

어휘의미패턴을 이용한 음성인식 오류 검출 및 수정

윤용욱⁰ 정한민* 이근배*
포항공과대학교 정보통신학과, 컴퓨터공학과*
{ywoon⁰, jhm, gblee}@postech.ac.kr

Error detection and correction in speech recognition by using lexico-semantic patterns

YongWook Yoon⁰
Graduate School of Information
Technology, POSTECH

HanMin Jung*, Gary Geunbae Lee*
*Dept. of Computer Science and Engineering,
POSTECH

요 약

음성인식기를 거친 결과는 오류를 포함할 수 있으며 이를 다른 자연어처리 응용에 이용하기 위해서는 오류의 검출과 수정과정이 필수적이다. 음성인식 오류 후처리는 그 성격상 문자인식 후처리와는 다른 접근 방법을 필요로 하며, 본 연구에서는 잡음환경을 제외한 특정 도메인에 국한된 음성발화 상황에 초점을 맞추고자 한다. 후처리 방법에 있어서는 통계적 접근과 패턴매칭에 의한 접근 방법이 있으며, 본 연구에서는 특정 도메인에서 사용되는 어휘의 의미정보를 포함하는 패턴을 자동으로 생성시켜 이에 의한 오류 검출 및 수정 방안을 제안한다. 본 실험에 사용된 도메인은 차량정보센터용 음성정보 제공 시나리오이며 상용 음성인식기를 후처리를 위한 개발 툴로 사용하였다.

1. 서론

기계에 의한 음성인식의 결과를 실생활의 여러 분야에 응용하는 사례가 나날이 늘어가고 있다. 예를 들어, 목소리를 가지고 특정인을 확인한다든지, 국제전화로 걸 때 송화자의 목소리를 인식하여 수화자의 국어로 통역할 때 사용한다든지, 가전제품을 동작시킬 때 사용자의 음성명령에 따라 구동시킨다든지 여러 사례를 생각해 볼 수 있다. 하지만 음성이 발화되는 환경은 보통 매우 열악하여(자동차 소음, 군중의 소음, 또는 여러 음원이 섞이거나 반사되는 등), 기계가 정확히 내용을 인식하여 그 결과를 출력한다는 것은 대단히 어려운 일이다.

따라서 지금까지 구현된 음성인식기의 출력은 어느 정도 오류를 포함하고 있으며, 이를 입력으로 하여 위에 열거한 응용에 사용하기 위해서는 오류를 검출하고 원문에 맞게 이를 수정하는 작업이 필요불가결하다.

인쇄체나 필기체등의 문자(character)인식 결

과의 후처리에 관하여는 많은 연구가 현재도 진행 중이며, 어느 정도 성과를 거두고 있다[10]. 그러나, 음성인식 오류의 후처리는 신호특성의 근본적인 차이, 학습과정의 차이등으로 말미암아 문자인식의 후처리와는 다른 접근방법이 필요하다.

인식기가 인식결과를 1-best로 출력하느냐 n-best로 출력하는가의 문제도 후처리 방법에 차이를 가져올 수 있다. 본 연구에서는 상용 음성인식기를 가지고 실험을 하였으며, 인식 결과로 1-best만을 출력하고 있다. 이는 상용인식기가 중간 개발 도구가 아닌 일반 사용자를 상대로 최종 결과를 출력하기 때문이라고 생각된다. 하지만 1-best출력에 대한 후처리의 장점은 인식기의 종류에 구애받지 않으므로 후처리의 모듈화가 가능하다는 점이다.

사용자 발화환경도 인식기의 성능을 좌우하는 중요한 요소이다. 다양한 발화환경이 존재한다. 자동차, 군중, 또는 채널의 소음등 인식대상이 아닌 잡음이 섞이는 경우를 생각해 볼 수 있다. 또한 특정 화자의 음성과 대화 내용에 따라 인식결과가 달라진다. 이 때문에 인식오류가 다양하며 예측 불가능성을 띠게 된다. 이러한 모든 요소를

다 감안하여 정확한 인식결과를 내는 인식기는 현재로서는 존재하지 않으며, 또한 그에 따른 후처리 방법에 대한 연구는 지난한 과제가 될 것이다.

따라서 본 연구에서는 정숙한 환경에서 특정 도메인에 국한된 문장이 발화된 경우로 인식환경을 제한하였으며, 구체적으로는 자동차용 정보센터에서 음성에 의한 정보제공 시나리오를 인식용 script로 사용하여 실험을 진행하였다. 하지만 후처리과정에서 응용 도메인 독립적 처리를 목표로 다양한 응용환경에 적용할 수 있는 방안을 제시하고자 한다.

2장에서는 지금까지의 국내의 음성인식 오류 검출/수정에 관한 연구결과를 소개하고, 3장에서는 본 연구의 주 테마인 어휘의미패턴을 이용한 새로운 후처리 방안을 구체적으로 소개한다. 4장에서는 차량정보센터 도메인에서 상용 음성인식기를 가지고 오류 후처리 실험을 수행하고 그 결과를 분석하며, 5장에서는 문제점과 향후 과제에 대해 검토한다.

2. 관련연구

2.1 Noisy channel model

Ringger et al.[1]은 음성인식기가 학습한 환경과 실제 사용되어지는 환경에 차이가 있음을 인식하고 이로 인해 발생하는 인식오류를 Noisy Channel Model로 해석하였다. 원래의 문장이 noisy channel을 거치면서 오류가 발생되나 이러한 채널의 특성을 잘 모델링하면 인식기나 사용환경에 독립적인 강건한 후처리 모듈을 따로 구현할 수 있다는 의미이다.

원래 문장의 word sequence를 W , 인식기를 거친후 오류를 포함한 문장의 word sequence를 W' 라 하면, 오류의 수정은 확률값 $P[W'|W]$ 를 최대로 하는 W 를 구하는 작업이 된다. 여기서,

- $P[W]$ 는 사용자가 W 를 발화할 확률로서 language model을 나타내고,
- $P[W'|W]$ 는 W 가 발화되었을 때 인식기가 W' 으로 출력할 확률을 가리키며 곧 error channel model이 된다.

$P[W'|W]$ 를 구하기 위하여 일대일 단어대치 가 일어날 경우 단어간 independence를 가정하고,

$$P[W'|W] = \prod_i P[w'_i | w_i]$$

로 계산하게 된다

이 연구에서는 [2]에서 제안한 language model을 결합하고 음성인식 오류의 후처리 과정을 체계적으로 모델링함으로써 확률모델에 의한 오류수정 방법을 제안했다는 데에 의미가 있다. 특히 인식기에 독립적인 오류 후처리 방안은 실제 응용에 적용성을 높이고 있다.

그러나, 기본적으로 1:1 단어간 에러확률에 기반함으로써 1:n이나 n:1 대치오류에 취약점을 갖고 있다(3.1 참조). 그래서 [1]에서는 단어의 fertility란 개념을 도입하여, 한 단어가 k개의 단어로 오 인식될 경우의 fertility를 k라 정의하고 그에 따라 확률을 재계산 하는 방법을 취하였으나 별다른 성능 향상은 보여주지 못하였다.

2.2 Error Pattern Matching

Satoshi Kaki[3]는 음성인식 결과로 나온 word string을 가지고서 error pattern matching에 기반한 후처리방안을 제안하고 있다. 이 방법은 에러는 일정한 유형을 가지고 발생하며 random한 에러 유형은 거의 없다는 가정하에 출발한다. 그래서 미리 error pattern을 모아서 database에 저장해 놓고 이를 이용한다.

처리는 크게 두 단계로 나뉘는데 첫 단계는 database에 저장된 error pattern을 인식 결과 string과 비교하여 error pattern과 일치할 경우 무조건 error로 판단하고 database에 쌍으로 저장된 correct string으로 치환한다. 여기서 미처리된 오류는 다음 단계로 넘어간다.

두번째 단계에서는 별도의 에러 검출 방법을 이용하여 일단 error string을 집어 내고 이의 수정을 위하여 일종의 corpus(string-database)를 이용한다. 구체적으로 설명하면 error string과 좌우 단어 문맥을 결합하고 이를 key로 string-database를 검색하여 이와 유사한 string이 있을 경우 이 string을 error string과 치환한다.

이 방법은 직관적이며 응용 도메인이 작을 경우에도 효율적인 에러 수정을 기대할 수 있다. 또한 에러 검출 및 수정이 단어군 즉 string단위로 이루어지므로 3.1에서 언급할 n:m 대치오류를 잘 처리할 수 있는 장점이 있다. 그러나 에러수정이 단순하게 이뤄지므로 올바른 단어가 에러로 오인될 가능성도 함께 내재되어 있다. 또한 [4]에서 밝혀진 대로 대부분의 음성인식기는 약간의 도메인 모델의 변화만 있어도 그 성능차이가 심해짐을 감안 할 때 화자나 응용 도메인의 변경에 적응하기 위해 더 많은 error pattern의 수집에 의존하여야 하며 이는 매우 많은 비용이 들어갈 것으로 예상된다

2.3 Error Pattern의 통계적 적용

김용현[5]의 방법은 인식된 결과에서 string 단위로 error pattern을 검출/수정하는 것은 [3]과 유사하나 pattern matching을 이용한 직접적인 방법이 아닌, error를 모아 놓은 database로부터 error pattern 확률을 산출하고 이를 여러 문맥 bigram정보와 결합한 통계적 방법을 사용한 점이 다르다.

여러 수정 후보는 다음과 같이 생성한다. 베이 스타인 음성인식기로부터 n-best문장 후보를 받아 이를 생성 가능한 단어군들의 lattice구조로 만든 후, 각 단어군의 pair마다 원본-오류 단어쌍의 확률을 계산한다. 이때 error pattern의 좌우 문맥도 같이 고려한다. 이 상대적 확률값이 가장 큰 pair에 대해 이를 오류 단어로 결정하고 해당 원문 단어로 복원하게 된다.

이 모델의 장점은 [3]에 비해 확률적 접근을 하므로 맞는 단어를 틀린 단어로 오인할 가능성은 적어진다. 그러나, 오류 검출에 있어 n-best문장 내 단어들중 후보가 없으면 검출이 아예 불가능하다. 여러 오류 후보중 하나를 결정하는 방법에 있어서도 두 단어군간의 상대적인 확률이 제일 높다고 해서 그것이 에러일 확률이 가장 높다는 보장은 없다. 충분한 양의 데이터가 없을 경우 에러패턴 학습결과가 민감하게 반영될 가능성이 높다. 또한 여전히 [3]에서와 같이 여러 학습시 사용된 도메인에 의존하므로 강건하지 못하다.

3. 의미정보를 이용한 에러 수정

3.1 오류 유형

보통 음성인식에서 발생하는 오류의 유형은 삽입(insertion), 삭제(deletion), 대체(replacement)의 3가지로 나누어 볼수 있다.

1. 삭제

원 문장에 있는 단어가 인식시 삭제된 경우 (예)

원문: 편의점 *이* 있는 주유소는

오류: 편의점 있는 주유소는

2. 삽입

원 문장에 없던 단어가 삽입된 경우 (예)

원문: 삼풍주유소의 위치는

오류: 삼풍주유소의 위치는 *이*

3. 대체

원래의 단어가 다른 단어로 바뀐 경우 (예 1)

원문: 세차 가능하고 편의점이 있는 ...

오류: *사찰* 가능하고 편의점이 있는 ...

(예 2)

원문: 영등포역 주변의 주유소를 찾아줘

오류: 영등포역 주변의 *스캐폴* 소를 찾아줘 (예 3)

원문: 저렴한 가격에 기름도 넣고 세차도...

오류: 저렴 *하계계* 기름도 넣고 세차도...

그 중에서도 대체 오류가 가장 빈번히 나타나며 본 실험에서는 전체 오류의 80%정도를 차지하였다(4장 참조). 대체된 단어의 내용을 분석해 보면 음향학적으로 유사한 단어가 선택될 확률이 높은 관계로 오인식이 되었을 경우 구문적, 의미적 유사성은 찾아보기 어려우며 예측 불가능한 경우가 많다. 또한 단어의 대체시 일대일 관계로 발생한 경우 모델링이 비교적 단순하나, 예 2 혹은 예 3에서처럼 n:1 또는 1:m의 관계로 원 단어와 오류 단어가 발생할 경우 복원을 하기 위해서는 먼저 문장의 정렬을 통하여 오 인식 단어의 대응관계를 찾아야 하며, 더군다나 1:1로 발생한 경우와 달리 오류의 유형이 훨씬 많아지므로 복원시 그만큼 어려움이 따른다.

3.2 시스템 구조

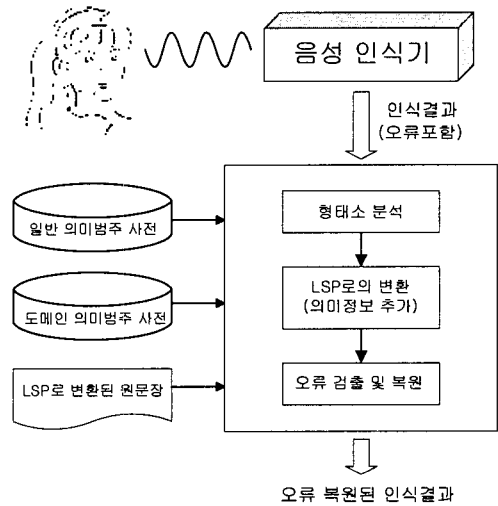


그림 1. 음성인식 오류 후처리 과정

위 그림 1에서 시스템의 대체적인 구조를 표시하였다. 먼저 사용자가 발화한 음성이 음성인식기를 거쳐서 텍스트 형태의 출력을 내게 되는데 이는 많은 오류를 포함하고 있다. 이를 후처리 과정에서 오류를 검출 및 수정하여 뒤에 이어지는 응용(예를 들어 정보검색, 질의응답, 기계번역등)

에서 보다 정확한 결과를 얻는 데 도움을 줄 있도록 하는 것이 후처리의 목적이다. 본 연구에서 제안하는 처리 과정은 특정 음성인식기에 종속되지 않고 독립적인 적용이 가능하며 또한 특정 응용 도메인에 적합성을 높이기 위하여 도메인 이식성을 높일 수 있도록 설계되었다.

3.3 어휘의미패턴

어휘의미패턴(LSP: Lexico-Semantic Pattern)은 일찌기 open-domain 질의응답시스템에서 사용자 질의를 미리 정의된 query type에 matching하고자 사용되었다[6][7]. Kim[8]은 사용자 질의를 입력으로 받아 이를 어휘, 품사, 유의어의 sequence로 변환하여 매칭에 사용하였다. 정한민[9]은 추가로 의미정보와 시소러스의 의미코드를 반영하여 어휘의 개념적 표현과 정보의 세밀한 기술을 가능하게 하였다. 본 연구에서는 음성인식용 시나리오와 출력 결과 문장을 어휘의미패턴으로 변환하여 정답문장과 출력결과를 비교하는 데 쓰인다.

본 실험에 사용된 입력 문장을 어휘의미패턴으로 변환한 예를 보이면 아래와 같다.

원문: 이용할 수 있는 가까운...
LSP: %action|xsp|ef|%수|@요구pa|ef|@길이

원문: 편의점 근방에 신촌역으로부터 가장 가까운...
LSP: %shop|%here|j|@subway|%from|%most-near|ef

‘%action’이나 ‘%shop’과 같이 ‘%’로 시작하는 것은 유의어를 가리키고, ‘@길이’나 ‘@subway’와 같이 ‘@’로 시작하는 것은 의미 범주 사전으로부터 가져온 의미정보이다. 또 xsp, ef, j 등은 각각 접미사, 어미, 조사를 가리키는 품사정보이다. 이와 같이 어휘의미패턴을 사용하면 어휘를 의미정보를 포함한 스트링으로 변환함으로써 보다 개념적이고 일반화된 표현을 할 수 있다. 아울러 어휘의미패턴 matching은 현재의 언어 분석 중심의 구문/의미 분석의 기술적 한계를 극복하고 실용적인 Finite State Automata(FSA) 수준의 구문/의미 처리를 채택한 것으로 보인 된다.

3.4 입력 문장에 대한 어휘의미패턴 구축

그림 1에서 음성을 입력받고 인식기를 거친후 후처리가 이루어지는 과정은 실시간으로 이루어진다. 맨 마지막 단계에서 어휘의미패턴의 비교에 의한 처리를 위해서는 입력 시나리오 대한 정답 문장의 어휘의미패턴이 off-line으로 구축돼 있어

야 한다. 어휘의미패턴을 구축하기 위해 1,011개의 시나리오 문장을 대상으로 윈도우 크기를 6으로 정한 후에 문장을 스캐닝해나가면서 패턴들을 수집한다. 총 42,385개의 패턴을 시나리오 문장들로부터 추출하였는데, 그 예는 그림 2와 같다.

%action %action * xsp ef %name	ncp
%action %action ncp * ef %name	xsp
%action %action ncp xsp * * name	ef
%action %action ncp xsp ef * * name	*
%action %action ncp xsp ef *	%name
%action %tol %action * ef %수	xsp
%action %tol %action xsp ef *	%수
%action %tol * xsp ef %수	%action
%action * %action xsp ef %수	%tol j
%action * %together pv ef %수	j
uknc j * ep ep @장소	@요구pv
uknc j @subway %near %from *	pv
uknc j @subway %near %from pv	*

그림 2. 입력문장에 대한 오류 검증/추정 패턴 구축

그림 2에서 각각의 라인은 크게 왼쪽 부분과 오른쪽 부분으로 나뉘어 진다. 왼쪽은 6개의 어휘의미패턴의 구성요소, 즉, 어휘, 품사, 또는 의미 범주로 구성되며, 오른쪽은 1개의 어휘의미패턴 구성 요소 또는 “*” (don't care)로 구성된다. 오른쪽 부분 “*”에 대응하는 왼쪽 어휘의미패턴 sequence는 시나리오 문장으로부터 추출한 정답 패턴을 의미하며, 오른쪽 부분이 임의의 어휘의미패턴 구성 요소인 경우 왼쪽 어휘의미패턴 sequence는 오류 추정 패턴을 의미한다. 오류 추정 패턴은 입력 문장에서 “*”에 해당하는 부분을 추정할 때 사용한다. 현재 위치에서 하나의 패턴을 추출한 후에 그 패턴내에서 위치를 옮겨가면서 “*”를 삽입하고, 그 위치에 해당하는 구성 요소를 오른쪽 부분에 두는 방식으로 패턴을 구축한다. 다음은 이 구축 방법에 대한 예이다.

(원문)
“세차장과 편의점이 함께 있는 ...”

(어휘의미패턴: 오류 검증 패턴)
%wash|%and|%shop|jcs|%together|@요구pa *

(오류 추정 패턴)
*|%and|%shop|jcs|%together|@요구pa %wash
%wash|*|%shop|jcs|%together|@요구pa %and
%wash|%and|*|jcs|%together|@요구pa %shop
%wash|%and|%shop|*|%together|@요구pa %jcs
%wash|%and|%shop|jcs|*|@요구pa %together
%wash|%and|%shop|jcs|%together|* @요구pa

만일 입력 문장에서 현재 문맥이 오류 검증 패턴과 매칭되면, 다음 윈도우로 넘어가며, 그렇지 않은 경우에는 오류 추정 패턴들을 이용하여 현재 윈도우 내에서 각 단어 위치를 옮겨가면서 6 종류의 오류 추정 패턴을 적용한다. 이러한 과정을 모든 윈도우에 대해 적용하면, 아래에서 설명하는 통계적인 오류 수정을 할 수 있게 된다.

3.5 어휘의미패턴에 기반한 통계적인 오류 수정

우선 어휘의미패턴을 이용한 작업전에 n:m 대치오류를 걸러내는 전처리 과정을 거친다. 빈번히 발생하는 n:m 오류에 대해 (오류:원형)의 쌍으로 모은 error pattern database를 미리 구축하고, 후처리 대상 문장에 이런 오류 단어가 발견되면 원래 단어열로 치환한다. 이는 Satoshi[3]의 error pattern 수정방법과 유사하다. 아래에 이러한 오류 수정의 예를 보인다.

<오류패턴> -> <복원패턴>
 "것으로 수 있는" -> "가스를 넣을 수 있는"
 "기름도들 수" -> "기름도 넣을 수"
 "길 소 중" -> "주유소 중"
 "김 짝뚱한" -> "기름값이 싼"
 "를 올 수 있는" -> "를 넣을 수 있는"
 "사자상과" -> "세차장과"
 "사전 선거" -> "세차장과"
 "세차고를 갈고 할" -> "세차하고 오일을 갈"
 "서울 교화하는" -> "오일교환을 할 수 있는"

다음 단계는 LSP를 이용한 일반적인 오류 수정 과정이다. 이 알고리즘은 오류 추정/수정 후보의 빈도와 어휘의미패턴의 구성 요소의 레벨에 따라 이루어진다. 빈도에 의한 우선 순위는 유일 수정 규칙이 다수 공통 수정 규칙에 우선하며, 즉, 음성 인식 오류 추정 위치가 동일한 곳에서만 일어나는 것을 최우선으로 하며, 두 곳 이상에서 오류가 추정되는 경우에는 빈도수가 높은 위치를 선택한다. 이 우선 순위는 수정에도 동일하게 적용된다. 수정될 수 있는 대상이 하나인 경우에는 그것을 선택하며, 여러 개인 경우에는 대상의 빈도에 의해 결정한다. 어휘의미패턴의 구성 요소로는 의미 범주 태그, 품사, 어휘가 있으며, 우선 순위는 의미 범주 태그 > 품사 > 어휘 순으로 정한다. 빈도가 동일한 오류 추정/수정 후보가 두 개 이상 존재하는 경우에는 어휘의미패턴 레벨로 수정 대상을 선정한다. 그림 3은 빈도와 레벨 정보를 이용한 음성 인식 오류 수정 예이다.

4. 실험 및 분석

본 실험에서는 음성인식기로 상용 제품인 ByVoice¹를 사용하였다. 이 제품은 Windows platform에서 실행되는 단독체 연속 음성인식기로서 1-best결과만을 제공하며 Windows의 일반 텍스트 편집기로 음성인식 결과를 출력하는 API를 제공한다. ByVoice는 실행전 화자적응을 위한 30분간의 구술학습이 필요하며, 특정 도메인 언어모델을 반영할 수 있도록 문장에 대한 학습기능이 있다. 실험을 위해 구현한 후처리 시스템은 Linux기반 서버 application이며 client로서 Windows상의 web application을 사용하였다.

음성인식 시나리오로는 차량정보센터용 정보제공 시나리오로서 주유소관련 정보 및 차량 navigation 정보 제공을 위한 사용자 질의문장 1,011문이다. 이 문장을 대상으로 3장에서 기술한 대로 어휘의미패턴을 구축하였다. 또한 오류 패턴 수집을 위하여 남녀 화자가 번갈아 인식실험을 실시하여 여러 패턴을 수집하고 database로 구축하였다.

음성인식결과와 후처리 이전의 baseline을 얻기 위해 위 시나리오에 대한 인식 실험결과는 아래와 같다(화자적응과 도메인 언어모델에 대한 학습을 마친 후 결과임).

표 1. 음성인식 결과 (후처리 전)

Level	# of input	# of incorrect	correctness
Sentence	540	224	58.5 %
Word	4243	341	92.0 %

입력 문장 540개중 107문장을 임의 추출하여 복원결과를 오류유형별로 아래 표 2에 나타낸다.

표 2. 어러 수정 결과 (word 단위)

오류유형	삽입	삭제	대치	계	복원율
Word 수	15	9	95	119	-
어휘복원	8	4	55	67	56.3
의미복원	15	9	71	95	79.8

위 표 2에서 어휘복원이란 lexical level에서 단어단위 복원결과를 말하며, 의미복원이란 후처리 결과를 가지고 다른 용응에 입력으로 사용했을 경우 그 시스템이 오류없이 동작한 경우를 말한다. 실험에서는 차량정보센터 질의응답시스템에 적용한 결과를 제시하였다.

본 오류 수정 알고리즘은 음성 인식 오류에 대한 수정을 위해 사용하는 지식의 범위를 넓힐 수 있어 보다 적용성이 높고, 수정 성능이 좋은 시스템을 만들 수 있게 한다. 기존의 음성 인식 내

¹ 제품에 대한 정보는 <http://www.byvoice.co.kr/> 참조

"세차장과 편의점이 함께 있는 주유소 중에서 가격이 가장 저렴한 곳"

```
# [0] [0->0]      세차장      ncn (1 0)      %wash
# [1] [1->1]      과          j (0 0)        %and
# [2] [2->2]      편의점      uknc (1 0)     uknc
# [3] [3->3]      이          j (0 0)        jcs
# [4] [4->4]      함께        ma (1 0)       %together
# [5] [5->5]      있          pa (1 0)       @요구pa
# [6] [6->6]      는          ef (0 0)       jxc
# [7] [7->7]      주유소      unoun (1 0)    %name
# [8] [8->8]      중          unoun (1 0)    nbn
# [9] [9->9]      에서        j (0 0)        %from
# [10] [10->10]   가격        ncn (1 0)      %price
# [11] [11->11]   이          j (0 0)        jcs
# [12] [12->14]   가장 저렴 하  ncp (1 0)     %most-cheap
# [13] [0->0]      (1 0)
# [14] [0->0]      (0 0)
# [15] [15->15]   ㄴ          ef (0 0)       ef
# [16] [16->16]   곳          nbn (1 0)     %location
```

```
# Speech Recognition Error: %wash|%and|uknc|jcs|%together|@요구pa
# Speech Recognition Hypothesis: %wash|%and|*|jcs|%together|@요구pa → %shop|ncn
```

```
# Speech Recognition Error: %and|uknc|jcs|%together|@요구pa|jxc
# Speech Recognition Hypothesis: %and|*|jcs|%together|@요구pa|j   그림 3. 어휘의미패턴을 이용한 오류 검출 및 수정 과
```

```
# Speech Recognition Error: uknc|jcs|%together|@요구pa|jxc|%name
# Speech Recognition Hypothesis: *|jcs|%together|@요구pa|jxc|%name → %shop|%wash|ncn
```

최다 위치: 편의점
 후보 통계 (빈도): %shop (3), ncn (3), %wash (2), %serv (1)
 LSP 레벨: %shop > ncn

Speech Recognition Correction: uknc → %shop at 2th

```
# shop: 식사, 음식, 편의점      ; %shop에 해당하는 의미범주 사전의 entry
# 최종복원후보 -> 편의점        ; by minimal edit distance from '편의점'
```

서의 음성 오류 후처리가 언어적 지식을 충분히 활용하지 못하였고[3][5], 문자인식 후처리가 어휘 수준에서 단어간 거리 계산을 통한 적용에만 그친 것과 비교할 때 오류 수정을 위한 지식 활용이 의미 범주까지 확장됨을 알 수 있다. 단어간 거리 계산에서는 사전에 있는 단어들과 편집 상에서의 거리 계산에 의존하여 철자 오류 수준을 벗어나지 못함으로 수정 성능의 저하에 영향을 미친다. 본 알고리즘은 오류가 발생한 경우에 인식된 단어가 전혀 다른 철자로 나타나더라도 단어의 음소간 거리 계산이 아닌 의미적 범주의 유사성을 검사함으로써 올바른 의미 범주를 발견할 수 있도

록 해준다.

5. 결론 및 향후과제

음성인식은 발화환경의 다양성으로 말미암아 인간이 인식하는 정도로 범용적인 환경에서 기계가 인식을 하기 위해서는 지속적인 연구가 필요하다. 다만 실생활에 응용하기 위하여 특정 도메인에서의 음성인식에 있어서는 의미정보를 이용한 후처리 방안이 높은 성능을 나타낼 수 있다. 본 연구에서는 방대한 corpus의 구축없이 특정 도메인에 국한된 시나리오를 가정하고 이를 미리 어휘의미패턴으로 구축함으로써 의미정보를 이용하는 후처리방안을 제시한다. 아울러 다른 도메인에

적용시에는 미리 해당 도메인의 인식 시나리오에 대한 어휘의미패턴을 구축함으로써 도메인 변화에 훌륭히 대처할 수 있다. 왜냐하면 어휘의미패턴의 구축은 자동으로 빠른 시간내에 이루어지기 때문에 기존의 [3][5]에서와 같이 새로운 도메인에서의 어휘 패턴 학습에 대량의 비용이 드는 일이 없기 때문이다.

그러나 현재 구축된 어휘의미패턴은 형태소 단위로 변환되었으므로 스트링 window내에 2개 이상의 오류가 발생하거나 단어의 삽입, 삭제가 발생 시에는 처리가 어려우므로 전처리 단계에서 이런 오류 유형을 걸러 내고 있다. 따라서 이런 n-to-m 관계의 오류를 처리할 수 있는 보다 유연한 어휘의미패턴 model의 도입이 필요하며, 또한 어휘의미패턴내의 의미정보를 나타내는 태그('@' 나 '%' 로 시작하는 것)를 POS 태그처럼 해당 단어 열의 feature로 간주하고 이 feature들을 포함하는 corpus를 구축한다면 어휘의미패턴 기반의 통계적 모델을 얻을 수 있으며 예외적인 상황에 유연하게 대처할 수 있다. 추후 연구대상에 이들을 포함한다.

6. 참고 문헌

[1] E.K.Ringger et al., "A Fertility Model for Post Correction of Continuous Speech Recognition" . *ICSLP' 96*, pp.897-900, 1996.

[2] F. Jelinek , "Self-Organized Language Modeling for Speech Recognition" , *Readings in Speech Recognition, Morgan Kaufmann Publishers, Inc., San Mateo, CA*, pp.450-506, 1990.

[3] Satoshi Kaki et al., 98, "A Method for Correcting Speech Recognition Using the Statistical features of Character Co-occurrence" . *COLING-ACL' 98*, p.653-657, 1998.

[4] James F. Allen et al., "A Robust System for Natural Spoken Dialogue" , *Proceedings of the 34th Annual Meeting of the ACL 96*, pp.62-70, 1996.

[5] 김용현, 정민화, "어휘패턴 학습과 후처리 모듈을 이용한 연속 음성 인식의 성능향상" 한국정보과학회 2000 봄 학술발표논문집, pp.441-443, 2000.

[6] Sanda Harabagiu, Dan Moldovan, Marius Pasca, Rada Mihalcea, Mihai Surdeanu, Razvan Bunescu, Roxana Girju, Vasile Rus and Paul Morarescu "The Role of Lexico-Semantic Feedback in Open-Domain Textual Question-Answering" , in *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL-2001)*, Toulouse France, pp.274-281, 2001.

[7] Gary Geunbae Lee, Jungyun Seo, Seungwoo Lee, Hanmin Jung, Bong-Hyun Cho, Changki Lee, ByungKwan Kwak, Jeongwon Cha, Dongseok Kim, JooHui An, Harksoo Kim, Kyungsun Kim." SiteQ: Engineering High Performance QA System Using Lexico-Semantic Pattern Matching and Shallow NLP ." In *Proceedings of the 10th Text Retrieval Conference (TREC-10)*, Washington D.C., Nov. 2001.

[8] Harksoo Kim and Kyungsun Kim and Gary Geunbae Lee and Jungyun Seo. "MAYA: A Fast Question-Answering System Based on a Predictive Answer Indexer." *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01) Workshop on Open-Domain Question Answering*. 2001.

[9] 정한민, 이근배, 최원석, 민경구, 서정연. "관계형 데이터베이스 상에서의 다국어 질의응답 시스템." 제 13회 한글 및 한국어정보처리 학술대회 논문발표집, pp.530-537, 동국대학교 2001년 10월.

[10] Geunbae Lee, Jong-Hyeok Lee, Jinhee Yoo. "Multi-level post-processing for Korean character recognition using morphological analysis and linguistic evaluation." *Pattern recognition, vol 30, no 8*, pp 1347-1360, 1997.