한글 문장의 자동 띄어쓰기를 위한 두 가지 통계적 모델

(Two Statistical Models for Automatic Word Spacing of Korean Sentences)

이 도 길 † 이 상 주 †† 임 희 석 ††† 임 해 창 ††††
(Do-Gil Lee) (Sang-Zoo Lee) (Heui-Seok Lim) (Hae-Chang Rim)

요 약 자동 띄어쓰기는 문장 내에서 잘못 띄어쓴 어절들을 올바르게 복원하는 과정으로서, 독자에게 글의 가독성을 높이고 문장의 뜻을 정확히 전달하기 위해 매우 중요하다.

기존의 통계 기반 자동 띄어쓰기 접근 방법들은 이전 띄어쓰기 상태를 고려하지 않기 때문에 잘못된 확률 정보에 의한 띄어쓰기를 할 수밖에 없었다. 본 논문에서는 기존의 통계 기반 접근 방법의 문제점을 해결할 수 있는 두 가지 통계적 띄어쓰기 모델을 제안한다. 제안하는 모델은 자동 띄어쓰기를 품사 부착과 같은 분류 문제(classification problem)로 간주할 수 있다는 착안에 기반하며, 은닉 마르코프 모델을 일반화함으로써 확장된 문맥을 고려할 수 있고 보다 정확한 확률을 추정할 수 있도록 고안되었다.

제안하는 모델과 지금까지 가장 좋은 성능을 보이는 기존의 방법을 비교하기 위해 여러 가지 실험 조건에 따른 다양한 실험을 수행하였고, 오류에 대한 자세한 분석을 제시하고 있다. 제안하는 모델을 복합 명사를 고려하는 평가 방식에 적용한 실험 결과, 98.33%의 음절 단위 정확도와 93.06%의 어절 단위 정확률을 얻었다.

키워드 : 자동 띄어쓰기, 확률 모델, 은닉 마르코프 모델

Abstract Automatic word spacing is a process of deciding correct boundaries between words in a sentence including spacing errors. It is very important to increase the readability and to communicate the accurate meaning of text to the reader.

The previous statistical approaches for automatic word spacing do not consider the previous spacing state, and thus can not help estimating inaccurate probabilities.

In this paper, we propose two statistical word spacing models which can solve the problem of the previous statistical approaches. The proposed models are based on the observation that the automatic word spacing is regarded as a classification problem such as the POS tagging. The models can consider broader context and estimate more accurate probabilities by generalizing hidden Markov models.

We have experimented the proposed models under a wide range of experimental conditions in order to compare them with the current state of the art, and also provided detailed error analysis of our models.

The experimental results show that the proposed models have a syllable-unit accuracy of 98.33% and Eojeol-unit precision of 93.06% by the evaluation method considering compound nouns.

Key words : Automatic word spacing, Probabilistic model, hidden Markov model

† 학생회원 : 고려대학교 컴퓨터학과 학사
dglee@nlp.korea.ac.kr
†† 종신회원 : (주)엔엘피솔루션 대표이사
zoo@nlpsolution.com
††† 종신회원 : 천안대학교 정보통신학부 교수
limhs@infocom.chonan.ac.kr
†††† 종신회원 : 고려대학교 컴퓨터학과 교수
rim@nlp.korea.ac.kr
논문접수 : 2002년 7월 10일
심사완료 : 2002년 12월 27일

1. 서 론

자동 띄어쓰기는 문장 내에서 잘못 띄어쓴 어절들을 올바르게 복원하는 과정이다. 한국어의 경우, 띄어쓰기는 독자에게 글의 가독성을 높이고 문장의 뜻을 정확히 전달하기 위해 매우 중요하다. 예를 들어, "아버지가 방에 들어가셨다(Father entered the room)"를 "아버지가방에 들어가셨다(Father entered the bag)"와 같이 띄어쓰면 전혀 다른 의미의 문장이 된다.

정보의 주요 수집원이 되는 인터넷상의 문서에는 많은 띄어쓰기 오류가 존재하며 이러한 문서를 제대로 처리하기 위해서는 자동 띄어쓰기 시스템이 반드시 필요하다. 뿐만 아니라 자동 띄어쓰기 시스템은 자연어처리 응용 시스템의 가장 기본이 되는 형태소 분석기의 전처리기, 문자인식기가 인식한 문서의 줄경계를 복원하기 위한 후처리기, 음성인식기로부터 생성된 연속음절 문장을 올바르게 띄어쓰기 위한 후처리기, 맞춤법 검사기의 한 모듈로서도 중요한 역할을 한다.

본 논문에서는 띄어쓰기 문제를 품사 부착 문제와 같은 분류 문제(classification problem)로 간주하고자 하며, 분류 문제를 해결하기 위해 지금까지 다양한 기계학습 기법이 개발되어 왔다.

이 중에서 은닉 마르코프 모델(hidden Markov model; 이하 HMM)은 품사부착[1,2,3,4], 정보추출[5], 개체명 인식[6], 외래어 추출[7] 등과 같은 자연어처리의 여러 문제를 해결하는 데에 많이 사용되는 모델이며 각 분야에서 높은 성능을 보이고 있다. HMM의 학습은 학습 말뭉치를 사용하는 지도 학습과 학습 말뭉치를 사용하지 않는 자율 학습이 있다. 일반적으로 적은 양의 학습 말뭉치를 사용하는 것이 많은 양의 원시 말뭉치를 사용하는 것보다 성능이 좋은 것으로 알려져 있다.

띄어쓰기 문제에서는 학습을 위해 따로 말뭉치를 구축할 필요가 없이 이미 존재하는 원시 말뭉치를 학습 말뭉치로 사용할 수 있다. 따라서 HMM이 띄어쓰기 문제에도 효과적으로 적용될 수 있을 것으로 기대된다. 본 논문은 띄어쓰기 문제에 적합하도록 HMM을 일반화하여 확장된 문맥을 고려할 수 있는 통계적 모델을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 자동 띄어쓰기 연구와 관련된 기존의 연구에 대해 논하고, 3장에서는 HMM에 기반한 품사 부착 모델을 설명한다. 4장에서는 제안하는 두 가지 통계적 자동 띄어쓰기 모델을 설명하고, 5장에서는 실험 및 평가, 마지막으로 6장에서 결론을 맺는다.

2. 관련 연구

자동 띄어쓰기와 관련해서 기존의 연구들은 입력 대상 문서에 따라 띄어쓰기가 부분적으로 잘못되어 있는 문서를 대상으로 한 경우[8,9]와 띄어쓰기가 전혀 되어 있지 않은 문서를 대상으로 한 경우[10,11,12,13,14,15,16,17]로 나눌 수 있다. 대부분의 기존 연구들은 후자에 해당된다. 후자는 부분적으로 띄어쓰기 오류가 있는 문서가 입력되더라도 모두 붙여쓰거나 띄어쓴 후에 일괄적으로 교정을 할 수 있으므로 전자에 비해 처리할 수 있는 대상 문서의 범위가 넓다. 그러나 입력 문장에서 올바르게 띄어쓴 부분을 잘못 수정할 수도 있다. 전자는 주어진 문장의 띄어쓰기 상태가 어느 정도 신뢰할 만한지를 시스템이 스스로 판단할 수 없으므로 동일한 내용의 문서라도 그 문서의 띄어쓰기 상태에 따라 생성되는 결과가 달라질 수 있다. 결국 입력 당시의 띄어쓰기 상태와 그 상태를 고려할 것인 지의 여부에 따라 각각 장단점이 있다. 본 논문에서는 띄어쓰기가 전혀 되어 있지 않은 문서를 대상으로 하고 있다.

자동 띄어쓰기의 방법은 접근 방법에 따라 크게 규칙 기반 방식과 통계 기반 방식으로 나눌 수 있다. 규칙 기반 방식은 사전 정보, 조사/어미 정보, 띄어쓰기 용례 사전 등과 같은 어휘 지식과 최장 일치나 최단 일치, 형태소 분석 규칙, 띄어쓰기 오류 유형 등과 같은 휴리스틱 규칙을 이용한다[9,10,11,12]. 이 방식은 통계 기반 방식에 비해 분석 과정이 복잡하고 어휘 지식의 구축과 관리에 있어서 많은 비용이 든다. 따라서 시스템의 구축과 유지, 보수가 어렵다. 이 방식은 어절의 경계를 인식하기 위해 대부분 형태소 분석기를 사용하는데, 형태소 분석기를 사용하여 어절의 경계를 인식하는 경우 다음과 같은 단점이 있다. 첫째, 중의적인 분석이 가능할 때 하나를 선택하여 탐색을 하다 문제가 발생했을 때 빈번하게 역행(backtracking)을 해야 하고, 한 번 잘못 분석한 부분으로 인해 다음 부분이 계속해서 영향을 받게 되는 전파 오류(triggered error)가 빈번하게 발생한다. 둘째, 분석 결과가 형태소 분석기에 매우 의존적이라는 점이다. 형태소 분석에 성공한 어절이라도 잘못 띄어 써진 어절일 가능성이 있고(형태소 분석의 과분석된 경우), 미등록어로 인해 분석에 실패하는 경우에도 문제가 발생한다(형태소 분석의 미분석된 경우). 특히, 전자의 경우에는 오류인 지의 여부도 알아낼 수 없다. 마지막으로, 만약 띄어쓰기 시스템이 형태소 분석기의 전처리로서 사용되는 경우라면 동일한 연산을 이중으로 해야하는 부담이 있다. 통계 기반 방식은 말뭉치로부터 인접한 두 음절간

의 띄어 쓸 또는 붙여 쓸 확률을 학습하여 이를 이용하여 띄어쓰기 오류를 교정하는 방식이다[13,14,15,1617]. 이 방식은 대량의 원시 말뭉치로부터 자동으로 음절 정보를 획득할 수 있으므로 어휘 지식이나 규칙을 작성하거나 유지, 보수에 드는 비용이 필요하지 않고, 형태소 분석기를 사용할 때와는 달리 문장에 존재하는 미등록어에 대해서도 견고한 분석이 가능하다. 그러나 학습 말뭉치와 유사한 종류의 문서에 대해서는 비교적 성능이 좋지만, 그렇지 않은 경우에는 낮은 성능을 보임으로써 학습 말뭉치의 영향을 크게 받고, 정확도 향상에 한계가 있다고 알려져 있다.

통계기반 방식의 한 예로 강승식(2001)[17]의 방법을 설명하면 다음과 같다. 이 방법은 임의의 두 음절 x_i와 x_{i+1}사이에 공백이 삽입될 확률 $P(x_i, x_{i+1})$을 식 (1)과 같이 계산하여 이 값이 경험적 임계치(threshold) 0.375를 넘으면 두 음절 사이에 공백을 삽입하는 방식이다.

$$P(x_i, x_{i+1}) = 0.25 \times P_R(x_{i-1}, x_i) + 0.5 \times P_M(x_i, x_{i+1}) + 0.25 \times P_L(x_{i+1}, x_{i+2}) \quad (1)$$

식 (1)에서 P_R은 두 음절의 오른쪽에 공백이 있을 확률, P_M은 두 음절의 중간에 공백이 있을 확률, P_L은 두 음절의 왼쪽에 공백이 있을 확률을 나타내고 각각 다음과 같이 계산된다.

$$P_R(x_{i-1}, x_i) = \frac{freq(x_{i-1}, x_i, SPACE)}{freq(x_{i-1}, x_i)}$$
$$P_M(x_i, x_{i+1}) = \frac{freq(x_i, SPACE, x_{i+1})}{freq(x_i, x_{i+1})} \quad (2)$$
$$P_L(x_{i+1}, x_{i+2}) = \frac{freq(SPACE, x_{i+1}, x_{i+2})}{freq(x_{i+1}, x_{i+2})}$$

위의 각 식에서 $freq(x)$는 학습 말뭉치로부터 문자열 x가 나타난 빈도를 의미하고, $SPACE$는 공백을 의미한다.

기존의 통계 기반 방식들도 이와 비슷하게 두 음절의 좌측, 중간, 우측에 공백이 나타난 빈도[1]에 따라 확률값 또는 상호정보를 구하고, 각 수치에 가중치를 주어 선형조합(linear combination)한 결과를 미리 정한 임계치와 비교하여 해당 음절의 띄어쓰기 여부를 판정한다[13,15]. 그런데 기존의 방법들은 이전 어절의 띄어쓰기 상태를 고려하지 않는 모델상의 문제점이 있다. 예를 들어, 문장 "공부할수있다"에서 두 음절 "수"와 "있"의 사이를 띄어 쓸 확률은 다음과 같다.

$$P(수, 있) = 0.25 \times P_R(할, 수) \times 0.5 \times P_M(수, 있) \times 0.25 \times P_L(있, 다) \quad (3)$$

1) 이러한 빈도를 음절 bigram이라 부른다.

여기서, $P_R(할, 수)$의 계산은 다음과 같이 한다.

$$P_R(할, 수) = \frac{freq(할, 수, SPACE)}{freq(할, 수)} \quad (4)$$

그러나 "할"과 "수" 사이를 띄어 쓰는 것이 올바르기 때문에 $freq(할, 수, SPACE)$ 대신 $freq(SPACE, 수, SPACE)$를 써야한다. 이러한 현상은 이전 띄어쓰기 상태를 고려하지 않기 때문에 발생한다. 물론, 바로 이전에 결정한 띄어쓰기 상태를 고려할 수도 있겠지만 그렇게 하게 되면 전파 오류가 발생할 수 있다. 결국 전파 오류를 막으려면 문장으로부터 생성 가능한 모든 띄어쓰기 상태로부터 최적의 띄어쓰기 상태를 결정해야 한다. 본 논문에서 제안하는 모델은 이전 띄어쓰기 상태를 고려함으로써 기존의 통계 기반 방법의 문제점을 해결하고자 한다.

3. HMM에 기반한 통계적 품사 부착 모델

HMM이 가장 많이 적용된 분야는 자동 품사 부착이다. 본 논문에서는 띄어쓰기 문제를 품사 부착 문제와 같이 분류 문제(classification problem)로 간주하여 해결하고자 한다. 제안하는 띄어쓰기 모델을 설명하기에 앞서 본 장에서는 품사 부착 모델을 설명하고자 한다.

품사 부착 함수, $\Gamma(W)$는 문장 내에 주어진 단어열 $W=(w_1, w_2, .., w_n)$에 대해 최적의 품사열 $T=(t_1, t_2, .., t_n)$를 찾는 것으로 식 (5)처럼 정의할 수 있다.

$$\Gamma(W) \stackrel{def}{=} argmax_T P(T \mid W) \quad (5)$$

식 (5)는 베이즈 정리(Bayes' rule)에 의해 다음과 같이 유도된다.

$$argmax_T \frac{P(T) \cdot P(W \mid T)}{P(W)} \quad (6)$$

여기서, 분모는 영향을 미치지 않으므로 아래 식과 같다.

$$argmax_T P(T) \cdot P(W \mid T) \quad (7)$$

결국, 다음과 같은 식이 유도된다.

$$argmax_T P(T \mid W) = argmax_T P(T) \cdot P(W \mid T) = argmax_T P(T, W) \quad (8)$$

위의 식은 품사열 T와 단어열 W의 결합 확률 $P(T, W)$를 최대로 하는 품사열 T를 찾는 것이다. 식을 전개하면,

$$P(T, W) = P(t_{1,n}, w_{1,n}) = \prod_{i=1}^{n} P(t_i \mid t_{1,i-1}, w_{1,i-1}) P(w_i \mid t_{1,i}, w_{1,i-1}) \approx \prod_{i=1}^{n} P(t_i \mid t_{i-K,i-1}) P(w_i \mid t_i) \quad (9)$$

과 같이 된다.

위의 식에서 사용된 마르코프 가정은 "품사의 발생은 이전 K개의 품사에만 의존한다"와 "단어의 발생은 해당

단어의 품사에만 의존한다"이다.

이 식에서 $P(t_i|t_{i-K,i-1})$을 품사 발생 확률 또는 전이 확률(transition probability)이라고 하고, $P(w_i|t_i)$를 어휘 발생 확률 또는 어휘 확률(lexical probability)이라고 한다. 여기서 K의 값이 얼마냐에 따라서 모델의 종류가 달라지는데, K가 클수록 더 많은 품사 문맥을 고려할 수 있으나, 자료부족 문제(data sparseness problem)로 인해 확률값을 구하기가 어렵기 때문에, 일반적으로 K가 1인 bigram 모델과 2인 trigram 모델을 많이 사용한다.

4. 통계적 자동 띄어쓰기 모델

띄어쓰기 문제도 품사 부착 문제와 유사하게 접근할 수 있다. 띄어쓰기 모델은 문장 내에 주어진 음절열 $S=(s_1, s_2, \ldots, s_n)$에 대해 최적의 띄어쓰기 태그열 $T=(t_1, t_2, \ldots, t_n)$를 찾는다.

$$argmax_T\ P(T|S) \tag{10}$$

띄어쓰기 태그는 해당 음절과 다음 음절의 사이를 띄어쓸 것인가 붙일 것인가에 대한 태그로서 이진 값 0 또는 1을 갖는다. 0은 해당 음절과 다음 음절을 붙여쓰라는 태그이고, 1은 띄어쓰라는 태그이다. 예를 들어 문장 "학교에서 공부를 참 열심히 합니다."의 경우 띄어쓰기 태그가 부착된 형태로 표기하면 "학/0 교/0 에/0 서/1 공/0 부/0 를/1 참/1 열/0 심/0 히/1 합/0 니/0 다/0 ./1"와 같다. 결국 띄어쓰기 모델은 품사 부착 모델에서의 식 (8)과 같이 띄어쓰기 태그열 T와 음절열 S의 결합 확률 $P(T,S)$를 최대로 하는 띄어쓰기 태그열 T를 찾는 것이다.

여기서 띄어쓰기 태그를 먼저 전개하느냐 음절을 먼저 전개하느냐에 따라 두 가지로 식을 전개할 수 있다. 연쇄 규칙(chain rule)에 의해 띄어쓰기 태그를 먼저 전개하면 식 (11)과 같다.

$$\begin{aligned}
P(T,S) &= P(t_{1,n}, s_{1,n}) \\
&= P(t_1)P(s_1|t_1)P(t_2|t_1, s_1)P(s_2|t_{1,2}, s_1) \\
&\quad P(t_3|t_{1,2}, s_{1,2})P(s_3|t_{1,3}, s_{1,2})\cdots \\
&\quad P(t_n|t_{1,n-1}, s_{1,n-1})P(s_n|t_{1,n}, s_{1,n-1}) \\
&= P(t_1)P(s_1|t_1)\prod_{i=2}^{n} P(t_i|t_{1,i-1}, s_{1,i-1})\cdot \\
&\quad P(s_i|t_{1,i}, s_{1,i-1}) \\
&\approx \prod_{i=1}^{n} P(t_i|t_{i-K,i-1}, s_{i-J,i-1})\cdot \\
&\quad P(s_i|t_{i-L,i}, s_{i-J,i-1})
\end{aligned} \tag{11}$$

음절이 먼저 발생하도록 전개를 하면 식 (12)와 같다.

$$\begin{aligned}
P(T,S) &= P(t_{1,n}, s_{1,n}) \\
&= P(s_1)P(t_1|s_1)P(s_2|s_1, t_1)P(t_2|s_{1,2}, t_1) \\
&\quad P(s_3|s_{1,2}, t_{1,2})P(t_3|s_{1,3}, t_{1,2})\cdots \\
&\quad P(s_n|s_{1,n-1}, t_{1,n-1}) \\
&\quad P(t_n|s_{1,n}, t_{1,n-1}) \\
&= P(s_1)P(t_1|s_1)\prod_{i=2}^{n} P(s_i|s_{1,i-1}, t_{1,i-1})\cdot \\
&\quad P(t_i|s_{1,i}, t_{1,i-1}) \\
&\approx \prod_{i=1}^{n} P(s_i|s_{i-K,i-1}, t_{i-J,i-1})\cdot \\
&\quad P(t_i|s_{i-L,i}, t_{i-I,i-1})
\end{aligned} \tag{12}$$

식 (11)에서 사용된 마르코프 가정은 "띄어쓰기 태그의 발생은 이전 K 개의 띄어쓰기 태그와 이전 J 개의 음절에만 의존한다"와 "음절의 발생은 이전 L 개의 띄어쓰기 태그와 현재 띄어쓰기 태그, 이전 I 개의 음절에만 의존한다"이다.

이에 대한 표기는 $\Lambda(T_{(K:J)}, S_{(L:I)})$와 같이 하기로 한다. 품사부착을 위한 마르코프 모델과 유사하게 확률 $P(t_i|t_{i-K,i-1}, s_{i-J,i-1})$을 띄어쓰기 태그 발생 확률 또는 태그 발생 확률, 확률 $P(s_i|t_{i-L,i}, s_{i-I,i-1})$을 음절 발생 확률이라고 한다. 기존의 HMM에 기반한 품사부착 모델은 품사 발생 확률은 이전 품사열만을, 어휘 발생 확률은 현재 품사만을 의존하는 마르코프 가정을 사용하는 데 비해, 띄어쓰기 모델은 품사부착 모델보다 완화된 마르코프 가정을 사용하여 확장된 문맥을 사용할 수 있도록 모델을 정의하고 있다.

식 (12)에서 사용된 마르코프 가정은 "음절의 발생은 이전 K 개의 음절과 이전 J개의 띄어쓰기 태그에만 의존한다"와 "태그의 발생은 이전 L개의 음절과 현재 음절, 이전 I개의 띄어쓰기 태그에만 의존한다"이다. 이에 대한 표기는 식 (11)과 구분하기 위하여 $\Phi(S_{(K:J)}, T_{(L:I)})$와 같이 표기하기로 한다. 식 (11)에서는 문장의 첫 띄어쓰기 태그가 발생하고, 이 띄어쓰기 태그에 의해 첫 음절이 발생한다는 가정을 사용하고 있으나, 식 (12)에서는 문장의 첫 음절이 발생하고, 이 음절에 의해 첫 띄어쓰기 태그가 발생한다는 가정에 기반하고 있다. 앞장에서 설명한 HMM은 식 (11)에 포함된다. 직관적으로 음절을 먼저 전개하는 식 (12)가 더 타당해 보이며, 각각의 식에 대한 모델의 성능 평가는 5장에서 제시한다. 본 논문에서는 식 (11)을 태그 우선 모델, 식 (12)

표 1 모델의 종류와 모델식의 예

모델 종류	모델식
$\Lambda(T_{(1:0)}, S_{(0:0)})$	$\prod_{i=1}^{n} P(t_i\|t_{i-1}) \cdot P(s_i\|t_i)$
$\Lambda(T_{(1:1)}, S_{(0:1)})$	$\prod_{i=1}^{n} P(t_i\|t_{i-1}, s_{i-1}) \cdot P(s_i\|t_i, s_{i-1})$
$\Lambda(T_{(1:1)}, S_{(1:1)})$	$\prod_{i=1}^{n} P(t_i\|t_{i-1}, s_{i-1}) \cdot P(s_i\|t_{i-1,i}, s_{i-1})$
$\Lambda(T_{(1:2)}, S_{(1:2)})$	$\prod_{i=1}^{n} P(t_i\|t_{i-1}, s_{i-2,i-1}) \cdot P(s_i\|t_{i-1,i}, s_{i-2,i-1})$
$\Lambda(T_{(2:2)}, S_{(2:2)})$	$\prod_{i=1}^{n} P(t_i\|t_{i-2,i-1}, s_{i-2,i-1}) \cdot P(s_i\|t_{i-2,i}, s_{i-2,i-1})$
$\Phi(S_{(1:0)}, T_{(0:0)})$	$\prod_{i=1}^{n} P(s_i\|s_{i-1}) \cdot P(t_i\|s_i)$
$\Phi(S_{(1:1)}, T_{(0:1)})$	$\prod_{i=1}^{n} P(s_i\|s_{i-1}, t_{i-1}) \cdot P(t_i\|s_i, t_{i-1})$
$\Phi(S_{(1:1)}, T_{(1:1)})$	$\prod_{i=1}^{n} P(s_i\|s_{i-1}, t_{i-1}) \cdot P(t_i\|s_{i-1,i}, t_{i-1})$
$\Phi(S_{(1:2)}, T_{(1:2)})$	$\prod_{i=1}^{n} P(s_i\|s_{i-1}, t_{i-2,i-1}) \cdot P(t_i\|s_{i-1,i}, t_{i-2,i-1})$
$\Phi(S_{(2:2)}, T_{(2:2)})$	$\prod_{i=1}^{n} P(s_i\|s_{i-2,i-1}, t_{i-2,i-1}) \cdot P(t_i\|s_{i-2,i}, t_{i-2,i-1})$

를 음절 우선 모델이라고 부르기로 한다.

두 모델 모두 K,J,L,I 의 값을 크게 할수록 더 많은 문맥을 고려할 수 있으나, 자료부족 문제(data sparseness problem)와 모델의 파라미터의 수가 급격히 증가하는 것을 방지하기 위해서 적절한 값을 선택하는 것이 중요하다. 논문에서는 이들의 값을 다음과 같은 범위로 제한하고 있다.

$$0 \le K,J,L,I \le 2$$

따라서 총 $3\times3\times3\times3=81$ 개의 모델이 가능하나 본 논문에서는 $(K,J)=(0,0)$인 경우는 사용하지 않으므로, 실제로는 72 개의 모델을 사용한다. 아직은 이들 중에서 어떤 모델의 성능이 가장 좋은지는 알 수 없으므로 실험을 통해 판단할 수밖에 없다. 여러 가지 모델의 종류와 각 모델식의 예는 표 1에 있다.

각 확률은 학습 말뭉치에 나타난 빈도로부터 최우추정(Maximum likelihood estimation)에 의해 계산한다. 표 2는 음절 발생 확률과 태그 발생 확률을 구하는 예를 보여준다.

두 가지 띄어쓰기 모델 $\Lambda(T_{(1:1)}, S_{(0:1)})$과 $\Phi(S_{(1:1)}, T_{(0:1)})$이 문장 "공부할수있다."를 "공부할 수 있다."와 같이 띄어쓰기 교정할 확률을 계산해 보자. "공부할 수 있다."를 띄어쓰기 태그가 부착된 형태로 표기하면 "공/0+부/0+할/1+수/1+있/0+다/0+./1"가 되고 주어진 문장을 띄어쓸 확률은 각각 다음과 같이 구할 수 있다.

$$\begin{aligned} P(T,S) &= \Lambda(T_{(1:1)}, S_{(0:1)}) \\ &= P(t_1=0\|s_0=\$, t_0=1) \cdot \\ &\quad P(s_1=\text{공}\|s_0=\$, t_1=0) \\ &\times P(t_2=0\|s_1=\text{공}, t_1=0) \cdot \\ &\quad P(s_2=\text{부}\|s_1=\text{공}, t_2=0) \\ &\times P(1\|\text{부}0) \cdot P(\text{할}\|\text{부}1) \times P(1\|\text{할}1) \cdot P(\text{수}\|\text{할}1) \\ &\times P(0\|\text{수}1) \cdot P(\text{있}\|\text{수}0) \times P(0\|\text{있}0) \cdot P(\text{다}\|\text{있}0) \\ &\times P(1\|\text{다}0) \cdot P(.\|\text{다}1) \end{aligned} \tag{13}$$

$$\begin{aligned} P(T,S) &= \Phi(S_{(1:1)}, T_{(0:1)}) \\ &= P(s_1=\text{공}\|s_0=\$, t_0=1) \cdot \\ &\quad P(t_1=0\|t_0=1, s_1=\text{공}) \\ &\times P(s_2=\text{부}\|s_1=\text{공}, t_1=0) \cdot \\ &\quad P(t_2=0\|t_1=0, s_2=\text{부}) \\ &\times P(\text{할}\|\text{부}0) \cdot P(1\|0\text{할}) \times P(\text{수}\|\text{할}1) \cdot P(1\|1\text{수}) \\ &\times P(\text{있}\|\text{수}1) \cdot P(0\|1\text{있}) \times P(\text{다}\|\text{있}0) \cdot P(0\|0\text{다}) \\ &\times P(.\|\text{다}0) \cdot P(1\|0.) \end{aligned} \tag{14}$$

표 2 음절 발생 확률과 태그 발생 확률 계산의 예

모델 종류	확률	확률식
$\Lambda(T_{(1:2)}, S_{(1:2)})$	태그	$P_{MLE}(t_i \mid t_{i-1}, s_{i-2,i-1}) = \frac{freq(s_{i-2}, t_{i-1}, s_{i-1}, t_i)}{freq(s_{i-2}, t_{i-1}, s_{i-1})}$
	음절	$P_{MLE}(s_i \mid t_{i-1,i}, s_{i-2,i-1}) = \frac{freq(s_{i-2}, t_{i-1}, s_{i-1}, t_i, s_i)}{freq(s_{i-2}, t_{i-1}, s_{i-1}, t_i)}$
$\Lambda(T_{(2:2)}, S_{(2:2)})$	태그	$P_{MLE}(t_i \mid t_{i-2,i-1}, s_{i-2,i-1}) = \frac{freq(t_{i-2,}s_{i-2}, t_{i-1}, s_{i-1}, t_i)}{freq(t_{i-2,}s_{i-2}, t_{i-1}, s_{i-1})}$
	음절	$P_{MLE}(s_i \mid t_{i-2,i}, s_{i-2,i-1}) = \frac{freq(t_{i-2,}s_{i-2}, t_{i-1}, s_{i-1}, t_i, s_i)}{freq(t_{i-2,}s_{i-2}, t_{i-1}, s_{i-1}, t_i)}$
$\Phi(S_{(1:2)}, T_{(1:2)})$	음절	$P_{MLE}(s_i \mid s_{i-1}, t_{i-2,i-1}) = \frac{freq(t_{i-2}, s_{i-1}, t_{i-1}, s_i)}{freq(t_{i-2}, s_{i-1}, t_{i-1})}$
	태그	$P_{MLE}(t_i \mid s_{i-1,i}, t_{i-2,i-1}) = \frac{freq(t_{i-2}, s_{i-1}, t_{i-1}, s_i, t_i)}{freq(t_{i-2}, s_{i-1}, t_{i-1}, s_i)}$
$\Phi(S_{(2:2)}, T_{(2:2)})$	음절	$P_{MLE}(s_i \mid s_{i-2,i-1}, t_{i-2,i-1}) = \frac{freq(s_{i-2,}t_{i-2}, s_{i-1}, t_{i-1}, s_i)}{freq(s_{i-2,}t_{i-2}, s_{i-1}, t_{i-1})}$
	태그	$P_{MLE}(t_i \mid s_{i-2,i}, t_{i-2,i-1}) = \frac{freq(s_{i-2,}t_{i-2}, s_{i-1}, t_{i-1}, s_i, t_i)}{freq(s_{i-2,}t_{i-2}, s_{i-1}, t_{i-1}, s_i)}$

여기서, "$"는 문장의 시작을 나타내는 의사(pseudo) 음절로서 띄어쓰기 태그는 항상 1이다.[2]

주어진 문장에 대해 최적의 띄어쓰기 태그열은 HMM에서와 같이 동적 알고리즘인 Viterbi 알고리즘을 이용하여 효율적으로 구할 수 있다.

5. 실험 및 평가

5.1 실험 및 평가 말뭉치

학습 말뭉치는 98년과 99년 세종계획 원시 말뭉치[18,19]를 사용하였다. 이 말뭉치는 약 2,600만 어절로 이루어져 있고, 균형 말뭉치(balanced corpus)로서 통계 기반 방식이 학습 말뭉치의 영향을 크게 받는 점을 줄이기 위함이다.

부분적인 띄어쓰기 오류의 수정에 대한 평가는 적절한 실험 말뭉치가 없어 용이하지 않으므로, 본 논문에서는 실험 말뭉치로 ETRI 28만 품사부착 말뭉치[20]에서 추출한 원시 어절을 모두 붙여쓴 형태로 가공한 것을 입력으로 하여 띄어쓰기가 수정된 말뭉치를 가공하기 전의 말뭉치와 비교함으로써 전체 음절과 어절에 대한 띄어쓰기 상태에 대해서 평가를 하였다. 표 3에는 실험 말뭉치의 통계가 제시되어 있다.

표 3의 1열과 2열은 말뭉치에서 해당 음절과 다음 음절의 사이를 붙여쓴 경우와 띄어쓴 경우를 각각 계산한 결과이다. 이 표에 따르면 어절 당 평균 음절의 수는 3.21이고, 전체 음절 중에서 붙여쓴 음절의 비율이 약 70%이다. 일반적으로 띄어쓰는 것보다 붙여쓰는 경우가 더 많음을 알 수 있다.

5.2 평가 척도

본 논문에서는 띄어쓰기의 평가 척도로 음절 단위 정확도(P_{syl}), 어절 단위 재현율(R_{word}), 어절 단위 정확률(P_{word})을 사용한다. 음절 단위 정확도는 정답 문서의 음절과 동일하게 띄어쓴 음절의 비율로써 구하고, 어절 단위 재현율은 정답 문서에 있는 어절과 동일하게 띄어쓴 어절이 얼마나 되는지를 측정하며, 어절 단위 정확률은 시스템이 띄어쓴 결과가 정답 문서와 비교하여 얼마나 정확한지를 측정한다. 음절 단위 평가에서 정확률과 재현율을 구분하지 않는 이유는 정답 문서의 음절의 수와 시스템이 출력한 음절의 수는 동일하기 때문이다. 각각의 값은 다음과 같이 계산된다.

표 3 실험 말뭉치 통계

붙여쓴 음절 수	띄어쓴 음절 수 (=총 어절 수)	총 음절 수 (token)	음절의 종류 (type)	붙여쓴 음절의 비율 (%)	띄어쓴 음절의 비율 (%)
638,546	288,291	926,837	2,456	68.90	31.10

2) 모든 문장은 이전 문장과 띄어쓰기 때문이다.

$$P_{syl} = \frac{\text{올바르게 띄어쓴 음절 수}}{\text{전체 음절 수}} \times 100\ (\%)$$

$$R_{word} = \frac{\text{올바르게 띄어쓴 어절 수}}{\text{정답 문서의 어절 수}} \times 100\ (\%)$$

$$P_{word} = \frac{\text{올바르게 띄어쓴 어절 수}}{\text{시스템이 출력한 어절 수}} \times 100\ (\%)$$

5.3 복합 명사를 고려한 평가 방법

한국어의 복합어는 붙여쓰거나 띄어쓰는 것을 모두 허용하는 경우가 있고, 특히 복합 명사에서 많이 발생한다. 지금까지 자동 띄어쓰기의 평가는 띄어써도 되고 붙여써도 되는 경우를 모두 맞는 것으로 간주하거나[12,17], 정답 문서와 다른 경우는 무조건 틀린 것으로 간주하는 방법[14]이 있었다. 전자는 수작업에 의존했기 때문에 적은 양의 실험 말뭉치에 적용할 수밖에 없었고, 후자는 사람이 작성한 문서 내에서도 일관성이 있게 띄어쓰지 않는다는 점과 동일한 복합 명사라도 사람에 따라 다르게 띄어쓸 수 있다는 점을 고려하면 성능을 과소 평가하는 경향이 있다.

본 논문에서는 품사 부착된 말뭉치로부터 복합 명사를 고려하여 띄어쓰기의 평가를 자동으로 하는 방법을 제안한다. 품사 부착 말뭉치로부터 연속으로 나타나는 두 명사가 있을 때, 앞 명사의 마지막 음절에 대한 띄어쓰기 태그를 0과 1 대신, 해당 음절의 띄어쓰기와 붙여쓰기를 모두 허용하는 태그인 2를 사용한다. 예를 들어, "고속의 대용량 정보전송이 가능하고,"의 품사 부착된 형태는 "고속/nc+의/jm 대용량/nc 정보/nc+전송/nc+이/jc 가능/nc+하/xsm+고/ec+,/s"이고 [3], 여기서 명사 "대용량", "정보", "전송"이 연속으로 나타난다. 이 문장을 복합 명사를 고려하여 띄어쓰기 태그 부착된 형태로 변환하면 "고/0+속/0+의/1+대/0+용/0+량/2-정/0+보/2+전/0+송/0+이/1+가/0+능/0+하/0+고/0+,/1"이 된다. 만약 예문의 두 번째와 세 번째 어절에 대한 띄어쓰기 결과가 "대용량 정보 전송이", "대용량 정보전송이", "대용량정보 전송이", "대용량정보전송이" 중 어느 하나에 해당되더라도 올바르다고 간주한다.

5.4 복합 명사를 고려하지 않는 실험

이 절은 복합 명사를 고려하지 않는 즉, 문서와 동일한 경우만 올바르다고 인정하는 평가 방법을 따르고 있다. 본 논문에서 제시한 두 가지 띄어쓰기 모델인 태그 우선 모델과 음절 우선 모델은 각각 $\Lambda(S_{(K:J)}, T_{(L:I)})$와 $\Phi(T_{(K:J)}, S_{(L:I)})$로 표기하고, K, J, L, I 의 값에 따라 여러 가지 모델이 존재한다. 태그 우선 모델에 대한

표 4 $\Lambda(S_{(K:J)}, T_{(L:I)})$에 대한 실험 결과

(K,J,L,I)	P_{syl}	R_{word}	P_{word}	(K,J,L,I)	P_{syl}	R_{word}	P_{word}	(K,J,L,I)	P_{syl}	R_{word}	P_{word}
(0,1,0,0)	84.26	41.28	44.06	(0,1,0,1)	88.93	55.38	57.10	(0,1,0,2)	88.45	53.83	55.88
(0,1,1,0)	89.44	56.91	61.34	(0,1,1,1)	95.58	79.31	82.58	(0,1,1,2)	95.74	79.76	83.68
(0,1,2,0)	84.44	42.15	47.02	(0,1,2,1)	92.86	70.26	71.63	(0,1,2,2)	94.97	76.90	79.45
(0,2,0,0)	85.48	45.65	47.52	(0,2,0,1)	88.93	56.24	57.21	(0,2,0,2)	89.59	58.23	59.88
(0,2,1,0)	90.22	59.12	63.74	(0,2,1,1)	95.60	79.26	82.94	(0,2,1,2)	95.92	80.41	84.56
(0,2,2,0)	86.46	47.62	52.15	(0,2,2,1)	93.44	72.06	73.90	(0,2,2,2)	95.22	77.84	80.59
(1,0,0,0)	85.75	47.05	48.96	(1,0,0,1)	90.24	60.73	62.20	(1,0,0,2)	89.74	58.68	61.09
(1,0,1,0)	89.28	59.80	59.98	(1,0,1,1)	95.64	81.17	81.81	(1,0,1,2)	95.90	81.50	83.56
(1,0,2,0)	82.85	45.10	45.38	(1,0,2,1)	93.30	73.04	73.39	(1,0,2,2)	94.94	77.52	78.88
(1,1,0,0)	85.83	49.95	50.43	(1,1,0,1)	90.96	63.18	64.89	(1,1,0,2)	90.21	62.99	62.58
(1,1,1,0)	89.85	61.47	62.80	(1,1,1,1)	96.15	82.88	84.10	(1,1,1,2)	96.17	82.67	84.86
(1,1,2,0)	84.21	49.44	49.29	(1,1,2,1)	94.07	75.54	76.87	(1,1,2,2)	95.62	80.32	82.13
(1,2,0,0)	87.21	54.25	54.85	(1,2,0,1)	90.83	63.34	64.59	(1,2,0,2)	91.54	66.39	67.00
(1,2,1,0)	90.74	64.14	65.63	(1,2,1,1)	96.07	82.44	84.09	(1,2,1,2)	96.39	83.51	85.91
(1,2,2,0)	86.96	55.50	55.95	(1,2,2,1)	94.67	77.53	79.28	(1,2,2,2)	95.90	81.39	83.42
(2,0,0,0)	86.18	50.25	51.42	(2,0,0,1)	90.44	61.97	63.61	(2,0,0,2)	89.77	61.52	62.17
(2,0,1,0)	89.49	61.07	61.32	(2,0,1,1)	95.83	82.11	82.73	(2,0,1,2)	95.91	82.09	83.39
(2,0,2,0)	83.37	46.52	47.15	(2,0,2,1)	93.55	73.91	74.63	(2,0,2,2)	95.03	78.36	78.96
(2,1,0,0)	86.51	52.60	53.46	(2,1,0,1)	91.10	64.81	65.85	(2,1,0,2)	90.69	65.11	65.10
(2,1,1,0)	90.34	64.04	64.90	(2,1,1,1)	96.29	83.73	84.74	(2,1,1,2)	96.28	83.43	85.21
(2,1,2,0)	85.07	52.32	52.63	(2,1,2,1)	94.31	76.69	77.82	(2,1,2,2)	95.91	81.51	83.45
(2,2,0,0)	88.58	58.94	59.84	(2,2,0,1)	91.78	67.07	68.32	(2,2,0,2)	92.44	69.88	70.54
(2,2,1,0)	91.65	67.82	69.14	(2,2,1,1)	96.26	83.46	84.88	(2,2,1,2)	96.69	84.93	86.82
(2,2,2,0)	88.97	61.20	62.28	(2,2,2,1)	95.01	78.99	80.60	(2,2,2,2)	96.04	82.05	83.96

3) 여기서 사용된 품사 집합은 ETRI 말뭉치의 품사집합이다.

표 5 $\Phi(T_{(K,J)}, S_{(L,I)})$에 대한 실험 결과

(K,J,L,I)	P_{syl}	R_{word}	P_{word}	(K,J,L,I)	P_{syl}	R_{word}	P_{word}	(K,J,L,I)	P_{syl}	R_{word}	P_{word}
(0,1,0,0)	89.09	55.03	58.79	(0,1,0,1)	89.28	59.81	59.99	(0,1,0,2)	89.34	60.54	60.96
(0,1,1,0)	92.64	67.57	70.70	(0,1,1,1)	92.88	70.44	72.62	(0,1,1,2)	93.17	71.68	73.88
(0,1,2,0)	92.34	68.97	68.69	(0,1,2,1)	94.22	76.18	76.18	(0,1,2,2)	95.48	80.61	80.70
(0,2,0,0)	90.46	62.41	63.85	(0,2,0,1)	90.82	66.22	65.05	(0,2,0,2)	91.02	67.25	66.06
(0,2,1,0)	93.67	72.90	74.42	(0,2,1,1)	93.77	74.87	75.08	(0,2,1,2)	94.08	75.98	76.32
(0,2,2,0)	93.57	73.79	73.50	(0,2,2,1)	94.94	79.26	78.59	(0,2,2,2)	95.84	82.47	81.76
(1,0,0,0)	83.86	38.24	42.27	(1,0,0,1)	83.54	42.06	44.05	(1,0,0,2)	83.90	43.20	46.48
(1,0,1,0)	88.94	55.38	57.14	(1,0,1,1)	88.95	57.49	60.01	(1,0,1,2)	89.91	60.43	63.17
(1,0,2,0)	89.92	58.16	60.96	(1,0,2,1)	92.53	69.02	71.04	(1,0,2,2)	94.51	76.11	77.70
(1,1,0,0)	94.71	75.48	79.06	(1,1,0,1)	95.17	78.98	80.08	(1,1,0,2)	95.22	79.45	80.38
(1,1,1,0)	95.94	81.31	83.33	(1,1,1,1)	96.14	82.84	84.04	(1,1,1,2)	96.21	83.23	84.57
(1,1,2,0)	95.64	81.16	81.47	(1,1,2,1)	96.35	84.08	84.56	(1,1,2,2)	97.00	86.57	87.24
(1,2,0,0)	96.17	83.06	84.02	(1,2,0,1)	96.57	85.21	85.20	(1,2,0,2)	96.42	85.16	84.54
(1,2,1,0)	96.92	86.19	86.82	(1,2,1,1)	97.12	87.35	87.56	(1,2,1,2)	97.16	87.56	87.85
(1,2,2,0)	96.61	85.39	85.29	(1,2,2,1)	97.08	87.22	87.33	(1,2,2,2)	97.34	88.28	88.41
(2,0,0,0)	83.86	38.24	42.27	(2,0,0,1)	83.54	42.06	44.05	(2,0,0,2)	83.90	43.20	46.48
(2,0,1,0)	88.94	55.38	57.14	(2,0,1,1)	88.95	57.49	60.01	(2,0,1,2)	89.91	60.43	63.17
(2,0,2,0)	89.92	58.16	60.96	(2,0,2,1)	92.53	69.02	71.04	(2,0,2,2)	94.51	76.11	77.69
(2,1,0,0)	95.42	77.57	82.46	(2,1,0,1)	95.54	80.06	82.53	(2,1,0,2)	95.69	80.54	83.06
(2,1,1,0)	96.46	82.28	86.52	(2,1,1,1)	96.73	84.55	87.15	(2,1,1,2)	96.81	84.93	87.51
(2,1,2,0)	96.16	81.85	84.97	(2,1,2,1)	96.44	83.19	86.14	(2,1,2,2)	97.10	86.35	88.40
(2,2,0,0)	96.41	84.70	86.04	(2,2,0,1)	96.62	85.92	86.54	(2,2,0,2)	96.81	86.02	87.06
(2,2,1,0)	97.42	87.35	89.64	(2,2,1,1)	97.47	87.85	89.76	(2,2,1,2)	97.48	87.89	89.79
(2,2,2,0)	97.20	86.79	88.57	(2,2,2,1)	97.29	87.26	88.95	(2,2,2,2)	97.39	87.69	89.31

실험 결과는 표 4에, 음절 우선 모델에 대한 실험 결과는 표 5에 있다.

실험 결과에 따르면 문맥을 확장할수록 높은 성능을 보인다. 표 4에서 품사 부착 문제에서 주로 사용되는 bigram HMM 모델은 $\Lambda(T_{(1,0)}, S_{(0,0)})$이고, trigram HMM 모델은 $\Lambda(T_{(2,0)}, S_{(0,0)})$으로서 낮은 성능을 보이고 있다. 이를 통해 띄어쓰기 문제에 이들 모델을 그대로 적용하는 것보다는 주어진 문제에 따라 적합한 모델을 선택하는 것이 바람직함을 알 수 있다.

태그 우선 모델에서는 $\Lambda(T_{(2,2)}, S_{(1,2)})$가 가장 좋은 결과를 나타내고, 음절 우선 모델에서는 $\Phi(S_{(1,2)}, T_{(2,2)})$가 가장 높은 어절 단위 재현율을, $\Phi(S_{(2,2)}, T_{(1,2)})$가 음절 단위 정확률과 어절 단위 정확률을 나타냈다. 일부 모델들이 가장 확장된 문맥을 사용하는 모델인 $\Lambda(T_{(2,2)}, S_{(2,2)})$이나 $\Phi(S_{(2,2)}, T_{(2,2)})$보다 조금씩 더 좋은 결과를 보인 이유는 자료 부족 문제가 발생하기 때문인 것으로 보인다. 태그 우선 모델과 음절 우선 모델 중에서는 음절 우선 모델이 더 좋은 결과를 나타내고 있다.

위의 실험과 동일한 학습 말뭉치와 실험 말뭉치에 대해서 강승식(2001)의 방법을 재구현하여 실험한 결과, 음절 단위 정확률은 93.06, 어절 단위 재현율과 정확률은 각각 76.71과 67.80을 나타냄으로써 본 논문에서 제안한 모델의 성능이 더 높음을 알 수 있다.

모델의 매개변수의 수는 K, J, L, I 의 값이 증가함에 따라 기하급수적으로 증가한다. 태그 우선 모델에서는 J와 I의 값이 음절 문맥을, K와 L의 값이 띄어쓰기 태그 문맥을 결정하고, 음절 우선 모델에서는 이와는 반대로 영향을 미친다. 그런데 모델의 매개변수의 수는 하나의 태그 문맥을 확장함에 따라 2배씩 증가하지만, 음절 문맥을 확장하는 경우에는 이론적으로 한글의 모든 가능한 음절의 수인 11,172배씩 모델의 매개변수가 증가한다[4]. 따라서 음절 문맥을 확장하는 것은 모델의 복잡도를 매우 증가시키게 된다. 태그 우선 모델에서는 I가 2인 경우에, 음절 우선 모델에서는 K가 2인 경우에 음절 trigram을 사용한다. 음절 bigram 사용하는 모델인 $\Lambda(T_{(K,J)}, S_{(L,1)})$과 $\Phi(S_{(1,J)}, T_{(L,I)})$의 경우에도 같은 음절 bigram을 사용하는 강승식(2001)보다 좋은 성능을 보이고 있다. 이를 통해 모델의 매개변수의 수가 유사하더라도 제안된 모델이 기존의 통계적 접근 방법보다 더욱 효과적이라는 것을 알 수 있다.

4) 초성 19자, 중성 21자, 종성 28자(무종성 포함)의 조합

표 6 $\Lambda(S_{(K:J)}, T_{(L:I)})$에 대한 복합 명사를 고려한 실험 결과

(K,J,L,I)	P_{syl}	P_{word}	(K,J,L,I)	P_{syl}	P_{word}	(K,J,L,I)	P_{syl}	P_{word}
(0,1,0,0)	85.96	46.56	(0,1,0,1)	90.55	60.03	(0,1,0,2)	89.97	58.61
(0,1,1,0)	90.99	64.29	(0,1,1,1)	96.65	85.82	(0,1,1,2)	96.79	86.90
(0,1,2,0)	85.73	49.41	(0,1,2,1)	93.78	74.86	(0,1,2,2)	95.88	82.62
(0,2,0,0)	87.15	50.08	(0,2,0,1)	90.49	59.95	(0,2,0,2)	91.08	62.54
(0,2,1,0)	91.76	66.77	(0,2,1,1)	96.69	86.11	(0,2,1,2)	96.97	87.60
(0,2,2,0)	87.74	54.28	(0,2,2,1)	94.37	76.94	(0,2,2,2)	96.13	83.59
(1,0,0,0)	87.41	51.61	(1,0,0,1)	91.65	65.06	(1,0,0,2)	91.03	64.12
(1,0,1,0)	90.75	62.60	(1,0,1,1)	96.64	85.33	(1,0,1,2)	96.89	87.05
(1,0,2,0)	84.08	47.98	(1,0,2,1)	94.19	77.14	(1,0,2,2)	95.86	82.50
(1,1,0,0)	87.45	53.02	(1,1,0,1)	92.41	67.92	(1,1,0,2)	91.52	65.33
(1,1,1,0)	91.32	65.48	(1,1,1,1)	97.13	87.43	(1,1,1,2)	97.15	88.18
(1,1,2,0)	85.42	51.72	(1,1,2,1)	94.97	80.57	(1,1,2,2)	96.50	85.54
(1,2,0,0)	88.75	57.38	(1,2,0,1)	92.24	67.40	(1,2,0,2)	92.80	69.64
(1,2,1,0)	92.15	68.27	(1,2,1,1)	97.07	87.25	(1,2,1,2)	97.35	88.91
(1,2,2,0)	88.14	58.27	(1,2,2,1)	95.57	82.71	(1,2,2,2)	96.76	86.57
(2,0,0,0)	87.73	53.80	(2,0,0,1)	91.75	66.39	(2,0,0,2)	90.98	64.94
(2,0,1,0)	90.86	63.71	(2,0,1,1)	96.79	86.14	(2,0,1,2)	96.84	86.73
(2,0,2,0)	84.60	49.54	(2,0,2,1)	94.44	78.36	(2,0,2,2)	95.89	82.46
(2,1,0,0)	88.04	55.83	(2,1,0,1)	92.41	68.66	(2,1,0,2)	91.90	67.74
(2,1,1,0)	91.71	67.41	(2,1,1,1)	97.25	88.01	(2,1,1,2)	97.23	88.41
(2,1,2,0)	86.28	54.77	(2,1,2,1)	95.17	81.30	(2,1,2,2)	96.76	86.76
(2,2,0,0)	90.00	62.32	(2,2,0,1)	93.07	71.08	(2,2,0,2)	93.58	73.15
(2,2,1,0)	92.96	71.74	(2,2,1,1)	97.22	87.95	(2,2,1,2)	97.59	89.78
(2,2,2,0)	90.09	64.76	(2,2,2,1)	95.87	83.92	(2,2,2,2)	96.88	87.08

표 7 $\Phi(T_{(K:J)}, S_{(L:I)})$에 대한 복합 명사를 고려한 실험 결과

(K,J,L,I)	P_{syl}	P_{word}	(K,J,L,I)	P_{syl}	P_{word}	(K,J,L,I)	P_{syl}	P_{word}
(0,1,0,0)	90.64	61.67	(0,1,0,1)	90.76	62.61	(0,1,0,2)	90.79	63.53
(0,1,1,0)	94.04	73.86	(0,1,1,1)	94.19	75.62	(0,1,1,2)	94.44	76.87
(0,1,2,0)	93.58	71.34	(0,1,2,1)	95.31	79.01	(0,1,2,2)	96.48	83.73
(0,2,0,0)	91.81	66.85	(0,2,0,1)	92.09	68.03	(0,2,0,2)	92.28	69.02
(0,2,1,0)	94.87	77.97	(0,2,1,1)	94.91	78.54	(0,2,1,2)	95.20	79.79
(0,2,2,0)	94.69	76.74	(0,2,2,1)	95.95	81.94	(0,2,2,2)	96.78	85.20
(1,0,0,0)	85.58	44.72	(1,0,0,1)	85.21	46.36	(1,0,0,2)	85.56	48.87
(1,0,1,0)	90.57	60.08	(1,0,1,1)	90.45	62.79	(1,0,1,2)	91.36	65.98
(1,0,2,0)	91.42	63.83	(1,0,2,1)	93.79	73.94	(1,0,2,2)	95.62	80.73
(1,1,0,0)	95.79	82.20	(1,1,0,1)	96.19	83.39	(1,1,0,2)	96.22	83.68
(1,1,1,0)	96.96	86.71	(1,1,1,1)	97.12	87.37	(1,1,1,2)	97.19	87.91
(1,1,2,0)	96.63	84.69	(1,1,2,1)	97.28	87.76	(1,1,2,2)	97.88	90.52
(1,2,0,0)	97.12	87.55	(1,2,0,1)	97.48	88.99	(1,2,0,2)	97.31	88.25
(1,2,1,0)	97.83	90.49	(1,2,1,1)	98.00	91.27	(1,2,1,2)	98.02	91.49
(1,2,2,0)	97.51	88.85	(1,2,2,1)	97.94	90.86	(1,2,2,2)	98.18	91.95
(2,0,0,0)	85.58	44.72	(2,0,0,1)	85.21	46.36	(2,0,0,2)	85.56	48.87
(2,0,1,0)	90.57	60.08	(2,0,1,1)	90.45	62.79	(2,0,1,2)	91.36	65.98
(2,0,2,0)	91.42	63.83	(2,0,2,1)	93.79	73.94	(2,0,2,2)	95.62	80.73
(2,1,0,0)	96.52	85.69	(2,1,0,1)	96.58	85.82	(2,1,0,2)	96.71	86.32
(2,1,1,0)	97.47	89.64	(2,1,1,1)	97.67	90.19	(2,1,1,2)	97.74	90.55
(2,1,2,0)	97.17	88.12	(2,1,2,1)	97.42	89.32	(2,1,2,2)	98.00	91.52
(2,2,0,0)	97.32	89.18	(2,2,0,1)	97.50	89.84	(2,2,0,2)	97.69	90.41
(2,2,1,0)	98.29	92.88	(2,2,1,1)	98.33	93.05	(2,2,1,2)	98.33	93.06
(2,2,2,0)	98.08	91.83	(2,2,2,1)	98.16	92.20	(2,2,2,2)	98.25	92.57

5.5 복합 명사를 고려한 실험

복합 명사를 고려한 실험 결과가 표 6과 표 7에 있다. 정답 문서의 어절의 수는 응답에 따라 가변적이기 때문에 어절 단위 재현율에 대한 평가는 하지 않았다.

문맥에 따른 각 모델간의 성능 차이는 5.4절의 실험과 거의 유사하다. 복합 명사에 대한 붙여쓰기와 띄어쓰기를 모두 허용하는 것이 그렇지 않은 경우보다는 음절 단위 정확도는 약 0.9~2.0%정도, 어절 단위 정확률은 약 3.5~5.8%정도 더 좋은 결과를 보였다. 강승식(2001)의 방법에 대한 음절 단위 정확률은, 94.01어절 단위 정확률은 71.22였다.

5.6 학습 말뭉치의 크기에 따른 실험

각 모델이 최적의 성능을 보이기 위해서는 어느 정도 크기의 학습 말뭉치가 필요한지, 어떤 모델이 주어진 크기의 학습 말뭉치에 가장 적합한 지는 아직 알지 못한다. 이러한 것을 알아보기 위해 그림 1, 그림 2, 그림 3에서는 학습 말뭉치의 크기에 따른 여러 모델의 성능을 비교하고 있다.[5] 그림에서 "TF"는 태그 우선 모델, "SF"는 음절 우선 모델, "KSS"은 강승식(2001)의 방법을 사용한 것이고 숫자는 모델의 종류를 의미한다. 음절 unigram을 사용하는 모델은 "TF 2110"과 "SF 0202"이고, 음절 bigram을 사용하는 모델은 "TF 2111", "SF 1212", "KSS"이고, 음절 trigram을 사용하는 모델은 "TF-2212"와 "SF-2212"이다.[6]

학습 말뭉치의 크기에 따른 성능의 변화를 관찰할 수 있는데, 음절 unigram을 사용하는 "TF 2110"과 "SF-0202"는 적은 양의 말뭉치에서 수렴하고 있고, 음절 bigram을 사용하는 "TF-2111"과 "SF 1212", "KSS"는 더 많은 말뭉치에서 수렴하고 있다. 음절 trigram을 사용하는 "TF-2212"와 "SF-2212"는 수렴하고 있지 않는 것으로 보아 더 큰 학습 말뭉치가 주어지면 성능이 더 높아질 가능성이 있다.

동일한 음절 n-gram을 사용하는 태그 우선 모델과 음절 우선 모델은 말뭉치의 크기와는 무관하게 음절 우선 모델이 일관되게 더 높은 성능을 보이고 있다. 이를 통해 띄어쓰기 문제에 대해서는 음절 우선 모델에서 사용된 마르코프 가정이 태그 우선 모델에서 사용된 마르코프 가정보다 더 견고하다는 것을 알 수 있다.

강승식(2001)의 방법의 어절 단위 정확률이 상대적으로 매우 낮은 이유는 주변 띄어쓰기 상태에 대한 문맥을 전혀 고려하지 않고 주변의 음절 문맥만을 고려하기 때문인 것으로 보인다. 반대로 제안한 두 모델은 동적 알고리즘의 일종인 Viterbi 알고리즘을 사용하여 최적의 상태열(띄어쓰기 태그열)을 찾기 때문에 어절 단위 평가에서 이러한 문제점을 해결하고 있다.

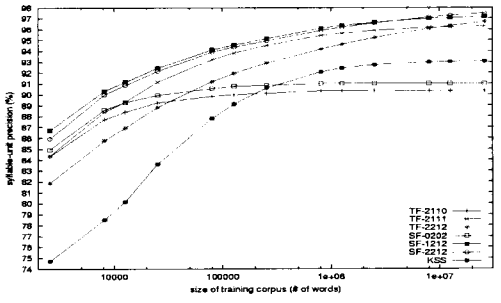


그림 1 학습 말뭉치의 크기에 따른 음절 단위 정확도 비교

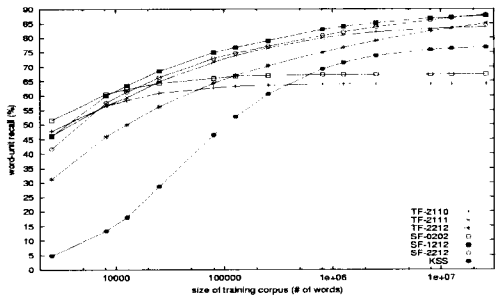


그림 2 학습 말뭉치의 크기에 따른 어절 단위 재현율 비교

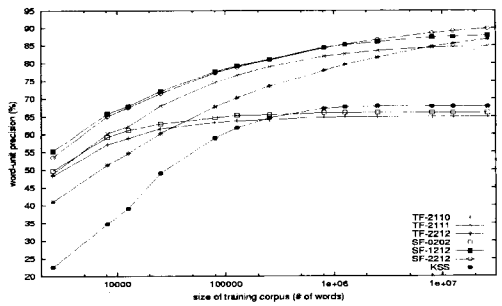


그림 3 학습 말뭉치의 크기에 따른 어절 단위 정확률 비교

5) 복합 명사를 고려하지 않은 실험이다.

6) 여기에 사용된 모델은 각 음절 n-gram을 사용하는 모델 중에서 가장 좋은 성능을 보이는 것을 선택한 것이다.

5.7 오류 분석

띄어쓰기의 오류에는 붙여써야 할 어절을 띄어쓰는 오류인 붙띄오류와 띄어써야 할 어절을 붙여쓰는 오류인 띄붙오류가 있다. 오류의 분석을 위해서 이 절에서 사용하는 기호와 용어는 다음과 같이 정의한다.

ⓐ f_0 = 띄붙오류가 발생한 음절의 수(응답은 0이고 정답은 1인 경우)
ⓑ f_1 = 붙띄오류가 발생한 음절의 수(응답은 1이고 정답은 0인 경우)
ⓒ t_0 = 붙여쓰기가 성공한 음절의 수(응답과 정답이 모두 0인 경우)
ⓓ t_1 = 띄어쓰기가 성공한 음절의 수(응답과 정답이 모두 1인 경우)
ⓔ $f_0+f_1+t_0+t_1$ = 전체 음절의 수
ⓕ f_0+f_1 = 전체 오류의 수
ⓖ f_0+t_0 = 붙여쓰기를 시도한 음절의 수 (응답)
ⓗ f_1+t_1 = 띄어쓰기를 시도한 음절의 수 (응답)
ⓘ f_0+t_1 = 띄어쓴 음절의 수 (정답)
ⓙ f_1+t_0 = 붙여쓴 음절의 수 (정답)
ⓚ ⓖ÷ⓔ×100 = 붙여쓰기 시도율
ⓛ ⓗ÷ⓔ×100 = 띄어쓰기 시도율
ⓜ ⓒ÷ⓖ×100 = 붙여쓰기 성공율
ⓝ ⓓ÷ⓗ×100 = 띄어쓰기 성공율
ⓞ ⓕ÷ⓔ×100 = 오류율

5.6 절에서 비교하는 있는 7가지 모델에 대해 실험 말뭉치로부터 구한 f_0, f_1, t_0, t_1 의 값은 표 8에 있다.

표 8 모델에 따른 음절 결과

	TF-2110	TF-2111	TF-2212	SF-0202	SF-1212	SF-2212	KSS
f_0	46,701	18,908	18,486	39,014	13,638	14,751	13,238
f_1	42,857	15,448	12,193	44,198	12,664	8,648	51,083
t_0	595,689	623,098	626,353	594,348	625,882	629,898	587,463
t_1	241,590	269,383	269,805	249,277	274,653	273,540	275,053

이 표로부터 그림 4와 같은 그래프를 얻을 수 있다.

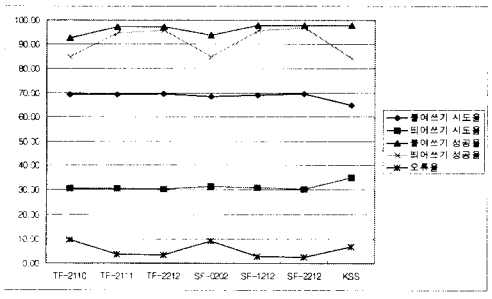


그림 4 모델에 따른 오류 비교

제안하는 모델은 문맥의 크기와 상관없이 거의 일정하게 약 70:30의 붙여쓰기/띄어쓰기 시도율을 보여준다. 이 것은 표 3에서 살펴본 실험 말뭉치에 대한 붙여쓰기와 띄어쓰기 간의 이상적인 비율과 유사한 것이다. 반면에, 강승식(2001)의 방법은 붙여쓰기/띄어쓰기 시도율이 이상적인 비율과 차이가 나며, 띄어쓰기 시도율이 높음에도 불구하고 띄어쓰기 성공률은 상대적으로 낮다.

그림 1, 그림 2, 그림 3에서 살펴본 바와 같이, 음절 unigram처럼 매우 제한된 문맥을 사용하는 모델이 일정한 양 이상의 학습 데이타에서 성능이 더 이상 향상되지 않는 것으로 보아 모델 자체의 한계로 인한 문제로 볼 수 있다. 그러나 문맥을 확장함에 따라 일정한 붙여쓰기/띄어쓰기 시도율을 유지하면서 띄어쓰기 성공율은 점차 향상되고 오류율은 감소하는 것을 관찰할 수 있다.

본 논문에서 가장 좋은 성능을 보이는 $\Phi(S_{(2:2)}, T_{(1:2)})$모델에 대해 복합 명사를 고려한 평가를 바탕으로 특정 음절이 자주 오류를 유발하는 지의 여부를 알아보고자 하였다. 표 9에서는 빈도가 10이상인 음절을 대상으로 오류율을 조사하였다.

표 9 음절 종류와 오류율의 관계

오류율	음절 종류	백분율 (%)	누적백분율 (%)
5% 이내	1,064	87.43	87.43
10% 이내	86	7.07	94.49
15% 이내	21	1.73	96.22
20% 이내	22	1.81	98.03
25% 이내	10	0.82	98.85
30% 이내	7	0.58	99.42
35% 이내	3	0.25	99.67
40% 이내	2	0.16	99.84
45% 이내	1	0.08	99.92
50% 이내	1	0.08	100.00
100% 이내	0	0	100

오류율이 가장 높은 상위 10개 음절을 조사한 결과 다음과 같은 표를 얻었다.

표 10에서 한글 음절 "롱"에 대한 오류 중에서 31개가 "고롱", "넥"에 대한 4개의 오류가 모두 "바넥"이라는 인명으로 사용되었다. 이러한 경우는 자료부족 문제로부터 기인한 오류이다. 또한, "꿔"의 오류는 모두 "바꿔주다", "바꿔버리다"와 같이 붙여쓰거나 띄어쓰는 것이 모두 허용되는 경우에 해당하므로 엄밀하게는 오류에 속하지 않는다고 볼 수 있다.

표 10 상위 10개 음절의 음절 통계 (오류율순)

음절의 종류	오류율	f_0	f_1	t_0	t_1	빈도
롱	46.58	34	0	32	7	73
~	41.38	6	6	10	7	29
네	40.00	0	4	6	0	10
-	36.84	12	2	10	14	38
꿔	35.00	4	3	10	3	20
5	34.41	207	6	393	13	619
P	31.62	37	0	78	2	117
릉	30.00	3	0	6	1	10
0	29.73	298	13	718	17	1,046
R	28.21	10	1	28	0	39

표 11 상위 10개 음절의 음절 통계 (빈도순)

음절의 종류	오류율	f_0	f_1	t_0	t_1	빈도
.	0.62	123	90	10,032	23,964	34,209
이	2.18	383	277	16,675	12,894	30,229
다	0.64	94	86	27,031	1,068	28,279
는	0.68	103	53	1,798	20,996	22,950
에	0.85	89	50	6,588	9,673	16,400
을	0.69	47	62	402	15,366	15,877
의	0.75	78	36	2,084	13,019	15,217
가	1.30	101	91	5,963	8,583	14,738
고	1.35	86	113	3,611	10,890	14,700
"	1.37	34	161	7,225	6,767	14,187

이 밖에는 기호, 숫자, 알파벳에 대한 오류율이 높게 나타났다. 특히 한자와 같은 저빈도 음절의 오류율이 높은데, 모든 음절에 대해 조사한 결과, 오류율이 100%인 다음과 같은 17개의 음절이 발견되었고, 이 중에서 13개 음절의 빈도가 1이었다.

ㅣ, ·-, 센, 닻, 끈, 쏴, 줘, 횅, 鶩, 達, 私, 想, 騷, 列, 潤, 點, 血

표 11에서는 빈도가 가장 높은 상위 10개 음절에 대한 빈도를 조사하였다. 고빈도 음절의 경우는 평균 오류율인 2.52보다 모두 낮은 오류율을 나타내고 있다.

이상의 결과를 종합하면, 고빈도 음절은 대체로 비교적 낮은 오류율을 보이는 반면에, 기호, 숫자, 알파벳, 저빈도의 한자나 한글 등은 상대적으로 높은 오류율을 보인다. 미등록어 특히 외래어의 일부로 사용된 한글 음절의 경우에도 오류율이 높다.

본 논문에서는 음절의 통계적 정보만을 이용하므로, 해당 음절이 어떤 형태소에 속하는 지는 알 수 없으나, 일부의 띄어쓰기 오류에 대한 유형을 표 12와 같이 수작업으로 분류하였다. 여기서 주의할 점은 제안하는 모델은 각각의 주어진 음절에 대해 띄어쓰기 태그를 독립적으로 결정하는 것이 아니라, 주어진 문장에 대해 모델의 확률값에 따라 최적의 띄어쓰기 태그열을 결정하기

표 12 띄어쓰기 오류의 유형과 예

오류 유형	정 답	응 답
본 용언과 보조 용언의 사이	생각해봐	*생각해 봐
	시켜주라는	*시켜 주라는
	도와 주는	*도와주는
명사와 용언의 사이	얘기 하세요	*얘기하세요
	상처준다	*상처 준다
관형어와 명사의 사이	먼 거리	먼거리
	세 쌍둥이	*세쌍둥이
	여러가지	*여러 가지
	저 녀석	저녀석
	우리 집	*우리집
의존명사의 앞	십년 만에	십년만에
	내친 김에	내친김에
	중국 쪽의	중국쪽의
부사와 용언의 사이	안하다	*안 하다
	못된다는	*못 된다는
	못 하나	*못하나
어미와 조사의 사이	아는 것도 많구나라는	아는 것도 많구나 라는
의미적 중의성	일어나 가버리고	일어나가 버리고
	얘기 가질 무렵	얘기가 질 무렵

때문에 동일한 음절열에 대해서도 문맥에 따라 다른 결과를 초래할 수 있다. 예를 들어, 실험 말뭉치에서 음절열 "우리집"에 대한 띄어쓰기 결과는 "우리 집"으로는 26번, "우리집"으로는 6번 분석되었다.

위의 예들은 대부분 사람도 혼히 범하기 쉬운 띄어쓰기 오류로서 이 중에서는 현재의 띄어쓰기 규정에서 허용하는 경우도 있으며[7], 실험 말뭉치 내에서도 항상 일관되게 띄어쓰기가 되어 있는 것은 아니다. 예를 들어 "도와주는"의 띄어쓰기는 "도와 주는"과 "도와주는"이 각각 3번씩 발견되었다. 이러한 현상을 제대로 반영하기 위해서는 복합 명사에 대해서 한 것과 같은 완화된 평가가 필요하다. 그러나 마지막 예에서 보듯이 의미적 중의성으로 인한 오류는 의미 정보를 사용해야만 해결할 수 있다.

6. 결론 및 향후연구

본 논문에서는 한글 문장의 자동 띄어쓰기를 위한 두 가지 통계적 모델을 제안하였다. 제안한 모델은 통계 기반 방식으로서 규칙 기반 방식에 수반되는 복잡한 분석 과정과 어휘 지식의 구축과 관리에 있어서 많은 비용을 필요로 하지 않으며, 원시 말뭉치에서 학습된 결과만을 이용하여 효과적으로 띄어쓰기 문제를 해결할 수 있다.

제안한 모델은 엄격한 마르코프 가정을 사용하는 HMM과는 달리 완화된 마르코프 가정을 사용함으로써 보다 확장된 문맥을 사용할 수 있다. 확장된 문맥을 고려하는 다양한 모델들에 대한 실험을 통하여 문맥을 확장함에 따라 성능이 높아지는 것을 관찰할 수 있었고, 기존의 방법보다 높은 성능을 나타내고 있다. 제안된 두 가지 모델인 음절 우선 모델과 태그 우선 모델 중에서 음절 우선 모델이 태그 우선 모델보다 실험 결과에서 일관되게 더 좋은 결과를 보이는 사실로부터 음절 우선 모델에 사용된 마르코프 가정이 직관적으로 더 타당할 뿐만 아니라 모델 자체도 견고하다는 것을 알 수 있다.

모델의 매개변수의 수는 음절 문맥의 크기에 의해 많은 영향을 받는다. 가장 높은 성능을 보이는 모델은 음절 trigram을 사용하고 있으나, 기존의 방법에서와 같이 음절 bigram을 사용하는 모델들도 기존의 방법보다 높은 성능을 보이고 있다. 이를 통해 제안하는 모델이 기존의 방법보다 효과적임을 알 수 있었다.

본 논문에서는 현재의 띄어쓰기 규정에 따라 띄어쓰기와 붙여쓰기가 모두 허용되는 경우를 반영할 수 있는 효과적인 자동 평가 방법으로서 복합 명사를 고려한 평가 방안을 제안하고, 이에 따른 실험 결과를 제시하였다. 복합 명사를 고려한 평가에서는 그렇지 않을 때보다 약 0.9~2.0%의 음절 단위 정확도가, 약 3.5~5.8%의 어절 단위 정확률이 향상되었다.

향후 연구로는 본 논문에서는 확률 추정 시 단순히 최우추정(MLE)을 이용하고 있으나, 문맥을 확장함에 따라 확률 추정 시 발생하는 자료 부족 문제를 완화하기 위해 효과적인 평탄화 방법을 적용하여 성능을 높이는 방안에 대한 연구가 필요하다. 또한 기호, 숫자, 알파벳, 한자 등과 같이 오류율이 높은 음절에 대해서는 좀 더 면밀한 관찰과 함께 후처리를 통해 오류를 줄이는 방안도 수행할 예정이다.

본 논문에서는 붙여쓰기와 띄어쓰기를 모두 허용하는 경우에 대한 평가는 복합 명사만을 고려하는 자동 평가를 수행하였으나, 오류 유형에 대한 분석에 의하면 실제로는 여러 가지 다양한 경우가 존재하므로 이를 고려하는 평가에 대한 연구가 필요하다.

7) 정답 어절과는 다르지만 응답 어절의 형태도 허용되는 경우에는 '*' 표시를 하였다.

참 고 문 헌

[1] E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz, "Equations for Part-of-Speech Tagging," In Proceedings of the 11th National Conference on Artificial Intelligence(AAAI-93), pp.784-789, 1993.

[2] B. Merialdo, "Tagging English Text with a Probabilistic Model", Computational Linguistics, 20(2), pp.155-172, 1994.

[3] 김진동, 임희석, 임해창, "Twoply HMM : 한국어의 특성을 고려한 형태소 단위의 품사 태깅 모델", 한국정보과학회 논문지(B), 제24권, 12호, pp.1502-1512, 1997.

[4] 이상주, 자동 품사 부착을 위한 새로운 통계적 모형, 고려대학교 컴퓨터학과 박사학위논문, 1999.

[5] K. Seymore, A. McCallum, and R. Rosenfeld, "Learning Hidden Markov Model Structure for Information Extraction," AAAI'99 Workshop on Machine Learning for Information Extraction, 1999.

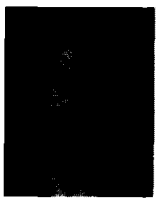
[6] D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. "NYMBLE: A High-Performance Learning Name-finder", In Proceedings of the Fifth Conference on Applied Natural Language Processing, pp. 194-201, 1997.

[7] 오종훈, 최기선, "은닉마르코프 모델(HMM)을 이용한 과학기술문서에서의 외래어 추출 모델", 제 11회 한글 및 한국어 정보처리 학술발표 논문집, pp.137-141, 1999.

[8] 박봉래, 대용량 한글 텍스트 데이타베이스 맞춤법 오류 교정 시스템의 구현, 고려대학교 전산과학과 석사 학위논문, 1995.
[9] 최재혁, "양방향 최장일치법을 이용한 한국어 띄어쓰기 자동 교정 시스템", 제9회 한글 및 한국어 정보처리 학술발표 논문집, pp.145-151, 1997.
[10] 김계성, 이현주, 이상조, "연속 음절 문장에 대한 3단계 한국어 띄어쓰기 시스템", 정보과학회논문지, 제25권 제12호, pp.1838-1844, 1998.
[11] 강승식, "한글 문장의 자동 띄어쓰기", 제10회 한글 및 한국어 정보처리 학술발표 논문집, pp.137-142, 1998.
[12] 강승식, "한글 문장의 자동 띄어쓰기를 위한 어절블록 양방향 알고리즘", 정보과학회논문지, 제27권 제4호, pp.441-447, 2000.
[13] 심광섭, "음절간 상호 정보를 이용한 한국어 자동 띄어쓰기", 정보과학회논문지, 제23권 제9호, pp.991-1000, 1996.
[14] 신중호, 박혁로, "음절 단위 bigram 정보를 이용한 한국어 단어인식모델", 제9회 한글 및 한국어 정보처리 학술발표 논문집, pp.255-260, 1997.
[15] 정영미, 이재윤, "한국어 텍스트 처리를 위한 줄 경계 띄어쓰기 복원", 제6회 한국정보관리학회 학술대회 논문집, pp.21-24, 1999.
[16] 전남열, 박혁로, "음절 Bi gram정보를 이용한 한국어 OCR 후처리용 자동 띄어쓰기", 제 12회 한글 및 한국어 정보처리 학술발표 논문집, pp.95-100, 2000.
[17] 강승식, "음절 bigram를 이용한 띄어쓰기 오류의 자동 교정", 음성과학회논문지, 제8권 2호, pp.83-90, 2001.
[18] 21세기 세종계획 국어기초자료 구축, 문화관광부, 1998.
[19] 21세기 세종계획 국어기초자료 구축, 문화관광부, 1999.
[20] 한국전자통신 연구원, "품사 부착 말뭉치 구축 지침서", 1999, http://aladin.etri.re.kr/~nlu/STANDARD/

이 도 길
1999년 2월 고려대학교 컴퓨터학과 학사. 2001년 2월 고려대학교 컴퓨터학과 석사. 2001년 3월 ~ 현재 고려대학교 컴퓨터학과 박사 과정. 관심분야는 한국어 정보처리, 기계학습, 정보검색, 생물 정보학

이 상 주
1992년 2월 고려대학교 컴퓨터학과 학사. 1995년 2월 고려대학교 컴퓨터학과 석사. 1999년 8월 고려대학교 컴퓨터학과 박사. 1999년 11월 ~ 2001년 3월 일본 동경대학교 정보과학과 연구원 (일본학술진흥회 지원). 1997년 3월 ~ 2002년 2월 고려대학교 기초과학연구소 연구원. 2002년 3월 ~ 현재 (주) 엔엘피솔루션 대표이사. 관심분야는 자연어처리, HCI, 기계학습, 정보검색



임 희 석
1988년 3월 ~ 1992년 2월 고려대학교 컴퓨터학과(이학사). 1992년 3월 ~ 1994년 2월 고려대학교 컴퓨터학과(이학석사). 1994년 3월 ~ 1997년 8월 고려대학교 컴퓨터학과(이학박사). 1997년 9월 ~ 1999년 2월 삼성종합기술원 선임연구원. 1999년 3월 ~ 현재 천안대학교 정보통신학부교수. 관심분야는 인공지능, 자연어처리, 인터넷 정보 시스템, 기계학습, 정보검색

임 해 창
1991년 ~ 현재 고려대학교 컴퓨터학과 교수. 1993년 인지 과학회 이사. 1994년 ~ 1998년 한국 정보과학회 편집위원. 1998년 5월 ~ 2000년 5월 한국정보과학회 한국어정보처리연구회 운영위원장. 1999년 3월 ~ 2000년 8월 고려대학교 컴퓨터과학기술연구소 연구소장. 관심분야는 자연어처리, 구문분석, 정보검색, 기계학습