
스마트폰 상에서의 개인화 학습을 위한 랭크부스트 알고리즘

강대기* · 장원태**

RankBoost Algorithm for Personalized Education of Chinese Characters on Smartphone

Dae-Ki Kang* · Won-Tae Chang**

본 연구는 2009년도 동서대학교 학술연구조성비 지원과제에 의해 지원되었습니다.

요 약

본 논문에서는 스마트폰 상에서 한자 학습 시스템을 랭크부스트 알고리즘을 이용하여 개인화하는 방법에 대해 논하고자 한다. 한자 학습의 일반적인 환경을 보면, 학습자는 급수에 따라 일정한 개수의 학습할 한자들이 있으며, 학습이 진행됨에 따라 그 한자들 중 자신이 잘 틀리는, 즉 자신에게는 난이도가 높은 한자들이 생기게 된다. 본 논문에서는 이러한 난이도의 측정을 랭크부스트 알고리즘을 통해 구현하였다. 알고리즘은 초기에는 모든 한자들에 대해 동일한 가중치를 가지고 학습을 시작하게 하지만, 사용자가 자주 틀리는 한자에 대해서는 가중치를 높여 나간다. 본 논문에서 제안하는 랭크부스트 알고리즘은 학습자에 개인화된 난이도 순위를 매겨줌으로, 학습자가 어려운 한자에 더 자주 노출되게 한다면 학습 효과를 높일 수 있다.

ABSTRACT

In this paper, we propose a personalized Chinese character education system using RankBoost algorithm on a smartphone. In a typical Chinese character education scenario, a trainee is supplied with a finite number of Chinese characters as an input set in the beginning. And, as the training session repeats, the trainee will notice her/his difficult characters in the set which she/he hardly answers. Those characters reflect their personalized degrees of difficulty. Our proposed system constructs these personalized degrees of difficulty using RankBoost algorithm. In the beginning, the algorithm start with the set of Chinese characters, of which each is associated with the same weight values. As the training sessions are repeated, the algorithm increase the weights of Chinese characters that the trainee mistakes, thereby eventually constructs the personalized difficulty degrees of Chinese characters. The proposed algorithm maximizes the educational effects by having the trainee exposed to difficult characters more than easy ones.

키워드

안드로이드, 스마트폰, 랭크부스트, 학습 알고리즘

Key word

Android, Smartphone, RankBoost, Learning Algorithms

* 동서대학교 컴퓨터정보공학부(주저자)
** 동서대학교 컴퓨터정보공학부(교신저자)

접수일자 : 2009. 07. 27
심사완료일자 : 2009. 09. 21

I. 서 론

한자 학습의 일반적인 환경을 보면, 학습자는 급수에 따라 일정한 개수의 학습할 한자들이 있으며, 학습이 진행됨에 따라 그 한자들 중 자신이 잘 틀리는, 즉 자신에게는 난이도가 높은 한자들이 생기게 된다. 이러한 난이도가 높은 한자들의 집합은 학습자 개인에 따라서 서로 다르게 나타난다. 그러나 대부분의 한자 학습 시스템들은 특정 학습자의 이력을 추적하여 그 학습자가 잘 틀리는 한자들에 대해 중점적으로 문제를 냄으로써 학습자의 실력 향상에 도움을 주는 기능들을 가지고 있지 않다. 또한 일반 PC의 경우, 여러 사람이 사용하는 경우도 많으므로 개인화 [1] 기능이 효과적으로 활용되지 않는 경우가 많다. 반면, 스마트폰[2]의 경우, 공간에서 대량 생산되지만 대부분의 경우 한 사람의 소비자에 의해서만 사용되므로, PC와 비교해 볼 때 개인화의 효과가 더 크다고 볼 수 있다. 이러한 배경에 근거하여, 본 논문에서는 스마트폰 상에서 개인화된 한자 학습 시스템을 랭크부스트 알고리즘[3]을 사용하여 설계하는 방법에 대해 논하고자 한다.

본 논문에서는 학습자 개개의 난이도의 측정을 랭크부스트 알고리즘을 통해 구현하였다. 알고리즘은 초기에는 모든 한자들에 대해 동일한 가중치를 가지고 학습을 시작하게 하지만, 사용자가 자주 틀리는 한자에 대해서는 가중치를 높여 나간다. 본 논문에서 제안하는 한자 학습을 위해 랭크부스트 알고리즘을 적용한 스마트폰 응용 시스템은 학습자에 개인화된 난이도 순위를 매겨줌으로, 학습자가 어려운 한자에 더 자주 노출되게 한다면 학습 효과를 높일 수 있다.

본 논문에서 제안하는 시스템은 안드로이드 운영체제[4] 상에서 구현되었으며, 사단법인 한자교육진흥회의 국가 공인 한자 급수 자격 시험[5] 중 5급 한자의 250자를 가지고, 4명의 초등학교 2학년 실험자들에 대해 실험을 수행하였다.

본 논문의 구성은 다음과 같다. 2장에서는 선행 연구들에 대한 소개를 한다. 3장에서는 본 논문에서 제안한 시스템을 설명한다. 마지막으로, 4장에서는 결론을 내리고 전체를 요약하도록 한다.

II. 선행 연구 (Previous Researches)

2.1 안드로이드 (Android)

안드로이드란 스마트폰과 같은 모바일 시스템을 위한 공개 운영체제 및 미들웨어와 응용 프로그램들의 집합으로 구글(Google)에서 만들었다. 구글에서는 안드로이드에서 응용 프로그램을 개발하기 위한 Software Development Kit (SDK) 를 제공하고 있으며, 이 SDK는 사용자가 응용 프로그램을 자바 언어로 개발할 수 있게 한다. 그림 1은 안드로이드 운영체제에 에뮬레이터에서 수행되는 모습을 나타내고 있다.

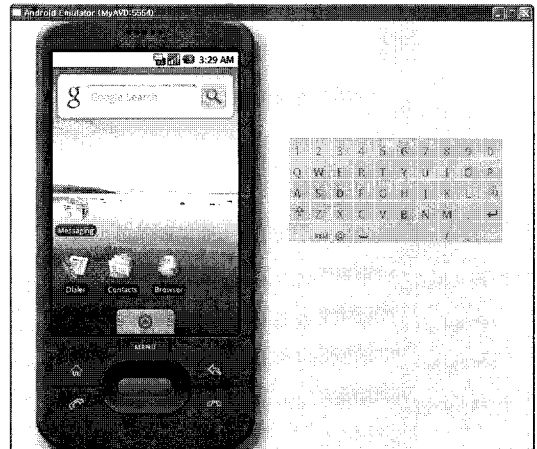


그림 1. 안드로이드 스마트폰 에뮬레이터
Fig. 1 Android Smartphone Emulator

2.2 개인화된 교육

현재, 여러 종류의 한자 학습 방식들이 널리 쓰이고 있다. 학습자의 과거 학습 경험(learning experience)이나 수업 집중도, 학습자의 수준에 적응적인 테스트의 관점에서, 이러한 한자 학습 방식들을 조망해 보면 다음과 같다.

- 학원이나 학교를 통한 한자 학습 - 강사를 상대로 다수의 학생이 배우기 때문에 현실감이 있고 수업의 집중이나 테스트에 있어 강사의 역할이 중요하며, 개인화된 학습이 어렵다.
- 교재나 책을 통한 한자 학습 - 학원을 통한 학습보다 더 융통성이 있지만, 수업 집중과 학습 효과, 태

스트의 개인화에 있어 학습자의 역량이 가장 중요하다.

- 컴퓨터 프로그램을 통한 온라인 한자 학습 - 근본적으로 교재를 통한 학습과 비슷하나, 학습 프로그램의 지능화 수준에 따라 수업 집중도와 테스트 기능의 개인화에 가능성이 높다. 특히 컴퓨터 프로그램이 인터넷에 연결되거나, 인터넷을 통한 온라인 학습의 경우, 학습 자료를 쉽게 갱신하거나 학습자의 개인화 정보를 저장하여 향후의 학습에 적용할 수 있다.

이러한 경향들을 배경으로 본 논문에서 관심을 가지는, 온라인 한자 학습 프로그램들이 매우 많으며, 특히 스마트폰 상에서는 아래의 그림 2와 같이 윈도우즈 모바일(Windows Mobile)의 Dio漢Dic이나 아이폰(iPhone)의 도전! 한자 2급 등의 뛰어난 상업용 학습 프로그램들이 있다.



그림 2. 상업용 한자 학습 시스템들
Fig. 2. Commercial Hanja Education Systems

이러한 학습 프로그램들은 체계적으로 구성된 학습 메뉴들과 풍부한 콘텐츠를 가지고 있다. 그러나 학습자의 개인적인 학습 경험(learning experience)을 반영하는 개인화된 학습 기능은 제공하지 않고 있다.

2.3 랭크부스트 알고리즘 (RankBoost Algorithm)

랭크부스트 알고리즘[3]을 설명하기 위해 우선 부스팅 알고리즘을 설명해야 한다. 이를 위해 대표적인 부스팅 알고리즘인 AdaBoost 알고리즘[6]을 소개하도

록 하겠다. AdaBoost는 (Adaptive Boost)의 약자로, 어떠한 학습 알고리즘에도 적용할 수 있는 메타 알고리즘이다. AdaBoost 알고리즘의 핵심은 여러 개의 분류기(classifier)를 차례로 생성하되, 나중에 생성되는 분류기를 위한 학습 데이터의 확률 분포를, 그 전에 생성된 분류기의 학습 데이터의 확률 분포에서 분류기의 성능 결과를 반영한 값으로 설정해 주는 것이다.

AdaBoost 알고리즘의 수행을 형식적으로 도시하면, 표 1과 같다.

표 1. AdaBoost 알고리즘
Table 1. AdaBoost Algorithm

<p>RankBoost 알고리즘: begin</p> <ol style="list-style-type: none"> 1. 입력 : 데이터 집합 $(X_1, Y_1), \dots, (X_m, Y_m)$, 여기서 $Y \in \{-1, 1\}$ 2. 데이터의 확률분포 $D_1(i) = \frac{1}{m}$으로 초기화함 3. 다음을 $t = 1, \dots, T$까지 반복함. <ul style="list-style-type: none"> 가. 분포 D_t에 근거하여 에러를 최소화하는 분류기 $h_t : X \rightarrow \{-1, 1\}$ 생성 나. 분류기의 에러 $\epsilon_t = \sum_{i=1}^m D_t(i) [y_i \neq h_t(x_i)]$를 구함 다. 만일 ϵ_t가 0.5보다 크다면 루프를 멈추거나 $D_t(i) = \frac{1}{m}$으로 초기화 라. 분포 갱신을 위해 $\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$ 계산 마. 정규화 계수 $Z_t = \sum_{i=1}^m D_t(i) e^{-\alpha_t y_i h_t(x_i)}$ 계산 바. $D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t}$로 분포 갱신 4. 출력 : $H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$ <p>end</p>
--

전체 데이터의 각각의 샘플들은 처음에는 동일한 확률 분포를 가지지만, 분류기가 제대로 분류하지 못하는 샘플들에 대해 단계적으로 확률 분포가 커지게 된다. 최

중적인 결과는 이러한 분류기들의 분류 결과의 평균으로 정해진다.

랭크부스트 알고리즘은 AdaBoost 알고리즘의 단순한 변형으로, 표 1의 알고리즘에서 최종적으로

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$$

$$\text{Score}(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

로 설정하여 각각의 샘플 x_i 에 대해 $\text{Score}(x_i)$ 를 구한 뒤, 내림차 순으로 정렬하는 것을 말한다. 본 논문에서 제안한 시스템은 위에서는 기계적으로 생성된 분류기를 인간인 학습자의 학습 성과로 대체한 것이다.

III. 시스템 (System)

그림 3은 제안된 시스템의 구성도이다. 스마트폰 내에 한자 가중치 데이터베이스, GUI 모듈, 그리고 문제를 생성을 담당하는 문제 출제 모듈이 있으며, 문제 출제 모듈은 변형된 랭크부스트 모듈을 통해 학습자에게 제시할 가장 적합한 문제를 생성한다.

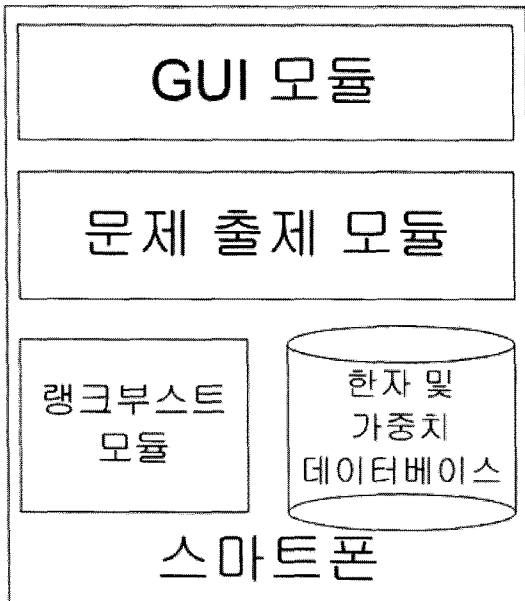


그림 3. 제안된 시스템의 전체 구성도
Fig 3. Structural Diagram of the System

학습자에게 제시할 가장 적합한 한자를 선정하기 위해 본 논문에서 제안하는 변형된 랭크부스트 알고리즘은 표 2에서 제시되고 있다.

표 2. 변형된 랭크부스트 알고리즘을 통한 학습 과정
Table 2. Learning Process by Modified RankBoost Algorithm

RankBoost 알고리즘을 통한 학습 과정:
begin

1. 입력 : 학습할 한자의 집합 X_1, \dots, X_m
2. 한자들의 확률분포 $D_1(i) = \frac{1}{m}$ 으로 초기화함
3. 다음을 $t = 1, \dots, T$ 까지 반복함.
 - 가. 분포 D_t 를 내림차 순으로 정렬하여 상위 $\frac{l}{2}$ 개의 분포값에 해당하는 한자들을 선정
 - 나. 그 외의 $m - \frac{l}{2}$ 개의 한자들 중 동일한 확률분포 $\frac{2}{2 \times m - l}$ 에 의해 $\frac{l}{2}$ 개의 한자들 선정
- 다. (가)와 (나)의 선정된 한자들을 합쳐 l 개의 한자들을 인간 학습자에게 학습을 시킴 ($l \leq m$)
- 라. 학습의 결과로 Y_i^t 생성, 여기서 i 는 (가) 단계에서 선택된 한자의 번호이며, $Y_i^t \in \{-1, 1\}$ 로 -1 은 학습자가 틀렸음을 의미하고 1 은 맞았음을 의미함
- 마. 학습자의 에러 $\epsilon_t = \sum_{i=\text{선택된 한자의 번호}}^{l \text{개의 선택된 한자}} D_t(i) [Y_i^t = -1]$ 를 구함 ($Y_i^t = -1$ 은 학습자가 틀렸음을 의미)
- 바. 분포 갱신을 위해 $\alpha_t = \frac{1}{2} \ln \frac{1 - \epsilon_t}{\epsilon_t}$ 계산
- 사. 정규화 계수 $Z_t = \sum_{i=1}^m D_t(i) e^{-\alpha_t Y_i^t}$ 계산하되, 선택되지 않은 한자의 경우, $Y_i^t = 1$ 로 간주함 ($Y_i^t = 1$ 은 학습자가 맞았음을 의미)
- 아. $D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t Y_i^t}}{Z_t}$ 로 분포 갱신
- 자. 출력: 점수 $(1 - \epsilon_t) \times 100$

end

처음에는 알고리즘은 동일한 확률 분포로 초기화된 한자들을 입력받는다. 전체 한자들의 개수가 m 이라면, 한번의 시험에서 학습자가 풀게 될 한자의 개수는 l ($l \leq m$)이다. 각 한자들마다의 확률 분포 D_i 에 근거하여 무작위로 $l/2$ 개의 한자를 선정하고, 선정되지 않은 그 외의 한자들에 대해서는 동일한 확률 분포를 가정하고 무작위로 $l/2$ 개의 한자를 선정하여, 총 l 개의 한자들을 학습자에게 학습시킨다. 즉, 절반은 부스팅된 확률 분포에 근거해서 선정된, 학습자가 그동안 잘 틀린 한자들이며, 나머지 절반은 무작위로 선정된 한자들이다. 물론 이 비율 γ 은 학습 패러미터에 따라 조정 가능하다. 표 2에서는 $\gamma = \frac{1}{2}$ 인 경우에 대해 알고리즘을 기술했다.

그림 4는 본 논문에서 제시한 설계 및 알고리즘을 통해 안드로이드 상에서 구현된 한자 학습 시스템의 실행 화면이다.



그림 4. 안드로이드 상의 한자 학습 프로그램
Fig. 4. Hanja Learning Program on Android

본 논문에서 제안하는 시스템의 효과를 검증하기 위해, 사단법인 한자교육진흥회의 국가 공인 한자 급수 자격 시험 중 5급 한자의 250자를 가지고, 4 명의 초등학교 2학년생 실험자들에 대해 학습 실험을 수행하였다.

실험 설정은 다음과 같다. 각각의 실험자를 위해 250자 중 100 자의 한자를 무작위로 선정한다. 선정된 100 자의 한자들 중, 50자는 대조군 설정을 위해 무작위로 문제를 제시하는 프로그램(Baseline)을 사용하여 공부하고, 다른 50자는 랭크부스트(RankBoost)를 사용하는 프로그램으로 공부한다($m=50$). 각 시험마다 30 개의 한자가 제시되고($l=30$), 시험은 20차례 반복되었다($T=20$).

그림 5는 각각의 실험 설정에 대해 학생들이 얻은 점수의 평균을 도시한 것이다.

그림 5에서 알 수 있듯이, 랭크부스트를 통해 틀린 문제를 집중적으로 학습한 학생들은, 단순히 무작위로 추출된 한자들을 가지고 학습한 학생들과 비교해 볼 때, 당연히 더 빠른 학습 속도를 보였다. 이러한 랭크부스트 방법은 기계 학습에서는 널리 쓰이고 있으며, 수학적으로도 형식적으로 잘 정립되어 있으나, 교육용 소프트웨어, 특히 개인화가 중요한 스마트폰 상의 교육용 소프트웨어에서는 그리 언급되지 않음으로 본 논문에서 제안하고자 하는 것이다.

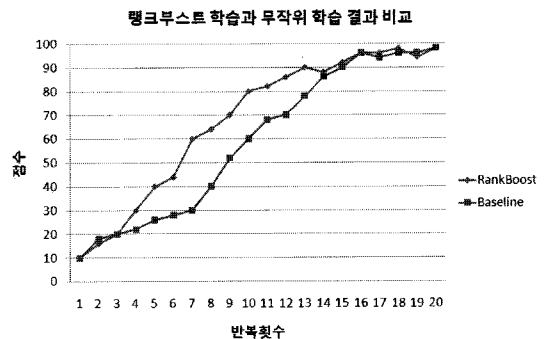


그림 5. 랭크부스트 한자 학습과 무작위 선정 한자 학습 속도 비교
Fig. 5. Comparison of Learning Speed Chinese Characters Selected By RankBoost and Random Pick

제안된 시스템의 전체 설정에 대해 정리하면 다음과 같다. 본 논문에서 제안하는 시스템은 구글의 안드로이드 운영체제 상에서 자바 언어로 구현되었으며, 스마트폰의 GUI 프로그램 개발을 위해서 구글 안드로이드 SDK 1.5를 사용하였다. 사용된 한자는 사단법인 한자교육진흥회의 국가 공인 한자 급수 자격 시험 중 5급 한자의 250자이며, 4명의 초등학교 2학년 실험자들에 대해

실험을 수행하였다.

IV. 결론 (Conclusion)

4.1 요약 (Summary)

본 논문에서는 스마트폰 상에서 한자 학습 시스템을 랭크부스트 알고리즘을 학습자를 대상으로 역이용하여 개인화하는 방법에 대해 논하였다. 이러한 개인화 시스템을 스마트폰에 적용하는 경우, 개인이 사용하는 스마트폰의 특성과 잘 들어맞으므로, 안드로이드 운영체제 상에서 시스템을 구현하였다.

본 연구에서는 제안된 알고리즘의 평가를 위해 4명의 초등학생을 대상으로 5급 한자 데이터를 사용하였다. 반복적인 학습 실험을 통해 다소간 의미있는 결과를 도출하였으나, 통계적으로 더욱 의미있는 결과를 내기 위해서는 보다 확장된 실험이 필요할 것으로 여겨진다. 이에 대해서는 향후 연구에서 더 자세히 서술하였다.

4.2 향후 연구 (Future Researches)

본 연구에서 제안한 시스템은 향후 다음 부분들에 대한 개선과 심화 연구를 계획하고 있다.

- 실험군 및 실험 데이터의 확장 - 실험 대상의 숫자를 더 늘리고 고학년 및 저학년 학생으로 계층화한 후, 다양한 급수의 한자 데이터를 적용하여, 통계적으로 더욱 의미가 있는 결과를 도출하도록 할 예정이다.
- 기존 연구 중 학습 효과를 높여주는 방법들과 제안한 방법의 정량적 비교 평가 - 기존 연구들을 심층적으로 살펴보고, 제안된 방법과 정량적인 비교를 통해 제안한 방법의 우수성을 평가하고자 한다.
- 필기 인식 시스템과의 연동 - 한자 급수 시험은 단순히 음훈 읽기만 테스트하는 것이 아니라, 쓰는 법도 테스트한다. 따라서 향후의 연구는 학습자가 쓴 한자를 인식하는 법에 대한 연구를 포함할 것이다. 이는 한자를 인식하여 검색하는 한자 필기 검색 엔진으로의 응용도 가능하다.
- 문맥 상의 쓰임에 대한 학습 방안 - 한자 급수 시험은 단순히 음훈 읽기만 테스트하는 것이 아니라, 문장 내에서 그 한자가 어떻게 쓰이는지에 대한 단어로

의 쓰임도 테스트한다. 따라서, 이 부분에 대한 연구도 진행할 예정이다.

- 다른 외국어 학습 - 제안된 시스템은 단순히 한자 뿐만 아니라 영단어 학습 프로그램의 경우에도 활용 가능성이 높으리라 여겨진다.

감사의 글

본 연구는 2009년도 동서대학교 학술연구조성비 지원과제에 의해 이루어졌습니다. 저자들은 논문을 자세히 심사하고 리뷰해 주신 심사위원님들께 감사드립니다.

참고문헌

- [1] A. Das, M. Datar, A. Garg, and S. Rajaram, "Google News Personalization: Scalable Online Collaborative Filtering," In the proceedings of 16th International World Wide Web Conference (WWW 2007), May 8 - 12, 2007, Banff, Alberta, Canada.
- [2] R. Ballagas, J. Borchers, M. Rohs, and J. G. Sheridan, "The Smart Phone: A Ubiquitous Input Device," IEEE Pervasive Computing, vol. 5, no. 1, pp. 70-77, January-March, 2006.
- [3] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. "An efficient boosting algorithm for combining preferences," Journal of Machine Learning Research, 4:933-969, Nov. 2003.
- [4] J. Agüero, M. Rebollo, C. Carrascosa1 and V. Julián1, "Does Android Dream with Intelligent Agents?," International Symposium on Distributed Computing and Artificial Intelligence, pp. 194-204, 2008.
- [5] 사단법인 한자교육진흥회, 국가 공인 한자 급수 자격 시험, <http://www.hanja114.org/>.
- [6] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, no. 55. 1997.

저자소개



강대기 (Kang, Dae-Ki)

1992년 한양대학교 전자계산학과
졸업 (공학사)

1994년 서강대학교 전자계산학과
졸업 (이학석사)

1994년~1999년 : 한국전자통신연구원 (연구원)

2006년 : Iowa State University Dept. of Computer Science
졸업 (PhD in Computer Science)

2007년 2월~2007년 8월 : 국가보안기술연구소
(선임연구원)

2007년 9월~현재 : 동서대학교 컴퓨터정보공학부
전임강사

※ 관심분야 : 기계학습, 관계학습, 통계적그래피컬모
델, 스마트폰, 온톨로지학습, 침입탐지, 웹방화벽,
웹마이닝, 컴퓨터비전



장원태 (Won-Tae Chang)

1989년 : 성균관대학교 전자공학과
(공학사)

1996년 : 서울시립대학교 제어계측
공학과 (공학석사)

1989년 ~ 2001 : Korea Telecom Authority International

2002년 ~ 현재 : 동서대학교 컴퓨터정보공학부 교수

※ 관심분야 : 모바일S/W, RFID, Smart Phone, 컴퓨터
응용