팅은, 기업의 정보자원에 대한 보안 유지 및 관리가 용이하고, 사용자 단말기의 하드웨어적 특성에 유연하게 대응이 가능하며, 프로그램 업데이트 및 패치가 서버 내에서 이루어지므로 관리상의 이점 또한 매우 크다. 무엇보다도 VDI는 기업들의 가장 큰 고민인 기밀 유출을 방지할 수 있고, 특정 PC가 아닌 어떤 PC에서라도 내 데스크톱 업무 환경을 이용할 수 있는 접근성을 제공하며 노후화된 PC를 교체하지 않고 새로운 운영체제나 소프트웨어를 사용할 수 있다는 점이 특징이다. 또한 해외 사업장이나 지역사무소 등 원거리 업무 현장의 PC 사용 문제를 해결할 수 있다는 점에서도 VDI가 환영을 받았다. 제로클라이언트는 이러한 VDI의 구현 목적 중 보안과 비용 절감에 더욱 초점을 맞추고 있는 것으로 알려져 있다.

제로 클라이언트 기술을 기업 문서 중앙화 측면으로 적용할 경우 핵심 요소로 인식되는 클라우드 스토리지 기술은 클라우드 컴퓨팅 서비스 분류 중 'IaaS', 즉 'Infrastructure as a Service' 유형에 속하는 대표적인 기술이다 (김미점, 2010). 사전 정의된 기업 문서 분류체계를 기반으로 구성되는 클라우드 스토리지는 기업 문서 보안 및 중앙화를 위하여 최근 많은 기업들이 도입하고 있다. 클라우드 스토리지의 가장 특징적인 장점은 클라우드 컴퓨팅 환경 하에 관리되는 정보 및 지식에 대해 사용자가 언제 어디서든 접근이 가능한 점이다 (Liu et al., 2011). 대표적인 상용화 클라우드 스토리지 서비스의 예는 외국의 경우 Amazon S3, Mosso, Wuala 등이 있으며, 국내의 경우 유클라우드, N드라이브 등이 있다.

클라우드 스토리지 기반 기업 문서 관리는 기업 내 존재하는 모든 정보 및 지식을 중앙화하여 정보 및 지식의 유출을 방지하고 이들에 대한 집중적인 관리를 가능하게 하는 장점이 있으나, 한편으로는 사용성의 문제점, 즉 클라우드 스토리지 저장 및 분류 체계의 복잡성으로 인한 사용자의 부담 및 혼란이 있다 (나문성 외, 2010). 즉, 기업 문서관리 시스템이 갖는 많은 수의 저장 및 분류 카테고리(Category)를 문서 작업자가 제대로 이해하지 못하거나 또는 생소한 경우 해당 문서를 어느 범주 하에 저장하여야 하는가를 결정하기 어렵다. 또한 많은 수의 카테고리를 작업자가 모두 기억할 수 없으므로, 작업 문서의 저장 카테고리를 결정할 때마다 카테고리를 반복적으로 검색해야 하는 불편함이 있다. 물론 자주 접근한 카테고리 또는 전체 카테고리의 구성에 대해 익숙한 작업자의 경우 이러한 문제가 발생되지 않을 수 있지만, 기존에는 다루지 않던 주제의 문서를 저장하는 경우 또는 새로운 주제의 자료를 검색 및 추출하는 경우에는 여전히 많은 사용자의 혼란을 가중시킬 수 있다. 따라서 작업 중인 문서의 내용을 파악하여 해당 문서가 저장될 수 있는 카테고리를 시스템이 자동으로 제안할 수 있는 기능이 필요하다. 즉, 작업자가 문서 작업을 완료하고 저장 버튼을 선택하는 순간 해당 문서의 내용을 자동으로 분석하여 키워드 및 주제문구 등을 추출하고 이를 작업자에게 게시하여 해당 문서 저장 카테고리의 결정을 지원한다. 시스템이 제안한 저장 카테고리에 이상이 없어 작업자의 승인 또는 확인을 득한 경우 자동으로 해당 문서에 해당 키워드 또는 주제문구를 태깅(Tagging)하여 해당 카테고리에 문서를 저장한다.

본 논문은 저장하고자 하는 문서의 내용을 분석하여 주제어를 추출하고 이를 문서에 태깅하여 저장하는 문서 아카이빙의 전체 과정을 자동화하는 방법론을 제시하고자 한다. 기존의 문서 아카이빙은 사람에 의해 정의된 키워드를 저장 카테고리에 매칭시켜 해당 문서를 정의된 카테고리에 저장하는 부분적으로 자동화된 방식으로 진행되었다면, 본 연구에서 제시하는 방법은 해당 문서의 키워드 및 주제문구를 시스템이 자동으로 추출하고 이를 기존 저장 카테고리와 비교하여 저장 위치를 자동으로 결정하는 '지능적인' 아카이빙 방식이다. 제시된 방법론의 유효성을 입증하기 위하여 프로토타입 시스템인 'ASICAS (Automatic Summarization-based Intelligent Cloud Archiving System)'을 구현하였다.

2. 관련 연구

2.1 Thin Client vs. Zero Client

'Thin Client'는 CD-ROM 드라이브, 디스켓 드라이브 및 확장 슬롯 등이 없이 오직 필수적인 장치들로

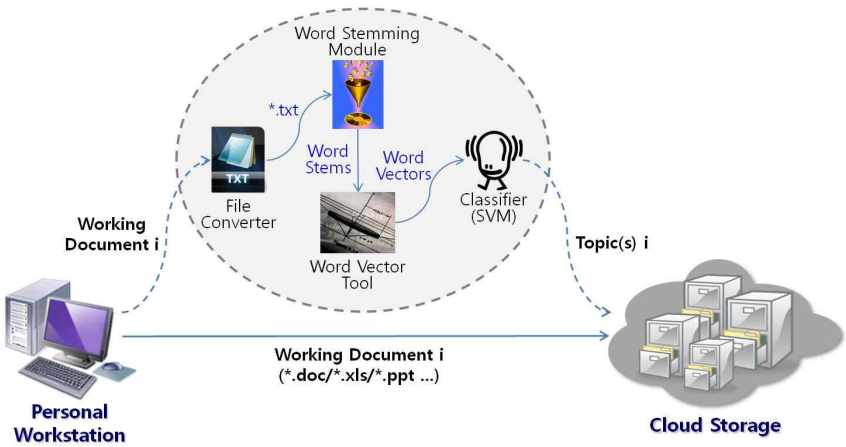
서 구성되어 중앙에서 관리할 수 있도록 설계된 업무용 PC로서, 넷PC 또는 네트워크 컴퓨터 등을 지칭하는 용어이다. 용어적인 정의는 '필수적 기능을 수행하는 장치와 애플리케이션만을 장착한 컴퓨터'로 정리될 수 있으나, 보다 광범위한 정의는, 최소한의, 그러나 필수적인 요소만을 갖춘 개인용 단말기를 강력한 기능과 권한을 가진 중앙 서버를 통하여 고차원적 관리하는 통합된 시스템 체제라 할 수 있다. 기업에 고성능 PC가 보급되면서, 애플리케이션의 설치나 업그레이드 또는 패치 작업이 진행되는 하드웨어의 유지보수에 드는 운용 및 관리 비용(TCO)을 무시 할 수 없게 되었다. 따라서 일반 작업자가 사용하는 클라이언트 컴퓨터로 고가의 고성능 PC를 사용하지 않고, 디스플레이 및 입력 등 최소한의 기능을 가진 저가의 전용 컴퓨터를 배치하고 애플리케이션 등의 자원은 서버에서 일원적으로 관리하여 운용 및 관리 비용의 절감을 유도하는 방안이 등장하였다 (송민규, 2009).

Thin Client 방식은 기본적으로 Client를 관할하는 Server의 컴퓨팅 자원, 즉 연산 및 저장 능력을 연결된 네트워크를 통하여 사용하는 방식으로, Client는 네트워킹과 최소한의 작동을 보장하는 운영체제 및 저장공간을 갖고 있다. 여기서 한 단계 더 나아가는 'Zero Client' 또는 'Ultra-thin Client' 방식은 운영체제조차 서버로부터 차용하여 사용하는데, 즉 커널(Kernel)이 운영체제를 대신하여 네트워크 초기화, 네트워킹 프로토콜 시동, 그리고 서버로부터 전송되는 결과값 디스플레이를 처리하는 방식이다. 따라서 Zero Client 방식은 과거의 'Dummy Terminal'과 매우 유사한 기능 및 특성을 갖고 있으며, 작업 문서 중앙 관리 및 보안 유지의 용이성으로 인하여 최근에 들어 많은 기업이 도입하고 있다. 메인보드에 임베디드 CPU와 플레시메모리 등을 장착하여 초박형으로 구성된 단말기를 사용하므로, 중앙처리장치, 메모리, 하드디스크 등의 부품없이 네트워크 접속과 주변기기 지원만 가능하다. 따라서 중앙처리장치, 메모리, 하드디스크 등을 장착한 Thin Client 단말기에 비해 Zero Client 단말기는 서버의 자원에 전적으로 의존한 컴퓨팅을 수행한다.

2.2 자동요약 (Automatic Summarization)

자동요약은 컴퓨터 프로그램을 이용하여 문서의 축약본을 산출하는 과정이다. 많은 종류의 다양한 정보로 인하여 이들 정보의 내용을 함축적으로 정리한 요약문은 매우 필수적이다. 이러한 자동요약 기술을 적용한 예로는 구글의 검색엔진이 대표적이다. 논리 정연하고 이해가 쉬운 요약문을 만들 수 있는 기술은 대부분, 사용할만한 요약문을 만들기 위하여 문장의 길이, 문장 스타일, 그리고 구문법 등을 함께 고려하여야 한다.

자동요약은 텍스트 기반의 문서 또는 문서의 광대한 코퍼스를 문서의 주된 의미를 포함하는 짧은 단어 또는 문구로 축약한다. 자동요약을 수행하는 방식은 크게 두 가지로 구분되는데, 문서의 주제문구 또는 주제어를 산출해내는 유형에 따라 '추출식(Exractive)'과 '추상식(Abstractive)'으로 나뉜다. 추출식 방법은 대상



<그림 1> 문서 주제어 자동 파악을 통한 클라우드 스토리지 기반 지능적 아카이빙 개념도

문서에 존재하는 단어, 문구, 또는 문장을 선택해내는 방식으로 요약을 수행한다. 반면 추상식 방법은 문서 내의 의미론적 표현(Semantic representation)을 만들어 자연어 생성 기술을 사용하여 요약을 수행하므로, 대상 문서에 존재하지 않는 단어 또는 문구를 만들어낸다. 추상식 방법은 최근 들어 대두된 방법이나 적용이 복잡하고 성능 또한 사전 학습 정도에 따라 다소 가변적이므로, 대부분의 연구들은 추출식 방법을 활용한다.

기존의 연구에서 주로 소개된 추출식 방법은 두 가지로 나뉠 수 있는데, 문서 태그에 사용되는 단어 또는 구절을 선택하기 위한 '주제문구 추출(Keyphrase extraction)'과 짧은 단락 형식의 요약문을 만들기 위해 문장 전체를 선택하는 '문서요약(Document summarization)'이다. 이 두 가지 방법 모두 '지도적(Supervised)' 방식과 '비지도적(Unsupervised)' 방식으로 진행될 수 있는데, 지도적 방식은 충분한 훈련(Training) 데이터를 바탕으로 사전 학습이 이루어져야 하는 면에서 비지도적 방법보다 비효율적일 수 있다. 지도적 방식에서 사용하는 대표적인 알고리듬으로는 문서 또는 텍스트 분류에서 탁월한 성능을 발휘하는 것으로 알려진 'SVM (Support Vector Machine)'이 있으며, 비지도적 방식에서는 최근 많은 관심을 받고 있는 'TextRank' (Mihalcea & Tarau, 2004)가 있다.

3. 방법론

<그림 1>은 본 연구에서 제안하는 클라우드 스토리지 기반 지능적 아카이빙의 개념도이다. 작업자의 워크스테이션에서 작성된 문서가 클라우드 스토리지에 저장될 때 해당 문서의 주제어(Topic) 파악이 동시에 진행된다. 파악된 주제어를 메타데이터로 하여 해당 문서를 클라우드 스토리지에 자동으로 저장할 수 있으며, 이를 통해 작업자에 의한 정의가 필요 없는 이른바 문서의 지능적 아카이빙이 가능해진다. 구체적인 절차는 다음과 같다.

- **작업 파일 형식 변환 (Converting)**

보통의 작업문서는 다양한 애플리케이션에 의해 작성되므로 이를 주제어 분석이 가능한 형식으로 변환한다. 분석이 가능한 파일 형식은 '텍스트(*.txt)' 형식이 일반적이므로, 모든 작업문서의 형식을 텍스트 형식으로 변환한다.

- **어근 추출 (Word Stemming)**

작업문서에는 다양한 문장성분의 단어들이 포함되어 있으므로 이 중 분석의 대상이 되는 단어들, 그리고 이들의 어근만 선별한다. 즉 복수 단어는 단수로, 과거형은 현재형으로, 부사 또는 형용사로 변환된 단어는 변환 이전의 어근만 선별하여 추출한다. 또한 대명사, 관사, 접두어/접미어 등은 삭제한다.

- **단어 벡터 (Word Vector) 생성**

추출된 단어의 어근을 이용하여 각 단어의 벡터를 생성한다. 분석 대상이 되는 단어의 벡터값을 이용하여 해당 문서의 벡터값이 결정되는데, 단어 벡터 생성을 위해 'TF/IDF (Term Frequency/Inverse Document Frequency)' 알고리듬을 이용한다. 이 때 단어 벡터 생성의 대상이 되는 기준 단어의 설정을 위해 기존의 문서 분류 체계, 즉 문서 저장 카테고리에서 사용하는 카테고리 명칭을 사용한다. 이러한 경우 기업 또는 도입 주체에서 정의하고 있는 문서 저장 카테고리에 맞추어 작업문서의 분류가 진행되므로, 추출된 주제어가 기업 또는 도입 주체에서 정의한 문서 저장 카테고리와 괴리되는 현상을 방지할 수 있다. 또한 이미 사용중인 카테고리 명칭을 기준 단어로 사용하므로, 작업문서 내에 포함된 단어들을 보다 수월하게 정의된 카테고리에 맞춰 정리할 수도 있다. 이러한 문서 저장 카테고리와 각 카테고리에 저장되어 있는 기존 문서들, 즉 'Corpus'를 이용하여 다음 단계인 Classifier의 사전 학습을 실시할 수 있다.

- **분류 및 예측 (Classification)**

추출된 문서의 벡터값을 기준 공간에 투영하여 해당 문서의 주제어를 최종적으로 결정한다. 본 연구에서는 텍스트마이닝 분야에서 매우 탁월한 성능을 발휘하는 기계학습 알고리듬으로 알려진 (Basu et al., 2002; Meyer et al., 2003; Dumais et al., 1998; Joachims, 1998; Rennie & Rifkin, 2001) SVM 알고리듬을 이용하여 Classifier를 구성한다. SVM은 사전학습에 의해 할당된 주제어별 공간에 해당 문서의 벡터값을 투영하여 주제어를 결정한다. 결정된 주제어에 이상이 없을 경우 해당 문서의 벡터값을 참고하여 기준 공간의 할당이 전체적으로 재조정되므로 과정이

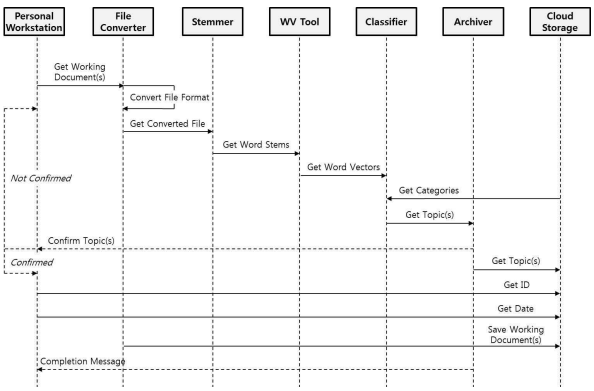
반복되어 데이터가 누적될수록 예측 성능이 향상된다. 벡터 공간의 구성을 위한 사전학습이 결과적인 성능에 직접적인 영향을 주므로 공식적이고 범용적인 Corpus가 필요한데, 아직까지는 공식적으로 인정되는 범용적인 Corpus는 존재하지 않으므로 일반적으로 'Reuter21578'과 같은 실험용 데이터를 사용하거나, 또는 적용하는 상황에 따라 직접 Corpus를 제작 및 수정하여 사용한다.

이상의 절차에 의해 자동 추출된 주제어는 해당 작업문서의 명칭으로 간주되므로, 추출된 주제어를 해당 작업문서에 태깅하는 간단한 프로그래밍을 통해 해당 문서의 자동 아카이빙이 완료된다. 동일 주제어를 갖는 다른 문서들이 존재할 수 있으므로, 작업자의 신분(ID 또는 직번) 정보와 문서 작업 완료시점(날짜 또는 시간) 정보를 함께 메타데이터로 태깅하여 인덱스화한다

4. ASICAS: 프로토타입 시스템

4.1 개요

본 연구에서 제시하는 주제어 자동 추출 기술을 적용한 클라우드 스토리지 기반 지능적 문서 아카이빙 방법론의 유효성 검증을 위하여 프로토타입 시스템인 ASICAS (Automatic Summarization-based Intelligent Cloud Archiving System)를 구현하였다.



<그림 2> ASICAS 작동 순서

ASICAS는 작업문서의 내용을 실시간 분석하여 해당 문서의 주제어를 자동 추출하고 문서작업자의 ID 및 작업완료 시간 정보와 함께 인덱스화하여 해당 문서를 클라우드 스토리지에 자동으로 저장한다. 즉, 문서작업이 완료되어 저장 버튼을 클릭하는 순간 해당 문서의 주제어 추출 분석이 자동으로 진행되어 작업자에게 추출된 주제어 및 저장 카테고리를 추천하고, 작업자의 저장 승인을 득하는 경우 해당 문서를 해당 카테고리에 자동으로 저장한다. [그림 2]는 ASICAS의 작동 절차를 보여준다.

ASICAS는 Java2 런타임 환경 하의 JDK v1.5.0_06을 이용하여 구현하였으며, 세부 모듈 중 'Stemmer'와 'WV Tool'은 KDD(Knowledge Discovery and Data Mining)와 기계학습을 위한 오픈 소스 환경인 'Yale' (Mierswa et al., 2006)의 'Word stemming tool'과 'Vector creating tool'을 활용하였다. 또한 Classifier인 SVM 모듈은 Chang & Lin(2001)에 의해 개발된 'LibSVM v2.81'을 사용하였다. ASICAS가 다루는 문제의 영역은 대학교 내에서 발생되는 각종 상황들로 정의하여 프로토타입 적용 범위를 한정하였는데, 이는 Classifier 사전학습의 부담을 최소화하되 그 기능성을 최대한 부각시키기 위한 조치이다. 단어 벡터 생성 및 Classifier의 사전학습을 위하여 사용한 사전 정의 카테고리는 메릴랜드 대학교 전산학과에서 정의한 'University Ontology'[1]를 참고하였다.

<표 1> ASICAS 기능 명세

주기능	세부기능	내용
작업문서 주제어 추출	Converting	문서 형식 변환
	Stemming	어근 추출
	Vectoring	벡터값 생성
	Classifying	주제어 추출
작업문서 자동 아카이빙	Indexing	Topic/ID/Date 태깅
	Searching	카테고리 검색
	Archiving	작업문서 저장
	Messaging	사용자 대화

이러한 사전 정의 카테고리는 작업문서 저장을 위한 카테고리로 활용됨과 동시에, 추출하는 주제어의 기준 단어로 활용되기도 한다. 즉, 작업문서에 포함된 여러 종류의 단어들 중 중요도가 높은 단어를 그대로 주제어로 추출하는 것이 아닌, 사전 정의된 카테고리에서 사용된 단어를 이용하여 작업문서의 주제어를 추출한다. 따라서 카테고리에서 사용된 단어들이 작업

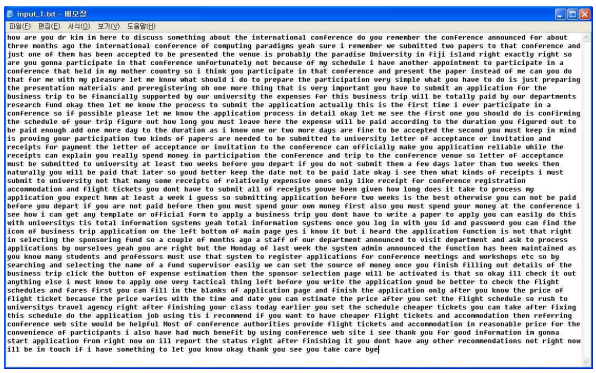
1) http://www.cs.umd.edu/projects/plus/SHOE/onts/univ1.0.html

문서의 내용을 대표하는 정도를 현실적이고 극대화시키기 위하여 충분한 수의 샘플 문서를 사용하여 SVM 모듈을 사전 학습시키는 과정이 필수적이다. ASICAS의 기능은 크게 작업문서 주제어 추출과 자동 아카이빙으로 구분된다. 세부적인 기능을 명세하면 다음의 <표 1>과 같다.

4.2 예제: 학회출장 준비에 관한 대화 녹취록

- **Converting: 문서 형식 변환**

ASICAS의 기능을 설명하기 위하여 '학회출장 준비'에 관한 두 화자 간의 대화를 녹취(Dictate)한 문서를 예제로 사용한다. ASICAS는 MS-Word(*.doc), MS-Excel(*.xls), 그리고 HTML 기반의 웹 페이지 문서 등을 텍스트(*.txt) 형식의 파일로 변환하여 분석을 시작한다. <그림 3>은 예제 문서를 텍스트 파일로 변환한 예이다.



<그림 3> 텍스트 형식으로 변환된 작업문서

- **Stemming: 어근 추출**

ASICAS의 Stemmer는 입력된 문서에 포함된 단어 중 분석 대상이 되는 단어의 어근만을 선별한다. 즉, 의미를 담고 있지 않는 단어(Stop word)를 제거하고 남은 분석 대상 단어들의 어근을 추출한다. <그림 4>는 Stemmer가 입력된 문서에서 추출한 분석 대상 단어들의 어근만 선별한 결과를 보여준다.

discuss	unfortun	expens	mind	accommod	main	finish	us
intern	schedul	total	prove	flight	page	fill	recommend
confer	appoint	paid	kind	ticket	heard	click	want
rememb	held	depart	letter	given	function	button	refer
announc	mother	research	invit	take	select	estim	site
month	countri	fund	receipt	expect	sponsor	activ	help
comput	think	okai	payment	guess	coupl	check	author
paradigm	instead	process	offici	includ	staff	els	provid
yeah	pleasur	actual	make	dailli	visit	tactic	reason
submit	know	time	reliabl	templat	mondai	fare	conveni
paper	prepar	pleas	explain	form	admin	blank	benefit
accept	simpl	detail	spend	appli	maintain	price	thank
present	materi	confirm	monei	write	student	vari	good
venu	regist	figur	week	easili	professor	rush	start
paradis	thing	leav	natur	inform	meet	travel	report
univers	applic	accord	date	system	workshop	agenc	statu
fiji	busi	durat	late	password	search	class	touch
island	trip	dai	rel	find	name	earlier	care
exactli	financi	fine	on	Icon	supervisor	cheaper	
particip	support	keep	registr	bottom	sourc	fix	

<그림 4> 문서 내 단어의 어근 추출 결과

- **Vectoring: 작업문서 벡터값 생성**

추출된 어근에 대해 TF/IDF 알고리듬을 적용하여 해당 문서의 벡터값을 생성한다. 이 때 해당 문서의 벡터값은 사전 정의된 카테고리를 기준으로 계산한다. 그러나, University Ontology는 온톨로지의 구현을 위해 정의된 카테고리이므로, 이를 그대로 주제어 추출의 기준 단어로 사용하지 않고 주제어의 특성을 고려하여 부분적으로 수정하는 것이 바람직하다. 즉, University Ontology의 내용 중 'Person'과 'Publication'이 상대적으로 상세화되어 있는데, 이는 다른 클래스에 비해 이 클라스들을 차별화하여 상세히 묘사하기 위함이다. 따라서 이를 주제어가 가져야 하는 의미상의 대표성 정도를 고려하여 [그림 5]와 같이 University Ontology를 기초로 선택 및 수정한 9개의 기준 단어를 사용한다. 이는 곧 선택된 9개의 기준 단어의 벡터값에 의해 작업문서의 벡터값을 생성함을 의미한다.

Category #0 --------> Admission
Category #1 --------> Course
Category #2 --------> Conference
Category #3 --------> Examination
Category #4 --------> Graduation
Category #5 --------> Payment
Category #6 --------> Publication
Category #7 --------> Research
Category #8 --------> Vacation

0:0 1:0.997223 2:-1 3:-1 4:-1 5:0.068952 6:-1 7:-0.92236 8:-1

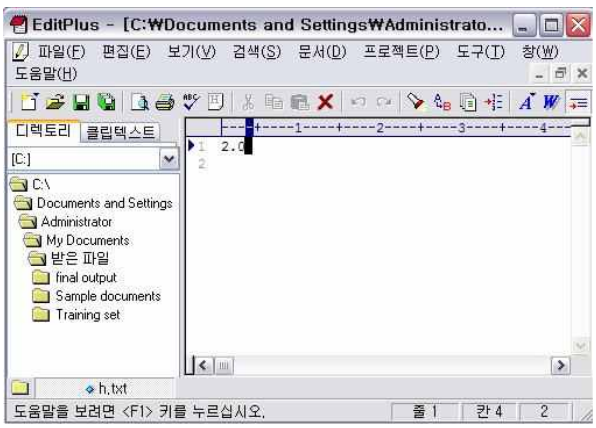
<그림 5> 벡터 생성 기준 단어 (상) 및 예제 문서의 최종적인 벡터값 (하)

- **Classifying: 주제어 추출**

생성된 문서 벡터를 9개의 기준 단어로 구성된 공간(9차원)에 투영하여 해당 문서의 주제어를 결정한다. 이를 위해서는 기준 단어와 실제 문서 내에 포함된 단어 간의 연관관계를 형성하기 위한 사전학습이 이

루어져야 한다. 사전학습을 위하여 9개의 기준 단어별로 약 80-90개 정도의 샘플 문서를 주제가 미리 정의되어 있는 웹페이지와 기존 문서를 통해 확보하여 사전 학습을 수행하였다. 주제 분류 예측(Category Prediction)의 정확도는 LibSVM를 통해 자동 계산되며, 본 연구의 사전 학습을 통한 예측 정확도는 총 40회의 테스트 중 37회가 옳은 분류값을 산출하여 92.5%의 만족할 수준의 정확도를 보였다 (MSE[2]=1.025,SCC[3]=0.825273).

해당 문서의 벡터값을 이용하여 추출된 최종 결과는 [그림 6]과 같다. 즉, 결과값은 기준 단어의 카테고리 차수로 표현되는데, 여기에서 '2.0'은 'Category #3'를 의미하여, 이는 곧 해당 문서의 주제어가 'Conference'임을 의미한다. 작업문서의 내용이 학회 출장 준비와 관련된 것이므로 제시된 기준 단어 중 'Conference'가 해당 문서의 주제어로 가장 적합하다.



<그림 6> 최종 결과 (주제어 카테고리 #2: Conference)

- **Indexing: Topic/ID/Date 태깅**

추출된 주제어는 작업문서의 저장 카테고리를 의미하므로 이를 해당 문서에 태깅하여 문서를 저장한다. 단, 동일 주제어를 갖는 또 다른 문서들이 존재할 수 있으므로, 해당 문서를 작성한 작업자의 ID와 작업완료 시점(Date)을 함께 태깅한다. 다음은 추출된 주제어를 문서 작성자 ID와 작업완료 Date와 함께 해당 문서에 태깅하는 프로그래밍 코드이다.

```
SimpleDateFormat dateFormat = new
SimpleDateFormat("yyyyMMdd", Locale.KOREA);
String recordedDate = dateFormat.format(new Date());
String tagName = topicName + "-" + userID + "-" +
recordedDate;
System.out.printf("Indexing tag is %s\n", tagName);
JOptionPane.showMessageDialog(null, tagName);
```

- **Searching: 카테고리 검색**

해당 문서에 대한 Indexing이 완료되면 사전 정의 카테고리([그림 3]) 중 저장 카테고리를 검색하여 저장 위치를 결정하여야 한다. 다음은 추출된 주제어에 해당되는 카테고리를 검색하고 결정하는 프로그래밍 코드이다.

```
HashMap<String, Integer> categoryHash = new
HashMap<String, Integer>();
categoryHash.put("Professor", 0);
categoryHash.put("Lecturer", 1);
categoryHash.put("PostDoc", 2);
categoryHash.put("ResearchAssistant", 3);
categoryHash.put("TeachingAssistant", 4);
categoryHash.put("Director", 5);
categoryHash.put("ClericalStaff", 6);
categoryHash.put("SystemsStaff", 7);
categoryHash.put("UndergraduateStudent", 8);
categoryHash.put("GraduateStudent", 9);
categoryHash.put("Department", 10);
categoryHash.put("Institute", 11);
categoryHash.put("Program", 12);
categoryHash.put("ResearchGroup", 13);
categoryHash.put("School", 14);
categoryHash.put("University", 15);
categoryHash.put("BookArticle", 16);
categoryHash.put("ConferencePaper", 17);
categoryHash.put("JournalArticle", 18);
categoryHash.put("WorkshopPaper", 19);
categoryHash.put("Book", 20);
categoryHash.put("Journal", 21);
categoryHash.put("Magazine", 22);
categoryHash.put("Proceedings", 23);
categoryHash.put("DoctoralThesis", 24);
categoryHash.put("MastersThesis", 25);
categoryHash.put("Course", 26);
categoryHash.put("Research", 27);
categoryHash.put("Schedule", 28);
categoryHash.put("Conference", 29);
```

2) Mean Squared Error
3) Squared Correlation Coefficient

```
int indexOfCategory = categoryHash.get(topicName);
System.out.printf("Searching result is %d index\n",
indexOfCategory);
JOptionPane.showMessageDialog(null,
indexOfCategory);
```

- **Archiving: 작업문서 저장**

작업문서의 인덱스 정보와 저장 카테고리가 결정되면 해당 문서의 명칭을 '주제어-ID-Date'로 설정하여 저장을 완료한다. 다음은 작업문서의 명칭을 '주제어-ID-Date'로 설정하여 주제어와 일치되는 카테고리에 저장하는 프로그래밍 코드이다.

```
try {
BufferedWriter tagFile = new BufferedWriter(new
FileWriter(filePath));
tagFile.write(tagName);
tagFile.close();
} catch (IOException e) {
System.err.println(e);
System.exit(1);
}
JOptionPane.showMessageDialog(null, "The document
is to be saved as '" + filePath + "'");
```

- **Messaging: 사용자 대화**

ASICAS와 사용자 간 이루어지는 대화는 두 종류로, 하나는 추출된 주제어가 맞는지 확인하기 위한 대화이고, 나머지는 어느 카테고리 하에 어떠한 명칭으로 작업문서가 최종 저장되었음을 알려주는 대화이다. 본 예제에서는 추출된 주제어 'Conference'가 해당 문서의 주제어가 맞는지 확인하는 대화(<그림 7>의 (a))와 해당 문서를 'Conference' 카테고리에 'Conference - ABCD00001 - 2012xxxx'이라는 명칭으로 저장하였음을 알려주는 대화(<그림 7>의 (e))가 이루어진다. 이러한 사용자 대화를 위한 프로그래밍 코드는 다음과 같다.

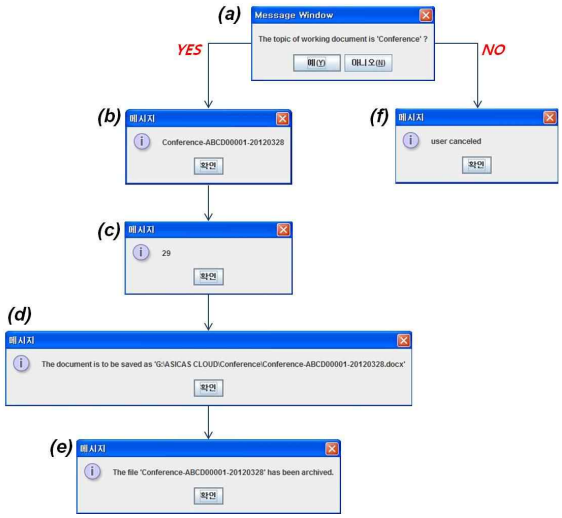
```
String msg = "The topic of working document is '" +
topicName + "' ?";
int ret = JOptionPane.showOptionDialog(null, msg,
"Message Window", JOptionPane.YES_NO_OPTION,
JOptionPane.PLAIN_MESSAGE, null, null, null);
switch (ret) {
case JOptionPane.YES_OPTION:
JOptionPane.showMessageDialog(null, "The file '" +
tagName + "' has been archived.");
break;

case JOptionPane.NO_OPTION:
JOptionPane.showMessageDialog(null, "user canceled");
break;
}
```

이상의 기능이 수행됨을 확인할 수 있도록 구현한 사용자 확인 창은 [그림 7]과 같다. 실제 ASICAS의 작동 시 사용자에게 제시되는 대화 메시지 창은 (a)와 (e)이고, 그 외의 창은 작동 절차를 확인하기 위해 임시적으로 구현되었다.

5. 결 론

문서 중앙화를 위한 기업의 노력은 기업 내 지적 자산의 보안 유지를 위한 시도임과 동시에 자원의 집중적인 관리를 통하여 업무의 효율성 및 효과성을 제



<그림 7> ASICAS 작동 확인 메시지 창

(a) 추출된 주제어 확인 메시지
(b) 주제어가 맞을 경우 파일 Indexing 결과
(c) 저장 카테고리 검색 및 결정 결과
(d) 저장 카테고리 및 파일 명칭
(e) 저장 완료 메시지
(f) 주제어가 틀릴 경우 작동 정지 메시지)

고하기 위한 방안이다. 제로 클라이언트 체제는 이러한 문서 중앙화를 위해 제시된 최신의 방법이며, 이를 기업의 현실에 보다 적합하도록 토착화시키는 작업은 필수적이다.

본 연구는 클라우드 스토리지를 활용하여 제로 클라이언트 체제가 갖고 있는 사용 상의 문제점을 보완하고자 고안되었다. 즉, 작업자가 작업한 문서를 기업 클라우드 스토리지에 저장하고자 저장 카테고리를 검색하는 과정의 부담과 정확성을 향상시키기 위하여, 작업 문서의 주제어를 자동으로 파악하고, 이를 바탕으로 해당 문서가 저장되어야 하는 카테고리를 자동으로 검색하여 작업자의 확인을 통해 자동 저장되도록 하는, 지능적 아카이빙 방식을 제시한다. 본 연구에서 제시하는 주제어 자동 파악 및 자동 아카이빙을 위한 방법론과 프로토타입 시스템은 바로 기업 환경에 적용이 가능할 정도로 정확성과 확장성을 갖추고 있다.

본 연구에서 주제어 자동 파악을 위해 적용된 SVM 알고리듬은 이미 텍스트마이닝 및 문서분류 분야에서 성능이 입증된 방법이다. 그러나 SVM 알고리듬을 이용한 주제어 추출의 정확성을 보장하기 위해서는 충분한 사전 학습을 위해 충분한 양의 샘플 데이터와 표준화된 사전 정의 카테고리가 필요하다. 이는 실제로 SVM 기반의 Classifier를 구성하는 데에 적지 않은 부담으로 작용하므로 실제 기업 환경 적용에 있어 걸림돌이 될 수도 있다. 또한 사용하는 언어에 따라 샘플 데이터와 사전 정의 카테고리를 별도로 구성하여야 하므로 다국어 기반의 문서를 다루는 기업에 있어서는 현실적이지 못한 방법이 될 수 있다. 이에 TextRank 알고리듬과 같이 사전 학습 과정이 필요하지 않고 다국어 기반의 문서를 처리할 수 있는 'Unsupervised Learning' 알고리듬을 적용하는 연구가 필요하다. 또한 이를 SVM과 같은 'Supervised Learning' 방식과 결과적인 성능 면에서 비교하는 연구 또한 의미가 있다.

본 연구에서 사전 정의 카테고리로 사용한 'University Ontology'와 사전 학습을 위해 사용한 샘플 문서들은 연구의 목적에 맞추어 수정된, 공식적으로 인정되지 못한 코퍼스를 사용하므로 이를 바탕으로 도출된 결과 또한 완전히 신뢰할 만한 수준의 것은 아니다. 온톨로지를 직접 구현하는 것이 바람직하나, 모든 상황을 포괄하는 온톨로지의 정의는 매우 방대한 작업이므로, 본 연구에서는 제시된 방법론의 타당성을 입증할 수 있는 정도의 온톨로지를 임시적으로 구성하였다. 본 연구에서 구현된 프로토타입의 주제어 예측 성능이 100% 수준까지 도달하지 않은 것 또한 이러한 이유에 기인하는데, 이는 추후 보다 공인된 온톨로지 카테고리와 코퍼스의 정의가 완료됨에 따라 보완될 것으로 기대한다. 매우 세분화된 특정 분야의 온톨로지 카테고리를 파악하는 정도의 작업일 지라도 수년의 시간을 요하는 작업이므로, 향후 어느 정도 신뢰할 만한 규모의 분류체계가 수립된다면 더욱 정확한 학습과 예측(추론)이 가능할 것으로 예상한다.

참 고 문 헌

[1] 김미점, "클라우드 스토리지 시스템 기술 고찰", 정보과학논문지, Vol.28 No.12, 50-58, 2010.

[2] 나문성, 김승훈, 이재동, "클라우드 환경에서 대규모 콘텐츠를 위한 효율적인 자원처리 기법", 한국산업정보학회논문지, Vol.15 No.4, 17-27, 2010.

[3] 송민규, "애플리케이션 공유 및 데이터 접근 최적화를 위한 씬-클라이언트 프레임워크 설계", 한국산업정보학회논문지, Vol.14 No.5, 19-32, 2009.

[4] 유기동, "지식근로자의 상황정보를 이용한 자율적 지식획득 방법론: 대화형 지식의 획득을 위한 차세대형 지식경영시스템", 지식경영연구, Vol.9 No.4, 65-75, 2008.

[5] Dumais, S., Platt, J., Heckman, D., & Sahami, M., "Inductive learning algorithms and representations for text categorization", Proceedings of the 7th International Conference on Information and Knowledge Management, 1998.

[6] Joachims, T., "Text categorization with support vector machines: Learning with many relevant features", Proceedings of the European Conference on Machine Learning, 1998.

[7] Liu, Q., Wang, G., & Wu, J., "Secure and privacy preserving keyword searching for cloud storage services", Journal of Network and Computer Applications, in press, 2011.

[8] Mierswa, I., Wurst, M., Klinkenberg, R., Scholz,

M., & Euler, T., “YALE: Rapid Prototyping for Complex Data Mining Tasks”, Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-06), 2006.
[9] Mihalcea, R. & Tarau, P., “TextRank: Bringing Order into Texts”, Proceedings of the Conference on Empirical Methods in Natural Language Processing, 404-411, 2004.
[10] Rennie, J.D.M. & Rifkin, R., “Improving multiclass text classification with the support vector machine”, CBCL Paper #210/AI Memo #2001-026, Massachusetts Institute of Technology, October 2001.
[11] Yoo, K., “Automatic Document Archiving For Cloud Storage Using Text Mining-Based Topic Identification Technique", Proceedings of the 2012 International Conference on Information and Computer Applications, 2012.

유 기 동 (Keedong Yoo)

- 정회원
- POSTECH 산업공학과 공학사
- POSTECH 산업공학과 공학석사
- POSTECH 산업경영공학과 공학박사
- 단국대학교 경상대학 경영학부 조교수
- 관심분야 : 지식경영 및 지식관리시스템, 유비쿼터스 컴퓨팅, 차세대형 경영정보시스템, 컨텍스트 기반 자율적 컴퓨팅, 지능적 지식 서비스, 정보전략 기획 및 성과평가

논 문 접 수 일 : 2012년 03월 29일
1차수정완료일 : 2012년 04월 09일
2차수정완료일 : 2012년 04월 20일
게 재 확 정 일 : 2012년 04월 23일