확률과 규칙을 사용한 품사 태깅

신 상 현　이 근 배　홍 남 희　이 종 혁
포항공과대학교 전자계산학과

Part-of-speech Tagging using Probability and Rules

Sanghyun Shin　Geunbae Lee　Namhee Hong　Jong-Hyeok Lee
Dept. of Computer Science and Engineering, POSTECH

요 약

한국어에 있어서 품사 태깅은 형태소 분석결과의 모호성을 제거하는 것으로, 기존의 방법을 보면, 확률을 이용하는 방법, 퍼지망을 이용하는 방법, 신경망을 이용하는 방법등 다양하다. 현재의 주류가 확률을 이용한 방법이다. 하지만, 이 방법은 제한된 윈도우 크기와 품사사이의 관계만을 이용한다는 한계점을 지니고 있다. 본 논문에서는 확률을 이용한 결과에, 확률에서 다루지 못하는 범위에 대하여 자동 학습된 규칙을 추가로 적용하여 이 한계점을 극복한다. 규칙 적용시 윈도우 크기를 임의로 정할 수 있고, 품사사이의 관계외에 어절사이의 관계도 고려할 수 있으므로 확률적 방법이 다루지 못하는 부분에 대하여 어휘단계에서의 교정이 가능하게 된다. 현재 20가지 정도의 규칙을 수작업 코딩하여 사용한 결과 확률적 방법의 성능을 3% 정도 향상시킬 수 있었으며, 앞으로 규칙생성을 자동학습할 경우 더 큰 성능향상을 기대해 볼 수 있다.

1. 서론

품사 태깅시스템은 입력문장에 대한 각 단어나 어절에 대하여 가능한 여러가지 품사(Part-of-speech)중 하나를 선택함으로써 모호성을 제거하는 것이다. 이는 자연어 처리의 기초 단계로서, 자동인덱싱, 문자/음성 인식, 정보추출등에 응용될 수 있다. 기존의 방법을 보면, 확률을 이용한 방법[2,3,7,9], 규칙을 이용한 방법[5,6,10],그리고 요즈음들어 신경망[4]이나 퍼지망[1]을 이용한 새로운 방법이 모색되고 있다. 이중 확률적 방법이 주로 이용된다. 한국어에 있어서 태깅은 한국어가 교착어의 특성을 가져 형태소의 활용범위가 넓어 영어에 비하여 어려운 점을 지니고 있다. 한국어 태깅시 형태소 분석기의 결과에 대하여 모호성을 제거하여 어절단위[2,3]나 형태소 단위[1]로 태깅을 한다. 본 논문에서는 기존의 확률적 방법에 대하여 추가적으로 학습된 규칙을 적용하여, 확률적 방법이 지니는 한계를 언어학적 측면을 최대한 고려하여 극복하고자 한다.

2. 태그집합

태그집합을 결정할 때 고려해야 할 것은 태깅시스템이 어떤 응용에 적용될 것인가하는 것이다.본 태깅시스템은 정보추출(information extraction)에 적용될 것을 고려하여, 형태소 단위의 태그집합 총 19개로 표1과 같이 정하였으나 앞으로 응용에 따라 변경될 여지가 충분히 있다.또, 응용에 관계없이 표준화된 태그집합을 사용할 수도 있다. 본 시스템은 어절 단위의 태깅으로, 형태소단위의 태그를 어절 단위로 결합하여 사용한다. 2만어절의 말뭉치를 이용하였을 때, 형태소 단위의 조합에 의해 생성되는 어절 단위의 태그는, 약 140개가 생성되었다.

태 그	품　　사	태 그	품　　사
MP	고유명사	jC	격조사
MD	의존명사	jJ	접속조사
MC	보통명사	jS	보조사
S	수 사	mC	연결서술형어말어미
T	대명사	mT	종결서술형어말어미
D	동 사	mj	전성어말어미
H	형용사	-	접미사
G	관형사	+	접두사
B	부사	e	선어말어미
y	조용보조어간		

표1 형태소 태그집합

3. 확률을 이용한 방법

확률을 이용하여 형태소 분석기 분석결과의 모호성을 해소하게 되는데, 이때, 사용하는 모델이 HMM이다.태깅이란 한 문장이(어절의 리스트-$e_{1,n}$)이 주어졌을 때,이 문장에 대한 확률을 최대로 하는 태그의 리스트($t_{1,n}$)를 구하는 함수로 볼 수 있다. 이를 수식으로 나타내면 다음과 같

다[7].

$$T(e_{1,n}) = argmax_{t_{1,n}}\ P(T_{1,n} = t_{1,n}|E_{1,n} = e_{1,n}) \quad (1)$$

여기서 확률변수를 생략하면

$$T(e_{1,n}) = argmax_{t_{1,n}}\ P(t_{1,n}|e_{1,n}) \quad (2)$$

$$= argmax_{t_{1,n}}\ P(t_{1,n}, e_{1,n})/P(e_{1,n}) \quad (3)$$

여기서 $P(e_{1,n})$은 모든 $t_{1,n}$에 대하여 상수이므로, 다음과 같이 된다.

$$T(e_{1,n}) = argmax_{t_{1,n}}\ P(t_{1,n}, e_{1,n}) \quad (4)$$

$$= argmax_{t_{1,n}}\ \prod_{i=1}^{n}\ P(t_i|t_{1,i-1}, e_{1,i-1})P(e_i|t_{1,i}, e_{1,i-1}) \quad (5)$$

여기에 현재의 어절은 현재의 태그에만 의존하고, 현재의 태그는 이전 2개까지의 태그에만 의존한다는 다음과 같은 trigram의 가정을 적용하면,

$$P(t_i|t_{1,i-1}, e_{1,i-1}) = P(t_i|t_{i-2}, t_{i-1}) \quad (6)$$

$$p(e_i|t_{1,i}, e_{1,i-1}) = P(e_i|t_i) \quad (7)$$

태깅의 수식은

$$T(e_{1,n}) = argmax_{t_{1,n}}\ \prod_{i=1}^{n}\ P(e_i|t_i)P(t_i|t_{i-2}, t_{i-1}) \quad (8)$$

이 된다. 여기에 현재의 어절은 현재의 태그에 의존한다는 조건을 현재의 태그는 현재의 어절에 의존한다는 조건으로 변화를 주면, 이론적 순수성은 사라지지만, 확률태깅에서 주로 사용되는 수식인

$$T(e_{1,n}) = argmax_{t_{1,n}}\ \prod_{i=1}^{n}\ p(t_i|e_i)P(t_i|t_{i-2}, t_{i-1}) \quad (9)$$

이 얻어 진다. 이식은 MLE(Maximum Likelihood Estimation)추정을 사용하면 다음과 같이 계산된다.

$$p(t_i|e_i) = f(e_i, t_i)/f(e_i) \quad (10)$$

$$\mathrm{p}(t_i|t_{i-2}, t_{i-1}) = f(t_{i-2}, t_{i-1}, t_i)/f(t_{i-1}, t_i) \quad (11)$$

(여기서 f는 빈도수)

$f(e_i)$과 $f(e_i, t_i)$은 어절테이블을 이용해서 빈도수를 구하고, $f(t_{i-2}, t_{i-1}, t_i)$과 $f(t_{i-1}, t_i)$은 각각 trigram 테이블과 bigram 테이블을 이용하여 구한다. 위의 수식에서 argmax에 해당하는 가장 큰 확률값을 가지는 태그의 리스트를 얻기 위하여 Viterbi Algorithm을 사용한다.

4. 규칙을 이용한 방법

확률을 이용한 태깅 방법은 현재의 어절과 현재의 태그, 그리고 현재의 태그과 앞의 2개 혹은 1개의 태그만을 고려하는 제약을 가지고 있다. 이는 실제 언어적 관점에서 본다면, 매우 단순화된 시각이다[11]. 언어학적 관점을 최대한 고려하기 위하여, 본 논문에서는 확률모델에서 고려하지 못하는 앞과 뒤의 어절사이의 관계,형태소 사이의 관계 및 현재의 태그와 앞뒤의 임의의 범위의 태그사이의 관계등을 고려한다.

*jC n 1 e *가 c *와 *jJ
*jC n 1 e *은 c *과 *jJ
MC n 1 t M* c 새 G
D* p 1 e *아 c 있* H*
D* p 1 e *어 c 있* H*
MC* p 1 t G c 바* MD*
D* n 1 t MD* c 한 G

표2 규칙 리스트

표2는 본 시스템에서 현재 사용하는 규칙의 일부이다. 이 규칙에서 첫번째 열은 현재 어절의 태그를 말한다. 두번째 열부터 4 번째 열까지는 고려하고자 하는 윈도우로, p는 이전의 어절이나 태그를 고려하는 것을 의미하며, 세번째 열은 윈도우 크기를, 네번째 열은 어절(e)을 고려할 것인지 태그(t)를 고려할 것인지를 말한다. 다섯번째 열은 윈도우에의 고려대상인 어절이나 태그를 말한다.여섯번째 열은 현재의 어절을 고려한다는 표시이고 일곱번째 열은 현재의 어절, 그리고 마지막 열은 변경될 태그이다.

규칙예) *jC n 1 e *가 c *와 *jJ

이 규칙은 현재의 어절이 "와"로 끝나고, 현재의 태그가 격조사일 경우 다음의 어절이 "가"로 끝난 다면, 현재 어절의 태그를 격조사에서 접속조사로 바꾸라는 의미이다. 위의 규칙은 확률태깅의 적용결과 접속조사 "와"가 격조사로 태그되는 경향이 있으므로 이를 수정하기 위한 것이다.

적용예) 입력문장 : 호랑이와 토끼가 말한 내용을 생각하자

확률방법에 의한 태깅 결과

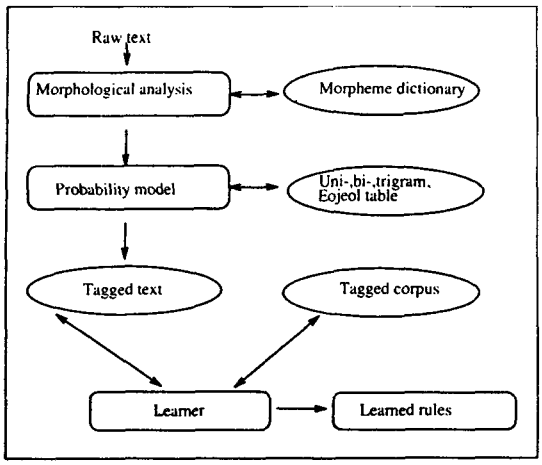

그림 1: 규칙 자동 학습

[호랑이와]:[MCjC] [토끼가]:[MCjC] [말한]:[MCymj] [내용을]:[MCjC] [생각하자]:[MCymT]

규칙을 추가한 방법에 의한 태깅 결과
[호랑이와]:[MCjJ] [토끼가]:[MCjC] [말한]:[MCymj] [내용을]:[MCjC] [생각하자]:[MCymT]

규칙예) D* p l e *어 c 있* H*

위의 규칙은 현재의 어절이 "있"으로 시작하고 동사로 태깅되었는데, 앞의 어절이 "어"일 경우, 동사를 형용사로 바꾸라는 의미이다. 위의 규칙은 "있다" 앞에 보조적 연결어미 "-어"로 끝나는 어절이 올경우 이 "있다"는 형용사이기 때문이다.

적용예) 입력문장 : 운동장도 텅 비어 있습니다

확률방법에 의한 태깅 결과
[운동장도]:[MCjS] [텅]:[B] [비어]:[HmC] [있습니다]:[DmT]

규칙을 추가한 방법에 의한 태깅 결과
[운동장도]:[MCjS] [텅]:[B] [비어]:[HmC] [있습니다]:[HmT]

위와 같은 방식으로 규칙을 이용 확률모델에서 다루지 못하는 범위에 대하여 규칙모델이 처리할 수 있게 되어, 태깅의 정확률이 올라가게 된다.

5. 규칙의 자동 획득 방법

그림1에서 보듯이 확률방법을 이용 문장단위로 태깅한다. 이 태깅 결과를 태그된 corpus와 비교하여 규칙 템플

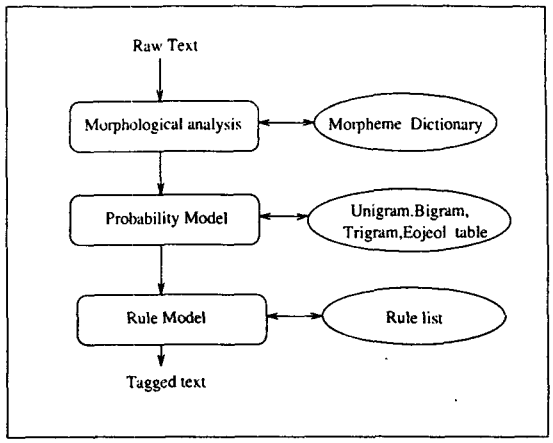

그림 2: 태깅시스템의 구조

리트을 참조해 가면서 규칙을 생성해 내는데, 이때 가장 적은 에러수를 낳는 규칙을 선택하게 된다. 이런식으로 기준치 이하의 에러수가 나올 때까지 학습을 계속한다[5,6].

6. 시스템 구성

본 논문에서 구현된 품사 태거는 그림2와 같이 크게 형태소 분석기, 확률태거, 규칙을 이용한 확률태거의 오류수정기의 3 부분으로 구성된다. 형태소 분석기는 입력 어절에 대하여 형태소 단위로 가능한 모든 품사를 생성한다.

확률태거는 형태소 분석기의 여러가지 가능한 품사에 대하여 HMM model을 이용하여 이들중 하나를 선택하게 된다.

규칙모델에서는 확률모델에서 다루지 못하는 범위나,어절사이의 관계를 이용하여, 확률모델의 결과를 수정하여 태깅의 정확률을 높인다.

7. smoothing 및 미등록어 처리

태깅된 말뭉치를 이용, 확률태깅을 위한 정보를 모을 때, 정보의 부족으로 문제점이 발생하게 된다. 즉 실제 적용시 말뭉치에 없는 정보가 출현하기 때문에 수식이 0이 되는 문제점이 발생하게 된다. 이를 해결하기 위하여 trigram에 대하여 bigram정보와 unigram정보를 추가로 적용한다[8].이때, 문맥확률은 다음과 같이 된다.

$$p(t_i|t_{i-1},t_{i-2}) = \lambda_1 * p(t_i|t_{i-1},t_{i-2}) + \lambda_2 * p(t_i|t_{i-1}) + \lambda_3 * p(t_i) \quad (12)$$

(여기서 $\lambda_1 + \lambda_2 + \lambda_3 = 1$)

본시스템에서는 여러 시험결과 $\lambda_1 = 0.5, \lambda_2 = 0.4, \lambda_3 =$

0.1로 놓았다.

어절에 관한 자료가 없을 경우 빈도수에 1을 더하여 이를 해결한다. 어절 테이블에 어절이 없을 경우 빈도수는 1이 되고, 있을 경우 빈도수는 하나 증가 하게 된다.

형태소 분석 결과 분석 불가능이 나올 경우 우선, 어절 테이블을 탐색하여 해당 어절이 있을 경우 빈도수가 가장 높은 태그를 선택한다.만약 어절 테이블에 없을 경우 trigram 테이블을 탐색하여 빈도수가 가장 높은 태그로 선택하게 된다.

8. 실험결과 및 분석

테스트 대상	확률모델	확률+규칙모델	성능향상
국민교육헌장 (133어절)	116 (87.2%)	123 (92.5%)	7 (5%)
국민학교교과서 (2123어절)	1898 (89.4%)	1958 (92.2%)	60 (2.8%)
총 계 (2256어절)	2014 (89.3%)	2081 (92.2%)	67 (2.9%)

표3 태그시스템 실험결과

확률정보를 얻기위한 태깅된 말뭉치로 국민학교 교과서 일부 약 2만 어절을 이용하였으며, 태거의 성능을 측정하기 위하여, 국민 교육 헌장과 국민학교 교과서 일부 약 2천 어절을 이용하였다. 확률모델에 규칙모델을 추가로 적용하여 약 3%의 성능향상을 볼 수 있었다. 이는 규칙을 확률모델의 잘못된 경우만 보정해 주도록 자동생성하면 훨씬 더 좋은 성능 향상을 기대할 수 있으리라 보며, 현재 이에 대한 실험이 진행되고 있다.

9. 결론

본 논문에서는 확률태깅이후에 규칙에 기반을 둔 태깅을 적용함으로써 태깅의 정확률을 높였다. 앞으로 본 시스템의 개선을 위해서는 다음의 사항이 필요하다.

첫째,현재 추출한 규칙들은 말뭉치와 확률적 방법에 의한 결과를 비교하여 사람이 만들것으로, 아직 규칙을 학습하는데 자동화를 실현하지 못하였다. 태그된 corpus와 비교하여 자동으로 생성시키는 부분을 추가시키는 것이 필요하다. 이러한 자동화된 생성규칙을 이용, 사람이 생각하지 못했던 유용한 규칙을 생성해 낼 수 있다.

둘째, 적은 말뭉치로 인하여 확률태깅의 비교적 낮은 정확률을 높이기 위하여 Baum-Welch의 HMM훈련 알고리즘을 추가로 적용하는 것이 필요하다.

셋째, 형태소 분석기가 분석을 하지 못하는 어절에 대하여 형태소 분석기의 부분적인 정보를 이용 최대한 추측의 확률을 높이는 것이 필요하다.

넷째, 태깅시스템이 형태소 분석기 결과에 의존하므로 형태소 분석기에서 잘못된 결과를 출력할 경우 무력하게 된다. 이에 대한 수정을 하는 부분이 필요하다.

참고 문헌

[1] 김 재훈, 조정미, 김창현, 서정연, 김길창, "퍼지망을 이용한 한국어 품사 태깅", 한글 및 한국어 정보 처리 학술 대회 학술 발표 논문집, 1993.

[2] 박 혜준, 윤 준태, 송 만석, "말뭉치 품사꼬리달기 시스템 구현", 한국정보과학회 봄 학술발표논문집, 1994.

[3] 이 운재 , "한국어 문서 태깅 시스템의 설계 및 구현", 한국과학기술원 석사학위논문, 1993.

[4] Benello, J.; Mackie, A. W.; Anderson, J.A., "Syntactic catogory disambiguation with neural networks", *Computer speech and language*, vol.3, 1989.

[5] Brill, E., "A simple rule-based part of speech tagger", *Proceedings of the DARPA speech and natural language workshop*, 1992.

[6] Brill, E., "A report of recent progress in transformation-based error-driven learning", *Proceedings of the DARPA workshop on human language technology*, March, 1994.

[7] Charniak, E.; Hendrickson, C.; Jacobson, Neil.; Perkowitz, M., "Equations for part-of-speech tagging", *Proceedings of the eleventh national conference on artificial intelligence*, July, 1993.

[8] Charniak, E., "Statistical language learning", *A Bradford book. The MIT press* , 1993.

[9] Church, K.,"A stochastic parts program and noun phrase parser for unrestricted text", *Second conference on Applied Natural language processing* 1988.

[10] Klein, s.; Simmons, R.F., "A computational approach to grammatical coding of English words", *Journal of association for computing machinery* vol.10, 1963.

[11] Tapanainen, P.; Voutilainen, A., "Tagging accurately-Don't guess if you know", *conference on Applied Natural language processing* 1994.