말소리와 음성과학 제1권 제3호 (2009)
pp. 109~115

레벤스타인 거리에 기초한 위치 정확도를 이용한 고립 단어 인식 결과의 비유사 후보 단어 제외

Exclusion of Non-similar Candidates using Positional Accuracy based on Levenstein Distance from N-best Recognition Results of Isolated Word Recognition

윤 영 선[1] · 강 점 자[2]
Yun, Young-Sun · Kang, Jeom-ja

ABSTRACT

Many isolated word recognition systems may generate non-similar words for recognition candidates because they use only acoustic information. In this paper, we investigate several techniques which can exclude non-similar words from N-best candidate words by applying Levenstein distance measure. At first, word distance method based on phone and syllable distances are considered. These methods use just Levenstein distance on phones or double Levenstein distance algorithm on syllables of candidates. Next, word similarity approaches are presented that they use characters' position information of word candidates. Each character‘s position is labeled to inserted, deleted, and correct position after alignment between source and target string. The word similarities are obtained from characters' positional probabilities which mean the frequency ratio of the same characters' observations on the position. From experimental results, we can find that the proposed methods are effective for removing non-similar words without loss of system performance from the N-best recognition candidates of the systems.

Keywords: isolated word recognition, N-best candidate selection, positional accuracy, word similarity

1. 서 론

음성은 사람과 컴퓨터 또는 기계와 가장 직관적이며 편리한 통신수단으로 여겨져 많은 연구가 진행되어 오고 있다. 음성 인식 기술은 대상 어휘와 발성 방법, 잡음 환경에 따라 다양하게 분류된다. 일반적으로 많이 사용되는 음성 인식 방법으로 소용량 고립단어 인식과 대용량 연속 음성 인식을 생각할 수 있다. 소용량 고립단어 인식은 인식 대상 어휘가 작고 단어 단위의 발성을 인식하기 때문에 PDA나 휴대폰, 텔레매틱스 단말기와 같은 소형 정보 단말기에 적재되어 사용된다. 반면 대용량 연속 음성 인식은 인식하려는 어휘의 수가 많고 연속적으로 발성되기 때문에 단말기는 음성을 수집하는 입력 장치 또는 특징 추출 장치로만 사용되고 인식 과정은 고성능 서버에서 수행한다. 최근에는 정보 단말기의 고성능화로 인하여 단말기에서도 대용량 어휘를 이용한 음성 인식을 실행하기도 한다.

소형 단말기의 성능이 높아지고 단말기에서 인식하는 어휘의 크기가 증가하면서, 인식 결과를 하나의 단어로 한정하지 않고 여러 후보 단어를 같이 출력하기도 한다. 특히 키패드 입력에 의하여 목적지나 상호를 입력하는 휴대폰이나 자동차 운행 보조 단말기(네비게이터) 등에서는 음성인식기능을 많이 채용하는 추세이나, 오인식의 가능성 때문에 하나만의 인식 결과만을 표시하는 것이 아니라, 여러 다중 후보들을 같이 제시하고 있다. 목적지나 상호를 입력하는 경우 인식 대상 어휘는 많고, 단어 단위의 발성이 요구되기 때문에, 일반적으로 유사 음소 단위의 음향 모델을 이용하여 음성 인식기를 구현하게 된다. 이 경우 인식 결과로 여러 후보 단어를 같이 출력할 경우 문맥 정

1) 한남대학교 ysyun@hnu.kr, 교신저자
2) 한국전자통신연구원 jjkang@etri.re.kr
(이 논문은 2009학년도 한남대학교 학술연구조성비 지원 및 정보통신연구진흥원의 IT 성장동력기술개발사업의 일환으로 연구되었음.[2009-S-036-01, 신성장동력산업용 대용량/대화형 분산/내장처리 음성인터페이스 기술 개발])

접수일자: 2009년 7월 31일
수정일자: 2009년 9월 11일
게재결정: 2009년 9월 15일

보를 적용하기 어려워 전혀 다른 유형의 단어가 화면에 표시되는 경우가 발생한다[1].

본 논문에서는 이와 같이 다중 후보 열에서 유사도가 다른 후보들에 비해 떨어지는 단어들을 인식 결과 목록에서 제외하여 사용자가 느끼는 심리적인 인식 성능의 향상과 신뢰도를 높이는 방법을 제안한다. 즉, 기존 인식 후보들 중에서 정답(정확하게 인식된 결과)을 최대한 보존하면서 후보 단어들 중에서 유사도가 가장 작은 (다른 후보 단어들과 이질적인) 후보를 제거하는 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에서 사용하는 음성 인식 시스템의 구성을 간단하게 설명하고, 3장에서는 후보 단어들을 비교하는 척도를 소개한다. 4장에서는 제안된 방법을 이용하여 후보 단어 열에서 비유사 단어들을 제외한 결과를 보이며, 마지막으로 결론을 맺는다.

2. 음성인식시스템

본 연구에서 사용한 음성 인식 시스템은 한국전자통신연구원에서 개발한 것으로서 텔레매틱스 시스템에 26만단어의 행선지 입력을 목적으로 개발되었다[2]. 텔레매틱스에 채용된 시스템의 한계로 인한 계산 량의 제약을 극복하기 위하여 2단계 인식 시스템을 채용하였다. 1단계 인식 과정에서는 연속 밀도 HMM(continuous density Hidden Markov Model)의 음향 모델을 변환한 모노폰 반연속 밀도 HMM(monophone semi-continuous HMM)을 사용하였으며, 1단계의 음성 인식 결과를 kNN (k-Nearest Neighbor) 알고리즘을 적용하여 2단계 인식 시스템에서 적용할 대상 어휘를 선정한다. 2단계 인식 시스템에서는 트라이폰(triphone) 반연속 밀도 HMM의 음향모델을 이용하여 축소된 대상 어휘(약 1000여개)에서 최대 10개의 후보 단어를 생성한다. 베이스라인 시스템의 성능은 10-best에 대해서 약 90%의 성능을 보이며, 환경 적응을 고려할 경우 약 93%의 성능을 보인다.

3. 후보 단어의 비교

2장에서 설명한 인식 시스템의 결과로 최대 10개의 단어 후보가 출력되는데, 이 후보 열은 각 음향 모델의 우도(likelihood) 값으로 결정되기 때문에 서로 관련이 없는 후보들이 선택될 수 있다. 본 연구의 목적은 이들 후보 열에서 관련성이 적은 후보 단어를 제외하는 것이기 때문에, 각각의 후보 열들에 대한 비교 방법을 정의하여야 한다.

본 연구에서는 후보 단어들의 비교를 위하여 레벤스타인(Levenstein) 거리척도[3]를 사용하였으며, 음소 및 음절 거리를 이용한 후보 단어 거리 척도를 살펴보고, 위치별 정확도를 이용한 후보 단어 유사도 방법을 제안한다. 제안하는 방법들 모두 레벤스타인 거리 척도를 이용하기 때문에 레벤스타인 거리 척도를 간략히 요약하고 각 방법에 대하여 구체적으로 설명한다.

3.1 레벤스타인 거리 척도

레벤스타인 거리 척도는 두 개의 문자열이 주어질 때, 한 문자열을 다른 문자열로 변환하는 데 필요한 비용을 산출한다.

$$D_{(i,j)} = \min \begin{cases} D(i-1,j-1), & s_i = t_i \\ D(i-1,j-1)+\alpha, & s_i \neq t_i \\ D(i-1,j)+\beta, & insertion \\ D(i,j-1)+\beta, & deletion \end{cases} \quad (1)$$

(1)식에서 s_i와 t_i는 비교 문자열의 위치를 나타내며, α, β는 치환오류와 삽입, 삭제오류 시 추가되는 비용을 의미한다.

레벤스타인 거리 방법은 정보 이론, 기계 번역 등에서 널리 사용되는 방법으로 그 변형도 많이 존재하나, 본 논문에서는 가장 일반적인 방법으로 삽입, 삭제, 치환 오류에 대해 동일한 비용으로 지정하였다($\alpha = \beta = 1$).

<그림1>은 레벤스타인 거리에 의하여 두 문자열 “SORRY”와 “OSORI”를 비교한 결과이다. 두 문자열의 비교 결과 두 문자열의 마지막 문자에서의 누적 거리 값인 ‘3’이 두 문자열의 비교 거리를 나타낸다.

	O	S	O	R	I
S	1	1	2	3	4
O	1	2	1	2	3
R	2	2	2	1	2
R	3	3	3	2	2
Y	4	4	4	3	3

그림 1. 레벤스타인 거리에 의한 두 문자열의 비교
Figure 1. Comparison of two strings using Levenstein distance

레벤스타인 거리에 의해 정렬된 두 문자열은 각 문자의 비교에 따라 삽입, 삭제, 치환 오류로 정의된다. 삽입(insertion) 오류는 원본 문자열에는 없으나 비교 문자열에 존재하는 경우이며, 삭제(deletion) 오류는 원본 문자열에는 존재하나 비교 문자열에 존재하지 않는 경우, 치환(substitution) 오류는 원본 문자열과 비교 문자열의 문자가 다른 경우이다.

3.2 음소 및 음절 거리를 이용한 단어 거리 정의

레벤스타인 거리 척도를 후보 단어의 비교에 이용하는 방법은 후보 단어를 구성하는 음소들을 문자열로 간주하여 레벤스타인 거리를 이용하는 것과, 후보 단어를 음절로 분리하고, 음

절간의 거리를 더하여 단어 거리로 사용하는 방법이 있다.

음소 문자열을 그대로 후보 단어의 거리 계산에 사용하는 방법은 구현하기 쉬우며, 음절 간의 경계를 고려하지 않아도 되는 장점이 있다. 반면에 음절간의 경계를 고려하지 않기 때문에 음절 단위로 출력되는 인식 단어와 약간의 차이를 보이게 된다. 즉, 제거되는 후보 단어가 인식 후보로 남는 후보에 비하여 음절 상으로는 더 유사할 수 있다.

따라서 단어를 구성하는 음절 단위의 동적 정합을 한 후, 음절 거리의 합에 의하여 단어를 비교하는 방법 또한 고려하였다. 음절 단위의 비교는 표시되는 인식 결과와 유사하다는 장점이 있다. 음소 문자열의 비교에 의한 단어 비교는 레벤스타인 거리 계산 방법과 동일하며 음절 단위의 비교에 의한 단어 거리 계산은 식 (2),(3)과 같다.

$$D^{w}_{(i,j)} = \min \begin{cases} D^{w}_{(i-1,j-1)} + D^{syl}_{(N_i,N_j)}, & correct \text{ or } substitution \\ D^{w}_{(i-1,j)} + \beta, & insertion \\ D^{w}_{(i,j-1)} + \beta, & deletion \end{cases} \quad (2)$$

$$D^{syl}_{(k,l)} = \min \begin{cases} D^{syl}_{(k-1,l-1)}, & s_k = t_l \\ D^{syl}_{(k-1,l-1)} + \alpha, & s_k \neq t_l \\ D^{syl}_{(k-1,l)} + \beta, & insertion \\ D^{syl}_{(k,l-1)} + \beta, & deletion \end{cases} \quad (3)$$

위 식에서 N_i와 N_j는 단어 i와 j를 구성하는 음절의 수이며, s_i와 t_i는 비교 문자열의 위치를 나타내며, α, β는 치환오류와 삽입, 삭제오류 시 추가되는 비용이다.

3.3 비교 위치에 따른 유사도 정의

다음으로 제안하는 후보 단어들의 비교 척도는 비교 위치에 따른 유사도이다. 레벤스타인 거리를 이용하여 두 문자열을 비교한 경우, 비교 위치에 따라 비교 문자열에 대하여 삽입(I), 정위치(C), 삭제(D)의 위치 속성을 표기한다. 정위치 표시는 비교 결과 맞음(correct)과 치환 오류가 발생할 수 있으나 비교 위치는 정확하다는 것을 의미한다.

원본 문자열과 비교 문자열의 비교 시, 비교 경로에 따라 삽입과 삭제, 정위치의 특성이 달라질 수 있다(문자열의 정렬이 달라진다)[4]. 이 경우 각 오류 특성에 따른 추가되는 비용을 다르게 한다면, 좀 더 정확한 문자열 정렬을 얻을 수 있을 것이다. 현재는 각 오류에 대해 동일한 비용으로 유사도를 정의하였으며, 직관적인 방법에 의하여 정렬 처리를 하였다. (식 (1)의 $\alpha = \beta = 1$).

<그림2>는 원본 문자열과 비교 문자열이 주어지면 비교 문자열의 각 문자에 대하여 위치 속성을 표기한 예를 보인다.

```
원본문자열 : -  S  O  R  R  Y
비교문자열 : O  S  O  R  -  I

누적 거리 :  1  1  1  1  2  3
문자 속성 :  I  C  C  C  D  S
```

그림 2. 레벤스타인 거리에 의한 문자열 정렬
Figure 2. Alignment of strings by Levenstein distance

문자열을 비교하기 위해서는 원본과 비교 문자열이 필요하나, 음성 인식의 경우 원본을 정확히 알 수 없다는 제약이 따른다. 따라서 본 논문에서는 10-Best 후보 열중에서 임의의 단어를 원본 문자열로 지정하고 나머지 후보들을 비교 문자열로 선정한다. 비교 문자열은 원본 문자열이 달라지면 기준 위치가 달라지기 때문에, 기준위치를 원본 문자열의 위치와 원본 문자열에 대한 삽입 위치로 선정한다. 이 과정을 모든 후보 단어에 대하여 적용한 후, 그 평균을 각 후보 단어의 유사도로 정의한다.

원본:		S		O		R		R		Y	
비교:	O	S	-	O	-	R	-		-	I	
속성:	I	C		C		C		D		S	

그림 3. 문자열 비교와 문자별 위치 속성
Figure 3. Comparison of strings and positional tag per character

<그림3>에서 보는 바와 같이 원본 문자열의 길이가 5라면, 문자간의 삽입 위치를 포함하여 총 11개의 문자열 길이가 후보 단어열의 기준 위치가 된다. 원본 문자열에 대한 삽입 문자는 원본 문자열에는 존재하지 않지만, 비교 문자열에 존재하고, 삭제 문자는 비교 문자열에 존재하지 않기 때문에 정위치와 삽입 오류가 발생한 경우에만 정확도 계산에 사용한다(<그림4> 참조).

후보 문자열	문자열 정렬
SORRY	-S-O-R-R-Y-
OSORI	OS-O-R---I
SCHOOL	-SCH-O-O-L
SOAP	-S-O---A-P

그림 4. 후보 단어와 기준 단어에 대한 문자열 정렬
Figure 4. Candidate words and their alignment to reference word

각 후보 단어를 원본 문자열로 간주하고 나머지를 비교 문자열로 삼아, 모든 후보 문자열들에 대해 위치별 정확도를 계산하나, 특정 후보 단어의 정확도를 설명하기 위해서 편의상 첫 번째 후보(<그림4>의 “SORRY”)를 원본 문자열로 정의하고, 나머지를 비교 문자열로 정의한다.

(1) 원본 문자열의 정확도 계산

원본 문자열의 비교 시 기준 위치가 자기 자신이기 때문에 모든 문자는 비교 위치가 정확하다(문자의 삽입과 삭제가 발생하지 않는다.). 이 경우, 같은 문자 위치에서의 문자열의 빈도수를 정위치에서의 확률이라 정의하고 원본 문자열의 정확도로 정의한다. (<그림5> 참조)

```
S : 4/4=1.00 (C)
O : 3/4=0.75 (C)
R : 2/3=0.66 (C)
R : 1/3=0.33 (C)
Y : 1/4=0.25 (C)
```

그림 5. 첫 번째 후보 단어의 정확도 계산
(C는 정위치를 의미)

Figure 5. Computation of accuracy of best candidate (C represents correct position)

<그림5>에서 보는 바와 같이 “SORRY” 문자열의 경우, 문자열의 기준이 자기 자신이기 때문에 모든 문자는 정확한 위치(correct position)에 있게 되므로, 정위치에서의 정확도를 계산한다. ‘S’문자의 경우 정위치에서 관측된 문자의 수가 4이며, 모두 ‘S’이기 때문에 정위치 정확도는 1.0이 된다. 비슷한 방법으로 ‘O’의 경우 관측된 문자열 중에서 ‘O’가 3개 관측되었기 때문에 0.75의 정확도를 가진다.

(2) 비교 문자열의 정확도 계산

비교 문자열은 원본 문자열과 비교하여 정위치와 삽입 위치로 구분되기 때문에 정위치에서의 정확도는 원본 문자열의 정확도 계산과 동일하다. 다만 삽입 위치에서의 정확도는 원본 문자열에 대한 삽입 위치에서의 출현 확률을 정확도로 정의한다. (<그림6> 참조)

```
O : 1/1=1.00 (I)
S : 4/4=1.00 (C)
O : 3/4=0.75 (C)
R : 2/3=0.66 (C)
I : 1/4=0.25 (C)
```

그림 6. 두 번째 후보 단어의 정확도 계산
(I는 삽입위치를 의미)

Figure 6. Computation of accuracy of second candidate (I represents inserted position)

<그림6>에서 보는 바와 같이 비교 문자열의 경우 원본 문자열과 비교하여 삽입 오류가 발생한다. 원본 문자열이 정답이고 삽입 오류가 발생하면 문자열의 정확도를 낮추는 방향으로 반영되어야 하나, 원본 문자열이 정답이라는 확신이 없기 때문에 삽입된 문자를 후보 단어의 유사도에 반영하여야 한다. 이 점을 고려하여 두 번째 단어의 경우 정위치에서의 정확도는 첫 번째 후보 단어의 경우와 동일한 방법으로 계산하고, 삽입 문자에 대하여 삽입 위치 정확도를 계산한다. ‘O’의 경우 삽입 위치에서 문자가 1번 관측되었으며, 관측된 문자가 ‘O’이기 때문에 삽입 위치에서의 정확도는 1.0이 된다.

위와 같이 각 문자의 특성에 따라 정위치 정확도와 삽입 위치 정확도가 정의되면, 후보 단어의 유사도가 정의되어야 한다. 후보 단어의 유사도는 삽입 위치와 정위치를 어떻게 고려하느냐에 따라 다음 두 가지로 구분할 수 있다.

첫 번째 경우, 후보 단어의 유사도는 원본 문자열과 비교 문자열의 정렬 후, 위치별 정확도를 이용하여 구한다. 모든 후보 단어를 원본 문자열로 지정하여 지정된 후보 단어와 비교한 후, 계산된 유사도의 평균을 현재 후보 단어의 유사도로 정의한다. 즉, 후보 단어 i에 대하여 원본 문자열을 j로 간주하고, 후보 단어 (i,j)쌍에 대한 유사도 $P_{acc}^{(i,j)}$를 계산한다. 이 과정을 모든 후보 문자열에 대하여 적용한 후 확률 평균을 구하면 $P_{acc}^{(i)}$를 계산할 수 있다.

$$P_{acc}^{(i,j)} = \left\{ \prod_{\substack{k=1 \\ k\in \text{correct}}}^{N_j} P_{correct}^{(i,j,k)} \right\}^{1/N_{correct}^{(j)}} \cdot \left\{ \prod_{\substack{k=1, \\ k\in \text{insert}}}^{N_j} P_{insert}^{(i,j,k)} \right\}^{1/N_{insert}^{(j)}}$$
$$P_{acc}^{(i)} = \left\{ \prod_{j=1, j\neq i}^{N_{cand}} P_{acc}(i,j) \right\}^{1/N_{cand}} \quad (4)$$

이 때 $P_{correct}$와 P_{insert}는 정위치 정확도와 삽입위치 정확도를 의미한다. N_j는 원본 문자열 j에 대한 기준 위치의 수를 나타내며, $N_{correct}^{(j)}$와 $N_{insert}^{(j)}$는 원본 문자열 j에 대한 정위치와

삽입위치의 수를 나타낸다. 또한 N_{cand}는 후보 단어의 수를 의미한다. 이 방법을 개별 유사도로 명명한다. 예를 들어 원본 문자열이 첫 번째 단어일 때 두 번째 단어 'OSORI'의 개별 유사도 $P_{acc}(2,1)$는 다음의 값을 갖는다.

$$P_{acc}(2,1) = (1 \cdot 0.75 \cdot 0.66 \cdot 0.25)^{1/4} \cdot (1.0)^{1/1} = 0.593$$

두 번째 경우에는 단일 후보 단어의 유사도를 구하는 과정은 비슷하나, 첫 번째 경우와 달리 위치별 정확도를 먼저 구한 후, 최종적으로 N_{cand}의 후보에 대하여 유사도 계산이 완료될 때, 후보 단어 i에서 발생되는 정위치와 삽입위치 정보를 이용하여 평균을 구한다. 이 방법은 식 (4)의 방법과 달리 전체 후보 단어의 위치별 정보를 이용하여 평균 유사도를 계산하였기 때문에 통합 유사도라 한다.

$$P_{acc}^{(i)} = \left\{ \prod_{j=1, j\neq i}^{N_{cand}} \prod_{\substack{k=1, \\ k\in correct}}^{N_j} P_{correct}^{(i,j,k)} \cdot \prod_{\substack{k=1, \\ k\in insert}}^{N_j} P_{insert}^{(i,j,k)} \right\}^{\frac{1}{N_{correct}+N_{insert}}} \quad (5)$$

식 (5)에서 $N_{correct}$와 N_{insert}는 i번째 후보 단어를 비교 문자열로 가정한 후, 각 후보 단어가 원본 문자열이라고 간주하고 순차적으로 비교하였을 경우에 관측되는 정위치와 삽입위치의 총 수를 나타낸다.

4. 실험 및 결과

제안한 방법의 유효성을 확인하기 위하여 한국전자통신연구원에서 개발한 음성인식시스템의 인식결과를 이용하여 비유사 후보의 검출 실험을 하였다. 제안된 방식을 이용하여 비유사 단어를 검출하고, 음성정보처리를 연구한 전문가에 의뢰하여 지정된 비유사 단어와 비교하였다. 전체 인식결과는 IV (In-Vocabulary) 3,675문장과 OOV(Out-Of-Vocabulary) 1,525문장으로 구성되었다. 이중 비유사 단어 제거 실험을 위하여 IV 3,675 문장에서 Best, 10-best, 오인식의 비율에 따라 선정된 500 문장을 사용하였다. 원래 인식 결과와 비유사 단어 제거를 위해 선택된 500 문장의 기본 인식률은 <표1>과 같다.

표 1. 평가 데이터의 기본 인식률
Table 1. Base recognition rate of evaluation data

전체 문장수	Best	10-Best	오인식
3,675	2,975	3,299	376
500	405	449	51
비율	80.9%	89.8%	10.3%

4.1 실험 자료 및 평가 기준

사람의 경우 정답의 유무에 따라 비유사 단어의 선택이 달라지기 때문에, 전체 평가 데이터에서 추출된 500문장에 대해 정답을 알고 있는 경우와 모르는 경우에 대해서 비유사 단어를 선택하도록 하였다. 각각의 경우에 대하여 비유사 단어를 제외하고 남은 후보 단어들을 대상으로 인식률을 다시 계산하면 <표2>와 같다.

표 2. 정답 제시 유무에 따른 평가 데이터 정보
Table 2. Evaluation data information with or without reference text

정답제시	인식률		비유사 단어
	Best	10-best	
제시	81.0%	89.8%	2,343
미제시	38.4%	44.2%	2.648

<표2>에서 알 수 있듯이 정답(기준 데이터)을 알고 있는 경우에는 정답을 비 유사단어로 전혀 선택하지 않았으며, 정답을 모르는 경우에는 약 40~45%의 정답이 비유사 단어로 선택되어 제거됨을 알 수 있다.

평가 방법은 전문가에 의하여 선택된 비유사 단어의 검출 여부(재현율, recall)로 판단하지 않고, 제안된 방법에 의하여 판단된 비유사 단어가 전문가에 의해 선택된 비유사 단어인지(정확도,precision)를 검사하였다. 일반적으로 재현율은 정보 검색에서 전체 존재하는 관련 데이터에서 검색엔진이 찾은 참(true) 데이터의 비율을 말하며, 정확도는 검색엔진에 의하여 찾은 데이터 중에서 참 데이터의 비율을 나타낸다.

비교 평가를 위한 기준 시스템의 정확도는 다음과 같이 계산하였다. 일반적으로 여러 다중 후보를 화면에 표시할 때 후보 단어의 우도 순위에 따라 표시할 단어 수를 결정한다. 따라서 본 논문에서는 기준 시스템의 비유사 단어를 우도 값에 따라 하위 단어로 선정하였다(예를 들어 10-Best 후보 단어일 때 9위와 10위의 후보 단어가 비유사 단어로 선정된다.). 즉, 기준 시스템의 정확도는 각 문장의 후보 단어 중에서 하위 우도를 갖는 비유사 단어를 선정하고, 이 단어가 전문가가 선택한 비유사 단어에 포함되었는 지로 계산하였다.

4.2 실험 결과

4.1절에서 설명한 500문장의 평가 데이터를 이용하여 본 논문에서 제안한 음소 거리와 음절 거리, 위치별 정확도에 의한 비유사 단어를 제거하는 실험을 하였다. 제거 방법은 각 시스템에서 검출하는 비유사 단어의 수를 지정하고 정답제시 유무에 따른 기준 시스템과 비교하였다. 전체 500문장이기 때문에 개략적으로 각 후보 단어에서 1개의 후보 단어를 제외하는 경우와 2개의 후보 단어를 제외하는 경우로 나누어 실험하였다.

<표 3>은 전체 평가 데이터 중에서 500개의 비유사 단어를 제거한 결과이다. 즉, 평균적으로 인식 결과에서 1개의 비유사 단어를 선택하고 제거하였다.

표 3. 500개의 비유사 단어를 지정한 경우 정확도
(기준시스템1과 정확도1은 정답이 제시된 경우,
기준시스템2와 정확도2는 정답이 제시되지 않은 경우)
Table 3. Precision for 500 non-similar words (Baseline1 and precision1 are evaluated with reference words, baseline2 and precision2 are without them)

시스템	Best	10-best	정확도1	정확도2
기준시스템1	81.0	89.8	82.6	-
기준시스템2	38.4	44.2	-	83.8
음소 거리	75.4	83.4	79.5	91.1
음절 거리	74.4	82.2	79.2	91.6
개별 유사도	75.8	84.2	76.9	86.4
통합 유사도	75.0	83.0	77.9	89.2

<표4>는 평균적으로 각 인식 결과에서 2개의 비유사 단어를 선택하고 제거한 결과이다.

표 4. 1,000개의 비유사 단어를 지정한 경우 정확도
(기준시스템1과 정확도1은 정답이 제시된 경우,
기준시스템2와 정확도2는 정답이 제시되지 않은 경우)
Table 4. Precision for 1,000 non-similar words (Baseline1 and precision1 are evaluated with reference words, baseline2 and precision2 are without them)

시스템	Best	10-best	정확도1	정확도2
기준시스템1	81.0	89.8	79.8	-
기준시스템2	38.4	44.2	-	81.5
음소 거리	69.4	76.8	78.0	88.8
음절 거리	69.0	76.4	78.2	88.3
개별 유사도	73.6	81.2	75.3	84.5
통합 유사도	69.0	76.4	76.8	87.9

실험 결과의 기준시스템1은 전문가들이 인식 결과 중에서 정답을 알고 있는 경우이기 때문에 Best와 10-best의 인식률은 이상적인 경우이며, 비유사 단어의 선택은 정답과 유사도 차이가 많은 단어들을 선택하게 된다. 반면 기준시스템2와 정확도2 척도는 전문가가 정답을 모른 상태에서 비유사 단어를 선택하였기 때문에 기존의 정답까지 제거하며, 정확도2의 값은 후보 단어들 중에서 유사성이 낮은 단어들을 얼마나 많이 선택하였는지를 나타낸다. 따라서 시스템에 의한 비유사 단어 제거의 성능은 기준 시스템1의 Best와 10-best 인식률에 가장 접근하면서, 기준 시스템2의 정확도2보다 큰 값을 보일수록 성능이 우수하다고 말할 수 있다.

실험 결과에서 음소 거리와 음절 거리를 이용한 단어 간의 거리를 채택한 경우, 음소 거리 방법의 성능이 우수함을 알 수 있다. 음절 거리를 이용하기 위해서는 어휘 사전에서 음절 간의 경계를 표시하여야 하나, 본 연구에서는 기존의 어휘 사전에서 대표 음절의 개수를 이용하여 음절 경계를 가정하였기 때문에 성능이 저하되었다고 판단한다. 또한, 유사도 방법을 사용한 경우 음소나 음절 거리에 의한 단어 간의 거리 방법보다는 정답 후보를 많이 보존하는 것으로 보이나, 비유사 단어의 정확도는 저하되었다. 이는 정위치 정확도와 삽입위치 정확도간의 가중치 적용에 의하여 보완될 수 있을 것이다.

5. 결 론

본 논문에서는 정보 단말기에 널리 사용되는 대용량 고립단어 인식 시스템의 인식 결과로 제시되는 다중 후보 단어들 중에서 유사성이 낮은 단어들을 제거하는 방법을 제안하였다. 제안된 방법은 후보 단어들 중의 하나를 원본 문자열로 간주하고, 나머지들을 비교 문자열로 간주한 후 원본 문자열과 비교 문자열을 레벤스타인 거리에 의하여 정렬시킨다. 정렬된 두 문자열의 구성 문자들의 속성을 정위치와 삽입위치로 규정한 후, 각 위치별 특성에 따른 관측 확률(위치별 정확도)을 계산하였다. 이 과정을 모든 후보 단어 쌍에 대하여 적용한 후, 개별 유사도 또는 통합 유사도로 평균값을 계산하여 각 후보 단어의 유사도를 계산한다. 이들 후보 단어의 유사도를 비교하여 유사도가 낮은 후보 단어들을 제외한 결과, 기존 시스템의 인식 성능의 큰 저하 없이 인식 결과들 중에서 비유사 단어를 제외할 수 있는 가능성을 보았다. 추후 연구로는 Best 인식률이 80%정도를 보이고 있어 상위 후보에 가중치를 둬 정답이 될 가능성이 높은 후보 단어를 최대한 보존하면서 비유사 단어를 찾는 방법에 대한 연구가 필요할 것이다.

감사의 글

음성인식결과를 사용하도록 허락하여 주신 한국전자통신연구원의 음성언어정보연구센터 관계자분들께 깊은 감사를 드립니다. 아울러 음성인식결과에서 추출한 500문장에 대해 비유사 단어를 결정한 한국과학기술원 김회린 교수님의 음성인식연구실 김명종・유주홍 학생에게도 깊은 감사를 드립니다.

참 고 문 헌

[1] Yun, Y.-S. (2008), “Exclusion of non-similar words form N-best word list of isolated word recognition system”, Proceedings of Conference of KSPS and KASS, pp. 183-186, November
(윤영선 (2008), “고립단어 인식 시스템의 후보 단어열에서 비유사 후보단어의 제외 연구”, 2008 대한음성학회・한국음성학회 공동학술대회 발표논문집, pp. 183-186)

[2] Park, J. G., Chung, H., Lee, Y. (2007), “Development of the Point-of-Interest Input System based on Large-vocabulary

Embedded Speech Recognition", Proceedings of Conference of KSPS, pp. 108-111, November
(박전규, 정훈, 이윤근 (2007), "내장형 대어휘 음성인식 기술에 기반하는 행선지 입력 시스템 개발", 2007 대한음성학회 가을학술대회 발표논문집, pp. 108-111)
[3] Ueffing, N., Macherey, K., Ney, H. (2003), "Confidence measures for statistical machine translation", Proceedings of MT Summit IX
[4] Wikipedia (2009), Sequence Alignment, en.wikipeida.org/wiki/Sequence_alignment

• **윤영선 (Yun, Young-Sun)** 교신저자
한남대학교 정보통신공학과
대전광역시 대덕구 오정동 133번지
Tel: 42-629-7569 Fax: 042-629-7843
Email: ysyun@hnu.kr
관심분야: 음성인식, 음성모델링, 발화검증
2001~현재 정보통신공학과 부교수

• **강점자 (Kang, Jeom-ja)**
한국전자통신연구원 음성/언어정보연구센터
대전광역시 유성구 가정동 161번지
Tel: 042-860-4880 Fax: 042-860-4889
Email: jjkang@etri.re.kr
관심분야: 음성인식, 발화검증, 잡음처리
1986~현재 한국전자통신연구원 책임기술원