www.calsec.or.kr
http://dx.doi.org/10.7838/jsebs.2013.18.2.095

워드넷 기반 특징 추상화를 통한 웹문서 자동분류시스템의 성능향상

Improving Hypertext Classification Systems through WordNet-based Feature Abstraction

노준호(Jun-Ho Roh)*, 김한준(Han-joon Kim)**, 장재영(Jae-Young Chang)***

초 록

본 논문은 기계학습 기법에 기반한 웹문서 자동분류 시스템의 성능을 높이기 위한 새로운 형태의 특징가공 기법을 제안한다. 제안 기법은 하이퍼텍스트 웹문서에 대한 자동분류를 효과적으로 수행하기 위해 하이퍼링크 관계를 활용하여 특징 집합을 확장시킨다. 웹문서는 하이퍼링크 관계를 통해 서로 연결된 구조를 가지며, 그 관계는 많은 경우 연관도가 높은 문서들 간에 존재한다. 이러한 링크 정보가 분류모델의 주요 인자가 되는 특징 집합의 질을 높이는 중요한 역할을 수행할 수 있다. 제안 기법의 기본 아이디어는 워드넷 온톨로지를 기반으로 분류 대상 문서와 인접 문서들에 포함된 단어(특징)들 간의 의미적 유사도를 평가함으로써 다수의 특징들로 구성된 추상화된 개념적 특징을 생성하는 것이다. 여기서 유사도 함수는 워드넷 안에서 특징들 간의 상/하위어 관계 정보를 정량적으로 계산하게 된다. 분류모델의 구축시 추상화된 개념 특징은 일반 특징과 동일하게 간주하여 보다 정확한 분류 모델을 구축하는데 기여한다. Web-KB 문서집합을 이용한 실험을 통해 제안 기법이 기존 기법 보다 우수함을 보였다.

ABSTRACT

This paper presents a novel feature engineering technique that can improve the conventional machine learning-based text classification systems. The proposed method extends the initial set of features by using hyperlink relationships in order to effectively categorize hypertext web documents. Web documents are connected to each other through hyperlinks, and in many cases hyperlinks exist among highly related documents. Such hyperlink relationships can be used to enhance the quality of features which consist of classification models. The basic idea of the proposed method is to generate a sort of abstracted concept feature which consists of a few raw feature words; for this, the method computes the semantic similarity between a target document and its neighbor documents by utilizing hierarchical relationships

본 연구는 2010년도 한국연구재단의 기초연구사업(과제번호 : NRF-2010-0025212) 지원으로 이루어졌으며, 또한 2011년도 한국연구재단의 기초연구사업(과제번호 : NRF-2011-0022445) 지원으로 이루어졌음.

* First Author, School of Electrical and Computer Engineering, University of Seoul
** Corresponding Author, School of Electrical and Computer Engineering, University of Seoul (E-mail : khj@uos.ac.kr)
*** Co-Author, Department of Computer Engineering, Hansung University

2013년 03월 18일 접수, 2013년 04월 09일 심사완료 후 2013년 04월 22일 게재확정.

in the WordNet ontology. In developing classification models, the abstracted concept features are equated with other raw features, and they can play a great role in developing more accurate classification models. Through the extensive experiments with the Web-KB test collection, we prove that the proposed methods outperform the conventional ones.

키워드 : 자동문서분류, 기계학습, 워드넷, 하이퍼텍스트, 월드와이드웹, 속성추상화
Text Classification, Machine Learning, WordNet, Hypertext, World Wide Web, Feature Abstraction

1. 서 론

하이퍼텍스트 웹문서는 반구조적 문서(semi-structured document)로서 일반 문서와는 달리 태그(tag) 정보와 하이퍼링크(hyperlink) 정보를 담고 있다. 여기서 하이퍼링크는 웹문서 간의 관계를 나타내는 중요한 정보이며 하이퍼링크로 연결된 문서들은 서로 유사한 내용을 공유하고 있거나 연관되는 정보를 가지는 것이 보통이다. 이러한 이유로 웹문서를 위한 자동분류시스템의 성능 향상을 위해 하이퍼텍스트의 구조적 특성을 활용하는 연구가 꾸준히 진행되어 왔다[1, 10, 15]. 자동분류(automated classification)는 주어진 문서에 대하여 기 정의된 클래스(class)에 자동으로 분류하는 기술을 말한다.

최근 대부분의 자동문서분류 기법은 기계학습(machine learning) 알고리즘을 활용하며, 이 중에서 문서 데이터에 주로 적용되는 알고리즘은 나이브베이즈(naïve Bayes), 서포트벡터머신(support vector machine), k-최근접이웃(k-nearest neighbors) 등이다[9]. 이러한 알고리즘들은 다른 기계학습 알고리즘에 비해 문서 데이터가 가지는 차원의 저주(curse of dimensionality) 문제를 어느 정도 극복할 수 있으며, 실제 문서의 표현을 위해서 각 차원을 단어로 표현하는 Bag-of-Words (BoW) 방식을 흔히 사용한다. BoW 방식은 문서에 포함된 단어 및 그것의 출현 횟수 정보만을 가지고 문서를 표현한다. 그래서 자동분류 알고리즘을 포함한 대부분의 텍스트마이닝(text mining) 알고리즘은 BoW 방식에 따라 문서를 가공・처리하며 그것의 단순한 표현방식에도 불구하고 합리적 성능을 얻는다고 평가받고 있다[16]. 하지만 BoW 방식은 단어가 가지는 의미를 표현하지 못하는 단점을 안고 있어서, 자동분류 알고리즘의 성능을 높이기 어렵게 만드는 근본적 원인이 된다. 자동분류 알고리즘의 성능 향상을 위해 흔히 사용하는 특징 선택(feature selection) 기법이 BoW 방식으로 생성된 특징 단어 집합을 정제하는 역할을 수행하지만 그것이 BoW 방식의 단점을 극복하지는 못한다.

최근 이러한 문제를 극복하기 위한 방안으로서 특징 가공이 효과적인 대안으로 연구되고 있다[3, 6, 8, 11, 14]. 특징 가공(feature engineering)이라 함은 주어진 분류 모델의 인자인 특징(feature)으로서의 문서 내부의 단어를 변형하거나 이와 관련된 단어를 인위적으로 추가하는 기법을 의미한다. 웹문서의 자동분류에 있어서 특징 가공은 두 가지 방법이 가능하다. 하나는 온톨로지(ontology)를 이용하여 의미적으로 유사한 외부 특징들을 추출, 활용하는 방안이다. 예들 들어, Mansuy and Hilderman[8],

Scott and Matwin[14]에서는 워드넷(WordNet) 온톨로지(http://wordnet.princeton.edu)를 이용하여 각 특징마다 동의어, 상위어 등을 추가하였으며, Elberrichi et al.[3], Lu et al.[6]에서는 특징 집합 내에서 동의어 관계를 맺는 특징들을 묶어내어 이들의 통계량을 계산하는 방법을 제안하였다.

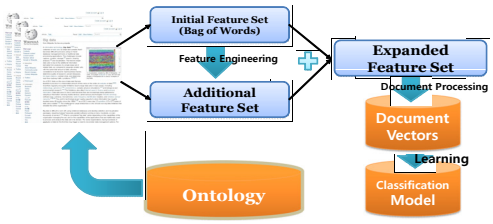
웹문서의 특징 가공을 위한 두 번째 방안은 인접문서의 관계 정보를 이용하는 것이다. 즉 주어진 웹문서의 분류를 위해서 그것의 인접문서에 속한 특징 정보를 분류 대상 문서의 특징 집합에 포함시키는 것이다. 인접 문서들 중 대상 문서와 연관도가 높은 인접 문서를 선별하여 그 문서와 분류 대상 문서에 모두 포함된 특징을 추출할 수 있고[10], 특징 집합의 규모를 줄이기 위해 인접 문서 내의 특징들 중에서 중요한 특징만을 선별할 수도 있다[1, 15]. 구체적으로, Oh and Myaeng [10]에서는 문서 간 유사도를 기반으로 신뢰도를 산정하여 인접 문서를 선별하고, 선별된 인접 문서와 대상 문서내의 공통된 단어들에 대해 가중치를 부여하는 방법을 제안하였다. Chakrabarti et al.[1]에서는 인접 문서들의 클래스 레이블(class label)만을 특징으로 사용하여 분류 정확도를 향상시켰고, Utard and Fürnkranz[15]에서는 인접 문서내의 앵커 텍스트(anchor text)와 앵커 텍스트 주변 단어를 특징으로 추출하는 방식을 제안하였다. 그러나 이러한 기존 기법들의 문제점은 분류 대상문서와 연관성이 낮은 특징 단어들까지 포함시킨다는 것이다.

본 논문은 웹문서의 자동분류 성능을 높이기 위해 워드넷 기반 의미적 유사도(semantic similarity)에 기반한 새로운 특징 가공 기법을 제안한다. 기본 아이디어는 추출한 특징 집합에서 의미적 연관도가 높은 특징들을 묶어 상위 개념으로 추상화한 새로운 개념 특징을 생성하는 것이다. 분류 대상 문서에 대하여 본래 보유한 특징 단어, 인접 문서의 단어, 그리고 추상화된 개념 특징 등 의미적으로 풍부한 특징 집합으로 표현함으로써 보다 정확한 자동분류 모델을 구성하자는 것이다.

본 논문의 구성은 다음과 같다. 제 2장에서는 본 논문이 제안하는 특징 가공 과정 및 특징 추상화 알고리즘을 설명하며, 제 3장에서는 이를 검증하기 위한 실험 및 성능 평가 결과에 대해 기술한다. 마지막으로 제 4장에서 결론을 맺는다.

2. 워드넷 기반 하이퍼텍스트 문서의 특징 가공

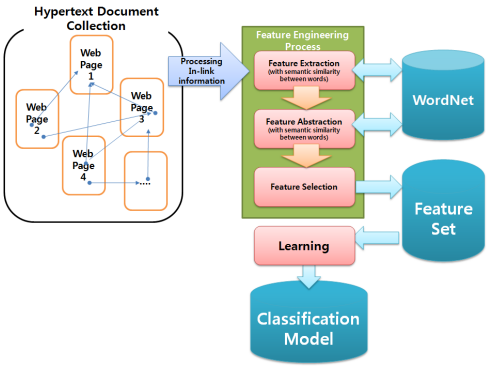
2.1 하이퍼텍스트 웹문서의 특징 가공



〈Figure 1〉 Feature Engineering Process for Automated Text Classification

기계학습 기반의 자동문서분류 시스템은 크게 '학습(learning)' 모듈과 '분류(classification)' 모듈로 구성된다. 학습 모듈은 학습 단계를 수행하는 것으로서, 클래스(class)가 미리 부여된 학습 문서집합(training document set)으로부

터 선별·추출한 양질의 주요 특징 단어들을 이용하여 기반 학습 알고리즘에 의해 분류모델(classification model)을 구축한다. 분류 모듈은 기구축된 분류모델을 활용하여 새로운 유입 문서들의 클래스를 예측하는 작업을 수행한다. 문서 데이터에 대한 분류모델은 많은 수의 특징을 포함하기 때문에 그 정확도가 기계학습 알고리즘 자체의 성능보다는 특징 집합의 질에 크게 의존한다. 따라서 자동문서분류 시스템의 성능을 높이기 위해서는 학습 알고리즘의 개선 측면 보다는 분류모델의 기본 요소가 되는 특징 단어의 질을 높이는 측면이 보다 효과적일 가능성이 크다[3, 6, 8, 14]. 이런 맥락에서 최근 주어진 문서의 문맥적 의미와 관련 깊은 요소를 추출하는 특징 가공 기법에 대한 연구가 중요해지고 있다. <Figure 1>은 본 연구가 고려한 특징 가공 기법을 도식화한 것이다. 특징가공은 학습 문서로부터 1차 추출된 특징 집합과 온톨로지를 활용하여 새롭게 얻어진 특징 집합을 병합하여 이를 분류 모델의 구성 인자로 사용하는 것이다.



〈Figure 2〉 Text Classification System Architecture with Feature Engineering

<Figure 2>는 본 연구에서 제안하는 웹문서의 자동문서분류 시스템의 구조도를 보여준다. 그림에서 보는 바와 같이, 특징 가공 과정은 '특징 추출(feature extraction)', '특징 추상화(feature abstraction)', '특징 선택(feature selection)' 세 단계로 이루어진다. 우선 '특징 추출' 단계에서는 웹문서의 진입링크 관계 정보와 워드넷 온톨로지의 단어 간 유사도를 계산함으로써 인접문서로부터 새로운 특징들이 추출된다. 그 후 '특징 추상화' 단계에서는 앞서 추출된 특징 집합으로부터 의미적 유사성이 높은 특징들을 묶어 개념 수준의 특징을 생성한다. 이 때 사용하는 유사도 함수는 이전 단계에서 사용한 것과 동일하다. 마지막으로 '특징 선택' 단계에서는 단어 및 개념 수준의 특징 전체집합으로부터 특정 기준에 따라 주요 특징을 선별한다. 본 연구에서는 특징 선택을 위해 카이제곱 통계량(χ^2-statistics)을 채택한다. 결국 본 논문의 특징가공 기법은 서론에서 기술한 특징 가공의 두 가지 방법을 모두 포함한 것이다.

<Figure 2>의 '특징 추출' 단계와 관련하여, 웹문서에 대한 특징 가공을 위해 하이퍼링크 관계 정보를 적극 활용한다. 하이퍼링크는 대상문서로 들어오는 진입링크(in-coming link)와 대상문서에서 나가는 진출링크(out-going link)로 나뉜다. 하이퍼링크 관계를 이용한 특징 추출의 대표적인 연구인 Utard and Fürnkranz [15]에서는 진입링크 문서의 앵커 텍스트(anchor text)[1]와 앵커 텍스트와 인접한 20~30개의 주변단어를 대상 문서의 특징으로 사용하여 문서 분류의 정확도를 향상시키려 하였다. 그러나

1) 앵커 텍스트는 HyperText Markup Language (HTML)에서 하이퍼링크를 생성하는 <a> 태그에 포함된 텍스트를 의미한다.

이는 주변단어를 설정하는 휴리스틱이 임의적이어서 모든 문서 도메인에 일반화하여 효과를 보기 어렵다. 본 논문에서는 의미 어휘사전인 워드넷 온톨로지를 활용하여 진입링크 문서의 특징 집합에서 대상문서와 연관이 높은 특징만을 추출한다. 제 2.2절에서 이에 대하여 자세히 기술한다.

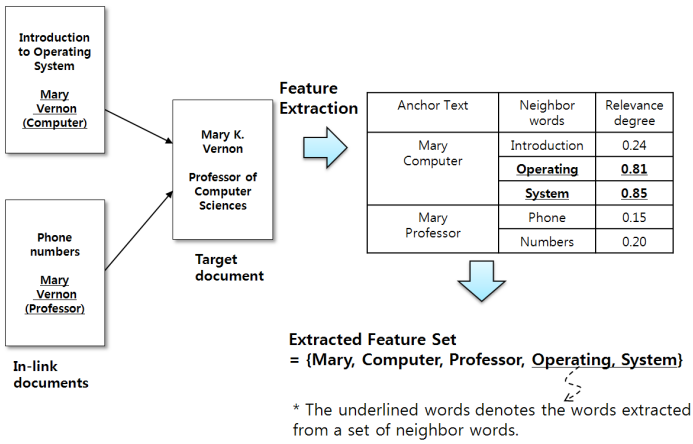
<Figure 2>의 '특징 추상화' 단계와 관련하여, 문서 자체에 포함된 특징 집합과 특징 추출 단계에서 추출된 특징 집합에 대하여 선별적으로 중요한 특징들의 가중치를 인위적으로 높이기 위해 의미적으로 연관있는 특징들을 묶어 이를 하나의 개념 수준의 특징으로 승화시킨다. 기본적으로 자동문서분류 시스템의 성능을 높이려면 학습문서가 그것의 소속 클래스의 주제와 상응하는 특징들로 표현되는 것이 바람직하다. 그리고 그러한 특징들은 서로 하나의 개념을 공유할 가능성이 높다. 예를 들어, 집합{'bus', 'car', 'auto'}에 속한 단어들은 '운송 수단'이라는 개념과 관련된, 즉 해당 개념을 정의할 수 있는 속성이라 할 수 있다. '특징 추상화' 작업은 하나의 개념으로 묶을 수 있는 단어들을 개별적인 특징으로 보지 않고, 그 단어 집합을 하나의 추상화된 개념 특징으로 간주하는 것이다. 특징 추상화를 통해 얻어진 개념 특징은 다른 단어 특징들보다 가중값을 높임으로써 분류 모델에서 클래스간의 분별도를 높일 수 있다.

2.2 새로운 특징의 추출

2.2.1 진입링크 문서로부터 특징 추출

하이퍼링크 관계에 있는 진입링크 문서에서 추출된 특징은 대상 문서와 연관성이 커야 한다. 그래서 보통 특징 추출 영역을 진입문서의 전체 영역으로 하기 보다는 클래스 이름, 앵커 텍스트, 앵커 텍스트 주변단어 등의 영역으로 한정한다[15]. 하지만 클래스 이름 또는 앵커 텍스트의 경우 한 문서의 내용을 간단한 단어로 축약하거나 대표 단어만을 사용한 것이기 때문에 특징 개수가 많지 않다. 반면에 앵커 텍스트 주변단어 영역으로 확장하는 경우에 대상 문서와 무관한 특징들이 다수 포함될 가능성이 크다. 본 논문에서는 양질의 특징 단어를 적정 개수 추출하기 위해, 추출 영역을 앵커 텍스트 주변의 문단(paragraph) 영역으로 하되 앵커 텍스트와 연관이 높은 용어만을 추출하는 방법을 제안한다.

<Figure 3>은 제안 기법을 통한 특징 추출의 예를 보여준다. 먼저 진입링크 문서에서 앵커 텍스트와 앵커 텍스트 주변단어를 추출하고, 앵커 텍스트에 속한 모든 단어와 주변단어 간 의미적 연관도를 계산한다. 그 후 연관도가 일정 임계값 이상인 특징들만을 추출한다. 여기서 임계값은 의미 있는 유사도를 가지는 특징의 추출 여부를 결정하며 0에서 1까지의 값을 가진다. 이 값이 0이면 연관 단어를 추출하지 않음을 의미하고, 1이면 완벽한 동의어만을 특징으로 추출함을 의미한다. 그림에서 보는 바와 같이, 앵커 텍스트 내의 단어인 'Mary', 'Computer', 'Professor' 그리고 앵커 텍스트와 의미적 연관도가 높은 단어인 'Operating', 'System'이 특징으로 추출되었다. 이렇게 추출된 특징들은 분류 대상 문서를 표현하는 주요 속성이 되며, 결국 해당 클래스에 대한 분류모델의 분별력을 높여주는 역할을 하게 된다. 이러한 특징 단어를

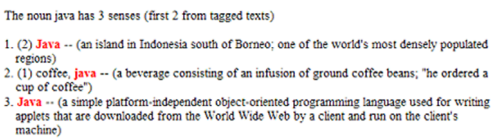


〈Figure 3〉 An Example of Features Extracted from in-link Documents

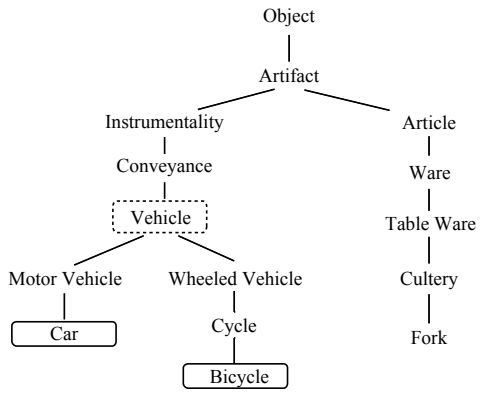
정확하게 추출하기 위해서는 의미적 유사도를 측정할 수 있는 함수가 필요한데, 본 연구에서는 기존 워드넷 기반 유사도 함수를 이용할 것이다.

2.2.1 워드넷 기반 단어 간 유사도

워드넷은 영어 의미 어휘목록 관계정보를 담은 사전으로서 소위 '동의어 집합(synset)' 단위체의 네트워크 구조를 이룬다. 이는 각 동의어 집합에 대한 상위어(hypernym), 하위어(hyponym), 등위어(coordinate term), 전체어(holonym) 등의 의미 관계들을 제공한다. <Figure 4>는 세 가지 의미를 가지는 단어 'java'를 워드넷을 통해 검색한 결과이다. 'java'는 'island', 'coffee', 'programming language'라는 3가지 의미를 가지는 중의적 단어이다. 워드넷은 각 의미에 대한 동의어가 synset으로 구성되어 각 의미와 관련된 연관 단어를 연결시켜 네트워크 구조를 구성한다. 워드넷의 연관 관계 정보 중 단어 간 연관도를 계산하기 위해 상·하위어 위상관계를 이용할 수 있다.



〈Figure 4〉 An Example of 'java' Synset in WordNet



〈Figure 5〉 An Example of hierarchical Relationship among Words in WordNet

<Figure 5>는 단어 'Car'와 'Bicycle' 간의 상·하위어 의미 관계도를 보여준다. 여기서

단어 'Vehicle'은 두 단어의 공통된 상위어이고 이 노드를 통해 두 단어가 연결되어 있다. 기본적으로 워드넷에서 단어간 의미적 거리는 서로 연결된 노드를 지나는 최소 거리로 정의할 수 있으며[4], 본 연구에서는 RiTa.WordNet[13]에서 제안한 식 (1)의 단어 a, f 간의 연관도 $sim(a, f)$를 사용한다.

$$sim(a,f) = 1 - \frac{mindist(a,f)}{mindist(commonparent, root) + mindist(a,f)} \quad (1)$$

여기서 a는 앵커 텍스트를 나타내고, f는 앵커 텍스트의 주변단어를 나타낸다. 그리고 mindist(s1, s2)는 워드넷 상·하위어 관계에서 단어 s1, s2 간의 최소 거리를 의미한다. 또한 commonparent는 두 단어의 공통의 상위어를 의미하고, root는 가장 최상위 개념의 단어를 의미한다. 이 식은 정규화된 식으로서, 두 단어가 동의어이면 최소 거리가 0이 되어 연관도는 1의 값을 가지며, 반대로 공통의 상위어가 없으면 최소 거리가 무한대의 값을 가지므로 0의 값을 가진다.

2.3 특징 추상화

서론에서 기술한 바와 같이, 텍스트마이닝에서 흔히 사용하는 Bag-of-Words 문서 표현방식은 단어가 가지는 의미를 표현하지 못하여 자동분류 알고리즘의 성능을 높이는데 한계를 준다. 즉, 문서 내 중요한 개념을 갖는 특징 단어라 할지라도 그 빈도수가 낮을 경우 학습 결과로서 만들어진 분류모델에서 가중치가 낮게 산정되어 그 특징의 영향력이 작을 수밖에 없다. 예를 들어, 컴퓨터 관련 문서에서 'c', 'java', 'php'같은 단어들은 컴퓨터 언어라는 공통의 상위 개념을 가진다. 만약 이러한 단어들이 한 문서에 존재한다면 빈도수가 낮더라도 의미적인 중요도는 높다고 볼 수 있다. 개념적으로 의미가 분명하거나 의미적으로 연관된 단어의 가중치를 높여준다면 분류모델의 성능을 높이는데 크게 기여할 것이라 보는 것이다. 이를 위해 본 논문에서는 워드넷을 사용해 추출된 특징으로부터 서로 연관도가 높은 단어들을 하나의 상위 개념으로 추상화(abstraction)하는 기법을 제안한다.

본 논문에서는 '개념(concept)'을 Formal Concept Analysis(FCA) 이론에 근거하여 개략적으로 정의한다. FCA 이론[12]에 따르면, 하나의 '개념(concept)'은 '외연(extent)'과 '내연(intent)'로 구성되는데, '외연'은 해당 개념에 포함되는 개체(instance)들의 집합이고, '내연'은 외연에 포함된 모든 개체들의 공통된 속성(attribute)들의 집합이다. 이 개념에 대한 정의를 문서 데이터에 적용하면, 외연은 특정 클래스에 속한 '개념'과 연관된 학습문서들의 집합이고, '내연'은 해당 학습문서들의 공통된 특징들의 집합이라 할 수 있다. 그래서 학습문서 데이터에서 의미적으로 유사한 주요 특징 단어들을 묶어낼 수 있다면, 그 '내연'적 특징 단어들을 가지고 하나의 개념을 정의할 수 있다. 제안하는 특징 추상화 기법은 유사한 특징 단어들이 묶여 하나의 개념을 구성하며, 각 개념에 속한 특징 단어들의 문서 내 빈도수를 모두 합하여 이를 그 개념의 가중값으로 정한다. 결국 하나의 개념은 관련 특징들의 집합과 그것의 가중값으로 표현한다. 본 논문에서 제안하는 특징 추상화 방식은 개념을 어떻게 정의하는지에 따라 '계층적 특

징 추상화' 방식과 '평면적 특징 추상화' 방식으로 구분한다.

2.3.1 계층적 특징 추상화

계층적 추상화 방식은 계층적 군집화(hierarchical agglomerative clustering) 알고리즘[17]과 유사한 방식으로 단어들의 군집(또는 클러스터)을 생성한다. 즉 초기에 1개 단어로 구성된 군집 집합에 대하여 가장 가까운 2개의 군집들을 연속적으로 묶어나가며, 군집 간의 거리가 임계값을 초과하기 전까지 군집화 과정을 진행한다. 이 때 사용하는 유사도 함수는 식 (1)과 동일하다. 결과적으로 생성된 각 군집이 하나의 개념을 정의한다. 각 개념은 한 개 이상의 특징으로 구성되며, 이를 분류모델의 요소로 삼는다. 결국 단어의 군집화는 개념을 생성하는 추상화를 의미한다. 각 개념은 자신의 가중값으로서 소속된 특징들의 빈도수의 합계를 취한다. 실제적으로 그 값을 소속 특징 단어에 반영함으로써 특징 추상화 이후 문서 d의 단어 f_i의 빈도수(term frequency) $tf(f_i, d)$는 식 (2)에 따라 반복적으로 보정된다.

$$tf(f_i, d) = tf(f_j, d) = tf(f_i, d) + tf(f_j, d) \quad (2)$$

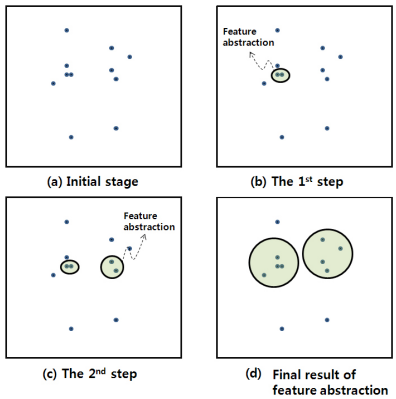
여기서 $tf(f_i, d)$는 문서 d에서 본래 특징

```
    INPUT : feature word set F = {f1, f2, ..., fN}
    OUTPUT : improved feature word set F
1   BEGIN
    /* Calculating the similarity values among features */
2   WHILE (max_sim > τ ) {            /* τ: threshold value */
3       FOR (i = 1; i < |F|; i++){          /* |F|: Size of feature word set */
4           FOR (j = 1; j < |F|; j++) fsim[i][j] = sim(fi, fj); /* Using similarity function Equation (1),
5       }  the similarity value between features fi and fj is kept in the 2-dimensional array fsim. */
    /* Developing the feature set while hierarchically grouping two feature words with high similarity */
6       max_sim = 0;
7       FOR (i = 1; i < |F|; i++) {
8           FOR (j = 1; j < |F|; j++) {
9               IF (max_sim < fsim[i][j]) {
10                  max_sim = fsim[i][j];  /* The maximum similarity value is saved in max_sim */
11                  fnew = {i, j};       /* The features fi and fj with the highest similarity is saved in  fnew*/
12                  F = F-{i, j}; F = F∪{fnew}; /* The abstracted feature is included in the final feature set*/
13              }
14          }
    /* Adjusting the frequency of feature for feature abstraction */
15          IF (max_sim > τ) {
16              tf(fi) = tf(fj) = tf(fi) + tf(fj); /*The frequency of fi and fj is added up to each other.*/
17          }
18      }
19  }
20  END
```

〈Figure 6〉 Algorithm of Hierarchical Feature Abstraction

단어 f_i의 빈도수를 의미한다. 이 식에 따라 하나의 개념으로 묶인 특징들은 초기의 빈도수가 반복적으로 보정된다.

문서에 존재하는 모든 특징 간 연관도를 계산하고 가장 높은 연관도를 가지는 2개의 특징을 임계값 이상이면 추상화한다. <Figure 6>은 계층적 특징 추상화 알고리즘을 보여준다. 이것의 입력은 진입링크 문서로부터 추출된 특징 집합이며, 출력은 특징 추상화를 적용하여 개선된 특징 집합이다. 우선 모든 특징 간 유사도를 계산한다(2~5행 참조). 그 후 가장 유사도가 높은 두 특징을 찾고, 이를 하나의 개념 특징으로 저장한다(6~15행 참조). 유사도가 임계값 이상이라면 개념 특징으로 추상화된 특징의 가중값은 소속 특징의 보정된 빈도수의 합으로 정한다(15~17행 참조). 새로이 생성된 개념 특징은 특징 집합에 첨가되어 반복적으로 추상화 과정에 참여한다. 중간 과정에서 생성되는 군집(개념)간의 거리는 average-linkage 방식[17]으로 계산하며, 그 반복 작업은 현재 유사도의 최대값이 임계값 미만이면 중단한다.



〈Figure 7〉 Hierarchical Feature Abstraction Process

<Figure 7>은 계층적 특징 추상화 과정을 도식화한 것이다. 각 점은 특징을 의미하고 점 사이의 거리는 특징 간 의미적 거리를 의미한다. (a)는 초기에 입력으로 주어진 특징 집합을 나타낸다. (b)와 (c)는 군집화 과정에서 특징 간 유사도를 계산하고, 가장 유사도가 높은 두 특징을 추상화하여 개념 특징을 생성한 중간 결과를 보여준다. (d)는 추상화 과정을 멈추어 최종적으로 2개의 개념 특징을 생성한 것을 보여준다.

2.3.1 평면적 특징 추상화

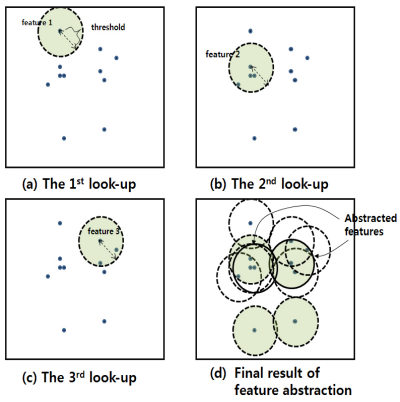
평면적 특징 추상화 방식은 주어진 특징 집합 내의 전체 특징을 대상으로 일정 거리 이내의 특징들을 하나의 개념으로 묶는 것이다. 예를 들어 특징 집합 {A, B, C, D}가 주어질 때, A를 기준으로 A를 제외한 모든 특징 B, C, D와의 거리를 계산한다. 이 때 B, C가 일정 거리 이내라면 {A, B, C}가 하나의 개념으로 묶인다. 마찬가지로 B를 기준으로 B를 제외한 A, C, D와의 거리를 계산하고, 그 결과 A, D가 일정 거리 이내이면 {B, A, D}가 하나의 개념으로 묶인다. 결국 평면적 특징 추상화 과정을 통해 최대 특징 개수와 동일한 개수의 개념을 생성한다. 다른 특징과 연관도가 작은 특징은 개념 생성에 참여하지 않는다. 각 개념은 한 개 이상의 특징으로 이루어져 있으며, 소속 특징 단어들의 빈도수를 모두 합하여 이를 개념의 가중값으로 정한다. 즉 특징 추상화를 생성된 개념 및 소속 특징 단어 f_i가 문서 d에서 가지는 가중값 $tf'(f_i, d)$은 식 (3)에 의해 계산된다.

$$tf'(f_i, d) = \sum_{\mathrm{sim(f,\ f_i)} > \tau} tf(f, d) \qquad (3)$$

여기서 우측항의 f는 일정 임계값 τ 이상의 연관도를 가지는 특징들을 의미하고, $tf(f, d)$는 그 특징 f가 문서 d에서 가지는 초기 빈도수를 의미한다.

평면적 추상화 알고리즘은 각 특징에 대해 문서에 존재하는 모든 특징 간 연관도를 계산한 후 일정 임계값 이상의 연관도를 가지는 특징들을 묶어 개념으로 추상화한다. <Figure 8>은 평면적 특징 추상화 알고리즘을 보여준다. 이것의 입력은 진입링크 문서로부터 추출된 특징 집합이며, 출력은 특징 추상화를 적용하여 개선된 특징 집합이다. 우선 임의의 특징을 기준으로 다른 모든 특징 간의 유사도를 계산한다(2~4행 참조). 이 결과를 이용하여 기준 특징과 다른 특징과의 유사도가 임계값 이상이면 해당 특징에 단어 빈도수를 합산한다(5~11행 참조). 이 과정을 모든 특징을 기준으로 반복 수행한다.

<Figure 9>는 평면적 특징 추상화 과정을 도식화한 것이다. (a)는 1번 특징을 기준으로 다른 특징과의 유사도를 계산한 후 거리가 임계값 이하인 특징들을 모두 추상화한 모습을 나타낸다. (b)와 (c)는 2번, 3번 특징을 기준으로 위 과정을 반복한 것이고, (d)는 모든 특징에 대해 특징 추상화를 끝낸 최종 결과를 보여준다.



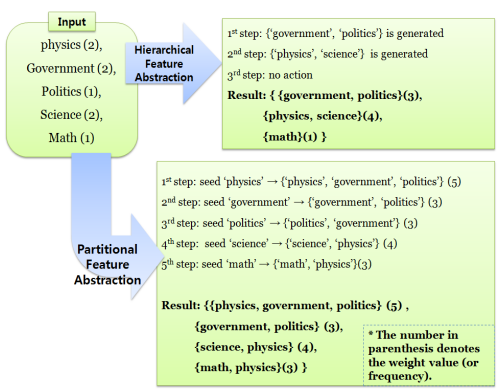
〈Figure 9〉 Partitional Feature Abstraction Process

	INPUT : feature word set F = {f_1, f_2, ..., f_N} OUTPUT : improved feature word set F
1	BEGIN
	/* Calculating the similarity values among features */
2	FOR (i = 1; i < |F|; i++) { /* |F| : Size of feature word set */
3	FOR (j = 1; j < |F|; j++) fsim[i][j] = sim(fi, fj); /*Using similarity function Equation (1),
4	} the similarity value between features *fi* and *fj* is kept in the 2-dimensional array *fsim*.*/
	/* A group of highly similar features is regarded as a single feature. */
5	FOR (i = 1; i < |F|; i++) {
6	FOR (j = 1; j < |F|; j++) {
7	IF (fsim[i][j] > τ) { /* τ: threshold value */
8	tf(fi) + = tf(fj); /* The frequency of *fj* which is highly similar to *fi* is added up
9	} to the frequency of *fi* */
10	}
11	}
20	END

〈Figure 8〉 Algorithm of partitional feature abstraction

2.3.3 특징 추상화의 예시

<Figure 10>은 두 가지 방식의 특징 추상화 알고리즘에 의해 생성된 개념 집합과 보정된 가중값을 예시한 것이다. 계층적 특징 추상화 방식에서는 1단계 반복에서 특징간 의미적 연관도를 계산하여 최고의 연관 관계를 가지는 {'government', 'politics'}가 하나의 개념으로 묶이고 이는 특성 집합에 포함된다. 2단계 반복에서 {'physics', 'science'}가 하나의 개념으로 묶이고, 3단계 반복에서 개념 특성간 거리가 임계값을 초과하여 추상화 과정을 중단한다. 결과적으로 생성된 개념은 {'government', 'politics'}, {'physics', 'science'}, {'math'}가 되고, 각 개념의 가중값은 각 개념에 포함된 특징들의 빈도수를 합산한 것이다. 평면적 특징 추상화 방식에서는 각 특징을 기준으로 연관도가 높은 특징을 묶어내어 결과적으로 {'physics', 'science', 'math'}, {'government', 'politics'}, {'science', 'physics'}, {'math', 'physics'}이 된다. 여기서 개념 {'politics', 'government'}은 개념 {'government', 'politics'}와 중복되므로 삭제되었다.



〈Figure 10〉 The Result of Feature Abstraction

3. 성능 평가

3.1 실험 방법

본 연구는 제안한 기법의 효능을 검증하기 위해서 Web-KB 문서집합을 활용하였다. 이 문서집합은 CMU text learning group이 웹문서 자동분류를 목적으로 구축한 것으로서, 4개 대학교의 웹페이지를 7개의 클래스로 구분하여 약 8,300개의 웹페이지로 구성되어 있다. 본 실험에서는 클래스에 포함된 문서 수가 10개 미만인 문서는 제외하여 결과적으로 4개의 클래스로 구분된 1,224개의 문서를 사용하였다. 특징(feature)으로 사용되는 것은 앵커 텍스트 및 앵커 주변에 존재하는 단어들의 집합이며, 분류 대상 문서에 포함된 일반 특징 단어는 제외한다. 이는 워드넷을 활용하여 웹 문서간 링크 관계로부터 특징을 확장 추출한 효과만을 확인하기 위함이다. 제안 기법은 자동분류 모델을 구성하는 주요 인자를 구성하는 것이므로 자동분류 알고리즘과 독립적이기 때문에, 그 성능이 자동분류 알고리즘의 특성에 의존하지 않는다. 그리고 특성 가공의 결과를 직접적으로 평가하는 것은 어렵기 때문에 분류 모델을 생성하여 이를 테스팅함으로써 가공 결과를 간접적으로 평가하게 된다. 본 실험에서는 제안 기법의 성능을 평가하기 위해 나이브베이즈 학습 알고리즘을 채택하였으며, 이를 실제 구동하기 위해 MALLET(MAchine Learning for LanguagE Toolkit) 시스템[7]을 활용하였다. 평가 척도는 분류 정확도(classification accuracy)이며, 이는 분류한 문서 개수에 대한 맞는 분류 문서의 개수의 비율값을 의미한다. 그리고 분류 모델의 공정한 검증을 위해 10겹 교차검

증(10-fold cross validation) 방식을 사용한다.

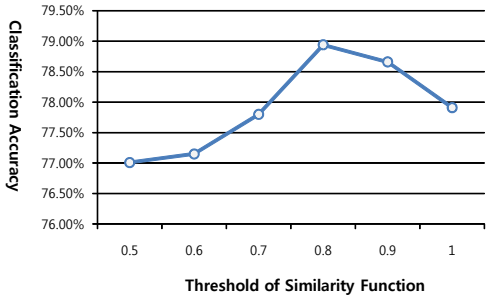
본 논문이 제시한 <Figure 2>의 시스템 구조에서 보는 바와 같이, 분류모델을 구축하기 위해 사전에 특징 추상화 단계 이후 특징 선택을 수행한다. 추상화된 특징 집합은 일반 단어 수준의 특징 집합을 병합하여 이로부터 특징 선택을 통해 양질의 특징을 추출한다. 이를 위해 본 실험에서는 카이제곱 통계량 기반 특징 선택 알고리즘을 사용하였다. 이를 통해 각 클래스별로 연관성이 높은 특징 집합을 선별할 수 있다.

3.2 실험결과

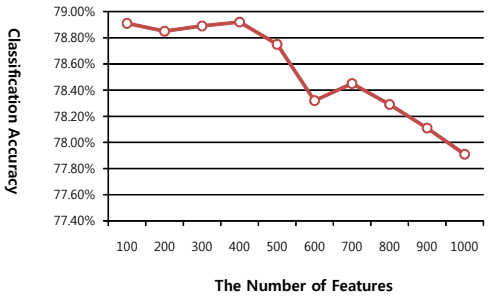
<Figure 11>은 특징 추상화 과정에서 임계값의 변화에 따른 자동분류 정확도의 변화를 보여준다. 그림에서 보는 바와 같이 임계값이 0.8일 때 가장 높은 분류 정확도를 보였다. 그리고 임계값이 0.8보다 작은 경우 급격한 성능 저하 현상을 보이고, 0.8보다 큰 경우 완만한 성능 저하 현상을 보였다. 전자의 경우는 임계값이 0.8보다 작고 0에 가까울수록 많은 특징들이 하나의 개념으로 통합되어 관련 특징들의 빈도수가 과도하게 보정되기 때문이다. 후자의 경우, 임계값이 1에 가까우면 특징 집합 내에서 완벽한 동의어만이 특징 추상화의 대상이 되어 제안 기법의 효과가 발휘되지 못하는 것이다.

<Figure 12>는 특징선택 과정을 통해 특징 개수의 변화에 따른 자동분류 정확도의 변화를 보여준다. 이는 특징 개수를 변화하면서 기존 기법과 제안 기법을 10회 실시하여 해당하는 자동분류 정확도 값의 평균값을 보여준다. 그림에서 보는 바와 같이 특징 개수가 많을수록 분류 정확도가 낮아지는 경향을 알 수 있다. 이는 자동분류시스템의 구축에 있어서 특징 선택 과정의 중요도가 크다는 것을 보여준다. 본 실험에서는 특징 개수가 400개일 때 가장 높은 정확도를 보였으며, 특징 개수가 500개를 초과하면서 분류 성능이 급격히 낮아지는 것을 확인할 수 있다.

위 실험에 근거하여 특징 개수는 400개, 임계값은 0.8로 고정하였으며, 최종적으로 <Table 1>에서 보는 바와 같이 특징 추출 방법에 따른 분류 정확도와 특징 추상화 전·후의 분류 정확도를 보여주고 있다. 여기서 '확장된



〈Figure 11〉 Classification Accuracy from Varying the Threshold Value



〈Figure 12〉 Classification Accuracy from Varying the Number of Features

〈Table 1〉 Comparison of Classification Accuracy (%)

	Type	No abstraction	Hierarchical abstraction	Partitional abstraction
Conventional methods	Anchor text	77.12	78.26	78.14
	Expanded anchor text	77.01	77.16	77.89
The proposed method	WordNet-based expanded anchor text	77.26	78.82	**78.94**

앵커텍스트(expanded anchor text)'는 앵커텍스트와 그것의 주변에 존재하는 모든 특징을 취한 기법을 의미한다. 비 추상화 부문에서 제안 기법인 '워드넷 기반 확장된 앵커 텍스트(WordNet-based expanded anchor text)'를 사용한 경우가 기존 기법에 비해 약 0.1~0.3%의 미약한 향상을 보였다. 주목할 점은 기존 기법에서 앵커 텍스트를 사용하였을 때가 확장된 앵커 텍스트를 사용할 때보다 분류 정확도가 높다는 점이다. 이는 웹문서의 앵커 텍스트 주변 단어들 중에서 대상 문서와 연관도가 떨어지는 단어들이 많이 존재한다는 것을 반증한다. 비교해서 제안 기법은 의미적 연관도가 큰 단어들을 선별적으로 추출함으로써 성능 저하를 방지하였다. 제안 기법은 평균적으로 계층적 기법이 1.7%, 평면적 기법이 1.8% 가량의 분류 정확도를 높였다. 제안하는 계층적, 평면적 추상화 기법에 따른 성능 차이는 크지 않았으며, 워드넷에 기반하여 의미적 연관도가 높은 특징 단어를 추출하여 평면적 특징 추상화를 수행한 방법이 상대적으로 좋은 효과를 보였다. 실험을 통해 얻은 제안 기법의 성능 향상 정도는 높지는 않지만 앵커 텍스트와 그 주변 텍스트만을 활용하여 문서 분류에 도움을 주는 특징을 추출할 수 있음을 보여준다. 특징 단어의 의미적 유사도를 평가하는 함수가 문서의 단어 분포를 고려한다면 추가적인 성능 향상을 기대할 수 있을 것이다.

4. 요약 및 결론

본 논문은 기계학습 기반 웹문서 자동분류 시스템의 성능을 높이기 위해 새로운 특징 가공 기법을 제안하였다. 본 연구는 웹문서에 대한 자동분류의 성능 향상을 위해 학습 알고리즘 보다는 특징 집합의 질을 높이는 것이 더 중요하다는 인식하에 특징 가공에 초점을 맞추었다. 제안 기법은 하이퍼링크 관계를 활용하여 개념적 특징을 생성하여 기존 특징 집합을 확장한 것이며, 이를 위해 하이퍼링크 정보와 단어의 의미 정보를 활용하였다. 하이퍼링크 정보를 통해 진입링크 문서내의 용어들 중 앵커 텍스트와 유사도가 높은 특징들을 추출하였으며, 의미 정보를 이용하여 유사도가 높은 특징들을 대상으로 개념 특징을 생성하는 특징 추상화를 수행하였다. 제안 기법은 Web-KB 문서집합을 이용한 실험을 통해 기존 기법보다 평균 1.7% 가량 자동분류 정확도를 높임으로써, 앵커 텍스트와 그 주변 텍스트에 존재하는 단어들만 가지고 새로운 특

징을 추출 확장하는 것이 의미가 있음을 보여준다. 제안 기법의 관건은 워드넷에 기반한 단어 간 의미적 유사도를 정확하게 산출하는 것이다. 향후 워드넷의 의미 정보와 분류 대상 문서의 단어 분포를 동시에 고려한 유사도 함수를 고안하여 현재 특징 추상화 기법을 개선하고자 한다. 또한 제안 기법을 베이지안 통계 기반 시멘틱 추천 기법[5]과 오피니언 마이닝 기법[2]에 적용하여 추천 및 감성 분석의 정확도를 높이고자 한다. 이는 시멘틱 추천 기법과 오피니언 마이닝 기법이 특징 집합의 질에 크게 좌우되기 때문이다.

References

[1] Chakrabarti, S., Dom, B., and Indyk, P., "Enhanced hypertext categorization using hyperlinks," Proceedings of the ACM SIGMOD International Conference, pp. 307-318, 1998.

[2] Chang, J. Y., "A Sentiment Analysis Algorithm for Automatic Product Reviews Classification in On-Line Shopping Mall," The Journal of Society for e-Business Studies, Vol. 14, No. 4, pp. 19-33, 2009.

[3] Elberrichi, Z., Rahmoun, A., and Bentaalah, M. A., "Using WordNet for Text Categorization," The International Arab Journal of Information Technology, Vol. 5, No. 1, pp. 16-24, 2008.

[4] Jiang, J. and Conrath, D., "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," Proceedings of International Conference on Research in Computational Linguistics, pp. 19-33, 1997.

[5] Lee, J. W., Park, S. C., Lee, S. K., Park, J. H., Kim, H. J., and Lee, S. G., "Semantic Search and Recommendation of e-Catalog Documents through Concept Network," The Journal of Society for e-Business Studies, Vol. 15, No. 3, pp. 131-145, 2010.

[6] Lu, Z., Liu, Y., Zhao, S., and Chen, X., "Study on Feature Selection and Weighting Based on Synonym Merge in Text Categorization," Proceedings of the 2nd International Conference on Future Networks, pp. 105-109, 2010.

[7] MALLET, MAchine Learning for Language Toolkit, http://mallet.cs.umass.edu/.

[8] Mansuy, T. and Hilderman, R., "Evaluating WordNet Features in Text Classification Models," Proceedings of the 19th International Florida Artificial Intelligence Research Symposium, pp. 568-573, 2006.

[9] Mitchell, T. M., Machine Learning, McGraw-Hill, 1997.

[10] Oh, H. J. and Myaeng, S. H., "A Hypertext Categorization Method using Incrementally Computable Class Link Information," Journal of Korean Institute of Information Scientist and Engineers, Vol. 29, No. 7-8, pp. 498-509, 2002.

[11] Oh, S. J., Ahn, J. H., and Park, J. S., "Ontology Selection Ranking Model based

on Semantic Similarity Approach," The Journal of Society for e-Business Studies, Vol. 14, No. 2, pp. 95-116, 2009.

[12] Priss, U., "Formal Concept Analysis in Information Science," Annual Review of Information Science and Technology, Vol. 40, No. 1, pp. 521-543, 2006.

[13] RiTa.WordNet, A WordNet library for Java/Processing, http://rednoise.org/rita/wordnet/documentation/index.htm.

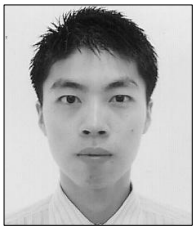
[14] Scott, S. and Matwin, S., "Feature engineering for text classification," Proceedings of 16th International Conference on Machine Learning, pp. 379-388, 1999.

[15] Utard, H. and Fürnkranz, J., "Link-Local Features for Hypertext Classification," Semantics, Web and Mining : Joint International Workshops, Lecture Notes in Computer Science, Vol. 4289, pp. 51-64, 2005.

[16] Zhang, B., Yan, J., Chen, Z., Liu, W., Bai, F., and Chien, L., "Text representation: from vector to tensor," Proceedings of 5th IEEE International Conference on Data Mining, pp. 725-728, 2005.

[17] Zhao, Y., Karypis, G., and Fayyad, U., "Hierarchical Clustering Algorithms for Document Datasets," Data Mining and Knowledge Discovery, Vol. 10, No. 2, pp. 141-168, 2005.

저 자 소 개



노준호 (E-mail : loece@naver.com)
2011년 서울시립대학교 전자전기컴퓨터공학부 (공학사)
2013년 서울시립대학교 전자전기컴퓨터공학부 대학원 (공학석사)
2013년~현재 LG전자
관심분야 데이터마이닝, 기계학습, 정보검색

김한준 (E-mail : khj@uos.ac.kr)
1994년 서울대학교 계산통계학과 (공학사)
1996년 서울대학교 전산과학과 (공학석사)
2002년 서울대학교 컴퓨터공학부 (공학박사)
2002년~현재 서울시립대학교 전자전기컴퓨터공학부 부교수
관심분야 정보검색, 텍스트마이닝, 데이터베이스, 기계학습, e-비즈니스 기술

장재영 (E-mail : jychang@hansung.ac.kr)
1992년 서울대학교 계산통계학과 (이학사)
1994년 서울대학교 계산통계학과 (이학석사)
1999년 서울대학교 계산통계학과 (이학박사)
2000년~현재 한성대학교 컴퓨터공학과 교수
관심분야 데이터베이스, 정보검색, 데이터마이닝