

음소인식 오류에 강인한 N-gram 기반 음성 문서 검색*

이수장(KAIST), 박경미(KAIST), 오영환(KAIST)

<차례>

- | | |
|------------------------------------|----------------------|
| 1. 서론 | 4. 실험 및 결과 |
| 2. 기존 SD를 이용한 검색 기법 | 4.1. 사용한 DB 및 음소 인식기 |
| 2.1. Slot Detection | 4.2. 질의어 및 정답 집합 |
| 2.2. Probabilistic String Matching | 4.3. 성능 평가 기준 |
| 3. 확률 기반 SD를 이용한 검색 기법 | 4.4. 실험 결과 |
| 3.1. Probabilistic Slot Detection | 5. 결론 |
| 3.2. N-gram Similarity Measure | |

<Abstract>

N-gram Based Robust Spoken Document Retrievals for Phoneme Recognition Errors

Sujang Lee, Kyungmi Park, Yung-Hwan Oh

In spoken document retrievals (SDR), subword (typically phonemes) indexing term is used to avoid the out-of-vocabulary (OOV) problem. It makes the indexing and retrieval process independent from any vocabulary. It also requires a small corpus to train the acoustic model. However, subword indexing term approach has a major drawback. It shows higher word error rates than the large vocabulary continuous speech recognition (LVCSR) system. In this paper, we propose an probabilistic slot detection and n-gram based string matching method for phone based spoken document retrievals to overcome high error rates of phone recognizer. Experimental results have shown 9.25% relative improvement in the mean average precision (mAP) with 1.7 times speed up in comparison with the baseline system.

* Keywords: Probabilistic slot detection, N-gram matching, Spoken document retrieval.

* 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.

1. 서 론

컴퓨팅 파워 및 네트워크 접근성의 증가와 저장 장치의 발달로 인터넷상에 많은 양의 멀티미디어 자료가 생산되고 있다. 이중에서도 특히 음성은 해당 멀티미디어 자료의 주제, 제목, 의미 등을 내포하고 있기 때문에 매우 중요한 정보 자원이다. 따라서 음성 문서 검색(spoken document retrieval: SDR) 기술은 점점 더 핵심 응용 분야로 떠오르고 있으며, 이에 대한 연구가 활발히 진행되고 있다.

가장 일반적인 SDR은 대규모 연속 음성 인식(large vocabulary continuous speech recognition: LVCSR) 시스템을 통하여 음성 문서를 단어열(word sequence)로 전사하고, 이를 전통적인 텍스트 검색 방법을 사용하여 정보를 추출하는 방법이다. 하지만 최신 인식기라 할지라도 특정한 도메인이 정해지지 않은 음성에 대해서는 30~50% 오류율이 발생하기 때문에, 부정확한 전사가 이루어지고 결국 검색 정확도의 감소로 이어지는 문제점을 가지고 있다[1]. 또한 미리 학습되지 않은 어휘(out-of-vocabulary: OOV)로 인한 성능 저하가 발생하고, 신조어가 나올 때마다 학습하려면 다량의 코퍼스(large corpus)가 필요하다.

서브워드(subword) 기반의 색인 기법을 사용하면 이러한 문제점을 해결할 수 있다[2][3]. 서브워드 기반의 색인 기법은 한 언어를 처리하기 위해 필요한 색인어(indexing term)수가 유한하며, 색인과 검색 과정이 해당 단어 어휘와 독립적이기 때문에 어떠한 질의어(query term)라도 처리할 수 있다.

하지만 서브워드 기반의 색인 방법은 음소 인식기의 낮은 인식률 때문에 LVCSR 시스템 기반의 단어 단위 색인 방법에 비해 검색 정확도가 현저히 떨어지는 치명적인 단점을 가지고 있다. 이런 문제점을 극복하기 위해 혼동 행렬(confusion matrix)을 사용한 문서 확장[3][4] 방법과 비슷한 어휘를 질의에 포함하여 검색하는 질의 확장[5] 방법이 제안되었다. [2]에서는 음성 문서 내에서 질의어의 위치를 찾아주는 슬롯 검출(slot detection: SD) 방법과 유사도 측정을 위한 확률기반 문자열 정합(probabilistic string matching: PSM) 방법을 사용하여 음성 문서 검색을 수행하였다. [3][4]에서는 기존 텍스트 검색에서 사용되는 벡터 공간 모델(vector space model: VSM)을 기반으로 음소의 오인식을 반영한 유사도 측정 기법을 제안하였다.

최근, VSM 기반의 검색방법과 SD+PSM 방법간의 성능 비교 실험[6]을 통해 SD+PSM 방법이 VSM 기반 검색방법 보다 검색 정확도는 우수하지만 상대적으로 계산량(computational cost)이 많아 검색 시간이 오래 걸리는 문제점이 있는 것으로 확인되었다.

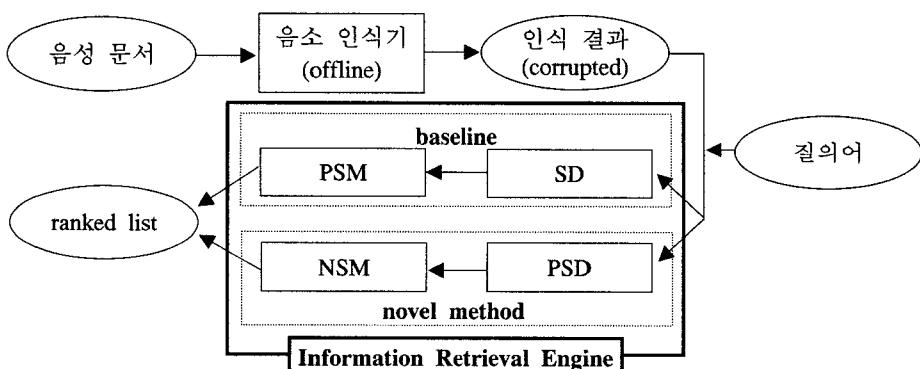
본 논문에서는 기존의 방법보다 높은 성능의 효율적인 확률기반 슬롯 검출(probabilistic slot detection: PSD) 방법을 제안한다. 이후 음소 n-gram의 특성을 이용한 n-gram 유사도 측정(n-gram similarity measure: NSM) 방법을 사용하여 빠르고,

정확한 검색 프레임워크를 실험을 통해 알아본다.

2장에서는 기존에 존재하는 SD와 PSM을 사용한 검색 방법에 대해 소개하고, 3장에서는 제안한 PSD 방법과 NSM 방법에 대해 설명한다. 4장에서는 실험에 이용하는 데이터베이스와 실험 환경에 대해 설명하고, 실험 결과를 제시한다. 마지막으로 5장에서 결론을 맺는다.

2. 기존의 SD를 이용한 검색 기법

본 장에서 소개할 기존 검색 시스템[2] 및 3장에서 제안한 검색 시스템의 구성도는 <그림 1>과 같다. 그림에서 화살표는 data flow를 의미한다.



<그림 1> 음성 문서 검색 시스템의 구성도

2.1. Slot Detection (SD)

서브워드 기반 색인 구조를 사용하려면 먼저 음성 문서를 음소 인식기로 인식하여 음소열로 바꿔야 한다. 이러한 음소열에는 단어의 경계지점에 대한 정보 (word boundary information)가 없기 때문에, 질의어가 어느 위치에서 발성되었는지 찾아 줄 SD 기법을 사용한다. SD는 질의어와 문서간의 유사도 계산을 할 영역을 줄여줌으로써 검색 속도 향상의 효과도 가져온다.

2.1.1. Scoring

SD의 첫 번째 단계에서는 인식된 음성 문서의 음소 위치(position)마다 슬롯 (음성 문서 내에서 질의어가 발성되었을 가능성의 높은 영역)이 시작될 위치인지

아닌지를 점수로 매긴다.

$$bin[k] = |\{x | 0 \leq x < l_Q \wedge D_j[k+x] = Q[x]\}| \quad (1)$$

l_Q 는 질의어의 길이(질의어를 음소열로 나타냈을 때 음소 수), $D_j[k]$ 는 j번째 문서의 음소열 중 k번째 음소를 나타낸다. $Q[x]$ 는 질의어의 음소열 중 x번째 음소를 의미한다. 식 (1)을 이용하여 음성 문서의 음소열 중 k번째 위치에서 슬롯이 시작된다고 가정했을 때 해당 위치의 슬롯과 질의어간에 공통 음소수가 몇 개인지 $bin[k]$ 에 계산하여 저장하며, 이 때 $bin[k]$ 값의 범위는 $0 \leq bin[k] \leq l_Q$ 이다. 다음으로 음소가 삽입(insertion) 또는 삭제(deletion)되는 오류를 고려하기 위해, 임의의 음소근처에서 질의어 길이에 따라 발생 가능한 삽입/삭제된 음소 수를 계산하고(식 (3)), 그 크기만큼 bin 값을 누적하여 식 (2)와 같이 $score[k]$ 를 구한다.

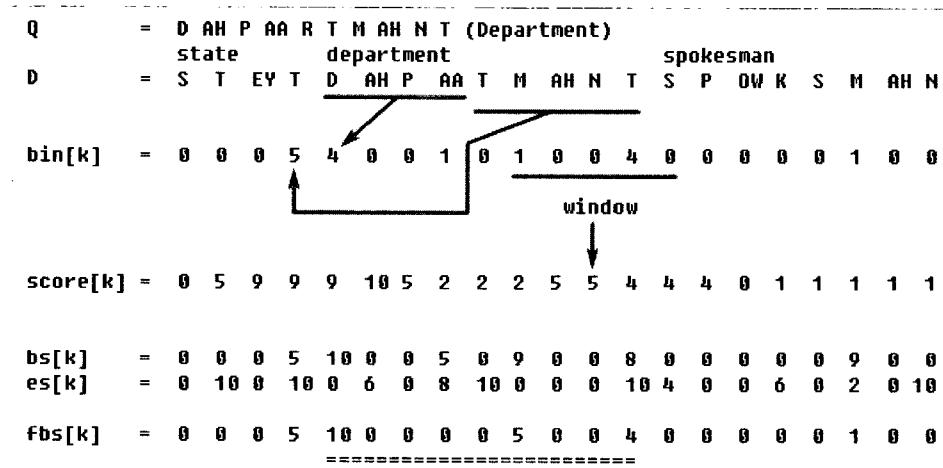
$$score[k] = \sum_{i=k-\left\lfloor \frac{w_Q}{2} \right\rfloor}^{k+\left\lfloor \frac{w_Q}{2} \right\rfloor} bin[i] \quad (2)$$

$$w_Q = \begin{cases} 1 & l_Q < 5 \\ 3 & 5 \leq l_Q < 10 \\ 5 & l_Q \geq 10 \end{cases} \quad (3)$$

여기서 k 는 음성문서의 음소열 인덱스이고, w_Q 는 윈도우(window) 크기를 나타내며 질의어의 길이에 따라 그 크기가 변한다.

<그림 2>는 음성 문서내의 department 어휘가 /R/ 음소가 탈락한 /D AH P AA T M AH N T/로 인식되었을 때의 SD 과정을 보여준다. 그림에서 보듯이 /R/음소의 탈락으로 인해 /R/음소 뒤쪽의 공통 음소수는 $bin[k]$ 가 아닌 $bin[k-1]$ 에 저장된다. 따라서 윈도우에 의한 bin 의 누적 값인 $score[k]$ 를 이용함으로써 음소의 삽입, 삭제 오류를 보상할 수 있다.

다음으로 슬롯의 영역을 좀 더 정확하게 구하기 위해 bs (beginning score)와 es (ending score)를 구한다(<그림 3>). bs 는 질의어의 길이와 슬롯 안에서 가장 왼쪽에 있는 공통 음소(left-most common phoneme)의 위치와의 차를 나타낸다. es 는 반대로 슬롯 안에서 가장 오른쪽에 있는 공통 음소(right-most common phoneme)의 상대적 위치를 의미한다. 마지막으로 식 (1)과 식 (2)에서 구한 bs 와 $score$ 를 바탕으로 k 위치에 슬롯이 위치할 가능성을 나타내는 $fbs[k]$ (final beginning score)를 계산한다.



<그림 2> SD 과정을 나타낸 예

```

FOR k= 0 TO lQ-1 DO
    R = {x|0 ≤ x < lDj ∧ Dj[x] = Q[k]}
    FOR each r ∈ R DO
        bs[r-k] = max(lQ-k, bs[r-k])
        es[r-k+lQ-1] = k+1
    END
END
FOR k = 0 TO lDj-1 DO
    beststart = argmax{bs[w]|k - ⌊ w_Q / 2 ⌋ ≤ w ≤ k + ⌊ w_Q / 2 ⌋}
    fbs[beststart] = max(score[k], fbs[beststart])
END

```

<그림 3> bs, es, fbs를 구하는 알고리즘

2.1.2. Slot detecting

슬롯을 검출하기 위한 두 번째 단계는 계산한 fbs 를 내림차순으로 정렬시키는 것이다. 가장 높은 점수의 시작 위치부터 차례로 fbs 가 임계값 $\tau \cdot l_Q$ ($0 \leq \tau \leq 1$)

보다 크면 해당 위치의 슬롯을 슬롯 집합 $S(Q, D_j)$ 에 포함시킨다. 이때 슬롯 집합에 이미 겹치는 영역을 가진 슬롯이 존재한다면 해당 슬롯은 집합에 포함시키지 않는다. 임계값은 질의어와 슬롯간의 공통 음소수의 하한값(lower bound)을 의미한다. 여기서 l_Q 는 질의어 Q 의 길이를 나타내고, τ 는 음소가 정확하게 인식될 비율을 나타내며 훈련 데이터로부터 구해진 음소 인식기의 인식률을 이 값으로 사용할 수 있다.

2.2. Probabilistic String Matching (PSM)

2.1절에 의해 구해진 슬롯 집합 $S(Q, D_j)$ 과 질의어와의 유사도를 동적 계획법(dynamic programming: DP)을 사용하여 식 (4)와 같이 계산한다.

$$\begin{aligned} 0 \leq m < l_Q, \quad 0 \leq n < l_S \\ A(0, n) &= C(Q[0], S[n]) \\ A(m, 0) &= C(Q[m], S[0]) \\ A(m, n) &= \max \begin{cases} A(m-1, n-1) + C(Q[m], S[n]) \\ A(m-2, n-1) + C(Q[m-1], \phi) \cdot C(Q[m], S[n]) \\ A(m-1, n-2) + C(\phi, S[n-1]) \cdot C(Q[m], S[n]) \end{cases} \quad (4) \end{aligned}$$

$$C(r, h) = \frac{M(r, h)}{\sum_{k \in h} M(r, h)} \quad (5)$$

S 는 슬롯 집합 $S(Q, D_j)$ 에 포함되어 있는 각각의 슬롯, l_Q 는 질의어 Q 의 길이, l_S 는 슬롯 S 의 길이를 의미한다. $C(r, h)$ 는 참조 음소(reference phoneme) r 이 주어졌을 때, 음성 인식 수행 결과에서 가설 음소(hypothesis phoneme) h 를 관찰할 확률을 나타내며, 음소 인식기로부터 얻어진 혼동 행렬 M 으로부터 근사값을 구할 수 있다[3]. ϕ 는 참조 또는 가설 음소가 없을 때를 의미한다. 행렬 A 의 계산이 끝나면 정규화(normalize)한 후 이 값을 해당 슬롯의 유사도 값으로 이용한다.

$$sim(Q, S) = \frac{\max\{A(l_Q, l) | 0 \leq l < l_S\}}{\sum_{i=0}^{l_Q-1} C(Q[i], Q[i])} \quad (6)$$

식 (6)과 같이 구해진 슬롯의 유사도 $sim(Q, S)$ 중에 가장 큰 값을 해당 음성 문서의 relevance score (RS)로 삼는다. 음성 문서의 RS에 의해 음성 문서들을 정렬

한 후 결과(ranked list)를 사용자에게 제공하면 모든 검색 과정이 끝나게 된다.

3. 확률 기반 SD를 이용한 검색 기법

2.1절에서 설명한 SD는 음소 인식과정에서 빈번하게 일어나는 음소의 삽입, 삭제 오류를 반영하여 가장 가능성이 높은 슬롯을 찾아준다. 그러나 그 판단기준으로 슬롯과 질의어간의 공통 음소수를 이용함으로써 음소의 치환(substitution)에 의한 오인식에는 강인하지 못하다는 문제점이 있다. 2.2절의 PSM의 경우 음소의 3 가지 오류(삽입, 삭제, 치환)를 모두 고려한 유사도 측정을 수행하지만, 실제 질의어가 발생된 슬롯보다 발생되지 않은 슬롯의 유사도 값이 높게 계산되는 경우가 종종 발생하여 검색 정확도가 하락한다. 이 문제들을 해결하기 위해 본 논문에서는 음소의 치환 오류를 고려한 PSD 방법을 제안한다. 또한 음소 n-gram을 이용하여 정확한 유사도를 측정할 수 있는 NSM 기법도 제안한다.

3.1. Probabilistic Slot Detection (PSD)

PSD 방법은 기존 SD 방법과는 다르게 음소간의 치환 확률을 점수(score)로 이용하며, 슬롯 영역을 추출하기 위해 추가적으로 계산해야 하는 여러 변수들이 필요하지 않다($bin, score, bs, es, fbs$ 대신 bs 만 계산). 따라서 슬롯 검출의 정확도를 높이면서 동시에 전체 시스템의 속도를 향상 시킬 수 있다.

3.1.1. Grouping

먼저 음소 p 가 주어졌을 때 이 음소와 치환될 확률이 높은 음소들을 다음과 같이 구하여 음소집합 $G(p)$ 을 만든다.

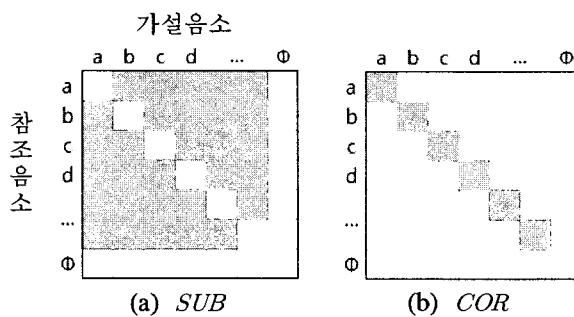
$$G(p) = \left\{ x \mid (C(p, \phi) \cdot C(\phi, x) < C(p, x)) \wedge \left(\frac{C(p, x)}{C(p, p)} > \frac{SUB}{COR} \right) \right\} \quad (7)$$

$$SUB = \frac{\sum_{i \in PS} \left\{ \sum_{j \in PS} C(i, j) \right\}}{(|PS|)^2 - |PS|}, \quad i \neq j \quad (8)$$

여기서 $C(p, \phi)$ 는 참조 음소 p 가 삭제될 확률이며, $C(\phi, x)$ 는 가설 음소 x 가 삽입될 확률, $C(p, x)$ 는 참조 음소 p 가 가설음소 x 로 치환될 확률을 나타낸다. SUB 는

$$COR = \frac{\sum_{i \in PS} C(i, i)}{|PS|} \quad (9)$$

전체 음소에 대한 치환 확률의 기대치, COR 는 전체 음소에 대한 정확하게 인식될 확률의 기대치를 나타낸다. PS 는 음소 인식기에서 사용하는 전체 음소 집합(phone set)을 나타내고, $|PS|$ 는 집합 PS 의 크기를 의미한다. 식 (7)에서 볼 수 있듯이 음소의 삽입, 삭제, 치환 오류들을 독립적인 사건으로 보고, 참조 음소 p 가 삭제될 확률과 가설 음소 x 가 삽입될 확률과의 곱보다 참조음소 p 가 가설음소 x 로 치환될 확률이 높으면, 그 가설 음소 x 는 참조음소 집합 $G(p)$ 에 포함 시킨다. 이때 치환될 확률이 너무 작으면 실제 음소 간에 치환이 일어날 가능성이 낮으므로, 이러한 가설 음소가 집합에 포함되는 것을 막기 위해 임계치(SUB/COR)를 설정한다. 각 음소마다 치환 확률의 분포가 다르기 때문에 치환 확률값 $C(p, x)$ 자체를 비교하는 것보다 정확하게 인식될 확률과 치환될 확률간의 비율(ratio)로 비교하는 것이 더 합리적이다.



<그림 4> 혼동 행렬에서 SUB와 COR을 계산하는데 사용되는 영역

SUB 와 COR 은 혼동 행렬을 이용하여 각각 계산하며, <그림 4>의 검게 표시된 영역의 평균값을 사용한다. 혼동 행렬의 맨 마지막 행과 열은 각각 음소의 삽입, 삭제 확률이 저장되어 있으므로 계산 영역에서 제외한다(혼동 행렬의 자세한 예제는 [2]를 참조).

기존의 음소 분류 방법[7]에서는 <표 1>의 (a)와 같이 전체 음소를 비슷한 발음끼리 몇 개의 그룹으로 묶고 각 그룹마다의 대표 음소를 정하여 사용하지만 본 논문에서는 음소 각각에 대해 유사음소 그룹을 생성한다. 이는 혼동 행렬이 대각 행렬을 기준으로 서로 대칭이 아니므로, 비슷한 음소들끼리 서로 음소집합 $G(p)$ 를 공유하기 어렵기 때문이다. (즉, $C(a, b)$ 와 $C(b, a)$ 는 서로 같지 않음.)

<표 1> 음소 분류(grouping) 기법 비교; (a) 기존 분류 기법, (b) 제안한 분류 기법

(a)			
Metaphone		Metaphone group	
Atl	AA, AE, AH, AO, AW		AA, AH
Ctl	CH, JH, SH		AE
Gtl	B, D, DH		AH
			AO
			...
			DH
			B, D, DH, T

3.1.2. Scoring

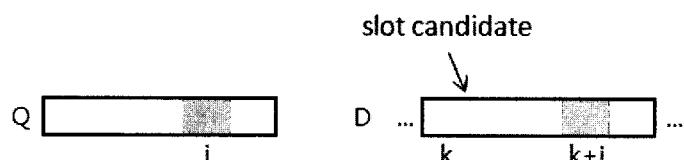
다음으로 전 단계에서 생성한 음소집합 $G(p)$ 를 이용하여 k 위치에서 슬롯이 시작될 가능성을 나타내는 $bs[k]$ 를 구한다.

$$bs[k] = \sum_{i=0}^{l_Q-1} \left\{ \sum_{p \in G(Q[i])} V(i, k, p) \right\}, \quad 0 \leq k < l_{D_j}$$

$$(10)$$

$$V(i, k, p) = \begin{cases} C(Q[i], p) & p = D_j[k+i] \\ 0 & p \neq D_j[k+i] \end{cases}$$

여기서 i, k 는 각각 질의어와 음성 문서의 음소열 인덱스를 표시하며, l_{D_j} 는 음성 문서 D_j 의 길이, p 는 음소집합 $G(Q[i])$ 에 포함된 한 음소를 나타낸다. $V(i, k, p)$ 는 질의어 Q 의 i 번째 음소와 음성문서에서 k 번째 음소로부터 시작되는 슬롯의 i 번째 음소간의 치환 확률값을 의미한다(<그림 5>). 제안한 PSD는 3.1.1절에서 구한 음소집합을 이용하여 $D_j[k+i]$ 음소가 음소집합 $G(Q[i])$ 의 원소이면 치환될 확률이 높은 음소로 판단하여 두 음소의 치환 확률값을 $bs[k]$ 계산에 적용한다. 이는 기존 SD 방법과 달리 음소의 치환 오류를 충분히 반영할 수 있다.



<그림 5> 질의어와 슬롯 예상 영역간에 서로 대응되는 음소의 위치

3.1.3. Slot detecting

이전 단계에서 계산한 bs 는 크기순으로 정렬하여 다음 임계값 θ 보다 크면 해당 부분을 슬롯 집합 $S(Q, D_j)$ 에 포함시킨다.

$$\theta = \lfloor \tau_Q \cdot l_Q \rfloor \cdot COR$$

$$\tau_Q = \frac{\sum_{i \in Q} C(i, i)}{l_Q} \quad (11)$$

여기서 Q 는 질의어의 음소열(음소들의 집합)을 나타내며, τ_Q 는 질의어 Q 내의 음소 한 개가 정확하게 인식될 확률이다. 임계값 θ 는 질의어 중 정확하게 인식될 음소수의 기대값 $\tau_Q \cdot l_Q$ 와 한 개의 음소가 정확하게 인식될 확률 COR 의 곱으로 계산하며, k 위치에서 슬롯이 시작된다고 할 때 $bs[k]$ 의 하한 기대치를 의미한다.

위 조건을 만족하면 슬롯의 시작 위치와 끝 위치를 결정하여 슬롯 집합 $S(Q, D_j)$ 에 해당 슬롯 S 를 추가한다. 기존 검색 시스템에서는 bs 와 es 를 추가로 계산하여 슬롯의 크기를 정하지만, 본 시스템에서는 식 (12)와 같이 질의어의 앞뒤로 윈도우 크기만큼을 덧붙여서 삽입, 삭제 오류를 고려하도록 슬롯 영역을 설정한다.

$$S(Q, D_j) = \left\{ S \left(k - \left\lfloor \frac{w_Q}{2} \right\rfloor, k + l_Q + \left\lfloor \frac{w_Q}{2} \right\rfloor \right) \mid bs[k] > \theta \wedge 0 \leq k < l_Q \right\} \quad (12)$$

여기서 w_Q 는 식 (3)에서 구해진 윈도우 크기를 의미한다. 슬롯집합 $S(Q, D_j)$ 에 저장되는 슬롯의 자료구조는 S (시작 위치, 끝 위치)이다.

식 (12)를 이용하여 슬롯 영역을 계산하면 <그림 6>과 같이 기본 슬롯 영역 이외에 앞뒤로 여분의 영역이 더 추가된다.

기존 SD와 비교하면 이 방식은 슬롯의 경계를 설정하는데 있어서 엄격하지 못하다. 하지만 이 방법이 효율적인 두 가지 이유가 있다. 첫째, 기존 방법도 오류가 많이 포함된 음소 인식 결과에 대해서는 슬롯 영역 설정이 정확하지 못하며, 추가적인 영역을 포함하더라도 실제 질의어가 발생된 영역을 슬롯에 포함하여야 유사도 측정과정에서 올바른 값이 계산된다. PSM 방법이나 앞으로 설명할 NSM 방법은 끝점 제약조건(endpoint constraint)이 없어서, $C(/D AH/, /D AH/) = C(D AH/, /T D AH E/)$ 과 같이 추가적인 음소가 포함되어 있어도 유사도는 올바르게 계산되기 때문이다. 둘째, 사용자의 입장에서 자신이 원하는 질의어의 발생 앞뒤로 몇 초간의 여분이 있어도 결과가 만족스럽다는 점이다. 이와 같이 슬롯

$Q = D \text{ AH } P \text{ AA } R \text{ T } M \text{ AH } N \text{ T}$ (Department)	
department	spokesman
$D = EY \quad T \text{ D } \quad \text{AH} \quad IY \text{ P } \text{AA} \text{ T } \quad M \text{ AH } \quad N \text{ T } S \quad P \text{ OW } \dots$	
$bs[k] = 0.2316 \ 0 \ 0.8375 \ 3.8930 \ 0 \ 0 \ 0 \ 0.6639 \ 0 \ 0.4789 \ 0 \ 0 \ 0.4985 \ 0 \ 0 \ \dots$	
original slot region	redundancy

<그림 6> 슬롯의 영역을 설정하는 예

영역 설정과정을 단순하게 처리하면 추가적으로 소모되는 계산량을 줄일 수 있고, 전체 시스템의 속도를 향상시킬 수 있다.

따라서 PSD는 음소 간의 치환 확률이 높은 지점을 슬롯으로 지정하고 음소의 삽입, 삭제에 의해 치환 확률이 제대로 계산되지 못하는 점을 극복하기 위해 슬롯 앞뒤로 여분의 공간을 더 포함한다.

3.2. N-gram Similarity Measure (NSM)

N-gram 구조는 문맥정보를 포함하고 있기 때문에 전통적인 텍스트 검색 분야, 생명정보학 등 여러 응용분야에서 많이 사용하고 있다[13]. 본 논문에서는 음소 n-gram을 이용하여 슬롯과 질의어 간의 유사도를 계산하는 n-gram similarity measure (NSM)를 이용한다. 슬롯 및 질의어의 음소열로부터 음소 n-gram을 구성할 때는 <표 2>처럼 한 음소씩 옆으로 이동하면서 겹쳐지게 만든다.

<표 2> 서브워드 단위 예

Subword Unit	Terms
word	department
phone ($n=1$)	D AH P AA R T M AH N T
phone ($n=2$)	D_AH AH_P P_AA AA_R R_T T_M M_AH AH_N N_T
phone ($n=3$)	D_AH_P AH_P_AA P_AA_R AA_R_T R_T_M T_M_AH M_AH_N AH_N_T

기존 시스템에서는 2-gram, 3-gram의 혼동 행렬을 구하기 위해 10시간의 방송 뉴스 데이터를 사용하였다[8]. 이처럼 음소 n-gram의 빈도수(frequency)를 구하기 위해 필요한 학습 데이터양은 n-gram의 크기가 커질수록 기하급수적으로 증가한다. 따라서 본 논문에서는 적은 양의 학습 데이터를 사용하여 간단하게 계산할 수 있는 음소간의 정합도(matching score: MS)를 식 (13)과 같이 제안한다.

$$MS_N = \frac{1}{N_Q} \cdot \sum_{i=0}^{l_Q-N} \max \left(\sum_{k=0}^{N-1} C(Q[i+k], S[j+k]) \right), \quad 0 \leq j \leq l_S - N \quad (13)$$

$$sim(Q, S) = \sum_{N=1}^4 \alpha_N \cdot MS_N \quad (14)$$

여기서 N 은 음소 n-gram의 크기를 나타내고 N_Q 는 질의어로부터 생성된 음소 n-gram의 수이다. 예를 들어 질의어가 /K AA IH S T/라고 할 때 $N=1$ 이면 $N_Q = 5(K, AA, IH, S, T)$ 가 되고, $N=2$ 이면 $N_Q = 4(K_AA, AA_IH, IH_S, S_T)$ 이다. 결국 MS_N 은 질의어와 슬롯을 음소 n-gram 단위로 비교했을 때 얻어질 수 있는 유사도의 최대 기대치를 저장한다.

[3][8]에서는 검색 정확도를 더욱 향상시키기 위해 음소 n-gram을 추출한 뒤, 이를 각각으로부터 계산된 유사도를 선형적으로 병합하는 방법을 제안하였다. 본 논문에서는 1-gram부터 4-gram까지를 사용하여 MS_N 을 구하고, 이를 식 (14)와 같이 병합하였다. α_N 는 각 n-gram에 곱해지는 가중치이다. MS_N 은 N 이 증가함에 따라 절대적인 수치가 증가하므로 이를 정규화하기 위해 가중치로 {1, 0.6, 0.5, 0.4}를 값으로 사용하였으며, 이 수치는 실험적으로 구하였다.

마지막으로, 식 (14)에 의해 계산된 슬롯의 유사도를 이용하여 음성문서를 높은 유사도 순으로 정렬하고 그 결과를 사용자에게 제공하면 검색이 완료된다.

4. 실험 및 결과

4.1. 사용된 DB 및 음소 인식기

본 연구에서는 LDC에서 배포한 3시간 분량의 1999 Hub4 Broadcast News Evaluation English Test Material을 평가 데이터로 사용하였다. 이들 데이터 중 최초 30분 분량은 검색 시스템 개발 및 혼동 행렬을 계산하는데 사용하였고[9], 나머지 2시간 30분은 평가에 이용하였다.

음소인식기는 Sphinx4[10]와 함께 제공되는 음향 모델을 사용하여 만들었다. 실험 결과 음소 인식기의 정확도는 53%로 나타났다.

4.2. 질의어 및 정답 집합

질의어는 일반성(generality)을 잃지 않기 위해 3개 이상의 음성 문서에 존재하

고 음소열 길이가 3 이상인 단일단어 288개를 사용하였다. 길이를 제한한 이유는 단어 내 음소수가 충분하여 음소 n-gram 효과를 잘 활용하기 위함이다. 실험에 사용된 질의어의 평균 길이는 6.8, 최소 길이 3, 최대 길이는 14이다. 사용자로부터 키보드를 통해 질의어가 입력되면 이를 FreeTTS[11]를 사용하여 음소열로 변환한다. 질의어가 해당 음성 문서 내에서 한번 이상 발생되었다면 그 문서를 정답으로 간주하여 정답 집합을 구성하였다.

4.3. 성능 평가 기준

검색 시스템은 최종 결과 Top-N 안에 질의어가 발생된 음성 문서가 포함되어 있는지 여부에 따라 검색 정확도가 평가된다. 이를 위해 [12]에서 제시한 11-point precision을 사용하였다. 이 방법은 각 음성 문서의 순위도 고려 대상에 포함되기 때문에 검색 시스템을 평가하기 좋은 기준이다. 본 실험에서는 Top 20까지의 문서를 가지고 평가하였다.

이 후 11-point precision으로 구해진 recall-precision 곡선으로부터 mean average precision (mAP)를 계산하여 각 시스템의 검색 정확도를 비교하였다. 이와 더불어 시스템들의 계산량을 비교하기 위해 검색에 소요된 시간도 측정하였다.

4.4. 실험 결과

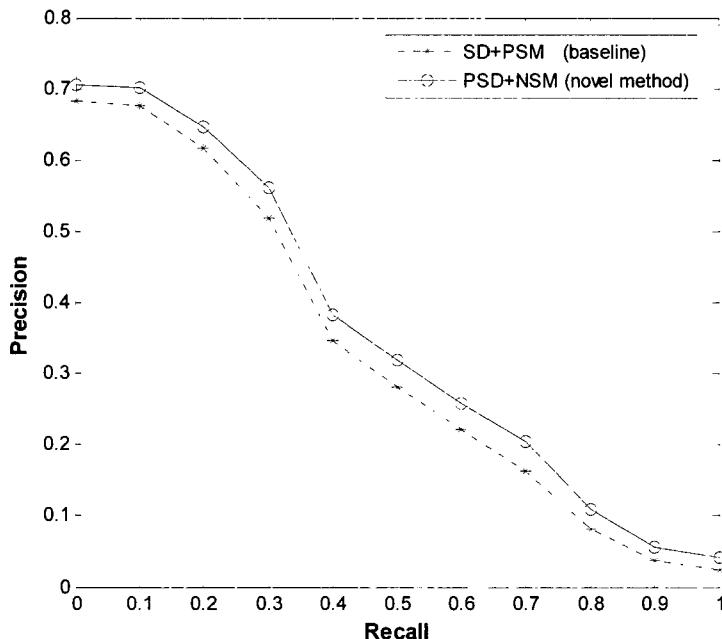
실험은 질의어 288개에 대한 11-point precision을 구한 후 그 평균을 계산하였다. 검색 시간은 질의어가 입력 된 후 정렬된 음성 문서 목록이 출력 될 때까지의 시간간격을 측정하며, CPU의 프로세스 스케줄링 및 입출력장치 상태에 따라 영향을 받으므로 실험을 10회 반복하여 그 평균값을 구하였다.

<그림 7>은 기존 시스템(SD+PSM)과 제안한 시스템(PSD+NSM)의 recall-precision 그래프이다. 곡선의 아래 면적은 mAP를 의미하며, 제안한 시스템이 기존 시스템보다 높은 검색 정확도를 나타내었다.

이와 더불어 제안한 PSD와 NSM이 검색 시스템에 미치는 영향을 알아보기 위해 기존 시스템과 조합한 실험을 하였으며, 그 결과는 <표 3>, <그림 8>과 같다.

<표 3> 각 시스템간의 검색 정확도 및 검색 시간 비교

	SD + PSM	PSD + PSM	SD + NSM	PSD + NSM
Time(sec.)	577.3	234.9	853.0	341.1
mAP	0.3320	0.3527	0.3556	0.3627

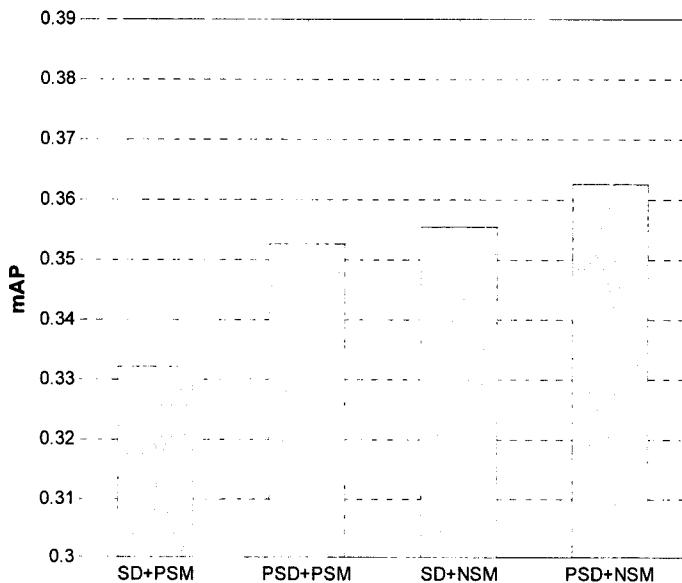


<그림 7> 기존 시스템과 제안한 시스템의 recall-precision 곡선

SD+PSM과 PSD+PSM 시스템을 비교해 보면 PSD+PSM이 mAP가 0.0207만큼 증가한 것을 확인할 수 있으며 이는 상대적으로 6.23%의 정확도가 향상된 것이다. 이는 <그림 9>에서 볼 수 있듯이 음소의 치환 오류가 존재하더라도 PSD는 SD보다 질의어가 포함된 슬롯을 효과적으로 검출할 수 있기 때문이다. <그림 9>의 (a)의 경우 임계치 3.36 보다 f_{bs} 3이 작기 때문에 해당 영역은 슬롯 집합에 포함되지 않는다.

이와 더불어 PSD는 기존 SD에 비해 불필요한 슬롯 수를 줄여 줌으로써, 이후 PSM 단계에서 처리할 계산량을 줄여주기 때문에 전체 검색 시간이 2.5배 빨라졌다. 288개 질의어에 대해 기존 SD는 296,241개, PSD는 142,723개의 슬롯을 검출한다. 또한 PSD+PSM에서 검출하는 슬롯 수가 더 적으면서도 SD+PSM 보다 정확도가 올라가는 이유는 질의어를 포함하지 않으면서도 유사도 값만 높은 가짜 슬롯 (false alarm)들을 제거해 주기 때문으로 판단된다.

SD+PSM과 SD+NSM의 결과를 비교해 보면 NSM이 PSM에 비해 계산량이 많지만, 검색 정확도는 상대적으로 7.1% 높아진 것을 확인할 수 있다. <표 4>를 보면 실제 질의어가 포함된 슬롯은 (b)이지만 PSM의 경우 가짜 슬롯인 (a)의 유사도를 (b)보다 높게 측정한다. 반면 NSM은 올바르게 유사도를 측정함을 확인할 수 있다.



<그림 8> 각 시스템의 mAP 비교

```

Q = F L AO R AH D AH (Florida)
                               florida      on sunday
D   = ... AH F L OY R DH AA N   S AY N D EY
fbs =             3           1

```

theta = 3.36

(a)

```

D   = ... AH F L OY R DH AA N   S AY N D EY
bs =           1.8406       0.3665

```

theta = 1.4444

(b)

<그림 9> SD와 PSD의 슬롯 검출 예제; (a) SD를 이용한 경우(슬롯 검출 실패),
 (b) PSD를 이용한 경우 (검정색 영역의 슬롯 검출)

<표 4> PSM과 NSM의 유사도 측정 예제

검출된 슬롯들		유사도 측정값	
(a)	Prince Norodom D = ... S N AO R AH T AH ...	유사도(sim)	
		PSM	0.6012
(b)	Florida some D = ... P L AO R N AH S AH ...	유사도(sim)	
		PSM	0.5202
		NSM	1.7534

5. 결 론

본 논문에서는 기존 시스템보다 빠르고 효율적인 PSD와 NSM을 이용한 검색 시스템을 제안하였다. 기존의 SD는 질의어와 음성 문서간의 공통 음소수를 이용하여 슬롯의 존재 여부를 판단하므로 음소의 치환 오류에는 강인하지 못하였다. 이 문제를 해결하기 위해 혼동 행렬로부터 얻어진 음소 치환 확률을 기준으로 슬롯을 판단하는 PSD를 제안하였으며, 기존 SD를 사용한 방법보다 mAP 기준 0.0207(상대적 정확도 향상 6.23%)의 검색 정확도 향상과 함께 2.5배 속도 향상 결과까지 얻을 수 있었다. 또한 기존 PSM은 정답이 아닌 슬롯을 정답인 슬롯 보다 높게 유사도를 측정하는 경우가 발생하는데 이를 극복하기 위한 NSM을 사용하여, mAP가 0.0235(상대적 정확도 향상 7.1%)만큼 높아진 것을 확인하였다.

결과적으로 제안한 방법은 검색 속도를 기존의 방법에 비해 1.7배 향상시켰으며, 검색 정확도 측면에서도 mAP 기준으로 0.0307(상대적 정확도 향상 9.25%)만큼의 성능 향상을 얻었다.

음소 인식기는 적은 양의 학습데이터를 가지고 만들 수 있고, 컴퓨팅 파워가 낮은 곳에도 적용할 수 있으므로, 제안한 방법은 휴대폰이나 PMP와 같은 각종 임베디드 시스템에도 사용할 수 있을 것으로 기대된다.

향후에는 음소 n-gram의 가중치를 학습적으로 구할 수 있는 방법에 대해 연구하고, [3]에서 제안한 VSM 기반 시스템에 PSD를 적용하여 검색 정확도 손실은 최소로 하면서 검색 시간은 더 빠르게 향상시킬 수 있는 방안을 연구할 예정이다. 또한 검색 정확도를 더 높이기 위해, 오류가 포함된 문자열간의 유사도를 정확하게 측정할 수 있는 방법에 대한 연구도 필요하다.

참 고 문 헌

- [1] C. Chelba, T. Hazen, M. Saracclar, "Retrieval and browsing of spoken content", *IEEE Signal Processing Magazine*, Vol. 25, No. 3, pp. 39-49, 2008.
- [2] M. Wechsler, *Spoken Document Retrieval Based on Phoneme Recognition*, Ph.D. Thesis, ETH Zurich, No. 12879, Oct. 1998.
- [3] K. Ng, *Subword-Based Approaches for Spoken Document Retrieval*, Ph.D. Dissertation, Mass. Inst. Technol., 2000.
- [4] N. Moreau, H.-G Kim, T. Sikora, "Phonetic confusion based document expansion for spoken document retrieval", *Proc. ICSLP*, pp. 542-545, 2004.
- [5] B. Logan, J. V. Thong, P. Moreno, "Approaches to reduce the effects of OOV queries on indexed spoken audio", *IEEE Transactions on Multimedia*, Vol. 7, No. 5, pp. 899-906, 2005.
- [6] N. Moreau, S. Jin, T. Sikora, "Comparison of different phone-based spoken document retrieval methods with text and spoken queries", *Proc. Interspeech*, pp. 641-644, 2005.
- [7] A. Amir, S. Srinivasan, A. Efrat, "Search the audio, browse the video- A generic paradigm for video collections", *EURASIP Journal on Applied Signal Processing*, Vol. 2, pp. 209-222, 2003.
- [8] U. V. Chaudhari, M. Picheny, "Improvements in phone based audio search via constrained match with high order confusion estimates", *Proc. Automatic Speech Recognition & Understanding (ASRU)*, pp. 665-670, 2007.
- [9] A. Lovitt, *Correcting Confusion Matrices for Phone Recognizers*, IDIAP Communication, No. 3, 2007.
- [10] K. Lee, H. Hon, R. Reddy, "An overview of the Sphinx speech recognition system", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 38, No. 1, pp. 35-45, 1990.
- [11] W. Walker, P. Lamere, P. Kwok, *Freetts - A Performance Case Study*, Tech. Rep., Sun Microsystems, 2002.
- [12] TREC, "Common evaluation measures", *Proc. Text REtrieval Conference*, pp. A-14, 2001.
- [13] G. Kondrak, "N-gram similarity and distance", *Proc. 12h Int'l. Conf. on String Processing and Information Retrieval*, pp. 115-126, 2005.

접수일자: 2008년 8월 11일

제재결정: 2008년 9월 3일

▶ 이수장(Sujang Lee) : 교신저자

주소: 305-702 대전광역시 유성구 구성동 373-1 한국과학기술원

소속: 한국과학기술원(KAIST) 전자전산학과 전산학전공 음성인터페이스 연구실

전화: 042) 869-5556

E-mail: lsujang@speech.kaist.ac.kr

▶ 박경미(Kyungmi Park)

주소: 305-702 대전광역시 유성구 구성동 373-1 한국과학기술원

소속: 한국과학기술원(KAIST) 전자전산학과 전산학전공 음성인터페이스 연구실

전화: 042) 869-5556

E-mail: kmpark@speech.kaist.ac.kr

▶ 오영환(Yung-Hwan Oh)

주소: 305-702 대전광역시 유성구 구성동 373-1 한국과학기술원

소속: 한국과학기술원(KAIST) 전자전산학과 전산학전공 음성인터페이스 연구실

전화: 042) 869-3516

E-mail: yhoh@speech.kaist.ac.kr