

언어생활을 반영한 지능적 끝말잇기 프로그램 구현

임희석^{1*}

An Implementation of Intelligent Word Relay Game Considering Characteristics of Real World Language

Heui-Seok Lim^{1*}

요 약 본 논문은 한국어의 사용 패턴을 고려한 지능적 끝말잇기 프로그램을 구현하였다. 끝말잇기 프로그램은 뇌 손상등을 통한 실어증 환자나 언어 능력 발달이 상대적으로 낮은 아동들의 언어 능력을 진단하고, 재미있게 재활치료를 하는데 기여할 수 있다. 하지만 컴퓨터는 사람과 비교하여 대용량의 단어 목록을 기억하기에 유리하므로 끝말잇기 프로그램에서 컴퓨터는 훨씬 유리한 조건이다. 따라서 컴퓨터의 보유 단어 목록에서 임의의 끝말을 생성하는 끝말잇기 프로그램은 지능적인 게임의 상대자로 부적합하며 게임자의 의욕을 저하시킬 수 있다. 게임자의 단어 사용 수준에 따라 컴퓨터의 대답의 난이도를 조절하고, 실제 언어생활의 사용 패턴을 반영한 끝말잇기 프로그램은 게임자로 하여금 컴퓨터 상대방이 사람인 것처럼 느끼게 할 수 있으며, 이를 통하여 끝말잇기 프로그램을 지속시킬 수 있다.

Abstract An word relay game contributes to rehabilitation and treatment of language disorders such as aphasia. As a computer is better than human in memorizing very large vocabularies, the computer has much advantage over people in word relay game. Such the game result in decrease the motivation of players and patients in treatment of language disorders. To make people to continue word relay and to be effective to remedy language disorders, the game need to be intelligent and familiar with a person. This paper proposes an implementation of intelligent Korean word relay game, which considers characteristics of Korean word usage patterns. The gaem is intelligent in constructing vocabulary database and choosing an answer considering the level of player.

Key Words : Word Relay Game, Aphasia, Language Disorder, Language Treatment

1. 서론

언어 능력 진단 및 평가는 인지 기능을 측정할 수 있는 중요한 요소 중 하나이다. 최근 언어 발달 장애 환자와 치매 또는 뇌졸중 등으로 인한 실어증 환자도 늘어나고 있는 추세이다[1, 2, 3]. 후천성 언어 발달 장애나 뇌손상으로 인한 실어증은 지속적인 재활 훈련과 치료로 인하여 그 증상이 완화될 수 있음이 보고되고 있다[6, 7, 8].

끝말잇기 게임은 전통적으로 사람들 사이에서 행해진 자연스러운 게임 방법이며 어휘를 사용하는 이유로 언어 능력 학습, 진단 및 평가를 위해서도 좋은 방법이다. 하지만 끝말잇기 프로그램은 저장 능력이 사람보다 훨씬

뛰어난 컴퓨터가 더 유리하다. 게임자의 입력 단어의 끝 음절로 시작하는 단순한 단어 출력은 게임자로 하여금 무료하고, 지루한 게임으로 느껴지며 지능적 게임이라 할 수 없다. 사람과 같은 특성을 가지며 지능적인 끝말잇기 프로그램이 되기 위해서는 컴퓨터가 응답하는 끝말의 생성이 실제 언어생활의 특성을 반영할 수 있어야 하며, 저장 능력이 뛰어난 컴퓨터를 이길 수 없다는 선입감도 해소될 수 있어야 한다. 인터넷 게임의 발전과 닌텐도 DS와 같은 휴대용 단말기가 발전했음에도 불구하고 게임의 기능과 교육적인 효과, 언어 능력 진단 및 평가 기능이 있는 한국어 끝말잇기 게임은 거의 개발된 적이 없다. 본 논문은 실제 사람이 생성한 것과 같은 끝말 생성과 인간 친화적 게임이기 위한 요소를 갖춘 끝말잇기 프로그램을 제안한다. 제안하는 끝말잇기 프로그램은 대용량 코퍼스에서 추출한 통계정보를 이용하여 컴퓨터가 생성할 단어의 난이도를 조절하며, 확률적으로 컴퓨터 대답을 만

본 논문은 2007년도 한신대학교 학술연구비 지원에 의하여 연구되었음

¹한신대학교 컴퓨터공학부

*교신저자 : 임희석 (limhs@hs.ac.kr)

들 수 있는 기법을 사용한다.

2. 끝말잇기 게임의 개요

제안하는 끝말잇기 게임 인간의 실제 언어생활을 반영한 지능적 난이도 조절과 look-ahead 알고리즘을 이용한 끝말 선택 알고리즘을 사용하는 지능적인 끝말잇기 게임이다.

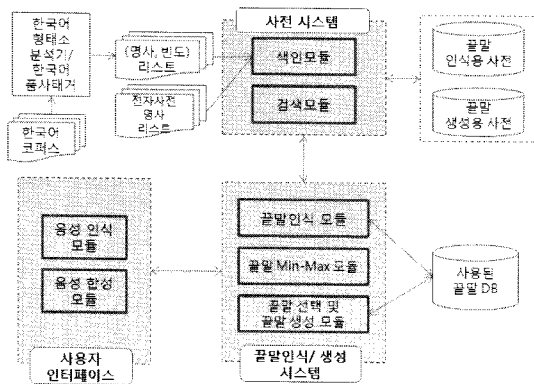


그림 1. 제안하는 끝말잇기 프로그램 구성도

[그림 1]은 제안하는 끝말잇기 게임의 시스템 구성도를 나타낸 것이다. 제안하는 게임은 크게 사용자 인터페이스, 사전 시스템, 그리고 끝말 인식/생성 시스템으로 구성된다. 사용자 인터페이스는 음성 인식 모듈과 음성 합성 모듈로 구성된다. 음성 인식 모듈은 사용자가 입력한

음성 단어를 해당 문자열로 출력하여 끝말 인식 모듈의 입력값을 생성하는 기능을 수행하고, 음성 합성 모듈은 끝말 생성 모듈에서 생성된 끝말을 음성으로 합성하여 사용자에게 출력하는 기능을 수행한다. 사전 시스템은 끝말 인식과 끝말 생성 모듈에서 사용하는 끝말 인식용 사전과 끝말 생성용 사전을 색인하고 검색하는 색인 모듈과 검색 모듈로 구성된다. 끝말 인식용 사전은 사용자가 입력한 끝말이 올바른 단어인지 탐색하기 위하여 사용되는 사전이며 일반적으로 자연어처리를 위하여 사용되는 전자사전 내에 포함된 명사류와 단어와 한국어 코퍼스로부터 추출한 명사류 단어로 구성된다. [그림 2]는 제안하는 끝말잇기 프로그램의 동작 방식을 슈도우 알고리즘(pseudo algorithm)으로 나타낸 것이며, 각 모듈의 세부 구성과 동작 원리는 뒷장에서 설명하도록 한다.

3. 사전의 색인 및 검색

제안하는 끝말잇기 프로그램에서는 끝말 인식용 사전과 끝말 생성용 사전이 사용된다. 끝말 인식용 사전은 게임자가 입력한 단어가 올바른 단어인지를 확인하기 위하여 사용되는 사전이며, 끝말 생성용 사전은 컴퓨터가 끝말을 생성하기 위하여 사용되는 사전이다.

끝말 인식용 사전은 사용자가 올바른 단어 중 어느 것이라도 입력할 경우, 이를 인식할 수 있어야 하므로 한국어에서 사용될 수 있는 가능한 모든 체언이 엔트리에 포함되어야 한다. 현재 끝말 인식용 단어로 사용되는 체언은 보통명사, 대명사, 그리고 보통명사와 대명사가 복합

```

1: 게임 모드 설정; // Easy, Medium, Hard
2: 게임 파라미터 설정; // t : 게임 모드에 따른 사용자 입력 대기 시간 임계치
3:                                     // p : 컴퓨터가 게임에 이길 확률값 설정
4: 사용자 음성 단어 대기 및 입력;
5: if ( 대기시간 > t ) 컴퓨터가 승리했음을 출력하고 게임 종료;
6: 음성 인식을 통한 사용자 음성 단어의 텍스트 단어(w) 생성;
7: if ( word_recognition(w) == FAIL ) 컴퓨터가 승리했음을 출력하고 게임 종료;
8: Min-Max 알고리즘을 이용한 끝말 후보집합(AC_list) 생성; // Answer Candidates list
9: do {
10:  AC = AC_list에서 임의의 끝말 선택;
11: } while ( AC ∈ 사용된 끝말 DB );
12: if ( AC == NULL ) 사용자가 승리했음을 출력하고 게임 종료;
13: [0, 1] 사이의 임의의 확률값(p) 선택;
14: if ( p > 확률 임계치 ) 음성 합성 모듈을 이용한 AC 음성 합성;
15: 합성된 음성 출력;
16: goto 4;
    
```

그림 2. 지능형 끝말잇기 프로그램 알고리즘

된 복합명사이다.

끝말 인식용 단어는 일반적인 자연어처리 시스템에서 사용하는 전자사전에서 해당 품사를 추출함으로써 획득할 수 있다. 하지만 전자사전에 등재되지 않았던 단어나 신조어의 생성으로 인한 단어들은 전자사전에서 획득할 수 없다. 전자사전에서 직접 획득할 수 없는 단어라도 게임자에 의하여 단어로 입력될 가능성이 있다. 제안하는 끝말잇기 프로그램에서는 대용량 한국어 코퍼스와 언어처리 도구를 이용하여 끝말잇기 단어를 추출하여 추가하였다. 한국어 코퍼스에서 추출한 단어는 사람들이 실제 사용하는 언어에서 단어를 추출한 것이므로 지능적인 게임을 위해서도 반드시 필요한 것이다. [그림 3]은 끝말 인식용 단어 리스트를 구축하는 과정을 도식화한 것이다.

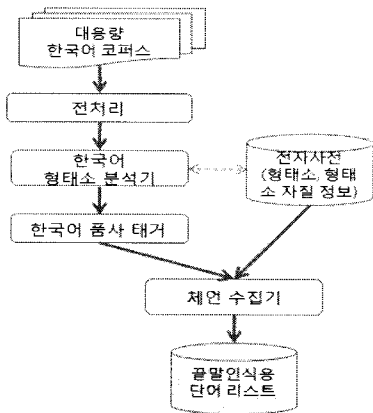


그림 3. 끝말 인식용 단어 리스트 구축 과정

대용량 한국어 코퍼스로는 1천만 어절 크기의 세종코퍼스가 사용되었다. 전처리기는 문장 기호, 한자 분리 등을 통한 어절 분리기 역할을 수행하며 한국어 형태소 분석기는 어절의 가능한 모든 한국어 형태소 결합열을 생성하는 역할을 수행한다[4, 5]. 한국어 품사 태거는 한국어 형태소 분석 결과 중 문장내에서 올바른 형태소 결합열을 생성하는 역할을 수행한다. 한국어 형태소 분석기와 한국어 품사 태거로는 [5]의 시스템을 사용하였다. 체언 수집기는 한국어 품사 태거 결과에서 명사와 고유명사, 그리고 복합명사를 추출하는 기능을 수행한다. 한국어 코퍼스로부터 추출된 끝말 인식용 단어의 개수는 86,542개였으며 전자사전에서 추출한 단어의 수는 142,036개이다. 코퍼스에서 추출된 단어와 전자사전에서 추출된 단어 중 중복된 것을 제외하여 통합한 단어 수는 총 182,478개이다.

끝말 생성용 사전은 컴퓨터가 끝말을 생성하기 위하여 사용되는 사전으로 끝말 인식용 사전과는 달리 (단어, 빈도)의 형태와 같은 레코드로 구성되어 있다. 이처럼 단어의 빈도가 필요한 이유는 생성하는 단어의 난이도 조절을 위하여 단어의 빈도를 사용하기 때문이다. 예를 들어 컴퓨터가 생성할 단어의 끝음절로 시작하는 단어 후보의 수가 적거나 각 단어 후보의 빈도가 적은 경우, 그 단어는 난이도는 높은 단어라고 할 수 있다. 난이도 조절에 대한 자세한 사항은 끝말생성 알고리즘에서 설명하도록 한다. 현재 끝말 생성용 사전의 단어 리스트로는 한국어 코퍼스에서 추출한 86,542개의 단어가 사용된다.

끝말 인식용 사전과 끝말 생성용 사전 구축을 위하여 추출된 데이터는 게임중 빠른 검색을 위하여 미리 색인

```

// word_recognition 함수
trial = 0;
while ( trial < 2 ) {
    사용자 음성 단어 대기 및 입력;
    if ( 대기시간 > t ) return FAIL;
    else {
        음성 인식을 통한 사용자 음성 단어의 텍스트 단어(w) 생성;
        if ( w ∈ 끝말인식용 사전 and w ∉ 사용된 끝말 DB ) return SUCCESS;
        if ( w ∉ 끝말인식용 사전 ) 부적합한 단어임을 출력;
        else if ( w ∈ 사용된 끝말 DB ) 이미 사용된 단어임을 출력;
        if ( trial == 1 ) return FAIL;
        else trial = trial + 1;
    }
}
    
```

그림 4. 끝말 인식 알고리즘

1 컴퓨터 다음 차례에 사람이 생성하여야 하는 단어 후보

하여 저장하는데, 현재 GDBM과 본 연구팀에서 자체개발한 B-tree 방식의 색인 방법이 사용된다.

4. 지능형 끝말잇기 게임 알고리즘

4.1 끝말 인식 알고리즘

게임자가 입력한 단어가 올바른 단어인지 조사하는 끝말 인식 알고리즘은 입력된 단어가 끝말 인식용 사전에 포함되어 있는 사전인지 검사하는 것으로 수행된다. 단어 입력 시간의 제한은 게임의 모드에 따라 다르며, 게임자는 입력 제한 시간 안에 단어를 입력하여야 한다. 만약 입력된 단어가 컴퓨터나 게임자가 이미 사용한 단어이면 해당 사항을 출력하고, 한번의 기회를 다시 주어 재입력하게 한다. 만약 1번째의 시도에서 제한된 시간 안에 단어를 입력하지 못하거나 이미 사용된 단어의 사용으로 인하여 주어진 2차 시도에서도 올바른 단어를 입력하지 못하는 경우 사용자는 게임에서 지게된다. 끝말 인식을 위한 알고리즘(word recognition)은 [그림 4]와 같다.

4.2 끝말생성 알고리즘

전자사전내의 단어와 코퍼스에 포함된 모든 단어를 끝말생성에 사용할 경우, 게임자보다 컴퓨터가 훨씬 더 많은 단어 정보를 가지고 있으므로 게임자가 매우 불리한 게임을 하게 된다. 또한 제안하는 프로그램을 이용한 언어 수준에 맞는 어휘 학습 훈련과 평가를 위해서는 게임자의 수준에 적합한 끝말이 생성될 수 있도록 하여야 한다.

제안하는 프로그램에서는 코퍼스에서 추출한 단어만을 컴퓨터가 생성할 수 있는 끝말단어로 사용하며, 컴퓨터의 대답을 사람 게임자의 난이도를 반영하여 계산할 후 있는 Min-Max 알고리즘을 이용한다. 난이도 조절을 위하여 코퍼스에 반영된 단어들의 실제 사용 빈도를 사용한다. 난이도 조절을 위해서는 컴퓨터가 생성할 단어의 끝음절로 시작되는 단어의 수와 각 단어의 빈도 정보가 사용된다. 예를 들어, [그림 5]처럼 게임자가 “학교”라는 단어를 입력했고, 컴퓨터가 “교”로 시작되는 단어를 생성하려는 경우를 생각해 보자. “게임자” 레벨은 게임자가 끝말을 생성하는 경우이고, “컴퓨터” 레벨은 컴퓨터가 끝말을 생성하는 경우이다. [그림 5]는 게임자가 “학교”를 제시했고, 컴퓨터가 “교”로 시작되는 단어와 해당 단어를 생성했을 때 다음 차례인 게임자가 선택할 수 있는 이웃 단어들의 예를 보이고 있다.

정의 1. 이웃단어 크기
 이웃단어 크기는 특정 음절로 시작되는 단어의 개수를 해당 음절의 이웃단어 크기라고 정의한다.

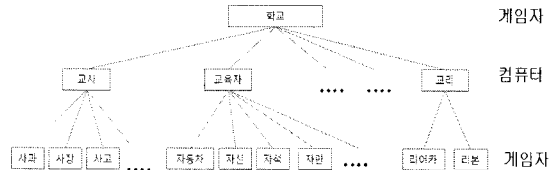


그림 5. 학교에 대한 생성 가능한 단어 예

[그림 5]에서 “교”로 시작되는 단어는 “교사”, “교육자”, “교장”, “교리” 등이 있을 수 있다. “사”의 이웃단어 크기는 50, “자”의 이웃단어 크기는 100, 그리고 “리”의 이웃단어 크기는 2라고 가정하자. 게임자가 “학교”라는 단어를 입력한 경우 “교사”, “교육자”, “교리” 등 “교”로 시작되는 단어를 컴퓨터가 응답으로 출력할 수 있다. 컴퓨터가 “교사”를 응답하는 경우 다음 차례의 응답자인 게임자는 “사”로 시작되는 단어 50가지 중 한 가지를 선택할 수 있으며, “교육자”인 경우 100가지 단어 중 한 가지를 선택할 수 있다. 만약 컴퓨터가 “교리”를 응답으로 선택한 경우 사용자는 “리”로 시작되는 단어 2가지 중 한 가지를 선택하여야 한다. 이럴 경우 게임자의 입장에서 보면 단어 선택의 폭이 넓은 “교육자”가 컴퓨터의 응답으로 출력되는 것이 가장 난이도가 쉽고, “교리”가 가장 난이도가 높은 단어가 된다. 따라서 컴퓨터가 응답한 다음 사람이 답변을 할 수 있는 단어까지 고려할 경우 난이도는 “교리” > “교사” > “교육자” 순이라 할 수 있다. 컴퓨터가 생성할 단어의 난이도는 해당 단어의 끝음절의 이웃크기(NS:Neighborhood Size)에 반비례한다고 할 수 있다.

생성된 끝말의 난이도에 영향을 미치는 또 다른 요소 중 하나는 이웃 단어들의 빈도이다. 이웃 단어 중 고빈도의 단어가 포함되어 있을 경우 게임자가 선택하기 쉬운 단어이고, 그렇지 않고 저빈도의 단어들인 경우 난이도가 높은 단어에 해당된다. 즉 생성될 끝말의 난이도는 이웃 단어 중 고빈도에 해당하는 단어의 수(HFNS), 중빈도에 해당하는 단어의 수(MFNS)에 반비례하고, 저빈도 단어의 수(LFNS)에 비례한다고 할 수 있다.

제안하는 프로그램은 끝말 생성을 위한 단어 난이도 계산을 위하여 [식 1]과 같은 난이도 계산식을 사용한다. α , β , γ , 그리고 δ 는 각 요소의 가중치를 결정하는 상수로서 현재 $\alpha=0.5$, $\beta=0.2$, $\gamma=0.1$, 그리고 $\delta=0.2$ 를 사용한다. 이 값들의 이상적인 값을 찾는 문제는 NP-complete

문제에 해당하므로 경험에 의해서 결정되었다.

$$\text{난이도}(w) = \alpha \log \frac{N}{NS} + \beta \frac{NS}{HFNS} + \gamma \frac{NS}{MFNS} + \delta \frac{NS}{LFNS} \quad [\text{식 1}]$$

where $\alpha + \beta + \gamma + \delta = 1.0$

[식 1]에 의해서 끝말 후보들의 난이도를 계산하고 난이도 값에 따라 끝말후보들을 정렬한다. 최종적으로 컴퓨터가 답변할 단어는 게임의 모드에 따라서 결정된다. 게임의 모드는 Easy 모드, Normal 모드, Hard 모드가 있으며, Easy 모드의 경우 난이도가 낮은 후보 중 임의로 하나를 생성하며, Hard 모드의 경우 난이도가 높은 후보 중 임의의 하나를 선택한다. Normal 모드의 난이도가 중간인 후보 중 임의의 단어를 선택한다.

난이도와 모드에 의하여 생성된 단어는 확률적으로 선택되어 게임자에게 제시함으로써 게임자로 하여금 컴퓨터가 사람과 같이 실수와 오류를 범할 수 있는 존재임을 느끼도록 하는 인간 친화적 게임이 될 수 있도록 하였다. 확률적으로 끝말을 생성하는 방법은 0~1 사이의 임의의 확률값 p를 선택하고, 이 값이 게임 모드에 따라서 미리 정해진 임계값(p')보다 클 경우에만 끝말을 생성하고, 그렇지 않은 경우 끝말이 생성되지 못하여 게임자가 승리할 수 있도록 한다는 것이다. 확률 임계값 p'은 게임 모드에 따라 결정될 수 있도록 설계하였다. 특정 단어의 난이도 계산식과 게임 상대방의 행위를 반영하여 지능적으로 끝말을 생성하는 Min-Max 알고리즘은 [그림 6]과 다음과 같다.

[그림 6]의 알고리즘이 동작하기 위해서는 MOVEGEN 함수와 STATIC 함수가 필요한데, 각 함수의 설명은 [표 1]과 같다.

표 1. Min-Max 알고리즘 보조 함수들

- | |
|--|
| <ol style="list-style-type: none"> 1. MOVEGEN(word, Player) <ul style="list-style-type: none"> - word 끝 음절로 시작하는 단어들 생성 - player에 의해서 현재 만들어질 수 있는 단어들의 리스트를 return 2. STATIC(word, Player) <ul style="list-style-type: none"> - 현재 단어가 사용자에게 있어서의 얼마나 유리한가를 나타내는 값 - [식 1] 난이도(w)값의 음수값 |
|--|

5. 구현 및 실험

끝말잇기 프로그램은 데스탑 버전으로 MS-windows XP, RedHat 9.0 버전으로 개발되었으며 PDA 버전으로는 ARM 프로세서 용도로도 개발되었다. [그림 7]의 실험 예는 음성인식 모듈과 음성 합성 모듈을 통합하지 않은 텍스트 버전이다. 개발된 끝말잇기 프로그램은 게임의 용도 뿐만 아니라 유아들의 언어 능력 평가 및 실어증 환자들의 진단 및 언어 재활 학습이 또 다른 용도이다. 유아와 실어증 환자의 언어 능력은 게임을 진행하면서 만들어지는 로그 파일을 분석함으로써 이루어 질 수 있으며, 언어

- | |
|--|
| <ol style="list-style-type: none"> 1: If DEEP-ENOUGH(word, Depth) then return the structure. 2: VALUE = STATIC(word, Player) 3: PATH = nil. 4: Otherwise, MOVEGEN(word, Player)을 호출하여 다음 게임자의 상태를 생성하고 return된 결과(structure)를 SUCCESSORS로 셋팅 5: If SUCCESSORS가 empty이면 해당 결과(structure)를 return. 6: IF SUCCESSORS가 empty가 아니면 각 요소를 조하하고 최상의 결과를 저장 7: Initialize BEST-SCORE를 STATIC가 return한 최소값으로 할당 8: SUCCESSORS의 각 SUCC에 대해서 다음과 같은 반복문 수행; 9: (a) RESULT-SUCC를 MINIMAX(SUCC, Depth + 1, OPPOSITE(Player))으로 할당 11: (b) NEW-VALUE를 -VALUE(RESULT-SUCC)값으로 할당 12: (c) If NEW-VALUE > BEST-SCORE then 13: (i) BEST-SCORE를 NEW-VALUE로 할당 14: (ii) BEST-SCORE를 PATH(RESULT-SUCC)의 앞에 SUCC를 붙인 결과를 할당 15: 해당 결과(structure)를 return 16: VALUE = BEST- SCORE 17: PATH = BEST - PATH |
|--|

그림 6. 난이도 계산을 이용한 끝말생성 Min-Max 알고리즘

능력 분석을 위하여 사용될 수 있는 기준은 [표 2]와 같다.

[표 2]의 기준을 이용한 언어 능력 분석은 정상 집단의 평균값과 유아 또는 환자와의 수행 값을 비교함으로써 이루어질 수 있다. 이러한 언어 능력에 대한 분석은 정상 집단의 통계적인 평균값 구축과 사례 연구가 이루어져야 하는 것으로 향후 연구 과제로 진행하고자 한다.

표 2. 언어 능력 분석 기준

기준	비고
게임을 수행할 수 있는 단계	게임의 지속 ply 측정
게임 중 사용하는 단어의 빈도 특성	고빈도, 저빈도 단어 사용 여부
사용되는 단어의 길이	사용단어의 난이도 측정
평균 반응 시간	반응 시간의 차이

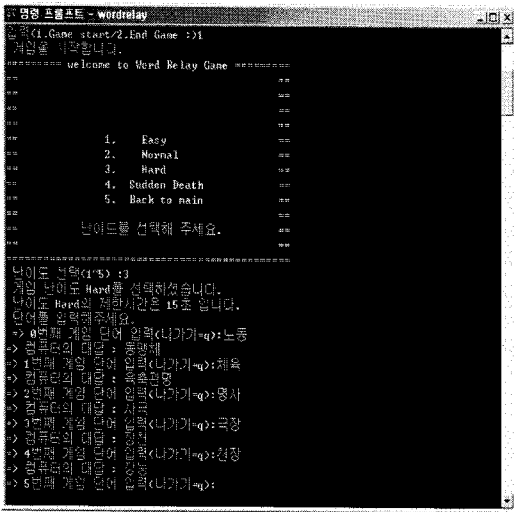


그림 7. 끝말잇기 프로그램 실행 예

6. 결론

본 논문은 실제 사람들의 언어 생활을 반영할 수 있는 대용량 한국어 코퍼스과 전자 사전을 이용하여 끝말을 생성하고, 코퍼스의 통계 정보를 이용하여 끝말의 난이도를 조절할 수 있는 지능적인 끝말잇기 프로그램을 제안하였다. 제안한 끝말잇기 프로그램은 끝말인식을 위하여 총 182,478개의 단어를 사용하며, 끝말 생성을 위하여 86,542개의 단어를 사용한다. 끝말 생성을 위하여 사용되는 단어 목록은 사람이 끝말잇기 게임을 하는 것처럼 모

사하기 위하여 대용량의 코퍼스에서 사람들의 언어 생활을 반영하여 추출하였으며, 컴퓨터의 끝말생성을 위해서는 Min-Max 알고리즘을 사용하였다. Min_Max 알고리즘은 끝말을 생성할 때 컴퓨터가 생성한 끝말의 끝음절을 이용한 사용자의 답변과 그 이후 다시 컴퓨터의 답변 등을 고려하여 난이도를 결정하는 지능적인 알고리즘이다.

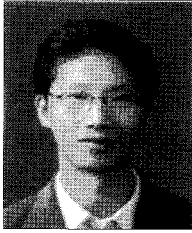
제안된 끝말잇기 게임은 게임의 목적뿐만 아니라 언어 능력 진단 및 평가를 위해서도 사용될 수 있는 도구로 정상인 집단의 수행 능력 측정 및 실어증 환자나 언어 발달 장애 환자와의 사례 연구가 뒤따를 예정이다.

참고문헌

- [1] 남기춘, 김동휘 (2002). 인간언어정보처리와 관련된 대뇌영역. 한국심리학회지:실험 및 인지, 14권, 4호, 291-307.
- [2] 이홍재, 편성범, 채수경, 남기춘(2000), “정상인과 명칭성 실어증 환자에게 보이는 의미적 관련성의 좌우 반구 편재화”, 한국심리학회지: 일반19권 1호, 103-118.
- [3] 남기춘, 편성범, 임희석, 김동휘, 정재범 (2004), 특허출원 : 명칭성 실어증 검사장치 (Apparatus for diagnosing anomia), 특허-2003-0078084(PATENT-2003-0078084).
- [4] 황유미, 남기춘 (2000). 한국어 관계대명사의 정보처리:실어증 사례를 중심으로. 2000년 한국뇌학회 학술대회 논문집, p. 25.
- [5] 임희석, 언어 지식과 통계 정보를 이용한 한국어 품사 태깅 모델, 박사학위논문, 고려대학교, 1997.
- [6] Silver, M. C., Perri, R., & Cappa, A. (2003). Grammatical class effects in brain-damaged patients: functional locus of noun and ver deficit. Brain and Language, Vol. 81, No. 1, p. 49-66.
- [7] Van Turenhout, M., Hagoort, P., and Brown, C. M. (1998). Brain activity during speaking: From syntax to phonology in 40 milliseconds. Science, Vol. 280, p. 572-574.
- [8] Zahn, R., Drews, E., Specht, K., Kemeny, S., Reith, W., Willmes, K., Schwarz, M., & Huber W. (2004). Recovery of semantic word processing in global aphasia: a functional MRI study. Cognitive Brain Research, Vol. 18, No. 3, p. 322-336.

임 희 석(Heui-Seok Lim)

[종신회원]



- 1992년 2월 : 고려대학교 컴퓨터학과 (이학사)
- 1994년 2월 : 고려대학교 컴퓨터학과 (이학석사)
- 1997년 9월 : 고려대학교 컴퓨터학과 (이학박사)
- 1997년 9월 ~ 1999년 2월 : 삼성종합기술원 HCI Lab.
- 1999년 3월 ~ 현재 : 한신대학교 컴퓨터공학부 부교수

<관심분야>

자연어처리, 인공지능, 인지신경계산학, 정보검색