어절의 중심어 정보를 이용한 한국어 기반 명사구 인식

KAIST/KORTERM 전자전산학과
서충원 · 오종훈 · 최기선

Korean BaseNP Chunking Using Head-word of Word Phrase

Seo Chung-Won, Oh Jong-Hoon, Key-Sun Choi
Depatment of EECS, KAIST/KORTERM, Deajeon, Korea

요 약

기반 명사구는 명사구 내부에 다른 명사구를 포함하지 않는 명사구로 정의된다. 이러한 기반명사구인식은 구문해석의 성능을 향상시키기 위한 방법으로 많이 사용되어 왔다. 효과적인 기반 명사구인식을 위해서는 올바른 학습자질의 선택과 적절한 문맥의 범위의 설정이 중요하다. 이러한 관점에서 기존의 연구에서는 여러 가지 학습자질과 문맥의 범위로 기반명사구를 인식하였다. 하지만 기존의 연구들에서는 학습자질로 단순한 어휘, 품사, 띄어쓰기 정보만을 사용하여 좁은 범위의 문맥정보만을 사용하였다. 본 논문에서는 한국어의 기반 명사구 인식을 위해 학습의 자질로 어절의 중심어를 사용하는 HMM모델을 제안한다. 본 논문의 방법을 통해 정확률 94.3%, 재현률 93.2%의 성능을 얻었다.

서 론

구문 해석은 문장 요소들 사이의 관계를 밝혀 문장의 구조를 파악하는 과정이다. 이는 기계 번역이나 의미 해석, 질의 응답과 같은 자연언어처리 응용의 기초 모듈로 사용이 되어 왔다. 구문 해석의 성능은 전체 자연언어응용 시스템의 성능을 좌우한다. 하지만, 전 문장을 대상으로 하는 구문해석은 높은 정확도를 얻기 힘들다. 이로 인해 실제 응용에서는 부분 구문 해석이나 기반구 인식(chunking) 등의 전처리 과정을 통하여 구문해석을 수행한다.

기반구 인식은 문장의 구성 요소들을 일정한 기준으로 묶어 더 큰 단위로 만들어 주는 것으로 이후 단계의 처리의 복잡도를 줄여 준다.[6] 기반구 인식의 대상은 명사구(NP), 동사구(VP), 형용사구(ADVP), 부사구(ADJP) 등이 있으며, 영어의 경우 표준화된 코퍼스를 이용하여 기반구 인식의 연구가 활발히 진행되고 있다.[2,3,7,8]

한국어의 기반구 인식에 대한 연구는 주로 명사구를 대상으로 하며, 명사구 중에서도 기반 명사구(base NP)에 대한 많은 연구가 있어 왔다. 기반 명사구는 명사구 내부에 다른 명사구를 포함하지 않는 명사구로 정의된다.[6,11,12] 영어의 경우 명사의 수식으로 전치/후치 수식이 모두 가능하지만, 전치 수식만을 기반 명사구에 포함시킨다. 한국어의 경우는 중심어가 항상 뒤에 나오기 때문에 전치/후치 수식의 구분이 없다. 한국어의 기반 명사구로는 "그 사이", "험하기", "뚜렷한 원칙", "더 큰 관심", "하얗고 깨끗한 옷"과 같은 것들이 가능하다.

기반 명사구 ::= {관형사| 동사구 + 관형사형 어미}*명사열

동사구 ::= {부사}* {용언+ 연결어미}* {용언}

용언 ::= 동사| 형용사| 동작/상태성명사+동사/형용사 파생조사

명사열 ::= {명사| 대명사| 수사} +| 동사+명사형 전성어미

기반 명사구에는 명사를 수식하는 관형어나 동사구가 포함된다. 하지만, "언어를 다룰 때"와 같이 명사를 수식하는 동사구 내부에 명사구가 포함 될 경우는 마지막 "때"만을 기반 명사구로 인정한다.

기반 명사구의 인식은 문장의 전체 구문 구조 중 부분적인 구문 구조를 파악하는 작업이다. 한국어의 경우 문법

E-mail : cwseo@world.kaist.ac.kr
E-mail : rovellia@world.kaist.ac.kr
E-mail : kschoi@world.kaist.ac.kr

적인 기능을 하는 단위와 내용을 구성하는 단위가 하나의 어절로 구성이 되기 때문에 기반 명사구 인식에서 어절의 구성 정보가 중요하다. 본 논문에서는 한국어의 기반 명사구 인식을 위해 어절의 구성 정보로, 내용어와 기능어의 중심어 정보를 사용하고, 인접 어절의 중심어 형태소 정보를 문맥 정보로 사용하여 기반 명사구를 인식하는 방법을 제안한다.

논문의 구성은 다음과 같다. 2장에서는 기반 명사구 인식의 표현 형식과 한국어 기반 명사구 인식에 대해서 설명한다. 3장에서는 기반 명사구 인식을 위한 확률 모델을, 4장에서는 실험 결과를 5장에서는 결론을 이야기한다.

관련연구

1. 기반명사구의 표현 형식

기반명사구의 표현 형식은 크게 괄호를 이용한 방법과, I(inside), O(out) 태그를 이용한 방법으로 나눌 수 있다.

괄호를 이용한 방법은 기반 명사구의 시작과 끝을 괄호로 표지하는 방식이다. 다음의 예에서 '[ ]'이 기반명사구를 나타낸다.

[삶/ncn]+의/jcm [모습/ncn]+,/sp [도시/ncn]+의/jcm [모습/ncn] [사라지/pvg+ㄴ/etm 시냇물/ncn] [불볕/ncn 더위/ ncn]+가/jcs

괄호를 이용한 방법은 부가적으로 표시되는 정보를 최소화 하면서 기반 명사구 정보를 표지할 수 있다는 장점이 있지만, 기계적으로 처리하기 위해서는 코퍼스를 따로 처리해서 명사구의 내부/외부 정보를 추출해야 하는 번거로움이 있다.

IO 태그를 이용한 방법은 형태소 태그에 부가적으로 IO 태그를 표지해 주는 방법이다. IO만을 가지고 표현을 할 경우, 두 개의 기반 명사구가 연달아 나올 때, 구분을 해 주기 힘들다는 문제가 있다. Ramshaw[5]는 B (begin) 태그를 추가하여 시작 부분을 구분 해 주는 방식을 제안 하였으며, Tjong[1,2]은 B(begin)와 E(end) 태그를 사용하여 다음과 같은 4가지 종류의 표현 방법을 제안하였다.

- IOB1 : 연속한 기반구에서 두 번째 기반구의 시작을 B로 표지
- IOB2 : 기반구의 시작을 B로 표지
- IOE1 : 연속한 기반구의 첫 번째 기반구의 끝을 E로 표지
- IOE2 : 기반구의 끝을 E로 표지

Uchimoto et al.,[4]는 일본어 개체명의 표현을 위해 B와 E를 모두 사용하고, B와 E가 겹치는 경우에 한하여 S 태그를 사용하는 BIOES 방법을 제시하였으며, 이는 기반 명사구를 위한 표현으로도 사용할 수 있다.

- B : 기반구의 시작을 나타내는 태그
- E : 기반구의 끝을 나타내는 태그
- I : 기반구의 내부를 나타내는 태그
- O : 기반구의 외부를 나타내는 태그
- S : 하나의 토큰으로 이루어진 기반구를 나타내는 태그

본 논문에서는 1), 2), 4)의 표현방법과 한국어의 수직코퍼스의 형식을 사용하여 한국어 기반명사구를 표현한다.[10]

1) 삶/ncn + E
2) 의/jcm . O
3) 모습/ncn + E
4) ,/sp . O
5) 도시/ncn + E
6) 의/jcm . O
7) 모습/ncn * E
8) 사라지/pvg + I
9) ㄴ/etm . I
10) 시냇물/ncn * E
11) 불볕/ncn * I
12) 더위/ncn + E
13) 가/jcs . O

띄어쓰기 정보는 다음 형태소와 같은 어절일 때는 '+'를, 어절의 끝일 경우는 "."를, 하나의 형태소로 이루어진 어절일 경우는 '*'를 사용하여 나타낸다.

2. 한국어의 기반 명사구 인식 연구

양재형[11]은 기반 명사구 인식 문제를 기반 명사구의 시작, 중간, 끝은 BIO로로 태깅하는 문제로 정의하고, 변형 기반의 학습 방법을 사용하여 기반명사구를 인식하였다. 11)의 연구에서는 사용한 변형규칙은 단어(W), 품사(P), 기반구 태그(T)를 바탕으로 +/− 2의 문맥 범위에서 규칙을 적용하고 있다.

예를 들어 "다르/AJ/O+ㄴ/EM/O"와 같은 태그가 부착이 되어 있는 부분은 다음과 같은 규칙에 의해 "다르/AJ/O+ㄴ/EM/I"로 바뀌게 된다.

$$P_{-1} = AJ, P_0 = EM, W_0 = \text{ㄴ}, T_0 = O : T_0 new = I$$

이 방법에서는 사용되는 규칙의 적용이 앞/뒤 2개까지의 형태소로 제한되기 때문에, 원거리 정보를 필요로 하는 문제는 해결하기 힘들다는 단점이 있다.

12)에서는 넓은 범위의 문맥 정보를 활용할 수 있는 상

태성 기반의 모델을 사용해 기반 명사구 인식을 수행하였다. 한국어는 기반 명사구의 끝 표식 인식이 기반 명사구의 시작 표식을 인식하는 것보다 쉽다고 알려져 있다.[9] 이러한 특성으로 인하여, 상태전이의 방향성에 따라 기반구 인식의 성능이 달라지게 된다.

"[영수]는 [이런 학생]입니다."의 문장에 대해 기반 명사구 인식을 수행한다고 할 때, 오른쪽 우선 방법의 경우는 "학생/ncn+이/jp+ㅂ니다/ef"에서 "이/jp"와 "학생/ncn"을 보고 기반구의 끝을 인식하여 "[이런 학생] 입니다"의 결과를 출력한다. 또, 왼쪽 우선 방법에서는 "[영수/nq]+는/jxc 이런/mmd 학생/ncn"에서 "는/jxc"와 "이런/mmd", "학생/ncn"을 보고 기반 명사구의 시작을 인식하여 "[이런 학생]입니다"의 결과를 출력한다. 오른쪽 우선 방법은 끝 표지 인식에, 왼쪽 우선 방법은 시작 표식 인식에서 좀 더 나은 성능을 보인다. 이신목[12]은 상태 전이의 방향성을 사용하여 오른쪽 우선 방법과 왼쪽 우선 방법을 결합하여 기반 명사구 인식의 성능을 높이고 있다.

황영숙[13]은 학습 자질로 어휘 정보와 품사, 띄어쓰기 정보를 사용하여 앞, 뒤로 두 개까지의 형태소를 문맥으로 구성하고, 학습 자질의 조합에 대해 비교하여 기반구 인식에 적합한 학습 자질을 밝히고 있다. 학습 방법으로는 결정트리 방법과 메모리 기반 방법을 사용하고 있으며, 명사구(AP), 동사구(VP), 부사구(AP), 독립어구(IP)의 4가지에 대하여 기반구 인식을 수행하였다. 하지만 기반구의 범위에서 조사나 연결/종결 어미, 기호들을 제외하고 있어서 11), 12)의 연구와는 대상으로 하는 명사구의 범위가 달라서 직접적인 비교는 힘들다.

기존의 연구들은 문맥의 자질로 어휘, 품사, 띄어쓰기 정보만을 사용하고 있다. 하지만, 기반 명사구 인식은 부분적인 구문 구조를 파악하는 작업으로 앞/뒤 형태소 정보만 사용으로는 한계가 있다.

[상식/ncn]+으로/jca 이해/ncpa+되/xsv+지/ecx+않/px+는/etm [표석/ncn+들/xsv]

[웡웡거리/pvg+는/etm 바람소리/ncn]+와/jcj[희뿌옇/paa+ㄴ/etm 하늘/ncn]+은/jxt

위와 같은 경우 '는/etm'을 봤을 때, 기반 명사구에 포함되는지는 인접해서 앞/뒤에 나타나는 형태소보다는 '이해/ncpa+되/xsv'나 '웡웡거리/pvg'와 같은 어절의 중심어와 관련이 있다.

본 논문에서는 어절의 구성 정보와 어절의 중심어 정보를 사용하여 문맥 정보를 확장하여, 기반 명사구 인식에 활용하였다.

기반 명사구 인식

1 Tri-gram HMM

기반 명사구 인식 문제는 기반명사구의 외부와 내부를 구분하는 태그를 부여하는 문제로 변환할 수 있다. 즉, 입력 문장인 형태소 열을 M, 그에 해당하는 기반 명사구 태그열을 F라고 할 때, 기반 명사구 인식은 주어진 형태소 열 M에 대해서 확률값을 최대화하는 태그열 F를 찾는 것으로 식 1)과 같이 나타낼 수 있다.

$$\arg\max_F P(F \mid M) = \arg\max_F \frac{P(M \mid F) * P(F)}{P(M)} = \arg\max_F P(M \mid F) * P(F) \quad 1)$$

이때, Markov 가정을 사용해 tri-gram 모델로 표현하면 다음과 같은 식 2)을 얻을 수 있다.

$$\arg\max_F P(M \mid F) * P(F) = \arg\max_F \prod_{i=1,n} P(m_i \mid f_i) * P(f_i \mid f_{i-1}, f_{i-2}) \quad 2)$$

여기에서, m은 형태소 어휘 정보, 품사 정보, 띄어쓰기 정보와 같은 학습 자질들로 구성된다 어휘 정보를 l, 품사 정보를 t, 띄어쓰기 정보를 c라고 표시하면 m은 다음과 같이 나타낼 수 있다.

$m_i = < l_i, t_i, c_i >$

이를 이용해서 식 2)를 다시 쓰면 다음과 같다.

$$\arg\max_F \prod_{i=1,n} P(m_i \mid f_i) * P(f_i \mid f_{i-1}, f_{i-2}) = \arg\max_F \prod_{i=1,n} P(l_i, t_i, c_i \mid f_i) * P(f_i \mid f_{i-1}, f_{i-2}) \quad 3)$$

기반 명사구 태그열에 대한 형태소 정보의 조건부 확률은 $P(l_i, t_i, c_i \mid f_i)$로 나타난다. 여기에 문맥 정보가 추가 되면, 확률값 추정에서 자료 희귀성 문제가 발생을 하여 정확한 확률 값을 구할 수 없다. 확률값의 부여는 기반 명사구에 대한 태그열로부터 어휘, 품사, 띄어쓰기를 유추하는 과정으로 생각할 수 있다.

어휘, 품사, 띄어쓰기, 기반 명사구 태그 사이의 확률적인 의존성을 그림으로 표현해 보면 다음과 같다.

Fig. 1과 같은 확률 의존성에 대한 모델을 사용하여 l_i, t_i, c_i에 대해 식을 전개하면 $P(m_i \mid f_i)$는 다음과 같이 나타낼 수 있다.

$$P(m_i \mid f_i) = P(c_i \mid f_i) * P(t_i \mid f_i, c_i) * P(l_i \mid f_i, t_i, c_i)$$

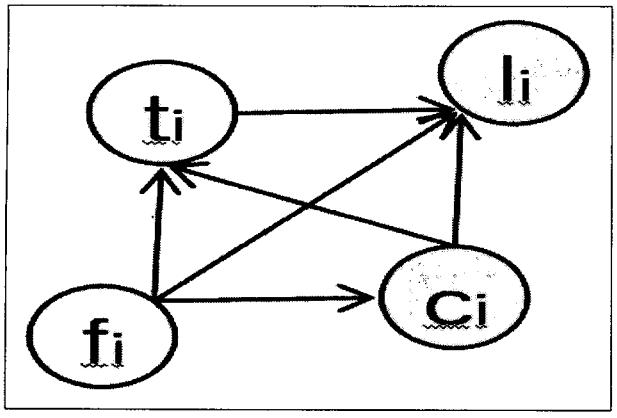

Fig. 1. 어휘(l), 품사(t), 띄어쓰기(c), 기반 명사구 태그(f)의 확률적 의존성 모델.

위 식을 이용해 tri-gram 모델을 다시 쓰면 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned}
&\arg\max_{F} P(M \mid F) * P(F) \\
&= \arg\max_{F} \prod_{i=1,n} P(m_i \mid f_i) * P(f_i \mid f_{i-1}, f_{i-2}) \\
&\qquad P(c_i \mid f_i) * P(t_i \mid f_i, c_i) * \\
&= \arg\max_{F} \prod_{i=1,n} P(l_i \mid f_i, t_i, c_i P(c_i \mid f_i) * \\
&\qquad P(t_i \mid f_i, c_i) * P(l_i \mid f_i, t_i, c_i) \\
&\qquad * P(f_i \mid f_{i-1}, f_{i-2}) \\
&=
\end{aligned} \tag{4}$$

HMM모델에서 정확도를 높이기 위해 어휘 확률과 전이확률에 문맥 정보를 추가하였다. 문맥 정보는 m_i 에 대해서 이전의 두 형태소 m_{i-1}, m_{i-2} 와 다음 형태소 m_{i+1} 를 사용하였다.

$$\begin{aligned}
&\arg\max_{F} \prod_{i=1,n} P(m_i \mid f_i, ctx1_i) * P(f_i \mid f_{i-1}, f_{i-2}, ctx2_i) \\
&= \arg\max_{F} \prod_{i=1,n} \begin{pmatrix} P(c_i \mid f_i, ctxq) * P(t_i \mid f_i, c_i, ctxb_i) * \\ P(l_i \mid f_i, t_i, c_i, ctx1_i) * \\ P(f_i \mid f_{i-1}, f_{i-2}, ctx2_i) \end{pmatrix}
\end{aligned} \tag{5}$$

여기서 ctx는 문맥 정보로, $l_{i-1}, l_{i-2}, l_{i+1}$, $t_{i-1}, t_{i-2}, t_{i+1}$ $c_{i-1}, c_{i-2}, c_{i+1}$ 의 조합으로 결정된다.

문맥 정보는 어떤 조합을 사용하느냐에 따라 성능이 많이 달라진다.[13] 문맥 정보를 적게 볼 경우는 성능에 큰 영향을 주지 않고, 문맥 정보를 너무 많이 볼 경우는 자료 희귀성 문제를 유발한다. 때문에, 적절한 문맥 정보를 선택하는 것은 어려운 일이다

본 논문에서는 3~4가지의 정보의 조합을 문맥정보로 참조하여 가장 좋은 성능이 보인 문맥 정보를 선택했다.

본 논문에서 사용한 문맥 정보는 다음과 같다.

$ctxa_i = t_{i-1}, t_{i-2}, c_{i-1}$

$ctxb_i = t_{i-1}, l_{i-1}, t_{i+1}$

$ctx1_i = t_{i-1}, t_{i-2}, c_{i-1}, l_{i-1}$

$ctx2_i = t_{i-1}, t_{i+1}, c_{i-1}$

어휘 확률의 문맥 정보의 경우, 어휘 정보가 2개 이상이 포함 될 경우는 자료 희귀성 문제가 발생하여 성능을 떨어뜨렸으며, 문맥정보에 품사 정보에 대한 정보가 추가 되었을 때가 성능의 차이가 가장 컸다.

전이 확률에 대한 문맥 정보는 어휘 정보가 들어 갔을 때 오히려 성능이 떨어지고, 문맥 정보로 3가지가 넘을 경우 어떤 조합을 사용해도 성능이 떨어졌다.

2 어절의 중심어 정보를 이용한 문맥정보 확장

한국어의 어절은 문법적인 단위로 하나 이상의 실질 형태소 열로 구성이 되거나 실질 형태소 열과 형식 형태소 열의 결합으로 구성된다. 이런 문법적인 기준으로 어절을 분리하면, 의미를 나타내는 내용어와 문법적인 역할을 나타내는 기능어로 나눌 수 있다.

기능어의 중심어와 내용어의 중심어정보를 각각 fh, ch로 표시하고, 중심어가 아닌 부분을 fx, cx로 표시하며, 기능어와 내용어의 중심어가 같은 부분은 cf로 표시를 하면 다음과 같이 나타낼 수 있다.

1) 삶/ncn	+	E	ch
2) 의/jcm	.	O	fh
3) 모습/ncn	+	E	cf
4) ,/sp	.	O	fx
5) 도시/ncn	+	E	ch
6) 의/jcm	.	O	fh
7) 모습/ncn	*	E	cf
8) 사라지/pvg	+	I	ch
9) ㄴ/etm	.	I	fh
10) 시냇물/ncn	*	E	cf
11) 불볕/ncn	*	I	cf
12) 더위/ncn	+	E	ch
13) 가/jcs	.	O	fh

어절에 대한 문맥 정보는 현재 어절의 내용어와 기능어에 대한 중심어의 품사 정보를 사용하여 나타내었다. 현재 형태소가 내용어의 중심어일 경우는 기능어의 중심어를, 기능어의 중심어일 경우는 다음 어절의 내용어의 중심어를 문맥 정보로 하고, 중심어가 아닐 경우는 현재 형태소가 속해 있는 부위의 중심어 정보를 사용하여 문맥 정보를 나타내었다(Fig. 2).

어절 문맥 정보를 사용하여 코퍼스를 나타내면 다음과 같다.

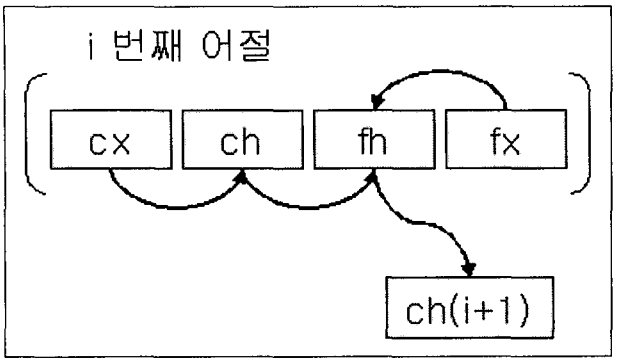

Fig. 2. 어절 문맥 정보.

1) 삶/ncn	+	E	ch	jcm
2) 의/jcm	.	O	fh	ncn
3 모습/ncn	+	E	cf	ncn
4) ,/sp	.	O	fx	ncn
5) 도시/ncn	+	E	ch	jcm
6) 의/jcm	.	O	fh	ncn
7) 모습/ncn	*	E	cf	pvg
8) 사라지/pvg	+	I	ch	etm
9) ㄴ/etm	.	I	fh	ncn
10) 시냇물/ncn	*	E	cf	ncn
11 불볕/ncn	*	I	cf	ncn
12) 더위/ncn	+	E	ch	jcs
13) 가/jcs	.	O	fh	paa

어절의 중심어를 P, 기능어와 내용어의 중심어 정보를 h라고 할 때, 어휘 확률 $P(m_i \mid f_i)$ 의 어휘 정보는 $m_i = < l_i, t_i, c_i, h_i, P_i >$로 표현된다.

이때, 어휘와 품사, 띄어쓰기, 어절 문맥, 어절의 중심어 정보를 위의 그림과 같은 확률적 의존성 모델로 생각하면 다음과 같은 식을 얻을 수 있다.

$$\begin{aligned} & P(m_i \mid f_i) \\ & = P(c_i, h_i \mid f_i) * P(t_i \mid f_i, c_i, h_i) * \\ & \quad P(P_i \mid f_i, t_i, c_i) * P(l_i \mid f_i, t_i, c_i, P_i) \end{aligned} \quad 6)$$

이를 문맥 정보를 사용하여 식을 확장하면, 다음과 같다.

$$\begin{aligned} & \underset{F}{\operatorname{argmax}} \prod_{i=1,n} P(m_i \mid f_i, ctx1_i) * P(f_i \mid f_{i-1}, f_{i-2}, ctx2_i) \\ & = \underset{F}{\operatorname{argmax}} \prod_{i=1,n} \begin{pmatrix} P(c_i, h_i \mid f_i, ctxa_i) * \\ P(t_i \mid f_i, c_i, h_i, ctxb_i) * \\ P(P_i \mid f_i, c_i, h_i, t_i, ctxc_i) * \\ P(l_i \mid f_i, t_i, c_i, h_i, P_i, ctx1_i) * \\ P(f_i \mid f_{i-1}, f_{i-2}, ctx2_i) \end{pmatrix} \end{aligned} \quad 7)$$

본 논문에서 사용한 문맥 정보는 다음과 같다.

$$ctxa_i = t_{i-1}, l_{i-1}, t_{i+1}$$
$$ctxb_i = l_{i-1}, t_{i-1}, t_{i+1}, P_i$$
$$ctxc_i = l_{i-1}, P_{i-1}, t_{i+1}, t_{i-1}$$
$$ctx1_i = t_{i-1}, t_{i-2}, c_{i-1}, l_{i-1}$$
$$ctx2_i = t_{i-1}, t_{i+1}, c_{i-1}$$

문맥 정보의 선택은 기본적으로 HMM1에서 사용한 문맥 정보를 바탕으로 새로 추가한 어절의 중심어에 대한 문맥을 추가하여 적절한 문맥 정보를 선택하였다.

본 논문에서는 어절의 형식 형태소와 실질형태소를 문법적인 역할을 바탕으로 내용어와 기능어열의 두 부분으로 구분하고 각각의 중심어에 대한 정보를 활용하여 문맥 정보를 확장하였다.

실험 및 평가

1. 실험집합

실험에 사용한 코퍼스는 한국과학기술원 국어정보베이스의 구문 표지 부착 코퍼스에서 기반 명사구를 추출한 코퍼스이다. 이는 총 12,083문장(322,057 형태소)으로 구성이 되어 있으며, 기존 연구 11), 12), 13)에서 사용한 것과 동일한 코퍼스이다. 품사는 총 54 품사로 구성된 KAIST 품사집합으로 품사 부착되어 있다.

평가는 정확률과 재현률, F-value를 사용하여 평가를 하였다.[11,12]

$$\text{Precision} = \frac{\text{출력한 명사구 중 정답의 수}}{\text{출력한 명사구 전체 수}}$$

$$\text{Recall} = \frac{\text{출력한 명사구 중 정답의 수}}{\text{정답 집합의 명사구 수}}$$

$$\text{F-value} = \frac{2 * precisoin * recall}{(precision + recall)}$$

2. 표기 형식에 따른 성능 비교

각 형식에 대한 비교 실험의 결과는 다음과 같다. 확률모델은 HMM에 문맥 정보를 사용한 HMM1(수식 5)을 적용하였으며, 확률값의 학습을 위해 8,903문장을 사용하였고, 평가용으로 3,180문장을 사용하였다(Table 1).

Table 1. 표기 형식에 따른 결과

	Precision	Recall	F-value
IOB1	90.98	90.53	90.76
IOB2	90.33	90.56	90.44
IOE1	86.71	88.45	87.57
IOE2	93.48	93.18	93.33
BIOES	91.79	91.64	91.72

OB2보다는 IOB1의 인식 결과가 좋으며, IOE1보다는 IOE2의 인식 결과가 좋다.

한국어의 경우 기반 명사구의 끝을 인식 하는 문제가 시작을 인식하는 문제보다 쉽다고 알려져 있다.[9] IOE2의 경우는 기반 명사구의 끝이 항상 'E'로 태깅이 되어 끝부분을 비교적 정확하게 인식 하기 때문에 끝부분이 'E'나 'I'로 표지되어야 하는 IOE1보다 좋은 성능을 보이고 있다.

시작 부분을 인식하는 IOB의 경우는 오히려 시작 부분이 'B' 또는 'I'로 태깅되는 IOB1의 결과가 IOB2의 결과보다 좋을 것을 볼 수 있다.

IOB1과 IOE2, BIOES를 비교 해 봤을 때는 IOE2의 경우가 훨씬 좋은 성능을 보이고 있다. BIOES의 경우는 IOB1보다는 좋은 성능을 보이지만 IOE2보다는 성능이 떨어진다. 이는 한국어의 기반 명사구 인식의 문제는 기반 명사구의 끝 표식을 인식하는 문제가 시작 표식을 인식 하는 문제보다 중요하다는 것을 말해 준다.

본 논문에서는 IOE2 형식을 사용해 실험을 하였다.

3. 실험결과

기본 tri-gram HMM과 문맥 정보를 통해 확장된 HMM, 어절에 대한 중심어 정보(cf head)가 추가된 HMM2와 어절의 중심어 정보가 추가된 HMM2에 대해서 각각 IOE2 방식으로 태깅을 수행하였다(Table 2).

문맥 정보를 사용하지 않은 기본 HMM0의 경우는 F-value가 83.96%으로 성능이 매우 낮다. 기본 모델에서 앞뒤 형태소의 품사와 어휘 정보만을 문맥 정보로 추가해도 크게 성능이 올라가고 있다.

문맥의 선택에 따라서 성능이 차이를 보이기 때문에, 문맥의 선택도 중요한 문제이다. 본 논문에서는 문맥 정보와 학습 자질의 선택에 대한 문제는 향후 연구로 남겨놓고, 몇 가지 조합 중에서 가장 좋은 성능을 보였던 문맥 정보를 사용하여 실험을 수행하였다.

문맥 정보로 단순히 어절의 내용어와 기능어에 대한 중심어 정보만을 추가했을 때, 정확률이 높아지는 것을 알 수 있다. 이것은 인접 형태소 사이의 관계만 사용하는 것보다는 간단한 구문적인 정보들을 같이 사용하는 것이 기반 명사구 인식에서 효과적이라는 것을 의미한다.

Table 2. 문맥 정보에 따른 실험 결과

	Precision	Recall	F-value
HMM0	84.74	83.20	83.96
HMM1	93.48	93.18	93.33
HMM1+cf_head	94.09	92.95	93.52
HMM2	94.31	93.21	93.76

형태소의 어절 내부에서의 역할 정보 외에 어절의 중심어 정보를 사용했을 때는 정확률과 재현률이 모두 높아지는 것을 볼 수 있다. 어절의 중심어 정보를 사용할 경우 문맥정보의 범위가 어절에 대한 정보로 커져서 형태소 단위로 문맥 정보를 보는 것보다 문맥 정보가 확장되는 효과가 있다(Table 3).

이신목[12]의 상태 기반 방법에서는 정확률과 재현률이 차이가 많이 나지만, HMM에서는 정확률과 재현률의 차이가 크지 않은 것을 볼 수 있다.

어절의 중심어 정보와 인접 어절에 대한 정보를 문맥 정보로 활용 했을 때, F-value는 93.76%로 양재형[11]보다는 2.75%, 이신목[12]의 연구 보다는 2.24%의 성능향상을 보였다. 황영숙[13]의 연구와 비교 했을 때, 0.54%의 성능향상을 보였다.

결 론

본 논문에서는 기반 명사구의 인식을 위해 tri-gram HMM을 제안하였다. 학습 자질로는 기존 연구에서 사용되던 어휘, 품사, 띄어 쓰기 정보 외에 어절의 구성 정보와 중심어 정보를 사용하였다. 어절은 구문 구조의 단위로 내용어와 기능어의 두 부분으로 나누어 진다. 어절의 구성 정보는 내용어에 대한 어절의 중심어와 기능어에 대한 어절의 중심어 정보가 된다.

문맥 정보를 어절의 중심어 정보로 확장을 하여 IOE2 방식으로 기반 명사구 인식을 시도 하였을 때, F-value는 93.76%가 나왔다.

한국어에서는 한 어절에서 중심어의 문법적인 역할을 바꿔주는 파생 조사나 계사와 같은 구문적인 특성을 갖는 형태소들이 있어서 구문적인 역할의 결정이 어절 단위로 제한되지 않는다. 어절 단위의 문맥 정보를 사용하기 위해서는 기본 단위를 문법적인 단위로 재 설정해 주는 작업이 필요하다.

영어의 경우 기반 명사구는 후치 수식구를 포함하지 않는 명사구로 한정이 된다. 하지만, 한국어의 경우는 전치/후치 수식어구의 구분이 없어서 기반 명사구도 길이가 길어지게 된다.[12] 이런 문제는 기반 명사구의 길이를 증가시킬

Table 3. 기존 연구와의 비교

	Precision	Recall	F-value
양재형[11]	91.8	90.7	91.25
이신목[12]	92.55	90.90	91.71
황영숙[13]	93.58	92.92	93.25
HMM1	93.48	93.18	93.33
HMM2(어절 문맥)	94.31	93.21	93.76

뿐 아니라 기반 명사구의 시작 표지에 대한 애매성도 증가시킨다. 본 논문에서 IOB와 IOE 표현 형식을 사용해 실험한 결과를 보면, 시작 표지의 경우 따로 구분 해주지 않았을 때가, 끝 표지는 구분 해 주었을 때가 좋은 성능을 보이고 있다.

향후 연구로 학습 자질의 조합에 대한 연구가 필요하다. 또, 상태 기반 학습 모델이나 MEMT, SVM과 같은 학습기의 적용에 대한 연구가 필요하다.또, 본 논문에서는 기반 명사구 인식만을 대상으로 하고 있기 때문에, 어절의 구성 정보와 중심어 정보가 다른 기반구 인식에도 적용이 가능한지에 대한 연구도 필요하다.

REFERENCES

1) Erik F, Tjong Kim Sang(1999) : *Representing text chunks. In Proceedings of EACL'99*
2) Erik F, Tjong Kim Sang(2002) : *Memory-Based Shallow Parsing, In Journal of Machine Learning Research, volume 2*
3) Erik F, Tjong Kim Sang, Herve Dejean(2001) : *Introduction to the CoNLL-2001 Shared Task : Clause Identification. In Proceedings of CoNLL-2001, Toulouse, France*
4) Uchimoto, Kiyotaka ; Qing Ma, Masaki Murata, Hiromi Ozaku, Hitoshi Isahara(2000) : *Named Entity Extraction Based on A Maximum Entropy Model and Transformation Rules. In Procedings of the ACL2000*
5) Ramshaw, Lance A, Mitchell P, Marcus(1995) : *Text chunking using transformation-based learning, In Proceedings of the 3rd Workshop on Very Large Corpora*
6) Abney, Steven(1990) : *Rapid Incremental Parsing with Repair. In Proceedings of the 8th New OED Conference*
7) Kudo Taku and Yuji Matsumoto(2001) : *Chunking wth Support Vector Machines. In : "Proceedings of NAACL 2001*
8) Tong Zhang, Fred Damerau, David Jhonson(2001) : *Text Chunking using Regularized Winnow, In Proceedings of CoNLL-2001, Toulouse, France*
9) 강인호 · 전수영 · 김길창(2000) : 최대 엔트로피 모델을 이용한 한국어 명사구 추출, 제 *12* 회 한글 및 한국어 정보처리 학술대회
10) 서충원 · 최용석 · 최기선(2001) : 세로 말모듬 변환과 관리도구에 관한 연구, 2001년 한국인지과학회 춘계 학술대회.
11) 양재형(2000) : 규칙 기반 학습에 의한 한국어의 기반 명사구 인식. 정보과학회 논문지 제 *27* 권 제 *10* 호
12) 이신목(2001) : 방향성을 이용한 상태 기반의 한국어 기반 명사구 인식, 한국과학기술원 석사학위 논문
13) 황영숙 · 정후중 · 박소영 · 곽용재 · 임해창(2002) : 자질집합선택 기반의 기계학습을 통한 한국어 기본구 인식의 성능향상, 정보과학회 논문지, *Vol 29. Number 9*