# 메모리 기반의 기계 학습을 이용한 한국어 문장 경계 인식

## Korean Sentence Boundary Detection Using Memory-based Machine Learning

임희석*, 한군희**
한신대학교 소프트웨어학과*, 천안대학교 정보통신학부**
Heui-Seok Lim(limhs@hs.ac.kr)*, Kun-Heui Han(hankh@cheonan.ac.kr)**

**요약**

본 논문은 기계 학습 기법 중에서 메모리 기반 학습을 사용하여 범용의 학습 가능한 한국어 문장 경계 인식기를 제안한다. 제안한 방법은 메모리 기반 학습 알고리즘 중 최근린 이웃(kNN) 알고리즘을 사용하였으며, 이웃들을 이용한 문장 경계 결정을 위한 스코어 값 계산을 위한 다양한 가중치 방법을 적용하여 이들을 비교 분석하였다. 문장 경계 구분을 위한 자질로는 특정 언어나 장르에 제한적이지 않고 범용으로 적용될 수 있는 자질만을 사용하였다. 성능 실험을 위하여 ETRI 코퍼스와 KAIST 코퍼스를 사용하였으며, 성능 척도로는 정확도와 재현율이 사용되었다. 실험 결과 제안한 방법은 적은 학습 코퍼스만으로도 98.82%의 문장 정확률과 99.09%의 문장 재현율을 보였다.

**Abstract**

This paper proposes a Korean sentence boundary detection system which employs k-nearest neighbor algorithm. We proposed three scoring functions to classify sentence boundary and performed comparative analysis. We uses domain independent linguistic features in order to make a general and robust system. The proposed system was trained and evaluated on the two kinds of corpus; ETRI corpus and KAIST corpus. As experimental results, the proposed system shows about 98.82% precision and 99.09% recall rate even though it was trained on relatively small corpus.

## Ⅰ. 서 론

전통 문법에서는 "문장이란 비교적 완전하고 독립된 의사 전달의 한 단위다"라고 정의한다. 품사 태거나 파서와 같은 대부분의 자연어처리 관련 도구는 기본 처리 단위로서 문장을 사용하며, 문서가 이미 문장 구분이 되어 있다고 가정한다.

그러나 문장을 구분하는 정확한 방법에 대해서는 언급하고 있지 않다[1]. 따라서 텍스트에서 문장의 경계를 결정하는 문장 경계 인식 작업이 반드시 요구된다. 그러나 마침표, 느낌표, 그리고 물음표 등의 문장 기호들이 문장 내에서 다른 의미로도 사용될 수 있으므로 문장의 경계를 인식하는 작업은 간단한 규칙에 의하여 처리될 수 있는 작업이 아니다. 비록 문장의 끝을 결정지을 수 있는 규칙 집합을 구축하여 사용하더라도 확장성과 적용 범위가 넓지 못한 단점을 야기할 수 있다. 예를 들어, 문장은 마침표, 느낌표, 물음표 중 어느 하나로 끝난다. 그러나 모든 마침표가 문장의 끝에만 존재하는 것은 아

접수번호 : #041021-002
접수일자 : 2004년 10월 21일
심사완료일 : 2004년 11월 08일
교신저자 : 임희석, e-mail : limhs@hs.ac.kr

니라 문장 내에서 다른 의미를 나타내기 위해서도 사용된다. 소수점, e-mail 주소, 생략 부호(...)나 약자 등에도 마침표가 사용된다. "!"나 "?"는 다소 덜 중의적이지만 고유 이름이나 강조를 위해 한 문장에서 여러 번 쓰일 수도 있다. 따라서 정확한 문장의 경계를 인식하기 위해서는 문장 기호가 사용된 문맥과 문맥 내의 언어 자질 정보를 이용하여 문장 경계를 의미하는 것인가에 대한 분류 작업이 필요하다.

문장 경계 인식을 위해서는 문장 경계를 결정할 수 있는 규칙을 전문가가 작성하여 이를 이용하는 규칙기반 방법이 사용될 수 있으나 이 방법은 규칙을 구축하는데 많은 비용이 소모되고, 성능이 규칙 작성을 위하여 사용된 언어 영역에 제한적일 수 있다.

기계 학습 기법 중에서 메모리 기반 학습과 언어 영역과 장르에 독립적인 자질 정보를 활용하는 범용의 한국어 문장 경계 인식기를 제안한다. 메모리 기반 학습은 나태(lazy) 학습의 일종으로서 학습 시에는 단순히 데이터를 저장하고, 분류 시에 메모리에 저장된 데이터 중에서 유사한 것들을 찾아내어 이들로부터 분류를 한다. 분류할 데이터 각각에 대해 다른 가설(hypothesis)을 생성할 수 있으므로 복잡한 목적 함수(target function)를 표현할 수 있다는 장점을 갖는다[2].

## II. 관련 연구

Grefenstette (1994)는 정규 표현식을 이용하여 약 95%의 정확도를 보였다. 그러나 규칙에 기반한 방법은 모든 경우를 포함할 수 없고 여러 규칙들이 서로 충돌을 일으킬 수 있다. 수작업에 의존해야 하므로 규칙을 작성하기가 어렵고, 비용이 매우 크다[3].

최근의 연구는 주로 기계 학습 방법을 이용하고 있다. Riley (1989)는 결정 트리를 이용하여 Brown 말뭉치에서 99.8%의 정확도를 보였다. 이는 2500만 단어를 가진 말뭉치에서 학습한 결과이다[4].

Palmer and Hearst (1997)는 결정 트리와 신경망, 품사 정보를 포함하는 사전을 이용하여 WSJ 말뭉치에서 98.5%의 정확도를 보였다[5]. 이 방법은 품사 부착 말뭉치가 필요하므로 장르나 언어에 대해 한계를 지닌다.

Ratnaparkhi (1996)은 최대 엔트로피 모델을 이용하여 WSJ 말뭉치에서 98.8%, Brown 말뭉치에서 97.9%의 정확도를 보였다[1].

위와 같이 기존의 문장 경계 인식에 관한 연구는 주로 영어에 대하여 수행되었으며 한국어 문장 경계 인식에 관한 연구는 미흡한 실정이다. 그러나 구문 분석과 의미 분석 등 전통적인 한국어 정보처리뿐만 최근 웹로봇에 의한 정보 자동 수집, 정보검색, 문서 자동 분류, 그리고 스팸 메일 필터링과 같은 비정형화된 문서 데이터의 처리를 위해서도 한국어 문장 경계 인식에 관한 연구는 매우 필요한 실정이다. 이에 본 논문은 메모리 기반의 학습 방법을 이용한 한국어 문장 경계 인식 방법을 제안한다.

## III. 메모리 기반 학습을 이용한 문장 경계 인식

### 1. kNN 알고리즘

메모리 기반 학습 중에서 가장 많이 사용되는 알고리즘은 kNN(k-nearest neighbor) 알고리즘이다[6, 7]. kNN 학습 알고리즘은 학습 데이터와 입력 데이터 모두가 n개의 자질 정보로 표현되는 벡터로 n차원의 공간의 한 점에 대응된다. 예를 들어, $\vec{x}_j = (x_{1j}, x_{2j}, x_{3j}, \ldots, x_{nj})$는 문장 경계 결정을 위하여 사용될 수 있는 n개의 자질 정보로 표현된 문장 경계 입력값을 나타낸다. 벡터로 표현된 문장 경계의 입력값들간의 거리 또는 유사도는 식 1과 같은 유클리디안 거리 또는 식 2와 같은 코사인 유사도 값으로 계산할 수 있다.

$$d(d_i, d_j) = \sqrt{\sum_{r=1}^{n}(d_{ir} - d_{jr})^2} \quad (1)$$

$$sim(d_i, d_j) = \frac{\vec{d_i} \cdot \vec{d_j}}{\left|\vec{d_i}\right| \times \left|\vec{d_j}\right|} \quad (2)$$

kNN 알고리즘은 (식 1) 또는 (식 2)를 이용하여 분류할 입력값과 유사한 k개의 이웃을 찾아 이 이웃들의 분류값이나 유사도 값을 이용하여 입력값의 분류값을 결

정한다.

따라서 kNN 알고리즘의 성능은 이웃의 크기인 k값과 k개의 이웃에 어떤 가중치 값을 주는가에 좌우된다. 본 논문은 다양한 k값에 따른 실험을 통하여 가장 우수한 성능을 보이는 k값을 결정하여 사용하며 가중치 부여 방법은 다음과 같은 3가지를 사용한다. 'SW(Same Weight)'는 모든 개체들에 똑같은 가중치를 부여하는 것으로 다수결에 의해 결정되는 방법이다. 'ID(Inverse Distance)'는 (식 3)과 같이 거리에 따라 가까운 것은 더 높은 score 값을 부여하고 먼 것은 낮은 값을 부여하는 방법이다.

$$w_j = \frac{1}{d_j} \text{ if } d_j \neq 0 \qquad (3)$$

'IL(Inverse Linear)'는 가장 가까운 것에는 score 값 1을 부여하고, 제일 먼 것에는 가중치 0을 부여한다. 나머지는 (식 4)와 같이 선형적인 계산법에 의해 가중치를 부여한다.

$$s_{ij} = \begin{cases} \dfrac{\max d_{jk} - d_{ij}}{\max d_{ik} - \min d_{il}} & \text{if } d_{ij} \neq 0 \\ 1 & \text{if } d_{ij} = 0 \end{cases} \qquad (4)$$

(식 3)과 (식 4)에서 $s_{ij}$는 입력 $x_i$를 분류하기 위한 이웃 $x_j$의 score 값을 의미하며 $\max d_{ik}$와 $\min d_{ik}$는 각각 입력 $x_i$에서 거리가 가장 먼 이웃까지의 거리와 가장 가까운 이웃까지의 거리를 의미한다.

## 2. 문장 경계 인식을 위한 자질 집합

본 논문은 학습된 문장 경계 인식기가 학습 데이터로 사용된 언어 영역에 제한적이지 않도록 언어 영역과 독립적인 다음의 6가지 언어 자질을 사용하였다.

### 2.1 해당 후보 구두점 자체

각 구두점마다의 특징이 다르다는 가정에 기반한 것이다. 후보 구두점은 ".", "?", "!", "〃", "'" 로서 모두 5가지이다. 단, 후보 구두점이 어절의 마지막 음절로 사용된 것만 학습 및 실험 데이터의 후보로 사용하였다.

### 2.2 후보 구두점의 바로 앞에 나타난 음절

각 구두점마다 사용되는 종결 어미의 종류가 다르다는 점을 이용한 것이다. 예를 들어, "."가 문장의 끝일 경우, 바로 앞 음절에 '다'가 많이 나타나고 "?"의 경우는 "니"나 "까"가 많이 나타난다.

### 2.3 후보 구두점의 바로 뒤에 나타난 음절

앞의 자질과 유사한 가정에 의한 것이다. 특히, """가 문장의 끝이 아닐 경우 그 뒤에는 "~라고", "~하고", "~하면서", "~하는" 등의 어휘가 많이 나타난다.

### 2.4 문장의 시작에서부터 후보 구두점이 나타난 위치까지의 거리

이 자질은 일반적으로 작은 음절수로 구성된 것보다 많은 음절수로 구성된 경우가 문장의 끝이 될 가능성이 많을 것이라는 가정에 기반한 것이다. 거리는 음절의 수로써 구하였다.

### 2.5 "의 개수에 대한 이진 값

"의 수에 따라 현재 열려 있는 상태인지 닫혀있는 상태인지를 파악하는 것이다. 그 수가 홀수이면 열려있는 상태이고 이 때 발견된 후보 구두점들은 문장의 경계가 될 수 없다는 특징을 이용한 것이다. 즉, 따옴표 내에 여러 개의 문장이 있더라도 이들 각각은 문장으로 간주하지 않는 것이다.

### 2.6 '의 개수에 대한 이진 값

위의 자질과 동일한 가정에 의해 사용된 자질이다. 예를 들면, 아래의 문장 그림 1에 대해서 윗 문장에 있는 후보 구두점에 대해 얻은 벡터가 그림 2이다.

"사과? 그거 조오치! 그래 사과해."
나는 빈정대듯 말했다.

그림 1. ETRI 말뭉치 예문

```
?,과,그,4,1,0,-
!,치,그,10,1,0,-
",0,.,나,16,0,0,+
```

그림 2 벡터화

## Ⅳ. 실험 및 결과

본 논문에서 사용한 말뭉치는 27,855개의 문장을 포함한 ETRI 원시 말뭉치[8]와 48,858개의 문장을 포함한 KAIST 언어 자원 의존 구조 부착 말뭉치[9]이다. KAIST 말뭉치는 여기에서 문장만을 따로 뽑아 원시 말뭉치를 구성하여 실험하였다. 이들 말뭉치들의 학습 양은 각각 32,381개, 49,358개였다. 문장으로 분리된 한국어 말뭉치는 부족한 실정이므로 학습 및 실험은 10-fold cross validation 방법을 사용하였다.

평가 척도로는 문장 정확률(P), 문장 재현율(R) 그리고 정확률과 재현율을 하나의 값으로 표현할 수 있는 F-measure를 사용하였으며 각 척도의 정의는 다음과 같다.

$$P = \frac{\text{시스템이출력한정답문장수}}{\text{시스템이출력한문장수}}$$

$$R = \frac{\text{시스템이출력한정답문장수}}{\text{전체문장수}}$$

$$F = \frac{2RP}{R+P}$$

### 1. ETRI 원시 말뭉치에 대한 실험

처음 실험은 ETRI 말뭉치만으로 하였다. 이 말뭉치의 특성은 대부분이 구어체로 되어있다는 것이다. 그러므로 KAIST 말뭉치와 같이 문어체가 대부분인 말뭉치보다는 한 문장 내에 후보 구두점이 많이 나타나고 문장 경계를 인식하기에 어렵다.

초기 실험은 이웃의 크기, k값을 1로 실험하였다. 즉, 실험 데이터와 가장 유사한 한 개의 데이터를 학습 데이터로부터 추출하여 실험 데이터를 분류하게 된다. 이렇기 때문에 개체에 대한 가중치 부여 방법은 의미가 없으나 k를 2이상으로 할 경우 가중치 부여 방법에 따라 성능이 다르게 나타나게 된다. 이에 대한 실험은 4.2절에 하였다.

문장 경계 인식 문제의 기준선(baseline)은 후보 구두점이 발견되었을 때 무조건 문장의 경계라고 간주하는 방법이다. 아래의 표는 10-fold Cross Validation에 의한 결과의 평균치이다.

표 1. ETRI 말뭉치의 실험 결과

| | 문장 정확률 | 문장 재현율 | F-measure |
|---|---|---|---|
| 기준선 | 82.87 | 100.00 | 90.63 |
| 본 시스템 | 96.41 | 97.23 | 96.82 |

메모리 기반 학습의 단점은 관련없는(irrelevant) 자질에 의해 결과가 쉽게 영향을 받는다는 점이다. 이에 따라 정보 획득률(gain ratio)을 기준으로 자질에 할당된 가중치를 조사하였고 그 결과는 표 2와 같다.

고려한 자질들을 비교해 본다면 표 2에서 볼 수 있듯이 6개의 자질 중, """가 열렸는지 닫혔는지를 파악하는 자질이 제일 좋다는 것을 알 수 있다.

가장 낮은 가중치가 할당된 자질 4를 제외한 후 실험한 결과는 표 3에 있다.

표 2. 자질의 가중치

| 자질 종류 | 가중치 |
|---|---|
| 자질 1 | 0.073 |
| 자질 2 | 0.104 |
| 자질 3 | 0.018 |
| 자질 4 | 0.008 |
| 자질 5 | 0.508 |
| 자질 6 | 0.024 |

표 3. 실험 결과

| 문장 정확률 | 문장 재현율 | F-measure |
|---|---|---|
| 96.73 | 97.42 | 97.07 |

표 3에서 보듯이 가중치가 낮은 자질을 제거한 실험이 모든 자질을 사용한 실험(표 1)보다 더 높은 성능을 내고 있다. 이후 실험에서는 자질4에 대해 고려하지 않

았다.

다음으로는 학습 데이터의 크기에 따른 성능의 변화를 살펴보았고, 그 결과는 그림 3과 같다. 전체 말뭉치를 10개로 나눈 후 이중에서 첫 번째 부분 말뭉치를 실험 데이터로 사용한 것이며, 1, 2, 3 등은 각각 사용된 부분 말뭉치의 수이고, 1/5, 2/5, 3/5 등은 각각 하나의 부분 말뭉치를 나누어 학습에 사용한 것이다.

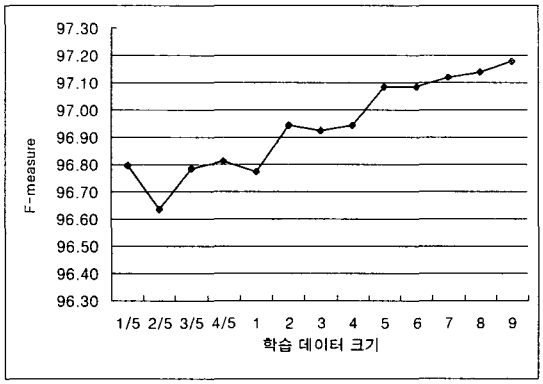


**그림 3. 학습 데이터의 크기에 따른 성능 변화**

그림에서 보듯이 학습 데능이 향상되는 것을 알 수 있으나, 일관성이 없는 경우도 관찰되었다.

## 2. 다른 말뭉치(KAIST 원시 말뭉치)와의 비교

지금까지는 대부분 구어체로 이루어진 ETRI 말뭉치에 대한 실험이었다. 기존의 관련 연구에 비하면 성능이 약간 낮은 편인데, 이는 말뭉치 특성상 한 문장 내에 여러 개의 후보 구두점이 나타나 성능이 떨어진 것으로 파악되었다. 그래서 대부분 문어체로 이루어진 KAIST 말뭉치에 대한 실험을 추가하였고 그 결과는 표 4와 같다.

**표 4. KAIST 말뭉치의 실험 결과**

| | 문장 정확률 | 문장 재현율 | F-measure |
|---|---|---|---|
| 기준선 | 98.64 | 100 | 99.32 |

기준선은 ETRI 말뭉치에서처럼 모든 후보 구두점에 대해 문장의 경계라고 가정한 것으로 KAIST 말뭉치에서는 높은 성능을 보인다. 이는 ETRI 말뭉치가 KAIST 말뭉치에 비해 경계 인식하는데 보다 어렵다는 것을 의미한다.

KAIST 말뭉치를 학습하면 자질들에 대한 가중치는 표 5에서와 같이 ETRI 말뭉치에서의 실험값과는 다소 차이를 보인다. 이 표에서 후보 구두점 자체의 가중치가 가장 높은 것은, KAIST 말뭉치의 문장이 후보 구두점을 두 개 이상 포함한 것이 적다는 것을 의미한다. 이 특징은 기준선(Baseline)이 높은 이유가 된다.

**표 5. 자질의 가중치**

| 자질 종류 | 가중치 |
|---|---|
| **자질 1** | **0.361** |
| 자질 2 | 0.077 |
| 자질 3 | 0.005 |
| 자질 5 | 0.278 |
| 자질 6 | 0.182 |

아래 표 6은 학습 말뭉치와 실험 말뭉치를 ETRI 말뭉치와 KAIST 말뭉치를 번갈아가며 실행한 것이다.

**표 6. 실험 결과**

| 학습 말뭉치 | 실험 말뭉치 | 문장 정확률 | 문장 재현율 | F-measure |
|---|---|---|---|---|
| KAIST | ETRI | 87.15 | 82.77 | 84.90 |
| ETRI | KAIST | 99.57 | 99.59 | 99.58 |

이 실험은 k는 2, 가중치는 거리에 반비례(ID)하게 부여하였다. ETRI 말뭉치가 KAIST 말뭉치에 비해 훨씬 복잡하여 문장 경계를 인식하는데 보다 어렵다는 것을 알 수 있다.

마지막으로 KAIST 말뭉치와 ETRI 말뭉치를 합쳐서 10-fold Cross Validation방법으로 실험하였으며 그 결과는 각각 표 7과 표 8과 같다.

표 7. 자질의 가중치

| 자질 종류 | 가중치 |
|---|---|
| 자질 1 | 0.071 |
| 자질 2 | 0.100 |
| 자질 3 | 0.007 |
| **자질 5** | **0.537** |
| 자질 6 | 0.070 |

표 8. 실험 결과

| 문장 정확률 | 문장 재현율 | F-measure |
|---|---|---|
| 98.82 | 99.09 | 98.95 |

KAIST 코퍼스와 ETRI 코퍼스를 통합하여 학습한 결과, 표 8에서 보인바와 같이 기존의 ETRI 말뭉치만을 사용하였을 때보다 성능이 많이 향상되는 것을 확인할 수 있었다. 자질의 가중치는 표 7과 같이 원래 ETRI 말뭉치만으로 얻은 결과와 유사하였다.

## V. 결 론

본 논문에서는 기계 학습을 이용한 범용의 한국어 문장 경계를 인식 방법을 제안하였다. 기계 학습 방법으로는 입력값과 유사한 k개의 데이터를 이용하는 kNN 알고리즘을 사용하였고, 문장 경계 인식을 위한 제안된 방법은 적용의 범위가 넓고 특정 영역에 종속적이지 않는 방법 개발을 위하여 특정 언어 또는 장르에 독립적인 자질 정보를 사용하였다. ETRI 코퍼스와 KAIST 코퍼스를 이용한 실험 결과 학습 코퍼스의 종류에 상관없이 높은 재현율과 정확률을 보였으며, 적절한 자질을 선택하는 것이 중요함을 보였다.

실험에 사용한 코퍼스는 ETRI와 KAIST에서 제공한 코퍼스를 원시 코퍼스로 변형하여 사용하였으며, 실험 결과 제안한 시스템은 적은 양의 학습 말뭉치만으로도 98.82%의 문장 정확률과 99.09%의 문장 재현율을 보였다.

본 논문에서 사용된 자질은 사전이나 품사 정보를 사용하지 않기 때문에 언어나 장르에 독립적이므로, 다른 말뭉치에 대해서도 적용이 가능하다. 단, 한 문장 내에 후보 구두점이 많은 것보다는 적은 것에 대해 성능이 더 좋게 나타남을 알 수 있었다. 향후 연구로는 다양한 기계 학습 기법을 적용하고, 보다 다양한 장르에서 실험해 보고자 한다. 또한 구두점이 생략된 경우 등의 문제도 해결할 수 있는 새로운 자질들도 고려하고자 한다.

## 참 고 문 헌


[1] Jeffrey C. Reynar and Adwait Ratnaparkhi, "A Maximum Entropy Approach to Identifying Sentence Boundaries," In Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 16-19, 1997.

[2] Tom M. Mitchell, Machine Learning, 1997.

[3] G. Grefenstette and P. Tapanainen, "What is a word, what is a sentence? problems of tokenization," In Proceedings of the 3rd International Conference on Computational Lexicography, pp. 79-87, 1994.

[4] Riley and Michael, D. "Some Applications of Tree-based Modeling to Speech and Language Indexing," In Proceedings of the DARPA speech and natural language workshop, pp. 339-352, 1989.

[5] David D. Palmer and Marti A. Hearst, "Adaptive Multilingual Sentence Boundary Disambiguation," Computational Linguistics, 23(2), pp. 241-267, 1997.

[6] Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch, *TiMBL: Tilburg Memory Based Learner, version 4.3, Reference Guide.* ILK Technical Report 02-10, 2002.

[7] 임희석, "경험적 정보를 이용한 kNN 기반 한국어 문서 분류기의 개선", 컴퓨터교육학회 논문지, 제5권 제3호, pp. 37-44, 2002.

[8] http://aladin.etri.re.kr/~nlu/STANDARD

[9] http://kibs.kaist.ac.kr/beginner/begin.htm.

## 저 자 소 개


**임 희 석(Heui-Seok Lim)** 정회원

- 1992년 2월 : 고려대학교 컴퓨터학과(이학사)
- 1994년 2월 : 고려대학교 컴퓨터학과(이학석사)
- 1997년 8월 : 고려대학교 컴퓨터학과(이학박사)
- 1997년 9월~1999. 2월 : 삼성종합기술원 전문연구원
- 1999년 3월~2004년 2월 : 천안대학교 정보통신학부 교수
- 2004년 3월~현재 : 한신대학교 소프트웨어학과 교수

<관심분야> : 자연어처리, 정보 검색, 인지신경언어

**한 군 희(Kun-Hee Han)** 종신회원

- 1989년 2월 : 충북대학교 컴퓨터공학과(공학사)
- 1994년 8월 :경남대학교 컴퓨터공학(공학석사)
- 2000년 8월 :충북대학교 컴퓨터공학과(공학박사)
- 1989년 1월~1994년 12월 : 대우정보시스템 연구원
- 1995년 3월~2000년 12월 : 대천대학 전기전자컴퓨터학부 교수
- 2001년 3월~현재 : 천안대학교 정보통신학부 교수

<관심분야> : 영상처리, 멀티미디어데이터베이스, 웹시스템개발