
게임에서의 지능적 NPC 구현을 위한 자연어 대화 처리 기법

우영운* · 박성대** · 박충식***

A Natural Language Conversation Method for Intelligent NPC Implementation in Games

Young Woon Woo* · Sungdae Park** · Choong-Shik Park***

이 논문은 2006년도 동의대학교 연구비를 지원받았음(연구과제번호 : 2006AA183)

요 약

최근에 인공지능 기법을 적용한 자연어 처리 프로그램을 개발하기 위한 연구가 많이 진행되고 있으나 아직까지는 자연어 형태소 분석 등에 대부분 많은 노력을 기울이고 있으며 형태소 분석 결과를 활용하기 위한 기법에 대한 연구가 부족한 실정이다. 본 논문에서는 자연어의 형태소 분석 결과와 규칙 추론 기법을 활용하여 게임에서 사용되는 NPC(Non-Player Character)가 사용자와 자연어 문장으로 대화를 가능하게 하는 자연어 대화 프로그램을 개발하였다. 이를 위하여 기존에 개발되어 있는 규칙 추론 엔진인 NEO를 이용하여 자연어 대화 처리에 적합한 규칙의 표현과 구현 기법을 제안하였다. 실험을 위하여 다이어트에 대한 상담을 해 주는 NPC를 가상으로 설정하여 다이어트에 관련된 지식을 규칙과 사실들로 생성하였으며 다이어트와 관련된 보편적인 문장들로 프로그램을 수행한 결과 자연스러운 대화 내용이 생성됨을 알 수 있었다.

ABSTRACT

Recently, there are many researches about natural language processing programs using artificial intelligence methods. But the researches mostly concentrate on Korean morphological analyses and there are few researches about application of the results of Korean morphological analyses. In this paper, we implemented a natural language conversation program that NPC in games can talk with users by natural language sentences using the results of morphological analyses and a rule-based inference method. We proposed representation and implementation methods of rules suitable for the processing of natural language conversation using NEO, a rule-based inference engine. In the experiment using rules and facts about knowledge of conversation for diet counselor NPC, we could verify that natural conversation results were produced.

키워드

자연어 대화, 규칙 기반 기법, 지능적 NPC
Natural language conversation, Rule-based method, Intelligent NPC

* 동의대학교 멀티미디어공학과

** 동의대학교 영상정보대학원

*** 영동대학교 컴퓨터공학과

I. 서론

현재 게임에서 사용되는 NPC는 대부분 미리 정해진 구조의 메뉴 방식에 의해 사용자와의 인터랙션(interaction)을 구현하고 있다. 이러한 방식의 NPC는 항상 동일한 방식에 의해 사용자와 인터랙션을 수행함으로써 게임에 하는 사용자가 따분하고 지루한 느낌을 쉽게 가지기 마련이다. 하지만 게임에서의 NPC가 사용자와 자연어로 대화를 할 수 있다면 상황에 따라 변화하는 내용으로 대화를 수행함으로써 매번 다른 내용의 이야기가 전개되기도 하고 이야기의 전개 순서도 매번 달라질 수 있어 사용자가 NPC를 대할 때마다 매번 새로운 기대를 가지게 되어 게임에 대한 흥미도가 더욱 높아질 수 있다.

최근에 인공지능 기법을 적용한 자연어 처리 프로그램을 개발하기 위한 연구가 많이 진행되고 있으나 아직까지는 자연어 형태소 추출 등의 자연어 분석에 대부분 많은 노력을 기울이고 있으며 형태소 분석 결과를 활용한 규칙 기반의 처리 기법에 대한 연구는 미비한 실정이다[1][2]. 따라서 인간에게 가장 친숙하고 편리하게 사용될 수 있는 자연어를 이용한 명령어의 입력이나 자연어를 이용한 대화 기능에 관한 연구가 미비하여 자연어 대화 기능을 가지는 지능적 NPC를 구현한 게임의 실용화가 현재까지도 담보 상태를 벗어나지 못하고 있다.

본 논문에서는 이러한 문제를 해결하기 위하여 게임에서의 지능적 NPC 구현에 적용될 수 있도록 자연어의 형태소 분석 결과와 규칙 형태의 지식 표현 기법을 활용한 규칙 기반의 자연어 대화 프로그램을 개발하였다. 본 논문에서 구현한 규칙 기반의 자연어 대화 프로그램은 자연어 분석 모듈인 HAM(Hangul Analysis Module)[3]과 기존에 개발되어 있는 규칙 추론 엔진인 NEO[4]를 활용하여 게임에서의 지능적 NPC에 적합한 자연어 대화 및 자연어 명령어 처리를 위한 규칙의 표현 기법을 제안하고 지식의 규칙화 및 규칙의 활용 방안을 제안하였다.

개발된 프로그램을 이용하여 다이어트에 대한 조언을 주는 NPC를 가상으로 설정하여 다이어트에 대한 지식들을 규칙과 사실들로 구현한 후 사람과의 대화를 수행해 봄으로써 가능성과 실용성을 보였다.

II. 형태소 분석

본 논문에서는 자연어 문장의 형태소를 분석한 결과를 이용하여 규칙 기반의 자연어 대화 프로그램을 개발하였다. 자연어 문장의 형태소를 분석하기 위한 도구로는 국민대학교 강승식 교수팀에서 개발한 HAM 4.0 라이브러리를 이용하였다[3]. HAM 형태소 분석 뷰어 프로그램은 그림 1과 같다.

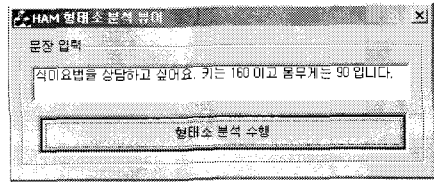


그림 1. HAM 형태소 분석 뷰어 프로그램
Fig. 1. HAM morpheme analysis viewer program

이상과 같이 자연어 문장을 분석하여 형태소를 파악하는 HAM 프로그램의 기능을 간략히 요약하면 다음과 같다.

- (1) 단어의 유형 인식
 - 명사 + 조사, 용언 + 어말어미, 용언 + 선어말어미 + 어말어미, 용언 + 명사형어미 + 서술격조사 + 선어말어미 + 어말어미 등
 - 모든 한국어 단어의 유형을 인식
- (2) 단어를 구성하고 있는 각 형태소의 기본형과 품사 인식
- (3) 복합명사 및 미등록어 처리 기능
- (4) 명사 사전 관리 기능
 - 불필요한 명사 삭제 기능
- (5) 사용자 정의사전 관리 기능
 - 미등록어 등록, 수정, 삭제 기능
- (6) 전문용어 사전 관리 기능
 - 전문용어 등록, 수정, 삭제 기능

(7) 기분석 사전에 의한 준말처리 기능

그림 1과 같이 형태소 분석 뷰어 프로그램에 자연어 문장을 입력하여 형태소를 분석한 결과는 그림 2와 같다.



그림 2. 자연어 문장에 대한 형태소 분석 결과
Fig. 2. Morpheme analysis results of natural language sentences

그림 2에서 알 수 있듯이 자연어 문장이 형태소 분석기에 입력되어 분석이 이루어지면 분리된 형태소와 그에 해당하는 의미의 태그(tag)가 붙어서 함께 출력된다. 본 논문에서 사용한 HAM 4.0에서는 다음과 같은 어휘 형태소 8 가지, 문법형태소 5 가지를 구별해 낸다.

(1) 어휘형태소

명사(N), 동사(V), 조동사(x), 부사(B), 한정사(D), 감탄사(X), 구두점(C), 특수기호(Q)

(2) 문법형태소

조사(j), 어말어미(e), 선어말어미(f), 체언접미사(s), 용언접미사(t)

본 논문에서는 규칙 기반의 자연어 대화 프로그램을 개발하기 위하여 이상과 같은 다양한 기능들 중 자연어 문장을 분석하여 추출되는 형태소들 중 일부를 활용하였다. 특히 다음과 같은 5 가지의 어휘형태소들이 규칙에서의 문장 패턴과 의미 파악을 위하여 사용되었다.

- 명사(N), 동사(V), 부사(B), 한정사(D), 구두점(C)

동사는 시제에 관계없이 동사 원형이 추출되므로 추출되는 동사 원형에 ‘다’를 붙여 규칙의 입력으로 적용되도록 함으로써 규칙 작성의 편의를 도모하였다. 또한 구두점(마침표(.), 쉼표(,), 의문부호(?), 감탄부호(!))을 활용하여 문장의 의미가 단순 서술인지 의문문인지 감탄문인지를 파악할 수 있도록 하였다.

III. 규칙 추론 엔진

자연어 대화를 구현하기 위하여 규칙을 사용할 경우에도 규칙 추론 엔진이 반드시 요구된다. 기존에 상업화되어 시판되고 있는 규칙 추론 프로그램들이 몇 가지 있으나 모두 고가이며 독립된 패키지 형태로 제공되고 있기 때문에 지능형 에이전트를 위해 기능 일부만을 커스터마이징(customizing)하여 활용할 수 없는 문제점이 있다 [5][6]. 또한 각 기능이 사용자에 의해 변경되거나 개선될 수 없는 구조이기 때문에 지능형 에이전트를 위해 기존의 패키지 도구를 활용할 수 없다.

따라서 본 논문에서는 저자들과 몇몇 동료들이 함께 개발한 NEO라는 규칙 추론 엔진을 이용하여 자연어 대화 기법을 제안하고 구현하였다. NEO는 LISP과 유사한 프로그래밍 기능을 갖고 있으며 규칙 기반의 추론 엔진을 함께 갖춘 도구이다. NEO에서 제공되는 대표적인 기능들을 요약하면 다음과 같다.

- (1) 리스트에 의한 규칙 정보의 처리 기능
- (2) 사실과 규칙의 저장 기능
- (3) 전향 추론과 후향 추론 기능
- (4) 패턴 매칭 기능
- (5) 조건 함수 사용
- (6) 사용자 정의 함수 사용
- (7) 대화식 GUI 개발 환경 제공

이상의 기능들이 제공되는 NEO 규칙 추론 엔진을 이용하여 자연어 대화를 위한 규칙을 구성하였다. NEO 개발 환경의 인터페이스는 그림 3과 같다.

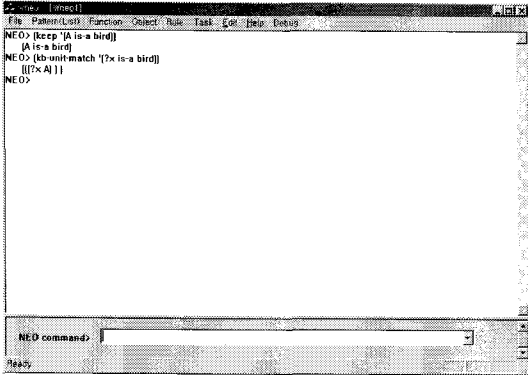


그림 3. NEO 개발 환경
 Fig. 3. NEO environment for development

응답이 중복되어 나오지 않게 된다.

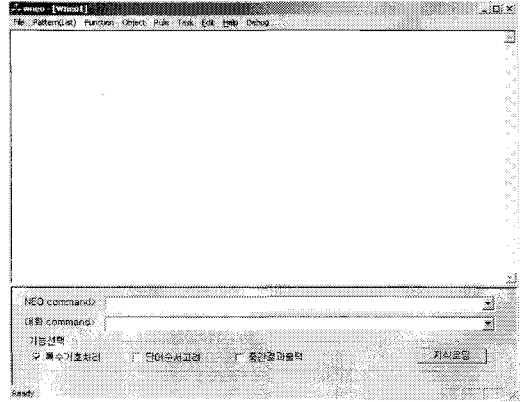


그림 4. 구현된 자연어 대화 프로그램 개발 환경
 Fig. 4. The Implemented natural Language conversation program for development

IV. 자연어 대화 프로그램 개발 환경

본 논문에서는 자연어 대화 프로그램을 개발하기 위하여 NEO 개발 환경에 형태소 분석 모듈을 결합하고 자연어 문장의 입, 출력이 가능한 기능을 보완하여 자연어 대화 프로그램 개발 환경을 구현하였다. 프로그램 개발 도구로는 Visual C/C++ 6.0을 이용하였다.

개발된 자연어 대화 프로그램에서는 형태소 분석 결과의 형태소 순서와 특수기호 고려 여부를 선택 사항으로 제공하여 대화 지식에서 어휘의 순서와 특수기호를 지식의 일부로 활용할 수 있도록 하였으며, 효과적인 디버깅을 위하여 중간 결과의 화면 출력 여부를 선택 사항으로 제공하였다. 또한 기존의 NEO 명령어를 이용하여 지식 베이스의 규칙과 사실, 그리고 사용자 정의 함수 등을 추가, 수정 삭제 할 수 있도록 NEO 명령어 입력창과 처리 모듈을 함께 제공하고 있다. 본 논문에서 구현한 자연어 대화 프로그램의 인터페이스는 그림 4와 같다.

기존에 개발되어 있던 NEO에서의 규칙 추론 메커니즘을 그대로 사용하게 되면 입력되는 사실들과 추론 결과로서 생성되는 새로운 사실들이 모두 저장되어 유지되는 구조로 동작되도록 되어 있다. 그러나 자연어 대화라는 특정 도메인을 위해서는 입력으로 들어오는 문장이 분석된 형태소들이 저장되어 그로 인한 추론이 이루어지고, 그 결과로서 생성되는 사실들이 한 가지의 대답을 마무리하고 나서는 저장되어 있던 형태소들과 추론 과정에서 생성된 사실들이 모두 삭제되어야만 다음 대화 문장 처리 시 앞 문장에 관련된 사실들로 인한 같은

본 논문에서는 이러한 문제점을 해결하기 위하여 기존의 NEO 규칙 추론 명령(foward)을 호출하기 전에 형태소 분석기에서 추출된 형태소들을 사실(facts)로 지식 베이스에 먼저 저장(그림 5의 (1), (1') 부분이 저장을 위한 루틴임)시킨 후에 추론을 수행하고 추론이 종료되고 나면 저장되었던 형태소들을 모두 삭제(그림 5의 (2))하고 reset-infer-native라는 명령을 마지막에 호출(그림 5의 (3))하도록 하는 자연어 대화 프로그램 개발 환경을 구현하였다.

본 논문에서 자연어 대화를 위하여 사용한 규칙의 형식은 다음과 같다. 이 규칙 구조는 NEO에서 기본적으로 제공하는 규칙 처리 메커니즘에 의한 구조이다[4].

$(rule-name1 \text{ if } (cond_1)(cond_2)...(cond_n))$
 $(rule-name1 \text{ then } (action_1)(action_2)...(action_n))$

if 절에 나타나는 조건(cond)들은 AND로 연결되는 조건들이며 then 절에 나타나는 행위(action)들은 실행될 수 있는 함수이거나 지식 베이스에 저장되는 사실 리스트 등이다. 조건과 행위를 구성하기 위한 리스트들은 NEO 문법을 따르며 사용자 정의 함수를 허용함으로써 다양한 기능을 지속적으로 추가할 수 있다.

```

void CCommandView::OnHamExecute()
// 형태소 분석 및 규칙 추론 함수
CString strHam, strTmp;
int i;

m_strToken.RemoveAll();
m_ctlHaminput.GetWindowText(strHam);
m_ctlHaminput.InsertString(0,strHam);
while (m_ctlHaminput.GetCount()>32)
m_ctlHaminput.DeleteString(32);
printf(">>> %s\n", strHam);
HamAnalysis(strHam); // 형태소 분석 모듈 호출

UpdateData(TRUE);
if(m_bWordOrder){ // 단어 순서를 고려하는 경우(1)
for(i=0; i < m_strToken.GetSize(); i++)
strTmp += m_strToken[i] + " ";
strTmp.TrimRight(' ');
if(m_bMidResult) // 중간 결과 출력
printf("%s\n", "(keep \"(" + strTmp + ")");
NonThreadCompuProc("(keep \"(" + strTmp + ")");
printf("NEO> ");
NonThreadCompuProc("(forward)"); //전향 추론 개시
printf("\n");
if(m_bMidResult) // 중간 결과 출력
printf("%s\n", "(forget \"(" + strTmp + ")");
NonThreadCompuProc("(forget \"(" + strTmp + ")");
// 추론 후 저장된 형태소 사실들을 모두 삭제 (2)
}
else{ // 단어 순서를 고려하지 않는 경우(1')
for(i=0; i < m_strToken.GetSize(); i++){
if(m_bMidResult)
printf("%s\n", "(keep \"(" + m_strToken[i] + ")");
NonThreadCompuProc
(" (keep \"(" + m_strToken[i] + ")");
}
printf("NEO> ");
NonThreadCompuProc("(forward)"); //전향 추론 개시
printf("\n");
for(i=0; i < m_strToken.GetSize(); i++){
if(m_bMidResult)
printf("%s\n", "(forget \"(" + m_strToken[i] + ")");
NonThreadCompuProc
(" (forget \"(" + m_strToken[i] + ")");
// 추론 후 저장된 형태소 사실들을 모두 삭제 (2)
}
}
NonThreadCompuProc("(reset-infer-native)");
// 추론 과정에서 발생한 임시 사실들을 모두 삭제 (3)
m_ctlHaminput.SetWindowText("");
}
    
```

그림 5. 형태소 분석모듈 호출 및 추론엔진 수행 함수

Fig. 5. The procedure for calling morpheme analysis module and executing inference engine

구현된 자연어 대화 프로그램에서는 단어순서의 고려 여부에 따라 다음 같은 2 가지 방식의 규칙 처리가 가능하다. 첫 번째는 단어 순서를 고려하지 않는 방식으로 다음과 같이 형태소 분석된 단어들에 개별적인 조건으로 입력되는 방식이다(그림 5의 (1')).

(r1 if (날말1)(날말2)...(날말n))

만약 낱말의 순서를 고려하여 규칙을 작성하게 되면 if 절은 다음과 같은 형태로 구성되며, 이 경우에는 입력된 전체 문장의 형태소 분석 결과가 하나의 리스트 형태로 되돌려 주도록 대화 프로그램이 구현되었다(그림 5의 (1)).

(r1 if (날말1 날말2... 날말n))

V. 실험 및 결과 고찰

본 논문에서는 낱말의 순서를 이용하지 않은 자연어 대화 규칙을 작성하였으며, 다이어트에 대한 조언을 주기 위한 지식을 규칙으로 구현하여 실험을 수행하였다. 실험을 위해 구축된 규칙의 개수는 81개이며 사실의 개수는 34개이다. 실험에 사용된 규칙과 지식의 일부는 다음과 같다.

```

/* 인사 */
(r-hi if (안녕)(NOI(이미인사))(NOT(이미인사2)))
(r-hi then(keep '(이미인사))(printf '(네, 안녕하세요! 다이어트 상담 전문 MRW입니다. 고객님의 무엇을 도와드릴까요? <'⊗>)))

(r-hiagain1 if (안녕)(이미인사)(NOT(이미인사2)))
(r-hiagain1 then (keep '(이미인사2))(forget '(이미인사))(printf '(네네, 안녕하세요! 고객님의 무엇을 도와드릴까요? <'⊗>)))
    
```

```

/* 하는일 */
(r-job if (무엇)(돕다)(주다)("??"))
(r-job then (printf '(저희 MRW에서는 음식 다이어트 종류 및 방법, 음식량조절, 운동방법, 신체질량지수체크 등에
    
```

관련된 상담해 드리고있습니다.)))

```
(r-job-1 if (?x)(어떤)(하다)(?"(equal ?x 일))
(r-job-1 then (printf '(저희 MRW에서는 음식 다이어트 종류 및 방법, 음식량 조절, 운동방법, 신체질량지수체크 등에 관련된 상담해 드리고있습니다.)))
```

/* 비용 */

```
(r-money if (?x)(?"(얼마)(equal ?x 상담)
(r-money then (printf '(네, 저희는 무료로 상담해 드리고 있습니다.)))
```

/* 상담시작 */

```
(r-consult if (?x)(다이어트)(싶다)(equal ?x 상담)
(r-consult then (printf '(다이어트 상담하고 싶으세요? 어떤 다이어트가 하고 싶으세요?)))
```

```
(r-weight if (?x)(빠다)(싶다)(equal ?x 살)
(r-weight then (printf '(살을 빼고 싶으세요? 어떤 부위의 살을 빼고 싶으신지 말씀해 주세요.)))
```

/* 뱃살 */

```
(r-abdomen if (?x)(빠다)(싶다)(equal ?x 뱃살)
(r-abdomen then (printf '(뱃살을 빼고 싶으세요? 윗배, 아랫배, 옆구리 이렇게 세부분으로 나눠서 운동을 할 수 있습니다. 어떤 부분을 빼고 싶으세요?)))
```

```
(r-abdomen-1 if (?x)(빠다)(어떻다)(되다)(?"(
(?y 부위 ?z)(equal ?x ?z)(equal ?x 뱃살)
(r-abdomen-1 then (printf '(\n %s) '(?y)))
```

/* 운동방법의 종류 */

```
(뱃살 종류 운동방법)
(허벅지살 종류 운동방법)
(종아리살 종류 운동방법)
(운동방법 조사 예는)
```

/* 허벅지살 운동방법 */

```
(간단한트레이닝 운동방법 허벅지살)
(스트레칭 운동방법 허벅지살)
(허벅지살 조사 예)
```

/* 음식 다이어트 종류 */

```
(덴마크다이어트 종류 다이어트)
(물다이어트 종류 다이어트)
(두부다이어트 종류 다이어트)
(다이어트 조사는)
```

이상과 같은 구조로 구현된 규칙과 사실들을 이용하여 다이어트와 관련된 자연어 대화를 실행한 결과 화면은 그림 6과 같다. 그림 6에서 ">>>" 프롬프트의 문장이 사용자가 입력한 문장이며 "NEO" 프롬프트의 문장이 컴퓨터가 생성하여 출력한 문장이다.

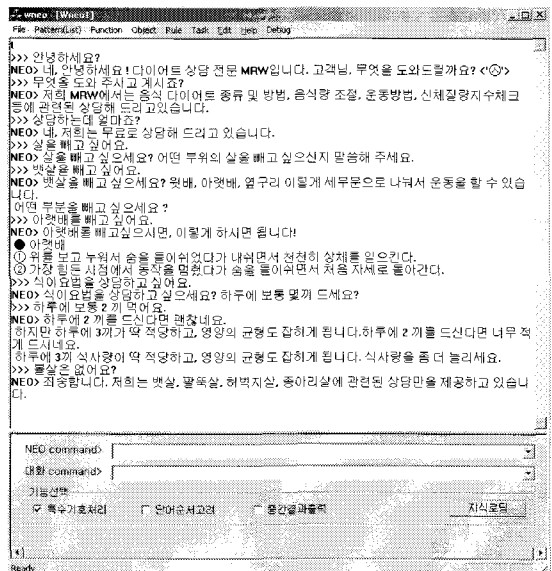


그림 6. 자연어 대화 실행 결과
Fig. 6. Results of natural language conversation

VI. 결 론

본 논문에서는 지능형 에이전트에서 활용 가능한 자연어 대화 프로그램을 개발하였다. 자연어 대화 프로그램 개발을 위해 자연어 형태소 분석 모듈인 HAM과 규칙 추론 기능을 갖는 NEO 개발 도구를 활용하였다. 자연어 형태소 분석 모듈을 통해 추출되는 형태소들 중 명사, 동사, 부사, 수사, 문장 부호의 5가지를 활용하였으며, 이 요소들을 이용한 규칙과 사실의 구현 방안을 제안하여

NEO 도구에서 전향 추론이 적용 가능하도록 하였다. 실험을 위하여 다이어트에 대한 조언을 주기 위한 규칙들과 사실들을 구축하여 실행해 본 결과 자연스러운 대화가 생성됨을 알 수 있었다.

본 논문에서 제안한 기법이 실용화되기 위한 향후 연구 과제로는 추출된 형태소의 순서나 추가적인 형태소들을 함께 활용할 수 있는 방법과 대화 범위에 알맞은 규칙들을 효과적이고 빠르게 생성할 수 있는 방법들이 추가되어야 할 것으로 생각한다.

참고문헌

- [1] 김도완, 박재득, 박동인, “자연어 대화 인터페이스를 이용한 정보검색(WWW)에 있어서 사용자 모델 에이전트,” 제8회 한글 및 한국어 정보처리 학술대회, pp.75-84, 1996
- [2] 김태훈, 최종민, “사용자 편의를 위한 지능형 인터넷 에이전트의 구현,” HCI '97 학술대회 발표 논문집, pp.261-266, 1997
- [3] <http://nlp.kookmin.ac.kr/>
- [4] 박충식, 우영운, 이민석, NEO Manual, 2005. [Not yet published]
- [5] <http://www.fairisaac.com/Fairisaac/Solutions/Enterprise+Decision+Management/Business+rules/Blaze+Advisor/>
- [6] <http://www.gensym.com/>

저자소개

우 영 운(Young Woon Woo)



1989년 2월 연세대학교 전자공학과 (공학사)

1991년 8월 연세대학교 본대학원 전자공학과(공학석사)

1997년 8월 연세대학교 본대학원 전자공학과(공학박사)

1997년 9월 ~ 현재 동의대학교 멀티미디어공학과 교수

2007년 ~ 현재: 한국해양정보통신학회 국제이사

※관심분야: 인공지능, 패턴인식, 퍼지이론, 의료정보

박 성 대(Sungdae Park)



2002년 2월 동의대학교 멀티미디어 공학(공학사)

2004년 2월 동의대학교 멀티미디어 공학(공학석사)

2008년 2월 동의대학교 컴퓨터공학(공학박사)

※관심분야: Digital Signal & Image Processing

박 충 식(Choong-Shik Park)



1985년 2월 한양대학교 전자공학과 (공학사)

1987년 8월 연세대학교 본대학원 전자공학과(공학석사)

1992년 8월 연세대학교 본대학원 전자공학과(공학박사)

1994년 3월 ~ 현재 영동대학교 컴퓨터공학과 교수

※관심분야: 인공지능, 지식기반시스템, 등