인지과학, 제22권 제2호
Korean Journal of Cognitive Science
2011, Vol. 22, No. 2, 217～233.

CRF를 이용한 한국어 자동 띄어쓰기*

심 광 섭†
성신여자대학교 IT학부

본 논문에서는 띄어쓰기가 전혀 되어 있지 않은 한국어 문장을 입력받아 자동으로 띄어쓰기를 해 주는 시스템을 제안한다. 띄어쓰기 문제는 주어진 문장의 각 음절에 대하여 띄어쓰기 여부를 나타내는 레이블을 부착하는 일종의 레이블링 문제이므로, 본 논문에서는 레이블링 문제 해결에 띄어난 성능을 보이는 것으로 알려진 CRF를 이용하여 자동 띄어쓰기를 시도하였다. 약 112만 음절 규모의 학습용 데이터로 학습을 하고, 2,114 문장(약 9.3만 음절)의 평가용 데이터로 띄어쓰기 정확도에 대한 평가를 하였다. 평가 결과 음절 단위의 정확도는 98.84%, 어절 단위의 정확도는 95.99%인 것으로 나타났다.

주제어 : 자동 띄어쓰기, CRF, 레이블링 문제

---

* 이 논문은 2010년도 성신여자대학교 학술연구조성비 지원에 의하여 연구되었음.
† 교신저자: 심광섭, 성신여자대학교 IT학부, 연구 분야: 한국어 정보 처리
E-mail: shim@sungshin.ac.kr

서 론

자연어 처리 분야에서는 문장을 단어 단위로 분리하여 처리하는 경우가 많다. 따라서 중국어나 일본어와 같이 띄어쓰기를 전혀 하지 않는 언어에서는 주어진 문장을 단어 단위로 분리하는 과정이 필수적이다. 이 때문에 이들 언어권에서는 문장을 단어 단위로 분리하는 방법에 대한 연구가 많이 진행되었다[1].

한국어에서는 기본적으로 어절 단위로 띄어쓰기를 하기 때문에 중국어나 일본어에 비하여 문장을 단어로 분리하는 것에 대한 필요성을 그다지 많이 느끼지 않았다. 하지만 실생활에서 쓰이는 한국어를 보면 띄어쓰기가 올바르게 되지 않은 경우가 많다. 또 문자 인식기에서는 문자가 공백으로 대치되거나 없는 공백이 삽입되는 등의 오류가 발생하면서 띄어쓰기가 잘못되는 상황이 생길 수 있다[2]. 음성 인식기에서도 비슷한 문제가 나타나는데, 발화에서는 어절과 어절 사이에 별도의 휴지가 부여되지 않기 때문에 음성 인식 결과는 띄어쓰기에 대한 정보가 없는 경우가 보통이다[3]. 요즘에는 메신저나 SMS 등을 통한 대화 시스템의 사용이 보편화되고 있는데, 이러한 대화 시스템의 사용자들은 띄어쓰기 규칙을 아예 무시하고 메시지를 작성하는 경우가 많다. 띄어쓰기를 올바르게 했다 하더라도 별도의 표지 없이 어절 중간에서 줄바꿈을 할 수 있는 한국어의 특성 상 줄바꿈을 한 부분에서 띄어쓰기를 한 것인지 아닌지에 대한 구분을 할 수가 없다.

이상에서 본 바와 같이 한국어는 중국어나 일본어와 달리 띄어쓰기를 하는 언어이면서 실제 상황에서는 띄어쓰기가 올바르게 되어 있지 않은 경우가 많이 발생한다. 따라서 실제 문장을 대상으로 하는 자연어 처리 응용에서는 띄어쓰기 오류를 제거하는 자동 띄어쓰기 단계의 도입이 필요하다. 한국어의 자동 띄어쓰기에 대한 지금까지의 연구 내용을 보면 부분적으로 띄어쓰기가 되어 있는 문장을 대상으로 한 경우도 있으나[4] 대부분은 띄어쓰기가 전혀 되어 있지 않는 문장을 대상으로 하고 있다[5, 6, 7, 8, 9]. 부분적으로 띄어쓰기가 되어 있는 문장을 띄어쓰기가 전혀 되어 있지 않은 문장으로 변환할 수 있으므로 후자와 같은 접근 방법이 훨씬 더 일반적이라 할 수 있다.

본 논문에서는 띄어쓰기가 전혀 되어 있지 않은 문장을 입력받아 자동으로 띄어쓰기를 해 주는 시스템을 제안한다. 본 논문에서 제안하는 시스템은 띄어쓰기가

되어 있는 말뭉치를 대상으로 CRF(Conditional Random Fields)에 기반한 학습을 하는데, CRF는 입력 데이터 열에 대하여 레이블을 부착하는 문제에 있어서 HMM(Hidden Markov Model)이나 MEMM(Maximum Entropy Markov Model)에 비하여 우수한 성능을 보이는 것으로 보고된 바 있다[10, 11].

관련 연구

한국어 자동 띄어쓰기에 대한 기존 연구는 크게 규칙 기반의 분석적인 접근 방법과 말뭉치 기반의 통계적인 접근 방법으로 나눌 수 있다.

먼저 분석적인 접근 방법을 살펴보면, [4]에서는 일반 문서에서 빈번하게 나타나는 띄어쓰기 오류 유형을 분석하고 이들을 처리하기 위한 여러 가지 휴리스틱을 제시하였다. [5]에서는 형태소 분석기를 이용하여 어절 경계를 인식하는 방법을 사용하였는데, 이 과정에서 어느 한 곳에서 어절 경계 인식이 잘못 되게 되면 다음에 나오는 다른 어절에 대한 경계 인식도 잘못 되는 이른바 전파 오류 문제가 발생한다. 이 문제를 해결하기 위하여 [5]에서는 어절 경계가 비교적 명확한 지점을 중심으로 주어진 문장을 여러 개의 어절 블록으로 나누고 각 어절 블록에 대하여 형태소 분석기를 이용한 띄어쓰기를 수행하는 방법을 제안하였으며, 93.2%의 정확도를 얻은 것으로 보고되었다. [6]에서는 음절 수준의 결합 범주 문법을 사용하여 띄어쓰기와 구문 분석을 동시에 수행하는 방법을 제안하였다. 이러한 방법을 제안한 것은 “누나가방에들어간다”를 “누나가 방에 들어간다”로 띄어 써야 할지 아니면 “누나 가방에 들어간다”로 띄어 써야 할지 모호하며, 이러한 모호성을 해소하려면 구문 분석이 필요하다고 보았기 때문이다. 하지만 모든 문장이 구문 분석까지 해야 할 정도의 모호성을 지니는 것이 아님에도 불구하고 항상 구문 분석을 통해 띄어쓰기를 해야 하기 때문에 처리 시간이 문제가 된다. 구문 분석 시간은 통상적으로 문장 길이의 세제곱에 비례하므로 이 방법에서는 문장 길이가 길면 길수록 처리 시간은 급격하게 증가하게 된다.

통계적인 접근 방법에 대하여 살펴보면, [7]에서는 말뭉치로부터 두 음절 사이에서 띄어 쓸 가능성, 두 음절 앞이나 뒤에서 띄어 쓸 가능성 등에 대한 바이그램

(bigram) 정보를 수집하고 이것을 자동 띄어쓰기에 적용하는 방법을 제안하였다. 바이그램 모델의 단점은 띄어쓰기에 사용되는 문맥의 크기가 고정되어 있다는 점이다. 문맥의 크기를 늘린다면 띄어쓰기 정확도가 높아지겠지만 이때에는 데이터 부족 문제가 나타난다. [8]에서는 데이터 부족 문제에 적절히 대처하면서도 띄어쓰기 정확도를 개선할 수 있는 자기 조직화 n-gram 모델을 제안하였다. 이 모델은 바이그램을 기본으로 하되 주어진 문제의 특성에 따라 문맥의 크기를 트라이그램(trigram)으로 확장하기도 하고 유니그램(unigram)으로 축소하기도 한다. 이렇게 함으로써 바이그램 정보를 이용할 때에 비하여 띄어쓰기 정확도가 약 3% 가량 개선되었다[8]. 한편, [9]에서는 기존의 통계적 접근 방법에서 이전 어절의 띄어쓰기 상태는 고려하지 않는다는 것을 문제점으로 지적하고, 띄어쓰기 문제를 품사 부착(POS tagging)과 같은 분류 문제로 간주하고 처리하는 HMM(Hidden Markov Model) 기반의 띄어쓰기 모델을 제시하였다. 이 모델은 Markov 윈도우의 크기에 따라 총 72개의 유형으로 분류되는데, 각 유형에 대한 성능을 비교하는 실험을 한 결과 가장 정확도가 높을 때가 93.06%인 것으로 나타났다[9]. 통계적 접근 방법에서 가장 큰 문제점 중의 하나는 축약어, 신조어 등 통계 정보가 없는 새로운 단어에 대해서는 잘못된 결과를 제시한다는 것이다. 이러한 문제점에 대한 해결 방안으로 말뭉치보다는 훨씬 방대한 웹문서로부터 통계 정보를 얻어 활용하는 방법에 대한 연구도 있다[10].

CRF

입력 데이터 열을 분할하고 각각에 레이블을 부여하는 문제에 대한 해결 방법으로 HMM(Hidden Markov Model)이 사용되어 왔다. HMM은 입력 데이터 열과 레이블 열 사이의 결합 확률(joint probability)을 이용하는 생성 모델(generative model)이다. 이 모델에서는 모든 가능한 입력 데이터 열을 나열해야 하며, 상호 작용하는 자질을 표현하거나 멀리 떨어진 입력 데이터 열 사이의 의존 관계를 표현하기 어렵다는 단점이 있다[11, 12, 13]. 이러한 단점을 극복하기 위하여 주어진 입력 데이터 열에 대하여 레이블 열의 확률을 이용하는 조건부 모델이 제안되었다. CRF

(Conditional Random Fields)는 조건부 모델의 일종으로, 일반적으로는 조건부 확률을 최대화하기 위해 훈련된 비방향성 그래프 모델이다[11, 14]. 일반적인 그래프 구조를 가진 CRF 모델의 특수한 형태인 선형 체인 구조의 CRF 모델은 입력 데이터 열에 레이블 (label) 열을 부여하는 문제에 적합하므로 지금부터는 선형 체인 구조의 CRF를 기준으로 설명한다.

$\mathbf{x} = x_1 \cdots x_n$를 입력 데이터 열에 대한 확률 변수(random variable)라고 하고, $\mathbf{y} = y_1 \cdots y_n$를 입력 데이터 열에 대응하는 레이블 열의 확률 변수라고 하자. 매개 변수 $\Lambda = (\lambda_1, \lambda_2, \cdots, \mu_1, \mu_2, \cdots)$를 갖는 선형 체인 구조의 CRF는 다음과 같은 조건부 확률로 정의된다[13].

$$P_\Lambda(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\left(\sum_j \sum_{i=1}^{n} \lambda_j t_j(y_{i-1}, y_i, \mathbf{x}, i) + \sum_k \sum_{i=1}^{n} \mu_k s_k(y_i, \mathbf{x}, i)\right) \quad \text{(식 1)}$$

여기서 $Z(\mathbf{x})$는 입력 데이터 열에 대한 레이블 열의 확률값의 합이 1이 되도록 하는 정규화 상수이다. $t_j(y_{i-1}, y_i, \mathbf{x}, i)$는 전이 자질 함수(transition feature function)이며, $s_k(y_i, \mathbf{x}, i)$는 상태 자질 함수(state feature function)이다. 다음은 전이 자질 함수와 상태 자질 함수의 예이다[1].

$$t_j(y_{i-1}, y_i, \mathbf{x}, i) = \begin{cases} 1 & y_{i-1} = NP,\ y_i = JO\text{이고 } i\text{번째 단어가 '는'인 경우} \\ 0 & \text{그 밖의 경우} \end{cases}$$

$$s_k(y_i, \mathbf{x}, i) = \begin{cases} 1 & y_i = NX\text{이고 } i\text{번째 단어가 '수'인 경우} \\ 0 & \text{그 밖의 경우} \end{cases}$$

1) 이 예에서 전이 자질 함수는 i번째 단어가 '는'이고 $i-1$, i번째 단어에 부여된 레이블이 각각 NP, JO인 경우와 그 밖의 경우를 구분하는 함수이며, 상태 자질 함수는 i번째 단어 '수'이고 이 단어에 부여된 레이블이 NX인 경우와 그 밖의 경우를 구분하는 함수이다. 이러한 레이블들은 학습용 데이터로부터 주어진다.

λ_j와 μ_k는 각 자질 함수에 대한 가중치로서 레이블링(labeling)이 된 학습용 데이터로부터 구할 수 있다. 매개 변수 Λ는 MLE(Maximum Likelihood Estimation)를 사용하여 구하는데, 다른 알고리즘보다 수렴 속도가 빠른 BFGS 알고리즘이 주로 사용된다[12, 13].

학습용 데이터로부터 매개 변수 Λ를 구하고 나면, 주어진 입력 데이터 열 $\mathbf{x}$에 대하여 가장 가능성이 높은 레이블 열 $\mathbf{y}^*$은 다음과 같이 구할 수 있다[14].

$$\mathbf{y}^* = \arg\max_{\mathbf{y}} P_{\Lambda}(\mathbf{y} \mid \mathbf{x}) \quad \text{(식 2)}$$

$\mathbf{y}^*$를 구하는 과정은 Viterbi 알고리즘을 사용한 동적 프로그래밍으로 계산할 수 있다.

입력 데이터 열을 분할하고 각각에 레이블을 부여하는 레이블링 문제에 대한 해결 방법으로 제안된 CRF는 그동안 여러 분야에 적용되어 왔다. [12]에서는 Base NP 청킹(chunking)을 BIO 태그를 부여하는 일종의 레이블링 문제로 보고 CRF를 적용하였다. [14]에서는 중국어 문장을 단어 단위로 분할하는 데 CRF를 적용하여 좋은 결과를 얻었다. 자연스럽고 알아듣기 쉬운 음성을 만들기 위하여 올바른 운율 경계를 추정하는 일은 TTS(Text-To-Speech)에서 매우 중요한 작업인데, [15]에서는 한국어 운율 경계를 추정하는 문제를 클래스 분류 문제로 보고 이 문제를 해결하는 데 CRF를 적용하였다. 이들은 CRF 학습을 위한 자질로 단어의 품사, 어휘, 단어의 길이, 문장에서의 단어 위치 등과 같은 다양한 언어적 정보를 사용하였다. [16]에서는 형태소 분석 결과로부터 구문 분석의 최소 단위인 어절 구문 태그를 예측하는 문제를 클래스 분류 문제로 보고 이 문제를 해결하는 데 CRF를 적용하였다.

CRF를 이용한 한국어 자동 띄어쓰기

자동 띄어쓰기 문제는 주어진 문장의 각 음절에 대하여 띄어쓰기를 할 것인가

말 것인가를 나타내는 레이블을 부여하는 일종의 레이블링 문제로 볼 수 있다. 예를 들어 “이번일을계기로한글의우수성이널리알려질수있었다”라는 문장의 각 음절에 대하여 다음과 같은 레이블을 부여하는 것이다. 여기서 B는 해당 음절에서 새로운 어절이 시작됨을 나타내며, I는 어절이 계속되고 있음을 나타내는 레이블이다.

이번일을계기로한글의우수성이널리알려질수있었다
B I B I B I I B I I B I I I B I B I I B B I I

위에서 살펴 본 바와 같이 한국어 자동 띄어쓰기는 일종의 레이블링 문제로 볼 수 있으므로, 본 논문에서는 레이블링 문제 해결에 탁월한 성능을 발휘하는 것으로 보고된 CRF를 한국어 자동 띄어쓰기에 적용해 보고자 한다. HMM은 입력 데이터 열과 레이블 열 사이의 결합 확률을 이용하는 생성 모델로, 상호 작용하는 자질들을 표현하거나 멀리 떨어진 입력 데이터 열 사이의 의존 관계를 표현하기 어렵다는 단점이 있다[11, 12, 13]. 반면에 CRF는 (식 1)에서 보듯이 입력 데이터 열 $\mathbf{x}$에 대한 조건부 확률로 정의되므로 상호 작용하는 자질들이나 멀리 떨어진 입력 데이터 열 사이의 의존 관계를 표현하는 것이 가능하다[2].

다음과 같이 띄어쓰기가 전혀 되어 있지 않은 n 개의 연속된 한국어 음절 열이 주어졌다고 하자. 이 음절 열이 몇 개의 어절로 분리되어야 하는가에 대해서는 알지 못하는 것으로 가정한다.

$$\mathbf{x} = x_1 x_2 \cdots x_{i-2}\ x_{i-1}\ x_i\ x_{i+1}\ x_{i+2} \cdots x_n$$

자동 띄어쓰기를 위해 우리는 CRF를 이용하여 주어진 음절 열에 대하여 다음과 같이 길이가 n인 레이블 열을 결정하려고 한다. 여기서 각 레이블 y_j는 B이거나 또는 I이다.

$$\mathbf{y} = y_1 y_2 \cdots y_{i-2}\ y_{i-1}\ y_i\ y_{i+1}\ y_{i+2} \cdots y_n$$

---

2) CRF는 입력 데이터 열 $\mathbf{x}$에 대한 조건부 확률로 정의된다고 했는데, 이는 입력 데이터에서 추출한 자질 정보에 대한 조건부 확률도 포함하고 있다.

이러한 결정을 하기 위해서는 레이블링이 된 학습용 데이터가 필요하다. 띄어쓰기가 제대로 되어 있는 원시 말뭉치(raw corpus)가 있다면, 이 말뭉치의 각 어절 경계를 기준으로 어절 시작에 해당하는 음절에 대해서는 B를, 나머지 음절에 대해서는 I라는 레이블을 부여하는 방법으로 학습용 데이터를 만들 수 있다. 그런데 원시 말뭉치는 구축 방법에 따라 다소간의 차이는 있겠지만 실제 문장을 그대로 모으다 보니 띄어쓰기 오류가 포함될 가능성이 상대적으로 높다. 따라서 본 논문에서는 사람이 직접 개입하여 품사를 부착하고 오류 검증을 한 코난 품사 태깅 말뭉치(tagged corpus)[3]를 사용하여 학습용 데이터와 평가용 데이터를 만들었다. 평가용 데이터는 코난 품사 태깅 말뭉치의 처음 33,128 어절(93,299 음절)을 발췌하여 만들었고, 학습용 데이터는 나머지 부분에서 400,282 어절(1,127,070 음절)을 발췌하여 만들었다[4].

띄어쓰기 문제에서는 어절 경계를 중심으로 전·후 몇 음절만 참조하는 것만으로도 충분할 것이다. 실제로 [7]에서는 어절 경계를 중심으로 전·후 한 음절 또는 두 음절을 참조하여 띄어쓰기를 하였고, [8]에서는 n-gram 모델을 사용하되 n이 2인 경우를 기준으로 상황에 따라 n 값을 축소 또는 확장할 수 있도록 하였다. 여기에서는 CRF를 사용했을 때 어절 경계를 중심으로 전·후 몇 음절까지 참조하는 것이 좋은지 판단하기 위하여 CRF++를 사용하여 간단한 실험을 수행하였다[5].

기존 연구 결과를 볼 때 어절 경계에서 멀리 떨어진 음절은 띄어쓰기 문제 해결에 별로 도움이 되지 않는 것으로 보인다. 그래서 여기에서는 어절 경계를 중심으로 전·후 두 음절의 범위에서 각 음절이 띄어쓰기 문제 해결에 어느 정도 기여를 하는가에 대한 평가를 하였으며, 결과는 표 1과 같다. 여기서 x_i는 새로운 어절이 시작되는 음절 위치를 나타내며, 나머지는 이 음절을 중심으로 한 상대적인 위

3) 코난 품사 태깅 말뭉치는 주식회사 코난테크놀로지(http://www.konantech.co.kr/)에서 연구개발용으로 구축한 여러 종류의 말뭉치 중 하나이다.

4) 학습 데이터가 더 큰 경우에는 메모리 용량 문제로 학습을 하지 못하게 되는 문제가 발생하였다.

5) CRF++는 CRF 모델에 따라 학습을 하고 학습된 결과를 레이블링 문제에 활용할 수 있도록 해주는 공개 소프트웨어이다[17]. CRF++에서는 학습을 위해 템플릿(template)을 정의하여야 하는데, 템플릿은 상태 자질 함수를 학습하는 데 사용되는 유니그램(unigram) 템플릿과 전이 자질 함수를 학습하는 데 사용되는 바이그램(bigram) 템플릿으로 구성된다.

치를 나타낸다. 표 1의 각 행은 해당 음절에 대한 정보만으로 띄어쓰기를 했을 때 얻어진 음절 단위의 정확도(A_{syl})와 어절 단위의 정확도(A_{word})를 보여 준다. x_i는 새로운 어절이 시작되는 음절 위치를 나타내므로, 이 표에서 x_{i-2}에 해당하는 행은 어절 경계에서 두 음절 앞에 있는 음절 정보만으로 띄어쓰기를 했을 때 얻어진 정확도를 보여 준다[6].

표 1. 띄어쓰기에 대한 각 음절의 기여도

x	상태 자질 (y_i)		전이 자질 ($y_{i-1}\, y_i$)	
	A_{syl} (%)	A_{word} (%)	A_{syl} (%)	A_{word} (%)
x_{i-2}	72.58	17.24	**88.40**	**61.75**
x_{i-1}	**84.13**	**45.73**	**92.16**	**75.51**
x_i	**80.12**	**42.06**	**88.78**	**65.59**
x_{i+1}	**73.46**	**23.64**	77.25	40.50
x_{i+2}	70.98	18.40	70.76	28.98

이 표에서 알 수 있듯이 한 음절만으로 띄어쓰기를 할 경우 이전 어절의 마지막 음절인 x_{i-1}과 새로운 어절의 첫 음절인 x_i가 띄어쓰기 문제 해결에 가장 많은 기여를 한다는 것을 알 수 있다. 그 다음으로 중요한 역할을 하는 음절로 상태 자질에서는 새로운 어절의 두 번째 음절인 x_{i+1}이고, 전이 자질에서는 이전 어절의 끝에서 두 번째 음절인 x_{i-2}이다.

다음은 연속한 두 음절이 띄어쓰기 문제 해결에 어느 정도의 기여를 하는가에 대한 실험을 하였다. 실험 결과는 표 2와 같은데, 이것을 표 1과 비교해 보면 연속한 두 음절을 참조하는 경우에는 보다 많은 음절 정보를 사용하기 때문에 정확도가 월등이 높아지는 것을 알 수 있다. 연속한 두 음절을 참조하는 경우에는 상태

6) 음절 단위의 정확도는 각 음절에 대하여 정답 문서에 부착된 레이블과 자동 띄어쓰기 시스템에 의해 부착된 레이블이 얼마나 일치하는가를 나타내는 평가 척도이다. 어절 단위의 정확도는 자동 띄어쓰기 시스템에 의해 얼마나 정확하게 어절이 식별되었는가를 나타내는 평가 척도이다.

표 2. 띄어쓰기에 대한 연속한 두 음절의 기여도

x	상태 자질 (y_i)		전이 자질 ($y_{i-1}\, y_i$)	
	A_{syl} (%)	A_{word} (%)	A_{syl} (%)	A_{word} (%)
x_{i-3}/x_{i-2}	75.16	29.87	91.28	70.24
x_{i-2}/x_{i-1}	**86.99**	**54.32**	**97.05**	**88.97**
x_{i-1}/x_i	**95.34**	**81.62**	**97.77**	**91.56**
x_i/x_{i+1}	**90.70**	**64.48**	**93.17**	**77.76**
x_{i+1}/x_{i+2}	81.17	43.52	83.00	49.75
x_{i+2}/x_{i+3}	70.39	23.46	71.64	30.41

자질이나 전이 자질 모두 x_{i-2}에서 x_{i+1} 사이의 음절이 띄어쓰기 문제 해결에 가장 많은 기여를 한다.

마지막으로 연속한 세 음절이 띄어쓰기 문제 해결에 어느 정도 기여하는가를 알아보는 실험을 하였다. 실험 결과는 표 3과 같은데 연속한 두 음절을 참조하여 띄어쓰기를 하는 것보다 오히려 좋지 않은 결과가 나왔다. 연속한 세 음절을 참조하는 경우에는 자료 부족 문제(data sparseness)가 발생하기 때문에 이러한 현상이 발

표 3. 띄어쓰기에 대한 연속한 세 음절의 기여도

x	상태 자질 (y_i)		전이 자질 ($y_{i-1}\, y_i$)	
	A_{syl} (%)	A_{word} (%)	A_{syl} (%)	A_{word} (%)
$x_{i-4}/x_{i-3}/x_{i-2}$	67.41	15.86	82.08	39.92
$x_{i-3}/x_{i-2}/x_{i-1}$	74.25	20.36	**88.37**	**52.16**
$x_{i-2}/x_{i-1}/x_i$	**86.28**	**41.10**	**93.18**	**66.95**
$x_{i-1}/x_i/x_{i+1}$	**84.05**	**32.33**	**90.04**	**53.45**
$x_i/x_{i+1}/x_{i+2}$	**79.45**	**24.68**	85.64	47.30
$x_{i+1}/x_{i+2}/x_{i+3}$	74.26	22.85	75.69	29.72
$x_{i+2}/x_{i+3}/x_{i+4}$	62.91	11.96	66.39	18.94

생한 것으로 추측된다. 연속한 세 음절을 참조하는 경우, 상태 자질에서는 x_{i-2}에서 x_{i+2} 사이의 음절이 띄어쓰기 문제 해결에 가장 많은 기여를 하며, 전이 자질에서는 x_{i-3}에서 x_{i+1} 사이의 음절이 띄어쓰기 문제 해결에 가장 많은 기여를 하고 있음을 알 수 있다.

이 실험 결과를 바탕으로 표 4와 같은 자질 집합을 정의하였다. 이 자질 집합은 표 1, 표 2, 표 3에서 굵은 글씨체로 나타낸 부분에 해당하는 음절 혹은 음절 열을 기반으로 하여 정의한 것이다.

표 4. 자질 집합 A

상태 자질 (y_i)	전이 자질 ($y_{i-1}\, y_i$)
· x_{i-1}, x_i, x_{i+1}	· x_{i-2}, x_{i-1}, x_i
· x_{i-2}/x_{i-1}, x_{i-1}/x_i, x_i/x_{i+1}	· x_{i-2}/x_{i-1}, x_{i-1}/x_i, x_i/x_{i+1}
· $x_{i-2}/x_{i-1}/x_i$, $x_{i-1}/x_i/x_{i+1}$, $x_i/x_{i+1}/x_{i+2}$	· $x_{i-3}/x_{i-2}/x_{i-1}$, $x_{i-2}/x_{i-1}/x_i$, $x_{i-1}/x_i/x_{i+1}$

다음은 표 1, 표 2, 표 3으로 주어진 결과를 무시하고 임의로 표 5와 같이 자질 집합을 만들어 보았다. 이 자질 집합은 x_i를 중심으로 대칭꼴로 동일한 개수의 음절을 참조하도록 한 것이다. 표 4와 표 5의 상태 자질 부분을 비교해 보면 x_{i+1}/x_{i+2}이 추가된 것만 제외하면 동일하다. 전이 자질 부분은 연속한 두 음절

표 5. 자질 집합 B

상태 자질 (y_i)	전이 자질 ($y_{i-1}\, y_i$)
· x_{i-1}, x_i, x_{i+1}	· x_{i-1}, x_i, x_{i+1}
· x_{i-2}/x_{i-1}, x_{i-1}/x_i, x_i/x_{i+1}, x_{i+1}/x_{i+2}	· x_{i-2}/x_{i-1}, x_{i-1}/x_i, x_i/x_{i+1}, x_{i+1}/x_{i+2}
· $x_{i-2}/x_{i-1}/x_i$, $x_{i-1}/x_i/x_{i+1}$, $x_i/x_{i+1}/x_{i+2}$	· $x_{i-2}/x_{i-1}/x_i$, $x_{i-1}/x_i/x_{i+1}$, $x_i/x_{i+1}/x_{i+2}$

을 제외하고 한 음절씩 뒤로 이동했다는 차이점이 있다.

앞에서 설명한 두 가지 자질 집합을 사용하여 학습한 경우 띄어쓰기 정확도가 표 6과 같이 나타났다. 표 6에서 보듯이 상태 자질만 사용하는 경우에는 어느 자질 집합을 사용하든 띄어쓰기 정확도에 거의 차이가 없으나, 전이 자질만 사용하는 경우 혹은 상태 자질과 전이 자질을 모두 다 사용하는 경우에는 자질 집합 A를 이용하여 학습한 경우에 띄어쓰기 정확도가 더 높아지는 것으로 나타났다.

표 6. 자질 집합에 따른 정확도 비교

	자질 집합	A_{syl} (%)	A_{word} (%)
상태 자질	A	98.62	95.04
	B	98.65	95.17
전이 자질	A	98.81	95.89
	B	98.70	95.48
상태 자질 + 전이 자질	A	98.84	95.99
	B	98.72	95.59

CRF를 사용하여 한국어 자동 띄어쓰기를 했을 때 얻어진 정확도를 다른 방법론에서의 결과와 비교하면 표 7과 같다.

표 7. 방법론에 따른 정확도 비교

학습 방법	A_{syl} (%)	A_{word} (%)
Markov 가정을 완화한 HMM	98.33	93.06
자기 조직화 바이그램 모델	N/A	94.71
CRF	98.84	95.99

Markov 가정을 완화한 HMM 모델은 기존의 HMM 모델에서는 고정된 문맥을 참조하는 데 반하여 확장된 문맥을 참조할 수 있도록 Markov 가정을 완화한 모델이다[9]. 문맥의 범위를 어디까지 확장하는가에 따라 성능에 차이가 발생하게 되는

데, [9]에서는 문맥 범위를 72 가지로 구분하고 각각에 대한 성능 평가를 수행하고 최고의 성능을 보이는 문맥 범위를 결정하였다. 실험 결과에 따르면 이 방법론에서는 최고 98.33%의 음절 단위의 정확도와 93.06%의 어절 단위 정확도를 얻은 것으로 나타났다.

자기 조직화 바이그램 모델은 문맥의 범위를 확장해야 하는지 축소해야 하는지를 판정하는 두 가지 함수를 정의하고 이 함수의 값이 일정한 값을 초과하는 경우 문맥의 범위를 확장 또는 축소하는 방법으로 주어진 문제에 따른 최적의 문맥 범위를 찾는 방법론이다[8]. 실험 결과에 의하면 자기 조직화 바이그램 모델의 어절 단위 정확도는 94.71%인 것으로 나타났다. [9]에서는 비록 문맥 범위를 확장하기는 하였으나 일반적인 HMM 모델을 전제로 하기 때문에 고정된 크기의 문맥만을 참조할 수 있는데 반하여, [8]에서는 HMM 모델에 자기 조직화 개념을 도입하여 문맥의 크기를 상황에 따라 가변적으로 조절할 수 있도록 함으로써 정확도가 향상될 수 있었던 것으로 판단된다.

CRF에서는 학습을 하게 되면 학습용 데이터 및 학습용 템플릿으로부터 전이 자질 함수와 상태 자질 함수가 정의되고 학습용 데이터에 대한 레이블링 오류를 최소화하도록 각 함수의 가중치 λ_j 및 μ_k 값이 결정된다. 학습 결과를 띄어쓰기 문제 해결에 사용할 때에는 가중치 값에 따라 자질 함수의 반영 정도가 달라지므로, [9]와 같이 고정된 문맥을 사용하는 방법이나 [8]과 같이 바이그램을 기준으로 문맥을 확장 또는 축소하는 방법보다는 유연하게 문맥 정보를 활용할 수 있기 때문에 띄어쓰기 정확도가 더욱 향상될 수 있었던 것으로 판단된다.

CRF++는 학습 시 줄 수 있는 인자(parameter)가 있는데, 이 인자 값에 따라 레이블링 성능이 영향을 받는다. 일반적으로 이 인자 값이 크면 클수록 학습용 데이터에 과적합(overfit)하는 경향이 있다. 표 4에서 주어진 자질 집합 A에서 상태 자질과 전이 자질을 모두 다 사용하는 것을 가정했을 때 이 인자 값에 따라 띄어쓰기 정확도가 어떻게 변화하는지에 대한 관찰을 하였다. 그 결과 인자 값이 1.5일 때 음절 단위의 정확도와 어절 단위의 정확도가 제일 높았으며, 인자 값이 커짐에 따라서 정확도는 대체로 감소하는 것으로 나타났다. 그림 1은 인자 값을 0.5에서부터 0.5씩 증가시키면서 6.5까지 변화시켰을 때 정확도가 변화하는 양상을 나타낸 것이다.

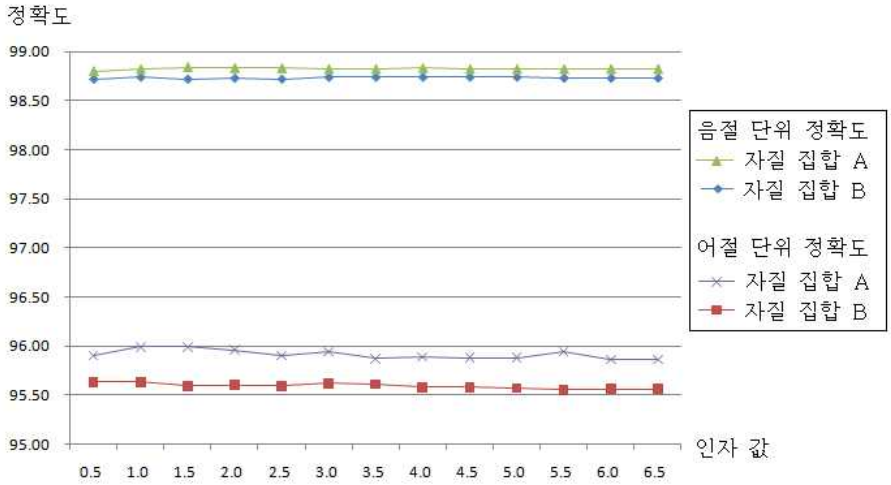

그림 1. 인자 값에 따른 성능 비교

결 론

한국어 자동 띄어쓰기 문제는 주어진 문장의 각 음절에 대하여 띄어쓰기를 할 것인가 말 것인가를 나타내는 레이블을 부착하는 일종의 레이블링 문제로 볼 수 있다. 본 논문에서는 레이블링 문제 해결에 띄어난 성능을 보이는 것으로 알려진 CRF를 이용하여 한국어 자동 띄어쓰기를 시도해 보았다. 띄어쓰기가 되어 있는 원시 말뭉치로부터 약 112만 음절 규모의 학습용 데이터를 만들어 학습을 하였다. 2,114 문장(약 9.3만 음절)의 평가용 데이터로 띄어쓰기 정확도에 대한 평가를 한 결과 음절 단위의 정확도는 98.84%, 어절 단위의 정확도는 95.99%인 것으로 나타났다. 이는 기존 연구 결과와 비교할 때 약 1.3 ~ 3.0% 정도 개선된 것이다.

사람들도 띄어쓰기 오류를 자주 범하긴 하지만 주로 착오에 의한 오류이기 때문에 오류 유형이 비교적 제한적이며 의미 전달에 크게 문제가 되는 경우는 드물다 할 수 있다. 그러나 기계 학습에 의한 자동 띄어쓰기의 경우에는 예상치 못한 곳에서 오류가 발생하기 때문에 문장의 의미가 훼손되는 경우가 종종 발생한다는 문제점을 안고 있다. 따라서 기계 학습에 의한 자동 띄어쓰기 결과를 실제 응용

에 적용하기 위해서는 이러한 문제점에 대한 해결 방안을 마련하여야 할 것으로 보인다.

참고문헌

[1] Jianfeng Gao, Mu Li and Chang-Ning Huang (2003), Improved Source-Channel Model for Chinese Word Segmentation, Proceedings of the 41st Annual Meeting of the ACL, pp.272-279.

[2] 전남열, 박혁로 (2000), 음절 Bi-gram정보를 이용한 한국어 OCR 후처리용 자동 띄어쓰기, **제12회 한글 및 한국어 정보처리 학술대회 논문집,** pp.95-100.

[3] 임동희, 강승식, 장두성 (2006), 음성 인식 후처리를 위한 띄어쓰기 오류의 교정, **한국컴퓨터종합학술대회 논문집,** 33(1), pp.25-27.

[4] 최재혁 (1997), 양방향 최장일치법을 이용한 한국어 띄어쓰기 자동 교정 시스템, **제9회 한글 및 한국어 정보처리 학술대회 논문집,** pp.145-151.

[5] 강승식 (2000), 한글 문장의 자동 띄어쓰기를 위한 어절 블록 양방향 알고리즘, **정보과학회논문지: 소프트웨어 및 응용, 27**(4), pp.441-447.

[6] 이호준, 박종철 (2002), 음절단위 결합범주문법을 이용한 한국어 문장의 자동 띄어쓰기**, 제14회 한글 및 한국어 정보처리 학술대회 논문집,** pp.47-54.

[7] 심광섭 (1996), 음절간 상호 정보를 이용한 한국어 자동 띄어쓰기, **정보과학회 논문지(B), 23**(9), pp.991-1000.

[8] 태윤식, 박성배, 이상조, 박세영 (2006), 자기 조직화 n-gram모델을 이용한 자동 띄어쓰기, **제18회 한글 및 한국어 정보처리 학술대회 논문집,** pp.125-132.

[9] 이도길, 이상주, 임희석, 임해창 (2003), 한글 문장의 자동 띄어쓰기를 위한 두 가지 통계적 모델, **정보과학회논문지: 소프트웨어 및 응용 30(3,4)**, pp.358-371.

[10] Gumwon Hong, Jeong-Hoon Lee, Young-In Song, Do-Gil Lee, Hae-Chang Rim (2009), Utilizing the Web for Automatic Word Spacing, IEICE Trans. Inf & Syst, Vol. E92-D, No.12, pp.2553-2556.

[11] John Lafferty, Andrew McCallum and Fernando Pereira (2001), Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data, Proceedings of the 18th International Conference on Machine Learning, pp.282-289.

[12] Fei Sha and Fernando Pereira (2003), Shallow Parsing with Conditional Random Fields, Proceedings of HLT-NAACL 2003, pp.134-141.

[13] Hanna M. Wallach (2004), Conditional Random Fields: An Introduction, CIS Technical Report MS-CIS-04-21, University of Pennsylvania.

[14] Fuchun Peng, Fangfang Feng, Andrew McCallum (2004), Chinese Segmentation and New Word Detection using Conditional Random Fields, Proceedings of the 20th International Conference on Computational Linguistics, pp.562-568.

[15] 김승원, 김병창, 정민우, 이근배 (2005), CRF를 이용한 한국어 운율 경계 추정, **제17회 한글 및 한국어 정보처리 학술대회 논문집**, pp.134-138.

[16] 오진영, 차정원 (2009), 엔트로피 지도 CRF를 이용한 한국어 어절 구문태그 예측, **정보과학회논문지: 컴퓨팅의 실제 및 레터, 15**(5), pp.395-399.

[17] http://crfpp.sourceforge.net/, “CRF++: Yet Another CRF toolkit.”

1 차원고접수 : 2011. 4. 28
2 차원고접수 : 2011. 6. 14
최종게재승인 : 2011. 6. 22

(*Abstract*)

Automatic Word Spacing based on Conditional Random Fields

Kwangseob Shim

School of Information Technology

Sungshin Women's University

In this paper, an automatic word spacing system is proposed, which assumes sentences with no spaces between the words and segments them into proper words. Segmentation is regarded as a labeling problem in that segmentation can be done by attaching appropriate labels to each syllables of the given sentences. The system is based on Conditional Random Fields, which were reported to show excellent performance in labeling problems. The system is trained with a corpus of 1.12 million syllables, and evaluated with 2,114 sentences, 93 thousand syllables. The best results obtained are 98.84% of syllable-based accuracy and 95.99% of word-based accuracy.

Key words : Automatic Word Spacing, Conditional Random Fields, CRFs, Labeling Problem