퍼지망을 이용한 한국어 품사 태깅*

김재훈, 조정미, 김창현, 서정연, 김길창
한국과학기술원, 전산학과

A Part-of-Speech Tagging Using Fuzzy Network

Jae-Hoon Kim, Jeong-Mi Cho, Chang-Hyun Kim,
Jungyun Seo, Gil Chang Kim
Department of Computer Science, KAIST

요 약

본 논문은 퍼지 망(Fuzzy Network)의 개념을 도입하여 한국어 단어의 품사 태깅에 관한 새로운 모델을 제시하고자 한다.

한국어 단어의 품사 태깅이란 여러 개의 품사를 가진 단어가 한국어 문장 속에 나타났을 때, 단어의 품사를 올바르게 결정하는 것이다.

여기서 가장 기본적인 문제는 여러 가지의 태그를 포함하고 있는 단어들의 나열을 어떻게 퍼지 망으로 표현하는가 하는 문제이다. 본 논문에서는 한국어 품사를 태깅할 때 사용한 퍼지 망을 정점(vertex)으로 단어 품사의 퍼지 집합을 표현하고, 연결선(edge)으로 품사와 품사 간의 퍼지관계를 표현한다. 일단 퍼지망으로 표현되면, 퍼지망에서의 최적의 경로를 찾는 문제와 동일하게 풀 수 있다. 일반적으로 퍼지 망에서 최적의 경로를 찾는 문제는 dynamic programming 방법에 의해서 효과적으로 해결할 수 있다.

약 2만 6천개의 형태소를 실험 데이타로 하여 실험한 결과 , 전체적인 품사 태깅 정확률은 95.6%로 비교적 좋은 결과를 보였다. 앞으로 좀 더 세분화된 태그 집합과 정확히 태깅된 실험 데이타로부터 추출된 소속함수를 이용한다면, 더 좋은 결과를 기대할 수 있다.

I. 서론

말은 그 쓰임새에 있어 어떤 공통적인 성질을 가지고 있다. 이들 공통적인 성질, 또는 다른 성질과 구별되는 표시를 **태그**(tag)라고 한다. 예를 들면, 품사(part-of-speech, POS)가 대표적인 것이다. 품사는 문법적 성질이 공통된 단어끼리 모아 놓은 단어의 갈래를 말한다. 품사는 형태소 해석이나 구문 해석을 하는데 있어서 중요한 정보이다. 그러나, 일반적으로 이들 품사는 주어진 단어에 대해서 유일하게 결정되는 것이 아니며 많은 단어는 품사의 애매성을 가지고 있다. 그러나, **그 단어가 문장 속에서 다른 단어들과 함께 사용되었을 경우에는 품사의 애매성을 줄일 수 있다**. 예를 들어, "몇"은 관형사인

*본 연구는 한국통신 장기 기초 연구과제의 하나인 "대화체 기계번역에 관한 연구"의 일부임.

동시에 수사이나, "학생이 몇 명입니까?"에서 "몇"은 관형사로 쓰인다. 또, 이와는 반대로 **단어가 문장 속에서(다른 단어와 결합하면서) 그 애매성이 늘어나는 경우도 있다**. 이는 애매성을 포함하고 있는 두 단어(또는 형태소)들이 결합했을 때에 발생된다. 이와 같은 특성은 한국어와 같은 교착어에서 자주 발생되는 현상이다. 예를 들면, "나는 학교에 가는 중이다."라고 하는 예문에서 '나'는 대명사(I)이고 동사(sprout)이다. 또, 기능어로서 '는'은 보조사이고 관형형어미이다. 이 두 단어가 결합하면 전체 4종류의 가능성이 발생되나, 한국어에서 동사와 조사, 대명사와 관형형 어미가 결합할 수 없기 때문에 두 가지만 발생할 것이다. 그러나, 실제로는 "나는"은 "나(I)+는(보조사)", "날(fly)+는(관형형어미)", "나(sprout)+는(관형형어미)" 3가지가 발생된다. 이와 같은 현상은 "날다"라고 하는 단어는 관형형어미 '는'과 결합하면서 불규칙 활용을 하기 때문에 일어나는 현상이다.

품사 태깅은 여러 가지의 품사를 가진 단어가 문장 속에 나타났을 때에 단어의 품사를 올바르게 결정하는 것이라고 할 수 있다. 태깅은 여러 가지 다른 방법에 의해서 처리되어 왔으며, 크게 규칙에 의한 방법[6, 10]과 통계적인 방법[3, 8, 9] 그리고 신경망을 이용한 방법[5]이 있다.

본 논문은 퍼지 망(Fuzzy Network)의 개념을 도입하여 한국어 단어(혹은 형태소)의 품사 태깅에 관한 새로운 퍼지 모델을 제시하고자 한다.

II. 한국어 품사 태깅에서의 문제점

영어는 어절과 단어의 구별이 없으나, 한국어의 경우에는 일반적으로 하나 이상의 단어가 모여 하나의 어절을 이룬다. 따라서 한국어에서 단어의 품사를 태깅하기 위해서는 어절을 단어 단위로 분리하는 일이 부가적으로 필요하다. 예를 들면, "나는"을 먼저 "나+는"과 "날+는"으로 분리해야 한다. 이것은 형태소 분리기의 일부 기능이다.

[3]에서는 어절 단위의 태깅을 한다. 어절 단위의 태깅의 문제점은 태그 집합의 종류가 많아지고 어절에 대한 확률정보의 추출이 불확실하게 된다는 것이다. 왜냐하면, 어절에 관한 확률정보를 추출할 때에 어절을 이루는 형태소들은 서로 독립이라는 가정을 기반으로 하기 때문이다. 또, 형태소의 해석이 격자 구조(lattice structure)로 생성되기 때문에 서로 다른 수의 형태소 결합으로 이루어진 어절에 대한 확률을 처리하기 위해 Euclidian 정규화 방법을 사용한다[3]. 이것의 문제점은 형태소와 형태소 사이의 의존 관계를 전혀 고려하지 않은 단순한 확률의 곱을 정규화하여 어절의 확률정보를 추출한다는 것이다. "지금도"가 형태소 해석이 될 경우, "지금(부사)+도(조사)"와 "지금(명사)+도(조사)"로 해석된다. 이 때 각각의 형태소가 가질 수 있는 품사의 가능성(어휘 확률정보)를 이용하여 어절의 확률정보를 구한다면, 부사와 조사, 명사와 조사 사이의 변이 확률정보(문맥 확률정보)를 무시한 결과를 가져오게 된다.

본 논문에서는 위에서 설명한 문제점을 해결하기 위해서 단어(혹은 형태소) 단위로 태그를 붙이고, 또 단어와 단어 사이의 의존관계를 최대한 반영할 수 있도록 모델링하였다.

III. 퍼지망을 이용한 한국어 태깅 모델

본 논문에서는 품사 태깅 문제를 해결하기 위해서 주어진 문장의 품사 격자 구조(lattice structure)를 퍼지 망[4]에 대응시켰다. 그림 1은 "나는 학교에 가는 중이다."에 대한 품사 격자 구조를 퍼지 망으로 표현한 것이다.

한국어 태깅을 위한 퍼지망 $\tilde{G} = (\tilde{V}, \tilde{E})$로 정의한다. 여기서 정점은 퍼지 집합, $\tilde{V} = (v, \mu_V(v))$이고, 연결선은 퍼지 관계, $\tilde{E} = ((v_1, v_2), \mu_E(v_1, v_2))$이다. 태깅 문제에서 $\tilde{V}$는 어떤 단어의 태그(혹은 형태소의 태그) v가 태그 V에 포함될 가능성의 정도를 소속함수로 표현한 것이고, $\tilde{E}$는 품사 v_1과 품사 v_2가 연속적으로 나타날 가능성을 소속함수에 의해서 표현한 것이다. 예를 들면, 명사 다음에는 조사가 나타날 가능성은 매우 높다. 또, 명사 다음에 용언의 어말어미가 나타날 가능성은 거의 없다. 이와 같은 소속 함수들을 이용하여 퍼지 망(fuzzy network)을 구성한다.

3.1 정점의 퍼지 집합

정점은 퍼지집합으로 $\tilde{V} = (v, \mu_V(v))$으로 표현된다. 여기서 $\mu_V(v)$는 품사 태그가 가질 수 있는 소속함수를 확률정보로 이용한다. 즉, $P(v|w)$를 그대로 이용한다. 여기서 w는 입력된 단어이고, v는 그 단어의 태그를 의미한다. $\mu_V(v)$는 말뭉치로부터 $freq(wv)$와 $freq(w)$를 조사하면 쉽게 구할 수 있다. $\mu_V(v) = \frac{freq(wv)}{freq(w)}$로 쉽게 구할 수 있다.

3.2 연결선의 퍼지 관계

연결선은 퍼지 관계로 표현된다. $\mu_E(v_1, v_2)$는 이것도 단순한 확률정보에 의해서 구할 수 있을 것이다. 그러나, 단순한 확률 정보는 여러 가지로 문제점을 가지고 있다. 가장 큰 문제는 학습 자료로 나타나지 않은 자료에 대해서는 어떤 가능성도 이야기할 수 없다는 것이다. 즉, 어떤 품사 v_1 다음에 또 다른 품사 v_2가 나타날 가능성을 말뭉치의 확률 정보에 의해서 추출하는데 이것이 한번도 나타나지 않았을 경우에 그 가능성이 0이므로 전혀 고려되지 않는다. 이와 같은 현상을 방지하기 위한 방법으로 신경회로망을 이용한다. 즉, 소속함수 $\mu_E(v_1, v_2)$를 결정하는데 신경망을 이용하였다. 이 소속함수는 태그 v_1 다음에 태그 v_2가 나타날 가능성을 의미한다. 따라서 이를 구하기 위한 신경망의 출력이 바로 v_1 다음에 v_2가 나타날 가능성이 되도록 모델링하여야 한다.

3.3 퍼지 연산자

t-conorm(∨)과 t-norm (∧)연산자를 사용한다. 먼저 t-conorm 연산자는 가장 일반적인 Max 연산자를 사용하고, t-norm 연산자는 확률적인 곱(·)을 이용한다. 이 확률적인 곱 연산자는 여러 번의 실험을 통해서 결정되었다. 이것에 대한 상세한 설명은 4.3.2절에서 기술할 것이다.

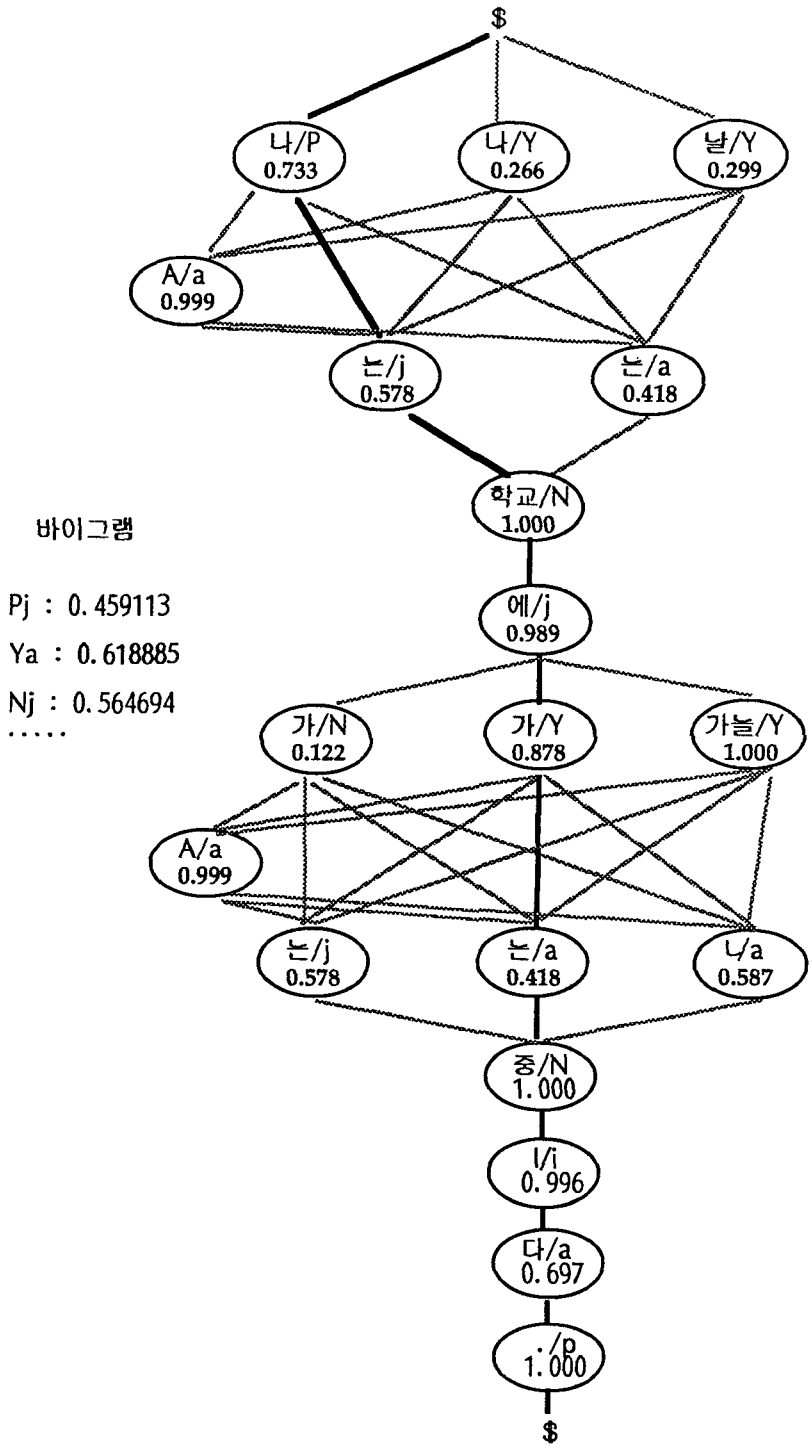

그림 1: 한국어 품사 태깅을 위한 퍼지망 모델

3.4 최적 경로 찾기 알고리듬

퍼지 망에 특별히 문장의 시작과 문장의 끝을 함께 표현할 경우에는 격자 구조를 가지게 된다. 이 때 모든 v_i의 join과 meet는 문장 종결을 표현하는 부호, \$가 된다. 이렇게 하면 그림 1과 같은 퍼지 격자 구조를 얻을 수 있다. 여기서 퍼지 격자 구조로부터 어떤 정점 v_i에 도달하기 바로 이전의 정점을 표현하기 위해 $pred(v_i) = \{v_j | < v_j, v_i > \in E\}$을 정의한다. 여기서 $< v_i, v_j >$는 순서쌍을 의미한다. 또한 정점 v_i의 바로 다음의 정점 v_j를 위해 $succ(v_i) = \{v_j | < v_i, v_j > \in E\}$를 정의한다.

위에서 정의된 succ와 pred를 이용하여 퍼지 망으로부터 최적의 경로를 찾기 위해서 dynamic programming의 일종인 Viterbi 알고리즘을 이용한다. 그림 1에서 굵은 선으로 표현된 경로를 찾는 것과 같다.

1. $succ(\$_s)$에 속하는 v_i에 대해서 다음과 같이 처리한다. 여기서 \$$_s$는 문장의 시작을 나타내는 기호이다.

$$\delta(i) = \mu_E(\$_s, v_i) \wedge \mu_V(v_i) \tag{1}$$

$$\psi(i) = 0 \tag{2}$$

2. 그외의 정점 v_i에 대해서는 재귀적으로 다음과 같이 계산된다.

$$\delta(i) = \max_{v_j \in pred(v_i)} [\delta(j) \wedge \mu_E(v_j, v_i)] \wedge \mu_V(v_i) \tag{3}$$

$$\psi(i) = arg \max_{v_j \in pred(v_i)} [\delta(j) \wedge \mu_E(v_j, v_i)] \tag{4}$$

3. 정점 \$$_e$의 $\psi(i)$에 대해서 경로를 따라 추적하면 그것이 최적의 경로가 된다. 여기서 \$$_e$는 문장의 끝을 나타내는 기호이다.

IV. 실험 및 평가

태깅 시스템의 성능은 정확률에 의해 결정된다. 본 논문에서의 퍼지 모델을 적용한 한국어 태깅 모델은 정확률에 중점을 두었다. 한국어를 태깅할 때 정확률을 결정하기 위해서 본 논문에서는 여러 종류의 퍼지 연산자를 대상으로 실험하였다.

4.1 시스템 구성

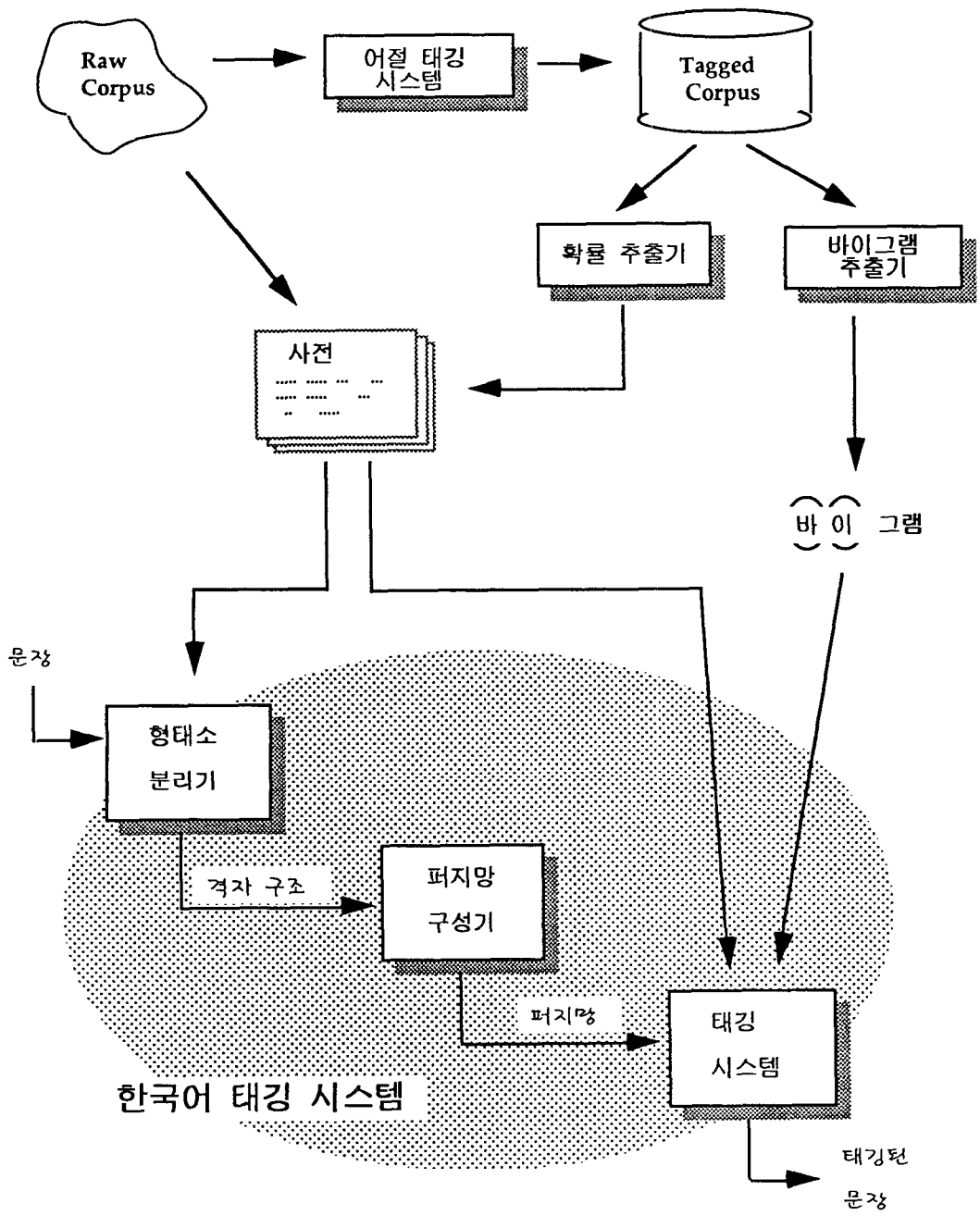

그림 2: 퍼지 망을 이용한 한국어 품사 태깅 시스템의 구성

본 논문의 시스템은 태깅을 하기 위한 여러 가지의 정보 추출 도구와 태깅 시스템으로 크게 구별할 수 있다. 전체의 구성도가 그림 2에 나타나 있다.

태깅되지 않은 말뭉치를 이용해서는 본 논문에서 이용될 여러 가지의 정보를 추출할 수 없기 때문에 가장 먼저 이루어져야 하는 일은 태깅된 말뭉치를 형성하는 것이다. 이는 Two-level 한국어 해석기[2,3]에서 제안된 어절 태깅 시스템을 이용하여 형성하였다. 이렇게 해서 태깅된 말뭉치가 구해지면 이를 이용하여 정점의 소속함수와 정점들간의 연결선의 소속함수를 구하기 위한 확률과 바이그램을 추출한다. 이렇게 추출된 정보는 태깅 시스템에 이용된다.

태깅 시스템의 가장 기본적인 한국어 품사 태그 집합은 [3]에서 사용한 기본 품사 태그 집합을 사용했다. 형태소 분리기의 기능은 어절을 형태소별로 분리하고 분리된 형태소에 여러 가지의 품사를 적재한다. 본 논문에서는 Two-level 한국어 형태소 분리기[2]를 이용하였다. 태깅 시스템은 3.4절에서 설명한 최적 경로 찾기 알고리듬으로 퍼지 망으로 나타난 경로 중 최적의 경로를 찾는다.

4.2 실험 환경

전체 시스템은 C 언어로 구현되었다. 한국어 태깅 시스템을 구현하기 위해서 비교적 문법에 잘 맞게 쓰여진, 정형화된 말뭉치를 사용하였다. 이 말뭉치로부터 추출된 형태소 수는 약 2만 6천개이다.

이 말뭉치를 통해서 각 단어가 가질 수 있는 각 품사의 가능성, 즉, 퍼지 망의 정점의 소속함수와 퍼지 망의 연결선의 소속함수를 구했다. 연결선의 소속함수 값은 신경회로망을 이용하는데, 여기서는 Nakamura에 의해서 제안된 NETgram을 이용하였다[11]. NETgram은 bigram network을 학습시킴으로써 소속함수 $\mu_E(v_1, v_2)$을 구할 수 있다. 이 때, bigram의 확률정보가 반영될 수 있도록 불균등 학습 방법[1]을 이용한다.

4.3 실험 결과의 분석

4.3.1 오류의 분석

본 한국어 자동 태깅 시스템의 평가는 다양한 문장에 대해서 분석하기 위해서 표 1과 같은 다양한 말뭉치들을 사용하였다.

먼저, 국민교육헌장은 매우 잘 정형화된 문장이며, 비교적 적은 오류를 가졌다. 여기서 발생되는 대표적인 오류는 다음과 같다. "이 땅에 태어나 ...."에서의 이는 수관형사이거나 대명사일 가능성이 있다. 그러나 이들은 사실은 단순한 품사 정보만으로 구별하기 어려운 것이다. 이들을 정확하게 분석하기 위해서는 의미 정보와 같은 더 많은 정보가 필요하게 된다. 이 말뭉치에서 발생되는 대부분의 오류는 이와 같은 것이다.

말뭉치	단어 수	오류 단어 수	오류율(%)	정확률(%)
국민교육헌장	280	7	2.50	97.50
실험 말뭉치의 일부	1,948	97	4.97	95.03
정보과학회 논문의 초록	2,052	85	4.14	95.86
소설 '용의 전설'의 일부	2,289	100	4.37	95.63
전체	6,569	289	4.40	95.60

표 1: 확률적인 곱에 의한 분석

다음으로, 실험 말뭉치는 비교적 잘 정형화된 문장들이나, 제목 등 일부 비문법적 문장들도 포함되어 있다. 빈도수가 많은 단어는 일반적으로 많은 변형이 있으며, 많은 애매성을 포함하고 있었다. 자주 나타나면서 애매성을 포함하고 있는 단어로는 "줍니다"가 있다. 문장 "생활하는데 도움을 줍니다."에서 어절 "줍니다"는 형태소 분리기에 의해 "주+ㅂ니다"와 "줍+니다"로 분리된다. 그러나, 올바른 분리 결과는 "주+ㅂ니다"로 분리되어야 한다. 이 단어에 대해서 8개의 오류가 발생되었다. 이 단어 하나에 의해 발생된 오류는 전체 오류 97개 중 8개로 8.2%에 해당한다. 많은 오류가 이와 같은 부류의 오류이다.

다음은 정보과학회 1993년 2월 논문 초록 14개이다. 이 말뭉치는 명사들의 나열이 많은 말뭉치이다. 명사들의 나열은 명사 다음에 명사가 나올 가능성이 0.104502로서 그 가능성이 매우 낮은 편이다. 그렇기 때문에 오류가 발생될 가능성이 높다고 할 수 있다. 예를 들면, "결과"라는 단어가 다른 명사와 함께 나열된 경우, "결과"가 하나의 명사로 해석되기 보다는 "결/N+과/j"의 형태로 해석된다. 이와 같은 오류의 부류로 "기존 → 기조/N+ㄴ/j", "분산 → 분사/N+ㄴ/j", "추른 → 추/N+른/j" 등이 있다.

다음은 소설부분이다. 소설은 축약된 어절을 많이 사용하고, 문맥 정보를 확실히 알아야 해결되는 많은 문제들이 내재되어 있다. 특히 도치와 같은 비문법적인 문장도 많이 포함되어 있기 때문에 퍼지 관계의 소속함수의 부정확성으로 발생되는 오류들이 많았다.

4.3.2 퍼지 연산자의 결정

t-conorm(∨) 연산자로 가장 일반적인 Max 연산자를 사용했다. 본 실험에서의 주 관심 대상은 t-norm (∧)연산자이다. 이는 가능하면 Yager가 주장한 교환성이 큰 연산자를 사용하는 것이 좋은 결과를 가져오기 때문이다. 그러나, 교환성이 큰 연산자는 한 어절 내에서 형태소의 길이가 다를 경우에 발생되는 정규화 문제(2절에서 언급한) 때문에 고려대상이 된다. 정규화 문제를 완전히 해결할 수 있는 연산자는 Min 연산자이다. 그러나, Min 연산자는 앞에서 설명한 교환성 문제가 있기 때문에 여러 여러 종류의 말뭉치를 통해서 결정했다. 본 실험에서 고려한 연산자는 확률적 곱(·), Yager의 I_{10} 연

말뭉치	전체 단어 수	확률적 곱 오류수	Yager의 I_{10} 오류수	Min 연산자 오류수	정규화 연산자 오류수
국민교육헌장	280	7	28	17	10
실험 말뭉치의 일부	1,948	97	235	202	220
정보과학회 논문의 초록	2,052	85	280	211	159
소설 '용의 전설'의 일부	2,289	100	228	198	159
전체	6,569	289	771	628	548
오류율(%)		4.40	11.74	9.56	8.04

표 2: 여러 연산자에 대한 오류율의 비교

산자, Min 연산자, 정규화 연산자($\sqrt{a \times b}$)들이다. 앞에서 설명한 말뭉치를 통해서 이들의 오류율을 비교하여 오류율이 가장 작은 확률적인 곱($\cdot$)를 선정하였다. 표 2는 이들 연산자에 대한 오류율을 비교한 것이다.

표 2에서 보는 바와 같이 확률적인 곱에 대한 연산자의 오류율이 가장 작다. 이 연산자는 교환성은 대단히 높으나 정규화에 문제를 가지고 있다. 그러나 정규화를 요구하는 전체의 어절 수가 상대적으로 작으므로 위와 같은 결과를 가져온 것으로 예측된다.

V. 결론

본 논문에서는 퍼지 망을 한국어 형태소의 품사를 태깅하는 문제에 적용하여 보았다. 전체적인 품사 태깅 정확률은 95.6%로 비교적 좋은 결과를 보였다.

문장 1)"나는 학교에 간다."와 2)"비행기가 나는 것은 새가 나는 방식과 다르다."에는 "나는"이라고 하는 동일한 어절이 존재한다. 그러나 문장 1)에서의 "나는"은 '대명사+조사'로 구성되며, 문장 2)에서의 "나는"은 '용언+어미'로 구성된다. 본 논문에서의 퍼지 망을 이용한 품사 태깅 시스템은 문장 내에서의 위치적 정보를 이용하여 문장 1), 2)에 대한 올바른 해석 결과를 출력한다. 그러나 문장 3)"싹이 나는 모습을 관찰하여라."에서 나타나는 어절 "나는"은 품사적인 정보에서는 '용언+어미'로 문장 2)의 경우와 동일하다. 이 경우 어절 "나는"이 "날(fly) + 는"으로 해석될 가능성이 "나(sprout) + 는"으로 해석될 가능성보다는 크기 때문에 문장 3)의 경우 올바른 결과를 생성할 수 없다. 문장 3)을 올바로 해석하기 위해서는 '비행기'와 '날다', '싹'과 '나다'가 의미적으로 밀접하게 연관되어 있다는 추가의 정보가 필요하다.

앞으로의 과제로는 첫째, 품사의 종류를 결정하는 것이다. 차후에 태그 집합이 변경될 경우에 말뭉치로부터의 정보 추출 작업을 다시 해야 하므로 우선적으로 해야 할 일은 품사 집합를 결정하는 일이라 할 수 있다. 또한 형태소 품사 분류를 지금보다 더 세분화하여 한다. 예를 들면, 본 논문에서는 대부분

의 문장부호에 대해 동일한 태그(k나 z)를 이용한다. 그러나, 문장에서 사용된 부호들은 대부분 특수한 용도로 사용되기 때문에 문장 내에서 구별할 필요가 있다.

둘째, 사전에 등록되지 않은 미등록어에 대한 처리를 해야 한다. 현재는 사전에 등록되지 않은 단어에 대해서 사전에 등록되지 않았다는 메세지를 내고 다음 처리를 하는데, 이것으로 말미암아 전체 문맥에 관한 정보를 정확하게 추론할 수 없을 경우가 발생된다. 이를 해결하기 위해서 미등록어일 경우에도 그 단어를 추측할 수 있는 기능이 부가적으로 필요하다. 한국어의 경우 어미를 보고 일부는 예측이 가능하다. 예를 들면 '-하게'로 끝나는 어절이 있을 경우에 대부분 그 앞에 나타나는 단어는 명사이다.

세째, 형태소 분리기의 오류를 수정하는 일이다. Two-level 한국어 형태소 해석기가 음운현상에 대해서는 비교적 잘 처리되나 때로는 너무 과다하게 분리하기 때문에 발생하는 오류도 적지 않다.

참고 문헌

[1] 김재훈, 김진형, 서정연, "단어 품사 예측을 위한 신경회로망에서의 불균등 학습," **춘계 인공지능 연구회 춘계 학술 발표 논문집**, pp. 20-24, 1993.

[2] 이성진, Two-level **한국어 형태소 해석**, 한국과학기술원, 전산학과,석사학위 논문, 1992.

[3] 이운재, **한국어 문서 태깅 시스템의 설계 및 구현**, 한국과학기술원, 전산학과,석사학위 논문, 1993.

[4] 이광형, 오길록, **퍼지 이론 및 응용: I권(이론)**, 홍릉과학출판사, 1991.

[5] J. Benello, A. W. Mackie, J. A. Anderson, "Syntactic Category Disambiguation with Neural Networks," *Computer Speech and Language*, vol. 3, pp. 203-217, 1989.

[6] E. Brill, "A Simple Rule_Based Part of Speech Tagger," *Proc. of the 3rd Conf. on Applied Natural Language Processing*, Trento, Italy, pp. 153-155, April, 1992.

[7] Kenneth Ward Church, "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," *Proceedings of Applied Natural Language Processing*, Austin, Texas, 1988.

[8] D. Cutting, J. Kupie, J. Pedesen, P. Sibun, "A Practical Part-of-Speech Tagger," *Proc. of the 3rd Conf. on Applied Natural Language Processing*, Trento, Italy, pp. 133-140, April 1992.

[9] S. J. DeRose, "Grammatical Category Disambiguation by Statistical Optimization," *Amer. J. of Computational Linguistics*, vol. 14, no. 1, pp. 31-39, 1988.

[10] K. Koskenniemi, "Finite-state Parsing and Disambiguation," *Int'l Conf on Computational Linguistics(Coling-90)*, pp. 229-232, Helsinki Univ., 1990.

[11] M. Nakamura, K. Maruyama, T. Kawanata, K. Shikano, "Neural Network Approach of Word Category Prediction for English Texts," *Int'l Conf on Computational Linguistics(Coling-90)*, pp. 213-218, 1990.

부록 1: 태깅의 예

입력문장:

본 논문에서는 최근에 연구가 활발한 객체지향 데이타 베이스시스템에서 새로운 효율적인 질의어 처리방안을 제시한다. 객체지향 데이타베이스에서 복합객체에 대한 질의어 처리 문제를 효율적으로 해결하기 위하여 의미지식 제약조건을 이용한 방법을 개발하였다. 질의어들이 찾게 될 데이타에 대한 유용한 정보가 있다면 그 정보를 적절히 이용하여 질의어 처리에 불필요한 부분을 생략하고 변형시켜, 보다 효율적인 처리를 수행함으로써 질의어 최적화를 이룰 수 있다. 의미지식 질의어 최적화의 방법으로 의미지식을 효과적으로 적용시켜 여러 가지 휴리스틱 기술을 개발하였다.

태깅된 문장:[1]

본/g 논문/N 에서는/j 최근/N 에/j 연구/N 가/j 활발/N 하$/y ㄴ/a 객체지향/N 데이타/N 베이스시스템/N 에서/j 새롭$/Y ㄴ/a 효율적/N I/i ㄴ/a 질의어/N 처리방안/N 을/j 제시/N 하$/y ㄴ다/a ./p 객체지향/N 데이타베이스/N 에서/j 복합객체/N 에/j **대하**/N ㄴ/j 질의어/N 처리/N 문제/N 를/j 효율적/N 으로/j 해결/N 하$/y 기/a 위하$/Y A/a 의미지식/N 제약조건/N 을/j 이용/N 하$/y ㄴ/a 방법/N 을/j 개발/N 하$/y A쓰/s 다/a ./p 질의어/N 들/W 이/j 찾/Y 게/a 되$/Y ㄹ/a 데이타/N 에/j **대하**/N ㄴ/j 유용/N 하$/y ㄴ/a 정보/N 가/j 있/Y 다면/a 그/g 정보/N 를/j 적절히/D 이용/N 하$/y A/a 질의어/N 처리/N 에/j 불필요/N 하$/y ㄴ/a 부분/N 을/j 생략/N 하$/y 고/a 변형시키/Y A/a ,/k 보다/D 효율적/N I/i ㄴ/a 처리/N 를/j 수행/N 하$/y ㅁ/a 으로써/j 질의어/N 최적화/N 를/j 이루/Y ㄹ/a 수/N 있/Y 다/a ./p 의미지식/N 질의어/N 최적화/N 의/j 방법/N 으로/j 의미지식/N 을/j 효과적/N 으로/j 적용시키/Y A/a 여러/g 가지/N 휴리스틱/N 기술/N 을/j 개발/N 하$/y A쓰/s 다/a ./p

[1]굵은 단어는 오류를 의미한다