

언어 장애인을 위한 한국어 지원 시스템의 구현

Implementation of Korean Support System for Language Disorders

최정환*, 추교남, 우요섭
J. H. Choi, K. N. Choo, Y. S. Woo

요 약

대부분의 보완·대체 의사소통은 별도의 전용기기를 사용하여 제공하거나 휴대가 어려운 데스크톱 또는 현재는 거의 사용하지 않는 Windows 기반의 태블릿 등에서 제공되는 연구들이 대부분이었다. 게다가, 앞선 연구들의 고유명사 처리는 사전을 만들어 제공하는 형식이었기 때문에 무수히 많은 고유명사에 대한 처리가 거의 불가능하였고 향후 연구과제로 많이 지적되고 있었다. 본 논문에서는 현재 급속도로 보급되고 있는 스마트 기기를 활용하여 과거의 연구 과제로 남았던 고유명사 처리에 대한 해결 방법을 제안한다. 또한, 신체장애를 가진 사람을 위하여 버튼을 이용한 입력방법과 키보드를 통한 입력 시 생길 수 있는 오류에 대한 보완 방법도 제안한다. 본 연구에서는 제안하는 보완·대체 의사소통 시스템의 유용성을 확인하기 위하여 안드로이드 기반의 모바일 어플리케이션을 구현하였다. 실험결과 사용자의 위치와 선택에 따라 주변에 있는 다양한 고유명사가 도출되어 문장을 생성하는데 도움을 주는 것을 확인할 수 있었다.

ABSTRACT

Most Augmentative and Alternative Communication(AAC) use exclusive equipment or studied desktop, tablet PC based windows. Besides, the preceding study offers proper noun dictionary so, henceforward study has problem to innumerable proper noun processing. This paper suggests a method of proper noun processing using a mobile smart equipment. And via the button with virtual keyboard input method and the errors that can occur is also proposing a complementary way. AAC system to check availability for application on Android has been implemented. Experimental results, depending on user location and selection of proper nouns in the around could produce a sentence is derived.

Keywords : Language Support, Korean Language Processing, Mobile System

1. 서론

본 논문에서는 스마트폰이나 태블릿을 이용해 후천적인 언어활동 장애나 선천적인 언어활동장애가

있는 사람들이 간단한 조작으로 일상생활에 필요한 문장을 생성해 일반인들과의 짧은 의사소통을 도와 줄 수 있는 시스템을 구현하고자 한다.

언어보조 기기의 목적은 필요한 문장을 생성하여 의미를 효과적으로 전달하는 것이다. 현재 청각장애인들이 사용하는 수화는 일반인들과의 의사소통에서 정확한 의미를 전달하기 힘들다. 현재 2000만대 정도의 보급률을 가진 스마트 폰의 텍스트 입력 어플리케이션을 활용해도 한문자를 입력을 하는 최소 2번에서 많게는 5번까지 입력을 하는 번거로움이 생긴다. 그리고 스마트폰의 제한된 입력방식은 오타를 발생시킬 수 있는 환경이어서 생각보다 많은 입력을 통해 문장을 생성할 수 있게 된다. 본 논문은 휴대성이 좋은 스마트 기기를 이용하여 일반적인

접 수 일 : 2012.05.14

심사완료일 : 2012.06.04

게재확정일 : 2012.06.18

* 최정환 : 한국전력 KDN

amagin1@hotmail.com (주저자)

추교남 : 인천대학교 정보전산원 초빙교수

kyonam@incheon.ac.kr (공동저자)

우요섭 : 인천대학교 정보통신공학과 교수

yswoo@incheon.ac.kr (교신저자)

※ 본 논문은 인천대학교 2011년도 자체연구비 지원에 의하여 연구되었음.

텍스트 입력과는 별도로, 아이콘과 가상키보드를 입력방법으로 하여 언어장애인과 비장애인의 의사소통에서 좀 더 빠르고 효율적인 방법을 제시한다.

2. 관련 연구

언어보조 시스템은 영구적이거나 일시적으로 말이나 글로 의사소통을 할 수 없는 사람들을 위한 표현 의사소통 방법이다. Beukelman 등은 보완·대체의사소통이란 모든 상황에서 독립적으로 의사소통을 할 수 없는 사람들의 의사소통을 지원해 주고, 향상시킬 수 있도록 계획된 모든 접근방법이라고 하였다[1]. 즉, 의사소통, 말하기와 쓰기에 심한 장애를 보이는 사람들의 장애를 일시적 혹은 영구적으로 보완해 주는 임상치료 행위의 한 영역으로 볼 수 있다. 동사예측에 의한 문장생성은 한글문장의 특성을 이용한 경우로서 명사에 따라 연결될 수 있는 동사를 단계적으로 제시해주는 방법이다[2]. 한글 문장의 경우 조사와 동사에 따라 명사의 종류를 한정할 수 있는 특성이 있기 때문이다.

하지만 이 방법은 한 개의 명사에 연결될 수 있는 동사의 수가 많기 때문에 사용자가 원하는 동사가 도출될 때 까지 여러 단계가 소요될 수 있다. 단어선택을 이용한 문장생성 방법은 사용자가 명사와 동사를 선택하여 문장이 완성되도록 하는 방법이다[3].

사용자가 명사와 동사를 선택하면 적절한 조사가 붙어 완전한 문장을 출력하는 것이다. 한글 문장에서 조사와 동사에 따라 명사종류를 한정 할 수 있다는 특징을 이용한 것이다. 하지만 거의 무한한 고유명사를 사전으로 가지고 있을 수 없다는 단점을 가지고 있다. 마지막으로 아이콘을 이용한 방법이 있다[4]. 이 방법은 키보드로부터 문자를 입력받아 제어하는 인터페이스보다 직관적이며 사용하기 쉬운 그림에 어휘 속성을 부여하여 문장을 생성할 수 있도록 한 언어보조도구이다. 자연언어의 중의성을 아이콘에 적용하여 아이콘 열이 이루는 정보에 따라 다양한 표현이 가능하도록 했다. 하지만 이 방법 역시 고유명사에 대한 처리가 미리 준비된 사전으로 밖에 할 수 없는 단점이 있다.

3. 모바일 환경에서의 보완·대체 언어 시스템

3.1 보완·대체 어휘데이터의 구축

언어처리 시스템은 언어에 대한 기반 지식 즉 어

휘데이터와 이를 활용하는 프로그램으로 구성된다. 시스템 구성에 필요한 어휘들을 결정하기 위해 먼저 시스템에 사용가능한 많은 문장들을 수집하여 코퍼스를 구축하였다. 다음에 이 코퍼스로부터 어휘들을 추출하여 어휘데이터 사전을 작성하였다. 명사들은 일반적으로 개수가 많기 때문에 의미적인 개념정보에 따라 분류할 필요가 있다. 논문에서는 세종프로젝트의 명사 분류를 기본으로 하였다[5].

고유명사 사전은 명사사전에 저장되어 있는 추상화된 명사를 세분화하여 저장한 XML 형식의 사전이다. 데이터베이스 안에는 사용빈도에 따라 값을 저장할 수 있는 엘리먼트를 만들어 Cost값에 따라 자주 사용되는 명사를 먼저 보여줄 수 있도록 구성하였다. Change Noun은 사용자가 선택하는 것으로 Origin Noun과 내용을 치환하여 보여주게 된다. 표 1은 XML 형식의 고유명사 사전의 예이다.

하위범주화 사전이란 용언을 중심으로 구성된 사전이다. 용언을 중심으로 체언으로 올 수 있는 명사의 개념범위를 제한시킨다[6]. 즉, 명사의 선택에 의해 체언에 대한 개념범위 규제를 주어 선택의 폭을 좁혀 사용자의 입장에서 체언을 선택하는데 도움을 주고 시스템적인 입장에서 문장 생성 시 조사를 결정하고 시스템이 생성할 수 있는 문장만을 제시하게 되므로 안정성을 보장할 수 있다. 표 2와 같이 하위범주화 사전의 구조는 ID, 용언, 원형, 동사에 사용될 수 있는 체언류, 조사로 구성되어 있다.

표 1. 고유명사 저장모습

```
<ID : 1>
  <Pnoun> BANK </Pnoun>
  <OriginNoun> 은행 </OriginNoun>
  <ChangeNoun>
    <Cost> 0 </Cost>
    우리은행 </ChangeNoun>
  <ChangeNoun>
    <Cost> 0 </Cost>
    국민은행 </ChangeNoun>
  ...
</ID>
```

표 2. 하위범주화 사전의 예

```
<ID : 1>
  <Verb> 있다 </Verb>
  <Origins> 있 </Origins>
  <Con11> C00B00 </Con11>
  <Con12> NULL </Con12>
  <Con21> D00A00 </Con21>
  <Con22> NULL </Con22>
  <Josa1> 은/는 </Josa1>
  <Josa2> 에 </Josa2>
</ID>
```

현재 우리가 쓰고 있는 고유명사의 수는 무한에 가깝다. 그런 고유명사를 사전으로 구성하여 제공하는 것은 사실상 불가능하다. 그래서 본 논문에서는 사용자의 위치에 따라 주변에 있는 고유명사를 처리하는 방법을 제시한다. 이 방법은 사용자가 실외에서 특정장소나 건물 등을 찾아갈 때 자신주변의 있는 찾고자 하는 종류의 고유명사를 보여주게 된다. 먼저 GPS나 네트워크, 통신망을 이용하여 자신의 위치좌표를 얻는다. 그 후 받은 위치정보를 가지고 사용자가 원하는 데이터를 네트워크를 이용하여 추출한 검색결과를 사용자에게 제공한다. 그림 1은 고유명사를 가져오는 흐름을 표현한 것이다.

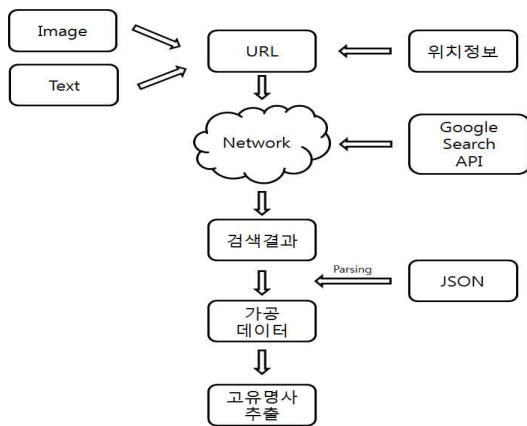


그림 1. 고유명사 추천 과정

3.2 문장생성 알고리즘

문장생성 알고리즘은 3.1절에서 정의한 다양한 사전을 바탕으로 화면에 나타난 이미지 버튼과 텍스트로 만들어진 버튼을 활용하여 원하는 문장을 생성한다. 일반적인 단문형태인 명사1개 + 동사1개, 명사2개 + 동사1개를 입력받아 하위범주화 사전에 있는 조사를 부여하여 완전한 문장의 형태를 이룰 수 있도록 하였다. 이렇게 단문으로 문장을 생성할 수 있게 한 것은 복잡한 문장의 입력은 문장 생성시 모호성이 커지기 때문에 문장에 왜곡이 있을 수 있고, 장애인이 왜곡된 문장을 수정하는 것은 상당히 번거로울 수 있기 때문이다. 따라서 복문이나 혼합문을 만드는 것보다 가급적 짧고 간단한 단문을 이용하여 반복하는 것이 유용하다고 판단하였다. 따라서 언어처리를 위한 코퍼스는 주로 단문형태의 문장을 수집하여 구축하였다. 그림 2에 문장생성에 관한 전체적인 알고리즘을 제시하였다.

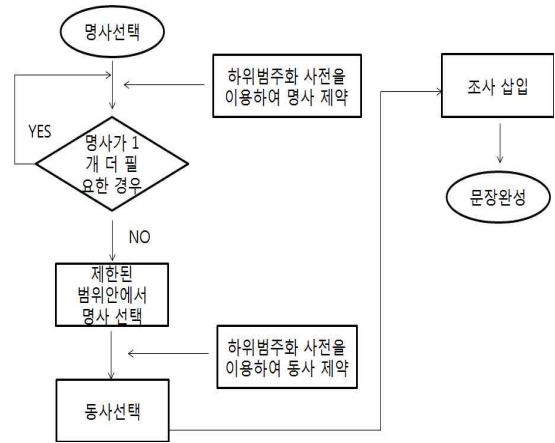


그림 2. 문장생성 알고리즘

3.3 문장비교 알고리즘

이 절에서는 버튼을 이용한 문장생성방법이 아닌 코퍼스에 입력되어 있는 문장을 사용자가 가상키보드를 이용하여 원하는 문장을 검색하여 출력하는 방법을 제시한다.

장애인은 키보드를 이용한 입력을 진행할 때 부정확한 입력을 할 수 있다. 그런 부정확한 입력을 수정하여 다시 입력하려면 입력시간이 증가하는 결과로 이어진다. 따라서 완벽한 단어나 문장이 들어오지 않아도 유사도에 따라 추천문장을 제시해줄 필요가 있다. 논문에서는 DNA염기서열의 유사도를 측정하는 서열정렬 알고리즘[7]을 문장비교에 적용하였다. 서열정렬은 보통 DNA들 간의 염기서열을 비교하여 두 DNA가 얼마나 유사한지를 알아보는 방법 중의 하나이다. 두 개의 DNA의 서열이 유사성이 크다면 기능적으로나 구조적으로 유사할 가능성이 많다는 것이다. 정렬의 방법에는 전역 정렬(Global Alignment)과 지역 정렬(Local Alignment)이 있다. 기본 정의는 문자 집합 Σ 상에서 정의된 길이가 n 인 서열 S 와 길이가 m 인 서열 T 가 존재할 때, $1 \leq i \leq n$ 인 i 에 대하여 $S[i]$ 는 S 의 i 번째 문자를 나타내고, $1 \leq j \leq m$ 인 j 에 대하여 $T[j]$ 는 T 의 j 번째 문자를 나타낸다.

주어진 서열 S 와 T 의 전역정렬은 S 와 T 에 '-'(공란)을 삽입하여 동일한 길이 n' 의 서열 S', T' 로 변환하고, $1 \leq i \leq n'$ 인 i 에 대해 $S'[i]$ 는 반드시 $T'[i]$ 에 대응되도록 한다. 정렬된 서열의 결과 값은 다음과 같이 구할 수 있다. 정렬된 서열을 $1 \leq i \leq n'$ 인 모든 i 에 대해 $(S'[i], T'[i])$ 의 짝으로 나타낸다. $(S'[i], T'[i])$ 에 대해서 다음 4가지 경우가 존재한다. $S'[i]$ 에 나타난 Σ 상에서 정의된 문자를 k , $T'[i]$ 에 나타난 Σ 상에서 정의된 문자를 p 라고 할 때, M .

Tompal[8]는 이 4가지 경우에 대해서 특정 값을 주는 점수 함수를 $\sigma(S'[i], T'[i])$ 로 정의하였다.

$$S'[i] = k, T'[i] = p \text{ 이고,} \\ k = p \text{인 경우 : } \sigma(k, k) \quad (1)$$

$$S'[i] = k, T'[i] = p \text{ 이고,} \\ k \neq p \text{인 경우 : } \sigma(k, p) \quad (2)$$

$$S'[i] = k, T'[i] = - \text{인 경우 : } \sigma(k, -) \quad (3)$$

$$S'[i] = -, T'[i] = p \text{인 경우 : } \sigma(-, p) \quad (4)$$

이 4가지 경우에 대해 각각 다른 가중치 값을 준다. 첫 번째 경우는 상대적으로 높은 값을 준다. 예를 들면, $\sigma(k, k) = 2, \sigma(k, p) = \sigma(k, -) = \sigma(-, p) = -1$ 과 같이 가중치를 다르게 준다. 다음 그림 3는 점수 함수가 위의 예처럼 주어졌을 때, 전역정렬의 예를 보여주고 있다.



그림 3. 전역정렬의 예

Needleman과 Wunsch의 알고리즘[9]은 가장 많이 사용되는 전역정렬 알고리즘 중의 하나이다. 이 알고리즘은 동적 프로그래밍 방식을 가지고 전역 최적 정렬을 구한다. 주어진 서열 S와 T의 최적 정렬 값을 계산하기 위해 $V(i,j)$ 를 $S[1]...S[i]$ 와 $T[1]...T[j]$ 의 최적 정렬의 값으로 정의한다. 그러면 S, T의 최적 정렬 값은 $V(n,m)$ 이 된다. 동적 프로그래밍 방식은 $0 \leq i \leq n, 0 \leq j \leq m$ 인 모든 $V(i,j)$ 의 값을 계산한다. $V(i,j)$ 를 계산 할 때, $V(i,j)$ 를 제외한 이전 값들, 즉, $V(0,0)...V(i-1,j-1), V(i-1,j), V(i,j-1)$ 은 모두 최적 정렬된 값으로 계산되어 있다고 가정한다. 앞에서 설명한 4가지 경우를 모두 고려하면 $V(i,j)$ 값은 (1),(2)와같이 계산된다.

$$i > 0, j > 0 \text{에 대해} \quad (1)$$

- $V(i,j) = \max(V(i-1,j-1) + \sigma(S[i],T[j]),$
 $V(i-1,j) + \sigma(S[i],-),$
 $V(i,j-1) + \sigma(-,T[j]))$

$$i = 0 \text{ 이거나 } j = 0 \text{인 경우는} \\ \text{다음과 같이 초기화 해 준다.} \quad (2)$$

- $V(0,0) = 0$
- $i > 0$ 일 때, $V(i,0) = V(i-1,0) + \sigma(S[i],-)$

- $j > 0$ 일 때, $V(0,j) = V(0,j-1) + \sigma(-,T[j])$

S, T의 전역 최적 정렬은 최적 정렬 값 $V(n,m)$ 로부터 $V(0,0)$ 까지 역추적해서 얻어진다.

I	j	0	1	2	3	4	5	6
0	0	0	-1	-2	-3	-4	-5	-6
1	A	-1	-1	-2	0	-1	-2	-3
2	G	-2	1	1	0	-1	1	0
3	G	-3	0	3	2	1	1	3
4	C	-4	-1	2	2	4	3	2
5	G	-5	-2	1	1	3	6	5

그림 4. 전역정렬을 위한 동적 프로그래밍의 예

그림 4는 전역 최적 정렬을 동적 프로그래밍 방식을 통해서 계산한 예를 보여주고 있다. $V(0,0)$ 은 (2)에서 보여주는 것과 같이 0이 되고 $V(2,2)$ 는 $\text{MAX}(1, -3, 0)$ 이기 때문에 $V(2,2)$ 는 1이 된다. 점수의 계산은 $V(5,6)$ 부터 $V(0,0)$ 까지 값을 역추적하며 올라간다.

위에서 정의한 서열정렬 알고리즘을 코퍼스내의 문장과 입력한 문장을 비교하는 방법으로 사용하였다. 각각의 문장을 S와 T로 보고 그 두 문장에 대하여 동적프로그래밍을 하여 가장 큰 점수를 받는 문장에 입력한 문장과 같은 확률이 높기 때문이다. 즉 입력된 문장의 성분이 코퍼스에 들어있는 문장 성분과의 유사도를 점수로 나타내는 것이다.

문장을 비교 할 때, 음절단위가 아닌 자소단위로 비교했다. 그 이유는 입력 시에 오류가 있어도 필요한 문장을 찾는데 있어서 정확성을 높이기 위해서이다. 우리나라의 글자는 자음과 모음이 합쳐져서 하나의 음절을 이루고 있다. 즉, 그 조합한 자음과 모음 중에 하나만 틀려도 전혀 다른 음절이 되고 만다. 문장을 음절단위로 비교를 하면 자음이나 모음에 오류가 없어야 정확한 비교가 이루어진다. 자소단위 분해하여 프로그래밍을 하면 음절에 오류가 있어도 음절을 이루고 있는 주변 자소의 비교를 통해 점수를 보상받을 수 있게 된다. 즉, 어느 정도의 오류를 포함하고 있는 입력이 들어와도 사용자가 원하는 문장을 찾을 수 있는 정확도가 높아진다. 자소단위의 또 하나의 장점은 완벽한 음절이나 어절, 문장이 입력되지 않아도 입력된 문자열을 가지고 문장을 찾을 수 있다는 점이다. 초성으로도 문장에 대한 검색이 가능해진다. 이런 서열정렬을 이용한 문장검색의 장점들은 문장을 입력하는 시간을 줄여주고 이동 중의 입력 시에 발생 할 수 있는 오타에 관해서도 영향을 덜 받을 수 있다.

4. 실험 및 결과

이 장에서는 지금까지 제안한 내용을 간단하게 시뮬레이션 할 수 있는 모바일 어플리케이션을 구현한다. 구현 환경은 안드로이드 OS를 기반으로 하는 스마트 폰이며 각종 데이터는 SQLite3을 이용하여 스마트 폰에 구성하였다[10].

4.1 이미지 버튼을 이용한 문장 생성

문장 생성 과정은 화면에 제시된 아이콘을 선택하면 어휘사전, 하위범주화 사전, 동사사전, Google Search API, JSON등을 이용하여 자동적으로 적절한 문장을 제시해주는 과정이다. 모바일 단말은 화면 크기가 제한되므로, 많은 명사들이 한 번에 제시될 수 없다. 따라서 명사들을 개념 별로 그룹화한 후 그룹의 아이콘을 그림 5와 같이 초기 화면에 제시하도록 하였다. 먼저 상위개념의 아이콘을 선택하면 그림 6과 같은 하위명사들이 제시되며 이를 선택하여 명사를 결정한다. 명사는 필요한 개수만큼 반복하여 입력할 수 있다. 그 다음 동사선택버튼을 누르면 그림 7과 같이 입력된 명사와 관련된 동사들이 제시된다. 동사를 선택하면 하위범주화 사전을 통해 각 명사의 격 정보, 즉 주어 또는 목적어 등을 선택제약 (Selectional Restriction) 알고리즘을 통해 결정한다.



그림 5. 초기 화면



그림 6. 후보 명사 범주 화면

실제 고유명사 같은 어휘들은 그 수가 너무 많으므로 어휘 사전에 모두 등록시키는 것이 불가능하여 언어처리 시스템의 큰 난점이 되고 있다. 본 실험은 모바일 환경에서 진행하였으므로 GPS와 네트워크 접속이라는 장점이 있다. 따라서 현 위치와 인터넷 검색을 활용하여 자동적인 고유명사 추천이 가능하도록 하였다. 그림 8은 Goggle Search API를 주변검색을 하여 자신의 위치를 중심으로 가장 가까운 은행을 순서대로 제시하는 예이다. 제시하는 어휘 수는 별도로 설정할 수 있도록 하였다. 너무 많은 후보를 제시하면 상대적으로 먼 곳까지 출력되기 때문에 적절한 개수로 제한할 필요가 있다.

그림 9는 최종적인 문장 생성 결과이다. 기본명사인 '은행'을 사용자가 선택한 '푸른신협'으로 대체하여 문장을 완성한 것이다. 이때 '은/는/을/를' 등 조사와 같은 문법형태소 생성이 필요하므로, 선행 형태소의 중성 유형과 하위범주화 사전을 활용하여 구현하였다.



그림 7. 동사 선택 화면



그림 8. Goggle을 통한 주변 장소 추천 화면

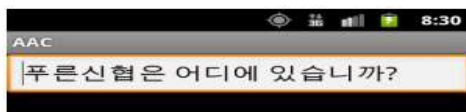


그림 9. 문장 생성 화면

4.2 텍스트 입력을 통한 문장생성

이 방법은 가상키보드로 입력한 문장을 코퍼스에 저장되어 있는 원시문장과 서열정렬을 이용하여 문장을 비교하고 고유명사 사진을 이용하여 추상화되어 있는 명사를 치환시켜주는 방법을 실험한 것이다. 각각의 비교점수를 내림차순으로 정렬하여 그 중 가장 높은 비교점수를 얻은 상위 3개의 문장을 사용자가 선택하여 출력하도록 설계 하였다. 문장을 선택 후 치환할 명사가 있으면 고유명사사전에서 명사를 찾아서 보여주는 모습이다. 이 부분은 API를 이용하여 주변을 검색한 결과가 아니라 사용자가 자주 사용하는 고유명사를 넣어서 사용자에게 맞게 DB를 구성 할 수 있도록 설계 하였다. 그림 10은 고유명사를 선택 후 일반명사와 치환하여 최종적으로 생성한 문장을 화면에 보여주는 모습이다.

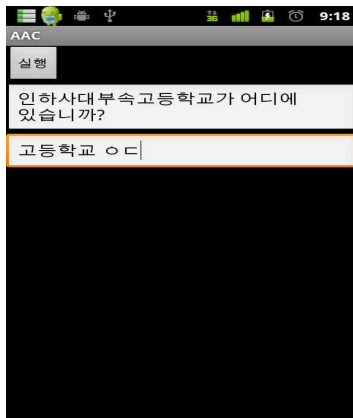


그림 10. 생성된 문장 출력 화면

본 논문에 사용한 이미지버튼을 이용한 방법에서는 최소 5번에서 최대 7번의 버튼 입력 시 문장을 생성할 수 있다. 문장을 생성 할 때 키보드를 이용하여 문장을 생성하는 것 보다 적게는 8번, 많게는 28번 정도 적게 키를 눌러도 같은 문장을 만들 수 있었다. 서열정렬 알고리즘을 이용한 문장생성 방법은 문장을 생성 시 원문의 30%정도의 텍스트만 입력을 하여도 충분히 우선순위가 3위안에 드는 것을 확인하였다. 중간에 섞인 오타가 있어도 큰 영향을 받지 않고 문장을 찾아주는 것을 보았으며 원시문장을 기본으로 하여 생성하기 때문에 이미지버튼을 이용한 방법보다 좀 더 자연스러운 문장을 만들 수 있다. 하지만 코퍼스의 문장이 많지 않으면 생성 할 수 있는 문장의 개수가 제한적이다. 코퍼스에 필요한 문장이 있어야 자신에게 맞게 생성을 하고 출력을 할 수 있기 때문이다.

5. 결론

본 논문에서는 기존에 연구되었던 언어 장애인을 위한 보완·대체 의사소통기술을 현대의 모바일기기에 맞게 설계하고 새로운 문장생성 방법을 제안 하였다. 특히, 과거 Window 기반의 태블릿 PC나 PDA에 없었던 GPS 및 무선네트워크 기술을 활용하여 주변에 있는 고유명사 검색하고 처리하여 한계가 있던 기존의 고유명사처리 방법에 대하여 새로운 접근 방법을 보였다.

문장의 생성방법을 한 가지 방법으로만 국한하여 하지 않고 이미지버튼을 이용한 방법과 텍스트 입력을 통한 방법 두 가지를 동시에 제공해 문장생성의 다양성을 주었다. 이미지버튼을 이용한 방법은 적은 입력키로도 문장을 생성 할 수 있는 장점이 있지만 API를 통한 고유명사 처리 시 원치 않는 결과가 같이 검색되어 보여주는 오류가 있음을 보였다. 텍스트 입력방식은 초성 및 불완전 문장을 입력해도 문장을 생성할 수 있어 오류에 강한 모습을 보이지만 코퍼스에 저장되어 있는 문장에 개수에 따라 생성범위가 정해지는 단점을 보이고 있다. 본 시스템은 현재의 사회에서 언어장애인과 일반인과의 의사소통에 도움을 줄 것이라고 기대한다.

참 고 문 헌

[1] 정해동 외 3인, “장애학생을 위한 보완·대체 의사소통지도”, 국립특수교육원, 1999.
 [2] 이은실, “제한된 공간내에서 동사예측기법을 이

용한 언어장애이용 문장발생장치의 구현”, 인하대학교 박사학위 논문, 2001.

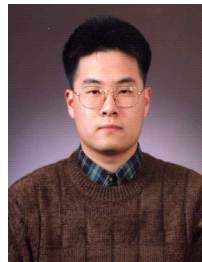
- [3] 황인정, 우요섭, 민홍기, “문장생성에 의한 통신 보조시스템의 설계 및 구현”. 멀티미디어학회논문지 제8권, 제9호, 2005, pp. 1248-1257.
- [4] 추교남, 우요섭, 민홍기, “언어 장애인의 언어보조 시스템을 위한 아이콘 언어의 구축”, 한국정보처리학회논문지B 제13-B권 제4호, 2006, pp. 479-488.
- [5] 홍재성, “21세기 세종계획 전자사전개발”. 국립국어연구원 보고서, 2006.
- [6] 추교남, “개념 기반 정보 검색을 위한 한국어 어휘의 의미 분석”, 인천대학교, 석사학위논문, 1999.
- [7] K. M. Chao, L. Zhang. “Sequence Comparison Theory and Methods”, Springer, 2009.
- [8] M. Tompa, “Lecture Notes on Biological Sequence Analysis”. Tech. Rep, Department of Computer Science and Engineering University of Washington, 2000.
- [9] S. B. Needleman, C. D. Wunsch. “A General Method Applicable to Proteins”. J. Mol. Biol. 48, 1970, pp. 443-453.
- [10] 최정환. 이동 스마트 기기에 적용 가능한 보완·대체 언어 시스템의 구현, 인천대학교 석사학위 논문. 2012.



최 정 환

2010년 2월 인천대학교 정보통신공학과 졸업 (학사)
 2012년 2월 인천대학교 정보통신공학과 졸업 (석사)
 2012년 6월 현재 한국전력 KDN 근무

관심분야 : 자연언어처리, 인공지능, 모바일 프로그래밍



추 교 남

1997년 2월 인천대학교 정보통신공학과(학사)
 1999년 2월 인천대학교 대학원 정보통신공학과(석사)
 2007년 8월 인천대학교 대학원 정보통신공학과(박사)

2009년~현재 인천대학교 정보전산원 초빙교수
 관심분야 : 자연언어처리, 인공지능, 모바일 프로그래밍, 빅데이터 마이닝



우 요 섭

1986년 한양대학교 전자통신공학과졸업(학사)
 1988년 한양대학교 대학원 전자통신공학과 졸업(공학석사)

1992년 한양대학교 대학원 전자통신공학과 졸업(공학박사)

1992년~현재 인천대학교 정보통신공학과 교수
 관심분야 : 자연언어처리, 데이터베이스, 네트워크 프로그래밍