k-NN 으로 확장된 한국어 단위화

박성배° 장병탁 김영택

서울대학교 컴퓨터공학부

sbpark@nova.snu.ac.kr {btzhang,ytkim}@cse.snu.ac.kr

Expanded Korean Chunking by *k*-NN

Seong-Bae Park° Byoung-Tak Zhang Yung Taek Kim

School of Computer Science and Engineering, Seoul National University

요 약

대부분의 자연언어처리에서 단위화는 구문 분석 이전의 매우 기본적인 처리 단계로, 텍스트 문장을 문법적으로 서로 관련된 단위로 분할하는 것이다. 따라서, 단위화를 이용하면 구문 분석이나 의미 분석 등에서 메모리와 시간을 효율적으로 줄일 수 있다. 일반적으로, 통찰에 의한 규칙을 사용해서도 비교적 높은 단위화 성능을 얻을 수 있지만, 본 논문에서는 기계 학습 기법인 *k*-NN 을 사용하여 보다 정확한 단위화를 구현한다. 인터넷 홈페이지에서 얻은 1,273 문장을 대상으로 학습한 결과, *k*-NN 으로 단위화를 확장했을 때에 확장하지 않았을 때보다 2.3%의 정확도 증가를 보였다.

1. 서론

텍스트 단위화(chunking)는 문법적으로 서로 관련된 단어 집합으로 텍스트를 나누는 것을 말한다. 예를 들어, 다음과 같은 예문을 살펴보자.

- 작은 세 마리의 곰이 마당에서 놀다가 잠든 듯하다.

이 문장은 다음과 같이 나누어 질 수 있다.

- [NP 작은 세 마리의 곰이] [NP 마당에서] [VP 놀다가] [VP 잠든 듯하다.]

이 문장에서 "작은 세 마리의 곰이", "마당에서", "놀다가", "잠든 듯하다"가 문법적으로 관련된 하나의 단위로 나뉘어질 수 있으므로, 이 단위를 사용하면 구문 분석과 같은 자연언어처리에서의 효율을 높일 수 있다.

Abney 가 서로 관련있는 단어들의 집합을 사용한 구문분석의 가능성을 처음으로 제시한 후[1], Ramshaw 와 Marcus 는 기계학습이 단위화 문제에 잘 이용될 수 있음을 보였다[2]. 이 후, 기계학습을 이용한 많은 연구들이 단위화 문제에 적용되었고 상당히 좋은 성과를 거두고 있다[7].

한편, 한국어는 교착어이기 때문에, 영어와 같은 언어에 비해서 단위화가 비교적 용이하다. Abney 는 단위(chunk)를 하나의 내용어와 함께 나타나는 기능어로 이루어진다고 하였지만, 한국어에서는 조사나 어미와 같은 기능어가 내용어와 함께 하나의 어절을 이루므로, 조사나 어미를 이용해서 쉽게 단위화를 수행할 수 있다. 신효필은 한국어에 대한 통찰을 통해 한국어 단위화를 위한 규칙을 처음으로 제시하였다[4]. 김미영도 비슷한 방법으로 단위화를 처리하여, 약 93% 정도의 정확도를 얻었다[5] . 이 정확도는 매우 높다고도 할 수 있지만, 보다 더 정확한 단위화의 필요성이 요구된다. 예를 들어, 약 30 어절 길이의 문장을 단위화한다면, 이정도 정확도로는 각 문장마다 평균 2 개 정도의 오류를 포함하게 된다. 대부분의 자연언어처리 시스템에서의 단위화는 구문 분석 이전의 매우 기본적인 단계에서 행해지므로, 이 단계에서의 오류는 전체 자연언어처리 시스템에서 매우 크게 나타난다. 따라서, 보다 완벽하게 단위화 문제를 해결할 수 있는 방법이 필요하다.

본 논문에서는 기존의 규칙 기반의 단위화에 기초하고 기계 학습 기법을 사용하여 한국어의 단위화 성능을 향상시키는 방법을 제시한다. 기본적으로 단위화는 규칙에 기반해서 처리되며, 단위화를 결정하기 힘든 모호한 부분에서는 *k*-NN(*k*-nearest neighbor) 학습 방법으로 관련 정보에서 추출한 정보를 이용하여 단위화를 처리한다. *k*-NN 학습이 예제기반 방법(instance-based method)이기 때문에, 학습 초기에 비교적 적은 수의 학습 예제만으로 높은 성능을 낼 수 있고 학습 예제의 수가 늘어남에 따라 보다 높은 성능을 기대할 수 있다.

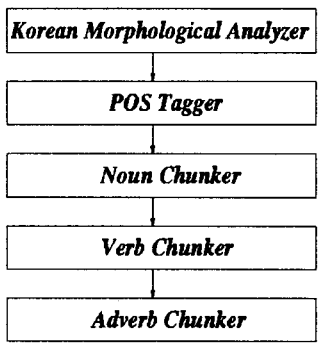


그림 1. 한국어 단위화 시스템의 구조.

2. 단위화 시스템의 개요

그림 1 은 한국어 문장을 단위화하는 시스템을 간략하게 보이고 있다. 일반적으로, 자연언어처리에서는 형태소

분석이 첫번째 단계이다. 또한, 형태소 분석기가 형태적 또는 품사적 모호성으로 인해 복수개의 분석 결과를 낼 수 있으므로, 이 중에서 올바른 것을 선택하는 품사 태깅 과정이 형태소 분석 다음에 수행된다.

한국어 문장은 크게 보아 명사구, 동사구[1], 부사구로 나누어 질 수 있기 때문에 명사구, 동사구, 부사구를 담당하는 각각의 단위화 모듈이 필요하다. 각각의 단위화 모듈은 다음 장에서 설명될 것이다.

3. 한국어의 단위화

3.1 명사구 단위화

신효필은 명사구 단위화를 위한 일반적인 규칙을 제시하였고[5], 김미영은 이 규칙에 기반하여 명사구 단위화를 위한 규칙을 다음과 같이 5 개로 정리하였다[6].

- 규칙 1: (관형격 조사)* 명사(구)
- 규칙 2: (관형형 전성어미)* 명사(구)
- 규칙 3: (관형사파생접사)* 명사(구)
- 규칙 4: (관형사)* 명사(구)
- 규칙 5: (체언)* 명사

본 논문에서는 한성대학교의 한국어 형태소 분석기 HAM 를 통해 명사구 단위화를 수행하기 위해서, 규칙 5 를 아래와 같이 확장하였다.

- 규칙 5: (체언)* 명사(구)

[6]에서는 규칙 5 에서 명사구를 다루지 않은 이유가 "[국어 [사전의 색깔]]"과 같은 잘못된 단위화를 막기 위해서라고 했는데, 이를 "[국어 사전의 색깔]"로 보면 문제가 되지 않으므로 명사구를 포함하였다.

규칙 3 에서는, 관형사파생접사 '-적'이 홀로 존재하는 경우 뿐만 아니라, 관형사파생접사 다음에 서술격조사 '이'와 관형형 전성어미가 나타나는 경우도 포함하였다.

3.2 동사구 단위화

한국어에서의 동사구 단위화는 "복합 동사 처리"라는 이름으로 비교적 오래 전부터 연구되어 온 문제이며, 일반적인 성능도 매우 높은 편이다[3]. [5]와 [6]에서는 이 문제에 대하여 유한상태 오토마타를 이용한 접근방식을 제시하였으나, 본 논문에서는 [3]에서 제시한 지식 정보를 바탕으로 한 규칙으로 이를 대신한다.

3.3 부사구 단위화

부사구는 연속으로 나타나는 부사만 단위로 인식하였다. 연속으로 나타나는 부사가 모두 단위를 이루는 것은 아니지만, 단위를 이루는 경우가 이루지 않는 경우보다 많으므로 연속으로 나타나는 부사는 단위를 이루는 것으로 가정한다. 연속으로 나타나는 부사가 단위를 이루지 않는 경우는 아래에서 설명할 기계학습 기법을 사용해서 처리한다.

4. 기계학습 기법을 사용한 확장

4.1 기존 단위화의 문제점

한국어 단위화는 위에서 설명한 규칙만으로도 비교적 높은 정확도를 보이지만, 단위화가 구문 분석의 전처리 단계로 사용된다면 더 정확해져야 한다. 이는 단위화에서의 오류가 구문 분석을 포함한 더 복잡한 분석에서 회복될 수 없기 때문이다.

실제 문장에 위에서 제시한 규칙을 적용한 후 얻은 단위화 오류의 종류는 그림 2 와 같다.

1. B-NP & I-NP 문제
   - [한국에][있는][동안 부모님을][뵈었다.]
   ⇒[한국에][있는][동안][부모님을][뵈었다.]
2. I-NP & I-NP 문제
   - [그는][오늘 아침 물가에서][떨고 있었다.]
   ⇒[그는][오늘 아침][물가에서][떨고 있었다.]
3. B-NP & B-NP 문제
   - [그녀는][코가][예쁜 소녀][이다.]
   ⇒[그녀는][코가 예쁜][소녀][이다.]
4. B-NP, I-NP & B-NP 문제
   - [이 황금][같은 기회를][놓치지 마라.]
   ⇒[이][황금 같은][기회를][놓치지 마라.]
5. B-ADVP & I-ADVP 문제
   - [나는][그를][그저 조용히][바라보았다.]
   ⇒[나는][그를][그저][조용히][바라보았다.]
6. I-NP & B-NP 문제
   - [하루에][수십 개씩 쇼핑몰이][생겨난다.]
   ⇒[하루에][수십 개씩][쇼핑몰이][생겨난다.]

그림 2. 단위화 오류의 종류 및 예제

각 오류의 종류는 규칙으로 잘못 결정된 레이블을 기준으로 하여 나뉘어 졌다. 예를 들어, 1 번 문제에서 "동안 부모님을"은 규칙에 의해 각각 B-NP[2]와 I-NP 로 결정되었다. 하지만, 이 경우의 정확한 레이블은 B-NP 와 B-NP 이다. 오류가 있는 예제에서 오류가 있는 부분을 정확한 레이블로 고친 후 각 오류의 종류마다 예제를 따로 저장하였다.

4.2 *k*-NN 을 이용한 한국어 단위화의 확장

본 논문에서 어절 w_i의 정확한 단위화 레이블을 결정하기 위해 사용되는 자질은 표 1 과 같다.

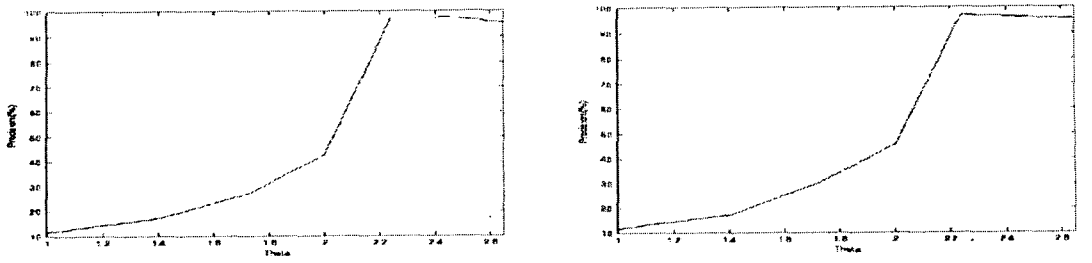
자 질	설 명
w_i	w_i의 어휘
w_{i-1}	w_{i-1}의 어휘
w_{i-2}	w_{i-2}의 어휘
POS_i	w_i의 품사
POS_{i-1}	w_{i-1}의 품사
POS_{i-2}	w_{i-2}의 품사
E_{i-1}	w_{i-1}의 어미 혹은 조사
R_{i-1}	w_{i-1}의 단위화 레이블
R_{i-2}	w_{i-2}의 단위화 레이블

표 1. 단위화를 위해 사용되는 자질

k-NN 에서는 각 학습 예제를 *n*-차원 공간 R^n 상의 점

[1] 본 논문에서 동사는 일반적으로는 동사와 형용사를 모두 나타내는 용언의 뜻으로 사용되나, 문맥에 따라서는 동사와 형용사를 분리하여 사용하기도 한다.

[2] 본 논문에서 사용된 단위(chunk)의 레이블은 B-NP, I-NP, B-VP, I-VP, B-ADVP, I-ADVP 이다. 이들은 각각 명사구, 동사구, 부사구의 시작을 끝을 나타낸다.



(a) $k = 1$ (b) $k = 3$.

그림 3. θ 값에 따른 단위화 정확도의 변화. (a)는 최근점의 수 k를 1로 했을 때이고, (b)는 이를 3으로 했을 때이다.

으로 보고, 어떤 예제와 가장 가까운 예제를 유클리드 거리로 결정한다.

본 논문에서는 학습 예제가 기본 규칙으로 정확하게 결정되지 않는 예제들이므로, 전적으로 k-NN의 결정을 믿는 것이 아니라 관심 예제와 k 개의 학습 예제 사이의 평균 거리가 미리 정해진 임계치 θ 이상일 때만 k-NN의 결정을 받아들이고 그렇지 않은 경우에는 기존 규칙의 결정을 받아들인다. 즉, 관심 예제를 x_q라고 할 때,

$$\frac{\sum_{i=1}^{k} d(x_i, x_q)}{k} \geq \theta$$

일 때만 k-NN의 결정을 받아들인다. 여기서, x_1, ..., x_k는 k-NN에 의해 선택되어진 가장 가까운 이웃들이다.

5. 실험

인터넷 홈 페이지에서 얻은 1,911 문장을 대상으로 단위화를 실험하였다. 이 문장의 평균 길이는 8.9 어절이다. 이 중, 1,273 문장을 학습 데이터로 사용하였고, 나머지 638 문장을 테스트 데이터로 사용하였다.

표 2는 한국어 단위화의 실험 결과이다. 3장에서 제시한 규칙만으로도 95.5%의 비교적 높은 정확도를 보이며, k-NN을 사용하여 확장했을 때에는 이보다 2.3% 더 높은 97.8%의 정확도를 보였다. 이 결과는 k = 1로 하였을 때 얻어진 결과이다.

	학습 데이터	테스트 데이터
기본 방법	95.3 %	95.5 %
k-NN 확장	N/A	97.8 %

표 2. 한국어 단위화 실험 결과.

그림 3은 k의 값과 임계치 θ가 k-NN으로 확장한 한국어 단위화에 미치는 영향을 보인다. (a)는 k = 1일 때이고, (b)는 k = 3일 때이다. 정확도의 변화는 k가 1일 때와 3일 때 모두 비슷한 결과를 보이며, θ의 값이 증가할수록 정확도가 높아지다가 어느 정도가 지나면 더 이상 좋아지지 않고 조금 나빠진다. 그 이유는 θ 값이 너무 높으면 k-NN이 적용되는 경우가 거의 없기 때문이다.

6. 결론

본 논문에서는 통찰에 의한 규칙과 이를 확장할 수 있는 k-NN 기법을 사용하여 보다 높은 한국어 단위화의 정확성을 얻을 수 있는 방법을 제시하였다. k-NN이 확실성을 가지고 답을 제시하는 경우에는 이를 받아들이고, 그렇지 않은 경우에는 기존의 규칙을 사용하여 단위화를 수행하였다. 실험 결과, k-NN을 사용하여 확장했을 경우가 k-NN을 사용하여 확장하지 않았을 때보다 2.3%정도 높은 정확도를 보인다. 일반적으로, 자연언어처리에서의 단위화는 비교적 초기단계에서 처리되고 이 단계에서의 오류는 후위 단계에서 회복되기 힘들기 때문에, 2.3%의 정확도 증가는 매우 중요한 의미를 갖는다.

감사의 글

본 연구는 정보통신부 대학기초지원사업인 "지능형 인터넷 정보서비스를 위한 대규모 텍스트 분류 및 검색 기술 개발"(과제번호 00-102)에 의하여 일부 지원되었음.

참고 문헌

[1] S. Abney, "Parsing by Chunks," In *Principle-Based Parsing*, Kluwer Academic Publishers, 1991.

[2] L. Ramshaw and M. Marcus, "Text Chunking Using Transformation-Based Learning," In *Proceedings of the Third ACL Workshop on Very Large Corpora*, pp. 82-94, 1995.

[3] 김기철, 이기오, 이용석, "형태소 분석 주도의 한국어 복합동사 처리," *정보과학회논문지*, 22권, 9호, pp. 1384-1393, 1995.

[4] H.-P. Shin, "The VP-Barrier Algorithm for a Robust Syntactic Parsing in Head-Final Languages," In *Proceedings of Natural Language Processing Pacific Rim Symposium*, pp. 475-478, 1999.

[5] 신효필, "최소자원 최대효과의 구문분석," 제 11 회 한글 및 한국어정보 처리 학술 대회 논문집, pp. 242-247, 1999.

[6] 김미영, 강신재, 이종혁, "단위분석과 의존문법에 기반한 한국어 구문분석," 제 27 회 정보과학회 봄 학술대회 논문집, pp. 327-329, 2000.

[7] Erik T.-K. Sang, "Noun Phrase Representation by System Combination," In *Proceedings of ANLP-NAACL 2000*, pp. 50-55, 2000.