음절 바이그램 단순화 기법에 의한 한국어 자동 띄어쓰기 시스템의 성능 개선*

국민대학교 컴퓨터학부, 첨단정보기술연구센터
강 승 식

Improvement of Automatic Word Segmentation of Korean by Simplifying Syllable Bigram

Seung-Shik Kang
Department of Computer Science, Kookmin University & AITrc, Seoul, Korea

요 약

한글 문서의 자동 띄어쓰기는 웹 문서와 검색 질의어, 법률안 제목, 문자 메시지 등에서 띄어 쓰지 않은 문장에 대해 자동으로 공백을 삽입해 주는 기능이다. 기존의 자동 띄어쓰기 기법은 각 문자 경계마다 공백 삽입 일치도를 비교하는 방식으로 평가되었으나, 실제 응용 시스템에서는 어절 인식 정확률이 높고, 공백의 과생성 오류가 적으며, 바이그램 데이터 크기가 작아야 한다. 본 논문에서는 이러한 요구 조건에 따라 새로운 평가 기준을 제시하고, 이에 따라 기존 방법보다 바이그램 데이터 크기가 매우 작고, 정확률이 높은 자동 띄어쓰기 방법을 제안하였다.

서 론

한글 문장의 띄어쓰기 오류는 '띄어쓴 오류'와 '붙여쓴 오류'로 구분된다. 그런데 띄어쓰기 오류에서 붙여써야 할 것을 잘못 띄어쓴 오류는 거의 발생하지 않으며, 띄어써야 할 것을 붙여쓴 오류가 대부분이다. 따라서 붙여쓴 오류에 대해 자동 띄어쓰기에 의해 자동으로 공백을 삽입해 주는 기능이 필요하다. 한글 자동 띄어쓰기는 고의로 붙여쓰거나 혹은 맞춤법 규정에 익숙하지 않은 사용자들이 띄어쓰기를 하지 않은 문서에 대해 자동으로 공백을 삽입해 주는 기능을 수행한다.

- 띄어쓴 오류 : 붙여써야 하는데 띄어쓴 오류.
- 붙여쓴 오류 : 띄어써야 하는데 붙여쓴 오류.

자동 띄어쓰기의 문제의 해결 방안은 "띄어쓴 오류에서 공백을 삭제하여 붙여주는 기능"이라는 접근 방식과 "붙여쓴 오류에서 공백을 삽입하여 띄어주는 기능"이라는 두 가지 방향에서 접근할 수 있다. 예를 들어, "한글은 과학적인 문자이다"에 대한 자동 띄어쓰기 문제는 (1)과 같이 모든 문자를 무조건 띄어쓴 초기 상태로부터 띄어쓰기 오류를 유발한 공백을 제거하는 방법과, (2)와 같이 공백이 모두 제거된 상태에서 붙여쓰기 오류가 발생한 위치에 공백을 삽입하는 방법이 있다.

(1) "한 글 은 과 학 적 인 문 자 이 다".
(2) "한글은과학적인문자이다".

띄어쓰기 문제의 두 가지 접근 방식 중에서 공백 제거 방식은 이웃한 두 개의 음절에 대한 붙여쓸 확률에 의해 공백 제거 확률이 높은 위치의 공백들을 제거하는 알고리즘으로 기술된다. 이에 비해, 공백 삽입 방식은 초기 상태로서 공백이 전혀 없는 상태에서 시작하여 필요한 위치에 공백을 삽입한다.

그런데 공백 제거 방식을 적용할 때 제거되어야 할 공백

*본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.
E-mail : sskang@kookmin.ac.kr

의 수는 공백 삽입 방식을 적용할 때 삽입되어야 할 공백의 수에 비해 매우 많기 때문에 공백 삽입 방식이 더 자연스럽다. 위 예제에서 공백 제거 방식은 8개의 공백이 제거되어야 하지만, 공백 삽입 방식은 2개의 공백만 삽입하면 된다. 이러한 차이점은 전체 문자수에 비해 공백 개수가 적기 때문이다. 한글 문장의 평균 어절 길이가 2.7~3.2 음절이므로 대략 3 음절마다 공백이 1개씩 삽입되고, 문장의 끝 어절 뒤에는 공백이 삽입되지 않는다. 따라서 평균적으로 공백 제거 개수는 공백 삽입 개수보다 2배 이상 많기 때문에 공백 삽입 방식이 더 자연스럽다.

자동 띄어쓰기 기법에는 형태소 분석 기법을 이용한 분석적인 접근법(analytic approach)과 말뭉치로부터 띄어쓰기 관련 정보를 습득하여 통계적으로 공백 삽입 확률을 계산하는 통계적 기법(statistical approach)이 있다. 분석적인 접근법은 조사/어미의 음절 특성과 어절 분리-결합 등 형태소 분석 기법을 이용하여 어절을 인식하는 방법이다(김계성 등, 1998 ; 강승식, 2000a). 즉, 형태소 분석에서 사용되는 분석 기법을 자동 띄어쓰기에 적용하는 것으로 강승식(2000a)은 띄어쓸 확률이 매우 높다고 판단되는 어절 블록을 설정하고 어절 블록 내에서 형태소 분석기를 이용하여 어절을 인식한다(강승식, 2000a).

통계적인 기법에서는 말뭉치에서 습득된 바이그램 단위의 띄어쓰기 정보인 좌공백, 우공백, 사이 공백 확률을 조합하여 공백 삽입 여부를 결정한다(심광섭, 1996 ; 강승식, 2000b). 이외에도 바이그램 정보와 동적 프로그래밍 기법을 이용한 어절 인식 알고리즘이 있다(신중호, 박혁로, 1997). 분석적 접근법은 정확도를 향상시키기 위해 알고리즘을 수정하거나 규칙을 정교하게 작성하는데 많은 노력이 필요한데 비해, 통계적 접근법은 통계 데이터에 의해 쉽게 일정 수준의 정확도를 달성할 수 있다.

통계적 기법은 음절 바이그램 데이터의 크기가 커서 응용 시스템에 따라 기억장소 사용 제약을 만족시키기 어려운 경우가 발생한다. 또한, 기존의 통계적 방법은 각 문자 경계마다 공백 삽입 확률을 계산하는 공백 삽입 일치도를 기준으로 알고리즘이 고안되었기 때문에 어절 인식 정확도가 낮다. 본 논문에서는 통계적 기법의 문제점을 해결하는 방안으로 음절 바이그램 데이터의 크기를 매우 작게 유지하고, 자동 띄어쓰기 시스템의 새로운 평가 방식에 의해 어절 인식 정확도를 높이는 방법을 제안한다.

자동 띄어쓰기 시스템의 평가 방식

자동 띄어쓰기 시스템의 평가 기준은 모든 문자와 문자 사이의 공백 삽입 가능 위치에서 공백을 삽입하거나 삽입하지 않는 정확도를 계산하는 '공백 삽입 일치도'와 실험 문서에 포함된 어절을 기준으로 띄어쓰기에 의해 정확히 인식된 어절 개수를 계산하는 '어절 인식 정확도'가 사용되어 왔다. 이 중에서 '어절 인식 정확도'가 유용한 평가 기준이라고 볼 수 있으나, 이 방식에서는 공백 1개의 삽입/삭제 오류가 어절 인식 정확도에 미치는 영향이 크기 때문에 주로 '공백 삽입 일치도'를 평가 기준으로 사용하고 있다(심광섭, 1996 ; 강승식, 2000b).

- 공백 삽입 일치도 : 모든 공백 위치에서 공백 일치도.
- 어절 인식 정확도 : 어절 일치도.

그런데 공백 삽입 일치도에 의한 평가 기준은 동일한 정확도라고 하더라도 실질적인 활용성 측면에서는 차이가 있다는 점을 반영하지 못하고 있기 때문에 자동 띄어쓰기 정확도를 평가하는데 부적합한 점이 있다. 예를 들어, 아래 예에서 (3)은 삽입된 공백의 수가 1개이고, (4)는 삽입된 공백이 3개로서 공백 삽입 일치도는 모두 90%(10개 중 9개 위치가 옳고 1개만 틀림)로서 동일하며, 공백을 과생성하거나 저생성한다는 차이점이 정확도 계산에 전혀 반영되지 않는다.

(3) "한글은 과학적인문자이다".
(4) "한글은 과학적인 문 자이다".

공백 삽입 일치도의 미흡한 점을 보완하기 위하여 새로운 평가 기준으로써 자동 띄어쓰기의 정답 기준을 "모든 문자 사이의 공백 삽입 가능 위치"에 대한 평가 기준이 아니라 "공백이 삽입된 개수와 그 위치"를 기준으로 옳게 생성된 공백의 개수에 의한 재현율과 공백의 과생성(over generation) 개수에 의한 정확도를 계산한다.

재현율=(옳게 생성된 공백 개수)/(생성되어야 할 공백 개수)
정확률=(옳게 생성된 공백 개수)/(시스템이 제시한 공백 개수)

새로운 평가 기준인 공백 재현율과 정확률은 정답 문서에서 공백의 개수를 기준으로 계산된다. 재현율은 해당 공백 위치에 재현된 공백의 개수에 의해 계산하고, 정확률은 생성된 공백의 개수 중에서 정답의 개수로 계산한다. 위 (3), (4)의 예에서는 재현되어야 할 공백 개수가 2이고 (3)은

1개, (4)는 3개의 공백을 생성하였으며 이에 대한 재현율과 정확률은 아래와 같이 계산된다.

(3) : 재현율 0.5, 정확률 1.0
(4) : 재현율 1.0, 정확률 0.67

공백 확률 단순화에 의한 자동 띄어쓰기

1. 바이그램 공백 확률 데이터의 단순화

띄어쓰기 확률 정보를 추출하기 위한 한글 바이그램 음절쌍과 그 빈도수는 기존의 연구에서 1,200만 어절 규모의 말뭉치에서 추출된 결과를 사용하였다(강승식, 2000b). 이 말뭉치에서 추출된 바이그램 개수는 약 29만개이고, 이 중에서 영문자와 숫자를 제외한 한글 음절쌍의 개수는 25만 6천여개이다.[1] 음절 X, Y에 대해 "XY"뿐만 아니라 "X Y" 유형이 포함되고, 문장 부호와 기호는 제외한다. 또한, 바이그램 빈도가 모든 한글 문서에 그대로 적용되는 것은 아니다. 인명, 회사명, 외래어 등 고유명사와 전문 분야 용어에는 기존의 바이그램 특성과 상이한 음절쌍이 출현될 수 있기 때문이다.

기존의 연구에서는 빈도수가 2 이하인 음절쌍을 제외하고, 빈도 3 이상인 15만 6천여개의 음절쌍에 대한 띄어쓰기 정보를 구축하였다. 말뭉치에서 추출된 모든 음절쌍이 현대 한글 문서에서 사용되는 것은 아닐 것으로 추정된다. 그 이유는 말뭉치에는 철자 오류로 인해 실제 문서에서 사용되지 않는 음절쌍이 포함된 경우가 있을 수 있기 때문이다.[2]

기존의 음절 정보는 각 음절쌍에 대해 공백 출현 위치에 따라 좌공백 빈도, 우공백 빈도, 사이 공백 빈도, 그리고 총 출현 횟수에 의해 각 경우에 대한 공백 삽입 확률을 계산한다.

- 좌 공 백 확률 : " XY"의 확률.
- 우 공 백 확률 : "XY "의 확률.
- 사이공백 확률 : "X Y"의 확률.

기존의 연구에서 각 경우에 대한 확률값을 저장할 때 각 확률값을 1바이트 혹은 2바이트로 구현할 때 데이터의 크기는 2배의 차이가 있다. 확률값을 1바이트 영역으로 변환하여 저장할 경우, 15만 6천여개 음절쌍에 대한 바이그램 데이터의 크기는 약 1M 바이트이다.

본 연구에서는 바이그램 음절쌍 데이터의 크기를 최소화하기 위하여 각 음절쌍마다 좌공백, 우공백, 사이 공백 확률을 단순화시켜 1바이트로 조합하여 저장하는 기법을 사용한다. 구체적인 방법으로 좌공백, 우공백, 사이 공백 확률값을 단순화하여 각각 2비트씩을 할당하였다.[3] 1바이트(8비트)를 좌공백과 우공백은 2비트씩 할당하고, 상대적으로 가중치가 높은 사이공백은 4비트를 할당하는 방법을 적용할 수도 있다.

즉, 각 확률값을 0~3까지 2비트로 변환하기 위하여 확률값이 0.2 미만이면 0, 0.2~0.4이면 1, 0.4~0.6이면 2, 0.6 이상이면 3으로 변환하는 방법을 사용한다. 또한, 음절쌍의 개수를 제한하여 음절쌍의 빈도가 3 이상인 경우, 6 이상인 경우, 14 이상인 경우, 37 이상인 경우, 98 이상인 경우로 구분하여 데이터를 구축하였으며, 각 경우에 대한 음절쌍 데이터의 크기는 Table 1과 같다.

2. 자동 띄어쓰기 방법

1) 확률 가중치 기법

기존의 연구에서는 아래와 같이 공백 삽입 여부를 결정하기 위해 3개의 확률값에 가중치를 적용한 합을 구하여 임계치 이상인 경우에 공백을 삽입하는 방법을 사용하였다(강승식, 2000b). 이 식에서 a+b+c=1이고 실험에 의해 a와 c의 값은 0.25, b=0.5이고, 임계치는 0.375이다. 사이 공백의 가중치를 좌공백 확률이나 우공백 확률의 2배로 한 것은 사이 공백 확률의 기여도가 높다고 추정되기 때문이다. 좌공백, 우공백, 사이공백 확률의 기여도는 실험적으로 그 가중치를 결정하여야 하나 그 기준이 모호하기 때문에 경험적으로 가중치를 변경하는 실험을 통하여 결정하였다.[4]

Table 1. 음절쌍 개수에 따른 데이터 크기(KB)

구 분 / 음절쌍 빈도	기존 구현 방법	단순화 기법
3	927	468
6	700	355
14	487	248
37	305	157
98	178	94

1) 음절 X, Y에 대해 "XY"뿐만 아니라 "X Y" 유형이 포함되고, 문장 부호와 기호는 제외한다. 또한, 바이그램 빈도가 모든 한글 문서에 그대로 적용되는 것은 아니다. 인명, 회사명, 외래어 등 고유명사와 전문 분야 용어에는 기존의 바이그램 특성과 상이한 음절쌍이 출현될 수 있기 때문이다.

2) 말뭉치에서 추출된 모든 음절쌍이 현대 한글 문서에서 사용되는 것은 아닐 것으로 추정된다. 그 이유는 말뭉치에는 철자 오류로 인해 실제 문서에서 사용되지 않는 음절쌍이 포함된 경우가 있을 수 있기 때문이다.

3) 1바이트(8비트)를 좌공백과 우공백은 2비트씩 할당하고, 상대적으로 가중치가 높은 사이공백은 4비트를 할당하는 방법을 적용할 수도 있다.

4) 좌공백, 우공백, 사이공백 확률의 기여도는 실험적으로 그 가중치를 결정하여야 하나 그 기준이 모호하기 때문에 경험적으로 가중치를 변경하는 실험을 통하여 결정하였다.

$P(xi, xi+1)=a\times PR(xi-1, xi)+b\times PM(xi, xi+1)+c\times PL(xi+1, xi+2)$

$PR(xi-1, xi)$: <$xi-1, xi$>의 우공백 확률.
$PM(xi, xi+1)$: <$xi, xi+1$>의 사이 공백 확률.
$PL(xi+1, xi+2)$: <$xi+1, xi+2$>의 좌공백 확률.

본 연구에서는 기존의 방법을 확률값을 단순화시킨 바이그램 음절 확률값에 적용한다. 다만, 확률값의 범위가 2비트값인 0~3으로 변환되었기 때문에 이 값을 기준으로 임계치를 설정해야 한다.

2) 투표 방식에 의한 공백 삽입

공백 삽입 여부를 결정하는 방법으로 좌공백, 우공백, 사이 공백 확률에 대해 좌공백과 우공백은 1표씩, 사이 공백은 2표를 부여하여 총 4표에 대한 투표 방식으로 공백 삽입 여부를 결정한다. 투표는 각 변환된 확률값이 3인 경우는 O, 그렇지 않으면 X로 한다. 즉, 변환하기 전의 확률값으로 계산하면 확률값이 0.6 이상인 경우에만 O로 투표하도록 한다. O표가 2 이상인 경우에 공백을 삽입하고, 1 이하이면 공백을 삽입하지 않는다.

실험 및 평가

자동 띄어쓰기 정확도 실험을 위하여 실험 문서로 컴퓨터 분야의 신문기사 400개를 수집하였다. 실험 문서는 126,700 어절로 구성되었으며, 그 크기는 약 1M bytes이다. 총 어절 중에서 문장 끝 위치는 공백이 삽입될 필요가 없으므로 실제로 공백이 자동 띄어쓰기 시스템에 의해 삽입되어야 할 공백의 개수는 121,700개이다.

기존의 연구에서는 띄어쓰기와 붙여쓰기가 모두 허용되는 복합명사의 특성을 고려하여 정답 집합을 입력 문서 자체(비가공된 정답)와 붙여쓴 복합명사를 정답으로 간주하기 위해 입력 문서를 수정하여 만든 '가공된 정답' 두 가지로 구성하였으나, 본 연구에서는 입력 문서 자체를 정답으로 간주하였다.

1. 확률 가중치 기법에 의한 실험

음절 바이그램으로 구성된 2비트 확률값은 0~3이고, 사이 공백은 가중치는 기존 방법과 달리 좌공백, 우공백과 동일한 가중치로 실험하였다.[5] 기존의 방법과 같이 사이 공백의 가중치를 2로 실험하였으나 동일한 가중치를 적용했을 때보다 전반적으로 성능이 약간 낮았으며, 확률값을 단순화했기 때문일 것으로 추정된다.

Table 2. 임계치 5 이상일 때 실험 결과

데이터 선택기준	공백 삽입 일치도	Recall	Precision
기존방법(빈도 3이상)	0.9260	0.8673	0.8535
빈도 3이상	0.9278	0.8623	0.8629
빈도 6이상	0.9270	0.8551	0.8655
빈도14이상	0.9251	0.8431	0.8683
빈도37이상	0.9215	0.8203	0.8734
빈도98이상	0.9154	0.7847	0.8805

Table 3. 임계치 6 이상일 때 실험 결과

데이터 선택기준	공백 삽입 일치도	Recall	Precision
기존방법(빈도 3이상)	0.9260	0.8673	0.8535
빈도 3이상	0.9254	0.8146	0.8924
빈도 6이상	0.9247	0.8087	0.8946
빈도14이상	0.9229	0.7987	0.8966
빈도37이상	0.9191	0.7802	0.8987
빈도98이상	0.9133	0.7513	0.9023

따라서 공백 삽입 여부를 판단하는 세 가지 확률값의 합은 0~9이다. 실험에 의해 임계치를 5와 6으로 설정하여 실험한 결과는 각각 Table 2, 3과 같다. 본 연구에서 제안한 방법과 기존 방법을 비교하기 위하여 기존 방법은 바이그램 음절쌍의 빈도가 3 이상(음절쌍 개수 15만 6천개)인 경우의 정확도를 계산하였다.

임계치 5일 때 기존 방법과 비교하면, '공백 삽입 일치도'의 경우 기존 방법에 비해 +0.18%(=0.9278−0.9260)의 성능이 개선되었으며, 새로운 평가 척도인 정확률은 +0.94%의 성능이 향상되었다. 이에 비해, 재현율은 −0.5% 감소에 그치고 있다. Table 1에서 바이그램 음절쌍 데이터의 크기는 기존 방법에 비해 2분의 1로 작아졌으므로 본 연구에서 제안한 방법이 매우 효율적임을 알 수 있다.

임계치를 6으로 상향 조절하면 공백 재현율은 감소하고 정확률은 높아진다. 이 실험에서는 기존 방법보다 재현율이 감소(−5.27%)하고 '공백 삽입 일치도'가 약간 감소(−0.06%=92.54−92.60)하였으나, 그 대신에 정확률은 3.89% 증가하였다. 임계치 5인 경우와 비교했을 때는 재현율은 −3%~−5% 감소하고 정확률은 2~3% 증가하였다.

따라서 재현율이 중요시되는 응용 분야에서는 임계치를 5로 하고, 정확률이 중요시되는 경우에는 임계치를 6으로 설정하였을 때 기존의 방법보다 성능이 향상될 뿐만 아니라 음절 바이그램의 크기는 2분의 1로 줄일 수 있음을 알 수 있다.

5) 기존의 방법과 같이 사이 공백의 가중치를 2로 실험하였으나 동일한 가중치를 적용했을 때보다 전반적으로 성능이 약간 낮았으며, 확률값을 단순화했기 때문일 것으로 추정된다.

Table 4. 투표 방식에 의한 실험 결과

데이터 선택기준	공백 삽입 일치도	Recall	Precision
기존방법(빈도 3이상)	0.9260	0.8673	0.8535
빈도 3이상	0.9236	0.8103	0.8892
빈도 6이상	0.9230	0.8039	0.8923
빈도14이상	0.9213	0.7937	0.8949
빈도37이상	0.9183	0.7763	0.8991
빈도98이상	0.9132	0.7498	0.9038

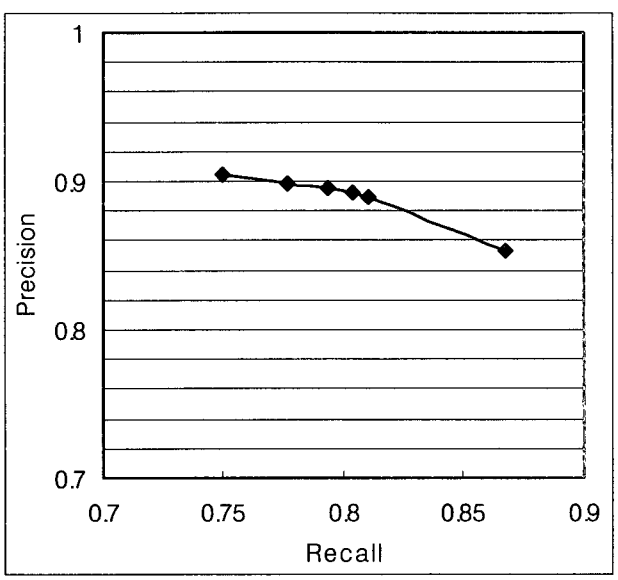

Fig. 1. 투표 방식의 재현율-정확률 실험 결과.

2. 투표 방식에 의한 실험

Table 4는 바이그램 데이터 크기별로 투표 방식에 의한 자동 띄어쓰기 실험을 수행한 결과이고, Fig. 1은 실험 결과에 대한 재현율-정확률 비교 그래프이다. 확률 가중치 방식의 임계치 6인 경우와 비교해 보면, 각 평가 기준별로 약간씩 성능 차이가 있으나 거의 유사함을 알 수 있다. 그러나 투표 방식의 경우 좌공백, 우공백, 사이 공백 확률값을 원래 확률값을 기준으로 0.6 이상인 경우와 아닌 경우로만 구분하였기 때문에 각 음절쌍마다 1비트씩 모두 3비트로 표현된다. 이는 확률 가중치 기법에서 각각 2비트씩 6비트를 사용할 때보다 바이그램 데이터의 크기를 매우 작게 유지하면서 거의 동일한 성능을 보임을 알 수 있다.

결 론

자동 띄어쓰기 시스템의 평가 기준으로 각 문자 경계마다 공백 삽입 여부를 평가하는 방식은 실제 응용 분야에서 어절을 인식하는 정확도를 정확히 반영하지 못하는 문제가 있다. 따라서 정답 문서의 공백 개수에 대하여 해당 공백 위치에 공백을 재현하는 새로운 평가 기준을 제시하였다. 재현율과 정확률로 평가하는 방식은 기존의 공백 삽입 일치도에 의한 정확도가 동일한 시스템이라 하더라도 재현율과 정확률이 다르게 평가된다.

자동 띄어쓰기 시스템은 공백 재현율을 높이는 것보다 공백 생성 정확도를 높이는 것이 더 중요하다. 따라서 재현율이 낮아지더라도 정확률을 높이는 방법으로 확률값을 단순화시켜 바이그램 데이터의 크기를 최소화하면서 정확률을 높이는 방법을 제안하였다. 실험에 의해 제안한 방법이 빈도 3 이상의 바이그램 데이터에서 기존의 방법에 비해 '공백 삽입 일치도'와 공백 생성 정확도를 높일 수 있음을 확인하였다.

또한, '공백 삽입 일치도'는 약간 낮아지더라도 공백 생성 정확률이 매우 중요한 시스템에서는 빈도 98 이상의 바이그램 데이터를 이용하는 방법이 매우 효율적임을 알 수 있었다. 특히, 빈도 98 이상의 바이그램 데이터만을 사용하는 경우에는 기존의 방법에서 빈도 3 이상의 데이터를 사용하는데 비해 데이터의 크기가 10분의 1로 작아지는 효과가 있다.

공백 생성 정확률이 낮은 자동 띄어쓰기 방법론은 성능 개선을 위한 추가 작업이 쉽지 않으나, 정확률이 높고 재현율이 낮은 방법론은 긴 어절로 인식된 문자열에 대해 2차적으로 형태소 분석이나 해당 문자열 내부에서만 임계치를 낮추는 방법에 의해 성능 개선이 쉬워진다. 향후 연구로는 공백이 삽입되지 않음으로 인해 두 어절 이상의 긴 문자열에 대해 띄어쓰기 오류를 교정하는 기법을 연구할 예정이다.

REFERENCES

김계성, 이현수, 이상조(1998) : "연속 음절 문장에 대한 3단계 한국어 띄어쓰기 시스템", 정보과학회 논문지(B), 25권 12호, *pp1838-1844*

강승식(2000) : "한글 문장의 자동 띄어쓰기를 위한 어절블록 양방향 알고리즘", 정보과학회 논문지(B), 27권 4호, *pp441-447*

심광섭(1996) : "음절간 상호 정보를 이용한 한국어 자동 띄어쓰기", 정보과학회 논문지(B), 23권 9호, *pp991-1000*

강승식(2000) : "음절 *Bigram* 특성을 이용한 띄어쓰기 오류의 인식", 제12회 한글 및 한국어 정보처리 학술발표 논문집, *pp85-88*

신중호, 박혁로(1997) : "음절 단위 *bigram* 정보를 이용한 한국어 단어 인식 모델", 한글 및 한국어 정보처리 학술발표 논문집, *pp255-260*