비형식적인 문서에 강건한 문장 경계 인식

김주희○ 서정연
서강대학교 컴퓨터공학과, 서강대학교 컴퓨터공학과/바이오융합기술협동과정
kong107@sogang.ac.kr, seojy@sogang.ac.kr

Robust Method for Sentence Boundary Identification in informal documents

Juhee Kim○ Jungyun Seo
Department of Computer Science and Engineering, Sogang University
Department of Computer Science and Engineering, and Interdisciplinary Program of Integrated Biotechnology, Sogang University
kong107@sogang.ac.kr, seojy@sogang.ac.kr

요 약

본 논문에서는 구두점이나 띄어쓰기가 없는 비형식적인 문서에서도 문장의 경계를 잘 인식할 수 있는 문장 경계 인식기를 제안한다. 기존의 문장인식기는 문장경계의 후보를 구두점 출현 위치만으로 하였는데 이는 잡음이 많은 웹문서를 처리하는데 한계가 있다. 반면에 제안한 방법은 문장 경계의 후보를 구두점의 출연 위치로 제한하지 않고 문장 경계 인식을 위한 자질로 구두점에 비 의존적인 음절 n-gram을 사용함으로써, 구두점이 잘 표현된 문서뿐만 아니라 구두점의 생략이 빈번한 웹문서의 문장 경계 인식까지 효과적으로 수행할 수 있다. 통계기반의 기계학습 기법으로 CRFs를 이용하여 하였고, 학습과 실험에 세종계획 말뭉치를 사용하였다. 제안한 문장 경계 인식기는 세종계획 말뭉치에서 99.99%의 정확률과 100.00%의 재현율을 보였고, 세종계획 말뭉치에서 문장 경계의 구두점을 제거한 경우에도 96.20%의 정확률과 87.51%의 재현율을 보여 구두점이 없는 경우에도 문장 경계 인식이 잘 이루어짐을 확인할 수 있었다.

1. 서론

'문장'이란 생각이나 감정을 말로 표현할 때 완결된 내용을 나타내는 최소의 단위로, 문장 단위의 작업을 수행하기 위해서는 우선 문장의 경계를 인식하는 것이 필요하다. 문서요약, 품사 tagging, Parsing, 기계번역과 같은 자연어 처리의 주요 작업들에서 문장이 기본적인 처리 단위가 되기 때문에 문장 경계 인식 작업이 중요하다.

문장의 정의에 의하면 문장은 반드시 문장 끝에 '.', '?', '!'와 같은 구두점을 갖는다. 기존의 연구들은 이러한 문장의 특징을 기계 학습 방법에 적용하여 문장 경계 인식 문제를 해결하였다. Riley, Michael D. (1989)는 구두점 주변에 나타난 단어의 출현 확률과 구두점이 발견된 어절의 클래스를 자질로 추출하였다. AP news 2500만 단어를 이용하여 확률 정보를 구축하였고, Decision Tree (C4.5)를 이용하여 Brown 말뭉치에서 99.8%의 정확률을 보였다.[1] 임희석, 한군희 (2004)는 구두점이 발생한 곳을 문장 경계의 후보로 보고, 구두점 종류에 따른 확률, 구두점 앞/뒤 출현 음절, 인용 부호 개수에 대한 이진값 등을 자질로 추출하였다. 구어체의 ETRI 말뭉치와 문어체의 KAIST 말뭉치를 섞어 10-fold cross validation 방법으로 평가하였으며, kNN 알고리즘을 이용하여 98.82%의 정확률과 99.09%의 정확률을 보였다.[2] 박수혁, 임해창 (2008) 역시 구두점이 발생한 곳을 문장 경계의 후보로 보고, 언어의 통계적인 특징을 이용한 범용 문장 경계 인식기를 제안하였다. 구두점 종류에 따른 확률, 구두점 후보 앞/두의 토큰 및 음절 정보를 자질로 추출하였으며, 세종 계획 구어말뭉치와 문어말뭉치를 동일한 비율로 구성하여 Random Forest에서 99.1%의 정확률과 99.2%의 재현율을 보였고, 영어의 경우 Wall Street Journal 말뭉치에서 Decision Tree를 이용하여 98.9%의 정확률과 94.6%의 재현율을 보였다.[3]

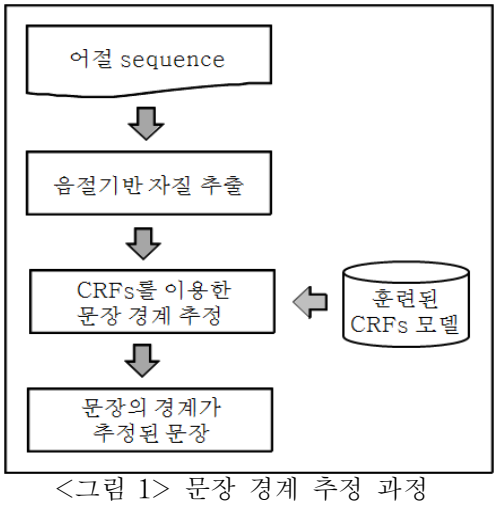
문장의 구두점 표현 규칙이 잘 지켜진 정형화된 문서를 처리할 경우에는 기존의 연구들과 같이 구두점과 관련된 자질을 이용하여 문장의 경계를 인식해 낼 수 있다. 그러나 최근에는 개인 blog 글, 댓글과 같은 형식이 자유로운

웹문서들로부터 정보를 얻는 작업이 점점 늘어나고 있다. 이러한 인터넷상의 글들은 문장마다 구두점을 포함하고 있는 경우도 있지만 그렇지 않은 경우도 매우 많기 때문에, 구두점이 발생한 위치만을 문장 경계 후보로 보는 기존의 구두점에 의존한 문장 경계 인식 방법들은 적용하기가 어렵다.

본 논문에서는 구두점이 없는 문장들에 대해서도 효과적으로 문장의 경계를 인식할 수 있는 문장경계인식기를 제안한다. 띄어쓰기나 구두점이 생략된 경우에는 모든 음절이 문장 경계의 후보가 될 수 있으므로 모든 음절을 문장 경계의 후보로 보았고, 음절기반의 자질을 사용하였다. 문장 경계 추정은 현재 음절의 문장 경계 여부가 다음에 나오는 음절들의 문장 경계 추정에 영향을 미치기 때문에 순차적인 labeling 문제로 볼 수 있다. 따라서 본 논문에서는 순차적인 labeling에 효과적인 CRFs를 문장 경계 추정에 적용하였고, 구두점만 생략된 경우와 구두점과 띄어쓰기가 모두 생략된 경우에 대해 90% 이상의 비교적 높은 성능을 얻을 수 있었다.

2. 문장 경계 추정

본 논문에서는 띄어쓰기 및 구두점의 규칙이 잘 지켜지지 않은 문서에 대해서도 문장 경계 인식이 잘 수행될 수 있도록 모든 음절을 문장 경계의 후보로 보았다. 또한 각각의 음절 다음이 문장의 경계이면 'T', 아니면 'O'로 tagging하는 방법으로 문장의 경계를 추정하였다. 일반적으로 문장이 하나의 음절로 이루어지는 경우는 매우 드물기 때문에, 만약 바로 이전의 음절이 문장의 경계로 추정되었다면 그 다음 음절은 문장의 경계가 될 가능성이 거의 없다. 즉, 이전의 음절이 문장의 경계인지 아닌지에 따라 그 다음에 오는 음절들의 문장 경계 추정 결과가 영향을 받을 수 있다.



<그림 1> 문장 경계 추정 과정

확률기반 기계학습 방법으로 CRFs(Conditional Random Fields)를 사용하였는데, 이는 CRFs가 연속적인 labeling 문제에 좋은 성능을 보이기 때문이다. <그림 1>은 CRFs를 이용한 문장 경계의 추정 과정을 보여 준다.

본 논문에서는 구두점에 의존적이지 않은 음절 n-gram과 음절의 유형을 자질로 사용하여 문장 경계를 추정하였다.

- 음절 n-gram

본 논문에서 음절 단위의 정보를 추출하였는데, 이는 음절 단위의 정보 추출이 단어나 어절 단위의 정보 추출보다 정보 추출 방법이나 자료부족 문제에 있어서 강점을 가질 뿐만 아니라 모든 음절을 문장 경계의 후보로 보기 위해서는 음절 단위의 정보 추출이 필요하기 때문이다. 단어 단위의 정보 추출의 경우 형태소 분석과 같은 복잡한 전처리 단계가 필요하고 형태소 분석의 에러 파급 가능성이 있는 반면, 음절 단위의 정보 추출은 추출 방법이 간단하다는 장점이 있다. 또한 어절 단위의 자질을 사용할 경우 data양이 적으면 자료부족의 문제가 생길 수 있는데 음절 bi-gram이나 tri-gram을 사용함으로써 자료부족 문제에 더 강건한 모델을 만들 수 있다. 일반적으로 문장 경계의 앞에 위치한 음절들이 문장 경계의 뒤에 위치한 음절들보다 문장 경계 추정에 도움이 되는 정보를 더 많이 가질 것이므로, 문장 경계의 앞에 위치한 음절들을 더 많이 반영할 수 있도록 자질을 구성하였다. <그림 2>의 자질1.7과 같이 T=-1인 시점에서의 n-gram을 추가로 사용함으로써 문장 경계 앞에 위치한 의미 있는 토큰을 자질에 추가할 수 있다.

- 음절 유형

'한글, 한자(H) / 알파벳(A) / 숫자(N) / 특수문자(C) / 빈칸(S)' 다섯 가지 유형으로 음절을 구분하여 이를 자질로 사용하였다. 음절의 유형을 자질로 사용할 경우, 똑같은 음절의 조합이 아니더라도 같은 음절 유형의 조합일 경우 이에 대한 정보를 제공할 수 있으므로 음절 n-gram만을 자질로 사용할 때 나타나는 자료부족 문제를 완화시킬 수 있다. 음절 유형의 자질 추출은 음절 n-gram와 같은 자질 추출 범위에서 똑같은 규칙을 적용하여 추출한다.

<그림 2>는 문장 경계 표현 방법 및 자질 추출의 예를 보여준다. 자질 집합 <1.6>의 경우 자질 추출 범위 T=-3 부터 T=3까지 T=0인 시점에서 T를 중심으로 T를 포함한 좌/우 n-gram을 자질로 추출한 것이고, 자질 집합 <1.7>은 T=-1인 시점에서 T를 중심으로 T를 포함한 좌/우 n-gram을 추출한 후 이를 자질 집합 <1.6>

에 추가한 것이다. 여기서 추가된 n-gram의 경우, 자질 추출 범위는 그대로 유지하면서 T=-1인 시점을 중심으로 T를 포함한 n-gram을 추출하였기 때문에 좌측에서는 tri-gram까지만 추출이 가능하고, 우측에서는 fifth-gram까지 추출이 가능하다.

Time(T)	...	-6	-5	-4	-3	-2	-1	0	1	2	3	4	...
음절	...	제	약	이	SP	있	다	SP	따	라	서	SP	...
음절유형	...	H	H	H	S	H	H	S	H	H	H	S	...
문장경계	...	O	O	O	O	O	O	T	O	O	O	O	...

<자질 1.6>
- SP, 다SP, 있다SP, SP있다SP, SP따, SP따라, SP따라서

<자질 1.7>
- SP, 다SP, 있다SP, SP있다SP, SP따, SP따라, SP따라서
  다, 있다, SP있다, 다SP따, 다SP따라, 다SP따라서

<그림 2> 문장 경계 및 자질 추출의 예

문장 경계 추정에 사용된 자질 집합은 <표 1>과 같다.

자질 집합 번호	자질	
	자질 추출 범위	자질 내용
1.0	음절 -2, +2	T=0를 중심으로 좌/우 n-gram
1.1	음절 -2, +2	T=0, -1를 중심으로 좌/우 n-gram
1.2	음절 -3, +1	T=0를 중심으로 좌/우 n-gram
1.3	음절 -3, +1	T=0, -1를 중심으로 좌/우 n-gram
1.4	음절 -4, 0	T=0를 중심으로 좌/우 n-gram
1.5	음절 -4, 0	T=0, -1를 중심으로 좌/우 n-gram
1.6	음절 -3, +3	T=0를 중심으로 좌/우 n-gram
1.7	음절 -3, +3	T=0, -1를 중심으로 좌/우 n-gram
1.8	음절 -4, +2	T=0를 중심으로 좌/우 n-gram
1.9	음절 -4, +2	T=0, -1를 중심으로 좌/우 n-gram
2.0	음절유형 -2, +2	T=0를 중심으로 좌/우 n-gram
2.1	음절유형 -2, +2	T=0, -1를 중심으로 좌/우 n-gram
2.2	음절유형 -3, +1	T=0를 중심으로 좌/우 n-gram
2.3	음절유형 -3, +1	T=0, -1를 중심으로 좌/우 n-gram
2.4	음절유형 -4, +0	T=0를 중심으로 좌/우 n-gram
2.5	음절유형 -4, +0	T=0, -1를 중심으로 좌/우 n-gram
2.6	음절유형 -3, +3	T=0를 중심으로 좌/우 n-gram
2.7	음절유형 -3, +3	T=0, -1를 중심으로 좌/우 n-gram
2.8	음절유형 -4, +2	T=0를 중심으로 좌/우 n-gram
2.9	음절유형 -4, +2	T=0, -1를 중심으로 좌/우 n-gram

<표 1> 문장 경계 추정에 사용된 자질 set

3. 실험 및 결과

본 논문에서는 문어체뿐만 아니라 구어체에서도 문장의 경계를 잘 찾아내는 문장 경계 인식기를 제안한다. 따라서 세종 계획 구어말뭉치와 문어말뭉치를 동일한 비율로 하여 데이터를 구성하였다. 말뭉치의 크기는 24000문장(260,736어절)과 56000문장(606,723어절)로 달리 하여 학습 데이터 크기에 따른 성능을 비교하였다. 10-fold cross validation 으로 평가 하였으며, 통계적 기계학습 tool로 CRF++v0.530을 사용하였다.

문장 경계 인식기의 성능은 문장 정확률과 문장 재현율, F1-measure로 평가하였으며, 이들은 다음과 같이 정의 하였다.

$$\text{문장 정확률}(P) = \frac{\text{시스템이 추출한 정답 문장 경계수}}{\text{시스템이 추출한 문장 경계수}}$$

$$\text{문장 재현율}(R) = \frac{\text{시스템이 추출한 정답 문장 경계수}}{\text{전체 문장 경계수}}$$

$$F1-measure(F) = \frac{2*\text{문장 정확률}*\text{문장 재현율}}{\text{문장 정확률}+\text{문장 재현율}}$$

3.1 최적의 자질 추출

구두점이 없는 문장의 경계 추정에 최적의 자질을 적용하기 위해 음절 n-gram 자질과 음절유형 n-gram 자질을 조합하여 성능을 측정하였다. 구두점이 없는 문장의 경계 추정을 위해 56000문장의 말뭉치에서 문장의 경계에 있는 모든 구두점을 제거한 후 성능을 측정하였고, 성능 측정 결과는 <표 2>와 같다.

<표 2> 자질 조합에 따른 문장 경계 추정 결과

자질 조합	Precision	Recall	F1-measure
1.0 + 2.0	0.9528	0.8616	0.9049
1.1 + 2.1	0.9546	0.8648	0.9075
1.2 + 2.2	0.9557	0.8678	0.9096
1.3 + 2.3	0.9552	0.8706	0.9109
1.4 + 2.4	0.9512	0.8615	0.9041
1.5 + 2.5	0.9480	0.8639	0.9040
1.6 + 2.6	0.9614	0.8730	0.9151
1.7 + 2.7	0.9620	0.8751	0.9165
1.8 + 2.8	0.9591	0.8724	0.9137
1.9 + 2.9	0.9612	0.8749	0.9160
1.7	0.9617	0.8749	0.9163

성능 평가 결과 문장 경계 앞, 뒤 3음절의 음절 n-gram과 음절유형 n-gram을 사용하고, T=-1 시점의 n-gram을 추가한 자질조합 '1.7 + 2.7'이 가장 좋은 성능을 보였다. <표2>를 통해 문장 경계 추정에서 문장 경계 앞의 음절이 문장 경계 뒤의 음절보다 더 가치 있는 것은 문장 경계 앞의 음절 3개까지만 이라는 것을 알 수 있었다. 또한 자질조합 '1.6 + 2.6'과 '1.7 + 2.7'을 비교함으로써 T=-1 시점의 음절 n-gram을 추가하는 것이 재현율의 향상에 기여함을 알 수 있었다. 이는 대부분의 문

장 경계가 어절과 어절 사이에 존재하여, T=0 시점의 음절 n-gram에는 빈칸이 포함되지만, T=-1시점에는 <그림2>의 예제와 같이 빈칸 이전의 의미 있는 음절 n-gram도 포함되기 때문에 성능이 향상된다. 음절유형 n-gram자질은 문장을 이루는 음절유형이 다른 두 문장 사이의 문장 경계를 추정하는데 도움이 되는데 이는 자질조합 '1.7 + 2.7'과 '1.7'의 성능비교를 통해 알 수 있다.

3.2 문장의 경계 추정 성능

최적의 자질 추출 실험을 통하여 구두점이 없는 문서의 문장 경계 추정이 91%이상의 성능을 보임을 확인하였다. 제안한 방법이 구두점과 띄어쓰기 모두 제거된 문서와 구두점과 띄어쓰기가 모두 표현된 형식적인 문서에서도 문장 경계 추정에 효과적임을 확인하기 위해, 구두점과 빈칸을 모두 제거한 데이터와 구두점과 띄어쓰기가 모두 있는 데이터에 대해서 각각의 모델을 생성하고 성능을 평가하였다. 각각의 모델에 생성된 자질 집합 조합은 '1.7 + 2.7'로, 구두점을 제거한 데이터에서 가장 좋은 성능을 보인 자질조합을 사용하였다. 이에 대한 성능은 <표3>과 같다.

<표 3> 구두점 및 띄어쓰기에 따른 문장 경계 성능

	어절 수	Precision	Recall	F1-measure
구두점 無 띄어쓰기 有	260,736	0.9631	0.8669	0.9125
	606,723	0.9620	0.8751	0.9165
구두점 無 띄어쓰기 無	260,736	0.9605	0.8105	0.8792
	606,723	0.9638	0.8264	0.8898
구두점 有 띄어쓰기 有	260,736	0.9998	0.9999	0.9999
	606,723	0.9999	1.0000	0.9999

<표3>을 보면, 제안한 방법은 구두점만 없는 경우 약 91%, 구두점과 빈칸이 모두 없는 경우에도 88%이상의 성능을 보이는 것을 알 수 있다. 이는 제안한 방법이 문장 경계 인식에 있어 구두점과 띄어쓰기 정보에 크게 의존하지 않고, 이를 제외한 어휘 정보만으로도 충분히 효과적으로 문장의 경계를 인식할 수 있음을 말해준다. 또한 구두점과 띄어쓰기가 모두 있는 데이터에 대해 99.99%의 성능을 나타내며 제안한 방법이 형식적인 문서의 문장 경계 추정에도 효과적임을 보였다.

구두점이 있는 경우와 구두점이 없는 경우를 비교하면, 문장 정확률보다 문장 재현율이 더 많이 하락하는 것을 볼 수 있다. 이는 구어체 문장에 문장의 종결이 일반적이지 않은 문장들이 다수 포함되어 있기 때문이다. "그게 뭐가 재밌냐고."와 같이 문장의 종결이 일반적이지 않은 문장은 주어진 데이터로 문장의 경계를 추정함에 있어 구두점에 의존적일 수밖에 없다. 또한 모든 문장의 경계에는 띄어쓰기가 있어서, 문장의 경계 추정에 띄어쓰기가 큰 단서가 된다. 따라서 띄어쓰기 없어질 경우 재현율은 하락하게 된다.

<표 4> 구두점 표현과 띄어쓰기가 잘 되어있는 데이터에서의 문장경계 추정.

	어절 수	P	R	F
baseline	260,736	0.9810	1.0000	0.9904
	606,723	0.9807	1.0000	0.9902
기존방법 (Random Forest)	256,529	0.9830	0.9860	0.9850
	619,567	0.9910	0.9920	0.9910
제안한 방법 (CRFs)	260,736	0.9998	0.9999	0.9999
	606,723	0.9999	1.0000	0.9999

<표 4>를 통해 구두점 표현과 띄어쓰기가 잘 되어있는 데이터에서도 제안한 방법이 효과적임을 확인할 수 있다. baseline은 제안한 방법에서 사용한 데이터에 대해 모든 구두점을 문장의 경계로 추정한 것이고, 기존방법(Random Forest)[3]은 세종 계획 구어말뭉치와 문어말뭉치를 동일한 비율로 구성하고, Random Forest를 이용하여 문장의 경계를 추정한 것이다. 기존 방법에서는 baseline과 비교하였을 때 정확률이 상승한 반면 재현율은 하락하였다. 그러나 제안한 방법에서는 정확률과 재현율 모두 99.99%이상의 높은 성능을 보였다.

4. 결론 및 향후 연구

본 논문에서는 CRFs(Conditional Random Fields)를 사용하여 정형화되지 않은 문서의 문장경계를 추정하는 방법을 제안하였다. 구두점 정보에 의존적이지 않은 어휘 정보와 문맥 정보를 자질로 사용함으로써 SMS, 웹문서 등 구두점이나 띄어쓰기가 생략된 문서의 문장경계를 효과적으로 추정하였다. 제안한 문장 경계 인식기는 구두점만 생략된 경우는 96.20%의 정확률과 87.51%의 재현율을 보였고, 구두점과 띄어쓰기 모두 생략된 경우는 96.38%의 정확률과 82.64%의 재현율을 보였다. 또한 구두점이 생략되지 않은 경우에는 99.99%의 정확률과 100.00%의 재현율을 보여 제안한 방법이 정형화된 문서와 정형화되지 않은 문서의 문장 경계 추정에 모두 효과적임을 알 수 있었다. 제안한 방법은 적은 양의 학습 말뭉치만으로도 비교적 높은 성능을 보였고, 구어체와 문어체를 아우르는 다양한 패턴의 문장 처리가 가능하다는 장점이 있다.

향후 연구로는 정형화되지 않은 문서의 문장 경계 인식에 있어 다양한 예외 사항을 파악하고 이를 개선할 수 있는 자질을 고려할 계획이다. 실제 웹 데이터 및 SMS 데이터에 대한 실험을 통해 데이터의 특징에 알맞은 자

질을 추출하고 이를 문장 경계 인식기에 적용하고자 한다.

참 고 문 헌

[1] M. D. Riley, "Some Applications of Tree-based Modeling to Speech and Language.", *In Proceedings of the DARPA speech and natural language workshop*, pp.339-352, 1989.

[2] H. S. Lim, K. H. Han, "Korean Sentence Boundary Detection Using Memory-based Machine Learning.", *Koreacontents*, vol.4, no.4, pp. 133-139, 2004. (in Korean)

[3] S. H. Park, H. C. Lim, "Sentence Boundary Detection Using Machine Learning Techniqes.", *KIISE,* vol.15, no.1, pp.241-267, 2008. (in Korean)

[4] D. D. Palmer and M. A. Hearst, "Adaptive Multilingual Sentence Boundary Disambiguation", *Computational Linguistics*, vol. 23, no. 2, pp. 241-269, 1997.

[5] J. C. Reynar and A. Ratnaparkhi, "A Maximum Entropy Approach to Identifying Sentence Boundaries", *In Proceedings of the Fifth Conference on Applied Natural Language Processing(ANLP'97)*, pp.16-19, 1997.