한국어 띄어쓰기 모델에서 사용자 입력을 고려한 베이지언 파라미터 추정[1)]

이정훈[0*] 홍금원* 이도길** 임해창*
*고려대학교 컴퓨터학과
**고려대학교 민족문화연구원
*{jhlee[0], gwhong, rim}@nlp.korea.ac.kr, **motdg@korea.ac.kr

Bayesian Parameter Estimation Considering User-input for Korean Word Spacing Model

Jeong-Hoon Lee[0*], Gum-Won Hong*, Do-Gil Lee**, Hae-Chang Rim*
*Dept. of Computer Science and Engineering, Korea University
**Institute of Korean Culture, Korea University

요 약

한국어 띄어쓰기에서 통계적 모델을 사용한 기존의 연구들은 최대우도추정(Maximum Likelihood Estimation)에 기반하고 있다. 그러나 최대우도추정은 자료부족 시 부정확한 결과를 주는 단점이 있다. 본 연구는 이에 대한 대안으로 사용자 입력을 고려하는 베이지언 파라미터 추정(Bayesian parameter estimation)을 제안한다. 기존 연구가 사용자 입력을 교정 대상으로만 간주한 것에 비해, 제안 방법은 사용자 입력을 교정 대상이면서 동시에 학습의 대상으로 해석한다. 제안하는 방법에서 사용자 입력은 학습 말뭉치의 자료부족에서 유발되는 부정확한 파라미터 추정(parameter estimation)을 방지하는 역할을 수행하고, 학습 말뭉치는 사용자 입력의 불확실성을 보완하는 역할을 수행한다. 실험을 통해 문어체 말뭉치, 통신환경 구어체 말뭉치, 웹 게시판 등 다양한 종류의 말뭉치와 다양한 통계적 모델에 대해 제안 방법이 효과적임을 알 수 있다.

1. 서론

한국어 자동 띄어쓰기는 띄어쓰기 오류가 포함된 사용자 입력 문장을 입력받아 띄어쓰기가 올바르게 수행된 문장을 출력하는 문제로 정의할 수 있다. 이러한 한국어 자동 띄어쓰기는 한국어 문장의 구성단위가 되는 어절의 경계 인식을 목표로 하며, 거의 모든 한국어 정보 처리에 전처리로서 수행되는 중요한 작업이다. 올바른 어절의 인식은 독자의 가독성 향상에 도움을 줄 뿐 아니라, 문장의 올바른 의미를 전달하는 데 중요하다. 이러한 한국어 자동 띄어쓰기의 예를 보자.

사용자 입력 문장 : “급히 드릴말씀이 있습니다.”
띄어쓰기 교정 문장 : “급히 드릴 말씀이 있습니다.”

첫번째 문장은 한국어 문장 작성 시 사용자가 야기하기 쉬운 띄어쓰기 오류를 포함하고 있다. 이에 대해 띄어쓰기 오류가 교정된 문장은 두 번째 문장이다.

이러한 한국어 띄어쓰기 문제를 해결하기 위해 제안된 기존 연구들[5, 6]은 음절 단위 정확률로 90% 이상의 성능을 보이는 통계적 모델을 발표하고 있다. 기존 연구들은 새로이 제안하는 모델의 우수성을 입증하기 위해 대량의 문어체 말뭉치를 학습과 테스트 말뭉치로 나눈 후에 성능 테스트를 한다. 이러한 실험은 모델간 비교를 위해선 유효한 실험이라 할 수 있지만, 다양한 종류의 말뭉치에 대해 적용하는 경우 발표된 성능보다 대부분 떨어지는 성능을 보이게 된다. 본 연구에서 수행한 실험에 의하면 90% 이상의 음절 단위 정확률을 보이는 모델이라도 전혀 다른 종류의 말뭉치에 대해 테스트 하는 경우엔 동종 말뭉치에 대해 실험한 경우와 비교해 공통적으로 성능이 하락하는 현상을 보이고 있다(6.2절 <그림 1> 참조). 우리는 이러한 현상이 모델 자체의 문제라기보다는 자료부족이 주요 원인이 되는 파라미터 추정의 부정확성에 기인한다고 간주한다.

따라서 본 연구에서는 부정확한 파라미터 추정에 의한 띄어쓰기 모델의 성능 저하를 완화하기 위해 사용자 입력 문장을 모델의 파라미터 학습 단계에 반영하는 방법을 제안한다. 제안하는 방법에서 사용자 입력 문장을 파라미터 추정 단계에 병합하기 위한 이론적 근거로 베이지언 파라미터 추정을 도입한다. 베이지언 통계학 관점에서 사용자 입력 문장을 파라미터의 사전 분포(prior)로 해석하고, 학습 말뭉치에서 획득한 우도함수(likelihood)와 결합하여 더 나은 사후 분포(posterior)를 획득하게 된다.

기존의 띄어쓰기 모델이 사용자 입력을 교정의 대상으

1) 이 논문은 2008년도 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2006-000-11162-0).

로만 간주한 것에 비해 본 연구는 교정의 대상임과 동시에 학습의 대상으로 간주한다는 점이 큰 특징이라 할 수 있다. 또한, 제안 방법은 특정 띄어쓰기 모델을 위한 방법이 아니라 최대 우도 추정을 사용하는 띄어쓰기 모델에 대해 적용 가능한 파라미터 추정 방법이다.

2. 한국어 자동 띄어쓰기 문제와 사용자 띄어쓰기 정보

한국어 자동 띄어쓰기는 띄어쓰기 오류가 포함된 사용자 입력 문장($s_{1,n}, t_{1,n-1}^{input}$)을 입력 받아 띄어쓰기 오류가 교정된 문장($s_{1,n}, t_{1,n-1}^{corrected}$)으로 자동 변환하는 것이다.

- 사용자 입력 문장 ($s_{1,n}, t_{1,n-1}^{input}$)
  = "$s_1, t_1^{input}, s_2, t_2^{input}, \ldots, t_{n-1}^{input}, s_n$"
- 띄어쓰기 교정 문장 ($s_{1,n}, t_{1,n-1}^{corrected}$)
  = "$s_1, t_1^{corrected}, s_2, t_2^{corrected}, \ldots, t_{n-1}^{corrected}, s_n$"

s_i는 i번째 음절, n은 문장 내 음절의 수, t_i는 s_i와 s_{i+1} 사이의 띄어쓰기 정보를 의미한다[5]. 단 t_i는 붙임을 의미하는 0이나 띄움을 의미하는 1 중 하나의 값을 취하는 이진 변수이다. t^{input}는 사용자 띄어쓰기 정보, $t^{corrected}$는 띄어쓰기 교정 모델이 출력한 띄어쓰기 결정열을 각각 의미한다.

주목할 점은 t^{input}으로 표현한 사용자가 입력한 띄어쓰기에 관한 정보이다. 사용자 띄어쓰기 정보(t^{input})는 주어진 음절열($s_{1,n}$)에 대해 사용자가 직접 입력한 띄어쓰기 태그를 의미한다. 이러한 사용자 띄어쓰기 정보(t^{input})는 다소 주관적이고, 그 정확률은 사용자의 띄어쓰기에 관한 지식 및 문장 입력 환경에 의해 변하는 다소 가변적인 속성을 가진다. 예를 들어 띄어쓰기 교육을 충분히 받지 못한 사용자가 작성한 경우, 사용자의 습관적인 오류 혹은 입력 환경의 제약 등 다양한 원인에 의해 사용자 띄어쓰기 정보는 오류를 가지게 된다. 그러나 이러한 오류에도 불구하고 사용자 띄어쓰기 정보(t^{input})는 음절 단위 정확률을 기준으로 측정할 경우 그다지 낮은 정확도가 아님을 6.1절의 <표 2>를 통해 알 수 있다.

3. 기존 연구의 고찰

띄어쓰기에 관한 연구는 규칙기반, 통계기반, 하이브리드 등 다양한 기법을 적용하여 연구가 이루어졌으나, 최근의 자동 띄어쓰기 연구는 학습 말뭉치로부터 통계정보를 추출하고 이에 대한 통계적 모델을 구성하는 연구[5, 6]가 주로 이루어지고 있다.

[6]은 음절 통계정보를 이용하여 각 음절간 띄어쓰기 여부를 결정하는 이진 분류 모델(binary classification model)을 제안하였고, [5]는 음절열을 관측 과정(observation process)으로 각 음절의 띄어쓰기 여부를 은닉 과정(hidden process)으로 간주하는 은닉 마르코프 모형(hidden Markov model) 기반 방식을 제안하였다.

이와 같이 기존 연구들은 띄어쓰기 문제에 적합한 모델의 구조를 파악하기 위해 집중하였으나, 사용자 입력에 대한 해석과 파라미터 추정은 모두 다음과 같은 전략을 취하고 있다.

기존 연구들은 사용자 입력 문장($s_{i,n}, t_{1,n-1}^{input}$)에서 사용자 띄어쓰기 정보($t_{1,n-1}^{input}$)를 제거하고, 음절 정보($s_{1,n}$)만을 띄어쓰기 모델의 입력으로 간주한다. 그 다음 단계에서 마르코프 가정(Markov assumption)을 적용하여 서로 독립인 조건부 확률 $p(t_{i,j}|context(t_{i,j}), \widehat{\theta_{i,j}})$의 집합으로 모델을 구성한다.

$context(t_{i,j})$는 각 띄어쓰기 t_i에 대응하는 조건부 문맥을 의미하며 각 모델에 따라 조건부 문맥의 구성은 달라진다. 각 조건부 확률 $p(t_{i,j}|context(t_{i,j}))$은 학습 말뭉치 D로부터 최대 우도 추정(maximum likelihood estimation)과 평탄화(smoothing) 기법을 이용해 획득한 $\widehat{\theta_{i,j}}$에 근거하여 값이 결정된다. 여기에 사용되는 학습 말뭉치는 띄어쓰기가 올바른 문장만으로 구성된 것이다.

4. 최대 우도 추정과 베이지언 파라미터 추정의 비교

[1]에 근거하여 베이지언 파라미터 추정과 최대 우도 추정을 비교하면 아래와 같이 설명할 수 있다.

관찰된 샘플의 집합 E가 주어진 경우, 사건 z의 출현 확률의 계산은 각각 다음과 같이 해석된다.

- 최대 우도 추정

$$p(z|\hat{\theta}),\ \hat{\theta} = \arg max_{\theta} p(E|\theta) \quad (1)$$

관찰된 샘플의 집합 E가 주어진 경우, 샘플과 파라미터 θ간의 우도 함수 $p(E|\theta)$를 구한다. 다음 단계는 우도 함수가 최대가 되는 $\hat{\theta}$를 구한다. 마지막으로 $\hat{\theta}$를 이용하여 알고자 하는 z의 출현확률을 예측한다.

- 베이지언 파라미터 추정

$$p(z|E) = \int_{\theta} p(z|\theta)p(\theta|E),\ p(\theta|E) = \frac{p(E|\theta)p(\theta)}{\int_{\theta} p(E|\theta)p(\theta)} \quad (2)$$

관찰된 샘플의 집합 E가 주어진 경우, θ의 사후 분포 $p(\theta|E)$를 구한 후에 θ 공간에 대해 적분하여 z의 출현 확률을 예측한다.

[1]에 의하면 최대 우도 추정은 계산이 간단하고 자료가 풍부한 경우 좋은 파라미터 추정을 수행하지만, 자료 부족 시 충분히 근거 있는 파라미터 추정을 수행하지 못하는 문제가 있다. 이에 대한 대안으로 많이 사용되는 방법 중 하나가 베이지언 파라미터 추정이다. 또한 베이지언 파라미터 추정은 사전 분포 $p(\theta)$를 통해 주관적 믿음이나 통계적 가설 등을 사후 분포 $p(\theta|E)$의 구축 단계에 병합하는 자연스러운 이론을 제공한다.

5. 제안 방법

5.1. 사용자 입력을 고려한 베이지언 파라미터 추정

기존 모델들이 각 조건부 확률 $p(t_{i,j}|context(t_{i,j}))$의 추정에 학습 말뭉치 D만을 사용하여 최대 우도 추정을 사용한 것에 비해, 본 연구는 사용자 입력 $userinput$을 파라미터의 사전 분포로 해석하는 베이지언 파라미터 추정을 제안한다. 이 과정은 베이즈 통계학에 기반하여 각 조건부 확률의 파라미터 $\theta_{i,j}$의 사후 분포를 구성하는 단계에 사용자 입력 $userinput$과 학습 말뭉치 D를 함께 병합한다. 사용자 입력은 이 단계에서 파라미터의 사전 분포의 역할을 한다. 즉 사용자 입력은 각 조건부 확률에 대해 학습 말뭉치로부터 학습하기 전에 주어지는 초기 확률 분포가 된다.

이와 같이 사용자 입력과 학습 말뭉치의 정보를 함께 반영하여 얻은 파라미터의 사후 분포를 이용한 각 조건부확률의 베이지언 파라미터 추정을 수식화하면 다음과 같다.

$$p(t_{i,j}|context(t_{i,j}),E)$$
$$= p(t_{i,j}|context(t_{i,j}),D,userinput) \quad (3)$$
$$= \int_{\theta_{i,j}} p(t_{i,j}|context(t_{i,j}),\theta_{i,j})p(\theta_{i,j}|D,userinput) \quad (4)$$

수식 (3)은 관찰된 샘플의 집합 E가 주어진 경우에 대해 $p(t_{i,j}|context(t_i))$의 출현 확률을 계산함을 의미한다. E는 서로 독립적인 학습 말뭉치 D와 사용자 입력 문장 $userinput$으로 구성된다. 수식 (4)의 $\theta_{i,j}$는 통계적 학습의 대상인 학습말뭉치 D와 사용자 입력 문장 $userinput$이 따르는 분포의 파라미터이며, 이에 근거하여 $p(t_{i,j}|context(t_{i,j}))$의 값이 결정된다. 따라서 주어진 입력에 대하여 올바른 $\theta_{i,j}$를 결정하는 것이 띄어쓰기 모델의 성능에 중요한 영향을 미친다. 일반적인 베이지언 파라미터 추정과 다른 점은 사용자 입력 문장 $userinput$을 학습의 대상으로 포함한다는 점이다.

[1]에 의하면 학습 말뭉치 이외에 사용되는 이러한 부가적인 정보는 일정 수준 이상의 신뢰성을 가지면 성능 향상에 도움이 된다. 6.1절의 <표 2>에 의하면 한국어 띄어쓰기 환경에서 사용자 입력은 휴대폰을 통해 입력한 경우에 77% 이상의 음절 단위 정확률을 보이고 있으며, 웹에서 수집한 문서의 경우에는 80~90% 이상의 정확률을 보이고 있다. 따라서 사용자 입력이 사전 분포로서 충분히 좋은 역할을 하리라 기대할 수 있다. 사전 분포로서 사용자 입력은 학습 말뭉치의 자료부족을 보완하는 추가 정보원으로서 역할을 한다. 물론 학습 말뭉치에 자료가 충분한 경우에는 학습 말뭉치의 정보를 강하게 반영하는 것이 유리할 것이다.

파라미터의 사후 분포인 $p(\theta_{i,j}|D,userinput)$는 사용자 입력문장 $userinput$과 학습말뭉치 D를 각각 순차적으로 관찰하면서 $\theta_{i,j}$를 갱신하여 구해진다. 이 과정에서 사용자 입력 문장이 해당 문장의 띄어쓰기에 대해 사용자가 제시한 샘플로 취급되어 $\theta_{i,j}$의 사후 분포 구축에 기여하게 된다. 이 과정은 [2]에 근거하였다.

- $p(\theta_{i,j})$: $\theta_{i,j}$에 대해 관찰된 샘플의 집합을 관찰하기 전에 알고 있는 사전 분포
- $p(\theta_{i,j}|userinput) \propto p(\theta_{i,j})p(userinput|\theta_{i,j})$: 사용자 입력 문장 $userinput$을 관찰한 후에 획득한 $\theta_{i,j}^t$의 사후 분포
- $p(\theta_{i,j}|userinput,D) \propto p(\theta_{i,j})p(userinput|\theta_{i,j})p(D|\theta_{i,j})$: 사용자 입력 문장 $userinput$과 학습 말뭉치 D를 함께 관찰한 후에 획득한 $\theta_{i,j}$의 사후 분포

즉 $p(\theta_{i,j}|userinput,D)$는 사용자 입력 $userinput$과 학습 말뭉치 D를 고려하여 얻은 $\theta_{i,j}$의 사후 분포이다.

이 과정을 언어모형 단계로 구체화하면 다음과 같다. 각 조건부확률을 획득할 때, 학습 말뭉치 D는 다항분포(Multinomial distribution)를 따르는 샘플의 집합으로 해석된다. 예를 들어 $p(t_{i=0,1}|\text{가나},D)$는 문맥 "가나" 사이에 붙일 확률과 띄울 확률을 의미한다. 이 두 확률의 합은 1이 되어야 하고, 각 조건부 확률은 말뭉치에서 발생한 빈도 정보에 의해 결정된다. 또한 서로 다른 문맥에 대해 독립적으로 확률값 추정을 하게 된다.

즉 $p(\theta_{i,j}|D)$은 다항분포의 집합으로 구성된다[3]. $p(\theta_{i,j}|D)$는 학습 말뭉치에서 관찰하여 얻은 확률 분포로서 우도함수가 되고, 이에 대한 conjugate prior[2]로 Dirichlet distribution을 취한다. 따라서 $p(\theta_{i,j})$과 $p(\theta_{i,j}|userinput)$는 Dirichlet distribution의 형태를 가정한다. 사용자 입력 문장에서 관찰하여 얻은 $p(\theta_{i,j}|userinput)$을 Dirichlet distribution의 관점에서 해석하면, 각 띄어쓰기 사건에 대해 가상의 출현빈도수만큼 믿는다고 표현된다. 이는 각 띄어쓰기 사건이 학습 말뭉치에서 출현한 빈도수와 대응되는 개념이다. 이는 학습 말뭉치 D에서 관찰한 사건에 대해서는 출현 빈도 1씩을 부여하지만, 사용자 입력 문장 $userinput$에서 관찰한 사건에 대해서는 이를 모두 신뢰하지 않음을 의미한다.

$$p(\theta_{i,j}|D,userinput)$$
$$= Dir(\theta_{i,j};\alpha_{i,j}) \cdot Dir(\theta_{i,j};\beta_{i,j}) \cdot Multi(\theta_{i,j};m_{i,j}) \quad (5)$$
$$= \frac{1}{c_1}(\theta_{i,j})^{\alpha_{i,j}-1} \cdot \frac{1}{c_2}(\theta_{i,j})^{\beta_{i,j}-1} \cdot \frac{1}{c_3}(\theta_{i,j})^{m_{i,j}} \quad (6)$$
$$= \frac{1}{c}(\theta_{i,j})^{\alpha_{i,j}+\beta_{i,j}+m_{i,j}-2}, \text{ 단 } \sum_B \theta_{i,j=B} = 1 \quad (7)$$

수식(5~7)의 유도는 [3]을 참조하여 유도했다. $c1$, $c2$, $c3$, c는 확률 1을 만들기 위한 상수이다. $m_{i,j}$는

2) conjugate prior란 베이즈 통계학에서 사후 분포 함수의 형태를 계산하기 용이한 형태로 만들기 위해 선택하는 사전 분포이다.

학습말뭉치 D로부터 관찰한 각 사건의 출현 빈도수이고, $\alpha_{i,j}$는 사전 분포의 가상의 출현빈도수이다. $\beta_{i,j}$는 각 사건을 사용자 입력 문장에서 관찰한 경우에 이에 대해 부여하는 신뢰도 표현하는 가상의 출현빈도수이다. 다항분포에서 $\theta_{i,j}$는 해당 사건이 발생할 확률을 의미하고, $m_{i,j}$는 해당 사건이 학습말뭉치 D에서 출현한 빈도수를 의미한다. 단 띄어쓰기의 경우 $\theta_{i,j=0}+\theta_{i,j=1}=1$이 된다.

지금까지 파라미터의 사후 분포를 구축했다. 이 단계에서 사용자 입력 문장과 학습 말뭉치의 정보가 각각 출현 빈도수 $m_{i,j}$와 가상의 출현 빈도수 $\beta_{i,j}$[3] 형태로 사후 분포 구축에 포함된다. 다음 단계는 파라미터의 모든 공간에 대해 적분을 취하는 것이다. 이에 대한 수식 (8)는 [3]을 참조하여 약간의 변형을 통해 얻었다.

$$\int_{\theta_{i,j}} p(t_{i,j}|context(t_{i,j}),\theta_{i,j})p(\theta_{i,j}|D,userinput) = \frac{m_{i,j}+\alpha_{i,j}+\beta_{i,j}}{\sum_j m_{i,j}+\sum_j \alpha_{i,j}+\sum_j \beta_{i,j}} \quad (8)$$

i는 각 음절의 index를 의미하고, $j=0$은 붙여 쓴 사건, $j=1$은 띄어 쓴 사건을 의미한다. $m_{i,j}$는 학습 말뭉치에서 각 조건부 확률에 대응하는 사건의 출현 빈도를 계산하여 쉽게 구할 수 있다. $\alpha_{i,j}$는 사전 분포로부터 나온 hyper parameter인데, [3]에서는 $\alpha_{i,j}$에 대한 최적화된 값을 추론하여 평탄화 효과를 얻는다. 그러나 본 연구는 평탄화보다는 사용자 띄어쓰기 정보의 활용성에 관한 연구이므로 $\alpha_{i,j}$ 값을 모두 0으로 설정하여 배제한다. 그 대신 기존의 띄어쓰기 모델의 평탄화 기법을 그대로 사용한다.

마지막 문제는 $\beta_{i,j}$의 값 결정이다. $\beta_{i,j}$는 사용자 입력 문장에서 관찰된 사건에 대해 부여하는 가상의 발생 빈도수이다. 만약 사용자 입력이 항상 올바르다면 $\beta_{i,j}$를 크게 할 수록 유리하고, 사용자 입력이 항상 틀리다면 $\beta_{i,j}$를 모두 0으로 설정하는 것이 바람직할 것이다.

수식 (8)에서 만약 $\sum_j m_{i,j}$이 크다면 이것이 사후 분포의 전체 성향을 지배하게 되고, 반대의 경우에는 $\sum_j \beta_{i,j}$가 상대적으로 큰 영향을 가질 것이다.

이와 같은 $\beta_{i,j}$값의 결정은 사용자 입력의 정확률에 의해 결정됨이 합리적이다. 5.2절에서 $\beta_{i,j}$의 값 결정에 대해 본 연구에서 제안하는 3가지 방법을 소개한다.

5.2. hyper parameter $\beta_{i,j}$의 결정

본 연구에서는 $\beta_{i,j}$ 값 선정을 위해 아래의 3가지 방법을 제안한다.

방법 1) 사용자 입력 오류에 대해 공평하게 처리

- $\beta_{i,j=0} = \beta_{i,j=1} = k_0$

방법 2) 사용자 입력 오류를 띈 경우와 붙인 경우를 분리하여 접근

- $\begin{cases}\beta_{i,j=0}=k_1\\ \beta_{i,j=1}=k_2\end{cases}$

방법 3) 사용자 입력의 오류를 문맥별로 모델링하여 접근

- $\begin{cases}\beta_i=k_3, & if\, p(t_i^{input}=t_i^{correct}|context(t_i))>threshold\\ \beta_i=0, & otherwise\end{cases}$

첫 번째 방법은 $\beta_{i,j}$의 값을 i와 j에 무관하게 일정하게 값을 부여하는 것이다. 이 의미는 사용자 입력에서 붙이거나 띈 경우에 대해 동등한 가중치를 부여함을 의미한다. 단 사용자 입력의 정확률에 대해 알지 못하기 때문에 1보다 작게 설정한다.

두 번째 방법은 사용자 입력에서 붙여 쓴 경우($\beta_{i,j=0}$)와 띄어 쓴 경우($\beta_{i,j=1}$)에 다른 가중치를 부여하는 것이다. 이는 한국어 띄어쓰기에서 사용자 입력의 띄어쓰기 오류가 대부분은 띄붙 오류가 붙띄 오류보다 훨씬 많다는 사전지식을 반영한 전략이다. <표 2>에 사용자 띄어쓰기 정보의 띄붙 오류와 붙띄 오류의 비율을 보면, 3가지 테스트 말뭉치 모두 띄붙 오류의 비율이 붙띄 오류에 비해 현저히 높음을 알 수 있다. 이는 사용자가 띈 경우에는 띄어쓰기 오류가 상대적으로 작다는 의미가 된다.

마지막 방법은 사용자 입력의 정확률을 예측하는 모델을 이용하는 것이다.

$$\begin{aligned}&p(t_i^{input}=t_i^{correct}|context(t_i))\\ &\quad=0.25\cdot p(t_i^{input}=t_i^{correct}|s_{i-1},s_i)+\\ &\quad\quad 0.5\cdot p(t_i^{input}=t_i^{correct}|s_i,s_{i+1})+\\ &\quad\quad 0.25\cdot p(t_i^{input}=t_i^{correct}|s_{i+1},s_{i+2})\end{aligned} \quad (9)$$

즉 t_i에 대한 조건부 문맥 $context(t_i)$ 별로 사용자 입력이 올바를 확률을 예측하고 임계치 이상이라면 사용자 입력에 신뢰도를 부여하는 방법이다. 본 연구에서는 [6]에서 제안한 띄어쓰기 모델을 응용하여 사용자 입력 정확률 예측 모델 수식(9)을 구현하였다. 이 방법은 사용자 입력의 띄어쓰기 오류 발생 확률은 문맥별로 크게 다를 것이라는 가정을 반영한다. 예를 들어 "할 수 있다"라는 문자열은 "나는 집에"에 비해 띄어쓰기 오류 확률이 높다.

6. 실험

6.1. 실험 환경 설정

본 연구를 위해 준비된 말뭉치는 sej(세종 말뭉치), cel(휴대폰 입력 말뭉치), web(커뮤니티 웹 게시판), gam(게임 웹 게시판) 4가지이다. sej는 품사부착된 세종

3) [3]에 따르면 $m_{i,j}$, $\alpha_{i,j}$, $\beta_{i,j}$는 파라미터 $\theta_{i,j}$를 지배하는 파라미터라는 의미에서 hyper-parameter로 불린다.

말뭉치를 띄어쓰기 정답 말뭉치로 재구성한 것으로 사용자 띄어쓰기 정보(t^{input})가 존재하지 않기에 띄어쓰기 모델의 학습에만 사용한다. cel은 통신환경 구어체 말뭉치로 휴대폰 기기를 사용하여 사용자가 입력한 말뭉치로 학습(9만)과 테스트(1천)로 나누어 사용한다. 학습 말뭉치는 "띄어쓰기가 교정된 문장"으로만 구성된다. 테스트 말뭉치는 {사용자 입력 문장, 띄어쓰기 교정 문장}으로 구성되며, cel, web, gam 3가지를 사용한다. web과 gam은 웹에서 수집한 게시판 문서이며 크기가 작아 테스트용으로만 사용한다. cel의 사용자 입력의 정확률은 77%이다. 이는 휴대폰을 통한 문자입력의 제약 조건 때문에 사용자가 대부분 붙여 쓰기하여 입력한 말뭉치이다. 모두 붙여 쓴 경우의 음절 단위 정확률이 약 70%임을 감안하면 띄어쓰기 오류가 많다고 할 수 있다. web과 gam은 모두 웹에서 수집한 게임 관련 게시글로서 오류가 많은 편이지만 cel처럼 입력기의 제약이 없기 때문에 cel에 비해 띄어쓰기 정확률이 우수하다고 할 수 있다. 지금까지 설명한 말뭉치 설정은 <표 1>과 <표 2>에 요약되어 있다.

<표 1> 학습 말뭉치

말뭉치 종류	sej	cel
문장 개수	9만	9만
어절 개수	113만	29만

<표 2> 테스트 말뭉치

말뭉치 종류	cel	web	gam
문장 개수	1천	1천	6백
$\mathrm{P}recision_{char}$	77%	89%	88%
띄붙 오류 비율	98.8%	97.0%	86.5%

띄붙 오류 비율= 띄붙 오류/(띄붙 오류+붙띄 오류)

모델의 성능 평가는 아래와 같이 음절 단위 정확률과 음절 단위 정확률의 성능변화율을 사용한다.

- 음절 단위 정확률($Precision_{char}$)
  $= \frac{\text{올바르게 띄어쓴 음절 개수}}{\text{총 음절 개수}} \times 100$

- 성능변화율 $= \frac{Precision_{char}^{proposed} - Precision_{char}^{base}}{Precision_{char}^{base}} \times 100$

$Precision_{char}^{proposed}$는 제안 방법을 적용한 경우의 음절 단위 정확률을, $Precision_{char}^{base}$는 제안 방법을 사용하지 않은 경우의 음절 단위 정확률을 각각 의미한다.
지금까지 설명한 실험 설정을 아래와 같이 정리하였다.

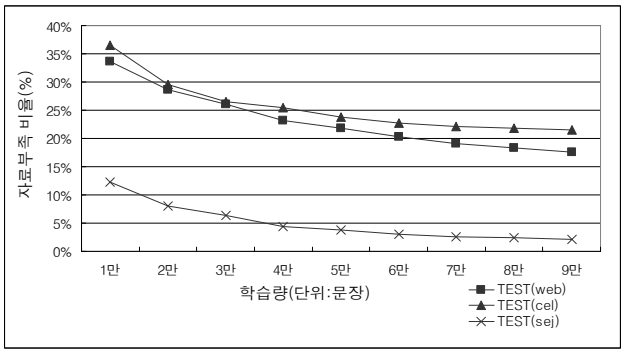
- $TEST$(말뭉치종류)
- $TRAIN$(말뭉치종류,학습량)
- $BCM(method)$
- $HMM(method)$

$TEST()$는 테스트 말뭉치에 대한 설정, $TRAIN()$은 학습 말뭉치에 대한 설정을 의미한다. BCM은 [6]의 모델을 의미하고, HMM는 [5]의 모델을 의미한다.
$method0$은 사용자 입력 문장의 정보를 전혀 보지 않은 기존 방법을 의미하며, [5]와 [6]을 그대로 재구현한 원 모델을 의미한다. $method1$은 첫 번째 β값 선정 방법인 사용자 입력 오류에 대해 공평하게 처리하는 방법을 $method2$는 띈 경우와 붙인 경우를 차등하는 경우, $method3$은 문맥별 사용자 입력 오류 모델을 사용한 경우를 의미한다.

6.2. 학습/테스트 말뭉치 종류의 이질성과 자료부족

첫번째 실험에서 $BCM(method0)$을 적용하는 경우, 학습과 테스트의 말뭉치 종류가 다른 경우와 학습량의 변화에 대한 자료부족 비율을 관찰한다. 자료부족 비율은 $BCM(method0)$의 조건부 확률을 계산하는 모든 항(term)에 대해 학습 말뭉치에서 관찰되지 않은 자료의 비율을 계산한다. $BCM(method0)$으로 실험한 이유는 $HMM(metohd0)$에 비해 단순한 평탄화 기법을 사용하여 직관적이기 때문이며, 같은 조건의 실험을 $HMM(metohd0)$에 수행하더라도 유사한 결과가 나온다.

<그림 1> 학습/테스트 말뭉치의 이질성과 자료부족



<그림 1>은 $BCM(method0)$에 대해 학습 말뭉치 sej를 1만, 2만, ..., 9만으로 각각 학습하고 테스트를 web, cel, sej에서 각각 수행한 경우에 측정한 자료부족비율을 표현하고 있다. 이를 통해 알 수 있는 점은 같은 양의 학습 말뭉치를 사용하더라도 테스트 말뭉치와 학습 말뭉치의 종류가 다른 경우에 자료부족의 비율에 큰 차이가 발생한다는 점이다. 이러한 자료부족은 띄어쓰기 모델의 성능 하락을 야기하게 된다.

6.3. 제안 방법에 의한 성능 향상

두 번째 실험에서 <표 1>의 학습 말뭉치(sej, cel)와 <표 2>의 테스트 말뭉치(cel, web, gam), BCM과 HMM 두가지 모델의 기존 방식과 제안 방법의 3가지 방법의 모든 조합에 대해 비교하였다. $method0$은 제안 방법을 사용하지 않은 방법(기저 성능)을 의미한다. $method1$, 2,

<표 3> 제안 방법에 의한 성능 향상

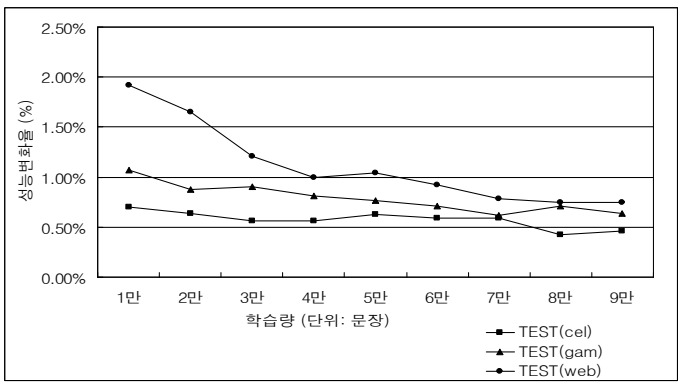
모델	학습 말뭉치	테스트 말뭉치	$method0(\%)$	$method1(\%)$	$method2(\%)$	$method3(\%)$
HMM	sej, 90000	cel	86.31	85.97 (-0.39)	86.71 (+0.46)	86.38 (+0.08)
		gam	89.57	90.52 (+1.06)	90.15 (+0.64)	90.00 (+0.48)
		web	88.49	90.42 (+2.18)	89.15 (+0.75)	89.05 (-0.12)
	cel, 90000	cel	93.99	94.08 (+0.09)	94.10 (+0.12)	93.91 (-0.09)
		gam	86.43	89.66 (+3.75)	87.70 (+1.47)	87.37 (+1.09)
		web	84.86	88.75 (+4.59)	86.56 (+2.01)	85.75 (-0.94)
BCM	sej, 90000	cel	83.36	85.71 (+2.82)	83.39 (+0.03)	83.44 (+0.10)
		gam	86.12	89.40 (+3.81)	86.14 (+0.02)	86.13 (+0.02)
		web	84.60	90.21 (+6.63)	84.64 (+0.05)	84.65 (+0.06)
	cel, 90000	cel	90.85	93.20 (+2.58)	90.98 (+0.14)	91.01 (+0.17)
		gam	79.67	89.54 (+12.40)	79.72 (+0.07)	79.73 (+0.08)
		web	78.21	89.52 (+14.46)	78.27 (+0.08)	78.24 (+0.04)

4,5,6,7 열은 음절 단위 정확률을 표현하고, 괄호 안은 성능 변화율을 표현함

3은 테스트 말뭉치의 각 테스트 문장을 사용자 입력으로 간주한다. <표 3>의 결과를 보면, 대부분의 경우에 대해 제안 방법이 효과적임을 알 수 있다. 제안 방법을 적용한 경우가 <표 3>에서 총 36개인데, 이 중에서 4개의 경우에만 약간의 성능 하락이 발생하고 나머지 경우는 모두 성능 향상을 보이고 있다.

제안 방법의 $method1$, 2, 3 간의 비교 분석을 하면 $method1$, 2, 3 중에서는 $method3$의 성능이 $method1$, 2에 비해 불안정하다. 이는 사용자 입력 오류 모델 수식 (9)의 학습을 cel 말뭉치에서 하였는데, 통계적으로 충분한 수준의 학습이 되지 못하여 부정확한 예측을 하기 때문이다. 그리고 $method1$은 $method2$에 비해 사용자 입력의 정확률에 더 큰 영향을 받는다는 점을 알 수 있다. 이는 $method1$은 사용자 입력에 대해 차등화된 전략을 적용하지 않기 때문이다. 따라서 cel과 같이 오류가 많은 경우에는 성능이 하락하기도 하지만 사용자 입력 오류가 상대적으로 적은 web, gam에 대해서는 $method2$에 비해 더 큰 성능 향상을 보이고 있다. $method2$는 3가지 $method$ 중에 가장 안정적인 성능 향상을 보이고 있다. 주요 원인은 사용자 입력 오류가 주로 띄붙 오류가 붙띄 오류보다 훨씬 많은 비중을 차지하는데 이러한 현상을 반영하기 때문이다.

<그림 2> 학습량에 따른 제안 방법의 성능 향상



<그림 2>는 학습량에 따른 제안 방법에 의한 성능 향상 정도를 실험한 결과이다. $HMM(method2)$로 sej 학습 말뭉치를 1만, 2만, ..., 9만으로 분할 학습하였으며 $HMM(method0)$을 기저성능으로 측정하였다. <그림 2>는 학습 말뭉치의 크기가 작은 경우, 최대 우도 추정에 비해 제안방법이 더욱 효과적임을 의미한다.

7. 결론

본 연구는 기존의 띄어쓰기 모델이 사용한 최대 우도 추정보다 한국어 띄어쓰기 문제에 더욱 적합한 사용자 입력을 고려하는 베이지언 파라미터 추정을 제안하였다. 기존 연구에서 배제하던 사용자 입력을 학습 말뭉치의 자료부족을 완화하는 유용한 자원으로 해석하였으며, 이에 대한 이론적 근거로 베이지언 파라미터 추정을 적용하였다. 또한, 제안 방법은 최대 우도 추정을 사용하는 모든 통계적 띄어쓰기 모델에 대해 적용가능하며 추가적인 말뭉치 없이 기존 띄어쓰기 모델의 성능을 향상시킬 수 있음을 보였다.

참고 문헌

[1] Duda, Hart, Stork, Pattern Classification 2nd Edition, Wiley-Interscience, 2001

[2] Peter M. Lee, Bayesian Statistics an introduction 3rd Edition, 2004

[3] David J.C. MacKay, Linda C. Bauman Peto, “A Hierarchical Dirichlet Language Model“, Natural Language Engineering, Volume 1, page 1-19, 1995

[4] Sharon Goldwater, Thomas L. Griffiths, “A Fully Bayesian Approach to Unsupervised Part-of-Speech Tagging”, Association of Computational Linguistics, page 744-751, 2006

[5] Do-Gil Lee, Hae-Chang Rim, Dongsuk Yook, “Automatic Word Spacing Using Probabilistic Models Based on Character n-grams”, IEEE Intelligent Systems, Volume 22, Issue 1, 2007

[6] 강승식, “음절 bigram를 이용한 띄어쓰기 오류의 자

동 교정”, 음성과학, 제 8권 제 2호, 2001