한국음향학회지 제21권 제3호 pp. 274~284 (2002)

마코프 체인 및 음절 N-그램을 이용한 한국어 띄어쓰기 및 복합명사 분리

Korean Word Segmentation and Compound-noun Decomposition Using Markov Chain and Syllable N-gram

권 오 욱*
(Oh-Wook Kwon*)

* 한국과학기술원 뇌과학연구센터

(접수일자: 2001년 3월 7일; 수정일자: 2001년 12월 7일; 채택일자: 2002년 1월 28일)

한국어 대어휘 연속음성인식을 위한 텍스트 전처리에서 띄어쓰기 오류는 잘못된 단어를 인식 어휘에 포함시켜 언어모델의 성능을 저하시킨다. 본 논문에서는 텍스트 코퍼스의 띄어쓰기 교정을 위하여 한국어 음절 N-그램을 이용한 자동 띄어쓰기 알고리듬을 제시한다. 제시된 알고리듬에서는 주어진 입력음절열은 좌에서 우로의 천이만을 갖는 마코프 체인으로 표시되고 어떤 상태에서 같은 상태로의 천이에서 공백음절이 발생하며 다른 상태로의 천이에서는 주어진 음절이 발생한다고 가정한다. 마코프 체인에서 음절 단위 N-그램 언어모델에 의한 문장 확률이 가장 높은 경로를 찾음으로써 띄어쓰기 결과를 얻는다. 모든 공백을 삭제한 254문장으로 이루어진 신문 칼럼 말뭉치에 대하여 띄어쓰기 알고리듬을 적용한 결과 91.58%의 어절단위 정확도 및 96.69%의 음절 정확도를 나타내었다. 띄어쓰기 알고리듬을 응용한 줄바꿈에서의 공백 오류 처리에서 이 알고리듬은 91.00%에서 96.27%로 어절 정확도를 향상시켰으며, 복합명사 분리에서는 96.22%의 분리 정확도를 보였다.

핵심용어: 띄어쓰기, 트라이그램 언어모델, 텍스트 전처리
투고분야: 음성처리 분야 (2.7)

Word segmentation errors occurring in text preprocessing often insert incorrect words into recognition vocabulary and cause poor language models for Korean large vocabulary continuous speech recognition. We propose an automatic word segmentation algorithm using Markov chains and syllable-based n-gram language models in order to correct word segmentation error in text corpora. We assume that a sentence is generated from a Markov chain. Spaces and non-space characters are generated on self-transitions and other transitions of the Markov chain, respectively. Then word segmentation of the sentence is obtained by finding the maximum likelihood path using syllable n-gram scores. In experimental results, the algorithm showed 91.58% word accuracy and 96.69% syllable accuracy for word segmentation of 254 sentence newspaper columns without any spaces. The algorithm improved the word accuracy from 91.00% to 96.27% for word segmentation correction at line breaks and yielded the decomposition accuracy of 96.22% for compound-noun decomposition.

Keywords: *Word segmentation, N-gram language model, Text preprocessing*
ASK subject classification: *Speech signal processing (2.7)*

책임저자: 권오욱 (owkwon@ucsd.edu)
Institute for Neural Computation, University of California, San Diego
9500 Gilman Dr, La Jolla, CA 92093-0523, USA
(전화: +1-858-822-1905; 팩스: +1-858-534-2014)

I. 서 론

한국어 대어휘 연속음성인식을 위한 사전 및 언어모델 구축을 위하여 사용되는 대규모의 말뭉치를 얻기 위해서는 자동 텍스트 처리 알고리듬이 필수적이다. 영어의 경우 20k단어 연속음성인식을 위한 WSJ 태스크에서는 약 38M의 단어로 이루어진 말뭉치를 사용하였으며 65k 단어급인 NAB 태스크에서는 약 300M 단어 규모의 말뭉치를 사용하였는데[1,2], 이 정도의 말뭉치를 수작업으로 오류를 수정하는 것은 거의 불가능하다. 특히 형태소 기반 대어휘 음성인식[3,4]을 위한 말뭉치에서는 띄어쓰기 오류는 형태소 태깅 오류를 야기하고 잘못된 형태소를 인식어휘에 넣게 되어 결과적으로 음성인식 성능을 저하시킨다.

신문기사 및 방송뉴스는 받아쓰기 엔진 및 방송뉴스 인식에서의 언어모델링을 위한 말뭉치로서 자주 사용된다[1,2]. 그러나 한국어 대어휘 연속음성인식을 위한 대량의 깨끗한 말뭉치는 아직까지 구할 수 없기 때문에 인터넷으로부터 구한 신문기사 또는 방송뉴스 대본을 전처리하여 말뭉치를 얻는다[3]. 그런데 웹 문서 또는 과거의 텍스트 파일에서는 종종 어절 경계가 아닌 곳에서 줄바꿈되어 있으며, 입력 오류 등에 의하여 틀린 단어 또는 띄어쓰기 오류가 포함된 경우도 있다. 본 논문에서는 이러한 띄어쓰기 오류를 자동으로 교정하기 위한 띄어쓰기 알고리듬을 제안하며, 이를 줄바꿈에서의 공백 처리, 긴 어절의 띄어쓰기 오류 수정, 복합명사 분리에 적용한다. 복합명사 분리는 최근에 인터넷 검색엔진에서의 질의어 처리에 자주 사용되고 있는데 이는 하나의 어절이 단위명사의 조합으로만 이루어진 자동 띄어쓰기의 특수한 예라고 볼 수 있다.

한국어에 대해서는 주로 자연어 처리 분야에서 띄어쓰기 자동 교정 및 정보 검색을 위한 색인어 처리를 위하여 띄어쓰기 알고리듬[5,6] 및 복합명사 분리 방법[7-11]이 연구되었다. 띄어쓰기에 대한 접근방법으로는 통계적인 정보만을 사용하는 통계적 방법과 한국어의 조사 및 어미의 특성을 반영하거나 복합명사의 구성 패턴을 참조하는 규칙기반 방법이 있다[5,7]. 규칙기반 접근에서는 형태소 해석 또는 사전 정보가 필수적이며, 사전에 들어 있지 않은 미등록어가 포함된 문장 및 복합명사의 처리시에 오류가 증가하게 된다. 그런데 신문기사나 방송뉴스와 같이 시사적인 내용을 많이 포함하는 영역에서는 새로운 인명, 지명, 회사명, 외국어 등 미등록어가 일반적인 문서보다 많기 때문에 띄어쓰기 성능이 저하된다. 통계적 접근방법은 언어적 지식이 없이도 적용할 수 있지만 통계적 정보를 얻는 데 사용한 영역과 다른 곳에 적용할 때 성능이 저하된다. 한국어 이외의 다른 언어에서도 복합명사 분리를 해결하기 위한 알고리듬이 연구되어 왔다[12-14].

통계적 접근방법 중에 음절간의 상호정보를 이용한 자동 띄어쓰기 알고리듬이 있다[5]. 상호정보에 의하여 전체 문장 중에서 띄어쓰기 가능성이 높은 부분에서 형태소 해석을 시도하고 성공하면 띄어쓰고 그렇지 않으면 다시 더 좁은 범위에서 가능성이 높은 위치를 찾는 과정을 반복한다. 이 알고리듬의 응용으로서 합성된 상호정보를 사용하여 띄어쓰기 위치를 찾은 다음 단위 명사가 명사사전에 등록되어 있는지를 확인하는 복합명사 분리 방법이 제안되었다[7]. 이 방법은 미등록어가 있을 때에 성능이 크게 감소한다.

규칙기반 접근방법에서는 먼저 조사/어미로 사용되는 음절의 특성을 이용하여 어절 블록의 경계를 찾고 어절 블록 내에서는 형태소 분석을 이용한 양방향 최장 일치법을 사용한다[6]. 블록경계 오류시에는 다음 어절에 붙여서 어절 인식을 다시 시도하는 후처리 적용한다. 정확도는 상호정보를 사용한 것보다 높게 나타났으며 성능이 사전에 크게 의존한다. 특히 외국어나 고유명사에 의한 미등록어가 많은 경우에는 성능이 저하된다. 복합명사의 분리에서는 분해 규칙과 예외 규칙을 구한 다음 이를 적용한다[8]. 이외에도 다른 방법을 사용한 복합명사 분리에 대한 연구가 있다[9,11].

본 논문에서는 대어휘 연속음성인식을 위하여 신문기사 및 방송뉴스로부터 구한 대규모의 말뭉치를 다룬다. 이러한 말뭉치에서는 고유명사 및 외국어와 같은 미등록어가 많기 때문에 통계기반의 접근방식을 사용하였다. 제안된 알고리듬에서는 주어진 입력음절열은 좌에서 우로의 천이만을 갖는 마코프 체인으로 표시되고 어떤 상태에서 같은 상태로의 천이에서 공백음절이 발생하며 다른 상태로의 천이에서는 주어진 음절이 발생한다고 가정하였다. 2개 이상의 공백이 연속으로 나타날 수 없다는 제한조건하에서 음절 n-그램에 의한 문장 확률이 최대가 되는 음절열을 찾음으로써 최적의 띄어쓰기 해를 찾았다. 성능 향상을 위하여 단어 길이에 대한 확률분포를 추가로 적용하였다. 제안 방법은 어휘 지식이나 휴리스틱을 사용하지 않고 통계적인 방법을 사용하여 임의의 길이의 띄어쓰기가 잘못된 문장을 바르게 고칠 수 있으며 문장의 일부 또는 전체가 띄어쓰기 되지 않은 문장에 대해서도 동작한다.

제II장에서는 제안된 음절 n-그램을 이용한 띄어쓰기

알고리듬을 기술한다. 제III장에서는 제안된 알고리듬을 응용한 전혀 띄어쓰기가 되어 있지 않은 문장의 띄어쓰기, 줄바꿈에서의 띄어쓰기, 복합명사 분리에 대한 실험 결과를 검토한다. 제IV장에서는 결론을 맺는다.

II. 띄어쓰기 알고리듬

2.1. 문장 발생 모델

음절열로 구성된 한국어의 문장 $S=(w_1\ w_2\ \cdots\ w_N)$이 주어져 있을 때 그 문장이 마코프 체인[15,16]으로부터 발생한다고 가정한다. 공백도 하나의 음절이라고 가정하며, 마코프 체인의 각 천이는 음절을 발생한다. 자신으로의 천이에서는 공백이 발생되고 다른 상태로의 천이에서 음절이 발생한다. 상태 s_i에서 s_{i+1}로의 천이는 $a_{i,i+1}$로 표시하며 하나의 음절 w_i을 발생하며, 상태 s_i에서 s_i로의 천이는 $a_{i,i}$로 표시하며 주어진 음절열에서 i번째 음절 다음의 공백을 발생한다. 각 천이는 주어진 문장의 각 음절을 심볼로 가지며 그 이전에 발생한 $n-1$개의 음절에 의존한다고 가정하면 $n-1$차 마코프 체인이 되며 천이 확률은 다음과 같이 n-그램 확률[17]로 주어진다. 특히 $n=1,2,3$인 경우 각각 유니그램, 바이그램, 트라이그램이라 한다.

$$p(a_t|a_1,a_2,\cdots,a_{t-1})\approx p(a_t|a_{t-n+1},\cdots,a_{t-1}) \quad (1)$$

$n=3$인 경우 2차 마코프 체인이 되고 천이 확률은 다음과 같다.

$$p(a_t|a_1,a_2,\cdots,a_{t-1})\approx p(a_t|a_{t-2},a_{t-1}) \quad (2)$$

어떤 문장 S가 주어진 경우 트라이그램 확률에 의한 문장 확률은 다음과 같이 계산된다.

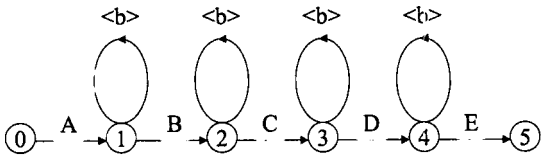

그림 1. "ABCDE"에 대한 문장 발생 모델
Fig. 1. Sentence generation model for "ABCDE".

$$p(S)=p(a_1,a_2,\cdots,a_N)$$
$$=p(a_1)p(a_2|a_1)p(a_3|a_1,a_2)\cdots p(a_N|a_{N-2},a_{N-1}) \quad (3)$$

음절발생 순서가 주어진 입력 음절들의 순서와 같아야 하고 주어진 모든 음절은 출력에도 나타나야 하며, 두개 이상의 공백이 연속해서 발생하면 안 된다는 상태천이의 제한조건이 있다. 따라서, 상태천이는 왼쪽에서 오른쪽으로의 천이만 허용되고, 주어진 음절이 공백이 아닌 경우에는 건너뛰기가 허용되지 않으며, 같은 상태로의 천이는 한번만 허용된다. 주어진 문장에 공백이 존재하는 경우에는 붙여쓰기를 위하여 공백음절을 발생하는 상태로의 천이를 건너뛸 수 있으며, 두 개의 공백이 연속적으로 나타나지 않게 하기 위한 추가적인 제한조건이 적용된다.

그림 1은 띄어쓰기가 전혀 되어 있지 않은 5개의 음절로 구성된 문장 "ABCDE"에 대한 마코프 체인을 나타낸다. A, B, C, D 및 E는 임의의 한글 음절을 의미하고, 〈b〉는 공백을 의미한다. 그림에서 0－5는 6개의 상태(state)를 나타내며, 상기 마코프 모델은 9개의 천이(transition)로 구성된다. 문장의 제일 앞과 뒤에 알고리듬 설명의 편의를 위하여 가상 상태를 추가한다.

그림 1에서 가능한 한글 띄어쓰기 형태는 "A B C D E", "AB C D E", "A BC D E", "AB CD E", "AB CDE", ..., "ABCDE"까지 총 16가지가 존재하고, 띄어쓰기 교정문제는 이들 패턴 중에서 최적의 패턴을 고르는 것이 된다. 상기 16가지의 패턴에서 나타나는 서로 다른 단어의 개수는 15개이다. 일반적으로 N개의 음절로 구성된 문장을 띄어쓰기

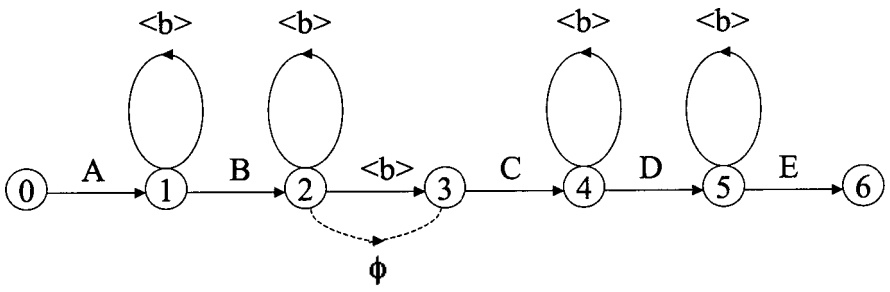

그림 2. "AB<b>CDE"에 대한 문장 발생 모델
Fig. 2. Sentence generation model for "AB<b>CDE".

하기 위하여 $2N-1$개의 패턴을 검사하여야 하며, 단어가 사전에 존재하는지를 검사한다면 $\frac{1}{2}N(N+1)$번의 사전검색이 필요하다. 만일, 음절개수가 증가하면 비교하여야 할 패턴의 수는 $O(N^2)$으로 증가하게 된다.

그림 2는 "AB⟨b⟩CDE"의 6개의 음절로 구성된 문장에 대한 마코프 체인의 예이다. 공백도 하나의 음절로 간주되어 체인을 형성하고, 이때 공백을 건너뛰기 위하여 공천이 (null transition) ϕ가 추가되었으며, 공백을 발생시키는 다른 상태로의 천이의 도착 상태 3에서는 그 자신으로의 천이가 없다. 여기서 공천이 (null transition)는 음절을 발생하지 않는 천이이다. 그림 2에서 공천이를 삭제하고 상태 2에서 상태 4로 음절 C를 발생하는 천이를 추가해도 동일한 효과를 갖는다.

2.2. 띄어쓰기 알고리듬

주어진 문장에 대하여 최적의 띄어쓰기 위치를 찾는 것은 문장에서 공백을 적절한 위치에 삽입하는 문제 또는 마코프 체인으로부터 문장이 발생한다고 가정할 때 발생할 수 있는 모든 가능한 음절열 중에서 가장 확률이 높은 천이 경로(가설)를 찾는 문제와 같다. 즉, $S=(w_1\ w_2\ \cdots\ w_N)$에 의하여 구성되는 마코프 체인으로부터 발생가능한 모든 가능한 가설 중에서 문장 확률 $p(x_1, x_2, \cdots, x_M)$가 최대인 문장을 찾는다. 여기서 x_t는 하나의 공백 또는 비공백 음절을 나타내며 M은 공백을 포함한 음절의 개수로서 가설에 따라서 달라진다.

$$S' = \arg\max_{x_1, x_2, \cdots, x_M} p(x_1, x_2, \cdots, x_M),\ M > N \tag{4}$$

그러나 n-그램 확률값은 항상 1보다 작거나 같기 때문에 가설에 포함된 음절의 개수가 증가할수록 (즉 삽입되는 공백의 숫자가 많을수록) 문장 확률은 작아진다. 이를 보상하기 위하여 본 연구에서는 다음과 같이 음절 개수에 정규화된 문장확률을 최대화하는 가설을 찾는다.

$$\hat{S} = \arg\max_{x_1, x_2, \cdots, x_M} p(x_1, x_2, \cdots, x_M)^{1/M},\ M > N \tag{5}$$

1차 마코프 체인은 음성인식에서 널리 사용되고 있는 은닉 마코프 모델 (HMM)[17]에서 상태가 관측 심볼을 나타낸다고 가정하는 것과 동일하므로 음성인식의 비터비 (Viterbi) 알고리듬[17]을 그대로 사용할 수 있다. 본 논문에서는 트라이그램 이상의 음절 이력을 고려하는 2차 마코프 체인을 사용하므로 비터비 알고리듬을 그대로 적용할 수 없다.

그림 3은 띄어쓰기 알고리듬의 설명을 위한 격자 그림이다. 상태 s에서 같은 상태로 천이할 때는 공백이 발생하고, 다른 상태 s'에서 상태 s로 천이할 때에는 음절 $A[s]$가 발생하며, $p(w_3 \mid w_1, w_2)$, $p(w_2 \mid w_1)$, $p(w_1)$는 주어져 있다고 가정한다. 각 노드에는 여러 개의 가설들이 저장될 수 있으며, 하나의 가설에는 최근 $n-1$개의 음절 정보 $(x_t, x_{t-1}, \cdots, x_{t-n+1})$, 누적 로그 확률 $L(t,s)$, 백 포인터 (back pointer) $bp(t,s)$를 갖는다. 여기서 백 포인터는 현재 가설이 이전 어느 가설에서 추출되었는지를

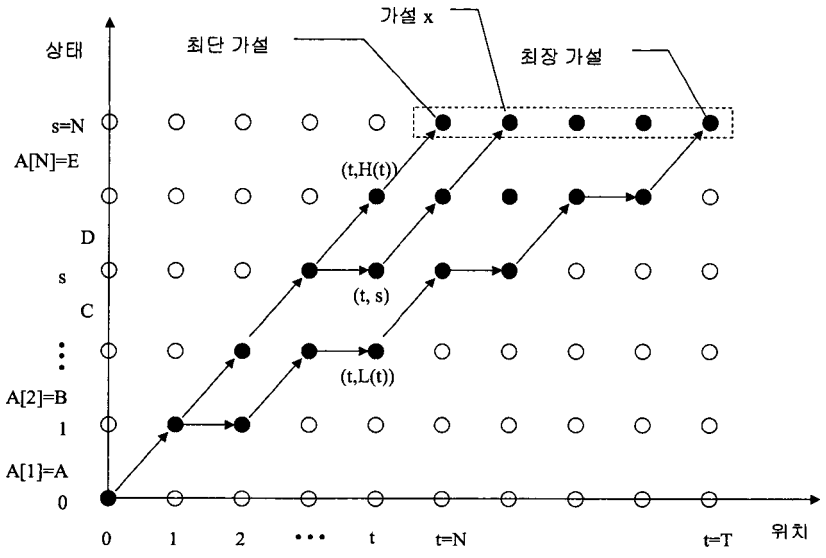

그림 3. 격자 그림
Fig. 3. Lattice diagram.

알아내기 위한 것으로 이전 가설의 위치, 상태, 포인터를 저장한다. 참고로 비터비 알고리듬에서는 하나의 가설(경로)만이 한 노드에 존재한다. 다음으로 $t=0$이고 $s=0$인 노드의 누적 로그 확률을 0으로 설정하고, 그 외 위치 $t=0$ 또는 $s=0$인 노드의 누적 로그 확률은 $-\infty$로 설정하며, 백 포인터는 모드 -1로 설정한다. 초기 최대 누적 로그 확률은 0으로 설정한다.

문장 발생모델의 제한조건에 의하여 음절개수가 N일 때 가능한 최장경로는 모든 음절사이에 공백이 삽입되는 경우로서 그 길이는 $T=2N-1$이 되며, 최단경로는 공백이 하나도 삽입되지 않은 경우로서 그 길이는 N이 된다. 시간 t에서 최장경로는 상태 $s=(t+1)/2$ 노드를 지나고, 최단경로는 상태 $s=t$ 노드를 지난다. 따라서 가능한 모든 경로는 최장경로와 최단경로 사이에만 존재하게 된다. 만일 상태 $s=N$에 도달하면, 입력으로 주어진 모든 음절이 발생된 것이다.

그림 4는 본 논문에서 사용한 띄어쓰기 알고리듬으로서, 시간 $t=1$부터 $t=T$까지 차례대로 가능한 최장경로와 최단경로 사이에 존재하는 노드만을 처리한다. 먼저 격자 그림에서 기술한 대로 초기화를 하고 최대 로그 확률을 갖는 가설을 찾은 다음 마지막 단계에서 그 가설을 역추적 (backtracking)하여 공백이 포함된 최적의 음절열을 찾는다.

시간 t와 상태가 s인 노드 (t,s)로의 천이가 가능한 t-1에서의 노드들 (t-1,s), (t-1,s-1)에 저장된 모든 가설을 추출하고, 노드 (t-1,s-1)에서 추출된 가설들에는 음절A[s]를 가설에 추가하고, 백 포인터에는 상기 상태 s-1을 저장한다. 노드 (t-1, s)에서 추출된 가설들에는 공백을 추가하고, 백 포인터에는 상기 상태 s를 저장한다. 만일 두개의 공백음절이 연속으로 발생하는 가설들은 제거한다.

n-그램에서의 음절 이력 $h=(x_{t-N+1},\cdots,x_{t-1})$는 이전에 나타난 n-1개의 음절을 의미한다. 입력 음절에 공백이 없을 경우, 연속된 2개의 공백이 올 수 없다는 다음과 같은 천이 제한조건에 따라 바이그램의 경우 2개, 트라이그램의 경우는 3개, 4-그램의 경우는 6개의 가설만을 저장하면 최적의 경로를 찾을 수 있다.

$$p(<b> \mid x, <b>)=0, \quad \forall x$$
$$p(w_s \mid w_{s-i})=0, \quad \forall i \geq 2 \tag{6}$$

그림 5는 바이그램 및 트라이그램을 사용하는 경우 가능한 천이를 나타낸다. 트라이그램을 사용하는 경우, 노드 (t,s)로의 천이가 가능한 t-1에서의 노드들은 (t-1,s),

```
/* 초기 가설 설정 */
상태 s에서 s로 천이시에는 공백이 그 외의 천이시에는 음절 A[s]가 발생한다고 가정한다.
t=0, s=0인 노드의 누적 로그확률은 0으로 하고 그 외 시간 t=0 또는 s=0인 노드들의 누적 로그확률은 -∞로 하고, back
pointer는 모두 -1로 설정한다.
최대 누적 확률은 0으로 한다.

/* 최대 누적 로그 확률 찾기 */
for t=1 to T
   for s=1 to S
            만일 s<(t+1)/2이거나 s>t이면 다음 상태로 간다.
            t-1의 노드로부터 제한 조건을 만족하는 가설들을 추출한다.
            추출된 가설에 현재 발생되는 음절과 back pointer를 기록한다.
            천이 로그 조건확률 log P(x_t | x_{t-2},x_{t-1})을 누적 로그 확률에 더한다.
            공백이 추가될 경우 단어 길이에 따른 벌칙에 더한다.
            최근 n-1음절이 같은 가설은 누적 로그 확률이 높은 것만 남긴다.
            최대 누적 로그 확률과 누적확률의 차이가 빔 크기보다 큰 가설은 제거한다.
   end
  최대 누적 로그 확률을 구한다.
end

/* Backtracking */
s=S이고 시간 S부터 T까지 노드에서 가장 큰 정규 누적 로그확률을 갖는 가설 h와 그때의 시간 t를 찾는다.
While(t>0)
            h의 back pointer를 이용하여 이전 가설 h' 을 찾는다.
            h, h' 의 상태 s, s' 이 같으면 공백을, 다르면 A[s]를 출력 스트링의 앞에 넣는다.
            t를 h' 이 속한 노드의 시간으로 설정한다.
end
```

그림 4. 제안된 띄어쓰기 알고리듬
Fig. 4. Proposed word segmentation algorithm.

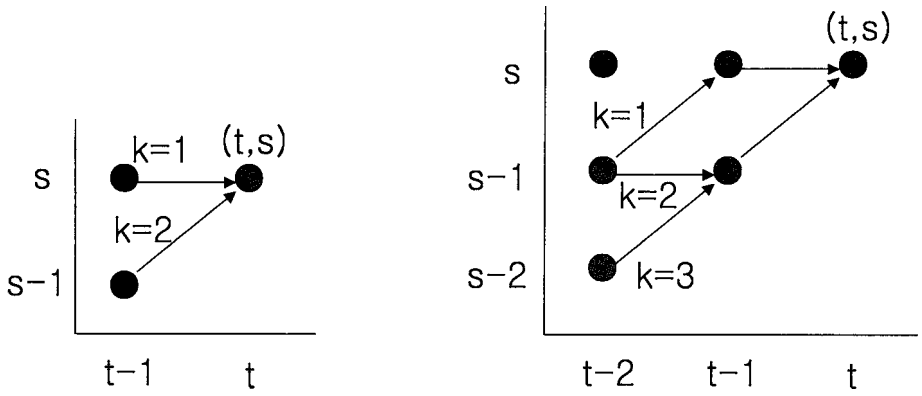

그림 5. 가능한 천이 경로
Fig. 5. Possible transition paths in case.

(t-1,s-1), (t-2,s-2)의 세 천이가 가능하다.

바이그램 및 4-그램의 경우에도 유사하게 구할 수 있으므로 여기에서는 트라이그램에 대해서만 설명한다. 노드 (t,s)의 음절 경로 k에 따른 누적 로그 문장 확률 $L(t,s,k)$는 식 (7)과 같은 회귀식으로 계산된다.

$$
\begin{aligned}
L(t,s,1) &= \max_{k'=1,2,3}\{L(t-1,s,k') \\
&\quad + \log\ p(<b> \mid x_{t-2,s-1}, w_s)\log\ p(d) + pendlty\} \\
L(t,s,2) &= \max_{k'=1,2,3}\{L(t-1,s-1,k') \\
&\quad + \log\ p(w_s \mid x_{t-2,s-1}, <b>)\} \\
L(t,s,3) &= \max_{k'=1,2,3}\{L(t-1,s-2,k') \\
&\quad + \log\ p(w_s \mid x_{t-2,s-2}, w_{s-1})\}
\end{aligned}
\tag{7}
$$

여기서 d는 그 가설이 속한 경로에서 최후 단어의 길이(음절 개수)를 의미한다. *pendlty*는 어절 삽입 벌점으로서 값이 커지면 공백의 개수가 감소한다. $\log\ p(x_t \mid h)$는 위에서 추출된 가설들에 대하여 새로운 음절 x_t의 조건 로그 확률이며, $\log\ p(d)$는 단어 길이에 대한 로그 확률이다. 각 노드에서 계산된 가설들 중에서 음절 이력이 동일한 가설들은 가장 높은 누적 로그확률을 갖는 가설만 남기고 나머지는 제거한다. 각 가설에서 최대 누적 로그 확률과 누적 로그 확률의 차이가 미리 주어진 빔 크기보다 큰 가설은 제거하고, 시간 t에서의 모든 가설 중에서 최대 누적 로그확률을 계산한다.

마지막으로 위에서 구한 최대 누적 로그 확률 및 백포인터를 이용하여 입력된 음절의 띄어쓰기 최적패턴을 탐색한다. 먼저, $s=N$이고 위치 N과 위치 T 사이에 있는 노드에 저장된 가설들 중에서 누적 로그 확률을 그 가설이 속한 경로 내의 음절 개수로 나눈 후, 상태개수 S를 곱한 정규 누적 로그 확률이 최대인 가설 h 및 그 때의 시간 t를 구하고, 상기 가설 h로부터 백 포인터를 이용하여 이전 가설 h'를 탐색한 후, 상기 이전 가설 h'로부터 가설 h로의 상태변화 결과에 따라 입력음절 또는 공백을 출력 문장의 앞에 삽입한다.

$$
(\hat{t},\hat{k}) = \arg\max_{k,N\le t\le 2N-1} L(t,N,k)/t \tag{8}
$$

정규 누적 로그 확률이 최대인 가설 h의 위치 t를 이전 가설 h'가 속한 노드의 위치로 설정하고 앞의 과정을 반복하여 입력 문장의 음절배열 A에 대하여 띄어쓰기가 교정된 최종 출력 문장을 얻는다.

III. 실험결과 및 토의

3.1. 음절 N-그램 및 어절 길이 분포

2년간의 동아일보 신문기사와 초등학교 교과서 문장을 훈련용 말뭉치로 사용하였다. 신문기사의 크기는 1.6M문장, 22M 어절, 174M 음절이고 교과서 문장의 크기는60k 문장 0.5M어절 3.7M 음절이었다. 신문기사 텍스트는 띄어쓰기 오류가 일부 포함되어 있으나 수작업으로 수정하지는 않았다. 규칙적인 오류가 아닌 타이핑 오류들은 통계적인 접근 방식에서는 문제시되지 않기 때문이다[5]. 초등학교 교과서 문장은 기본 단어로 이루어져 있으며 띄어쓰기가 정확하다. 이 말뭉치로부터 언어모델링 툴킷[18]을 사용하여 음절단위 트라이그램을 구하였다. 훈련용 말뭉치에서 나타나지 않은 트라이그램 확률은 Katz

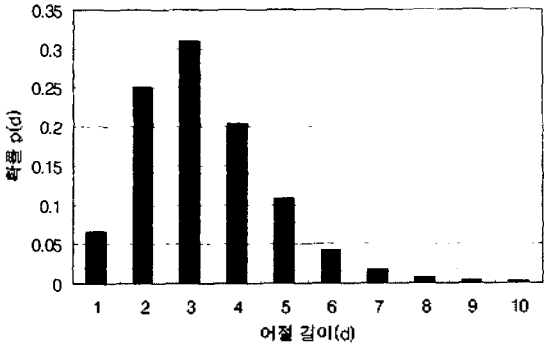

그림 6. 어절 길이의 확률 분포
Fig. 6. Probability distribution of eojeol length.

백오프 방법[19]을 사용하여 구한다. 유니그램의 개수는 영어 알파벳을 포함하여 2406개, 바이그램은 172895개, 트라이그램은 1310468개이었다. 트라이그램의 메모리는 7.1 MB 정도를 차지한다.

훈련용 말뭉치로부터 어절 길이에 따른 확률 분포를 계산하였다. 그림 6은 어절 길이에 따른 확률 분포를 나타낸다. 대부분의 경우 어절 길이는 2에서 5사이에 분포하며 가장 빈도가 높은 것은 3이다. 이 확률 분포는 띄어쓰기 향상을 위하여 식 (7)에서 사용되었다.

3.2. 문장 띄어쓰기

테스트에 사용한 문장은 조선일보 칼럼에서 선택한 문장단위로 분할된 254문장 3622어절 공백을 포함하여 24250음절이었다. 그림 7은 시험에 사용한 일부 문장과 띄어쓰기 결과를 나타낸 것으로서 고유명사가 많이 포함되어 있음을 보여준다. 띄어쓰기 정확도는 음절 단위 또는 어절 단위로 표현한다[5,6]. 복합어 및 보조용언은 가능하면 분리됨을 원칙으로 하였다. 띄어쓰기 성능을 조사하기 위하여 텍스트에서 문장 단위로 공백을 모두 제거하고 띄어쓰기 알고리듬을 적용하였으며 $penalty=0$을 사용하였다.

띄어쓰기 정확도는 기준 단위에 따라서 음절 단위 정확도 (SA) 및 어절 단위 정확도 (WA)를 정의하였다.

$$SA=(1-\frac{S+M}{N_s})\times 100(\%) \tag{9}$$

$$WA=\frac{C}{N_w}\times 100(\%) \tag{10}$$

여기서 S는 붙여써야 하는 어절을 띄어쓴 오류 (붙띄오류)의 갯수이며 M은 띄어써야 할 어절을 붙여쓴 오류 (띄붙오류)의 갯수를 나타내며, C는 띄어쓰기 위치가 맞은 개수이다. N_s와 N_w는 각각 오류가 없는 전체 테스트 문장의 음절 개수 및 어절 개수를 나타낸다.

표 1. 음절단위 n-그램 언어모델에 따른 띄어쓰기 성능
Table 1. Performance of word segmentation with varying n-gram language models.

음절 언어모델	어절 단위 정확도 (%)	음절 단위 정확도 (%)
바이그램 (n=2)	78.11	94.08
트라이그램 (n=3)	91.58	96.69
4-그램 (n=4)	91.77	97.17

표 1은 바이그램, 트라이그램, 4-그램을 적용한 경우의 어절단위 및 음절 단위 정확도를 나타낸다. 트라이그램을 사용하는 경우 어절 단위 정확도는 91.58%이며 음절 단위로는 96.69%를 나타내었다. 두 음절만의 정보를 사용하는 경우에 비하여 트라이그램을 사용하는 경우 성능 향상이 두드러지며 4-그램을 사용한 경우는 25 MB의 메모리 크기에 비하여 성능 향상은 크지 않음을 알수 있다. 말뭉치에서 나타난 모든 트라이그램을 사용한 경우 (컷오프 값 0)가 1번 발생하는 트라이그램은 제거한 경우 (컷오프 값 1)보다 0.04% 정도 어절 정확도가 크기 때문에 메모리를 작게 차지하도록 컷오프 값을 1로 선택하였다.

일반적인 텍스트에서의 성능을 조사하기 위하여 고등학교 교과서에서 발췌한 256문장, 3684어절, 24788음절로 이루어진 두 편의 한국어 수필과 번역된 외국 단편 소설 (알퐁스 도데의 '별')에 대하여 테스트하였다. 실험결과 트라이그램을 사용한 경우 어절단위 정확도 91.31%, 음절단위로는 97.93%를 얻었다. 어절단위 정확도가 신문기사보다 낮게 나타난 것은 외국 소설에서 나타나는 외국인명 및 지명에서 오류가 많았기 때문이며, 한국어 수필의 경우는 신문기사보다 띄어쓰기 결과가 더 우수하였다.

전산학 분야의 말뭉치를 사용한 두 음절간 상호 정보를 이용한 기존의 연구 결과에서 형태소 분석기를 사용한 경우에 훈련 영역과 같은 영역의 테스트 문장에 대하여 93.6% (음절단위 98.4%), 다른 영역의 경우 84.7%의 단어 정확도를 나타내었으며 평균적으로 87.2% (음절단위 96.4%)를 나타내었다[9]. 형태소 해석기를 사용하지 않았을 때에는 각각 같은 영역 및 다른 영역의 테스트 문장에 대하여 90.9%와 74.4%의 단어 정확도를 얻었다. 본 논문의 결과는 형태소 해석기 및 어휘를 참조하지 않고 음절 트라이그램만를 사용하여 다른 영역의 테스트 문장에 대하여 약 91.5%의 어절단위 정확도를 나타내므로 기존의 통계적 접근방식보다 우수한 성능을 나타냄을 알 수 있다.

조사/어미로 사용되는 음절의 특성을 이용한 연구에서

입력 텍스트 (띄어쓰기 없음)

1. 지리산연맥을종주하다보면별나게광활한능선평원을만나게된다
2. 잔돌이많이깔렸다해서세석평전인것이다
3. 이평전에이런신화가깃들어있다
4. 지리산정상인천황봉에좌정을한여신마야고는역시지리산반야봉에좌정한남신반야를사랑하여결합될날만을기다린다
5. 그사랑의보금자리로서바로그천황봉이래신방을꾸미고잔돌을바닥에깔고쇠별꽃을만발하게해놓았다는것이다
6. 그세석평전에등산객들을위한우림한산장이일월일일을기해개장했다
7. 신들의신방을인간이가로챈셈이다

띄어쓰기 결과

1. 지리산 연 맥을 종주하다 보면 별나게 광활한 능선 평원을 만나게 된다
2. 잔 돌이 많이 깔렸다 해서 세석평전인 것이다
3. 이평 전에 이런 신화가 깃들어 있다
4. 지리산 정상인 천황봉에 좌정을 한 여신 마야고는 역시 지리산 반야봉에 좌 정한 남신 반야를 사랑하여 결합될 날만을 기다린다
5. 그 사랑의 보금자리로서 바로 그 천황봉 이래 신방을 꾸미고 잔 돌을 바닥에 깔고 쇠별 꽃을 만발하게 해놓았다는 것이다
6. 그 세석평전에 등산객들을 위한 우림한 산장이 일월 일일을 기해 개장했다
7. 신들의 신방을 인간이 가로챈 셈이다

그림 7. 띄어쓰기 예제
Fig. 7. Example of word segmentation.

는 음절단위 정확도 97.3%, 어절 단위 정확도 93.2%로 나타났다[6]. 이 연구에서는 전산관련 논문 및 여러 유형의 문장을 사용하였으나 미등록어 및 외국어의 존재 여부 등이 나타나 있지 않아서 비교하기가 어렵다. 그러나 본 논문에서는 어휘 정보나 형태소 해석을 사용하지 않은 경우의 성능이라는 점을 고려할 때 앞으로 형태소 해석과 결합되면 좋은 결과를 나타낼 것으로 본다.

3.3. 실제 응용에서의 띄어쓰기

말뭉치의 전처리에서 실제로 띄어쓰기 알고리듬이 필요한 경우는 앞절에서와 같이 전혀 띄어쓰기가 되어 있지 않은 경우는 거의 없다. 따라서 실제적인 응용에서는 줄바꿈에서의 오류 또는 긴 어절의 띄어쓰기 고치기에 사용된다. 여기에서는 실제의 경우에서의 띄어쓰기 성능을 조사하기 위하여 텍스트가 주어질 때 정해진 길이 이상의 어절만을 띄어쓰기 하는 경우와 줄바꿈 위치에서만 띄어쓰기 하는 경우의 성능을 조사하였다.

표 2. 띄어쓰기 적용한 최소 어절길이의 변화에 따른 띄어쓰기 정확도
Table 2. Word-based accuracy with varying minimum eojeol lengths for word segmentation.

띄어쓰기 적용한 최소 어절길이 (음절 갯수)	3	4	5	6
어절 단위 정확도 (%)	95.53	95.93	96.00	95.06

표 2는 앞에서 사용한 신문 칼럼 기사를 주어진 길이 이상의 어절만을 띄어쓰기 한 경우의 성능을 나타낸다. 5음절 이상의 어절만을 띄어쓰기하는 것이 성능이 가장 우수하다. 띄어쓰기 알고리듬을 적용하지 않은 경우의 정확도는 96.19%이었다. 이것은 주어진 텍스트의 띄어쓰기 정확도가 이미 96% 이상인 경우에서는 자동 띄어쓰기 알고리듬으로 오류를 감소할 수 없으며 96% 이하의 정확도를 갖는 텍스트에만 유효함을 나타낸다.

다음으로는 HTML문서 처리에서 종종 나타나는 줄바꿈 위치에서의 띄어쓰기 교정에 대한 성능을 조사하였다. 줄바꿈에서의 공백 삽입 문제는 종종 오래된 워드 프로세서를 사용한 텍스트에서 나타난다. 그림 8은 매 줄끝에는 공백이 없는 신문 칼럼의 띄어쓰기 결과를 보여준다. 띄어쓰기 알고리듬은 줄바꿈 위치에 있는 어절에 대해서만 적용된다. 줄바꿈위치에서 무조건 모두 붙여쓰기 한 경우의 띄어쓰기의 어절단위 정확도는 91.00%로 나타났다. 띄어쓰기 알고리듬을 사용하여 줄바꿈 위치에서 띄어쓰는 경우와 붙여쓰는 경우의 정규화된 어절 확률을 계산하여 확률이 높은 경우를 선택하도록 하였다. 붙여쓰기한 경우의 어절 길이가 5 이상인 경우에 대하여 앞의 방법을 적용하는 경우에 가장 성능이 우수하였으며 96.27%의 어절단위 정확도를 나타내었다. 짧은 길이의 어절에 대해서 띄어쓰기 알고리듬을 적용하는 경우에는 오히려 성능이 저하되었다.

입력 텍스트: HTML 문서 (줄바꿈 있음)

1. 지리산 연맥을 종주하다 보면 별나게 광활한 능선 평원을 만나게 된다.
2. 잔 돌이 많이 깔렸다 해서 세석평전인 것이다. 이 평전에 이런 신화가
3. 깃들어 있다. 지리산 정상인 천황봉에 좌정을 한 여신 마야고는 역시 지
4. 리산 반야봉에 좌정한 남신 반야를 사랑하여 결합될 날만을 기다린다.
5. 그 사랑의 보금자리로서 바로 그 천황봉아래 신방을 꾸미고 잔돌을
6. 바닥에 깔고 쇠별꽃을 만발하게 해놓았다는 것이다. 그 세석평전에 등산
7. 객들을 위한 우람한 산장이 일월 일일을 기해 개장했다. 신들의 신방을 인
8. 간이 가로챈 셈이다.

띄어쓰기 결과

1. 지리산 연맥을 종주하다 보면 별나게 광활한 능선 평원을 만나게 된다
2. 잔 돌이 많이 깔렸다 해서 세석평전인 것이다
3. 이 평전에 이런 신화가 깃들어 있다
4. 지리산 정상인 천황봉에 좌정을 한 여신 마야고는 역시 지리산 반야봉에 좌정한 남신 반야를 사랑하여 결합될 날만을 기다린다
5. 그 사랑의 보금자리로서 바로 그 천황봉아래 신방을 꾸미고 잔돌을 바닥에 깔고 쇠별꽃을 만발하게 해놓았다는 것이다
6. 그 세석평전에 등산객들을 위한 우람한 산장이 일월 일일을 기해 개장했다
7. 신들의 신방을 인간이 가로챈 셈이다

그림 8. 줄바꿈에서의 띄어쓰기 예제
Fig. 8. Example of word segmentation at line breaks.

3.4. 복합명사 분리

띄어쓰기 알고리듬을 이용하여 복합명사를 분리하였다. 복합명사는 길이가 4 이상에 대해서만 고려하며 단위명사는 모두 길이가 2 이상이라고 가정한다. 먼저 최소 어절 길이를 2로 제한한 띄어쓰기 알고리듬으로 공백의 위치를 찾은 다음 단위명사가 사전에 존재하는지를 검사한다. 단위명사가 사전에 존재하지 않으면 인접한 단위명사와 병합하여 사전에 존재하는지를 반복적으로 검사한다.

테스트를 위하여 한국일보 신문기사 경제면에서 나타나는 복합명사들을 사용하였다. 단위명사의 중복을 제한하지는 않았다. 400개의 외국어가 포함된 복합어 (주로 외국어 회사명, 지명을 포함)를 포함하여 4음절 이상의 복합어 10376개를 사용하였다. 복합명사는 6,238개의 단위 명사로 구성되었으며 단위명사로 분리하였을 때의 어절 개수는 23713개이었다. 즉 하나의 단위명사는 평균 2.3개의 음절로 이루어졌다. 본 연구에서 사용한 4-12음절 복합명사에서도 인명, 지명, 외국어를 포함한다. 사전의 크기는 66,563개이었으며 미등록어 비율은 5.7%이었다. 400개의 외국어 포함된 복합어의 미등록어 비율은 52.4%이었다. 복합어 분리시에 작은 길이의 단위명사로 분리되는 것을 방지하기 위하여 최고의 성능을 나타내는 $penalty = 6$을 사용하였다. 표 3은 복합명사의 길이에 따른 분리 정확도 (어절단위 정확도)[8]를 나타낸다. 모든 복합명사에 대한 분리 정확도는 96.22%로 나타났다. 5% 정도의 미등록어를 갖는 복합명사에 대해서도 82.0%의 성능을 얻었다.

다른 연구 결과와의 비교를 위해서는 공통의 텍스트를 사용하여야 하지만 한글 정보처리를 위한 공통 말뭉치가 없기 때문에 정확한 비교는 어렵다. 심광섭은 합성된 상호정보를 이용하여 미등록어가 없는 상태에서 복합명사

표 3. 복합명사 띄어쓰기의 성능
Table 3. Performance of compound noun decomposition.

복합명사 길이	복합명사 개수 (비율 %)	분리 정확도 (%)
4	5306 (51.1)	98.7
5	1764 (17.0)	93.6
6	1721 (16.6)	97.1
7	580 (5.6)	93.8
8	350 (3.4)	93.3
9	135 (1.3)	92.0
10	65 (0.6)	93.2
11	31 (0.3)	87.9
12 이상	24 (0.2)	94.6
외국어 포함	400 (3.9)	82.0
전체	10376 (100)	96.22

분리 정확도는 98.6%를 나타내었다[7]. 미등록어가 있을 경우에는 분리 정확도는 90.6%이었고 최장일치법을 사용한 경우 76.1%이었다. 본 연구의 결과가 미등록어율을 고려하였을 때 훨씬 나은 성능을 나타낸다. 최재혁은 복합명사의 길이에 따라서 구성패턴을 파악하고 정해진 순서에 따라 복합명사 분리한다[9]. 4음절 89%, 6음절 83% 8음절 81% 5 음절 78% 7음절 73%로 나타났다. 강승식은 규칙 기반의 방법을 사용하여 97.95%의 정확도를 얻었다[8]. 이것은 음절 분포 (4음절 60.6%, 5음절 21.9%, 6음절 9.9%, 7음절 4.4%, 8음절 1.8%, 9음절 음절 0.8%, 10음절 이상 0.6%)[11]를 고려한 경우 96.57%의 정확도를 얻었다. 본 연구의 경우 앞의 음절 분포를 고려하면 외국어 복합어 400개를 제외한 경우 97.01%의 정확도를 나타내어 다른 방법에 비해 우수함을 보여준다. 이전의 연구결과와 마찬가지로 본 연구에서도 5음절의 정확도가 감소하는 현상이 나타났다.

IV. 결 론

대규모 말뭉치에서의 띄어쓰기 오류 교정을 위한 자동 띄어쓰기 알고리듬을 제안하였다. 제안된 알고리듬에서는 한국어 문장이 마코프 체인으로부터 발생한다고 가정하고 최대 정규 문장확률을 갖는 경로를 찾음으로써 최적의 공백 삽입 위치를 결정하였다. 미등록어가 많이 존재하는 신문 칼럼으로부터 추출한 254문장을 사용하여 알고리듬을 테스트하였다. 전혀 공백이 없는 문장의 띄어쓰기에 적용한 결과 91.58% 어절 정확도를 나타냈다. 이것은 음절단위로는 96.69%의 정확도에 해당한다. 다음으로는 줄바꿈에서의 띄어쓰기 성능을 조사하였다. 길이가 5 이상인 어절에 대해서만 띄어쓰기 알고리듬을 적용한 경우에 96.27%의 어절 정확도를 나타내었다. 제안 알고리듬을 복합명사의 분리에 적용하여 성능 테스트를 한 결과 96.22%의 정확도를 얻었다. 이 결과는 기존의 통계 기반 방식보다 우수하며 규칙 기반 접근 방식보다 우수하거나 근접하는 결과이다. 그러나 이 결과는 아직도 수작업 띄어쓰기 정확도에 미치지 못한다. 성능 개선을 위하여 형태소 해석과의 결합 또는 언어적 지식을 활용하는 방안에 대한 연구가 필요하다.

이 알고리듬은 대어휘 연속음성인식을 위한 언어모델 계산에 사용되는 대규모 말뭉치의 정규화 과정의 하나로서 수작업으로 행해지던 띄어쓰기 교정작업의 반자동화에 기여할 수 있으며, 텍스트의 띄어쓰기 자동 검사 및 교정, 한국어 정보검색을 위한 색인어 처리, 문자인식에 의하여 대량의 텍스트 입력시 줄바꿈 위치에서의 공백 처리에 유용할 것이다.

감사의 글

이 연구는 과학기술부가 지원하는 뇌과학연구개발사업의 일부로서 이루어진 것입니다.

참 고 문 헌

1. R. Bakis, S. Chen, P. Gopalakrishnan, R. Gopinath, S. Maes, L. Polymenakos, and M. Franz, "Transcription of broadcast news shows with the IBM large vocabulary speech recognition system," *Proc. 1997 DARPA Speech Recognition Workshop*, Feb. 1997.
2. J. L. Gauvain, L. Lamel, G. Adda, M. Jardino, "The LIMSI 1998 HUB-4E Transcription system," *Proc. DARPA Broadcast News Transcription*, Feb. 1999.
3. O. W. Kwon, K. Hwang, J. Park, "Korean large vocabulary continuous speech recognition using pseudomorpheme units," *Proc. EUROSPEECH'99*, Budapest, Hungary, Sept. 1999.
4. P. Geutner, "Using morphology towards better large-vocabulary speech recognition systems," *Proc. ICASSP'95*, Detroit, USA, May 1995.
5. 심광섭, "음절간 상호 정보를 이용한 한국어 자동 띄어쓰기," 정보과학회논문지(B), 23 (9), 991-1000, 9, 1996.
6. 강승식, "한글 문장의 자동 띄어쓰기," 제10회 한글 및 한국어 정보처리 학술대회논문집, 137-142, 1998.
7. 심광섭, "합성된 상호 정보를 이용한 복합 명사 분리," 정보과학회논문지(B), 24 (11), 1307-1317, 11, 1997.
8. 강승식, "한국어 복합명사 분해 알고리즘," 정보과학회논문지(B) 25 (1), 172-182, 1, 1998.
9. 최재혁, "음절수에 따른 한국어 복합명사 분리 방안," 제8회 한글 및 한국어 정보처리 학술발표대회논문집, 262-267, 1996.
10. 강승식, "한국어 형태소 분석을 위한 복합 명사의 인식 방법," 인지과학회 춘계학술발표논문집, 175-189, 1993.
11. 윤보현, 임희석, 임해창, "통계정보를 이용한 한국어 복합명사의 분석 방법," 한국정보과학회 봄학술발표논문집, 925-928, 1995.
12. K. H. Chen, S. H. Liu, "Word identification for Mandarin Chinese sentence," *Proc. 14th Int. Conf. Computational Linguistics*, 101-107, 1992.
13. T. Hisamitsu, Y. Nitta, "Analysis of Japanese compound nouns by direct text scanning," *Proc. 16th Int. Conf. Computational Linguistics*, 550-555, 1996.
14. T. Pachunke, O. Mertineit, K. Wothke, R. Schimidt, "Broad coverage automatic morphological segmentation of German words," *Proc. 14th Int. Conf. Computational Linguistics*, 1218-1222, 1992.
15. A. Popoulis, Porbability, *Random Variables, and Stochastic Processes*, McGraw-Hill, 1984.
16. W. Feller, *An Introducion to Probability Theory and Its Applications*, vol. I, 3rd ed. Princeton Univ., 1970.
17. L. R. Rabiner, B. H. Juang, *Fundamentals of Speech*

Recognition, Prentice Hall, 1993.
18. P. Clarkson, R. Rosenfeld, "Statistical language modeling using the CMU-Cambridge toolkit," *Proc. EUROSPEECH'97*, 2707-2710, 1997.
19. S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 35, 400-401, 1987.

저자 약력

● 권 오 욱 (Oh-Wook Kwon)

1986년 2월: 서울대학교 전자공학과 학사
1988년 2월: 한국과학기술원 전기 및 전자공학과 석사
1997년 2월: 한국과학기술원 전기 및 전자공학과 박사
1988년 3월~2000년 4월: 한국전자통신연구원 책임연구원
2000년 5월~2001년 3월: 한국과학기술원 정보전자연구소 연구교수
2001년 3월~현재: University of California, San Diego 포스트닥 연구원
※ 주관심분야: 음성인식, 음성신호처리, 패턴인식, 언어처리