한국어 경량형 띄어쓰기 교정 시스템의 구현

송영길† · 김학수††

요 약

본 논문에서는 기존의 규칙 기반 방법과 통계 기반 방법의 장점을 취하면서도 메모리 사용량이 적은 한국어 띄어쓰기 교정 시스템을 제안한다. 또한 철자 오류와 조사 생략이 빈번히 발생하는 모바일 구어체에 강건하도록 모델을 학습시키기 위해서 일반 구어체 말뭉치로부터 가상의 구어체 말뭉치를 자동으로 구축하는 방법을 제안한다. 제안 시스템은 새로운 음절 패턴에 대한 적용 범위를 증가시키기 위해서 음절 유니그램 통계 정보를 이용하며, 정밀도 향상을 위해서 음절 바이그램 이상의 오류 교정 규칙을 이용한다. 가상의 모바일 구어체 문장에 대한 실험 결과에 따르면 제안 시스템은 1MB 내외의 적은 메모리를 사용하면서도 92.10%(일반 구어체 말뭉치에서 93.80%, 일반 균형 말뭉치에서 94.07%)라는 비교적 높은 정밀도를 보였다.

주제어 : 한국어 띄어쓰기 교정, 경량형 모델, 가상 모바일 구어체 말뭉치

An Implementation of a Lightweight Spacing-Error Correction System for Korean

Yeong-Kil Song† · Hark-Soo Kim††

ABSTRACT

We propose a Korean spacing-error correction system that requires small memory usage although the proposed method is a mixture of rule-based and statistical methods. In addition, to train the proposed model to be robust in mobile colloquial sentences in which spelling errors and omissions of functional words are frequently occurred, we propose a method to automatically transform typical colloquial corpus to mobile colloquial corpus. The proposed system uses statistical information of syllable uni-grams in order to increase coverages on new syllable patterns. Then, the proposed system uses error correction rules of two or more grams of syllables in order to increase accuracies. In the experiments on fake mobile colloquial sentences, the proposed system showed relatively high accuracy of 92.10% (93.80% in typical colloquial corpus, 94.07% in typical balanced corpus) spite of small memory usage of about 1MB.

Keywords : Korean Spacing-error Correction, Lightweight Model, Fake Mobile Colloquial Corpus

† 준 회 원: 강원대학교 컴퓨터정보통신공학전공 석사과정
†† 정 회 원: 강원대학교 컴퓨터정보통신공학전공 교수(교신저자)
논문접수: 2009년 01월 06일, 심사완료: 2009년 03월 23일
* 본 연구는 부분적으로 삼성전자 산학협력 과제의 지원을 받아 수행되었음. 또한 본 논문은 부분적으로 2008년도 정부(교육과학기술부)의 재원으로 한국학술진흥재단의 지원을 받아 수행된 연구임(KRF-2008 -313-D00907).

1. 서 론

웹기반 전자상거래 시스템의 등장으로 사용자와 사업자 모두의 요구를 충족시킬 수 있는 효과적인 정보검색 도구의 필요성이 제기되고 있으며, 이러한 요구는 무선 환경으로 빠르게 확산되고 있다. 그러나 현재 상용화되어 있는 PDA(Personal Digital Assistants)나 휴대폰 단말기의 사용자 인터페이스는 복잡한 메뉴들로 이루어져 있기 때문에 편리성을 추구하는 사용자들의 요구를 충족시켜 주지 못하고 있다. 이러한 문제를 해결하기 위해서는 문자메시지에 포함된 일정 정보를 자동으로 추출하여 데이터베이스에 저장해주는 정보추출 시스템이나 모바일 장치 내부의 컨텐츠(contents)를 빠르게 접근할 수 있도록 도와주는 정보검색 시스템과 같은 다양한 자연어처리 응용 프로그램들의 개발이 필요하다. 그러나 붙여쓰기가 빈번한 모바일 환경에서 사용자 입력에 대한 띄어쓰기 교정이 되지 않는다면 형태소 분석을 비롯한 상위 단계의 모든 언어 분석이 매우 어려워지고, 그로 인하여 대부분의 자연어처리 응용 프로그램들의 개발이 현실적으로 불가능하게 된다. 그러므로 모바일 환경에서 자연어처리 응용 프로그램들을 개발하기 위해서는 띄어쓰기 교정 시스템의 개발이 선행되어야 한다.

자연어처리 응용 프로그램의 하나인 띄어쓰기 교정 시스템의 관점에서 살펴봤을 때, 모바일 환경은 두 가지 면에서 일반 PC(Personal Computer) 환경과 크게 다르다. 첫 번째로 모바일 기기의 성능이 PC와 비교했을 때 현저하게 낮다는 점을 들 수 있다. PDA나 휴대폰과 같은 모바일 기기들은 중앙처리장치의 속도, 연산 능력, 메모리 용량 등 여러 면에서 PC에 비해 낮은 컴퓨팅 파워(computing power)를 가지고 있다. 특히 매우 제한적인 가용 메모리는 대용량의 언어자원이 필요한 자연어처리 응용 프로그램 개발에 큰 약점으로 작용한다. 두 번째로 모바일 단말기의 입력 장치는 작고 불편하기 때문에 입력 문자열에 많은 철자 오류와 조사 생략 현상이 포함된다는 점을 들 수 있다. 이러한 특성은 모바일용 자연어처리 응용 프로그램을 개발하는데 있어서 기존의 언어 자원들(주로 문어체나 방송대본과 같은 일반적인 구어체 말뭉치들)을 이용하는 것을 어렵게 만든다.

본 논문에서는 모바일 기기용 한국어 띄어쓰기 교정 시스템을 개발할 때 필연적으로 발생하는 위에서 기술한 두 가지 문제(메모리 제약 문제와 언어자원 부족 문제)를 해결하는데 초점을 맞춘다. 먼저 가능한 한 적은 메모리를 사용하면서도 일정 수준 이상의 성능을 보장하기 위해서 2단계에 걸쳐서 띄어쓰기를 교정하는 새로운 형태의 하이브리드(hybrid) 모델을 제안한다. 다음으로 제안 모델을 학습시키는데 꼭 필요한 모바일 언어 말뭉치를 일반 구어체 말뭉치로부터 자동 구축하는 방법을 제안한다.

2. 관련 연구

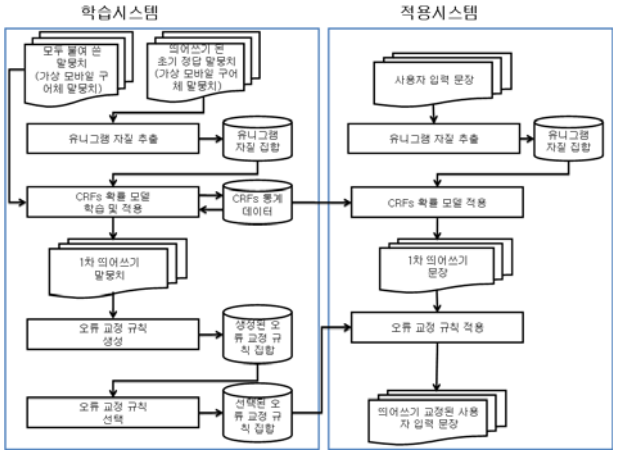
한국어 띄어쓰기 교정에 대한 기존의 연구는 분석적인 방법[1][2][3]과 통계적인 방법[4][5][6]으로 나눌 수 있다. 분석적인 방법은 형태소 분석 규칙이나 띄어쓰기 오류 유형 등의 휴리스틱(heuristic)을 이용하여 띄어쓰기 오류를 교정하는 것이다. 이 방법은 제한된 영역에서 매우 높은 정확률을 보이며, 튜닝(tuning)이 가능하다는 장점이 있다. 그러나 형태소 분석을 위해서 여러 가지 언어 자원이 필요하며, 그것을 구축하고 관리하는데 많은 비용이 든다는 단점이 있다. 또한 새로운 패턴에 대해서는 매우 낮은 정확도를 보인다는 단점이 있다. 통계적인 방법은 인접한 두 음절 주변의 n-그램(n-gram) 확률 정보를 바탕으로 띄어쓰기 오류를 교정하는 것이다. 이 방법은 대량의 원시 말뭉치로부터 자동으로 음절 정보를 얻어 사용하기 때문에 별도의 지식 구축비용이 들지 않고, 새로운 음절 패턴에 대해서도 비교적 강건하게 작동한다는 장점이 있다. 그러나 신뢰성이 높은 확률 정보를 얻기 위해서는 실제 적용 영역과 비슷한 특성을 보이는 대용량의 학습 말뭉치가 필요하다는 단점이 있다. 또한 이 방법은 n의 값

이 클수록 높은 신뢰성을 보이지만 더 많은 메모리를 필요로 한다는 단점이 있다. 현대 한국어 음절의 수는 약 10^4개이며, 이것을 바이그램(bi-gram)으로 저장한다면 약 400MB의 메모리가 필요하다. 여기에 부가적인 통계 정보를 추가하면 데이터의 크기는 더욱 커지게 된다. 그러므로 바이그램 이상의 통계 정보를 사용하는 기존의 방법들을 가용 메모리가 제한적인 모바일 기기에 그대로 적용하는 것은 사실상 불가능하다. 본 논문에서는 위에서 기술한 기존 연구 결과들을 바탕으로 분석적인 방법과 통계적인 방법의 장점을 살리면서도 메모리 사용량은 적은 모바일 기기용 띄어쓰기 교정 모델과 학습데이터 구축 방법을 제안한다.

3. 하이브리드 방법의 띄어쓰기 교정 시스템

3.1 제안 시스템 개요

<그림 1>은 본 논문에서 제안하는 모바일 기기용 띄어쓰기 교정 시스템의 전반적인 처리 과정을 보여주는 구조도이다.



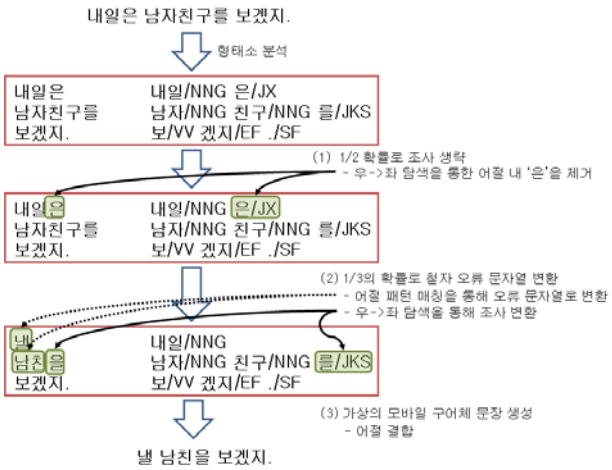
<그림 1> 제안 시스템의 구조도

<그림 1>에서 보는 것과 같이 제안 시스템은 학습 시스템과 적용 시스템으로 구성된다. 학습시스템은 가상의 모바일 구어체를 기반으로 음절 유니그램 통계 데이터와 음절 바이그램 이상의 교정 규칙을 생성한다. 적용 시스템은 2단계에 걸쳐서 입력된 문장의 띄어쓰기 오류를 교정한다. 1단계에서는 음절 유니그램 통계 데이터를 기반으로 하는 확률 모델을 이용하여 띄어쓰기 오류를 교정한다. 2단계에서는 음절 바이그램 이상의 교정 규칙을 적용하여 1단계에서 교정에 실패한 것들을 재교정한다.

3.2 가상 모바일 구어체 말뭉치 구축

일반적으로 기계 학습 시스템들은 학습 데이터가 실제 적용 데이터와 얼마나 유사하냐에 따라서 많은 성능 차이를 보인다. 그러므로 제안 시스템의 성능 향상을 위해서는 모바일 기기 사용자들이 직접 입력한 구어체 말뭉치를 대량으로 수집하여 학습하는 것이 가장 바람직하다. 그러나 연령별, 남녀별로 다른 현상을 보이는 모바일 구어체 균형 말뭉치를 대량으로 수집하는 것은 통신 사업자가 아니면 매우 어렵다. 또한 최근에 발효된 개인 정보 보호법으로 인하여 통신 사업자라고 할지라도 말뭉치 수집이 현실적으로 불가능하다. 이러한 문제를 해결하기 위해서 본 논문에서는 일반 구어체 말뭉치에 철자 오류와 조사 생략 현상을 임의로 발생시켜서 실제 모바일 구어체와 유사한 가상의 말뭉치를 만드는 방법을 제안한다. 가상의 모바일 구어체 말뭉치를 만들기 위해서 사용한 시스템 자원은 형태소 분석기, 철자 오류 사전, 그리고 조사 변환 규칙이다. 철자 오류 사전은 문화 관광부에서 발간한 통신 언어 어휘집을 바탕으로 구축하였다. 철자 오류 사전의 형태는 '파이팅 -> 팟팅'과 같이 올바른 문자열과 오류 문자열의 쌍으로 구성된다. 조사 변환 규칙은 양방향성을 가지며 조사 앞에 위치한 문자열의 종성 유무와 'ㄹ'종성 여부에 따라 'O 은 <-> X 는'이나 'ㄹ 로 <-> O 으로'와 같은 형태로 기술된다. 상기 예에서 'O 은 <-> X 는'은 종성이 있는 체언 다음에 오는 '은'이라는 조사는 종성이 없는 체언 다음에 올 경우에 '는'으로 바뀌며, 그 역으로도 해석이 가능하다는 뜻이다. 'ㄹ 로 <-> O 으로'는 'ㄹ' 종성으로 끝나는 체언 다음에 오는 '로'라는 조사는 종성이 있는 체언 다음에 올 경우에 '으로'로 바뀌며, 그 역으로도 해석이 가능하다는 뜻이다.

본 논문에서 제안하는 가상의 모바일 구어체 말뭉치를 구축하는 방법은 <그림 2>와 같다. 먼저, 원시 구어체 말뭉치에 대해서 형태소 분석을 수행한다. 두 번째로 각 어절의 형태소 분석 결과를 바탕으로 조사의 존재 여부를 판단한다. 만약 조사가 존재하면 1/2의 확률로 조사를 생략시킬 것인지 결정한다. 조사 생략이 결정되면 형태소 분석 결과에 나타난 조사 문자열을 해당 어절의 오른쪽에서부터 매칭(matching)하여 삭제한다. 조사 생략 여부에 대한 확률을 1/2로 잡은 것은 조사 생략 현상이 지식이나 연령, 성별 등에 영향을 받지 않으며 개인적인 성향에 따라 임의로 발생한다고 가정을 했기 때문이다. 세 번째로 특정 어절이 철자 오류 사전에 등재된 문자열을 포함하고 있다면 1/3의 확률로 해당 어절 내의 문자열을 오류 문자열로 변환한다. 그리고 오류 문자열로 변환된 어절에 조사가 존재하는지 여부를 검사한다. 만약 조사가 존재하면 조사 문자열을 해당 어절의 오른쪽에서부터 매칭하여 찾고, 조사 변환 규칙을 적용하여 올바른 조사로 바꾼다. 문자열 변환의 확률을 1/3로 잡은 것은 연령대를 청년, 중년, 노년으로 분류하고, 청년층이 주로 인위적인 오류 문자열을 많이 사용한다고 가정을 했기 때문이다. 마지막으로 변형된 어절들을 결합하여 문장을 구성한다.



<그림 2> 가상 구어체 문장 생성의 예

3.3 1단계: 통계 기반의 띄어쓰기 교정

제안 시스템은 새로운 음절 패턴에 대한 강건성을 높이기 위해서 통계 정보를 기반으로 1단계 띄어쓰기 교정을 수행한다. 1단계 띄어쓰기 교정을 위해서 본 논문에서 제안하는 통계 모델은 다음과 같다. n개의 음절로 구성된 문장 $X=x_1x_2x_3\cdots x_n$이 주어졌을 때, 각 음절 뒤의 띄어쓰기 정보 $Y=y_1y_2y_3\cdots y_n$을 찾는 문제는 수식 (1)과 같은 확률 모델로 정의될 수 있다.

$$Y^* = \arg max_Y P(Y|X) = \arg max_{y_{1,n}} P(y_{1,n}|x_{1,n}) \quad (1)$$

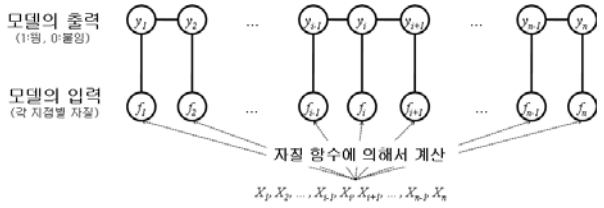
수식 (1)에서 $x_{1,n}$은 문장을 구성하는 n개의 음절 열(syllable sequence)을 의미하며, $y_{1,n}$은 각 음절 뒤의 띄어쓰기 여부를 나타내는 레이블 열(label sequence)을 의미한다. 입력 문장을 구성하는 모든 음절 열과 레이블 열을 고려하여 특정 음절의 띄어쓰기 정보를 확률적으로 계산하는 것은 데이터 희소성 때문에 매우 어렵다. 이러한 문제를 해결하기 위해서 본 논문에서는 1차 마코프(Markov) 가정을 적용하여 수식 (1)을 수식 (2)와 같이 변경한다.

$$T^* = \arg max_Y P(Y|X) \approx \arg max_{y_{1,n}} \prod_{i=0}^{n} P(y_i|x_i)P(y_i|y_{i-1}) \quad (2)$$

그리고 수식 (2)의 조건부 확률을 수식 (3)과 같은 CRFs(Conditional Random Fields)[7][8]를 이용하여 계산한다.

$$P(Y|X) = \frac{1}{Z(X)} exp\left(\sum_i \sum_k \mu_k s_k(y_i, X, i) + \sum_i \sum_k \lambda_k t_k(y_{i-1}, y_i, X, i)\right) \quad (3)$$

<그림 3>은 수식 (3)을 확률 그래프 형태로 표현한 것이다.



<그림 3> CRFs를 이용한 띄어쓰기 모델의 그래프 구조

<그림 3>에서 보는 것과 같이 CRFs는 다양한 입력 노드의 값이 주어졌을 때 지정된 출력 노드의 조건부 확률을 계산하기 위한 무방향성 그래프 모델이다. CRFs는 HMM(Hidden Markov Model)의 단점인 독립 가정을 완화시키는 효과가 있으며, MEMM(Maximum Entropy Markov Model)의 단점인 레이블 편향 문제(label bias problem)를 극복할 수 있다는 장점을 가지고 있어서 최근 자연어처리 분야에서 많이 사용되는 통계기반의 기계 학습 모델이다. 수식(3)에서 $Z(X)$는 입력 문장에 대한 정규화 요소이고, $s_k(y_i,X,i)$는 i번째 띄어쓰기 레이블에 대한 관찰 확률을 계산하기 위한 자질 함수로써 입력 문장에서 해당 자질 s_k가 나타나면 1을, 그렇지 않으면 0의 값을 가진다. $t_k(y_{i-1},y_i,X,i)$는 i-1번째 띄어쓰기 레이블과 i번째 띄어쓰기 레이블 사이의 전이 확률을 계산하기 위한 자질 함수로써 입력 문장에서 해당 자질 t_k가 나타나면 1을, 그렇지 않으면 0의 값을 가진다. <표 1>은 자질 함수의 입력이 되는 자질들을 보여준다.

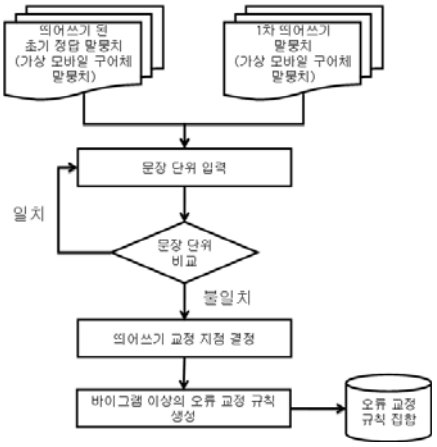
<표 1> 입력 자질의 구성

자질	설명
W_{-3}	띄어쓰기 지점을 기준 앞 3번째 음절 또는 일반화 심볼
W_{-2}	띄어쓰기 지점을 기준 앞 2번째 음절 또는 일반화 심볼
W_{-1}	띄어쓰기 지점을 기준 앞 1번째 음절 또는 일반화 심볼
W_{+1}	띄어쓰기 지점을 기준 뒤 1번째 음절 또는 일반화 심볼
W_{+2}	띄어쓰기 지점을 기준 뒤 2번째 음절 또는 일반화 심볼

<표 1>에서 보는 것과 같이 제안 모델은 유니그램 형태의 좌우 문맥 정보를 자질로 사용함으로써 메모리 사용량을 최소화한다. 또한 데이터 희소성 문제를 완화하기 위해서 한글과 5가지 특수문자(", ', ?, !, .)를 제외한 숫자, 영문자, 기타 문자를 *NU*, *EN*, *SY*라는 심볼(symbol)로 일반화하여 사용한다. 예를 들어, '그는 1910년 Berlin에서'라는 문자열이 입력되면 '1910'은 *NU*로, 'Berlin'은 *EN*으로 변환한 후 자질을 추출한다. 그러므로 'Berlin'과 '에' 사이의 띄어쓰기를 결정하기 위한 입력 자질은 앞 3음절과 뒤 2음절을 나타내는 'NU 년 EN 에 서'가 된다.

3.4 2단계: 규칙 기반의 띄어쓰기 재교정

본 논문에서 제안한 통계 기반의 1단계 띄어쓰기 교정 모델은 새로운 음절 패턴에 대해 높은 강건성을 보이지만 음절 유니그램이라는 제한된 문맥 정보를 바탕으로 하기 때문에 정확률이 낮다. 이러한 문제를 해결하기 위해서 제안 시스템은 바이그램 이상의 오류 보정 규칙을 자동 학습하여 통계 모델에 의해서 교정되지 못했거나 잘못 교정된 오류들을 2단계에서 재교정한다. <그림 4>는 2단계 교정을 위해서 본 논문에서 제안하는 오류 교정 규칙 생성 방법을 보여준다.



<그림 4> 오류 교정 규칙 생성 방법

오류 교정 규칙을 생성하는 방법을 자세히 살펴보면 다음과 같다. 먼저 띄어쓰기가 올바른 초기 정답 말뭉치를 모두 붙여 쓴다. 그리고 제안한 통계 모델을 적용하여 1차 띄어쓰기 말뭉치를 구성한다. 다음으로 초기 정답 말뭉치와 1차 띄어쓰기 말뭉치를 비교하여 재교정이 필요한 부분을 선정하고, '$W_{-1}W_{+1}$ -> 1/0', '$W_{-2}W_{-1}W_{+1}$ -> 1/0', '$W_{-2}W_{-1}W_{+1}W_{+2}$ -> 1/0', '$W_{-3}W_{-2}W_{-1}W_{+1}W_{+2}$ -> 1/0'와 같이 4개의 규칙을 추출한다. 규칙에서

W_{+n}/$_{-n}$는 띄어쓰기 교정 지점을 기준으로 상대적인 거리 n에 위치한 음절 나타내고, 1/0은 띄어쓰기 정보로 1은 띄어씀을, 0은 붙임을 의미한다. 예를 들어, “매우더운여름날농부는밭을멨다”라는 문장에 통계 모델을 적용하여 자동 띄어쓰기를 수행한 문장이 “매우 더운 여름날농 부는 밭을 멨다”라고 하자. 그리고 정답 문장이 “매우 더운 여름날 농부는 밭을 멨다”라고 하면, ‘날’과 ‘농’사이의 오류를 교정하기 위한 규칙으로 ‘날농 -> 1’, ‘날농부 -> 1’, ‘날농부는 -> 1’, ‘름날농부는 -> 1’이 생성된다. 그리고 ‘농’과 ‘부’ 사이의 오류를 교정하기 위한 규칙으로는 ‘부는 -> 0’, ‘농부는 -> 0’, ‘농부는밭 -> 0’, ‘날농부는밭-> 0’이 생성된다. 오류 교정 규칙이 모두 생성되면 각 규칙마다 수식 (4)를 적용하여 신뢰도(confidence score)를 계산하고, 상위 n개를 선택하여 띄어쓰기 오류를 재교정하는데 사용한다. 띄어쓰기 오류 재교정은 입력 문장과 규칙 문자열을 단순 매칭하고, 규칙에 부여된 띄어쓰기 정보를 실행하는 방식으로 이루어진다.

$$Score(Rule) = \frac{Positive(Rule)}{Positive(Rule)+Negative(Rule)} \times \frac{\log_2(Positive(Rule))}{MaxScore} \quad (4)$$

수식 (4)에서 $MaxScore$는 각 규칙들에 부여된 신뢰도 중에서 가장 큰 점수를 의미한다. 그리고 $Positive(Rule)$은 해당 규칙을 1차 띄어쓰기 말뭉치에 적용하여 옳게 수정된 띄어쓰기 개수이고, $Negative(Rule)$은 해당 규칙을 적용하여 잘못 수정된 띄어쓰기 개수이다.

4. 실험 및 평가

4.1 실험 데이터 및 환경

실험 대상 말뭉치로는 21세기 세종계획(http://www.sejong.or.kr) 원시 구어체 말뭉치 50만 문장을 대상으로 구축한 가상의 모바일 구어체 말뭉치를 사용하였다. 그리고 정확한 평가를 위해서 모든 실험에 대해서 10배 교차 검증을 수행하였다. 성능 측정 방법은 입력 문장을 모두 붙여 쓴 후에 음절과 음절 사이를 띄어쓰기 후보 구간으로 생각하여 정밀도(accuracy)를 측정하였다. 예를 들어, 5음절로 구성된 문장이 있다면 4개의 띄어쓰기 후보 구간이 존재하며 그 중 3개의 구간에서 띄거나 붙여 쓴 결과가 맞았다면, 정밀도는 3/4=0.75가 된다. 제안 시스템에서 사용한 CRFs의 학습 인자는 가우시안 값을 10으로, 반복 횟수를 30회로 설정하였다. 그리고 메모리 사용량을 고려하여 수식 (4)에서 $Positive(Rule)$이 $Negative(Rule)$의 2배 이상인 오류 교정 규칙들만을 선택하도록 하였다.

4.2 자질 선택을 위한 실험

메모리 사용량은 적으면서 CRFs에 적합한 자질을 선택하기 위해 본 논문에서는 가상 모바일 구어체 말뭉치에서 1만 문장(361,295 음절)을 임의로 선택하여 기존 연구[4][5][6][9]에서 많이 사용한 자질들에 대한 3가지 실험을 수행하였다. 첫 번째로 <표 2>에서 보는 것과 같이 n-그램 음절 자질 사용에 따른 모델 크기와 정밀도를 측정하였다. <표 2>에서 W_xW_y와 $W_xW_yW_z$는 각각 띄어쓰기 교정 지점을 기준으로 x, y 위치에 존재하는 음절 바이그램 자질과 x, y, z 위치에 존재하는 음절 트라이그램(tri-gram) 자질을 의미한다. 정밀도는 학습 데이터를 평가 데이터로 활용하여 계산한 것이다.

<표 2> 음절 자질 사용에 따른 모델 크기 및 정밀도

모델	자질 패턴	모델 크기 (MB)	정밀도 (%)
1	W_{-1} W_{+1} W_{+2}	0.36	87.54
2	W_{-3} W_{-2} W_{-1} W_{+1} W_{+2}	0.62	89.43
3	W_{-3} W_{-2} W_{-1} W_{+1} W_{+2} $W_{-3}W_{-2}$ $W_{-2}W_{-1}$ $W_{-1}W_{+1}$ $W_{+1}W_{+2}$	27.79	91.61
4	W_{-3} W_{-2} W_{-1} W_{+1} W_{+2} $W_{-3}W_{-2}$ $W_{-2}W_{-1}$ $W_{-1}W_{+1}$ $W_{+1}W_{+2}$ $W_{-3}W_{-2}W_{-1}$ $W_{-1}W_{+1}W_{+2}$	141.22	91.69

<표 2>에서 보는 것과 같이 CRFs는 다양한 음절 자질을 사용하면 할수록 더 좋은 성능을 보였다. 그러나 음절 바이그램 자질이나 음절 트라이그램 자질을 사용한 경우에 모델의 크기가 각각

28MB, 141MB 정도로 매우 크기 때문에 가용 메모리가 제한적인 모바일 기기에는 적합하지 않았다. 모델 (3)과 모델 (4) 사이의 정밀도 차이가 크지 않은 이유는 데이터 부족 문제에 기인한 것으로 보인다. 즉, 자질 선택 실험을 위해 사용한 1만 문장으로는 신뢰할 수 있는 수준의 음절 트라이그램 통계값을 얻을 수 없었기 때문인 것으로 생각된다. 이러한 실험 결과를 바탕으로 본 논문에서는 음절 바이그램과 음절 트라이그램을 자질에서 배제하였다.

두 번째로 <표 3>에서 보는 것과 같이 입력 문장에 존재하는 띄어쓰기 정보를 사용하는 것에 따른 모델 크기와 정밀도의 변화를 측정하였다. <표 3>에서 W_xSW_y는 띄어쓰기 교정 지점을 기준으로 x, y 위치에 존재하는 두 음절 사이의 띄어씀이나 붙임 정보를 의미한다. 모델 (3)은 올바르게 띄어쓰기된 실험 말뭉치의 각 어절 사이를 임의로 붙이거나 띄어서 재구성한 말뭉치를 이용하여 학습한 후, 재구성 말뭉치를 평가 데이터로 활용하여 정밀도를 계산한 것이다. 모델 (2)는 모델 (3)과 동일한 방법으로 학습한 후, 실험 말뭉치를 모두 붙여 쓴 것을 평가 데이터로 활용하여 정밀도를 계산한 것이다.

<표 3> 띄어쓰기 정보 사용에 따른 모델 크기와 정밀도

모델	자질 패턴	모델 크기(MB)	정밀도(%)
1	W_{-3} W_{-2} W_{-1} W_{+1} W_{+2}	0.62	89.43
2	W_{-3} W_{-2} W_{-1} W_{+1} W_{+2} $W_{-2}SW_{-1}$ $W_{+1}SW_{+2}$	0.63	88.27
3	W_{-3} W_{-2} W_{-1} W_{+1} W_{+2} $W_{-2}SW_{-1}$ $W_{+1}SW_{+2}$	0.63	91.45

<표 3>에서 보는 것과 같이 입력 문장의 띄어쓰기 정보는 정밀도 향상에 기여하였다. 그러나 모델 (2)의 정밀도에서 보듯이 입력 문장에 띄어쓰기가 거의 포함되지 않은 경우에 오히려 나쁜 결과를 보였다. 모바일 기기 사용자들은 좁은 입력 공간 때문에 띄어쓰기를 거의 하지 않는 경향이 있다. 그러므로 띄어쓰기 정보를 포함하는 것이 급격한 성능 하락을 초래할 수 있으므로 실제 응용에는 적합하지 않은 것으로 생각된다. 이러한 실험 결과를 바탕으로 본 논문에서는 띄어쓰기 정보 자질을 사용하지 않았다.

세 번째로 <표 4>에서 보는 것과 같이 품사 정보를 사용하는 것에 따른 모델 크기와 정밀도의 차이를 측정하였다. <표 4>에서 J, E, N은 조사, 어미, 의존명사 사전에 해당 문자열이 포함되었는지 여부를 나타내며, 두 자리 숫자로 표현된다. 앞의 숫자는 띄어쓰기 구간 앞 음절들(W_{-3},W_{-2}, W_{-1})이 해당 사전에 포함되어 있는지 여부를 나타내며, $W_{-3}W_{-2}W_{-1}$이 포함되어 있으면 3, $W_{-2}W_{-1}$이 포함되어 있으면 2, W_{-1}이 포함되어 있으며 1의 값을 가진다. 뒤의 숫자는 띄어쓰기 구간 뒤 음절들(W_{+1}, W_{+2})이 해당 사전에 포함되어 있는지 여부를 나타내며, $W_{+1}W_{+2}$가 포함되어 있으면 2, W_{+1}이 포함되어 있으면 1의 값을 가진다. 품사 정보로 조사, 어미, 의존명사만을 사용한 이유는 해당 품사의 리스트는 비교적 적은 수의 닫힌 집합이기 때문이다.

<표 4> 품사 정보 사용에 따른 모델 크기와 정밀도

모델	자질 패턴	모델 크기 (MB)	정밀도 (%)
1	W_{-3} W_{-2} W_{-1} W_{+1} W_{+2}	0.62	89.43
2	W_{-3} W_{-2} W_{-1} W_{+1} W_{+2} J E N	0.62	89.29

<표 4>에서 보는 것과 같이 품사 정보는 성능 향상에 크게 기여하지 못했다. 이러한 실험 결과를 바탕으로 본 논문에서는 품사 정보 자질을 사용하지 않았다. 결과적으로 본 논문에서는 위에서 기술한 3가지 실험 결과를 바탕으로 <표 1>과 같은 자질을 선정하여 사용하였다.

4.3 성능 측정을 위한 실험

제안 시스템의 성능 평가를 위해서 2가지 실험을 진행하였다. 첫 번째로 제안 시스템의 경량화 정도에 따른 성능 평가를 진행하였다. <표 5>는 교정 신뢰도에 따른 정밀도의 차이를 보여준다. <표 5>에서 1.0, 2.0, 4.0은 수식 (4)의 최소값으로 신뢰도가 해당 점수 이상인 오류 교정 규칙들만을 선택하여 띄어쓰기 재교정에 사용한 모델을 의미한다.

<표 5> 교정 신뢰도에 따른 제안 시스템의 정밀도

평가 방법	클로즈 테스트(closed test)			
모델	CRFs	교정 신뢰도		
		1.0	2.0	4.0
정밀도 (%)	89.44	96.75	94.08	92.24
평가 방법	오픈 테스트(open test)			
모델	CRFs	교정 신뢰도		
		1.0	2.0	4.0
정밀도 (%)	89.43	93.84	93.35	92.10

<표 5>에서 보는 것과 같이 최소 신뢰도가 1.0인 경우에 가장 높은 정밀도를 보였다. 그러나 최소 신뢰도에 따른 메모리 사용량을 측정한 결과, 1.0인 경우에 44.8MB, 2.0인 경우에 4.48MB, 4.0인 경우에 0.68MB로 신뢰도가 2.0인 경우와 4.0인 경우가 모바일 기기에 적합한 경량형 모델이었다. 특히 신뢰도가 1.0인 경우와 2.0인 경우를 비교했을 때, 오픈 테스트에서 거의 성능 차이가 없었다. 이러한 실험 결과를 바탕으로 모바일 기기의 가용 메모리 용량이 1MB 내외일 경우에 교정 신뢰도 4.0인 모델이 적당하며, 그 이상일 경우에는 교정 신뢰도 2.0인 모델이 가장 적합하다는 것을 알 수 있었다. <표 6>은 제안 시스템과 대표적인 기존 시스템들('Lee-2007'[9], 'Kang-2001'[10])의 성능을 비교한 것이다.

<표 6> 성능 비교 결과

모델	오픈 테스트 정밀도(%)	학습 말뭉치	평가 말뭉치
제안 시스템 (CRFs)	91.14	가상 모바일 구어체 말뭉치	ETRI 품사 부착 말뭉치
제안 시스템 (CRFs+교정신뢰도 2.0)	95.04		
제안 시스템 (CRFs+교정신뢰도 4.0)	94.07		
Kang-2001 (음절 바이그램)	93.06	21세기 세종계획 말뭉치	
Lee-2007 (HMM-음절 유니그램)	91.02		
Lee-2007 (HMM-음절 트라이그램)	97.48		
제안 시스템 (CRFs)	89.43	가상 모바일 구어체 말뭉치	가상 모바일 구어체 말뭉치
제안 시스템 (CRFs+교정신뢰도 2.0)	93.35		
제안 시스템 (CRFs+교정신뢰도 4.0)	92.10		

'제안 시스템 (CRFs)'는 동일한 평가 말뭉치에서 비슷한 입력 자질을 사용하는 'Lee-2007 (HMM-음절 유니그램)'과 비교했을 때, 0.71% 낮은 정밀도를 보였다. 이것은 철자 오류와 조사 생략이 포함된 가상의 모바일 구어체 말뭉치로 학습한 것에 기인한 것으로 생각된다. '제안 시스템 (CRFs)'를 일반 구어체 말뭉치로 학습했을 경우에는 'Lee-2007 (HMM-음절 유니그램)'보다 다소 높은 91.74%의 정밀도를 보였다. 'Lee-2007 (HMM-음절 트라이그램)'은 '제안 시스템 (CRFs+교정신뢰도 2.0)'보다 높은 정밀도를 보였다. 그러나 메모리 사용량이 많은 트라이그램 자질을 사용하고 있기 때문에 모바일 기기용으로는 부적합할 것으로 생각된다. 'Kang-2001 (음절 바이그램)'은 '제안 시스템 (CRFs+교정신뢰도 4.0)'과 비교하여 1%정도 낮은 정밀도를 보였다. 성능 차이는 크지 않지만 이것 역시 바이그램 자질을 사용하고 있기 때문에 모바일 기기에서 사용하기에는 무리가 따를 것으로 생각된다. 이러한 것들을 모두 고려했을 때, CRFs와 교정신뢰도 4.0 이상을 채택한 제안 시스템이 1MB 내외의 메모리만을 사용하면서도 비교적 높은 성능을 보인다는 것을 알 수 있었다. 또한, 5MB 정도의 가용 메모리가 있다면 95% 정도의 정밀도를 보이는 자동 띄어쓰기 시스템을 만들 수 있음을 알 수 있었다. 결과적으로 바이그램 이상의 자질을 통계 모델의 입력으로 사용하는 것보다는 제안 시스템과 같이 유니그램 자질을 통계 모델의 입력으로 사용하고 바이그램 이상의 규칙으로 재보정하는 것이 효과적인 경량화 방안이라는 것을 알 수 있었다.

두 번째로 말뭉치에 따른 제안 시스템의 성능을 평가하였다. <표 7>에서 '세종 구어체 말뭉치'는 모바일 언어적 특성이 반영되지 않은 본래의 구어체 말뭉치를 말하며, '모바일 구어체 말뭉치'는 '세종 구어체 말뭉치'에 포함되어 있는 문장들을 자동으로 변형하여 구축한 가상의 모바일 구어체 말뭉치를 말한다.

<표 7> 말뭉치에 따른 제안 시스템의 정밀도

학습 말뭉치	평가 말뭉치	오픈 테스트 정밀도 (%)
21세기 세종계획 구어체 말뭉치	21세기 세종계획 구어체 말뭉치	93.81
21세기 세종계획 구어체 말뭉치	가상 모바일 구어체 말뭉치	89.91
가상 모바일 구어체 말뭉치	21세기 세종계획 구어체 말뭉치	93.80
가상 모바일 구어체 말뭉치	가상 모바일 구어체 말뭉치	92.10

<표 7>에서 보는 것과 같이 모바일 구어체로 학습한 모델은 모바일 구어체로 평가한 것이나 세종 구어체로 평가한 것이나 비슷한 성능을 보였다. 즉, 철자 오류나 조사 생략이 일부 포함된 말뭉치로 학습한 경우에 그런 것들이 포함되지 않은 말뭉치가 입력되더라도 일정한 수준의 정밀도를 유지함을 알 수 있었다. 그러나 세종 구어체로 학습한 모델을 모바일 구어체로 평가한 경우에는 많은 성능 하락을 보였다. 이것은 모바일 환경에서 사용되는 구어체와 일반 구어체 사이에 많은 차이가 있음을 말해준다. 또한 간단한 규칙에 의해서 가상으로 구축된 말뭉치라도 모바일 환경에 적합한 띄어쓰기 시스템을 구현하는데 매우 유용하게 사용될 수 있다는 것을 보여준다.

5. 결론 및 향후 과제

본 논문에서는 모바일 기기에 적합한 한국어 띄어쓰기 교정 시스템을 제안하였다. 제안 시스템은 하이브리드 방법을 이용하여 2단계로 띄어쓰기를 교정한다. 1단계로 적은 메모리를 사용하면서도 새로운 패턴에 대한 강건성을 보장하기 위해서 음절 유니그램 기반의 통계 모델을 이용하여 띄어쓰기를 교정한다. 2단계로 정밀도 향상을 위해서 음절 바이그램 이상의 오류 교정 규칙을 이용하여 1차 교정에 실패한 것들을 재교정한다. 현실적으로 수집이 어려운 모바일 구어체 말뭉치를 구축하기 위해서는 형태소 분석기, 철자 오류 사전, 조사 변환 규칙을 이용하여 일반 구어체를 가상의 모바일 구어체로 변환하는 방법을 제안하였다. 가상의 모바일 구어체 말뭉치를 대상으로 한 실험 결과에 따르면 제안 시스템은 1MB 내외의 메모리를 사용하면서 92.10%(일반 구어체 말뭉치에서 93.80%, 일반 균형 말뭉치에서 94.07%)의 정밀도를 보였다. 향후 연구 과제는 다음과 같다. <표 7>의 결과가 실제 모바일 구어체에 대한 것이 아니기 때문에 다양한 연령대로부터 실제 수집된 말뭉치를 대상으로 한 실험이 뒤따라야 할 것으로 생각된다. 다양한 연령대로부터의 말뭉치 수집이 필요한 이유는 나이에 따라서 철자 오류나 언어 변이 현상이 매우 다르게 나타날 수 있기 때문이다.

참 고 문 헌

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

[2] 김계성・이현주・이상조 (1998). 연속 음절 문장에 대한 3단계 한국어 띄어쓰기 시스템. **정보과학회논문지, 제25권 제12호**, pp.1838-1844.

[3] 강승식 (2000). 한글 문장의 자동 띄어쓰기를 위한 어절 블록 양방향 알고리즘. **정보과학회논문지, 제27권 제4호**, pp.441-447.

[4] 심광섭 (1996). 음절간 상호 정보를 이용한 한국어 자동 띄어쓰기. **정보과학회논문지, 제23권 제9호**, pp.991-1000.

[5] 신중호・박혁로 (1997), 음절 단위 bigram 정보를 이용한 한국어 단어인식모델. **제9회 한글 및 한국어 정보처리 학술발표 논문집**, pp.255-260.

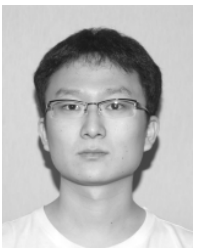
[6] 태윤식・박성배・이상조・박세영 (2006). 자기 조직화 n-gram모델을 이용한 자동 띄어쓰기. **제18회 한글 및 한국어 정보처리 학술대회 논문집**, pp.125-132.

[7] Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *in Proceedings of ICML 2001*.

[8] Pinto, D., McCallum, A., Wei, X., & Croft, W. B. (2003). Table extraction using conditional random fields. *in Proceedings of SIGIR 2003*.

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

[10] Kang, S. & Woo, C. (2001). Automatic segmentation of words using syllable bigram statistics. *in Proceedings of NLPRS 2001*, pp. 729-732.



송영길

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

김학수

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