DOI: 10.3745/KIPSTB.2009.16-B.4.333

저사양 기기를 위한 한국어 자동 띄어쓰기 시스템

송 영 길[†] · 김 학 수[††]

요 약

대부분의 기존 자동 띄어쓰기 시스템들은 많은 시스템 자원을 필요로 하기 때문에 상대적으로 낮은 컴퓨팅 파워를 가진 모바일 기기에 사용하기에는 적합하지 않다. 본 논문에서는 저사양 모바일 기기에 맞도록 메모리 사용량이 적고 수치 계산이 단순한 자동 띄어쓰기 시스템을 제안한다. 제안 시스템은 통계 기반 시스템과 규칙 기반 시스템으로 구성된 2단계 모델이다. 메모리 사용량을 줄이기 위해서 통계 기반 시스템이 음절 유니그램 기반의 개량된 은닉 마코프 모델을 사용하여 띄어쓰기 오류를 1차로 수정한다. 다음으로 정밀도 향상을 위해서 규칙 기반 시스템이 음절 바이그램 이상의 어휘 규칙을 이용하여 잘못 수정된 띄어쓰기 오류를 재보정한다. 실험 결과에 따르면 제안시스템은 1MB를 조금 넘는 메모리 사용하면서도 94.14%라는 비교적 높은 정밀도를 보였다.

키워드 : 자동 띄어쓰기, 저사양, 모바일 기기, 2단계 모델

An Automatic Korean Word Spacing System for Devices with Low Computing Power

Yeongkil Song[†] · Harksoo Kim[††]

ABSTRACT

Most of the previous automatic word spacing systems are not suitable to use for mobile devices with relatively low computing powers because they require many system resources. We propose an automatic word spacing system that requires reasonable memory usage and simple numerical computations for mobile devices with low computing powers. The proposed system is a two step model that consists of a statistical system and a rule-based system. To reduce the memory usage, the statistical system first corrects word spacing errors by using a modified hidden Markov model based on character unigrams. Then, to increase the accuracy, the rule-based system re-corrects miscorrected word spaces by using lexical rules based on character bigrams or more. In the experiments, the proposed system showed relatively high accuracy of 94.14% in spite of small memory usage of about 1MB.

Keywords : Automatic Word Spacing, Low Computing Power, Mobile Device, Two Step Model

1. 서 론

고성능 모바일 기기의 등장으로 쾌적한 모바일 컴퓨팅 환경이 제공되면서 기존에는 사용할 수 없었던 유용한 응용 프로그램의 활용이 가능해졌다. 그러나 대부분의 응용 프로그램들은 복잡한 UI(user interface)를 가지고 있어서 사용자들의 정보 접근성 및 활용성이 좋지 못하다. 이러한 복잡한 UI 문제를 해결하기 위한 방법으로 SMS(short message service)로부터의 정보 추출[1], 내부 콘텐츠 검색[2] 등의 다양한 자연어처리 응용 프로그램들이 개발되고 있다. 그러나 기존의 자연어처리 응용 프로그램들은 입력문장의 띄어쓰기가 정확하다는 가정 하에 만들어졌지만 현실은 그렇지 못하다. 실제 모바일 환경에서는 사용자들이 입출력 장치의 불편함으로 인하여 거의 띄어쓰기를 수행하지 않는 것으로 알려져 있다[3]. 그러므로 본격적인 자연어처리에 앞서 자동 띄어쓰기를 수행하는 것이 필요하다. 그러나 기존의 한글 자동 띄어쓰기 시스템들은 서버급 컴퓨터를 대상으로 개발되었기 때문에 상대적으로 성능이 떨어지는 모바일 기기에 그대로 적용하기에는 무리가 따른다. 특히 모바일 기기는 제한된 메모리를 가지고 있기 때문에 기존의 많은 시스템에서 사용하는 바이그램(bigram) 이상의 음절 자질을 그대로 사용하는 것은 매우 어렵다. 본 논문에서는 모바일 기기와 같이 저사양(low computing power) 시스템을 대상으로 하여 기존의 모델보다 경량화된 띄어쓰기 시스템을 제안한다.

※ 이 연구는 삼성전자 산학협력 과제의 지원을 받아 수행되었음. 또한, 이 논문은 2008년도 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2008 -313-D00907).
† 준 회 원 : 강원대학교 컴퓨터정보통신공학전공 석사과정
†† 정 회 원 : 강원대학교 컴퓨터정보통신공학전공 교수
논문접수 : 2009년 4월 17일
수 정 일 : 1차 2009년 6월 25일

2. 관련연구

한국어 자동띄어쓰기에 대한 기존 연구는 분석적인 방법[4,5]과 통계적인 방법[6-10]으로 나눌 수 있다. 분석적인 방법은 사전 정보, 어미/조사 정보 등의 어휘 지식을 사용하여 최장일치, 최단일치, 형태소 분석규칙, 띄어쓰기 오류 유형 등의 휴리스틱을 이용하여 띄어쓰기를 하는 방법이다. 이러한 방법을 사용하기 위해서는 여러 가지 언어학적 자원이 필요하고 그것을 구축 관리하는데 많은 비용이 든다. 또한 구축되지 않은 새로운 패턴에 대해서는 매우 낮은 정확도를 보인다는 단점이 있다. 통계적인 방법은 대량의 말뭉치로부터 인접한 두 음절의 띄어 쓰거나 붙여 쓸 확률을 학습하여 띄어쓰기 오류를 교정하는 것이다. 이 방법은 대량의 원시 말뭉치로부터 자동으로 음절 정보를 얻어 사용하므로 별도의 지식 구축비용이 들지 않고 미등록어에 대해서도 높은 정확도를 보인다는 장점이 있다. 하지만 신뢰성 있는 통계 정보를 얻기 위해서는 대용량의 학습 말뭉치가 필요하게 되고, 학습된 말뭉치의 유형에 따라 띄어쓰기 결과가 달라진다는 단점이 있다. 최근의 연구는 주로 통계적인 방법을 사용하거나 통계적인 방법과 분석적인 방법을 결합하는 방향으로 이루어지고 있다. 통계적인 방법에서 확률 정보를 얻기 위해 주로 사용하는 방법은 띄어쓰기 대상 지점 주위 n개의 음절을 학습데이터로 사용하는 n-그램(n-gram) 방법이다. n-그램 방법을 통해서 추출된 통계 정보는 n의 값이 클수록 높은 신뢰도를 보이지만 그에 따라 메모리 사용량이 기하급수적으로 늘어나게 되고, 데이터 희소성 문제가 커진다는 단점이 있다. 이러한 문제는 제한된 메모리 사용공간을 가진 모바일 기기에서는 큰 문제로 작용한다. 이러한 문제를 해결하기 위해 본 논문에서는 상대적으로 적은 메모리 용량을 사용하는 유니그램(unigram) 통계 기법을 기반으로 하면서 바이그램 이상의 오류 교정 규칙을 결합한 경량형 한국어 띄어쓰기 교정 시스템을 제안한다.

3. 경량형 띄어쓰기 시스템

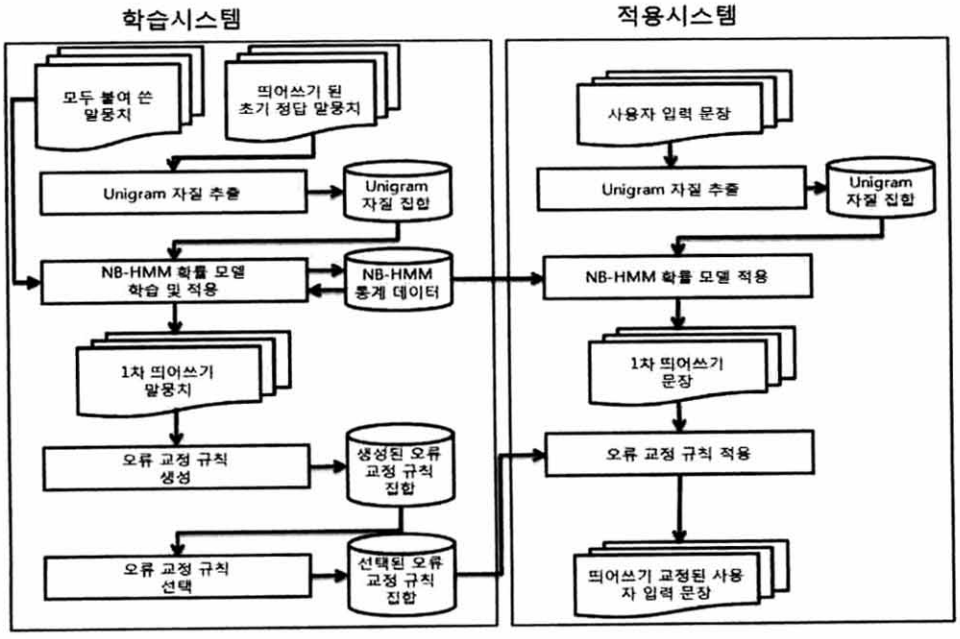
3.1 시스템 개요

제안 시스템은 (그림 1)과 같이 학습 시스템과 적용 시스템으로 구성된다. 학습 시스템은 유니그램 통계데이터와 바이그램 이상의 교정 규칙을 생성한다. 1단계인 유니그램 통계 데이터는 초기 띄어쓰기 정답 말뭉치를 바탕으로 3.2절의 통계모델로 학습한다. 2단계인 바이그램 이상의 교정 규칙을 생성하기 위해서, 유니그램 통계 학습에 사용된 말뭉치를 통계 학습된 모델을 이용하여 1차 띄어쓰기를 한다. 이렇게 생성된 결과와 정답 말뭉치를 비교하여 3.3절의 과정으로 규칙을 생성한다. 이렇게 생성된 통계데이터와 규칙 사전을 이용하여 적용 시스템은 2단계에 걸쳐 입력된 문장을 띄어쓰기 한다. 1단계로 유니그램 자질의 NB-HMM 확률 모델을 이용하여 자동 띄어쓰기를 수행하고, 2단계에서는 바이그램 이상의 오류 교정 규칙을 적용하여 1단계에서 띄어쓰기에 실패한 것들을 수정한다.

3.2 통계 기반의 띄어쓰기 선교정

문장을 구성하는 각각의 음절에 대하여 띄어쓰기 여부를 찾는 문제는 수식 (1)과 같이 각각의 음절에 대하여 띄어쓰기 여부를 태깅하는 문제로 볼 수 있다. 수식 (1)에서 $W_{1,n}$은 n개의 음절열 $w_1 w_2 \cdots w_n$이고, $C_{1,n}$은 띄어쓰기 태그열 $c_1 c_2 \cdots c_n$을 이다.

$$C(W_{1,n}) = \arg max_{C_{1,n}} P(C_{1,n} | W_{1,n}) \quad (1)$$



(그림 1) 제안 시스템 구조도

입력 문장을 구성하는 모든 음절열과 태그열을 고려하여 특정 음절의 띄어쓰기 정보를 확률적으로 계산하는 것은 데이터 희소 문제를 초래한다. 이러한 문제를 해결하기 위해 본 논문에서는 1차 마코프(Markov) 가정을 적용하여 수식 (1)을 수식 (2)와 같이 고쳐 쓴다.

$$C(W_{1,n}) = \arg max_{C_{1,n}} P(C_{1,n}|W_{1,n}) \approx \arg max_{C_{1,n}} \prod_{i=0}^{n} P(w_i|c_i)P(c_i|c_{i-1}) \quad (2)$$

수식 (2)에서 $P(w_i|c_i)$는 관측확률로 c_i가 나왔을 때 w_i일 확률이고 $P(c_i|c_{i-1})$는 전이 확률로 이전 태그가 c_{i-1}이었을 때 현재 태그가 c_i일 확률이다. 관측확률에서 w_i는 하나의 음절을 나타내기 때문에 그대로 사용하기에는 정보량이 많이 부족하다. 그러나 본 논문에서는 나이브 베이즈 분류(Naïve Bayesian Classification) 기법을 적용하여 수식 (2)의 관측확률을 수식(3)과 같이 확장한다.

$$P^*(w_i|c_i) = \frac{1}{Z}P(c_i)\prod_{j=1}^{f} P(w_{ij}|c_i) \quad (3)$$

수식 (3)에서 f는 자질의 수를 나타내고, Z는 값을 정규화하기 위한 $p^*(w_i|c_i)$의 최고값이다. 이렇게 변형된 수식 (3)을 수식 (2)에 반영하면 수식 (4)가 된다. 본문에서는 수식 (4)와 같이 주변 문맥을 고려하도록 변형된 히든 마코프 모델(HMM;Hidden Markov Model)을 NB-HMM(Naïve Bayesian Hidden Markov Model)이라 한다.

$$P(C_{1,n}, W_{1,n})^* = \prod_{i=1}^{n}\left(P(c_i|c_{i-1}) \cdot \frac{1}{Z}P(c_i)\prod_{j=1}^{f} P(w_{ij}|c_i)\right) \quad (4)$$

모바일 기기의 CPU는 개인용 컴퓨터의 CPU에 비해 연산속도가 느리고, 많은 기능이 빠져있다. PDA에서 많이 사용하는 XSCALE PXA270의 경우 부동소숫점(floating point) 연산을 위한 하드웨어가 없어 소숫점 연산을 소프트웨어적으로 구현하여 사용한다. 그러므로 모바일 기기를 대상으로 한 프로그램에서는 소숫점 연산이 없는 것이 유리하다. 그래서 계산량을 줄이기 위해 수식 (4)에 log연산을 하고, 부동소수점 연산을 없애기 위해 10^6을 곱하여 정수화한다. 이때 유효숫자는 소숫점 6째 자리로 한다. 최종적으로 얻은 수식은 수식 (5)와 같다.

$$P(C_{1,n}, W_{1,n})^* \approx \sum_{i=1}^{n}\left(\begin{array}{l}\lceil \log(P(c_i|c_{i-1}))\times 10^6 \rceil \\ + \lceil \log(P(c_i))\times 10^6 \rceil \\ + \sum_{j=1}^{f} \lceil \log(P(w_{ij}|c_i))\times 10^6 \rceil\end{array}\right) \quad (5)$$

수식 (5)에서 보는 것과 같이 제안 모델은 기존 HMM에 나이브 베이지안 기법을 적용하여 주위 문맥을 고려한다. 이와 같은 방법은 기존의 연구에서도 있었는데 대표적으로 수식 (6)으로 표현되는 CRFs(Conditional Random Fields) [11,13]와 수식 (7)로 표현되는 MEMMs(Maximum Entropy Markov Models)[12,13]이 있다.

$$P(C_{1,n}|W_{1,n}) = \frac{1}{Z(W)}exp\left(\sum_{i=1}^{n}\sum_{j}\lambda_j f_j(c_i, c_{i-1}, W_i^?)\right) \quad (6)$$

$$P(C_{1,n}|W_{1,n}) = \prod_{i=1}^{n}\frac{1}{Z(c_{i-1}, w_i)}exp\left(\sum_{j}\lambda_j f_j(c_i, c_{i-1}, w_i)\right) \quad (7)$$

이 방법들은 높은 성능을 보이지만 비교적 복잡한 수식을 사용하기 때문에 컴퓨팅 파워가 비교적 낮은 모바일 기기에 사용하기에는 무리가 따른다. 본 논문에서 제안하는 모델은 HMM과 나이브 베이지안 기법을 단순히 결합하여 CRFs와 MEMMs의 장점을 HMM에 적용하였다. 비록 기존의 두 모델보다 성능은 조금 떨어지지만 보다 간단한 연산으로 사용할 수 있기 때문에 컴퓨팅 파워가 낮은 모바일 기기에서 사용하기에 적합하다고 생각된다.

제안된 통계 모델은 메모리 사용을 줄이기 위해서 음절 유니그램 자질을 사용한다. <표 1>은 통계 기반의 띄어쓰기 선교정을 위해 본 논문에서 사용한 자질의 구성을 보여준다. 유니그램 자질에서도 데이터 희소성 문제가 발생할 수 있다. 본 논문에서는 이때 발생할 수 있는 데이터 희소성 문제를 한글만으로 제안하기 위해 한글 및 문법적 역할을 하는 6가지 특수문자(" / ' / ? / ! / , / .)를 제외한 영문자, 숫자, 기타 특수문자의 열을 각각 *'EN'*, *'NU'*, *'SY'*로 변경하여 사용한다. 한글과 제외된 특수문자가 아닌 경우로 시작되어 한글이 나오기 직전까지를 하나의 묶음으로 사용하고 시작과 끝의 기호를 사용하여 표현한다. 예를 들어 '그는 *19c Berlin*에서'에서 *'19c Berlin'*은 하나의 묶음으로 표현되고 그 시작은 숫자이므로 'NU', 끝은 영문자이므로 *'EN'*이 되어 '그는 *NUEN*에서'가 된다. 한글 자질의 데이터 희소성 문제는 라플라스 스무딩(Laplace Smoothing)기법을 적용하였다. 1회 이상 출현한 자질에 대해서는 1을 더한 값을 빈

〈표 1〉 통계 기반의 띄어쓰기 선교정을 위한 자질의 구성

자질	설명
w_{-3}	띄어쓰기 교정 지점을 기준으로 앞 3번째 음절 또는 일반화 심볼
w_{-2}	띄어쓰기 교정 지점을 기준으로 앞 2번째 음절 또는 일반화 심볼
w_{-1}	띄어쓰기 교정 지점을 기준으로 앞 1번째 음절 또는 일반화 심볼
w_{+1}	띄어쓰기 교정 지점을 기준으로 뒤 1번째 음절 또3.3.3는 일반화 심볼
w_{+2}	띄어쓰기 교정 지점을 기준으로 뒤 2번째 음절 또는 일반화 심볼

〈표 2〉 음절 빈도 예

음절	사용 위치별 빈도									
	w_{-3}		w_{-2}		w_{-1}		w_{+1}		w_{+2}	
	띄	붙	띄	붙	띄	붙	띄	붙	띄	붙
<ST>	27,682	116,817	6,248	66,002	1	1	1	1	59	72,191
의	11,380	41,222	4,277	48,359	44,664	7,976	3,983	48,532	7,083	44,933

구분	이전띄어쓰기	
	띄	붙
띄	63,927	771,955
붙	771,956	1,176,306

$$
\begin{aligned}
P(1,나) &\approx \left(\begin{array}{l}\lceil \log(P(1|0))\times 10^6 \rceil + \lceil \log(P(1))\times 10^6 \rceil \\ +\left(\begin{array}{l}\lceil \log(P(<ST>_{-3}|1))\times 10^6 \rceil + \lceil \log(P(<ST>_{-2}|1))\times 10^6 \rceil + \\ \lceil \log(P(나_{-1}|1))\times 10^6 \rceil + \lceil \log(P(의_{+1}|1))\times 10^6 \rceil + \\ \lceil \log(P(<ST>_{+2}|1))\times 10^6 \rceil\end{array}\right)\end{array}\right) \\
&= \left(\begin{array}{l}(-402,055)+(-522,547) \\ +\left(\begin{array}{l}(-1,479,948)+(-2,126,404)+(5,922,145) \\ +(-2,321,935)+(4,151,293)\end{array}\right)\end{array}\right) \\
&= -16,926,326 \\
P(0,나) &\approx \left(\begin{array}{l}\lceil \log(P(0|0))\times 10^6 \rceil + \lceil \log(P(0))\times 10^6 \rceil \\ +\left(\begin{array}{l}\lceil \log(P(<ST>_{-3}|0))\times 10^6 \rceil + \lceil \log(P(<ST>_{-2}|0))\times 10^6 \rceil + \\ \lceil \log(P(나_{-1}|0))\times 10^6 \rceil + \lceil \log(P(의_{+1}|0))\times 10^6 \rceil + \\ \lceil \log(P(<ST>_{+2}|0))\times 10^6 \rceil\end{array}\right)\end{array}\right) \\
&= \left(\begin{array}{l}(-219,127)+(-155,044) \\ +\left(\begin{array}{l}(-1,222,141)+(-1,470,090)+(6,289,647) \\ +(-1,603,619)+(1,431,164)\end{array}\right)\end{array}\right) \\
&= -12,390,834
\end{aligned}
$$

(그림 2) '나의'에 대한 계산 결과

도로 사용하고, 존재하지 않은 자질은 1회 출현한 것으로 본다.

이와 같은 방법으로 생성된 음절 빈도가 <표 2>와 같을 때 '나의'라는 문장이 들어왔다고 하면 (그림 2)와 같이 계산이 이루어진다. <표 2>에서 '<ST>'는 문장의 시작과 끝을 나타내는 기호이다.

'나의'에는 띄어쓰기 후보가 총 1개('나'와 '의'의 사이)가 존재한다. (그림 2)는 수식 (5)를 이용하여 각각 '나'의 뒤에 띄어 쓰는 경우와 붙여 쓰는 경우를 계산한 것이다. $P(1,나)$보다 $P(0,나)$의 값이 크므로 '나'와 '의' 사이는 붙인다는 결과를 얻을 수 있다.

3.3 규칙 기반의 띄어쓰기 후교정

통계 기반의 띄어쓰기 선교정의 결과를 정답 문장과 비교하여 틀린 부분에 대한 규칙을 생성한다. 후교정 규칙은 <표 1>의 자질들을 바이그램 이상으로 묶은 총 4가지의 패턴 ($w_{-1}w_{+1}s$', '$w_{-2}w_{-1}w_{+1}s$', '$w_{-2}w_{-1}w_{+1}w_{+2}s$', '$w_{-3}w_{-2}w_{-1}w_{+1}w_{+2}s$')으로 구성된다. 여기서 s는 띄어쓰기 여부(1: 띄어 씀, 0: 붙임)이다. 예를 들어 '연구의목적은무엇인가.'에 대한 1차 띄어쓰기 결과가 '연구의목적은 무엇인가.'이고 정답이 '연구의목적은 무엇인가.'라면 오류구간 '의목'에 대하여 '의목1', '구의목1', '구의목적1', '연구의목적1'의 4가지 패턴을 생성한다. 이와 같은 방법으로 생성된 규칙들을 전체 말뭉치에서 해당 규칙이 맞는 경우($Positive(Rule)$)와 틀린 경우($Negative(Rule)$)의 빈도를 측정한다. 측정한 빈도를 이용하여 각각의 규칙의 신뢰도(confidence score)를 구하게 되는데, 여기서는 정보추출기법에서 추출 패턴의 점수를 계산하는 수식[14]을 수정한 수식 (8)을 사용하였다. Riloff[14]의 식은 추출 패턴의 신뢰도와 정확하게 추출한 빈도를 모두 고려한 식으로, 띄어쓰기 규칙의 신뢰도를 판단하는데도 용이하다.

$$
Score(Rule) = \frac{\dfrac{Positive(Rule)}{Positive(Rule)+Negative(Rule)} \times \log_2(Positive(Rule)+1)}{Max.Score} \quad (8)
$$

$$
where\ \ Positive(Rule) > k \times Negative(Rule)
$$

수식 (6)에서 $Positive(Rule)$은 해당 규칙을 1차 띄어쓰기 말뭉치에 적용하였을 때 옳게 수정된 띄어쓰기 수이고, $Negative(Rule)$은 잘못 수정된 띄어쓰기 수이다. $Max.Score$는 각 규칙들에 부여된 신뢰도 중에서 가장 큰 값으로 규칙 신뢰도를 정규화하는데 사용된다. k는 맞을 확률이 높은 규칙을 얻기 위해 오류 패턴을 발생 빈도를 이용한 필터링(filtering)을 위해 사용되는 상수이다. 규칙의 신뢰도를 계산할 때 동일한 규칙이 띄어쓰기와 붙여쓰기 모두의 규칙을 가지고 있다면 높은 점수를 가진 것만을 남긴다. 만약 점수가 같다면 두 규칙을 모두 없앤다. 계산된 신뢰도를 바탕으로 내림차순 정렬하여 상위 N개의 규칙을 선택하거나 일정 점수 이상의 규칙을 선택하는 방법으로 최종 규칙 사전을 구축한다. 본 논문에서는 상위 N개의 규칙을 선택하는 방법으로 실험을 진행하였다. 구축된 규칙은 통계기반의 선교정 후의 결과에 패턴 매칭의 방식으로 교정된다. 이때의 대상

은 각각의 띄어쓰기 구간이 되며, 최장규칙 우선으로 선택하게 된다. 예를 들어 선교정의 결과가 '연구의목적'라면 '연구', '구의', '의목', '목적'의 구간에 대해 후교정 규칙 사전을 검색한다. 만약 '의목' 구간의 띄어쓰기를 하고 후교정 규칙 사전에 '의목0', '구의목적1'의 두 가지 규칙이 있다면 '구의목적1'의 규칙을 우선적으로 선택하여 적용하여 '연구의 목적'의 결과를 나타내게 된다.

4. 실험 및 평가

4.1 실험 데이터 및 환경

<표 3>은 학습 및 실험에 사용한 말뭉치의 구성을 보여준다. 21세기 세종계획 원시 말뭉치는 10차 교차 검증(10-fold cross validation)방법으로 실험하였고 기존실험과의 비교를 위해 ETRI 품사 부착 말뭉치를 사용하였다.

성능 평가를 위해 수식(9)의 정밀도(accurcy), 수식 (10)의 재현율(recall), 수식 (11)의 정확률(precision)을 측정하였다. 예를 들어 '아버지가 방에 들어가신다.'와 같은 문장을 '아버지 가방에 들어 가 신다.'와 같이 띄어 썼다면, 정밀도는 $\frac{8}{11}$, 재현율은 $\frac{1}{2}$, 정확률은 $\frac{1}{3}$이 된다.

$$\text{정밀도} = \frac{\text{올바르게띄어쓴후보구간수}}{\text{전체띄어쓰기후보구간수}} \quad (9)$$

$$\text{재현율} = \frac{\text{올바르게띄어쓴어절수}}{\text{정답문서의어절수}} \quad (10)$$

$$\text{정확률} = \frac{\text{올바르게띄어쓴어절수}}{\text{시스템이출력한어절수}} \quad (11)$$

4.2 실험 결과

4.2.1 통계를 이용한 선교정 모델 실험

통계를 이용한 선교정 모델의 성능을 알아보기 위해서 대표적인 통계 모델인 CRFs와 비교하였다.

<표 4>는 일반 PC에서 CRFs와 NB-HMM을 이용하여 각각 모델을 구성하였을 때의 결과이다. CRFs와 NB-HMM의 자질은 제안 모델에서 사용한 자질을 동일하게 사용하였고, 동일한 학습 및 실험 말뭉치를 사용하였다. CRFs에 비해 NB-HMM은 정밀도와 재현율이 각각 1.86%, 10.18% 떨어졌다. 그러나 수행시간은 약 3배 빨랐고, 모델 크기는 약 45% 수준으로 속도와 용량 면에서 더 좋은 성능을 보였다.

<표 5>는 NB-HMM과 CRFs, MEMM의 연산을 모바일 기기(XSCALE PXA270 CPU, 51.26MB memory, Windows Mobile 5.0)와 일반 PC에서 수행하였을 때의 속도를 비교한 결과이다. 약 5만 문장의 테스트 데이터를 돌렸을 때를 가정하여 실험하였으며, 모델에서 자질의 값을 찾는 것을 고려하지 않고 계산 속도만을 측정하였다.

위 결과를 보면 일반 PC에서 NB-HMM이 CRFs와 MEMM에 비해 약 5~10배 빠르고, 모바일 기기에서는 약 70배 빠르다는 것을 알 수 있다. 모바일 기기에서 실험 했을 때가 더 큰 차이를 보이는 것으로 보아 모바일 기기에서의 부동소숫점 연산이 큰 영향을 미친다는 것을 알 수 있다.

4.2.2 규칙 기반의 후교정 모델 실험

<표 6>은 규칙 기반 후교정 모델의 성능을 보여준다. 통계 모델이 잘못 띄어쓴 구간의 수 323,808개 중에 215,574개를 찾아서 재현율이 66.57%이고 이 중 185,776개가 맞아서

〈표 3〉 사용된 말뭉치 종류

구분	말뭉치	문장 수 (개)
학습 및 실험	21세기 세종계획 원시 말뭉치	809,914
실험	ETRI 품사 부착 말뭉치	27,858

〈표 4〉 모델 성능 비교 (일반 PC)

확률모델	모델 성능(오픈테스트)				모델 크기(Bytes)	학습 말뭉치	실험 말뭉치
	정밀도	재현율	정확률	수행시간(ms/1문장)			
CRFs	91.14	84.65	85.86	0.78	595,618	21세기 세종계획 말뭉치	21세기 세종계획 말뭉치
NB-HMM	89.28	74.47	89.59	0.26	266,268		

〈표 5〉 모델별 계산 속도 간접 비교

확률 모델	모바일 기기		일반 PC	
	시간 (초)	비교(배)	시간 (초)	비교(배)
NB-HMM	2	1.0	0.19	1.0
CRFs	137	68.5	0.93	4.89
MEMM	141	70.5	1.81	9.53

〈표 6〉 규칙 기반 후교정 모델 결과($k=2.5$, 단위:개)

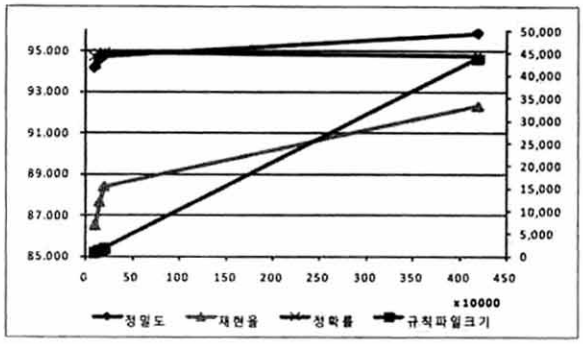
학습 말뭉치			21세기 세종계획 말뭉치
실험 말뭉치			21세기 세종계획 말뭉치
사용된 규칙 수			150,000
통계 모델이 틀린 경우			323,808
전체 적용 규칙	규칙을 적용한 수		1,219,335
	규칙이 맞은 경우		1,156,603
	규칙이 틀린 경우		62,732
	통계모델이 틀린 경우	규칙을 적용한 수	215,574
		규칙이 맞은 경우	185,776
		규칙이 틀린 경우	29,798
	통계모델이 맞은 경우	규칙을 적용한 수	1,003,761
		규칙이 맞은 경우	970,827
		규칙이 틀린 경우	32,934
재현율 (%)			66.57
정확률 (%)			86.18

정확률이 86.18% 이었다. 오류가 아닌 경우 1,003,761개 중에 규칙이 적용되어 틀린 경우가 32,934개로 3.28%가 발생하였다.

4.2.3 통합 모델 실험 (선교정 모델 + 후교정 모델)

<표 7>은 수식 (8)의 k값에 따른 통합 모델의 실험 결과이다. 필터링 기법을 사용하지 않았을 때($k=0.0$)보다 필터링 기법을 사용했을 때($k=2.5$) 정밀도가 약 0.38% 향상되었다. 본 논문에서는 최적의 값으로 $k=2.5$를 사용하였다.

<표 8>은 통합모델에서 사용하는 후교정 규칙 개수에 따른 성능 변화를 보여준다. <표 8>을 그래프로 나타내면 (그림 3)과 같다. (그림 3)의 왼쪽 y축은 정밀도, 재현율, 정확률의 단위이고, 오른쪽 y축은 규칙파일의 크기의 단위이다. 그래프의 x축은 규칙의 수이다. 그래프를 보면 정밀도와 재현율의 향상은 어느 정도 이상의 규칙에서는 변화폭이 줄어드는 것을 확인할 수 있다. 그리고 10만 개의 규칙을 사용했을 때와 120만 개의 규칙을 사용했을 때의 정밀도 차이가 1.72%, 재현율 차이가 5.79% 나지만 용량은 약 41MB 차이 나는 것을 확인할 수 있다. 그러므로 메모리 사용량을 고려했을 때 10만 개의 규칙 또는 20만개의 규칙을 사용하는 것이 효율적임을 알 수 있었다. 전체 규칙을 사용하는 경우 10만 개 규칙을 사용했을 때 보다 약 1.5배 느리지만 <표 4>의 CRFs를 사용했을 때보다 1.39배 빠르다.



(그림 3) 사용하는 규칙 개수에 따른 성능변화 그래프

<표 9>는 제안 시스템과 'Kang-2001[6]', 'Lee-2007[10]'의 성능을 비교한 것이다. 시스템간의 간접 비교를 위해 비교 대상 시스템과 같은 종류의 학습 말뭉치와 평가 말뭉치를 사용하였다.

Lee-2007[10](HMM)'과 제안 '시스템(NB-HMM)'을 비교했을 때 '제안 시스템(NB-HMM)'이 4.62% 높은 정밀도를 보였고 재현율과 정확률도 더 높은 성능을 보였다. 'Lee-2007[10](HMM)'은 음절 유니그램 자질을 관측확률로 사용

〈표 7〉 k값에 따른 통합 모델의 실험 결과 (패턴 발생 빈도를 이용한 필터링 실험)

k	규칙 수 (개)	성능(오픈테스트) (%)			학습 말뭉치	실험 말뭉치
		정밀도	재현율	정확률		
0.0	150,000	94.13	88.16	93.19	21세기 세종계획 말뭉치	21세기 세종계획 말뭉치
1.0	150,000	94.21	88.21	93.32		
2.0	150,000	94.51	87.91	94.56		
2.5	150,000	94.51	87.63	94.84		
3.0	150,000	94.49	87.44	95.00		

〈표 8〉 규칙 수에 따른 통합 모델의 실험 결과 ($k=2.5$, 일반 PC)

규칙 수 (개)	성능(오픈테스트) (%)			규칙파일 크기(Bytes)	속도 (ms/1문장)	학습 말뭉치	실험 말뭉치
	정밀도	재현율	정확률				
100,000	94.14	86.52	94.71	807,344	0.37	21세기 세종계획 말뭉치	21세기 세종계획 말뭉치
150,000	94.51	87.63	94.84	1,249,216	0.38		
200,000	94.74	88.38	94.90	1,699,375	0.40		
4,191,470	95.86	92.31	94.80	43,730,685	0.56		

〈표 9〉 성능 비교 결과

모델	성능(오픈테스트) (%)			학습 말뭉치	평가 말뭉치
	정밀도	재현율	정확률		
제안 시스템 (NB-HMM)	90.37	76.72	89.61	21세기 세종계획 말뭉치	ETRI 품사부착 말뭉치
제안 시스템 (NB-HMM, F=2.5, 10만개 규칙)	94.47	87.12	94.37		
제안 시스템 (NB-HMM, F=2.5, 15만개 규칙)	94.72	88.02	94.38		
제안 시스템 (NB-HMM, F=2.5, 20만개 규칙)	94.89	88.61	94.38		
Kang-2001[6] (음절 바이그램)	93.06	76.71	67.80		
Lee-2007[10] (HMM)	85.75	47.05	48.96		
Lee-2007[10] (음절 트라이그램)	97.48	87.89	89.79		

하는 일반 HMM 모델이다. 여기서 보는 바와 같이 일반 HMM에 나이브 베이즈 분류 기법을 결합한 모델이 일반 HMM 보다 좋은 성능을 보임을 알 수 있었다. <표 8>에서 가장 높은 성능을 보이는 'Lee-2007[10](음절 트라이그램)'은 '제안 시스템(NB-HMM, F=2.5, 10만개 규칙)' 보다 정밀도가 3.01%, 재현율이 0.77% 더 높은 성능을 보였다. 비교적 큰 성능 차이지만 음절 트라이그램 정보(NB-HMM으로 실험시 약 141MB였음)를 사용하기 때문에 모바일 기기에서 사용하기에는 부적합할 것으로 생각된다. 'Kang-2001[6](음절 바이그램)'은 '제안 시스템(NB-HMM, F=2.5, 10만개 규칙)'과 비교하여 1.41% 낮은 정밀도를 보였고, 재현율과 정확률도 낮은 성능을 보였다. 메모리 사용량을 고려했을 때 NB-HMM과 F=2.5의 10만 규칙을 사용한 제안 시스템이 1MB 내외의 메모리만을 사용하면서도 비교적 높은 성능과 효율을 보인다는 것을 알 수 있었다.

5. 결론 및 향후 과제

본 논문에서는 모바일 기기에 적합하도록 경량화된 2단계 한국어 자동 띄어쓰기 시스템을 제안하였다. 1단계에서 메모리 사용량을 줄이고 데이터 부족문제 해결을 위해서 음절 유니그램 자질 기반의 통계 모델을 이용하여 자동 띄어쓰기 선교정을 수행한다. 2단계에서는 바이그램 이상의 교정 규칙을 이용하여 1단계에서 잘못 교정한 것들을 후교정한다. 동일한 자질로 CRFs와 비교한 실험에서 제안 시스템의 선교정 통계모델의 정밀도가 0.25% 떨어졌지만 수행 속도는 2.87배 향상되었고, 모델 크기는 약 54% 수준으로 줄었다. 21세기 세종계획 원시말뭉치를 이용한 실험 결과에 따르면 제안 시스템은 1MB 내외의 메모리를 사용하면서 94.02%의 정밀도를 보여서 기존 시스템보다 모바일 기기에 더 적합함을 알 수 있었다.

향후 연구 과제는 다음과 같다. 선교정과정의 정밀도를 향상시키기 위해서 시스템 자원을 크게 필요로 하지 않는 추가 자질에 대한 연구를 진행할 예정이다. 그리고 후교정 과정의 정밀도 향상을 위해 후교정 규칙 신뢰도 측정 방법에 대한 연구를 진행할 예정이다.

참 고 문 헌

[1] Seon, C., Kim, H., Seo, J., "Information extraction using finite state automata and syllable n-grams in a mobile environment," Proceedings of the ACL-08: HLT Workshop on Mobile Language Processing, pp.13-18, 2008.

[2] Johnston, M., "Multimodal Voice Search for Interactive Media," Demo of the ACL-08: HLT Workshop on Mobile Language Processing (http://mobilenlpworkshop.org/Demos.html), 2008.

[3] 강승식, 장두성, "SMS 변형된 문자열의 자동 오류 교정 시스템", 정보과학회논문지 : 소프트웨어 및 응용 제35권 제6호, pp.386-391, 2008.

[4] 김계성, 이현주, 이상조, "연속 음절 문장에 대한 3단계 한국어 띄어쓰기 시스템", 정보과학회논문지(B) 제25권 제12호, pp.1938-1844, 1998.

[5] 강승식, "한글 문장의 자동 띄어쓰기를 위한 어절 블록 양방향 알고리즘", 정보과학회논문지:소프트웨어 및 응용 제27권 제4호, pp.441-447, 2000.

[6] 강승식., "음절 bigram를 이용한 띄어쓰기 오류의 자동교정", 음성과학회논문지, 제8권 제2호, pp.83-90, 2001.

[7] 최성자, 강미영, 허희근, 권혁철, "음절 N-Gram과 어절 통계 정보를 이용한 한국어 띄어쓰기 시스템", 한국정보과학회 언어공학연구회 학술발표 논문집, pp.47-53, 2003.

[8] 임동희, 전영진, 김형준, 강승식, "확장된 음절 바이그램을 이용한 자동 띄어쓰기 시스템", 한국정보과학회 언어공학연구회 학술발표 논문집, pp.189-193, 2005.

[9] 태윤식, 박성배, 이상조, 박세영, "자기 조직화 n-gram모델을 이용한 자동 띄어쓰기", 한국정보과학회 언어공학연구회 학술발표 논문집, pp.125-132, 2006.

[10] Lee, D., Rim, H., and Yook, D., "Automatic word spacing using probabilistic models based on character n-grams," IEEE Intelligent Systems, Vol.22 No.1, pp.28-35, 2007.

[11] Lafferty, J., McCallum, A., Pereira, F., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," Proceedings of ICML 2001, pp.282-289, 2001.
[12] McCallum, A., Freitag, D., Pereira, F., "Maximum entropy Markov models for information extraction and segmentation," Intl. Conf. on Machine Learning, pp.591-598, 2000.
[13] http://www.cs.brandeis.edu/~cs114/Spring2006/slides/CRFs_MEMMs.pdf (2009. 6. 16 방문)
[14] Riloff, E., Jones, R., "Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping," Proceedings of the 16th National Conference on Artificial Intelligence, 1999.

송 영 길

E-Mail : nlpyksong@kangwon.ac.kr
2008년 강원대학교 컴퓨터학부(공학사)
2008년~현 재 강원대학교 컴퓨터정보통신공학전공 석사과정
관심분야 : 한글 자동 띄어쓰기, 형태소 분석기, 사용자 모델링

김 학 수

E-Mail : nlpdrkim@kangwon.ac.kr
1996년 건국대학교 전자계산학과(공학사)
1998년 서강대학교 컴퓨터학과(공학석사)
2003년 서강대학교 컴퓨터학과(공학박사)
2004년~2005년 CIIR in UMass, Amherst (박사후연구원)
2005년~2006년 한국전자통신연구원(선임연구원)
2006년~현 재 강원대학교 컴퓨터정보통신공학전공 교수
관심분야 : 자연어처리, 대화시스템, 정보검색, 질의응답시스템