메모리 제약적 기기를 위한 음절 패턴 기반 띄어쓰기 시스템
(A Word Spacing System based on Syllable Patterns for Memory-constrained Devices)

김 신 일 † 양 선 † 고 영 중 ††
(Shinil Kim) (Seon Yang) (Youngjoong Ko)

요 약 본 논문에서는 메모리 제약적인 기기에 적합한 한국어 띄어쓰기 시스템을 제안한다. 본 연구에서는 최신 선행 연구들에 비해 성능의 저하가 없게 하면서 동시에 메모리 사용량을 탁월하게 줄이는 데에 초점을 맞추었다. 규칙 정보는 전혀 사용하지 않고, 은닉 마르코프 모델(Hidden Markov Model)의 이론에 근거하여 확률 정보를 적용하였으며, 두 가지의 자질을 사용하는데, 1) 첫 번째 자질은 각 음절이 개별적으로 가지는 띄어쓰기 패턴 자질이며, 2) 두 번째 자질은 두 음절 패턴 자질 사이의 전이 확률 값 정보이다. 실험 결과에서, 첫 번째 자질만 사용한 경우 모바일에 적용하기 위해 제안된 다른 연구보다 약 53% 정도 적게 메모리를 사용하면서 약 91% 정도의 정밀도를 보였다. 두 가지 자질을 모두 사용한 경우 음절 바이그램을 사용한 다른 연구와 비교하여 약 76% 정도 메모리를 적게 사용하면서 약 94%가 넘는 우수한 성능을 나타내었다.

키워드 : 띄어쓰기 시스템, 메모리 제약적 기기, 음절 패턴

Abstract In this paper, we propose a word spacing system which can be performed with just a small memory. We focus on significant memory reduction while maintaining the performance of the system as much as the latest studies. Our proposed method is based on the theory of Hidden Markov Model. We use only probability information not adding any rule information. Two types of features are employed: 1) the first features are the spacing patterns dependent on each individual syllable and 2) the second features are the values of transition probability between the two syllable-patterns. In our experiment using only the first type of features, we achieved a high accuracy of more than 91% while reducing the memory by 53% compared with other systems developed for mobile application. When we used both types of features, we achieved an outstanding accuracy of more than 94% while reducing the memory by 76% compared with other system which employs bigram syllables as its features.

Key words : Word Spacing System, Memory-constrained Devices, Pattern of Syllables

· 이 논문은 동아대학교 학술연구비 지원에 의하여 연구되었음

† 학생회원 : 동아대학교 컴퓨터공학과
pirate2003@naver.com
syang@donga.ac.kr
†† 종신회원 : 동아대학교 컴퓨터공학과 교수
yjko@dau.ac.kr

논문접수 : 2010년 1월 6일
심사완료 : 2010년 6월 18일

정보과학회논문지: 소프트웨어 및 응용 제37권 제8호(2010.8)

1. 서 론

한국어에서 하나의 음절은 많은 의미정보를 가진다. '은', '는', '이', '가'와 같은 음절은 앞의 단어가 대부분 주어로 사용된다는 정보와 함께, 뒤에 띄어쓰기가 있어야 한다는 정보 역시 가지고 있다. 때로는 단어의 한 음절에 속하기도 하지만, 이러한 정보들 역시 띄어쓰기를 위해서는 중요한 정보이기도 하다. 본 연구에서는 음절들이 가지는 정보들을 통계 기반 방법에 적용시켜서 적은 메모리만 사용하면서도 우수한 성능을 산출하는 띄어쓰기 시스템을 제안한다.

자연어처리에서 기본이 되는 연구인 띄어쓰기 시스템의 경우, 오래 전부터 많은 연구들이 진행되어왔다. 하

지만 대부분의 연구가 성능에 초점을 맞추어 진행되었기 때문에 메모리 사용량은 당연히 많아질 수밖에 없었다. 모바일 기기의 사용이 증가되고 있는 상황에서 기존의 연구들은 이런 기기에 적용하기에는 어려운 점이 많다. 그래서 최근에는 경량화에 초점을 맞춘 띄어쓰기 시스템도 활발히 연구 역시 진행되고 있고, 본 연구에서도 확률 모델인 Hidden Markov Model(HMM)의 이론적 근거를 바탕으로 실험을 하여 선행 연구들보다 메모리 사용량을 줄이면서 성능 역시 우수한 결과를 확인할 수 있었다.

그림 1은 본 논문에서 음절 정보를 이용하는 제안 시스템의 전체 흐름을 도식화한 것이다.

먼저 띄어쓰기가 올바른 학습 말뭉치에서 음절에 대한 정보를 추출하기 위해 '0'과 '1'로 구성된 말뭉치와 공백을 제거한 말뭉치로 가공시킨다. 이렇게 구성된 말뭉치로부터 두 가지 자질을 추출하는데, 이것은 표 1과 같다.

이 두 자질은 본 시스템에서 각각 관찰확률과 전이확률에 해당되며, 관찰확률의 경우는 총 8개의 자질 패턴('000'~'111')을 추출하여 사용한다. 이렇게 추출된 정보를 바탕으로 사용자 입력 문장에 대해서는 각각의 음절이 가지는 자질 패턴의 곱으로 가능한 모든 시퀀스의 확률 값을 구하게 되고, 그렇게 구해진 시퀀스의 확률 값 중 가장 높은 값을 가지는 시퀀스를 적용시킨다. 입력된 모든 문장에 대해 각각의 시퀀스를 계산하여 적용시킨 뒤 최종적으로 띄어쓰기가 완료된 문장을 출력하게 된다.

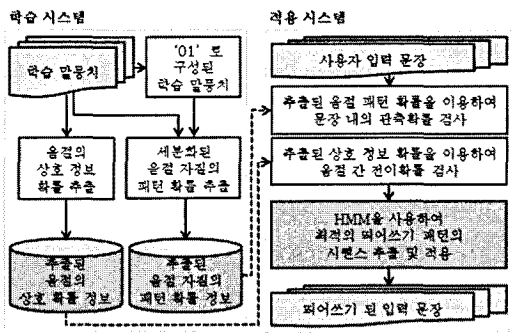

그림 1 전체 시스템 흐름도

표 1 실험에서 사용하는 자질에 대한 설명

자질	설명
자질1	음절 별로 기록된 앞 두 음절과 뒤 한 음절에 대해 띄어쓰기 유무를 표현하는 3비트 형태 패턴의 확률 값 (3.1절 : 음절 자질 패턴)
자질2	음절과 음절 사이에 띄어쓰기/붙여쓰기에 대한 확률 값 (3.2절 : 두 음절 간 확률 정보)

본 연구에서는 자질1만을 사용한 실험과 자질1과 자질2를 모두 사용한 실험을 병행하였다. 자질1만 사용한 실험에서는 마찬가지로 HMM을 응용하여 띄어쓰기에 적용한 '송영길 외(2009)[1]'보다 메모리 사용량이 약 53% 이상 줄고 성능은 약 2.18%이 향상된 것을 확인할 수 있다. [1]은 띄어쓰기 교정 지점을 기준으로 앞 3번째 음절로부터 뒤 2번째 음절까지 총 5음절의 상태를 고려하지만, 제안 시스템은 공백을 제외한 음절을 기준으로 앞 2번째 음절로부터 뒤 1번째 음절까지 총 3 음절의 상태만을 고려한다. 그리고 음절 자질 각각에 대해 확률 값을 저장하여 사용하는 기존 시스템에 비해, 제안 시스템은 음절 자질의 패턴을 확률 값으로 저장하여 사용하기 때문에 메모리 사용량을 획기적으로 줄일 수 있었다.

그리고 자질1과 자질2를 모두 적용해서 띄어쓰기를 실시했을 때는 94% 이상의 결과를 얻었는데, 음절 바이그램을 이용한 '강승식(2001)[2]'의 연구와 비슷한 성능을 내면서도 메모리 사용량이 약 76% 이상 줄었다. 여기에 대해 원인을 분석해본 결과, 기존 연구에서는 두 음절의 왼쪽에 공백이 있을 확률, 중간에 공백이 있을 확률, 오른쪽에 공백이 있을 확률에 대한 정보를 저장한다. 하지만 제안시스템은 두 음절 사이에 대한 확률 값만 저장하여 사용하기 때문에 그만큼 메모리를 절약할 수 있었다.

이처럼 본 연구에서는 기존의 다른 연구들보다 더 적은 용량의 확률 데이터를 사용하면서도 성능 면에서도 손색없는 결과를 산출할 수 있었다. 이는 본 연구에서 제안한 방법이 기존의 시스템들보다 모바일 기기처럼 메모리 제약이 있는 기기에 더 적합할 수 있음을 시사한다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 관련 연구에 대해 기술하며, 3장에서는 음절 정보 추출 방법, 그리고 HMM을 이용하여 문장 띄어쓰기 하는 방법에 대해 기술한다. 4장에서는 실험 결과에 대해 기술하며, 5장에서는 본 연구의 결론 및 향후 연구 계획을 기술한다.

2. 관련 연구

띄어쓰기는 자연어 처리에서 기본이 되는 작업 중 하나로, 이에 대한 연구가 오래 전부터 진행되고 있으며 방법은 크게 규칙 기반 방법[3,4], 통계 기반 방법[2,5,6], 그리고 두 기법을 통합한 방법[1]으로 구분할 수 있다. 규칙 기반 방법은 학습 말뭉치에 크게 영향을 받지 않고 비교적 일관적인 성능을 내지만, 어휘 지식을 구축하고 관리하는데 많은 비용이 든다. 통계 기반 방법은 어휘 지식 관리를 위한 정보량이 많지 않기 때문에 전체

적인 관리가 용이하다는 장점이 있으나, 학습 말뭉치의 종류에 따라 적용 시스템에서 띄어쓰기되는 문장의 성격이 바뀔 수 있다. 최근에는 이 두 방법을 혼합한 방법이 연구되어지고 있다.

먼저 규칙 기반 방법을 사용한 방법의 예로 '강승식(2000)[4]'를 들 수 있다. 이 연구에서는 '공백 인식 접근법'과 '어절 인식 접근법'의 관점에서, 문장을 어절 블록으로 분할하여 어절 경계를 인식하는 알고리즘을 제안하였고 이에 대한 공백 재현율과 어절 재현율은 각각 97.3%, 93.2% 였다. 최근에는 규칙 기반 학습을 하면서 경량화에도 초점을 맞춘 '박성배 외(2005)[3]'의 연구도 제안되었다. 이 연구는 경량화에 초점을 맞추면서 메모리 기반 학습도 추가한 MODIFIED-IREP 방법을 사용하여 96.8%의 띄어쓰기 정확률을 보였다.

통계 기반 방법은 '이도길 외(2003)[5]'의 연구를 예로 들 수 있다. 여기서 제안하는 모델은 자동 띄어쓰기를 품사 부착과 같은 분류 문제로 간주하여, HMM을 일반화함으로써 확장된 문맥을 고려할 수 있고 보다 정확한 확률을 추정할 수 있도록 고안되었다. 여기서 제안한 모델의 실험 결과 복합명사까지 고려하여 98.33%의 음절 단위 정확도와 93.06%의 어절 단위 정확률을 얻었다. 하지만 음절 트라이그램까지 이용하며 성능 위주의 실험을 진행하였기에 시스템의 크기가 약 63.7MB 정도 된다는 것을 다른 논문[6]을 통해 알 수 있다.

마지막으로 두 방법을 혼합한 연구로는 '송영길 외(2009)[4]'의 연구가 있는데, 통계 기반으로 음절 유니그램 기반의 개량된 HMM을 사용하여 띄어쓰기 오류를 1차로 수정한다. 그리고, 정밀도 향상을 위해서 규칙 기반 시스템을 사용하여 음절 바이그램 이상의 어휘 규칙을 이용하여 잘못 수정된 띄어쓰기 오류를 재보정한다. 이에 대해 세종 코퍼스로 실험한 결과는 1MB 정도의 메모리 사용과 함께 94.14%의 비교적 높은 정밀도를 보였다.

3. 음절 정보 추출

3.1 자질1: 음절 자질 패턴

문장의 성격에 따라 하나의 음절이라도 각기 다른 자질을 가진다. 예를 들어, '의'라는 음절을 알아보자. '이 연구의 목적은 무엇인가?', '의사소통을 위해서 어떻게 해야 할까?'라는 문장이 있을 때, 앞 문장에서 사용된 '의'는 조사로서 사용되었고, 뒤 문장에서는 명사에 속하는 하나의 음절로 사용이 되었음을 알 수 있다. 이처럼 하나의 음절이라도 여러 개의 자질을 가지고 있다고 판단하여, 그 자질을 각각 패턴 형태로 만들고 음절 빈도를 이용한 확률을 계산하여 학습시킨다.

음절 자질을 패턴으로 만드는 구체적인 학습 방법은 다음과 같다. 여기서 w는 공백이 될 수 있는 하나의 음절을 의미한다.

표 2 '의' 음절에 대한 자질1 추출 예

문장	자질1 (w-2w-1w+1)
이 연구의 목적은 무엇인가?	001
의사 전달을 명확히 해주세요.	010
내 의견을 들어 주세요.	010
여의도역은 여기서 먼가요?	100
저의 생각은 다음과 같습니다.	001

표 3 세분화된 음절 자질의 구성

자질	설명
w-2	음절을 기준으로 앞 2번째 띄어쓰기 상태
w-1	음절을 기준으로 앞 1번째 띄어쓰기 상태
w+1	음절을 기준으로 뒤 1번째 띄어쓰기 상태

표 4 세분화된 음절 자질 패턴

음절	음절 자질 집합							
	0	1	2	3	4	5	6	7
...	...							
의	0	2/6	2/6	0	1/6	0	0	0
...	...							

※ 0~7은 음절이 가지는 '000' 패턴부터 '111' 패턴을 의미함

표 2는 '의'에 대한 자질의 패턴을 추출하고자 할 때 문장에서 가지는 패턴 번호를 나타내며, 표 3은 표 2에서 사용된 '자질'에 대한 설명이다. 0인 경우는 해당 위치에 띄어쓰기가 되어있지 않다는 의미이고, 1은 띄어쓰기가 되어있다는 의미이다. 하나의 음절에 대해 이렇게 세분화된 음절 자질 패턴은 표 4와 같은 형식으로 저장이 된다. 여기서, 한글 문장에 대한 명확한 띄어쓰기 확인을 위해 영어, 숫자, 특수 문자는 각각 E, N, S의 심볼로 변환하여 띄어쓰기를 수행한다. 하지만 문장을 구분 지을 수 있는 특수 문자(. / , / " / ' / ? / !)의 경우, 한글 음절과 마찬가지로 각각의 음절 자질 패턴을 추출한다. 입력된 문장에서 각각의 음절에서 계산되는 관찰확률(b_{io})은 식 (1)과 같다.

$$b_{io} = P(f_o|w_i), \qquad 0 \le o \le 7 \qquad (1)$$

즉, HMM에서 사용될 관찰확률은 학습된 음절 정보에서 i번째 음절(w_i)이 나왔을 때 해당 음절이 가진 패턴(f_o)이 나올 확률이다. 여기서 패턴 f는 0~7까지 8개의 자질 패턴을 나타낸다.

3.2 자질2: 두 음절 간 확률 정보

음절 자질 패턴만을 이용해서 띄어쓰기를 결정할 때 학습 말뭉치의 종류에 따라 음절이 가지는 띄어쓰기 특성이 달라질 수 있다. '이'라는 음절을 보면 대부분 주어

를 뒷받침해주는 조사로 사용되고, 무엇인가 가리키는 대명사로 사용되기도 있다. 하지만 '이론'이나 '이용'과 같이 앞이나 뒤에 나오는 음절에 따라 '이'라는 음절이 특정 단어의 일부분으로 사용될 경우가 있는데, 음절 하나의 자질 패턴만으로는 이러한 경우에 대해 띄어쓰기를 명확히 할 수 없다. 본 연구에서는 두 음절 간의 확률 정보를 이용하여 띄어쓰기에 적용 시킨다.

두 음절 간의 확률 정보 값은 HMM에서 전이확률(a_{ij})로 계산되어지는데, 추출하는 방식은 다음과 같다.

$$a_{ij} = P(1 \mid w_i, w_j), \qquad w_i, w_j \in W \tag{2}$$

식 (2)는 학습된 음절 정보에서 i번째의 음절 w_i와 j번째 음절 w_j가 있을 때, 두 음절 사이에 띄어쓰기가 들어갈 확률을 의미한다. 이렇게 전이확률을 계산해가는 과정에서 학습 말뭉치에 출현하지 않은 음절이 사용자 입력 문장에 나타날 수 있는데, 그럴 경우에는 띄어쓰거나 붙여 쓸 확률에 대해 동일한 값(0.5)을 넣어서 식(3)과 같이 계산한다. 여기서 W는 학습 데이터를 통해 얻어진 음절들의 집합이다.

$$a_{ij} = 0.5, \qquad w_i \notin W \text{ or } w_j \notin W \tag{3}$$

음절을 저장할 때는 데이터 부족 문제로 인해 음절쌍이 학습 데이터에 안 나오는 경우가 많으므로, 학습 데이터에 나타난 음절들에 대해서만 사전에 저장을 한다. 그리고 기준 음절(w_i)은 '_'로 구분하여 처음에만 저장하고 그 이후로는 뒤에 붙는 음절(w_j)과 확률 값(a_{ij})만 저장한다. 추가적으로 학습데이터에서 붙여쓰기만 나타난 음절쌍에 대해서는 '-1'의 값을 넣어 다른 값과 구분을 한다. 그림 2는 저장 방법에 대한 예이다.

```
...
_학
가      0.01
간      0.07
고      0.33
과      -1
교      0.03
...
```

그림 2 저장 방법의 예(a_{ij})

3.3 문장 띄어쓰기

최종적으로 문장 띄어쓰기를 하기 위해서는 앞서 추출한 자질1에 해당하는 관찰확률(b_{io})과 자질2에 해당하는 전이확률(a_{ij})을 이용한다. 두 가지 확률을 이용한 HMM 수식은 다음과 같다.

$$s_j(1) = \pi_j \tag{4}$$

$$s_j(t+1) = \max s_i(t) a_{ij} b_{io}, \qquad 0 \le o \le 7 \tag{5}$$

식 (4)는 식 (5)에서 사용할 초기상태를 지정하는 것으로, 제안 시스템에서 π의 의미는 문장 첫 음절의 자질 확률을 의미한다. 여기서 각각의 자질 패턴의 값은 앞서 설명하였듯이 8개를 사용하고, i, j는 학습 말뭉치에서 나온 음절에 해당한다. 식 (5)는 입력 문장에서 t번째까지의 음절에 대한 띄어쓰기 패턴들의 곱으로 시퀀스의 확률 값($s(t)$)을 구하고, 시퀀스의 확률 값($s(t)$)에서 w_i에서 w_j로의 전이확률(a_{ij})과 w_i에서의 관찰확률(b_{io})을 곱하여 $t+1$번째 시퀀스의 확률 값($s(t+1)$)을 구한다.

표 5 자질1의 예

음절	음절 자질 패턴 집합							
	0	1	2	3	4	5	6	7
학	0.16	0.10	0.17	0.05	0.25	0.16	0.10	0.01
교	0.13	0.18	0.12	0.02	0.18	0.26	0.10	0.01
간	0.15	0.13	0.32	0.08	0.17	0.12	0.02	0.01
다	0.28	0.12	0.17	0.07	0.20	0.09	0.05	0.02

표 6 자질2의 예

음절	음절 자질 패턴 집합	
	띄어 쓸 확률	붙여 쓸 확률
학교	0.03	0.97
교간	0.43	0.57
간다	0.12	0.88

'학교간다'라는 문장이 있을 때, 음절에 대한 확률이 표 5 및 표 6과 같이 나왔다고 가정하자.

그림 3을 통해 알 수 있듯이 제안 시스템은 다음 음절의 자질을 구할 때 모든 가능성을 고려하지 않는다. 만약 '학'의 자질 패턴이 '010'이 선택되었다면, 자질1은 '교'에 대해서 '100'과 '101'의 확률만 계산한다. 그리고 자질2는 '학교' 사이에 띄어 쓸 확률만 계산한다. 이러한 방식으로 자질 패턴마다 각각의 제한된 경로를 모두 계산하여 입력 문장에 대해 가장 높은 값을 가지는 시퀀스(S)를 출력한다. 최종적으로 이 문장에서 나오는 패턴의 시퀀스는 '2→5→2→4' 이다.

확률을 곱하는 과정에서 모바일 기기에 적용할 것을 감안하여 [1]의 연구를 참조하였다. 식 (5)에 log연산을

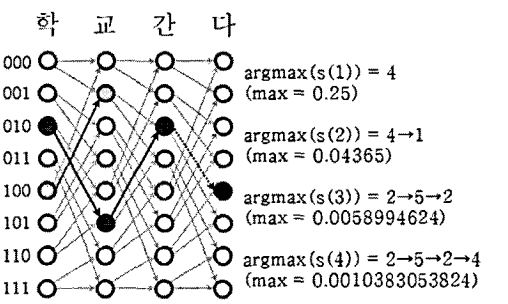

그림 3 음절 자질 종속성을 이용한 띄어쓰기 예

하고 부동소수점 연산을 없애기 위해 10^6을 곱하여 정수화 시킨다. 이는 확률을 log로 사용함에 있어서 언더플로우 현상을 막고자 하는 이유도 있다. 위의 과정을 통해 얻어진 수식은 식 (6)과 같다.

$$s_j(t+1) \approx \max \begin{pmatrix} [\log s_i(t) \times 10^6] \\ +[\log a_{ij} \times 10^6] \\ +[\log b_{io} \times 10^6] \end{pmatrix} \quad (6)$$

문장의 길이가 T인 입력 문장에 대해 최종적으로 나오는 시퀀스(S)에 대한 확률 값은 식 (7)과 같다.

$$P(S) = \max s_i(T) \quad (7)$$

4. 실험 및 성능 평가

4.1 문장 띄어쓰기

실험 환경은 같은 모바일 환경에서 적용시킬 목적으로 연구되어진 NB-HMM(Naive Baysian-Hidden Markov Model)을 통해 실험을 진행한 다른 연구[1]와 동일한 실험 환경을 구축하였다. 학습 말뭉치로는 21세기 세종계획 구어체 말뭉치 중 일부를 사용하였고, 실험 말뭉치는 객관적인 평가를 위해 MATAC '99'에서 사용한 ETRI 품사 부착 말뭉치를 사용하였다. 이에 대한 말뭉치의 특성은 표 7과 같다.

표 7 문장 띄어쓰기 실험 말뭉치의 특성

구분	말뭉치	문장 수(개)
학습 말뭉치	21세기 세종계획 말뭉치	809,914
실험 말뭉치	ETRI 품사 부착 말뭉치	27,858

4.2 평가 척도

문장 띄어쓰기에 대한 평가 척도로 식 (8)의 정밀도(accuracy)를 측정한다. 예를 들어 '이 연구의 목적은 무엇인가?'와 같은 문장의 경우 11개의 띄어쓰기 구간으로 구성되며 그 중에 8개의 구간이 맞았고 3개의 구간이 틀렸다면 8/11이 된다.

$$\text{정밀도} = \frac{\text{올바르게띄어쓴후보구간수}}{\text{전체띄어쓰기후보구간수}} \quad (8)$$

4.3 성능 비교 실험

4.3.1 자질1을 사용한 실험

자질1을 이용한 문장 띄어쓰기 성능을 알아보기 위해 '송영길 외(2009)[1]'의 모델과 비교하였다.

표 8은 [1]에서 제안한 시스템 중, 규칙 기반 후교정 모델 실험을 하기 전, NB-HMM만을 이용하여 통계적인 방법으로 띄어쓰기를 한 결과와 비교한 것이다. 위 실험에서 알 수 있듯이, 동일한 환경에서 학습 및 실험을 진행했을 때 제안한 시스템의 정밀도가 2.18% 더 높았고, 모델 크기는 약 53% 정도 감소했음을 확인할 수 있었다. 메모리 사용량을 줄일 수 있었던 원인을 분석해

표 8 자질1을 사용한 경우의 시스템 성능 비교

확률 모델	정밀도 (%)	메모리 사용량 (Byte)	학습 및 실험 말뭉치
제안시스템	*91.46*	*125,375*	21세기 세종계획 말뭉치
[1] (확률정보)	89.28	266,268	

본 결과 [1]에서 제안한 시스템은 필요한 확률 정보가 띄어쓰기 교정 지점을 중심으로 앞 3번째부터 뒤 2번째 음절 또는 일반화 심볼의 확률 정보를 필요로 한다. 하지만 제안시스템에서는 음절을 중심으로 앞 2번째부터 뒤 1번째까지 공백에 대한 확률 정보만을 필요로 하기 때문에 그만큼 메모리 사용량을 줄일 수 있었다.

4.3.2 자질1과 자질2를 모두 사용한 실험

표 9는 자질1과 자질2를 모두 사용한 경우와 '강승식(2001)[2]', '이도길 외(2007)[5]', '송영길 외(2009)[1]'의 성능을 비교하였다. [2]은 음절 바이그램을 사용하였다는 점에서 자질2를 추가적으로 사용한 제안시스템과 유사하기에 비교를 하였다. 그리고 [1,5]는 제안시스템과 같이 HMM을 사용하였는데, 수식에 사용되는 값들의 차이만으로 모델 크기에 차이가 많이 난다는 것을 나타내기 위해 비교를 실시하였다. 구체적으로, [5]는 음절간의 확률을 트라이그램까지 적용하여 실험을 진행하였지만, 본 연구에서 자질1은 앞의 음절과 상관없이 해당 음절이 가지는 주위 음절의 띄어쓰기 패턴만을 사용하고 그 방법이 메모리 면에서 굉장히 효율적임을 보이기 위해 비교를 실시하였다. 그리고 [1]은 교정 방법으로 사전 정보를 사용하지만, 본 연구에서는 오직 확률 정보만을 이용한다는 것을 강조하기 위해 비교하였다. 또한 시스템 간 비교를 일관성 있게 진행하기 위하여 동일한 학습 말뭉치와 평가 말뭉치를 사용하였다.

먼저 [2]와 비교했을 때, 제안시스템이 성능은 비슷하지만 메모리 사용이 약 76% 정도 적게 사용함을 알 수 있다. 이를 통해 제안시스템이 음절 바이그램 만을 사용한 시스템보다 모바일에 적합함을 알 수 있다. [5]는 제

표 9 자질1, 2 모두 사용한 경우의 시스템 성능 비교

확률 모델	정밀도 (%)	메모리 사용량 (Byte)	학습 말뭉치	실험 말뭉치
제안시스템 (확률정보)	94.05	1.0MB	21세기 세종계획 말뭉치	ETRI 품사부착 말뭉치
[2] (확률정보)	93.06	4.1MB		
[5] (확률정보)	97.48	63.7MB		
[1] (확률+사전)	94.47	1.1MB		

안시스템보다 3.5% 정도 높은 성능을 보이고 있지만, 사용된 모델크기에서 알 수 있듯이 모바일에서 사용하기는 힘든 점이 있음을 알 수 있다. 마지막으로 [1]의 경우, 정밀도는 제안시스템보다 약 0.4% 정도 높지만 제안시스템보다 10% 정도 많은 메모리를 사용한다는 사실을 알 수 있다. 두 시스템 모두 모바일 기기에 적합하다고 판단할 수 있으나, [1]는 후교정으로 오류 교정 규칙을 사용하였다. 본 연구의 시스템은 음절 자질, 상호 정보와 같은 확률 값만을 이용하여 위와 같은 성능을 내었기 때문에, [1]보다 띄어쓰기 시스템의 관리 및 확장이 용이하다고 볼 수 있다.

5. 결론 및 향후 연구

본 논문에서 적은 양의 메모리만 사용하는 한국어 자동 띄어쓰기 시스템을 제안하였다. 본 시스템은 학습 말뭉치에서 음절별 띄어쓰기 패턴 자질 및 음절 자질 간의 확률 정보를 추출하여 띄어쓰기를 실시한다. 이 두 가지 자질을 결합하여 사용함으로써 적은 양의 메모리만 사용하면서도 성능 면에서 최신 선행 연구에 견줄 수 있는 우수한 결과를 산출하였음을 확인할 수 있었다. 따라서 본 연구에서 제안한 시스템이 모바일 폰과 같이 메모리 사용 한도에 제한을 받을 수 있는 시스템에서 효과적으로 응용될 수 있을 것으로 판단할 수 있다.

향후 연구과제는 [1]에서 후교정으로 사용한 방법과 마찬가지로 오류 교정 규칙을 추가하여 성능을 향상시키는 방법에 대해 연구를 계획하고 있으며, 추가적으로 학습 말뭉치의 특성에 따라 띄어쓰기 문장의 성격이 어떻게 달라지는지에 대해서 비교 및 분석을 실시할 예정이다.

참 고 문 헌

[1] Y. Song, H. Kim, "An Automatic Korean Word Spacing System for Devices with Low Computer Power," *Journal of KIPS*, vol.16(B), no.4, pp.333-340, 2009. (in Korean)

[2] S. Kang, "Automatic Correction of Word-spacing Errors using by Syllable Bigram," *Journal of KSSS*, vol.8, no.2, pp.83-90, 2001. (in Korean)

[3] S. Park, E. Lee, Y. Tae, "Automatic word spacing in Korean for small memory devices," *Proc of the 18th international conference on Innocations in Applied Artificial Intelligence*, pp.249-258, 2005.

[4] S. Kang, "Eojeol-Block Bidirectional Algorithm for Automatic Word Spacing of Hangul Sentences," *Journal of KIIS : Software and Applications*, vol.27, no.4, pp.441-447, Apr. 2000. (in Korean)

[5] D. Lee, S. Lee, H. Lim, H. Rim, "Two Statictical Models for Automatic Word Spacing of Korean Sentences," *Journal of KIISS : Software and Applications*, vol.30, no.4, pp.358-371, Apr. 2003. (in Korean)

[6] S. Choi, M. Kang, H. Kwon, "Improving Korean Word-Spacing System Using Stochastic Information," *Proc. of the KCC-2004*, vol.31, no.1(B), pp.883-885, Apr. 2004. (in Korean)

김 신 일

2010년 동아대학교 컴퓨터공학과(학사)
2010년~현재 동아대학교 컴퓨터공학과 석사과정. 관심분야는 자연어처리, 문장 자동 띄어쓰기, 의존 파싱 등

양 선

정보과학회논문지 : 소프트웨어 및 응용 제 37 권 제 2 호 참조

고 영 중

정보과학회논문지 : 소프트웨어 및 응용 제 37 권 제 2 호 참조