Digital Contents Society Review Vol. 1. No. 1. October. 2005(pp. 73-85)

웹 문서 군집화: 기술 현황

김재훈* 박은진* 옥철영**

1. 서론

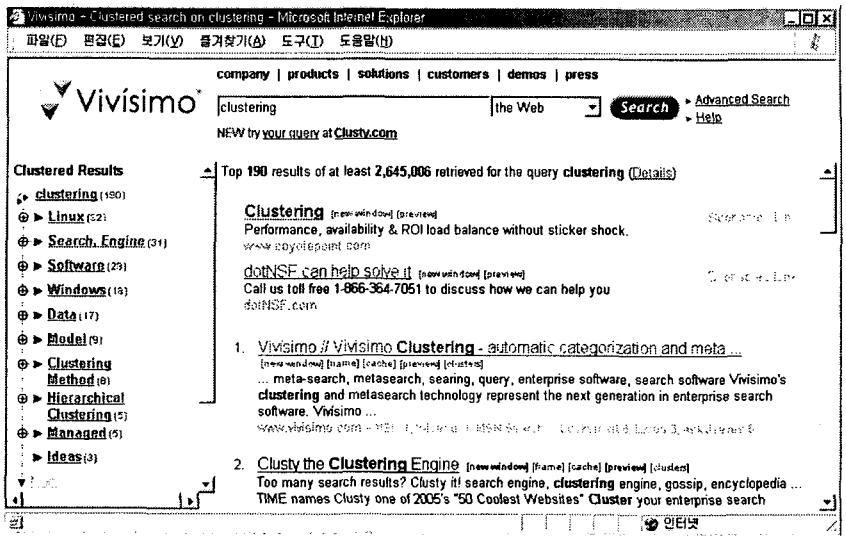
본 논문은 인터넷 정보검색(Internet *information retrieval*)의 문제점을 살펴보고 그 문제점을 해결하는 방안 중 하나로 웹 문서 군집화(Web document clustering)에 대해서 기술한다.

정보검색(information retrieval)이란 대량의 문서로부터 사용자 질의(user query)에 가장 적합한 문서(relevant document)를 찾는 것이다. 적합한 문서란 다소 추상적인 개념이지만, 이를 구체화하고 객관화하고자 하는 노력이 정보검색 분야에서 꾸준히 행해지고 있다(TREC2005, SIGIR2005). 적합 문서(relevancy document)를 판단하는 기준은 여전히 주관적일 수밖에 없다. 적합 문서는 사용자 질의에 적합한 내용이나 제목이 포함되어야 하고, 시기적으로 적절해야 하며, 또한 믿을 만한 출처로부터 검색되어야 하고, 사용자가 의도한 정보요구(information need)를 만족해야 한다. 이와 같은 적합 문서를 찾아주는 시스템이 정보검색 시스템(information retrieval system)이며, 가장 간단한 정보검색 방법은 키워드 검색(keyword search)이다. 키워드 검색은 문서와 질의를 단어의 집합으로 가정하고, 사용자 질의를 포함하는 문서를 적합 문서로 가정하여 검색하는 방법이다.

한편, 웹 문서 검색(Web document search)은 일반적인 정보검색을 웹에 적용한 것이다. 즉 검색 대상 문서가 바로 웹 문서라는 것이다. 웹 문서 검색과 일반적인 정보검색과는 몇 가지의 다른 점이 있다. 첫째, 웹 문서 수집기(Web crawler)를 이용해서 문서가 수집된다. 둘째, 대부분은 문서는 HTML(HyperText Markup Language) 혹은 XML(Extensible Markup Language)과 같은 마크업 언어로 구조화되어 있다. 셋째, 문서들은 상호 참조(cross-reference)가 가능한 연결구조(link structure)로 되어 있다. 넷째, 대상문서들이 지속적으로 변한다. 웹 문서 검색 시스템을 일반적으로 검색엔진(*search engine*)라고 한다. 한국에서 널리 사용되는 검색엔진으로는 구글(Google)[1], 야후(Yahoo!)[2], 다음(Daum)[3], 네이버(Naver)[4], 엠파스(Empas)[5], 네이트(Nate)[6], 알타비스타(AltaVista)[7] 등이 있다. 이와 같은 검색엔진의 가장 큰 문제점은 검색 결과의 문서 수가 너무 많다는 것이다(Weiss, 2002; Zamir and Etzioni, 1998). 일반적으로 한 질의어에 대해서 수천 건의 문서가 된다. 즉, 대부분의 검색엔진은 높은 재현율(recall)을 가지나 정확률(precision)은 매우 낮다[8]. 그 다음 문

* 한국해양대학교 IT공학부
** 울산대학교 컴퓨터정보통신공학부

1) http://www.google.co.kr/
2) http://kr.yahoo.com/
3) http://www.daum.net/
4) http://www.naver.com/
5) http://empas.com/
6) http://www.nate.com/
7) http://www.altavista.com/
8) 재현율(recall)이란 검색엔진이 소장하고 있는 적합 문서 중에서 검색된 적합문헌의 비율을 말하며, 정확률이란 검색된 문서 중에서 적합 문서의 비율을 말한다. 예를 들어 재현율 80%라는 것은 검색엔진이 소장하고 있는 전체 10개의 적합 문서 중에서 8개의 적합



<그림 1> 웹 문서의 군집화 예

제는 동형이의어(homograph)가 구별되지 않는다는 것이다. 예를 들면, "은행"을 검색하면 일반적으로 금융기관이 검색되지만 은행(*ginkgo nut*)나 은행나무에 관련된 문서가 검색된다. 이와 같은 이유로 원하는 문서가 검색되었다 하더라도 원하는 문서를 검색된 목록에서 찾는 데는 많은 시간이 소요된다.

이 문제를 해결하는 방법으로 검색된 문서를 군집화하는 방법이 많이 사용된다. 검색된 문서가 군집화되면 의미적으로 유사한 문서들이 하나의 덩어리(cluster)를 형성함으로써 검색 결과를 좀더 빨리 훑어볼 수 있게 되어 많은 시간을 절약할 수 있게 된다. <그림 1>은 Vivísimo라는 사이트에서 "clustering"라는 질의에 대한 군집화 결과이다. 이 시스템은 군집화 결과에 적절한 이름을 붙여 메뉴로 제공하고 있으며, 사용자가 "clustering method"을 원한다면 일곱 번째 덩어리를 선택하면 바로 원하는 결과를 찾을 수 있을 것이다. 이 결과는 일반 검색엔진에서 제공하는 검색 결과와는 많은 차이를 보인다. 첫째, 검색 결과가 계층구조를 가진다. 둘째, 검색된 결과가 의미적으로 유사한 문서들끼리 한 군데 모여 있다. 셋째, 군집화 결과로 형성된 덩어리를 관찰함으로써 좀더 정확한 질의를 생성할 수 있다.

Vivísimo 이외에도 현재 웹에서 서비스를 제공하는 시스템으로는 Clusty[1], KartOO[2], iBoogie[3], Mooter[4], WebClust[5] 등이 있다. 이들 시스템에 대한 특징과 구체적인 소개는 4장에서 기술할 것이다.

본 논문의 구성은 다음과 같다. 2절에서 웹 문서 군집화에 대해서 살펴보고, 3절에서는 웹 문서 군집화 시스템의 구조에 대해서 기술한다. 4절에서 웹 문서 군집화의 응용을 살펴보고, 5절에서 군집화 도구에 대해서 간단히 소개한다. 끝으로 6절에서 웹 문서 군집화에 관해서 토의하고 결론을 맺고자 한다.

2. 웹 문서 군집화란?

문서가 검색되었다는 것을 의미하고, 정확률이 80%라는 것은 10개의 검색된 문서 중에서 8개의 적합 문서가 검색되었다는 의미이다.

1) http://clusty.com/
2) http://kartoo.com/
3) http://iboogie.com/
4) http://mooter.com/
5) http://www.webclust.com/

군집화(clustering)는 여러 개체를 서로 비슷한 개체를 같은 덩어리를 형성하는 과정이며, 같은 덩어리에 속한 개체들 사이의 유사도를 최대화하고 다른 덩어리에 속한 개체들 사이의 유사도를 최소화하는 과정이다(Berkhin, 2002; Han and Kamber 2001; Jain *et al.* 1999). 군집화는 결과의 덩어리가 무엇이며, 덩어리의 총 수가 얼마인지는 미리 알 수 없다. 이 점이 분류(classification)와 큰 차이이다. 군집화 결과의 덩어리를 미리 알 수 없다는 점에서 비지도학습의 분류 방법(unsupervised classification)이라고 하기도 한다.

문서 군집화(document clustering)는 군집화 대상이 일반 문서이며, 의미적으로 비슷한 문서를 같은 덩어리에 할당하는 것이다. 웹 문서 군집화(Web document clustering)는 그 대상 문서가 웹 문서라는 것이다. 그렇다면 왜 문서 군집화 방법을 바로 웹 문서 군집화에 그대로 적용할 수 없을까? 그 이유는 몇 가지로 요약할 수 있다. 첫째, 대상 문서가 수억 개가 된다. 둘째, 문서가 지속적으로 변한다. 셋째, 문서의 형식이 매우 다양하고 대부분의 문서는 비구조적일 수 있다. 넷째, 군집화를 위해서는 링크와 같은 부가적인 정보가 더 필요하다.

웹 문서의 군집화가 응용되는 사례를 중심으로 웹 문서 군집화의 필요성을 설명하고자 한다. 이를 요약하면 다음과 같다. 첫째, 사용자들이 대량의 웹 문서를 효과적으로 훑어보고 분석할 수 있는 인터페이스를 제공한다. 두꺼운 책에서 원하는 정보를 빠르게 찾는 방법은 두 가지 방법으로 생각할 수 있다. 하나는 일반적으로 책의 뒤 부분에 나오는 색인을 이용하는 방법이고, 다른 하나는 책의 앞 부분에 나오는 목차를 이용하는 방법이다. 전자는 주로 용어의 정의나 단순한 내용을 찾을 경우이고 후자는 좀더 포괄적으로 내용을 알고자 할 경우에 사용된다. 전자는 일반적인 정보검색 기능이고, 후자는 문서 군집화 기능으로 볼 수 있다. 웹 문서에 대한 목차의 역할을 하는 것이 디렉토리 서비스이며, 대표적인 예가 야후 디렉토리이다. 이 디렉토리는 주제 영역별로 계층적으로 분류되어 있으면 대부분의 경우는 수작업으로 분류된다. 이것의 문제점은 새로운 문서가 들어올 때마다 주제를 할당해야 한다. 또한 경우에 따라서는 새로운 주제를 생성하여 할당하기도 한다. 군집화는 자동으로 계층구조를 생성하여 야후 디렉토리처럼 사용할 수 있다. 그러나 아직 실용화되기 위해서는 각 덩어리들에 적절한 이름 붙이기 등과 같은 연구(Kulkarni and Pedersen, 2005)가 지속적으로 이루어져야 할 것이다.

둘째, 검색 시스템의 재현율(recall)을 높이기 위해서 사용된다. 이를 위해서 문서들을 먼저 군집화해 두고, 사용자 질의에 의해서 덩어리 내의 한 문서가 검색되었을 경우 덩어리에 포함된 전체 문서를 검색된 것으로 간주한다. 이렇게 함으로써 사용자 질의에 "학교"가 포함되어 있을 때, "학원"이나 "유치원" 등이 포함된 문서도 자연스럽게 검색될 수 있다. 이렇게 검색되는 이유는 문서 군집화의 기본적인 가정이 비슷한 문서가 같은 질의에 관련될 가능성이 높기 때문이다(van Rijsbergen, 1979).

셋째, 검색 결과를 효과적으로 훑어볼 수 있도록 한다. 문서 군집화는 검색 결과의 문서를 주제별로 재분류할 수 있다. 이런 시스템의 예가 앞에서 간단히 소개한 *Vivísimo*와 *Clusty* 등과 같은 시스템이 있다. 이들 시스템은 단순한 군집화뿐 아니라, KartOO와 같이 HCI(human-computer interaction) 기술을 더하여 시각적인 효과를 더한 시스템들도 있다(<그림 5> 참조). 이와 같이 군집화를 수행함으로써 많은 경우 "은행"과 같은 단어처럼 동형이의어에 대한 처리가 자연스럽게 이루어질 수 있다. 또한 이와 같은 문서 군집화 시스템은 대부분 메타검색엔진(meta-search engine)[14]을 이용하고 있기 때문에 여러 검색엔진으로부터 검

14) 메타검색엔진은 스스로 검색 자료를 가지고 있지는 않지만, 다른 검색 사이트에 질의를 의뢰하고, 그 결과를 종합하여 사용자에게 알려주는 검색엔진이다.

색해야 하는 번거로움도 크게 완화할 수 있다(Selberg and Etzioni, 1997).

넷째, 많은 검색 시스템이 채용하는 벡터 모델의 검색 속도를 개선한다. 벡터 모델에서 주어진 질의와 모든 문서 사이의 유사도를 구하는 것은 많은 계산량을 요구한다. 이를 개선하기 위해서 사전에 문서를 군집화하여 모든 문서와 유사도를 계산하는 것이 아니라 군집화된 덩어리와 유사도를 계산함으로써 계산량을 크게 줄일 수 있다(Jardine and van Rijsbergen, 1971). 이 경우는 정확률은 다소 감소될 수 있으나 계산량은 크게 개선할 수 있을 것이다.

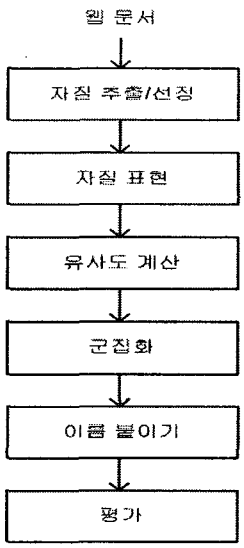
3. 웹 문서 군집화 시스템의 구조

웹 문서 군집화 시스템은 웹 문서 수집기(Heydon and Najork, 1999; Brin and Page, 1998)를 통해서 문서를 수집하는 것과 링크와 같은 웹 문서의 속성이 자질(feature)에 포함될 수 있다는 것을 제외하고는 일반적인 군집화 시스템과는 큰 차이가 없다. 웹 문서에서 추출되는 자질은 일반 문서 군집화에서 사용되는 자질 외에서 링크 수, 이미지 링크 수 등이 있으며(Sinka and Corne, 2004). 이는 목적에 따라 크게 다를 수 있다. 일반적으로 문서 군집화 방법은 몇 가지의 요구조건이 필요하다(Zamir *et al.* 1997). 첫째, 군집화의 결과로부터 한 덩어리의 내용을 쉽게 파악할 수 있어야 한다(ease-of-browsing). 그래서 자신의 원하는 내용이 해당 덩어리 내에 있는지를 빨리 판단할 수 있어야 한다. 둘째, 아주 짧는 시간 내에 군집화 결과를 보여줄 수 있어야 한다(speed). 인터넷 사용자들은 대개 질의의 결과를 수초 이상 기다리지 않는다. 셋째, 수천만 건의 문서에 대해서도 같은 방법으로 적용할 수 있어야 한다(scalability). 웹 문서가 하루가 다르게 증가하고 있기 때문에 대량의 문서에서도 빠른 시간 내에 군집화하지 못하면 실시간 지원은 사실상 어려운 일이다. 넷째, 실시간 처리를 바탕으로 하기 때문에 전처리 방법에 따라서 군집화 알고리즘의 성능이 달라서는 안된다(no preprocessing). 다섯째, 클라이언트에서 군집화 알고리즘이 실행되어야 한다(client side execution). 서버는 수많은 사용자 요구를 지원해야 하므로 가능하면 계산 부담을 줄여야 하고, 또한 클라이언트에서 군집화를 수행함으로써 개인적인 특성을 고려한 군집화가 가능하게 된다. 여섯째, 문서의 모든 내용을 보지 않고 일부의 내용만으로도 신뢰성 있는 군집화 결과를 생성할 수 있어야 한다(snippet-capable). 일반적으로 검색엔진에서 제공하는 문서의 일부분을 제공하는데 이 정보만으로 군집화를 수행할 수 있다.

<그림 2>는 일반적인 군집화 시스템의 구조이며, 총 여섯 단계로 구성된다(Jain *et al*, 1999; Han and Kamber, 2001). 이하의 절에서는 군집화의 각 단계에 대해서 개략적으로 기술한다.

3.1 자질 추출

문서를 군집화의 가장 이상적인 과정은 문서에 포함된 문장들을 이해하여 내용이 비슷한 문장끼리 같은 덩어리를 형성하는 것이다. 이 같은 방법



<그림 2> 군집화 시스템의 구조

은 현실적으로 불가능하다. 주어진 문서를 다양한 방법으로 추상화하여 재표현한다. 이 과정에서 원

래 문서가 가지고 있는 의미들이 어느 정도 소실될 수 있기 때문에 군집화의 결과가 항상 정확하다고는 말할 수 없다. 일반적으로 군집화에서 문서는 자질의 집합으로 가정한다. 그렇다면 도대체 무엇이 자질이 될 수 있을까? 자질은 크게 두 가지로 나누어 생각할 수 있다. 하나는 문서 내용에 관련된 자질이며, 이를 언어적 자질(linguistic feature)라고 한다. 언어적 자질에는 문서에 포함된 단어의 집합 혹은 아주 간단한 구문관계(syntactic relation)를 이용한다. 일반적으로 문서에 포함된 모든 단어를 자질로 선정하지는 않는다. 언어적 자질을 선정하는 방법은 일반적으로 정보검색에서 사용하는 색인어(index) 추출 방법(Baeza-Yates and Ribeiro-Neto, 1999)과 동일한데 빈도수가 중간 정도인 단어가 선정된다.

다른 하나는 문서의 부가정보에 관련된 자질이며, 이를 메타데이터(meta-data)라고 한다. 문서의 메타데이터로는 URL, 저자, 상호참조(cross reference), 각종 서지정보(bibliographic information) 등이 포함될 수 있다. 메타데이터는 수집된 문서에 따라 추출할 수 있는 자질들이 다소 제한될 수 있다.

3.2 자질 표현

문서 군집화에서 문서의 표현 방법은 정보검색에서 주로 사용되는 벡터 공간 모델(vector space model)을 이용한다(Baeza-Yates and Ribeiro-Neto, 1999). 각 문서에서 추출된 집합을 하나의 벡터로 표현하는 과정을 자질 표현이라고 한다. 벡터의 원소값은 다양한 방법(Salton and Buckley, 1988)으로 구할 수 있으나, 가장 널리 사용되는 모델은 BOW(Bag Of Words)로 각 단어가 하나의 독립 변수로서 적절한 가중치를 지니고 있다. 각 단어의 가중치를 결정하는 방법으로 가장 널리 사용되는 방법은 TFIDF 방법이다(Salton and Buckley, 1987).

3.3 유사도 계산

유사도 계산은 두 문서 간에 가까움의 정도를 계산하는 방법으로 두 가지 방법이 있다. 하나는 문서 간의 거리(distance)를 측정하는 방법이고, 다른 하나는 문서 간의 유사도(similarity)를 나타내는 방법이다. 본질적으로 유사도는 거리에 반비례함으로 사실상 같은 개념이다. 거리 측정법은 자질의 성질에 따라 다양하게 계산된다. 벡터 공간에서 거리 측정법을 다양한 분야에서 다양한 측정법이 제시되었으며 문서 군집화에서도 이들을 널리 사용하고 있다(Jain *et al.* 1999; Han and Kamber, 2001). 문서 군집화에서 주로 사용되는 방법은 내적(inner product), 코사인 계수(cosine coefficient), 다이스 계수(Dice coefficient), 자카드 계수(Jaccard corfficient) 등이 있다.

3.4 군집화 알고리즘

군집화는 크게 계층 군집화(hierarchical clustering)와 분할 군집화(partition clustering)로 나눈다. 계층 군집화는 큰 덩어리 안에 여러 개의 작은 덩어리들이 반복적으로 포함시키는 방법이며, 분할 군집화는 모든 문서를 k개의 분할로 나누며 각 분할을 하나의 덩어리로 간주하는 방법이다(Berkhin, 2002; han and Kamber, 2001; Steinbach *et al.* 2000). 계층 군집화에는 상향식 방법(agglomerative technique)과 하향식 방법(divisive technique)이 있으나, 하향식 방법은 문서 군집화에서는 거의 사용되지 않는다. 상향식 방법은 모든 문서를 하나의 덩어리로 가정하고 덩어리와 덩어리 사이의 유사도를 계산하여 가장 가까운 유사도를 가진 두 덩어리를 하나의 덩어리로 결합하는 방법이며, 이 과정을 원하는 종료조건이 되었을 때 군집화가 종료된다. 여기서 종료조건은 정해진 덩어리의 수가 되었을 경우나 유사도가 임계값 이하일 경우이다. 상향식 방법에는 단일 링크(single-link), 완전 링크(complete-link), 그룹 평균 링크(group average link) 방법이 있다.

분할 군집화에 가상 중심점 방법(centroid technique)과 실제 중심점 방법(mediod technique)로 나눈다. 가상 중심점 방법은 하나의 덩어리에 대표가 실제 문서가 아니고 임의의 새로운 벡터를 의미하고 실제 중심점 방법은 덩어리에 포함된 한 문서가 대표가 되는 방식이다. 일반적으로 가장 널리 사용되는 방법은 분할 군집화 알고리즘으로 K-mean 방법이 있으며 이 방법은 가상 중심점 방법이다.

3.5 이름 붙이기

군집화 결과를 사용자들이 보다 더 쉽게 사용할 수 있는 방법은 각 덩어리에 적절한 이름을 부여하는 방법이다(Kulkarni, 2005; Kulkarni and Pedersen, 2005). 이 문제는 아직도 해결되지 않는 문제들이 산재되어 있다. 최근에 덩어리에 이름을 붙이는 방법은 크게 세 가지로 생각해볼 수 있다. 첫째는 덩어리 내에서 빈도수가 가장 높은 몇 개의 단어를 추출하고 이를 의미 있는 순서로 재배열하여 덩어리의 이름(제목)으로 간주한다. 둘째, 덩어리 내에서는 빈도수가와 다른 덩어리에는 빈도수의 차이가 큰 단어를 선정한다. 셋째, 덩어리 내에 있는 문서를 요약하고 압축하여(내용 압축) 해당 덩어리의 제목으로 제공한다.

3.6 평가

군집화의 결과를 평가하는 것은 쉬운 일이 아니다. 왜냐하면 일반적으로 정답을 가지지 않기 때문이다. 그러나 군집화 시스템을 단순히 평가하기 위해서는 Reuter 말뭉치와 같은 문서 분류 말뭉치[15]를 사용하기도 한다. 문서 분류 말뭉치를 이용한 평가 방법은 정보검색 시스템에서 평가측도로 사용되는 정확률과 재현율을 주로 사용한다. 따라서 일반적인 군집화의 경우, 군집화 결과가 다른 시스템의 결과에 비해서 얼마나 좋은 것인지 혹은 나쁜 것인지를 말하는 것은 쉬운 일이 아니다. 군집화 평가 방법은 크게 5 가지 방법으로 생각할 수 있다.

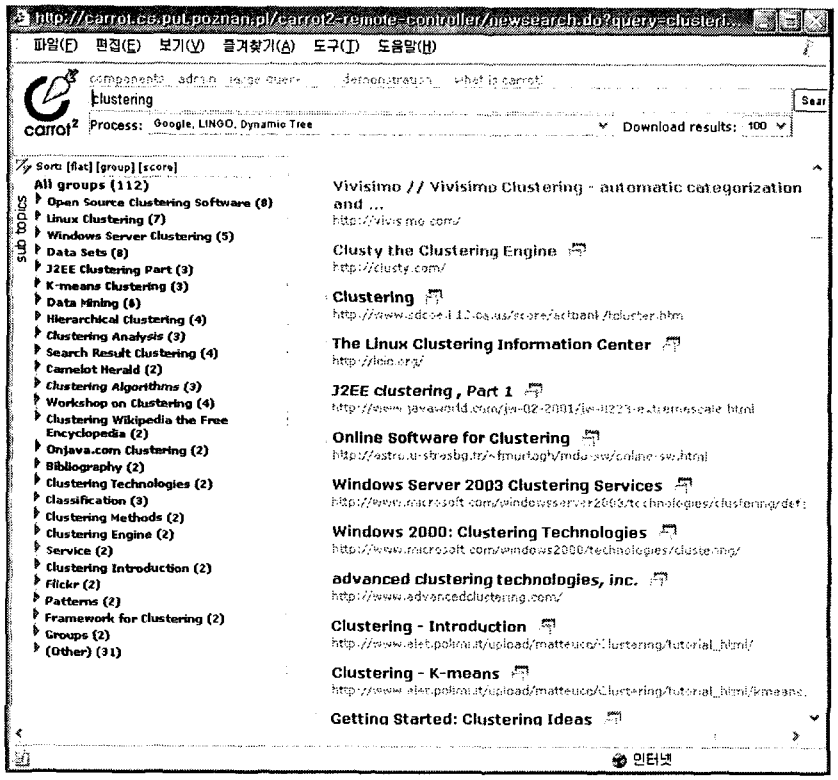
첫째 방법은 군집화 알고리즘 개발자가 직접 평가하는 것이다. 이 방법은 전혀 객관적인 방법이 아니므로 거의 사용되지 않는다. 둘째, 군집화 결과를 해당 분야의 전문가에게 의뢰하여 평가를 받는 것이다. 이 방법은 실질적으로 많은 시간이 소요되며 또한 많은 비용이 들어간다. 세 번째 방법은 야휴와 같은 디렉토리와 비교하는 방법이다. 웹 문서가 동적이기 때문에 많은 경우 디렉토리 내에 포함되어 있지 않아서 정확하게 평가할 수 없다. 네 번째 방법은 정보검색 시스템과 같은 다른 시스템을 이용하는 방법이다. 즉 군집화 결과를 정보검색 시스템에 반영하여 정보검색 시스템의 결과가 얼마나 좋아졌는지를 평가하는 방법이다. 이 방법은 정보검색 시스템에 다소 의존적이며, 모든 응용 분야에서 항상 같은 결과를 가져올지는 다소 의문이다. 다섯 번째 방법은 군집화의 성질을 이용한 군집화 지표(clustering index)를 사용하는 것이다. 이 지표는 다른 결과에 대해서 상대적으로 좋고 나쁨을 말할 수 있기 때문에 어느 정도 객관적인 측도가 될 수 있다(Bolshakova, 2005).

4. 웹 문서 군집화의 응용

본 절에서는 정보검색 결과의 군집화를 통해서 문서 군집화가 어떻게 응용되는지를 살펴보고자 한다. 먼저 초기 검색 결과 군집화 시스템으로서 Scatter/Gather(Cutting *et al.* 1992)와 Grouper (Zamir and Etzioni, 1999)를 간단하게 살펴보고, 공개된 문서 군집화 시스템으로 Carrot(Weiss, 2001)에 대해서 간단히 살펴본다. 그리고 나서 인터넷으로 통해서 직접 서비스되고 있는 Vivísimo와 Clusty 그리고 KartOO에 대해서 간단하게 살펴볼 것이다.

Scatter/Gather이 검색 결과를 군집화하여 보여주는 최초의 시스템이다. 이 시스템은 사용자 질의에 대한 검색 결과를 주제별로 군집화하고, 약간의

15) 문서 분류 말뭉치에는 Reuter-21578, OHSUMED, AP TREC, WebKB 등이 있다.



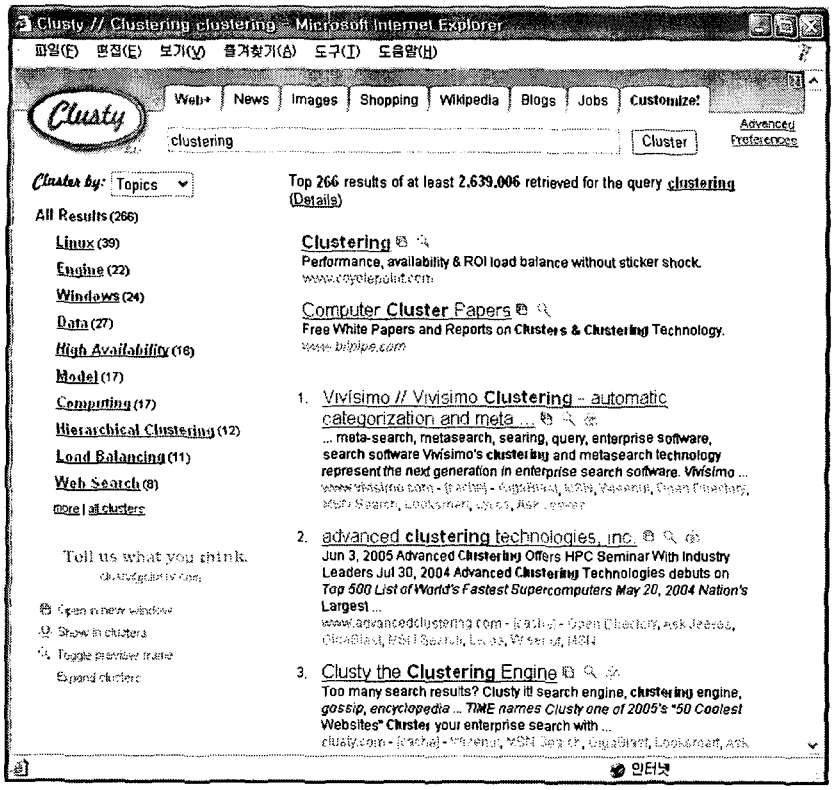
<그림 3> "Clustering"라는 질의에 대한 Carrot²의 화면 속사

요약 정보를 사용자에게 제공한다. 이 시스템은 웹 문서를 대상으로 한 시스템은 아니다. 웹 문서 검색 결과를 처음으로 군집화한 시스템은 Grouper가 있다. 이 시스템은 STC(Suffix Tree Clustering) 알고리즘을 사용해서 군집화한다. 이 알고리즘은 같은 구절(phrase)를 포함하는 문서를 하나의 덩어리로 결합하는 방법을 사용하고 있으며 실시간(선형 시간, linear time) 처리가 가능하다. Grouper에 영향은 받아서 STC를 구현하여 공개한 시스템이 Carrot이다. 현재는 Carrot²를 SourceFORGE.net를 통해서 공개하고 있으며 <그림 4>는 실험적으로 운영하고 있는 웹 사이트를 통해서 얻은 화면속사(screen snapshot)이다.

웹 문서 군집화 시스템의 대부분은 메타검색엔진(metasearch engine)을 통해서 검색된 결과에 대해서 군집화를 수행한다. 이런 시스템에는 1장에서 언급했듯이 Vivísimo, Clusty, KartOO, iBoogie, Mooter, WebClust 등이 있다. 본 논문에서 이들 중에 Vivísimo, Clusty, KartOO의 특징에 대해서 간단히 살펴보고자 한다.

Vivísimo은 상업용 시스템(commercial clustering engine)이다(Vivísimo, 2005). 이 시스템은 계층 군집화 알고리즘을 사용하나, 구체적으로 어떤 알고리즘을 사용하는지는 잘 알려지지 않았다. 이 시스템은 15개의 검색엔진에 질의를 보내고 그 결과를 받아서 중복된 결과는 제거하고 군집화한다. 군집화 결과의 각 덩어리에는 적절한 이름을 붙여서 사용자에게 전달한다(<그림 1>).

Clusty는 Vivísimo 시스템을 사용하여 다양한 사용자의 요구를 충족할 수 있도록 확장하였다. 일반 검색엔진과 같이 웹 문서뿐 아니라, 뉴스기사, 이미지, 백과사전, 블로그 등에서 검색하여 군집



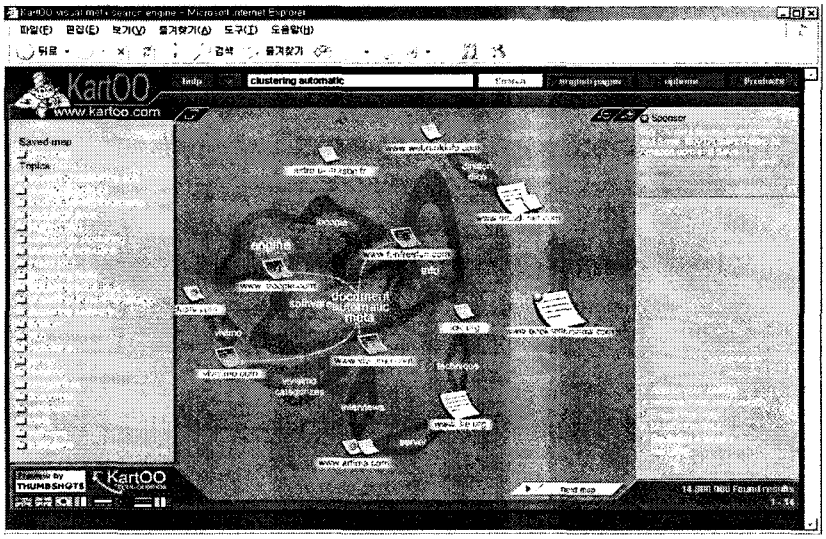
<그림 4> "Clustering"라는 질의에 대한 Clusty의 화면 속사

화할 수 있도록 한 시스템이다(<그림 4>). 특이 시스템은 사용자가 검색할 검색엔진 등을 선택할 수 있도록 하여 사용자의 요구에 최대한 부합될 수 있도록 하였다.

KartOO는 이제까지 소개한 군집화 시스템과는 자못 다른 인터페이스를 가진 군집화 시스템이다(<그림 5>). 이전에 소개한 시스템은 군집화 결과를 왼쪽에는 각 덩어리의 이름인 주제를 나열하고 오른쪽에는 그 덩어리에 속한 문서를 순위별로 보여준다. KartOO는 왼쪽은 다른 시스템과 같이 주제를 나열하고 오른쪽은 후원자(sponsor)와 프린터와 같은 보조기능을 바로 수행할 수 있도록 하였으며, 가운데에는 지도와 같은 GUI(Graphic User Interface)를 사용해서 사용자에게 더욱 친근하게 사용할 수 있도록 해준다. 지도의 등고선을 통하여 그 중요도를 표현하고 있으며 각 덩어리가 서로 어떻게 연결되었는지를 표현해서 서로의 연관도를 눈으로 쉽게 확인할 수 있도록 하였다.

5. 군집화 도구

공개된 군집화 도구들을 간단하게 소개하고자 한다. 웹 문서 군집화 시스템으로 Carrot^2(Weiss, 2001), 문서 군집화 시스템으로 CLUTO(Karypis, 2003)으로 간단하게 소개하고 단어 군집화(word clustering) 시스템으로 SenseCluster(Purandare and Pedersen, 2004)를 살펴본다. 또한 기계학습 도구의 일환으로 군집화 알고리즘을 내장하는 Weka(Witten and Frank, 2005)와 LNKnet (Kukolich and Lippmann, 2004)를 살펴볼 것이다.



<그림 5> "Clustering"라는 질의에 대한 KartOO의 화면 속사

Carrot[2][1]는 질의 기반으로 동적 군집화를 수행하고 그 결과를 시각화하는 시스템으로 웹 문서 군집화시스템이다. 기본적인 개념을 Vivísimo과 같으며 Java로 구현되었다. 시스템은 크게 입력부(input), 여과부(filter), 출력부(output), 제어부(controller)로 구성되었다. 입력부은 사용자 질의를 받아들이고, 여과부는 검색 결과에서 중복된 문서 등과 같은 불필요한 요소를 제거하고, 출력부은 군집화된 결과를 사용자에서 전달한다. 제어부는 입력부, 여과부, 출력부 사이의 통신을 제어하고 군집화한다. 이 시스템은 시스템을 컴포넌트 기반으로 설계되어 여러 응용 분야에 쉽게 사용할 수 있도록 하였다. 또한 Java로 구현하여 여러 플랫폼이나 운영체제에 무관하게 사용할 수 있다.

CLUTO[2]는 자질의 수가 많은 문서들에 대해서 효과적으로 군집화하는 도구이며 덩어리들의 다양한 특성을 분석할 수 있다(Karypis, 2003). 또한 정보검색, 데이터마이닝, 정보생물학 등 다양한 분야에 쉽게 적용할 수 있다. GUI 인터페이스를 가진 gCLUTO와 웹에 적용할 수 있는 인터페이스를 가진 wCLUTO도 함께 공개되어 있다. 다양한 군집화 알고리즘, 유사도 계산 기능을 가지고 있으며, 군집화 결과를 시각화하여 사용자들에게 제공한다.

SenseClusters[3]는 단어를 군집화하는데 초점을 맞추고 있으며 Perl로 작성된 군집화 알고리즘이다. 동형이의어가 서로 다른 의미일 경우 서로 다fms 문맥을 가진다고 하는 가정에 따라서 단어를 군집화한다. 이 도구의 기본적인 군집화 알고리즘은 CLUTO이다.

Weka[4]는 뉴질랜드 University of Waikato에서 개발된 데이터마이닝을 위한 기계학습 도구이다(Witten and Frank, 2005). 이 도구는 군집화 알고리즘뿐 아니라, 많은 종류의 분류 알고리즘(classification algorithm)도 함께 내장되어 있으며, Java로 구현되었다. LNKnet[5]는 MIT Lincoln Laboratory에 의해서 개발된 기계학습 도구이다

1) http://carrot.cs.put.poznan.pl/index.html
2) http://www-users.cs.umn.edu/~karypis/cluto/
3) http://www.d.umn.edu/~tpederse/senseclusters.html
4) http://www.cs.waikato.ac.nz/ml/weka
5) http://www.ll.mit.edu/IST/lnknet/index.html

(Kukolich and Lippmann, 2004). 이 도구는 최근에 개발된 많은 분류 알고리즘의 일환으로 군집화 알고리즘이 포함되어 있다.

6. 토의 및 결론

웹 문서 군집화는 웹 문서 검색 결과의 군집화 이외에도 다음과 같은 분야에 응용될 수 있다.

연관성 피드백(relevance feedback)(Iwayama, 2000): 일차적으로 검색 결과를 군집화하과 사용자 질의와 가장 연관성 있는 덩어리를 이용해서 연관성 피드백을 수행함으로써 사용자가 찾고자 하는 정보를 보다 더 빨리 찾을 수 있게 된다.

추천 시스템(recommender system)(Sarwar *et al.*, 2002): 인터넷을 통한 전자상거래에서는 수많은 고객이 수많은 상품들 중에서 자신이 원하는 상품을 찾아서 구매하기를 원하다. 여기서 군집화는 상품들을 먼저 군집화하여 몇 개의 덩어리로 나눈다. 그리고 나서 고객이 상품을 찾기 위해서 질의하여 상품을 찾을 때 그 상품이 속해 있는 덩어리를 함께 보여줌으로써 구매요구를 유발시킬 수 있을 뿐 아니라 사용자로 하여금 2차적인 질의를 줄일 수도 있다.

주제 검출과 추적(topic detection and tracking, TDT)(Allan *et al.*, 1998): 일반적으로 TDT는 어떤 사건을 찾고 추적하는 과정이다. 문서를 적절한 사건단위로 분리하고, 분리된 문서를 군집화하여 특정 사건이 어떤 과정으로 전개되는지를 찾을 수 있다.

지난 30년 동안 문서 군집화가 정보검색에 이모저모로 이용되어 왔다. 처음에는 정보검색의 효율성을 높이기 위해 사용되다가 90년대 초반에는 GUI 개선이나 검색 결과의 탐색을 도와주는 일에 치중하였으며, 최근에 와서는 동적 군집화에 초점을 맞추면서 실시간으로 질의에 따른 검색 결과를 군집화하는데 많은 연구 인력이 투자되고 있다. 30년 동안 꾸준히 연구해오고 있지만 여전히 풀리지 않은 많은 문제를 가지고 있다. 예를 들면, 자질 추출 및 축소(Dash and Liu, 2000; Sinka and Corne, 2004; Ruger and Gauch, 2000), 덩어리에 가장 적절한 이름 붙이기(Kulkarni and Pedersen, 2005; Zeng *et al.* 2004), 검색 결과의 시각화, 질의를 통한 실시간 군집화, 효율적인 알고리즘, 의미적인 자질 축소(semantically dimension reduction) 등이 있다. 군집화는 정보검색을 위한 매우 유용한 도구이며 이론적으로나 실용적으로나 매우 안정되어 있지만 아직도 해결되지 않는 문제들이 도처에 존재하고 있다.

참고 문헌

[1] Allan J., Carbonell, J., Doddington, G., Yamron, J. and Yang, Y. (1998). "Topic detection and tracking pilot study". *Proceedings of the Broadcast News Understanding and Transcription Workshop*, pp. 194-218.

[2] Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Addison Wesley.

[3] Berkhin, P. (2002). *Survey of clustering data mining techniques*, Technical report, Accrue Software, San Jose, California.

Bolshakova, N. (2005) https://www.cs.tcd.ie/Nadia.Bolshakova/validation_algorithms.html

[4] Brin, S. and Page, L. (1998). "The anatomy of a large-scale hypertextual web search engine". *Proceedings of the Seventh International World Wide Web Conference*, pp. 107-117.

[5] Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. (1992) "Scatter/gather: a cluster-based approach to browsing large document collections". *Proceedings of the 15th Annual International ACM SIGIR conference on*

Research and development in information retrieval, pp. 318-329.

[6] Dash, M. and Liu, H. 2000. "Feature Selection for Clustering". *Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Current Issues and New Applications*. T. Terano, H. Liu, and A. L. Chen, Eds. *Lecture Notes In Computer Science*, vol. 1805. Springer-Verlag, London, 110-121

[7] Dhillon, I. S., Kogan, J. and Nicholas, M. (2002). "Feature Selection and Document Clustering", Book chapter in *Text Data Mining and Applications*, 2002.

[8] Han, J. and Kamber M. (2001). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

[9] He, X., Zha, H., Ding, C., and Simon, H. (2001). *Web document clustering using hyperlink structures*, Technical Report. CSE-01-006, Department of Computer Science and Engineering, Pennsylvania State University.

[10] Heydon, A. and Najork, M. (1999). "Mercator: A scalable, extensible Web crawler". *World Wide Web*, vol. 2 no. 4 , pp. 219-229.

[11] Iwayama, M. (2000). "Relevance feedback with a small number of relevance judgements: incremental relevance feedback vs. document clustering", *Proceedings of ACM SIGIR '00*, pp. 10-16.

[12] Jain, A.K., Murty M.N., and Flynn P.J. (1999). "Data Clustering: A Review", *ACM Computing Surveys*, vol 31, no. 3, pp. 264-323.

[13] Jardine, N. and van Rijsbergen, C. J. (1971). "The use of hierarchical clustering in information retrieval". *Information Storage and Retrieval*, vol. 7, pp. 217-240.

[14] Karypis. G (2002). *CLUTO: A Clustering Toolkit,*. Technical Report no. 02-017, Department of Computer Science, University of Minnesota, Minneapolis, USA.

[15] Kukolich, L. and Lippmann, R. (2004). *LNKnet User's Guide,*. MIT Lincoln Laboratory.

[16] Kulkarni, A. and Pedersen, T. (2005). "SenseClusters: Unsupervised clustering and labeling of similar contexts", *Proceedings of the ACL Interactive Poster and Demonstration Sessions*.

[17] Kulkarni, A. K (2005). "Unsupervised Discrimination and labeling of ambiguous names", *Proceedings of the ACL Student Research Workshop*, pp.145-150.

[18] Lan , M., Tan, C.-L., Low, H.-B. and Sung S.-Y. (2005). "A comprehensive comparative study on term weighting schemes for text categorization with support vector machines". *Proceedings of WWW '05: Special interest tracks and posters of the 14th international conference on World Wide Web*, pp. 1032-1033.

[19] Lang, H. C. (2003). *A Tolerance Rough Set Approach to Clustering Web Search Results*. Master Thesis, Faculty of Mathematics, Informatics and Mechanics, Warsaw University.

[20] Mureasan, G. (2002). *Using Document Clustering and Language Modelling in Mediated Information Retrieval*. School of Computing, The Robert Gordon University, Aberdeen, Scotland.

[21] Purandare, A. and Pedersen T. (2004). "SenseClusters - Finding clusters that represent word senses". *Proceedings of Fifth*

Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-04).

[22] Ruger, S. and Gauch, S. (2000). "Feature reduction for document clustering and classification". Technical Report, Computing Department, Imperial College London, UK.

[23] Salton, G. and Buckley, C. (1987). "Term weighting approaches in automatic text retrieval, Technical Report COR-87-881, Department of Computer Science, Cornell University, USA.

[24] Salton, G., & Buckley, C. (1988). "Term weighting approaches in automatic text retrieval". *Information Processing and Management,* vol. 24, no. 5, pp. 513-523.

[25] Sarwar, B. M., Karypis, G., Konstan, J. and Riedl, J. (2002). "Recommender systems for large-scale e-commerce: scalable neighborhood formation using clustering". *Proceedings of the 5th International Conference on Computer and Information Technology.*

[26] Selberg, E. and Etzioni, O. (1997). "The MetaCrawler architecture for resource aggregation on the web". *IEEE Expert,* vol. 12, no. 1, pp. 8-14.

[27] SIGIR (2005). The 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, http://www.sigir2005.org/

[28] Sinka, M. P. and Corne, D. W. (2004). "Evolving document features for Web document clustering: A feasibility study", *Proceedings of the IEEE Congress of Evolutionary Computation.*

[29] Steinbach, M., Karypis, G. and Kumar, V. (2000), *A Comparison of Document Clustering Techniques,* Technical Report no. 00-034, Department of Computer Science and Engineering, University of Minnesota.

[30] TREC (2005). The 14th Text REtrieval Conference, http://trec.nist.gov/

[31] van Rijsbergen, C. J. (1979). *Information Retrieval.* Butterworths, London, second edition.

[32] Weiss, D. (2001). *A Clustering Interface for Web Search Results in Polish and English.* Master's thesis, Poznan University of Technology, Poland.

[33] Weiss, D. (2002). "Introduction to search results clustering". *Proceedings of the 6th International Conference on Soft Computing and Distributed Processing,* Rzeszów, Poland.

[34] Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques,* 2nd Edition, Morgan Kaufmann, San Francisco.

[35] Zamir, O. and Etzioni, O. (1998). "Web document clustering: A feasibility demonstration". *Proceedings of the 21st annual international conference on research and development in information retrieval (SIGIR'98),,* pp. 46-54.

[36] Zamir, O., and Etzioni, O. (1999). "Grouper: A dynamic clustering interface to web search results". *Computer Networks (Amsterdam, Netherlands: 1999)* 31, pp. 1361-1374.

[37] Zamir, O., Etzioni, O., Madani, O., and Karp, R. (1997). "Fast and intutive clustering of web documents. *Proceeding of the 3rd International Conference on Knowledge Discovery and Data Mining,* pp. 287-290.

[38] Zeng, H.-J., He, Q.-C., Chen, Z. and Ma, W.-Y. (2004). "Learning to cluster web search results". *Proceedings of the 27th annual*

international conference on research and development in information retrieval (SIGIR'04), pp. 210-217.

[39] 김병희 (2003). 정보이론 기반의 병합식 클러스터링 기법 연구, 공학사 학위논문, 서울대학교 컴퓨터공학부.

[40] 이경순 (2001). 정보검색에서 벡터공간 검색과 클러스터 분석을 통한 문서 순위 결정 모델, 한국과학기술원, 전산학과, 박사학위 논문.

김 재 훈

1986년 계명대학교 전자계산학과(학사)
1988년 한국과학기술원 전산학과(공학석사)
1996년 한국과학기술원 전산학과(공학박사)
1988년-1997년 한국전자통신연구원, 선임연구원
1997년-현재 한국해양대학교 컴퓨터공학과 부교수
2000년 - 2002년 2월 한국과학기술원 첨단정보기술연구센터, 연구원
2001년 - 2002년 2월 USC, Information Sciences Institute, 방문연구원
관심분야: 자연언어처리, 한국어 정보처리, 정보검색, 정보추출, 지식공학
E-mail: jhoon@mail.hhu.ac.kr

박 은 진

2003 한국해양대학교 자동화 정보 공학부(학사)
2002 - 2004 (주) 블루코드테크놀로지 재직
2004 - 현재 한국해양대학교 컴퓨터 공학과 대학원 자연언어처리 전공 재학
E-mail: bakeunjin@hhu.ac.kr

옥 철 영

1982 서울대학교 공과대학 컴퓨터공학과(학사)
1984 서울대학교 대학원 컴퓨터공학과(공학석사)
1993 서울대학교 대학원 컴퓨터공학과(공학박사)
1984- 현재 울산대학교 컴퓨터정보통신공학부 교수
관심분야: 자연어처리, 기계학습 및 지식처리, 정보검색, 웹 기반 정보시스템
E-mail: okcy@ulsan.ac.kr