확장된 음절 bigram을 이용한 자동 띄어쓰기 시스템

임동희, 전영진, 김형준, 강승식
국민대학교 컴퓨터학부
{nlp,terrius7}@cs.kookmin.ac.kr, dictions@nate.com, sskang@kookmin.ac.kr

Word Segmentation System Using Extended Syllable bigram

Dong-Hee Lim, Young-Jin Chun, Hyoung-Joon Kim, Seung-Shik Kang
Dept. of Computer Science, Kookmin University

요 약

본 논문은 통계 기반 방법인 음절 bigram을 이용한 자동 띄어쓰기를 기본 방법으로 하고 경우의 수를 세분화한 확장된 음절 bigram을 이용한 공백 확률, 띄어쓰기 통계를 바탕으로 최종 띄어쓰기 임계치 차등 적용, 에러 사전 적용 3가지 방법을 추가로 사용하는 경우 기본적인 방법만을 쓴 경우보다 띄어쓰기 정확도가 향상된다는 것을 확인하였다. 그리고 해당 음절에 대한 bigram이 없는 경우 확장된 음절 unigram을 통해 근사적으로 계산해 데이터부족 문제를 개선하였다. 한국어 말뭉치와 중국어 말뭉치에 대한 실험을 통해 본 논문에서 제안하는 방법이 한국어 자동 띄어쓰기뿐만 아니라 중국어 단어 분리에 적용할 수 있다는 것도 확인하였다.

1. 서론

띄어쓰기가 없거나 잘못된 문장에 대해 정보처리를 할 경우 잘못된 결과가 나오게 된다. 한 예로 정보 검색을 위해 수집된 문서에 대해 띄어쓰기를 기준으로 하여 색인어를 추출할 때 잘못된 띄어쓰기가 존재하면 잘못된 색인어가 추출되고 이는 정보 검색 시스템의 성능을 떨어뜨린다. 그리고 띄어쓰기는 한국어뿐 아니라 띄어 쓰기가 없는 중국어에 대한 정보처리를 할 경우에도 매우 중요한 변수로 작용된다.

자동 띄어쓰기 방법은 크게 통계 기반 방법, 규칙 기반 방법으로 구분할 수 있다. 통계 기반 방법은 손쉽게 구현할 수 있지만 학습 말뭉치에 의존적인 결과가 나오며 자료 부족 문제도 발생한다. 규칙 기반 방법은 구현 시 어휘 지식이 필요하다. 그러나 어휘 지식 구축 및 유지보수에는 많은 비용이 든다.

본 논문은 통계 기반 방법으로 확장된 음절 bigram을 이용한 띄어쓰기 시스템을 구현하였다.

2. 자동 띄어쓰기

음절 bigram을 이용하여 자동 띄어쓰기를 하는 방법, 확장된 음절 bigram을 구축하는 방법, 그리고 음절 bigram 데이터 부족 문제를 개선하기 위해 확장된 음절 unigram을 사용하는 방법을 설명한다.

2.1. 음절 bigram

강승식(2001)은 말뭉치에서 각 bigram 음절쌍 <X,Y>에 대해 공백의 출현 위치에 따라 좌공백 빈도, 우공백 빈도, 사이공백 빈도, 총 출현 횟수를 계산하여 임의의 두 음절 사이에 공백이 삽입될 확률을 계산하였다. 최종적으로 계산한 값이 실험에 의해 결정한 경험적 임계치를 넘으면 공백을 삽입하는 것으로 결정하였다.

$$P(X_i, X_{i+1}) = 0.25 \cdot P_R(X_{i-1}, X_i) + 0.5 \cdot P_M(X_i, X_{i+1}) + 0.25 \cdot P_L(X_{i+1}, X_{i+2})$$

$$P_R(X_{i-1}, X_i) = freq_R(X_{i-1}, X_i) / freq(X_{i-1}, X_i)$$

$$P_M(X_i, X_{i+1}) = freq_M(X_i, X_{i+1}) / freq(X_i, X_{i+1})$$

$$P_L(X_{i+1}, X_{i+2}) = freq_L(X_{i+1}, X_{i+2}) / freq(X_{i+1}, X_{i+2})$$

위의 식에서 $freq_R(x_{i-1}, x_i)$, $freq_M(x_{i-1}, x_i)$, $freq_L(x_{i+1}, x_{i+2})$는 음절 쌍에 해당하는 우공백 빈도수, 사이공백 빈도수, 좌공백 빈도수이다. 그리고 $freq(x_{i-1}, x_i)$, $freq(x_i, x_{i+1})$, $freq(x_{i+1}, x_{i+2})$는 음절 쌍에 해당하는 총 빈도수이다.

[1] 본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

2.2. 확장된 음절 bigram 구축

2.1에서는 좌공백 빈도, 우공백 빈도, 사이공백 빈도, 그리고 총 출현횟수에 대해 빈도 정보를 구성하였다. 하지만 음절 bigram 정보는 총 8가지 경우가 생기고 그 빈도를 말뭉치에서 계산할 수 있다. 음절쌍 <X, Y>의 8가지 경우에 대한 빈도수를 계산하는 구체적인 예는 다음과 같다.

- "XY", "XY ", "X Y", "X Y ",
  " XY", " XY ", " X Y", " X Y "

위의 예에서 공백이 없다는 것을 $_{(0)}$으로 공백이 있다는 것을 $_{(1)}$로 표기하면 다음과 같이 표기할 수 있다.

- $_{(0)}X_{(0)}Y_{(0)}$, $_{(0)}X_{(0)}Y_{(1)}$, $_{(0)}X_{(1)}Y_{(0)}$, $_{(0)}X_{(1)}Y_{(1)}$,
  $_{(1)}X_{(0)}Y_{(0)}$, $_{(1)}X_{(0)}Y_{(1)}$, $_{(1)}X_{(1)}Y_{(0)}$, $_{(1)}X_{(1)}Y_{(1)}$

위의 표기는 다음과 같이 공백 표시를 우측에 함께 표기할 수 있다.

- $XY_{(000)}$, $XY_{(001)}$, $XY_{(010)}$, $XY_{(011)}$,
  $XY_{(100)}$, $XY_{(101)}$, $XY_{(110)}$, $XY_{(111)}$

"나는 학생이고, 너는 학생이 아니다." 라는 문장을 가지고 확장된 음절 bigram을 구축하는 과정은 다음과 같다. 단, 문장 처음과 끝은 띄어 쓴다고 가정한다.

연속된 두 음절(문장부호도 하나의 음절로 취급)씩 음절쌍으로 묶고 위에서 설명한 8가지 중 어느 경우에 해당하는지 표시를 하면 "나는$_{(101)}$ 는학$_{(010)}$ 학생$_{(100)}$ 생이$_{(000)}$ 이고$_{(000)}$ 고,$_{(001)}$,너$_{(010)}$ 너는$_{(101)}$ 는학$_{(010)}$ 학생$_{(100)}$ 생이$_{(001)}$ 이아$_{(010)}$ 아니$_{(100)}$ 니다$_{(000)}$ 다.$_{(001)}$" 이 된다. 음절 쌍을 오름차순으로 정렬을 하면 ",너$_{(010)}$ 고,$_{(001)}$ 나는$_{(101)}$ 너는$_{(101)}$ 는학$_{(010)}$ 는학$_{(010)}$ 니다$_{(000)}$ 다.$_{(001)}$ 생이$_{(000)}$ 생이$_{(001)}$ 아니$_{(100)}$ 이고$_{(000)}$ 이아$_{(010)}$ 학생$_{(100)}$ 학생$_{(100)}$" 을 얻을 수 있고 같은 음절쌍끼리 빈도를 합산하면 최종적으로 표1을 얻을 수 있다.

표 1. 확장된 음절 bigram의 예

음절쌍	(000)	(001)	(010)	(011)	(100)	(101)	(110)	(111)
,너	0	0	1	0	0	0	0	0
고,	0	1	0	0	0	0	0	0
나는	0	0	0	0	0	1	0	0
너는	0	0	0	0	0	1	0	0
는학	0	0	2	0	0	0	0	0
니다	1	0	0	0	0	0	0	0
다.	0	1	0	0	0	0	0	0
생이	1	1	0	0	0	0	0	0
아니	0	0	0	0	1	0	0	0
이고	1	0	0	0	0	0	0	0
이아	0	0	1	0	0	0	0	0
학생	0	0	0	0	2	0	0	0

위와 같은 방법을 이용해서 말뭉치에 대한 확장된 음절 bigram을 구축한다.

2.3. 데이터 부족 문제 개선

통계 기반 방법에서 발생하는 문제인 데이터 부족 문제를 확장된 음절 unigram을 통해 간접적으로 계산할 수 있다는 것을 이용하였다. 말뭉치가 작더라도 시스템을 구성할 수 있게 한다는 점에서 말뭉치의 의존 문제를 개선한다고도 볼 수 있다.

확장된 음절 unigram 정보는 총 4가지 경우가 생기고 그 빈도를 말뭉치에서 계산할 수 있다. 음절 <X>의 4가지 경우에 대한 빈도수를 계산하는 구체적인 예는 다음과 같다.

- "X", "X ", " X", " X "
- $_{(0)}X_{(0)}$, $_{(0)}X_{(1)}$, $_{(1)}X_{(0)}$, $_{(1)}X_{(1)}$
- $X_{(00)}$, $X_{(01)}$, $X_{(10)}$, $X_{(11)}$

"나는 학생이고, 너는 학생이 아니다." 라는 문장을 가지고 확장된 음절 unigram을 구축하는 과정은 2.2와 비슷한 과정을 가진다.

"나$_{(10)}$ 는$_{(01)}$ 학$_{(10)}$ 생$_{(00)}$ 이$_{(00)}$ 고$_{(00)}$,$_{(01)}$ 너$_{(10)}$ 는$_{(01)}$ 학$_{(10)}$ 생$_{(00)}$ 이$_{(01)}$ 아$_{(10)}$ 니$_{(00)}$ 다$_{(00)}$.$_{(01)}$" 를 음절에 대해 오름차순으로 정렬 하여 ",$_{(01)}$.$_{(01)}$ 고$_{(00)}$ 나$_{(10)}$ 너$_{(10)}$ 는$_{(01)}$ 는$_{(01)}$ 니$_{(00)}$ 다$_{(00)}$ 생$_{(00)}$ 생$_{(00)}$ 아$_{(10)}$ 이$_{(00)}$ 이$_{(01)}$ 학$_{(10)}$ 학$_{(10)}$" 을 얻는다. 같은 음절끼리 빈도를 합산하면 표2를 얻을 수 있다.

표 2. 확장된 음절 unigram의 예

음절	(00)	(01)	(10)	(11)
,	0	1	0	0
.	0	1	0	0
고	1	0	0	0
나	0	0	1	0
너	0	0	1	0
는	0	2	0	0
니	1	0	0	0
다	1	0	0	0
생	2	0	0	0
아	0	0	1	0
이	1	1	0	0
학	0	0	2	0

음절쌍 <X,Y>에 대한 bigram이 없는 경우 확장된 unigram을 통해 $P_{(000)}(X,Y)$을 계산하는 방법은 다음과 같다. 이 때 X와 Y의 발생 확률은 독립적이라고 가정한다. 하지만 중간에 띄어쓰기 유무가 X에 종속되는지 아

니면 Y에 종속되는지에 따라 두 가지 방법이 나온다. 본 논문에서는 X에 종속된다고 보고 첫 번째 식을 사용하였다.

$$P_{(000)}(X,Y)=\frac{freq_{00}(X)}{freq_{00}(X)+freq_{01}(X)+freq_{10}(X)+freq_{11}(X)}\times\frac{freq_{00}(Y)}{freq_{00}(Y)+freq_{01}(Y)}$$

$$P_{(000)}(X,Y)=\frac{freq_{00}(X)}{freq_{00}(X)+freq_{10}(X)}\times\frac{freq_{00}(Y)}{freq_{00}(Y)+freq_{01}(Y)+freq_{10}(Y)+freq_{11}(Y)}$$

$P_{(001)}(X,Y)$, $P_{(010)}(X,Y)$, $P_{(011)}(X,Y)$, $P_{(100)}(X,Y)$, $P_{(101)}(X,Y)$, $P_{(110)}(X,Y)$, $P_{(111)}(X,Y)$도 위와 같은 방법으로 계산할 수 있다.

3. 띄어쓰기 알고리즘

음절 bigram을 이용한 자동 띄어쓰기를 방법(1)로 하고 확장된 음절 bigram을 이용해 공백 확률을 구하는 것을 방법(2), 띄어쓰기 통계를 바탕으로 최종 띄어쓰기 임계치 차등 적용을 방법(3), 에러 사전을 적용하는 것을 방법(4)로 하여 단계별로 적용한다.

3.1. 확장된 음절 bigram 이용

우선적으로 확실하게 띄어 쓰거나 붙여 쓴다는 것이 부분적으로 결정이 되어 있어야 경우의 수를 세분화한 확장된 음절 bigram을 이용해서 좀 더 그 경우에 맞는 통계정보를 적용할 수 있다. 이를 위해서 2.1에서 소개한 방법(1)에서 하나의 임계치를 가지고 띄어 쓰거나 붙여 쓰는 것을 단순하게 결정 내리지 않고 확실하게 붙여 쓰는 임계치와 띄어 쓰는 임계치를 추가로 설정해서 확실하게 붙여 쓰는 것과 띄어 쓰는 것을 결정한다. 그리고 확실하게 결정되지 않은 것들에 대해 그 경우에 해당하는 확장된 음절 bigram만을 가지고 공백 확률을 구하여 최종적인 임계치를 가지고 띄어쓰기를 결정한다.

임의의 음절열 "$S_0S_1S_2S_3S_4S_5S_6S_7S_8$" 이 주어 졌을 때 우선 방법(1)을 이용하여 공백 삽입 확률을 구하고 확실하게 붙여 쓰는 임계치보다 작은 경우 확실하게 붙여 쓴다는 것을 결정하고 확실하게 띄어 쓰는 임계치보다 큰 경우 확실하게 띄어 쓴다는 것을 결정한다. 확실하게 결정한 결과가 "$_{(1)}$ S_0 $_{(1)}$ S_1 S_2 $_{(0)}$ S_3 S_4 $_{(0)}$ S_5 S_6 S_7 $_{(0)}$ S_8 $_{(1)}$" 인 경우 확실하게 결정되지 않은 "S_1S_2" 의 사이 공백확률을 확장된 음절 bigram을 이용해서 재계산하는 구체적인 방법은 다음과 같다.

$$P(S_1,S_2)=0.25\cdot P_R(S_0,S_1)+0.5\cdot P_M(S_1,S_2)+0.25\cdot P_L(S_2,S_3)$$

$$P_R(S_0,S_1)=\frac{freq_{\underline{1}11}(S_0,S_1)}{freq_{\underline{1}10}(S_0,S_1)+freq_{\underline{1}11}(S_0,S_1)}$$

$$P_M(S_1,S_2)=\frac{freq_{\underline{1}10}(S_1,S_2)}{freq_{\underline{1}00}(S_1,S_2)+freq_{\underline{1}10}(S_1,S_2)}$$

$$P_L(S_2,S_3)=\frac{freq_{\underline{1}00}(S_2,S_3)+freq_{\underline{1}01}(S_2,S_3)}{freq_{\underline{0}00}(S_2,S_3)+freq_{\underline{0}01}(S_2,S_3)+freq_{\underline{1}00}(S_2,S_3)+freq_{\underline{1}01}(S_2,S_3)}$$

위 식에서 $_{\underline{0}}$과 $_{\underline{1}}$은 방법(1)을 이용하여 확실한 임계치를 적용했을 때 확실하게 결정된 것을 나타낸다. 나머지 "S_3S_4", "S_5S_6", "S_6S_7" 의 사이 공백확률도 위와 같은 방법으로 재계산할 수 있다.

3.2. 임계치 차등 적용

한글 말뭉치(세종계획 원시 말뭉치[7], 1998년과 1999년)에서 한 어절의 평균 음절의 수를 계산해 보면 약 3.2가 되는 것을 알 수 있다. 이는 음절 3.2개마다 공백이 나온다는 것이다. 방법(1)과 방법(2)에 의해 연속적으로 붙여 쓰는 것으로 결정이 되었더라도 그 어절의 음절 수가 5음절 이상인 경우에 한에서 완화된 임계치를 적용하여 두 어절(3음절 이상과 2음절 이상 또는 2음절 이상과 3음절 이상)로 띄어쓰기가 이루어지도록 한다. 연속된 두 어절의 음절수가 둘 다 2음절 이하인 경우에는 위와 반대로 엄격한 임계치를 적용하여 한 어절로 붙여 쓰게 한다.

SIGHAN 2005에서 제공된 중국어 말뭉치[8]에서 어절 길이의 평균을 계산해 보면 약 1.6이 된다. 따라서 중국어 띄어쓰기에 대해서는 한 어절의 음절 수가 3음절 이상인 경우에 한에서 완화된 임계치를 적용하여 두 어절(1음절 이상과 2음절 이상 또는 2음절 이상과 1음절 이상)로 띄어 쓰게 한다. 1음절과 1음절 형태로 띄어 쓰는 것으로 결정된 것도 엄격한 임계치를 적용하여 2음절로 붙여 쓰게 한다.

3.3. 에러 사전 구축

시스템이 결정을 잘못 내리는 오류들에 대해 에러 사전을 구축하여 성능을 향상시킬 수 있다. 기본적인 에러 사전 구축은 학습이 완료된 자동 띄어 쓰기 시스템에 학습 말뭉치를 입력으로 하여 출력된 결과에 대한 띄어쓰기 에러를 찾아낸다. 에러는 붙여 쓰는 곳을 띄어 쓴 삽입 에러, 띄어 쓰는 곳을 붙여 쓴 삭제 에러 두 가지가 있다. 각각의 에러에 대해 발생한 위치 앞뒤 2음절씩 총 4음절을 에러 종류 표시와 함께 에러 사전에 추가를 한다.

언어적으로 띄어쓰기가 두 가지 이상으로 허용되는 경우나 말뭉치의 띄어쓰기 오류로 인해 에러로 판단되어 에러사전에 포함되는 것을 피해야 한다. 이를 위해서 에러로 판단되는 음절열을 가지고 말뭉치에서 띄어 쓴 경우와 붙여 쓴 경우의 빈도를 조사하였다. 그 결과를 바탕으로 빈도를 비교하여 높은 쪽으로 결정되도록 에러사전을 구축하였다.

에러 사전에 있는 음절열에 대해서는 정해진 띄어 쓰

기만을 하게 된다. 삽입 에러로 표시된 경우는 무조건 붙여 쓰고, 삭제 에러로 표시된 경우는 무조건 띄어 쓴다.

4. 실험 및 결과 분석

학습과 테스트 말뭉치는 한글 말뭉치(세종계획 원시 말뭉치, 1998년과 1999년) 2개와 중국어 말뭉치(Second International Chinese Word Segmentation Bakeoff에서 사용된 말뭉치 - City University of Hong Kong, Peking University, Microsoft Research) 3개를 사용하였다. 한글에 대한 객관적인 테스트 말뭉치가 없으므로 한글 말뭉치를 99:1로 나누어 학습과 테스트 말뭉치로 사용하였고 중국어에 대해서는 학습 말뭉치와 테스트 말뭉치가 각각 존재하므로 그대로 사용하였다.

테스트 말뭉치는 공백을 모두 없애고 자동 띄어쓰기 시스템의 입력으로 사용하여 시스템이 출력한 결과와 공백을 없애기 전의 테스트 말뭉치와 비교하였다.

표 3. 학습 말뭉치 통계

말뭉치	어절 종류	총 어절 수	음절 종류	총 음절 수
세종98	1,220,248	7,241,808	6,754	23,125,003
세종99	1,435,377	7,837,668	6,563	25,826,091
cityu	69,085	1,455,629	4,923	2,403,355
pku	55,303	1,109,947	4,698	1,826,448
msr	88,119	2,368,391	5,167	4,050,469

표 4. 테스트 말뭉치 통계

말뭉치	어절 종류	총 어절 수	음절 종류	총 음절 수
세종98	39,725	73,110	2,900	233,348
세종99	44,378	79,052	2,570	260,774
cityu	9,000	40,936	2,701	67,689
pku	13,148	104,372	2,934	172,733
msr	12,923	106,873	2,838	184,355

방법 (1), (2), (3), (4)를 단계별로 적용하였을 경우 어절 단위 recall, precision, 그리고 F-score 값의 변화를 확인하였다. F-score는 다음과 같이 계산된다.

$$\text{F-score} = \frac{2\times\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}$$

평가를 위해 First International Chinese Word Segmentation Bakeoff에서 사용된 간단한 Perl script [9]를 사용하였다.

임계치를 변화시키면서 F-score가 가장 높은 값을 임계치로 결정하여 사용하였고 각각의 말뭉치마다 각각의 임계치를 구하였다. 말뭉치 하나마다 확실한 붙여 쓰기 임계치(T_1), 확실한 띄어쓰기 임계치(T_2), 최종 띄어쓰기 임계치(T_3), 완화된 띄어쓰기 임계치(T_4), 엄격화된 띄어쓰기 임계치(T_5) 총 5개의 임계치를 가진다. 방법(1)에서는 T_3으로만 띄어쓰기가 결정되고 방법(2)에서는 T_1, T_2가 추가적으로 사용된다. 방법(3)에서는 T_4, T_5가 사용된다. 방법(4)는 임계치가 사용되지 않는다.

표 5. 사용된 임계치

말뭉치	T_1	T_2	T_3	T_4	T_5
세종98	0.27	0.61	0.45	0.38	0.51
세종99	0.29	0.60	0.43	0.43	0.51
cityu	0.35	0.69	0.47	0.44	0.51
pku	0.33	0.72	0.47	0.46	0.49
msr	0.31	0.68	0.46	0.46	0.48

각 말뭉치의 결과를 살펴보면 방법(1)만을 적용할 때보다 방법(2), 방법(3), 방법(4)를 단계별로 적용할 때 성능이 좋아지는 것을 확인할 수 있다. 한국어 말뭉치에 대한 성능이 중국어 말뭉치에 대한 성능보다 안 좋은 이유는 한국어 말뭉치가 정제되지 않아서 띄어쓰기 오류가 학습 말뭉치와 테스트 말뭉치에 상당한 비율로 존재하기 때문이라고 생각된다.

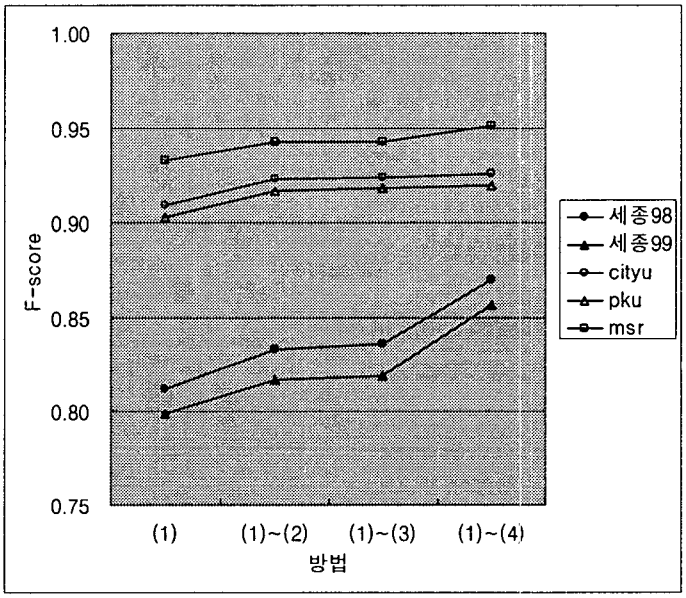

그림 1. 방법에 따른 성능 차이

표 6. 세종 말뭉치에 대한 결과

방법	recall		precision		F-score	
	98	99	98	99	98	99
(1)	0.807	0.799	0.818	0.798	0.812	0.799
(1) ~ (2)	0.835	0.824	0.831	0.811	0.833	0.817
(1) ~ (3)	0.843	0.824	0.829	0.814	0.836	0.819
(1) ~ (4)	0.870	0.856	0.870	0.857	0.870	0.857

표 7. cityu 말뭉치에 대한 결과

방법	recall	precision	F-score
(1)	0.919	0.899	0.909
(1) ~ (2)	0.930	0.916	0.923
(1) ~ (3)	0.931	0.917	0.924
(1) ~ (4)	0.932	0.920	0.926

표 8. pku 말뭉치에 대한 결과

방법	recall	precision	F-score
(1)	0.898	0.908	0.903
(1) ~ (2)	0.910	0.924	0.917
(1) ~ (3)	0.911	0.925	0.918
(1) ~ (4)	0.912	0.929	0.920

표 9. msr 말뭉치에 대한 결과

방법	recall	precision	F-score
(1)	0.940	0.927	0.933
(1) ~ (2)	0.948	0.937	0.943
(1) ~ (3)	0.948	0.938	0.943
(1) ~ (4)	0.951	0.951	0.951

은닉 마르코프 모델을 기반으로 하는 띄어쓰기 시스템[4]에서 제시된 성능을 알아 본다면, [4]의 복합 명사를 고려하지 않은 결과 중 음절 bigram 사용 모델에 대한 결과는 음절 단위 정확도 0.973, 어절 단위 재현율 0.883, 어절 단위 정확률 0.884이다. 테스트 말뭉치, 평가 도구, 평가 항목 등 실험 환경이 다르므로 본 시스템의 성능과 객관적인 비교는 어렵지만 복잡한 모델을 사용하지 않더라도 비슷한 성능을 가지는 것을 알 수 있다.

5. 결론

이전의 음절 bigram 통계 기반 띄어쓰기 시스템이 어느 정도의 성능을 보이지만 완벽하게 모든 경우에 대해서 올바른 결과를 주지는 못한다는 것을 관찰할 수 있었다. 이를 보안하는 방법을 추가로 제안하고 적용해 성능 향상을 확인하였다. 이후 규칙 기반 방식을 추가로 적용하는 것을 고려해 보고 그에 따라 성능을 향상시키는 방안을 모색할 필요가 있다.

참고 문헌

[1] 강승식, "음절 bigram을 이용한 띄어쓰기 오류의 자동 교정", 음성과학회논문지, 제8권 제2호, pp.83-90, 2001.

[2] Kang, S. S. and C. W. Woo, Automatic Segmentation of Words using Syllable Bigram Statistics, Proceedings of NLPRS'2001, pp.729-732, 2001.

[3] 심광섭, "음절간 상호 정보를 이용한 한국어 자동 띄어쓰기", 정보과학회논문지, 제23권 제9호, pp.991-1000, 1996.

[4] 이도길, 이상주, 임희석, 임해창, "한글 문장의 자동 띄어쓰기를 위한 두 가지 통계적 모델", 정보과학회논문지, 소프트웨어 및 응용 제30권 제4호, pp.358-370, 2003.

[5] Nakagawa, Tetsuji, Chinese and Japanese Word Segmentation Using Word-Level and Character-Level Information, In Proceedings of COLING, pp.466-472, 2004.

[6] Richard Sproat and Tom Emerson, "The First International Chinese Word Segmentation Bakeoff", In proceedings of the Second SIGHAN Workshop on Chinese Language Processing, 2003.

[7] 21세기 세종계획 국어기초자료 구축, 문화관광부 (1998, 1999).

[8] Second International Chinese Word Segmentation Bakeoff(2005), http://www.sighan.org/bakeoff2005/

[9] http://www.sighan.org/bakeoff2003/score