입력 문장의 띄어쓰기를 고려한 음절 바이그램 띄어쓰기 모델

조한철, 이도길, 임해창
고려대학교 자연어처리 연구실
{johanc, dglee, rim}@nlp.korea.ac.kr

Automatic Word Spacer based on Syllable Bi-gram Model using Word Spacing Information of an Input Sentence

Han-Cheol Cho, Do-Gil Lee, Hae-Chang Rim
Natural Language Processing Lab, Korea University
{johanc, dglee, rim}@nlp.korea.ac.kr

요약

현재까지 제안된 자동 띄어쓰기 교정 모델들은 그 중의 대다수가 입력 문장에서 공백을 제거한 후에 교정 작업을 수행한다. 이러한 교정 방식은 입력 문장의 띄어쓰기가 잘 되어 있는 경우에 입력 문장보다 좋지 못한 교정 문장을 생성하는 경우가 있다.
본 논문에서는 이러한 문제점을 해결하기 위하여 입력 문장의 띄어쓰기를 고려한 자동 띄어쓰기 교정 모델을 제안한다. 이 모델은 입력 문장의 음절단위 띄어쓰기 오류가 5%일 때 약 8%의 성능 향상을 보였으며, 10%의 오류가 존재할 때 약 5%의 성능 향상을 보였다.

1. 서론

띄어쓰기 교정이란 입력 문장 내에서 잘못된 띄어쓰기를 수정하는 작업이다. 한글 맞춤법에는 띄어쓰기에 대한 자세한 규정이 있기는 하지만 방대한 양의 규칙과 모호성 때문에 정확한 띄어쓰기를 하는 것은 매우 어렵다. 이러한 이유로 많은 사람들이 글을 쓸 때 띄어쓰기 실수를 하는 경우가 발생한다. 또한 종이 문서를 전자 문서화 할 때 필요 한 OCR 작업 및 두 행에 걸쳐 나누어진 단어의 복원 작업 중 띄어쓰기 오류가 발생할 수 있다.
많은 자연어처리 과정들이 효율적인 처리를 위해서 입력 문장이문법적으로 옳다고 가정하기 때문에 이러한 띄어쓰기 오류는 견고한 자연어처리에 걸림돌이 된다.

지금까지 제안된 대부분의 띄어쓰기 교정 방식들은 입력 문장에서 공백을 모두 제거한 후에 띄어쓰기 교정 작업을 수행하는 특징을 갖는다. 입력 문장의 띄어쓰기 상태를 고려하려면 띄어쓰기 교정 문제에 적합한 특수한 형태의 학습 데이터가 필요하기 때문에 기존의 방식들은 입력 문장 내에 존재하는 음절만을 이용하여 띄어쓰기 교정을 수행한 것이다. 이런 방식을 사용할 경우, 원시 말뭉치 데이터만을 사용하여 띄어쓰기 교정 작업을 할 수가 있다.

하지만 입력 문장의 띄어쓰기를 고려하지 않을 경우에는 입력 문장에서 올바르게 띄어쓰기가 된 부분을 틀리게 고치는 오류가 발생하여, 교정 결과가 입력 문장보다 더 나빠지는 경우가 생길 수 있다. 이러한 오류를 감소시키기 위해서는 입력 문장의 띄어쓰기 상태를 고려해야만 한다.

본 논문에서는 기존에 제안된 띄어쓰기 교정 모델들과 동일한 원시 말뭉치 학습 데이터를 사용하면서도 입력 문장의 띄어쓰기를 고려할 수 있는 모델을 제안한다.

2. 관련 연구

현재까지 띄어쓰기 오류를 교정하기 위해서 많은 방법들이 제안되었다. 이는 크게 규칙 기반 교정 방식과 통계 기반 교정 방식으로 나누어진다.

규칙 기반 접근 방식[1,2]은 어휘 사전, 조사/어미 사전, 띄어쓰기 용례사전 등과 같은 어휘 지식과 최장/최단 일치, 형태소 분석, 띄어쓰기 오류 유형과 같은 휴리스틱을 사용하여 띄어쓰기 교정을 수행한다. 이러한 규칙 기반 접근 방식은 띄어쓰기를 교정한 경우에 그 결과가 매우 정확하다는 장점이 있다. 하지만 규칙에 해당되는 오류만을 교정하기 때문에 통계 기반 접근 방식에 비해 교정 대상이 적으며, 띄어쓰기 교정을 위한 규칙들을 유지, 관리하는데 많은 비용이 드는 단점이 있다.

이에 반해, 통계 기반 접근 방식은 대량의 말뭉치로부터 인접한 두 음절의 띄어 쓸 확률과 붙여 쓸 확률을 계산하고, 이를 이용해 띄어쓰기 교정 작업을 수행한다[3,4,5,6,7]. 이 방식의 장점은 대량의 말뭉치로부터 자동으로 학습을 수행하기 때문에, 어휘 지식이나 띄어쓰기 규칙을 유지, 보수할 필요성이 없다는 것이다. 하지만 학습 결과가 말뭉치에 영향을 크게 받기 때문에 교정 대상이 학습 말뭉치와 유사하지 않은 경우에는 비교적 낮은 성능을 보인다. 또한 통계 기반 접근 방식은 정확도를 향상시키는데 있어서일정한 한계점이 있다고 알려져 있다. 통계 기반 띄어쓰기 교정의 초기 연구로는 음절간 상호 정보를이용한 모델[3]과 음절 바이그램을 이용한 모델[4]이 있다. 위의 두 모델은 주어진 두 음절 사이의 띄어쓰기를 결정할 때 주변 네 음절만을 사용한다. [6]은 기존의 통계 기반 띄어쓰기 교정방법들이 교정된 띄어쓰기 상태를 고려하지 않아 발생하는 문제점들을 해결하고자 확장 문맥 HMM 모델을 이용한 띄어쓰기 교정 모델을 제안하였다. 이 모델은 이전에 교정된 띄어쓰기 상태를 고려함으로써 교정 성능을 향상시켰다.

3. 입력 문장의 띄어쓰기를 고려한 음절 바이그램 띄어쓰기 모델

입력 문장의 띄어쓰기를 고려하려면 띄어쓰기 확률을 추정 과정에 있어서 입력 문장의 띄어쓰기 상태가 반영되어야 한다. 하지만 이러한 방법은 띄어쓰기 오류가 포함된 문장과 그것이 교정된 문장의 쌍으로 된 학습 데이터와 같이 특수한 형태의 학습 데이터를 요구한다. 이러한 형태의 학습 데이터는 현재 존재하지 않기 때문에, 원시 말뭉치 데이터를 사용하면서도 입력 문장의 띄어쓰기 상태를 고려할 수 있는새로운 접근 방법이 필요하다.

본 논문에서 제안하는 모델은 기존에 제안된 모델들과 동일하게 입력 문장에 존재하는 공백을 제거하고, 인접한 두 음절의 띄어 쓸 확률을 구한다. 하지만 이 확률을 입력 문장의 띄어쓰기 상태에 기반하여 재조정함으로써 입력 문장의 띄어쓰기 상태를 반영할 수 있도록 했다. 이 모델은 크게 나누어 세 부분으로 구성된다. 첫 번째 부분은 입력 문장의 띄어쓰기 상태가 얼마나 정확한가를 추정하는 부분이다. 입력 문장의 띄어쓰기가 올바를수록 1에 가까우며, 잘못될수록 0에 가까운 확률을 추정한다. 두 번째 부분은 입력 문장에서 인접한 두 음절 사이의 띄어쓰기 확률을 추정하는 부분이다. 이 확률은 음절 바이그램을 사용하여 추정하였다. 세 번째 부분은 음절 바이그램 방식으로 추정한 띄어쓰기 확률을 입력 문장의 띄어쓰기 상태에 기반하여 재조정하는 부분으로, 이를 통해서 입력 문장의 상태를 반영하는 것이 가능하다. 최종적으로 재조정된 띄어쓰기 확률이 임계치 0.52일 때 최고 성능을 보였다.보다 높을 경우에는 두 음절 사이를 띄어 쓰며, 낮거나 같을 경우는붙여 쓴다.

3.1 절에서는 음절 바이그램을 이용하여 인접한 두 음절 사이의 띄어쓰기 확률을 추정하는 방법을 설명한다. 그리고 3.2 절에서는 입력 문장에 존재하는 띄어쓰기 상태가 얼마나 정확한가를 추정하는 방법을 소개한다. 마지막으로 3.3 절에서는 띄어쓰기 확률을 입력 문장의 띄어쓰기 상태에 기반하여 재조정하는 방법을 설명한다.

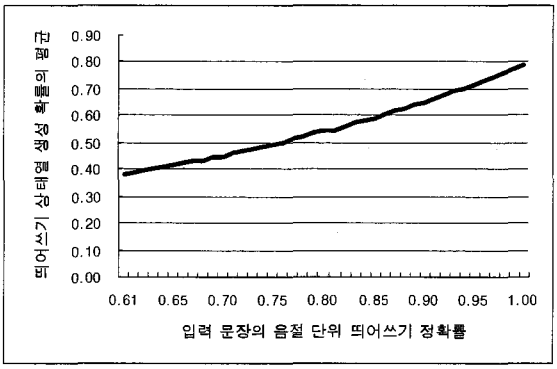
3.1 음절 바이그램을 이용한 띄어쓰기 확률의 추정

입력 문장 S의 연속된 두 음절 S_i와 S_{i+1} 사이를 띄어 쓸 확률은 음절 바이그램을 이용하여 계산한다. 식 (1)에서 P_R, P_M, P_L은 각각 주어진 두 음절의 우측, 중간, 좌측을 띄어 쓸 확률이다.

$$P_{space}(s_i, s_{i+1}) = 0.25P_R(s_{i-1}, s_i) + 0.5P_M(s_i, s_{i+1}) + 0.25P_L(s_{i+1}, s_{i+2}) \quad \text{--- (1)}$$

음절 바이그램을 이용한 띄어쓰기 교정 모델에 대한 자세한 설명은 논문 [4]에 나와있다.

3.2 띄어쓰기 정확률의 추정



<그림 1. 실제 정확률 vs. 추정 정확률>

입력 문장에서 인접한 두 음절 사이의 띄어쓰기가 얼마나 정확한가는 띄어쓰기 정확률로 판단할 수 있다. 하지만 실제 띄어쓰기 정확률은 입력 문장에서 띄어쓰기 오류가 모두 교정된정답 문장이 있어야 계산할 수 있기 때문에 사용하는 것이 불가능하다. 그러나 음절 바이그램 모델[4]의 경우, 교정 결과가 약 97%의 매우 높은 음절 단위 띄어쓰기 정확률을 보이기 때문에 입력 문장과 이 모델의 교정 결과를 이용하여 비교적 정확히 띄어쓰기 정확률을 추정하는 것이 가능하다. 한 가지 문제점은 입력문장이 매우 짧을경우, 교정 문장 에 발생할 수 있는 오류에 의해 추정된 정확률이 매우 급격히 변할 수 있다는 점이다. 이

점을 해결하기 위해서 띄어쓰기 정확률에 비례하는 입력 문장의 띄어쓰기 상태의 생성 확률을 대신 사용하였다. 생성 확률은 입력 문장의 길이가 길어질수록 값이 감소하기 때문에, 문장의 길이로 정규화하여 사용하였다. 이 확률은 식 (2)를 사용하여 계산할 수 있다.

$$P_{mean}(S_{1,n}, t_{1,n-1}) = \sqrt[n]{\prod_{i=0, w(s_i,s_{i+1})=true}^{n-1} P_{space}(s_i, s_{i+1}) \prod_{i=0, w(s_i,s_{i+1})=false}^{n-1} (1 - P_{space}(s_i, s_{i+1}))} \quad \text{--- (2)}$$

s_i = 입력 문장의 i번째 음절

t_i = 입력 문장의 i번째 띄어쓰기 상태 (1 - 띔, 0 - 붙임)

$$w(s_i, s_{i+1}) = \begin{cases} \text{true if } t_i = 1 \\ \text{false if } t_i = 0 \end{cases}$$

식(2)를 이용해 추정한 정확률이 실제 입력 문장의 띄어쓰기 정확률을 잘 반영하는가를 확인하기 위해서 검증 데이터를 이용해 두 값의 관계를 비교해 보았다. 검증 데이터는 띄어쓰기가 완벽한 문장에 0%~39%의 띄어쓰기 오류를랜덤하게 추가하여 만들었다. 랜덤 띄어쓰기 오류를 포함한 데이터를 사용한 가장 큰 이유는 현재 띄어쓰기 오류가 있는 문장과그 오류가 수정된문장으로 구성된 데이터가 존재하지 않기 때문이다. 또한 띄어쓰기 오류는 발생하는 원인이 매우 다양하기 때문에 특정 패턴의 오류들을 포함한 데이터를 사용하는 것 보다 랜덤 띄어쓰기 오류를 추가한 검증 데이터를 사용하는 것이 올바른 선택이라고 판단했다.

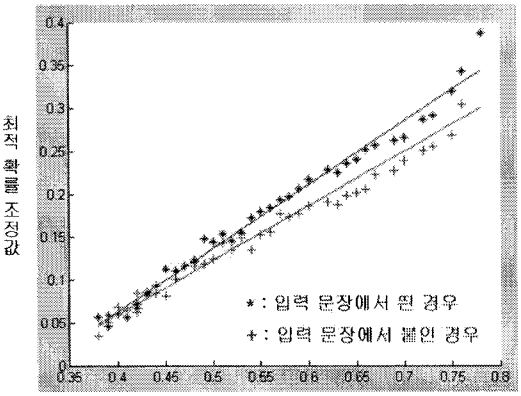
그림 1은 검증 데이터의 실제 음절단위 띄어쓰기 정확률과 식(2)로 추정된 정확률이 갖는 관계를 표시한그래프이다. 이 그래프에서 두 값이 거의 선형적으로 비례하는 것을 볼 수 있다. 이를 통해서 식 (2)를 이용하여 추정한 띄어쓰기 정확률이 입력 문장의 실제 띄어쓰기 정확률을 잘 반영함을 알 수 있다.

본 논문에서는 비슷한 용어로 인한 혼란을 피하기 위하여, 식(2)를 사용하여 추정한 입력 문장의 띄어쓰기 정확률를 입력 문장의 띄어쓰기 신뢰도라고 명명하겠다.

3.3. 띄어쓰기 확률의 조정

입력 문장의 띄어쓰기 상태를 고려하는 방법은 크게 두 가지로 나눌 수 있다. 첫 번째는 특수한 학습 데이터를 사용하여 확률 추정 과정에서 고려하도록 하는 것이며, 두 번째는 입력 문장의 띄어쓰기를 고려하지 않고 구한 확률을 입력 문장의띄어쓰기 상태를 고려하도록 재조정하는 방법이다. 첫 번째 방법은 현재 존재하지 않는학습 데이터를 필요로 하기 때문에 실제적으로 불가능하므로, 본 논문에서는 두 번째 방법을 사용하였다.

음절 바이그램을 이용하여 구한 띄어쓰기 확률이 입력 문장의 띄어쓰기를 반영하도록 하기 위해



<그림 2. 입력 문장의 띄어쓰기 신뢰도와 추정된 최적 확률 조정값 함수>

서는 입력 문장의 띄어쓰기 신뢰도에 기반한 확률 재조정이 필요하다. 만약 입력 문장에서 인접한 두 음절 s_k와 s_{k+1} 가 붙여 써진 경우라면 입력문장의 상태를 반영하기 위해서 띄어쓰기 확률을 감소시켜야 하며, 띄어 써진 경우라면반대로 증가시켜야 한다.

이 때, 띄어쓰기 교정 성능을 최대로 하기 위해서는 입력 문장의 신뢰도에 따른 최적의 확률 증가값(P_a)과 감소값(P_b)을 알아야 한다. 이는 검증 데이터를 이용하여 실험적으로 결정하였다. 우선 음절 단위의 띄어쓰기 오류를 0%~39%까지 추가한 40개의 검증 데이터를 만들고, 검증 데이터마다 띄어쓰기 신뢰도와 가장 높은성능의 확률 조정값을 빔 탐색을 이용하여 계산했다. 그리고 최소제곱오류 방식을 사용한 함수 추정 방법[8]을 이용하여 임의의 띄어쓰기 신뢰도에 대한 최적 확률 조정값을 계산할 수 있는 다항 함수를 추정하였다. n차로 추정된 함수들을 비교해 본결과, 1차 함수로도 충분한 성능을 보였기 때문에 최적 확률 조정을 위한 함수는 1차 함수를 이용하였다.

그림 2는 입력 문장의 띄어쓰기 신뢰도(x축)에 대한 최적 확률 조정값(y축)을 빔 탐색을 이용해 계산하여 나타낸 것이다. 플러스(+) 모양의 점이 입력 문장에서 붙여 쓴 경우에 대한 최적 확률 조정값이며, 스타(*) 모양의 점이 띄어 쓴 경우에 대한 최적 확률 조정값이다. 두 개의 선은 각 경우에 최적 확률 조정값을 생성하는 함수의 그래프이다.

추정된 최적 확률 조정값 생성 함수는식 (3)에 있다. $s_{1,n}$는 입력 문장의 음절열, $t_{1,n-1}$은 입력 문장의 띄어쓰기 상태열, 그리고 $P_{mean}(s_{1,n}, t_{1,n-1})$는 입력 문장의 띄어쓰기 신뢰도를 나타낸다. 그리고 P_a는 입력 문장에 두 음절을 띄어 쓴 경우에 음절 바이그램 모델의 띄어쓰기 확률에 더할 값(reward)이며, P_b는 입력 문장에서 두 음절을붙여 쓴 경우에 빼줄 값(Penalty)이다.

$$\begin{cases} P_a(S) = (0.7423 \times P_{mean}(s_{1,n}, t_{1,n-1})) - 0.2341 \\ P_b(S) = (0.6366 \times P_{mean}(s_{1,n}, t_{1,n-1})) - 0.1953 \end{cases} \quad \text{--- (3)}$$

입력 문장의 띄어쓰기 상태를 고려한 띄어쓰기 확률은 식(4)를 이용하여 계산한다. $P_{new}(x_i, x_{i+1})$는 0~1 사이의 값이며, 함수 $w(x_i, x_{i+1})$는 식 (2)에서 사용한 것과 같다.

$$P_{new}(x_i, x_{i+1}) = \begin{cases} P_{space}(x_i, x_{i+1}) + P_a & \text{if } w(x_i, x_{i+1}) = 1 \\ P_{space}(x_i, x_{i+1}) - P_b & \text{if } w(x_i, x_{i+1}) = 0 \end{cases} \quad \text{--- (4)}$$

4. 실험

이 절에서는 입력 문장의 띄어쓰기 상태를 고려한 띄어쓰기 교정 시스템을 테스트하였다. 학습 데이터는98년과 99년 세종계획 원시 말뭉치[9,10]를 사용하였다. 약 2,600만 어절로 구성된 균형 말뭉치이다. 테스트 데이터는 ETRI 28만 품사 부착 말뭉치[11]에 포함되어 있는 원시 문장에 1%~20% 음절 단위띄어쓰기 오류를 추가하여 사용하였다.

표 1은 테스트 데이터의 어절 단위정확률과 본 논문에서 제안한 모델의 어절 단위 정확률 그리고 음절바이그램 모델[4]의 어절 단위 정확률

Error Rate	Test Corpus	Proposed Model	Bigram Model
1%	0.9484	0.9755	0.8583
5%	0.7646	0.9425	0.8583
10%	0.5858	0.9146	0.8583
15%	0.4454	0.8972	0.8583
20%	0.3370	0.8837	0.8583
공백없음	X	0.8271	0.8583

<표 1. 교정 방식 별 어절 정확률 비교>

$$\text{어절 단위 정확률} = \frac{\text{올바르게 띄어 쓴 어절 수}}{\text{교정된 문장의 어절수}} \quad \text{--- (5)}$$

을 비교한 것이다. 실험 결과에 의하면, 입력 문장의 띄어쓰기 정확률이 높을수록 제안하는 방법의 교정 정확률도 높게 나옴을 알 수 있다. 또한 약 20%의 음절 단위 띄어쓰기 오류를 추가한 테스트 데이터에서도 기존의 모델보다 좋은 성능을 보이고 있다. 이것은 입력 문장의 띄어쓰기 상태를 반영하는 것이 띄어쓰기 교정 성능을 향상시키는데 도움이 된다는 것을 의미한다고 볼 수 있다.

마지막으로 기존의 띄어쓰기 교정 모델들이 사용했던 입력 문장에 띄어쓰기를 모두 제거한 테스트 데이터를 이용하여 실험을 해보았다. 이 경우에는 기존의 모델보다 약 3% 낮은 성능을 보였다. 일반적으로 한글 문장은 띄어 쓴 부분보다 붙여 쓴 부분이 약 3배 정도 많기 때문에 입력 문장에서 띄어 쓴 부분을 모두 없애더라도 약 70% 이상의 신뢰도를 갖게 되어 이러한 결과가발생하게 된다. 이 점을 해결하기 위해서는 이러한 특수한 경우에도 입력 문장의 띄어쓰기 신뢰도를 정확히 측정할 수 있는 방법을 고안할 필요성이 있다.

5. 결론

본 논문에서는 입력 문장의 띄어쓰기 상태를 반영하는 것이 띄어쓰기 교정 성능 향상에 많은 도움이됨을 보였다. 실험 결과에 의하면 입력 문장에 띄어쓰기 오류가 상당한 양이 포함되어 있더라도 기존 모델보다 나은 교정성능을 보였다.

향후에는 입력 문장의 음절들이 모두 붙어있거나 하는 특수한 경우에도 띄어쓰기 신뢰도를 정확히 추정하는 방법을 고안할 필요성이 있으며, 음절 바이그램 모델보다 더 높은 교정 성능을 보이는 HMM 모델[6]을 사용한다면 더 높은 성능을 이끌어내는 것이 가능할 것이라고 본다.

이 논문 또는 저서는2005년 정부(교육인적자원부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2005-041-D00737)

6. 참조 문헌

[1] 최재혁, "양방향 최장일치법을 이용한 한국어 띄어쓰기 자동 교정 시스템", 제9회 한글 및 한국어 정보처리 학술발표 논문집, pp.145-151, 1997.

[2] 강승식, "한글 문장의 자동 띄어쓰기를 위한 어절블록 양방향 알고리즘", 정보과학회 논문지(B), 27권 4호, pp.441-447, 2000.

[3] 심광섭, "음절간 상호 정보를 이용한 한국어 자동 띄어쓰기", 정보과학회 논문지 제23권 9호, pp.991-1000, 1996.

[4] 강승식, "음절 Bi-gram을 이용한 띄어쓰기 오류의 자동 교정", 음성과학회 논문지 제8권 2호, pp.83-90, 2001.

[5] 강승식, "음절 바이그램 단순화 기법에 의한 한국어 자동 띄어쓰기 시스템의 성능 개선", 제15회 한글 및 한국어 정보처리 학술대회, pp.

[6] 이도길, 이상주, 임희석, 임해창, "한글 문장의 자동 띄어쓰기를 위한 두 가지 통계적 모델", 정보과학회논문지 소프트웨어 및 응용 제30권 4호, pp.358-371, 2003.

[7] 박봉래, "대용량 한글 텍스트 데이터베이스 맞춤법 오류 교정 시스템의 구현", 석사학위 논문, 1995.

[8] Fausett, "Applied Numerical Analysis Using MATLAB".

[9] 21세기 세종 계획 국어기초자료 구축, 문확관광부, 1998.

[10] 21세기 세종 계획 국어기초자료 구축, 문화

관광부, 1999.
[11] 한국전자통신 연구원, "품사 부착 말뭉치 구축 지침서", 1999.