

# None Play Character 문장 자동 생성 시스템을 적용한 어드벤처 게임

김용범, 정호민, 송창근, 고영웅, 김유섭  
한림대학교 컴퓨터 공학과

e-mail : {stylemove, yskim01}@hallym.ac.kr

## An Adventure Game using Dynamic Sentence Generation for None Play Character

Young-Bum Kim, Ho Min Jeong, Chang Geun Song, Young Woong Ko, Yu-Seop Kim  
Department of Computer Engineering , Hallym University

### 요 약

본 논문에서는 NPC(None Play Character)가 어휘를 난이도에 따라 자동으로 확장하는 방법을 사용하여 동일한 의미의 다양한 문장을 동적으로 구술할 수 있도록 하는 시스템을 제안하였다. 이를 위해서는 NPC에게 주어진 문장에서 형태소 분석을 통해 키워드를 추출하고, 각 키워드에 대하여 워드넷의 계층적 특성에 따라 유사한 후보 단어를 제시한 후 게임에 활용할 수 있도록 XML 형식의 문서를 제작하였다. 이러한 방법을 통하여 NPC에게 부여할 어휘의 개발비용을 줄일 수 있으며 동적으로 어휘를 사용자에게 제공함으로써 게임 복잡도를 증가시켜 게임 난이도 조절이 가능하도록 하였다. 이와 더불어 이 게임은 다양한 유사도 측정 방법에 대한 성능 평가 모델로의 사용할 수 있다.

### 1. 서론

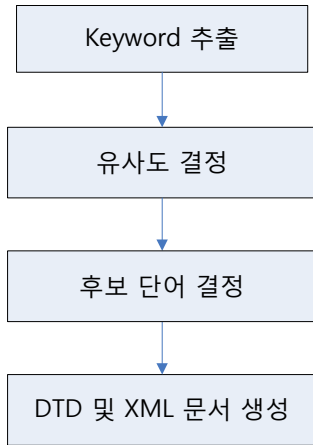
플레이어가 직접 조작하지 않는 순수 AI로만 움직이는 캐릭터를 언급하는 NPC(Non Play Character)[1]는 많은 게임들에서 주인공의 동료로 혹은 정보를 주는 도우미로써 등장하여 플레이어에게 도움을 주는 역할을 한다. 최근의 게임에서는 NPC의 역할이 다양하여 게임의 주어진 목표를 달성하기 위해서는 특정 NPC와의 대화에서 얻은 정보의 비중이 높아졌다. 하지만 사용자는 NPC가 구술할 수 있는 문장은 고정되어 있는 한계 때문에 종종 지루함을 호소한다. 이러한 문제를 해결하기 위해서는 다양한 어휘 목록을 구성하는 전문가 인력 및 시간적 비용이 요구된다. 본 논문에서는 워드넷[2]상에서 추정할 수 있는 다양한 어휘 유사도를 활용하여 어휘 목록을 자동으로 생성하여 게임에 적용할 수 있는 방법을 고안하

여 게임의 질을 향상시키는 것이 가능하도록 하였다.

### 2. NPC 어휘 구성

#### 2.1. Keyword 추출 단계

시나리오에 따라 작성된 문장을 NPC의 역할에 맞게 사상시킨 문장에 대하여 HAM(Hangul Analysis Module)[3]을 적용시켜 주요 Keyword를 추출한다. 예를 들어 “젊은 친구 시계 탑 9시 부근 무기 점 뒤 쪽으로 행운의 열쇠가 있을 걸세”라는 문장이 입력되면 {시계 탑, 9시, 부근, 무기, 점, 왕궁, 뒤쪽, 행운, 열쇠}라는 키워드가 추출된다.



(그림 1) NPC 어휘 구성단계

## 2.2. 유사도 결정 단계

이 단계는 게임의 난이도를 통하여 유사도를 결정하는 단계로 게임의 난이도 1 ~ N을 유사도 0 ~ N-1 로 사상하여 유사도를 결정한다. 예를 들어 게임 목표 달성을 위해 획득해야 할 키가 5개일 경우 유사도 도메인은 0 ~ 4 로 결정되며 현재 찾은 키의 개수가 1이면 난이도는 1로 결정이 된다.

## 2.3. 후보 단어 추출 단계

1 단계에서 추출한 키워드와 2 단계에서 결정된 유사도를 이용하여 아래의 측정방법 2가지를 이용한다. [4][5]에서의 어휘 개념  $c_1$ ,  $c_2$ 의 관계는 아래의 ①, ②와 같이 추정한다.

$$rel(c_1, c_2) = C - len(c_1, c_2) - k * d \quad ①$$

①식은 어휘 개념간의 길이와 패스의 방향 전환개수를 고려한 유사도 측정 방법으로 C와 k는 상수이고,  $len(c_1, c_2)$ 은 두 개념사이의 가장 짧은 간선의 개수, d는 패스의 방향 전환 회수를 의미한다. 이 방법은 두 개념 사이의 간선 개수와 패스의 방향전환이 적은 경우에 더욱 유사한 개념으로 정의한다.

$$sim(c_1, c_2) = -log \quad ②$$

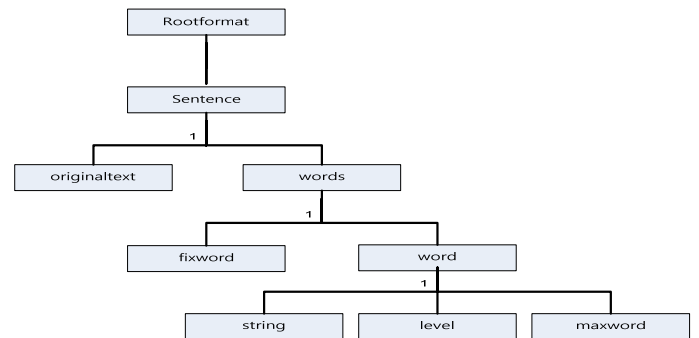
②식은 어휘 개념간의 간선의 개수와 어휘가 포함되어 있는 트리의 차수를 고려한 유사도 측정방법으로  $len(a, \beta)$ 은 개념 a 와 개념  $\beta$  사이의 간선의 개수를 의미하며, D는 WordNet에 존재하는 여러 개의

트리 가운데 두 개념이 존재하는 트리의 차수를 의미하며 다른 트리에 존재하는 두 개념은 유사도가 없음을 의미한다. ①, ②식의 유사도 측정 방법에 따라 후보단어를 추출하여 난이도에 따른 원본 문장의 키워드들이 추출된 사상을 S로 하였을 때 아래와 같은 확률을 얻을 수 있다.

$$P(S_i) = P(W_0) * P(W_1) \dots * P(W_{n-1}) \quad ③$$

③식의 i는 난이도에 따른 입력 변수이며 n은 키워드의 개수를 의미하며  $W_j$ 는 j의 난이도에 따른 후보 단어와 키워드를 합한 표본 공간에서 키워드가 추출될 사상을 의미한다. 예를 들어 “젊은 친구 시계 탐 9시 부근 무기 점 뒤쪽으로 행운의 열쇠가 있을 걸세”라는 원본 문장이 난이도 1에서 다시 나올 경우는 키워드 10개에 따라 후보 단어가 9개씩 나왔다고 가정하면  $P(S_1)$ 은  $\frac{1}{10^{10}}$  로 원본 문장이 나올 경우가 희박하다. 위의 문장의 난이도 1일 경우 시계의 후보 단어는 ①식에 의하여 {장치, 설비, 이그니션키, 만능 열쇠}를 얻을 수 있으며 ②식에 대하여 {만능 열쇠, 이그니션키}를 얻을 수 있다.

## 2.4. DTD 및 XML 문서 생성 단계



(그림2) NPC 어휘 Tree

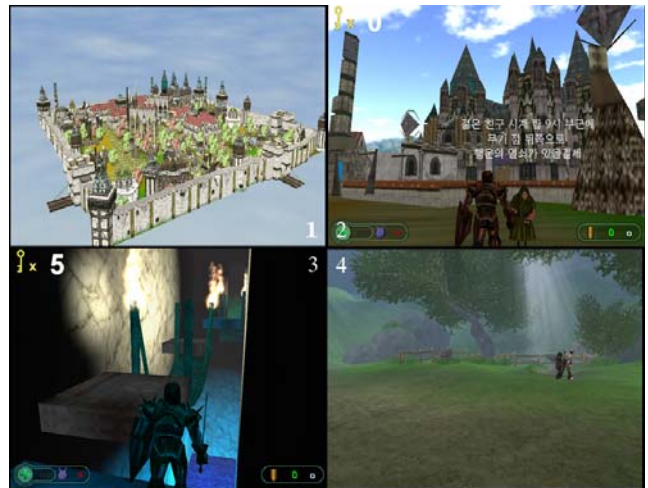
난이도에 따른 문장의 키워드의 후보단어들이 결정되면 (그림2)와 같은 Tree구조의 자료구조를 얻을 수 있다. Rootformat 필드는 NPC들의 모든 문장을 포함하는 최상위 계층을 의미하고, Sentence 필드는 특정 NPC의 문장을, words 필드는 원본 문장의 키워드를, fixword 필드는 조사나 공백과 같이 고정되어 있는 어휘를, word는 후보단어를, level 필드는 유사도를, 그리고 maxword는 후보단어가 몇 개인지를 각각 의미한다. (그림2)를 이용하여 NPC 어휘DTD(그림3)와 NPC 어휘 XML스키마(그림4)를 구성한다. (그림3)과 (그림4)를 이용하면, 효율적인 자원 관리를 통해 다

중 사용자 접속시 문제없이 어휘의 동적 생성이 가능하다.

```

<!ELEMENT Rootformat(Sentence*)>
<!ELEMENT Sentence(originaltext, words*)>
<!ELEMENT originaltext (#PCDATA)>
<!ELEMENT words(fixword|word*)>
<!ELEMENT fixword (#PCDATA)>
<!ELEMENT word(level, maxword, string*)>
<!ELEMENT level(#PCDATA)>
<!ELEMENT maxword(#PCDATA)>
<!ELEMENT string(#PCDATA)>
    
```

(그림 3) NPC 어휘 DTD



(그림 5) 게임 인터페이스

```

<Rootformat>
  <Sentence>
    <Originaltext>젊은 친구 시계 탑 9시 부근에 무
    기 점 뒤쪽으로 행운의 열쇠가 있을걸세</Originaltext>
    <words>
      <originalword>열쇠 </originalword>
      <word>
        <level>1</level>
        <maxword>4</maxword>
        <string>장치</string>
        <string>설비</string>
        <string>이그니션키</string>
        <string>만능 열쇠</string>
      </word>
      ...
    </words>
    <words>
      <fixword>.</fixword>
    </words>
  </Sentence>
</RootFormat>
    
```

(그림 4) NPC 어휘 XML 스키마

### 3. 게임

(그림 5)는 본 논문에서 제안한 게임의 화면들이다. 1번 화면은 소개 화면으로 사용자에게 앞으로 진행할 탭에 대하여 소개한다. 2번 화면은 마을 안에서 키를 획득하기 위해 NPC 들에게 정보를 얻는 모습이다. 3번 화면은 키를 5개 다 획득 하여서 사탄이 살고 있는 성으로 들어가 사탄의 함정을 피해나가는 모습이다. 마지막으로 4번 화면은 엔딩화면으로 공주를 만나 기쁨을 만끽하는 모습이다.

#### 3.1 게임 소개

본 게임은 아케이드 와 어드벤처를 결합한 게임이다. 사탄에게 납치된 공주를 찾기 위해 왕자는 사탄이 살고 있는 성으로 들어가야만 한다. 성을 들어가기 위해서는 키가 5개 필요하며 이를 위해서는 이 지역에 살고 있는 주민(NPC)들의 도움이 필요하다. 주민들에게 키의 위치 및 사탄이 살고 있는 성안의 함정들을 피할 방법에 대한 정보를 얻어 공주를 구출하는 게임이다.

#### 3.2 게임 의 주요 요소

게임을 위해서는 게임 엔진 이외에 XML형식의 NPC 어휘문장을 해석할 XML 파서(그림 5)가 요구된다. (그림6)는 난이도와 문장을 입력 값으로 하여 게임서버에 전송을 하면 게임서버는 이에 해당하는 문장을 전송해준다. 서버는 그 문서에서 각 키워드에 대하여 유사도에 따른 후보 어휘를 추출하고 또한 조사와 같은 고정된 어휘 역시 동적으로 추출하여 완성된 문장을 클라이언트에게 보내준다.

### 4. 실험 및 평가

본 연구에서는 워드넷의 유사도 측정 방법을 게임에 적용하는 것이 가능한지를 판단하고, 어느 유사도 방법이 본 게임에 가장 적합한지를 알아보기 위하여, 반복적으로 게임을 경험하게 하는 방식으로 실험을 진행하였다. 본 실험을 위하여 아래와 같이 3가지의 게임 환경을 준비하였다.

```

string getSentence(level, sentence)
begin
  string completesentence;
  search(sentence);
  while(get(words))
  begin
    if( level = words.level )
      string word;
      int wordcnt = getwordcnt();
      if(wordcnt == 0)
        word = getfixword();
      else
        int randcnt = random(wordcnt);
        word = getword(cnt);
      endif
      stringcat(completesentece, word);
    end if
  end while
end function

```

(그림 6) XML 파서 슈도 코드

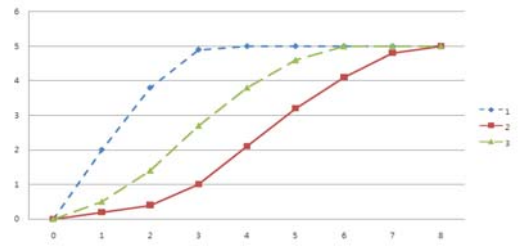
1. 문장이 고정된 경우
2. 유사도 방법 1
3. 유사도 방법 2

(표 1) 처음 수행한 사용자의 결과

	0	1	2	3	4	5
1	10	14	26	0	0	0
2	9	39	2	0	0	0
3	11	16	23	0	0	0

(표 1)은 처음 게임을 접한 사용자들이 키를 획득한 개수를 의미한다. 예를 들어 1번 실험의 경우에는, 처음 수행한 사용자 중에서 키를 전혀 획득하지 못한 경우가 10명, 1개 획득한 경우가 14명, 그리고 3개를 획득한 경우가 26명이었다. 전체 실험에서 처음 게임을 접한 사람들이 키를 획득하는 경우는 2개가 최고이다 1,3의 경우를 제외하고 2의 경우는 유사도 3의문장이 나올 경우 원뜻과 많이 떨어진 문장이 나오기 때문에 키를 획득하기가 다소 힘든 결과를 보인다. (그림 7)은 각 경우마다 키를 다 찾는 데 소요되는 시간을 보여주는 그래프이다. 정적으로 문장을 제공하였을 경우에는 매번 사용자가 문장에 대해 이미 암기를 하기 때문에 키를 다 획득하는 데 많은 시간이 걸리지 않았다. 반면에 난이도에 따라 동적으로 문장을 생성하였을 경우에는 문장 해석의 어려움으로 열쇠를 모두 찾는 것에 많은 시간이 요구되었으나 사용자는 다소 어려운 문장임에도 불구하고 어느 시간이 경과되면 열쇠를 다 획득하는 결

과를 얻을 수 있었다.



(그림 7) 시간에 따른 열쇠 획득 그래프

## 5. 결론

워드넷상에서 어휘 간 유사도 측정하는 방법을 게임에 적용하여 NPC가 다양한 어휘 생성이 가능하게 하여 보았다. 이를 통해서 다양한 유사도 측정 방법들의 테스트 모델이 되어 어느 방법이 가장 게임에 적용하기 적합한 지 알 수 있었다.

## 6. 감사의 글

이 논문은 2006년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음(KRF-2006-331-D00534)

## [참고문헌]

- [1][http://en.wikipedia.org/wiki/Non-player\\_character](http://en.wikipedia.org/wiki/Non-player_character)
- [2]G. A. Miller, "WordNet: An On-Line Lexical Database", International Journal of lexicography, 1990.
- [3]강승식, 범용 형태소 분석기 "HAM Ver 6.0.0", 국민대학교 자연 언어 정보 검색 연구실 <http://nlp.kookmin.ac.kr>
- [4]Graeme Hirst and David St-Onge, "Lexical chains as representations of context for the detection and correction or malapropisms," In Fellbaum, pp 305-332, 1998.
- [5]Claudia Leacock and Martin Chodorow, "Combining local context and WordNet similarity for word sense identification," In Fellbaum, pp 265-283, 1998.