工學碩士 學位論文

유한상태 오토마타를 이용한 수식 표현 인식

Identification of Numerical Expressions using Finite-State Automata

指導敎授 朴 炘 讚

2002年 2月

韓國海洋大學校 大學院

컴 퓨 터 工 學 科

白 億 種

本 論文을 白億種의 工學碩士 學位論文으로 認准함

委員長 工學博士 孫 周 永 印

委 員 工學博士 辛 沃 根 印

委 員 工學博士 朴 烋 讚 印

2001年 12月

韓國海洋大學校 大學院

컴퓨터工學科 白 億 種

목 차

Identification of Numerical Expressions using Finite-State Automata

Ock-Jong Baek

Department of Computer Engineering, Korea Maritime University, Pusan, Korea

Abstract

There have been many trials to parse sentences in text to search complete and exact parses, but it is very hard because of unavoidable incompleteness of lexicon and grammar. Recently, to alleviate these difficulties, partial parsing appears as an alternative in the field. Partial parsing aims to recover syntactic information efficiently and reliably from unrestricted text, by sacrificing completeness and depth of analysis.

As a part of partial parsing, the identification of Korean numerical expressions in text is described in this paper. Numerical expressions are required in several systems such as information extraction systems and question-answering systems. One of desired characteristics of these systems is the fastness. To achieve this goal, we use a finite-state automaton, for which we could use a tool like lex. So that we could rapidly implement the system. We observed that the system is fast and correct through several experiments. To evaluate our system, we used newspaper as test collection. We achieved the recall of 90.8%, and the precision of 86.9%. Experiments show that our system is comparatively correct.

제 1 장 서 론

자연언어는 인공언어와 달리 표현이 다양하며 사용되는 어휘도 제한되지 않는다. 따라서 모든 문장을 표현하는 문법을 기술한다는 것은 매우 어려운 일이다. 이와 같은 문제를 다소 완화하기 위해서 중의성(ambiguity)이 적은 표현을 인식하기 위한 문법을 기술하고 이를 이용해서 문장의 구조를 분석하는 부분 구문분석(partial parsing)에 대한 연구가 활발히 진행되고 있다[1-4]. 부분 구문분석은 중의성이 적은 부분부터 먼저 분석을 시작해서 완전한 문장의 구조를 분석한다. 따라서 때로는 완전한 문장의 구문구조를 분석하지 못할 경우도 있다. 그러나 부분적인 구문 구조만으로도 충분히 정보의 가치가 있는 정보 검색이나 정보 추출 등의 분야에서 부분 구문분석은 사용되고 있다[5-7].

자연언어에서 중의성이 적은 부분은 여러 가지 표현이 있을 수 있으나 본 논문에서는 수사와 관련된 표현만을 다루며, 이를 수식 표현(numerical expressions)이라고 한다. 수식 표현이란 수사와 함께 쓰여 의미 있는 정보를 나타내는 것으로 시간 표현(temporal expression)과 수량 표현(quantity expression)이 있다. 본 논문에서는 시간 표현을 다시 날짜(date), 시간(time), 기간(duration)으로 분류하고, 수량 표현을 금액(money), 비율(percent), 측도(measure), 계수(cardinal)로 분류한다[8].

본 논문에서는 수식 표현을 표현하기 위해서 정규표현(regular expression)[1]을 사용하고, 수식 표현을 인식하기 위해서 유한상태 오토마타(finite-state automata, FSA)를 이용한다. 수식 표현의 구조는 비교적 단순하고 수식 표현의 형태적인 정보도 또한 명확하기 때문에 정규표현으로 표현이 가능하다. 또한 문서에 나오는 수식 표현들을 관찰하여 각 표현들을 정규표현으로 기술하며, 이

1) 정규표현은 정규문법으로 변환할 수 있으며 시스템 구현을 위한 유한상태 오토마타로도 변환이 가능하다[12,13,14].

정규표현을 lex를 이용해서 유한상태 오토마타로 변환한다[9- 11]. 이렇게 변환된 시스템의 입력은 문장이고, 출력은 수식 표현이 된다. 유한상태 오타마타를 이용한 수식 표현 인식 시스템의 장점은 처리 속도가 빠르고, 정확율 또한 비교적 높다. 그리고, 시스템 구현 시간이 매우 짧은 것도 하나의 장점이 된다. 단점으로는 정규표현의 표현 능력의 한계로 중의성의 해결 능력이 다소 떨어지며 의미정보의 표현 능력이 다소 떨어진다는 것이다.

수식 표현 인식 시스템은 주가 정보, 환율정보, 일기예보 정보 등을 추출하는 정보추출 시스템[5, 7]이나 질의응답 시스템[12]에 널리 사용될 수 있다. 그 밖에도 주어진 문장에 수식 표현을 먼저 인식함으로써 자연언어처리 시스템의 처리 부담을 덜어줄 뿐 아니라 속도도 크게 개선할 수 있을 것이다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구로서 정규표현과 유한상태 오토마타의 관계, 자연언어처리에서 유한상태 오토마타를 이용한 사례, 그리고 수식 표현 인식 시스템의 응용에 대해서 살펴본다. 제3장에서는 한국어 수식 표현의 형식에 대해서 살펴본다. 제4장에서는 유한 상태 오토마타를 이용한 수식 표현 인식 시스템에 대해서 구체적으로 기술한다. 제5장에서는 제안된 시스템의 성능을 평가하고, 마지막으로 제6장에서 결론과 향후 연구 방향에 대해서 기술한다.

제 2 장 관련 연구

본 장에서는 정규표현과 유한상태 오토마타에 대하여 살펴보고, 유한상태 오토마타를 이용한 자연어처리 방법에 대해서 살펴본다. 그리고 수식 표현 인식 시스템이 응용될 수 있는 분야에 대해서 살펴본다.

2.1 정규표현과 유한상태 오토마타

정규표현(regular expression)을 말하기에 앞서 정규문법(regular grammar)이 정의되어야 한다. 정규문법은 중첩되지 않는 언어를 표현하는 수단 중 하나이며, 주로 컴파일러의 어휘 분석 과정에서 토큰 구조를 표현하는데 이용된다. 정규문법에 의해 생성될 수 있는 언어를 정규언어(regular language)라 한다. 정규언어를 표현하기 위한 또 다른 수단으로 정규표현이 사용되며, 정규표현은 정규언어에 속해 있는 문장을 직접 기술할 수 있다는 특징을 갖는다. 정규문법이 생성하는 언어와 같은 종류의 언어를 인식하는 인식기를 유한상태 오토마타라 한다. 유한상태 오토마타가 인식하는 언어를 정규표현으로 나타낼 수 있고, 역으로 주어진 정규표현을 인식하는 유한 오토마타를 고안할 수 있다. 즉 그림 1과 같이 정규문법, 정규표현, 유한상태 오토마타는 서로 변환이 가능하다[9].

렉스(lex)는 1975년에 레스크(Lesk, M. E)에 의해 발표된 어휘 분석기이다. 이것은 입력 문자열에서 정규표현으로 기술된 토큰들을 찾아내는 프로그램을 작성하는 데 유용한 도구이다. 렉스의 기능은 사용자가 정의한 정규표현과 패턴-액션을 입력으로 받아 C 언어 프로그램을 생성한다. 이 프로그램은 입력 문자열에서 정규표현에 해당하는 토큰을 찾아 준다.

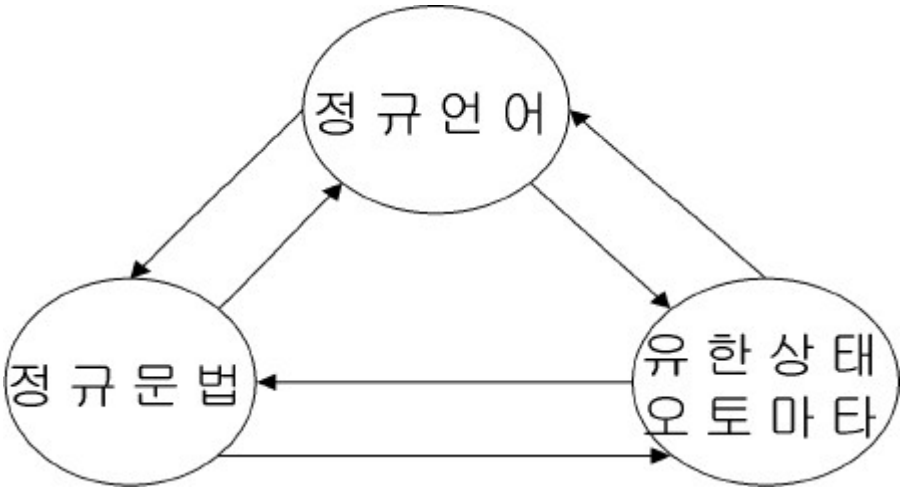

그림 1 정규표현와 정규문법과 유한상태 오토마타

Fig. 1 Regular expression, regular grammar, and finite-state automata

렉스의 입력(lex source)**은 다음과 같이 세 부분으로 구성되어 있다**.

<**정의 부분**>
%%
<**규칙 부분**>
%%
<**사용자 부프로그램 부분**>

여기서 %%**가 각 부분의 구분자이며**, **각 부분은 반드시 순서적으로 기술되어야 한다**. <**정의 부분**>**에서** %{**와** }% **사이에는 실행 코드를** C **언어로 기술할 때 필요한 자료 구조**, **변수**, **상수를 정의할 수 있는 부분이다**. **렉스는** %{**와** }% **사이에 있는 프로그램 부분을 렉스의 출력인** lex.yy.c**의 앞부분에 그대로 복사한다**. <**규칙 부분**>**은 렉스 입력의 핵심 부분으로 문자열을 표현하는 정규표현과 그 표현이 인식되었을 때 처리할 행위를 기술하기 위한 부분인 실행 코드로 구성된다**. **렉스 입력의 세 번째 부분인** <**사용자 부프로그램 부분**>**은 렉스의 입력 작성시 사용되는 부프로그램들을 정의하기 위한 곳으로 렉스에 의한 어떤 처리없이 그대로 렉스의 출력인** lex.yy.c**에 복사된다**.

형식 언어 이론에서 정의한 정규표현을 바탕으로 실제로 렉스에서 제공된 방법을 사용하여 토큰의 형태를 정확하게 표현할 수 있어야 한다. 렉스의 정규표현은 크게 텍스트 문자(text character)와 연산자(operator character)들로 구성된다. 텍스트 문자는 입력 문자열에서 실제로 매칭되는 부분이고, 연산자 문자는 반복 또는 선택 등을 나타내는 특수 문자들이다. 예를 들어, 정규표현 a*에서 a는 텍스트 문자이고, *는 연산자 문자이다. 일반적으로 영문자와 숫자는 항상 텍스트 문자이고, 연산자 문자는 특수 문자로 나타낸다. 렉스에서 토큰의 구조를 쉽게 표현할 수 있도록 제공한 연산자 문자들은 표 1과 같다.

표 1 패턴 지정을 위한 메타 문자 및 의미

Table 1 Meta characters for patterns and their meanings

메타 문자	설명
.	newline문자(\n)을 제외한 임의의 하나의 문자(single character)를 의미
*	* 앞에 있는 패턴의 0회 이상의 반복을 의미
[ ]	Bracket [ ]내에 지정된 character class내의 임의의 한 문자
^	정규식 내에서 첫 문자로 사용시 라인의 시작을 의미, [] 내에서는 negation의 의미로도 사용
$	정규식 내에서 첫 문자로 사용시 라인의 끝을 의미.
{ }	{} 직전 패턴의 반복 매칭 횟수의 범위를 지정
\	메타 문자의 의미를 disable하거나, C의 escape sequence 지정
+	+ 앞에 있는 패턴의 1회 이상의 반복을 의미
?	? 직전 패턴의 0 또는 1회 반복을 의미
|	| 양쪽에 있는 좌측 패턴과 우측 패턴 중 하나에 대응.
“...”	quotation mark 내에 있는 모든 문자를 문자 그대로 해석.
/	slash "/" 디에 있는 정규식에 일치하는 경우에만 "/"앞에 있는 정규식에 매칭
( )	( )는 일련의 정규식들을 하나의 정규식으로 그룹화

2.2 유한상태 오토마타를 이용한 자연어 처리 방법

유한상태 오토마타를 이용하는 방법은 여러 연구자들에 의해서 수행되었다 [1-3]. 이들의 공통점은 단계형 유한상태 오토마타(cascaded finite-state automata)를 이용한다는 것이다. 단계형 유한상태 오토마타는 유한상태 오토마타가 여러 단계를 이루고 있는 경우를 말하며, 각 단계에서는 정규표현에 의해서 특별한 구를 인식한다. 주어진 입력은 여러 개의 정규표현에 일치될 수 있다. 이 경우에는 일반적인 정규표현에서 중의성을 해결하는 방법과 같이 최장일치된 정규표현을 선호하도록 한다. 어떤 정규표현에도 일치되지 않은 입력은 무시

되어 그대로 출력된다. 최장일치 방법은 전체 문장을 분석하지 않고도 구를 매우 신뢰성 높게 찾을 수 있으며, 단계형 오토마타는 신뢰성이 가장 높은 구에 대해서 먼저 수행하여 점점 더 큰 구를 형성해 갈 수 있다. 또한 이 방법은 추가적인 정보를 쉽게 첨가할 수 있기 때문에 유연성이 매우 좋다.

2.2.1 Cascaded Analysis of Syntactic Structure(CASS)

부분 구문분석은 구조적 중의성이 많이 발생하지 않는 지엽적인 부분에 해당하는 구조적 정보만을 이용하여 문장을 분석한다. 그렇기 때문에 결과 구조는 하나의 전체 구문 트리가 아닌 부분 트리들의 집합이 된다. CASS에서는 하나의 구를 인식하기 위한 부분 구문분석 규칙이 정규 문법으로 기술되어 있으며, 각 규칙은 규칙 레벨에 따라 단계적으로 적용된다. 다음의 예제는 부분 구문분석 규칙의 한 예제이다[1].

```
1 : NP -> D? A* N+ | Pron
    VP -> Md Vb | Vz | Hz Vbn | Bz Vbn | Bz | Vbg
2 : PP -> P NP
3 : SV -> NP VP
4 : S  -> (Adv | PP)? SV NP? (Adv | PP)*
```

위 문법에서 쓰인 각 심볼이 지니는 의미는 다음과 같다:

D : 관사, A : 형용사, N : 명사, Pron : 대명사, P : 전치사, Adv : 부사
NP : 명사구, VP : 동사구, PP : 전치사구,

규칙 앞에 붙어 있는 각 숫자는 그 규칙의 적용 레벨을 의미한다. 품사 태깅의 결과를 레벨-0이라고 했을 때, 레벨 L의 규칙은 레벨 L-1의 규칙의 출력을 입력으로 받아 처리하게 된다. 한 개 이상의 규칙이 적용될 경우에는 최장일치(longest match)를 이용하여 중의성을 해결하였다. CASS에서 사용한 부분 구문 규칙은 사람에 의해 손으로 작성되었으며, 약 95%의 정확도를 얻었다.

2.2.2 FASTUS(Finite-State Automation Text Understanding System)

Hobbs et al.[6]은 처리 속도를 개선하기 위해 유한상태 오토마타를 이용해서 정보 추출 시스템(information extraction system)을 구축하였다. FASTUS는 모든 단계에 유한상태 변환기를 이용하였으며, 다음과 같은 단계로 처리되었다.

1. 이름을 인식한다.
2. 간단한 구를 인식한다.
3. 복잡한 구를 인식한다.
4. 영역에 의존적인 구나 절을 인식한다.
5. 정보 추출에 대상이 되는 구나 절을 결합한다.

FASTUS는 단순한 패턴 매칭기법을 사용하였기 때문에 개념적으로 간단하고, MUC(message understanding conference) 평가에서 100개의 신문기사를 12분만에 처리하였기 때문에, 빠른 속도가 그 특징이다.

2.2.3 Fidditch

Fidditch는 Hindle[13]에 의해서 개발되었으며, 비록 오래되었지만, 가장 성공적인 부분 구문분석기 중 하나이다. Fidditch는 원래 부분 구문분석을 목적으로 개발한 것은 아니고, 비제한적인 문장(unrestricted text)를 분석하기 위해서 개발되었다. Fidditch는 Marcus 구분 분석기에 펀트(punt)라고 하는 행위를 첨가하였다. 펀트는 구의 역할이 정확하지 않을 경우에 해당하는 구를 입력에서 제거하고(skip and fit), 즉 부착을 하지 않은 상태로 두고 파싱을 계속 진행하는 방법을 말한다. 파서는 절의 핵심요소(보통 절의 경계표지, that, which 등), 주어, 술어, 펀트된 절에 부착된 요인들을 인식한다.

Fidditch는 매우 빠르기 때문에 대량의 문장 즉 말뭉치에 대한 구문분석에 매우 유용하다. 또한 파서가 결정적으로 수행되기 때문에 유한상태 오토마타를 이용해서 구현할 수 있다.

2.2.4 한국어 시간 표현 인식

유한상태 오토마타를 한국어에 적용한 예에는 정보 추출[5]과 시간 표현을 인식[14]하는 것과 같은 지역적인 정보를 처리하는 분야가 있다. 시간 표현은 정보 추출과 같은 응용분야에서 필히 인식해야 하는 중요한 요소이다. 언제 사건이 발생했는지에 대한 정보를 시간 표현을 통해서 알 수 있기 때문이다. 시간 표현을 인식하기 위해서, 김윤관[14]은 대량의 말뭉치로부터 시간 관계 표현을 획득하고, 이들을 부분 문법(local grammar)으로 표현한 후, 이것을 FST (finite state transducer)를 이용하여 부분 구문분석한다.

2.3 수식 표현 인식 시스템의 응용

수식 표현 인식을 이용할 수 있는 시스템은 크게 정보 추출 시스템, 시간 혹은 날짜에 대한 질의 응답 시스템, 그리고 가격 정보에 민감한 증권 관계 비교 쇼핑 에이전트 등이 있다.

2.3.1 정보 추출 시스템

인터넷의 발달로 인해 사용자가 접할 수 있는 자연어 텍스트의 양이 증가됨에 따라 필요한 정보만을 추출하는 정보 추출 시스템의 필요성이 증대되고 있다. 정보 추출이란 특정 분야의 내용을 담은 자연어 텍스트로부터 원하는 정보만을 찾아내어 데이터베이스화하는 작업을 말한다. 즉 추출할 정보의 종류를 미리 틀(template)로 정의하고, 텍스트 분석을 통해 틀을 메꾼 후 채워진 틀을 DB에 저장하는 작업이다[5, 7]. 틀은 해당 분야의 전문가가 정의하기도 하고, 학습 데이터로부터 문서의 주제를 학습한 후 각 주제에 해당하는 틀을 만들기도 한다. 정보 추출 시스템의 대상 텍스트로는 테러 사건과 야구 경기 결과와 같은 문서에서 필요한 정보만을 추출하는 것인데, 이런 분야에서의 틀은 가해자와 피해자 정보, 발생 지역에 대한 정보, 사건이 발생한 시점에 대한 정보, 발생 피해액에

관련된 정보가 필요하다. 이 중에서 수식 표현은 사건이 발생한 시점에 대한 정보와, 발생 피해액에 대한 정보를 제공할 수 있다.

2.3.2 질의 응답 시스템

자연언어에 의한 질의 응답 시스템은 질문자가 그 내용을 형식에 구애받지 않는 일상의 회화문으로 입력하면 의미를 파악하여 원하는 정보를 제공해 주는 시스템이다. 질의 응답 시스템에서 자연스러운 대화가 이루어지기 위해서는 화자의 의도나 대화의 배경이 되는 상식 등을 시스템 내에 형식화하여야 하고 대화의 전후 관계나 지시사 문제, 화제의 관리 및 일관성 유지 등이 필요하다. 적절한 소규모의 적용 영역을 대상으로 하는 연구가 현실적인 것이다. 질의 응답 시스템은 주로 호텔예약과 같은 소규모의 영역에서 이루어지며, 수식 표현은 예약 날짜 혹은 금액, 그리고 방 번호와 같은 정보를 처리하는데 이용될 수 있다[12].

제 3 장 한국어 수식 표현의 형식

본 장에서는 한국어 문장에서 쓰이는 수식 표현에 관련된 수사에 대해서 살펴보고 수식 표현의 형식과 그 종류에 대해서 살펴본다.

3.1 한국어 수사

한국어 문장에서 수사란 사물의 수량이나 차례를 가리키는 말로 수량을 가리키는 양수사와 차례를 가리키는 서수사로 나뉘며 정확한 수량이나 차례를 나타내는 정수와 개략적인 수량이나 차례를 나타내는 부정수가 있다. 표 2는 한국어 수사의 예를 보이고 있다. 이 밖에도 극히 일부이기는 하지만 '원', '투'와 같이 영어를 음차하여 표현하는 수사도 있으나, 본 논문의 범위를 벗어나기 때문에 특별히 다루지는 않는다.

표 2 한국어 수사 분류

Table 2 Classification of numerals in Korean

분 류	구 별		보기
숫자			0, 1, 2, 3, ..., 9
고유어 수사	양수사	정수	'하나', '둘', '셋', ...
		부정수	'한둘', '두셋', ...
	서수사	정수	'첫째', '둘째', ...
		부정수	'한두째', ...
한자어 수사	양수사	정수	'일', '이', ...
		부정수	'일이', '이삼', ...
	서수사	정수	'제일', '제이', ...
		부정수	(없음)

3.2 수식 표현의 형식

수식 표현에서 계수표현을 제외하고는 거의 모든 표현의 형식이 수사 다음에 단위성 의존명사가 오며 그 단위성 의존명사의 종류에 따라 수식 표현의 종류를 결정할 수 있다. 그러나 경우에 따라서는 특별한 의미를 가지는 단위성 의존명사나 실마리 단어가 생략되는 경우가 있는데, 이 경우가 중의성 발생의 한 원인이 된다. 또 다른 형식으로는 "제5회"에서 '제'와 같은 접두어 다음에 수사가 나오는 형식이 있을 수 있다. 이 밖에도 기호나 특수한 단위성 의존명사에 따라서 조금씩 그 형식이 변할 수 있는데 이것에 대해서는 이하의 절에서 기술한다.

3.2.1 수식 표현의 기호와 그 중의성

수사와 함께 쓰여서 의미 있는 정보를 나타내는 것으로 가장 먼저 기호가 있다. 단순히 문장의 종결을 나타내는 마침표(.)가 숫자의 중간에 쓰이는 경우는 소수점 수를 표현하며, 문장의 중간에 나타나서 주절과 종속절을 나누는 쉼표(,)가 숫자들 사이에서 수를 식별하기 쉽게 자릿수를 구분하기도 한다. 이 밖에도 부연 설명하기 위한 하이픈(-)이 숫자들 사이에서 날짜표현을 하기도 하고, 콜론(:)의 경우는 시간을 나타내기도 한다. 달러($) 기호와 퍼센트(%) 기호도 각각 수사와 함께 쓰여 금전을 표시하거나, 비율을 표시하기도 한다. 틸드( ) 기호는 기간 혹은 범위를 나타내며, 그 외의 기호들은 숫자와 함께 사용될 때 특별한 의미를 가지기도 한다. 아래는 이들 기호들이 사용된 예를 보이고 있다.

(3-1) 2.7%의 경제 성장률을 보였다.
(3-2) 2,000,000원까지만 지급한다.
(3-3) 2001-3-20 (2001년 3월 20일)
(3-4) 12:30 (12시 30분)

3.2.1 단위성 의존명사의 분류

수사와 함께 쓰여서 의미 있는 정보를 나타내는 것으로 단위성 의존명사는 그

비중이 상당하다. 대부분의 수사는 수관형사의 형태로 다음에 오는 명사를 수식하는 경우를 제외하고, 수사와 단위성 의존명사의 쌍으로 존재하는 경우가 대부분이다. 단위성 의존명사는 그 주된 기능이 셀 수 있거나(countable), 셀 수 없는(uncountable) 명사의 셈에 관여하여 그 명사를 셀 수 있게 해주는 것과, 셈의 대상이 되는 명사의 의미론적 특성을 명시해 주는 것이다. 명사에 의해 표현된 의미 특성을 다시 되풀이 명시해 준다는 점에서는 잉여적인 요소로 간주한다[15]. 언어학 사전에서 단위성 의존명사는 "단어들의 의미론적 또는 형태론적 종류를 가리키기 위해 쓰이는 보조적 기호(예 : 새 한 '마리', 연필 한 '자루' 등) 혹은 그것이 속해 있는 단어의 범주를 가리키는 형태"로 정의된다. 표 3은 한국어 단위성 의존명사를 그 의미별로 분류한 것이다.

표 3 단위성 의존명사 하위분류

Table 3 Subcategory of unit bound nouns

척도	길이	자, 척, 치, 뼘, 리 .....
	넓이	평, 마지기, 정보 .....
	부피	섬, 가마니, 말, 홉, 되, 아름 .....
	무게	근, 돈, 관, 푼 .....
	시간	시, 분, 초, 년, 월, 일 .....
	횟수	바퀴, 번, 회 .....
	속도	마력 .....
	화폐	원, 달러, 엔, 프랑, 마르크
수량	사람	분, 사람, 명, 놈, ...
	사물	개, 모, 그루, 촉(난초), 톳(김), 거리(오이, 가지) 량(열차), 쌈(바늘), 접(마늘, 무), 송이(꽃류) 권(서적류), 자루(총, 연필류), 척(선박류), 대(차량)
	동물	마리, 두, 필(말), 손(생선류), 축(오징어), 두름(조기), 벌(의류, 그릇류) .....

3.3 수식 표현의 분류

문서 내에 사용되는 수식 표현은 크게 시간 표현과 수량 표현으로 나눈다. 시간 표현은 날짜(DTE), 시간(TME), 기간(DUR)로 나누고 수량 표현은 통화(MNY), 비율(PCT), 측도(MSR), 계수(CRD)로 나눈다[8].

시간 표현에는 절대적인 표현과 상대적인 표현이 있으며, 절대적인 표현은 "2001년 12월 25일"과 같이 구체적인 시간을 나타내는 표현이며, 이와는 반대로 상대적인 시간 표현은 "오늘", "어제", "내일", "금년", "작년" 등과 같이 어떤 시간을 기준이 있어야 정확한 시점을 찾을 수 있는 시간 표현이다. 본 논문에서는 절대적인 시간 표현은 물론 상대적인 시간도 그 시간과 날짜를 예측할 수 있으므로 인식의 범위에 포함시켰다. 시간 표현은 정보 추출과 같은 시스템에서 어떤 사건의 발생한 시점을 표현하기에 아주 유용한 정보로 사용될 수 있다.

수량 표현에는 수사와 단위성 의존명사와 함께 표현되며 그 구조는 비교적 단순하다. 그러나 단위성 의존명사에 따라 수사가 제약된다.

3.3.1 날짜(Date) 표현

날짜가 표현하는 시간의 범위는 하루를 넘어서는 범위의 시간이다. 그래서 날짜는 년, 월, 일의 시간 단위로 나타나고, 전체, 혹은 부분으로 표현된다. 부분 부분으로 표현될 경우에는 가장 큰 덩어리를 인식하는 것이 올바른 정보를 전달한다. 예를 들어, "2001년 3월 2일"이라는 입력 데이터에 대해서 띄어쓰기 단위로 구분되는 날짜 표현에 대해서 각각의 태그를 부여하기보다는 전체를 하나의 덩어리로 인식하는 것이 의미있는 정보를 전달하기 때문이다.

보통 날짜의 표현은 " 년 월 일"의 형태로 표현되며, 될 수 있으면 큰 덩어리를 인식하기 위해서 가장 큰 규칙을 먼저 정의한다. 그리고 중간 규칙을, 그리고 마지막으로는 가장 작은 표현을 정의한다. 날짜 표현의 종류는 다음과 같

은 전체 혹은 부분 표현이 있으며 인식기에서는 이들 모두를 인식한다.

(3-5) 1999년 12월 25일은 금세기 마지막 크리스마스이다.
(3-6) 76년 3월은 중요한 일이 있었다.
(3-7) 76년에 태어난 사람들의 모임은 용우회이다.
(3-8) 지난달 7일에 합격자 발표가 있었다.

수사가 아니지만 날짜를 표현하는 경우가 있는데 이들도 인식의 범위에 포함시킨다. 연도를 나타내는 표현으로 10간12지를 사용하는 경우가 있는데 이들도 간단한 정규식을 사용해서 나타낼 수 있다. 달을 나타내는 고유어도 포함시켰으며, 어떤 특정한 날을 나타내는 표현도 포함시켰다.

(3-9) 임진왜란은 임진년에 일어났기 때문에 붙여진 이름이다.
(3-10) 정월에는 새로운 마음으로 한 해를 시작하는 사람이 많다.
(3-11) 어린이날과 어버이날은 5월에 있다.
(3-12) 제헌절은 7월 17일이다.
(3-13) 식목일은 나무를 심는 날이다.
(3-14) 크리스마스는 성탄절의 또 다른 이름이다.
(3-15) 이번 하반기에 있었던 일 중에는 9월 11일에 테러가 있었다.
(3-16) 3월 초순에 합격자 발표가 있을 예정이다.

3.3.2 시간(Time) 표현

시간을 표현하는 범위는 하루를 넘지 않는 범위의 시간이다. 그래서 시간은 시, 분, 초의 시간 단위로 나타나고, 전체 혹은 부분으로 표현된다. 날짜 표현에서와 마찬가지로 부분 부분의 표현일 경우에는 가장 큰 덩어리를 인식하는 것으로 한다. 비록 수사는 아니지만 시간을 나타내는 표현이 있으며, 이들은 시간의 표현을 좀 더 명확히 해주는 경우가 많다. 시간 인식에서는 이들의 인식도 포함한다. 시간에 대한 표현으로는 다음과 같은 전체 혹은 부분의 시간이 있다.

(3-14) 11시 11분 11초에 축제는 시작된다.
(3-15) 10시 30분까지 도착해야 한다.
(3-16) 7분 30초가 소요되었다.
(3-17) 모든 수업은 오전 9시에 시작한다.
(3-18) 밤 12시에 전화가 걸려왔다.
(3-19) 오후 4시에 만나기로 약속했다.

3.3.3 기간(Duration) 표현

어떤 일이 지속되는 기간을 표현하는 말이다. “2 3월”과 같이 틸드( )기호가 사용되어서 시간의 범위를 나타내기도 하고, “ 부터 까지”와 같이 표현하기도 하며, “ 일간”, “ 년동안”에서처럼 특정한 접사를 사용하기도 한다. 기간을 나타내는 구체적인 표현에는 다음과 같은 것들이 있다.

(3-20) 지난 12년간의 노력이 결과로 나타났다.
(3-21) 10시부터 12시까지 회의가 진행된다.
(3-22) 1월부터 3월까지는 새로운 정책을 구상하는 기간이다.
(3-23) 지난 열흘동안의 계획이 드러났다.
(3-24) 기간 : 1994 2000
(3-25) 시간 : 14 16시

3.3.4 금전(Money) 표현

화폐 단위는 그 숫자가 비교적 적다. 대표적으로 원, 달러, 엔, 마르크, 파운드, 프랑과 같은 글자와 $, ₩ 弗 같은 기호 등이 있으며, 수사와 결합하는 형식도 그렇게 복잡하지 않다. 그리고 화폐와 결합하는 수사의 종류는 고유어가 사용되는 경우는 거의 드물고, 숫자와 한자어 수사가 함께 사용되는 경우가 있지만, 여기에도 어느 정도의 제약이 따른다. 화폐와 함께 사용되는 기호는 마침표가 쓰여서 소수점 수를 표현하고, 쉼표가 쓰여서 자릿수를 쉽게 인식할 수 있게 한다. 금전을 나타내는 구체적인 표현에는 다음과 같은 것들이 있다.

(3-26) 2천만원으로는 턱없이 부족하다.
(3-27) 2,000달러의 상금이 걸려있다.
(3-28) 17만 마르크를 가지고 독일을 갔다.
(3-29) 1$는 1200.36원의 가치를 가진다.

3.3.5 비율(Percent) 표현

비율은 어떤 수나 양의 다른 수나 양에 대한 비이며, 보통 수사에 퍼센트 혹은 %와 같은 기호와 함께 쓰인다. 숫자와 한자어 수사와 결합하여 비율을 표현하며, 사용되는 수사의 크기는 그렇게 크지 않은 것이 특징이다. 결합하는 형식은 그렇게 복잡하지 않으며, 쉼표가 쓰여서 소수점 수를 쉽게 표현한다. 비율을 나타내는 구체적인 표현에는 다음과 같은 것들이 있다.

(3-30) 우리학교 특차 입학조건은 수능 전체에서 상위 5%의 학생이면 가능하다.
(3-31) 그 기업은 매년 경제 성장률 7.6%로 성장하고 있다.
(3-32) 백퍼센트는 거의 불가능하다.

3.3.6 측도(Measure) 표현

특정한 단위를 가지고 나이, 면적, 거리, 속도, 온도, 부피, 무게, 에너지 등의 양이나 정도를 표현하는 말이다. 단위성 의존명사와 함께 사용되는 경우를 의미한다. 단위성 의존명사는 분류하는 기준에 따라 여러 가지가 있을 수 있다. 유동준[16]은 척도, 모양, 배열, 인성, 수량의 5범주로 나누고 다시 14개의 하위 범주로 세분화하였고, 서정수[17]는 길이, 넓이, 부피, 무게, 액수, 시간, 수량, 사람, 동물 등 10가지 분류를 하고, 각각에 대해서 단위성 의존명사의 항목을 평면적으로 나열하고 있다. 측도를 나타내는 구체적인 표현에는 다음과 같은 것들이 있다.

(3-33) 평균 수명은 일본이 79세로 1위이다.
(3-34) 이 집은 18평이다.
(3-35) 여기서 우리 집까지는 100미터이다.
(3-36) 다섯 명이면 할인 대상이 된다.
(3-37) 나무 한 그루를 보았다.
(3-38) 고등어 한 마리로는 부족하다.
(3-39) 장미 한 송이를 선물했었다.

3.3.7 계수(Cardinal) 표현

개체의 양이나 수를 표현하거나 수 자체를 표현하는 말을 계수라고 하며, 다른 품사와 어울리지 않고 혼자서도 쓰일 수 있는 표현을 다룬다. 주로 수식에서 이런 표현이 사용되며, 그 구체적인 표현에는 다음과 같은 것들이 있다.

(3-40) 2는 자연수이며, 2.7은 소수점 자릿수이다.
(3-41) 3십여 차례 공연이 있었다.
(3-42) 이번 사업에 1백25억을 투자하였다.

제 4 장 유한상태 오토마타를 이용한 수식 표현 인식 시스템

본 장에서는 한국어 수식 표현 인식기에 대해서 기술한다. 본 논문에서는 빠르고 비교적 정확한 한국어 수사 인식기를 구현하기 위해 유한상태 오토마타를 이용한 방법을 선택하였다.

4.1 시스템 구성

본 논문에서 수사 인식을 위해 사용한 방법은 유한상태 오토마타를 단계형으로 구성해서 각 단계에서는 특정한 형식의 수식 표현을 인식해 나가는 방법을 사용하였다. 입력은 아무런 처리를 거치지 않은 원문서가 입력되고, 크게 세 부분으로 나누어진 각각의 단계에서는 특정한 형태의 수식 표현을 인식하고, 인식된 태그가 붙은 결과가 최종적으로 출력된다.

제1층은 시간 표현을 인식하고, 제2층에서는 계수표현을 제외한 수량 표현을 인식하며, 제3층에서는 계수표현을 인식한다. 각 단계의 수식 표현은 정규표현으로 기술한다. 본 논문에서는 각 단계의 인식 결과는 다음 단계의 입력이 되며 인식된 결과는 특별한 처리 없이 다음 층으로 넘기는 우회 기법을 사용한다. 또한 특수한 경우를 제외하고는 수식 표현의 시작이 어절의 시작과 동일하므로 어절 단위 인식 방법을 사용한다. 하나의 입력에 두 개 이상의 규칙이 동시에 일치할 경우에는 가장 큰 구절을 인식하게 하여 모호성을 해소하였으며, 수사와 단위성 의존명사가 결합할 때는 가능한 모든 경우가 허용되기보다는 약간의 제약이 존재하기 때문에, 이를 위해서 단위성 의존명사를 분류하여 인식하게 하였다.

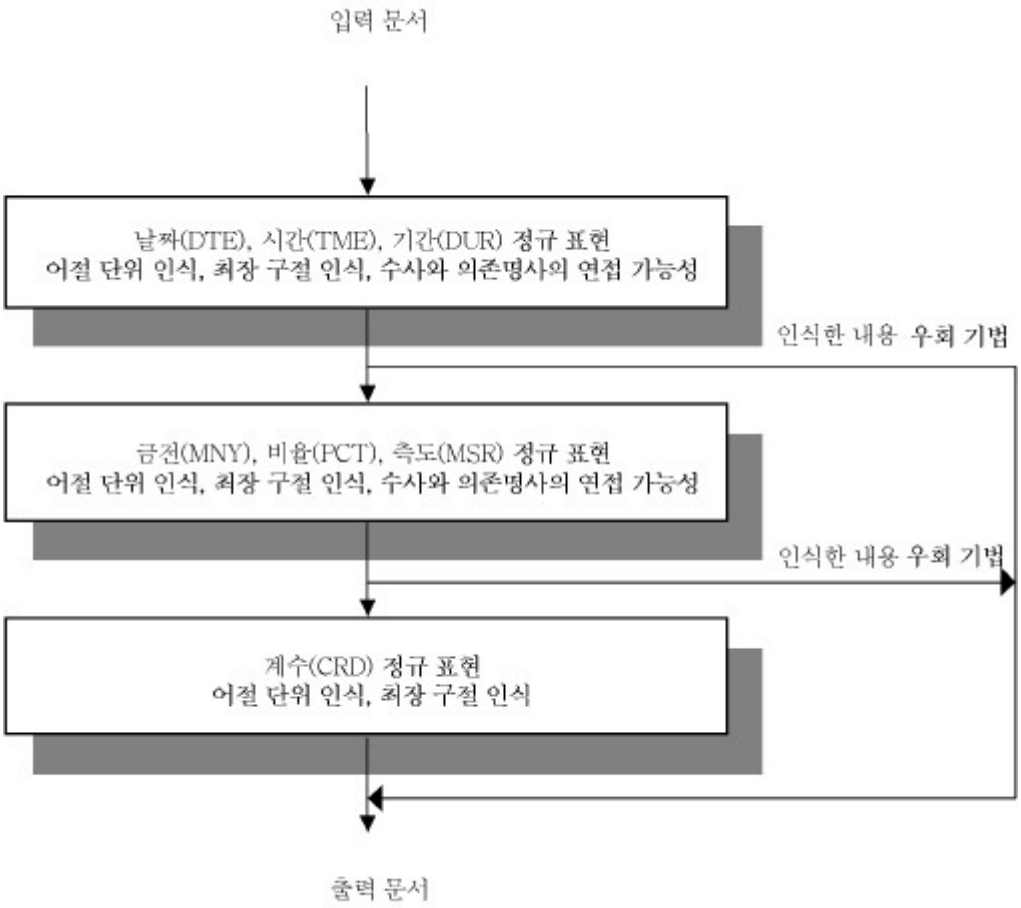

그림 2 제안된 수식 표현 인식기 개념도

Fig. 2 Overview of a numerical expression identifier

4.2 수식 표현의 상태 전이도

수식 표현 중에서 계수표현을 제외하고 모든 표현의 형식이 수사 다음에 단위성 의존명사가 나오며 그 단위성 의존명사의 종류에 따라 수식 표현의 종류를 결정할 수 있다. 수식 표현의 개략적인 유한상태 오토마타를 그림으로 표현하면 그림 3과 같다.

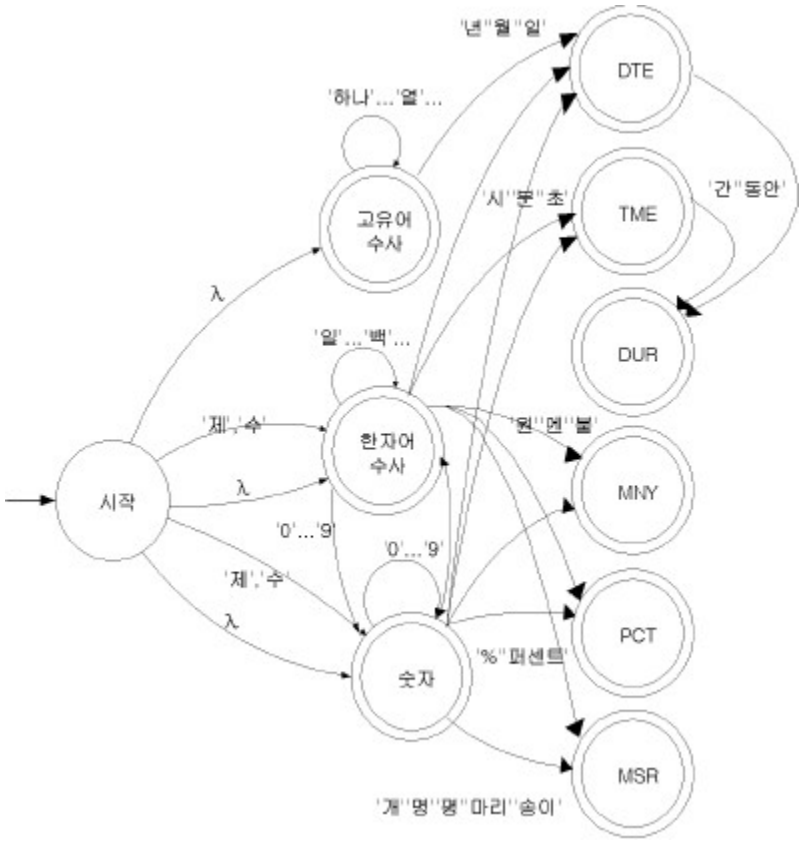

그림 3 상태 전이도로 표현한 수식 표현 인식

Fig. 3 A finite automata for recognizing numerals

그림 3에서 들어오는 화살표가 있는 원은 시작상태를 의미하고, 이중 원은 종결상태를 의미한다. 화살표는 상태 전이를 나타내며, λ는 아무런 입력이 없이도 상태 전이를 할 수 있는 기호이다.

4.3 정규표현 : 한국어 수사 인식을 위한 문법

본 논문에서 제안하는 유한상태 오토마타를 이용한 한국어 수식 표현 인식 시스템은 자연어 문서에서 수사가 포함되어 특별한 의미를 담고 있는 표현을 인식하는 시스템이다. 구현의 편의를 도모하기 위해 한국어 수사 인식을 위한 문법은 아래와 같은 정규표현에 의해서 표현되었다. 정규표현은 표현력은 제한적이

지만 표현이 용이하고, 이미 개발된 여러 형태의 도구를 이용할 수 있다는 장점을 가지고 있다. 또한 정규표현은 자동적으로 유한상태 오토마타로 변환될 수 있으며 유한상태 오토마타는 수행 속도면에서 다른 어떤 알고리즘보다 좋다. 본 논문에서는 널리 사용되고 있는 lex의 정규표현을 그대로 채용하여 한국어 수사 인식기의 문법을 기술하였다. 아래에는 그 예의 일부를 보이고 있다.

```
NNN  ([0-9])+
NNC  ("일"|"이"|"삼"|"사"|"오"|"육"|"칠"|"팔"|"구"|"십"|"백"|"천"|"만")+
NNK  ("한"|"두"|"세"|"네"|"다섯"|"여섯"|"일곱"|"여덟"|"아홉"|"열")+

NDy  ({NNN}({NNN}|{NNC})*{SP}?)+"여"?"년"
NDm  ({NNN}{SP}?)+"월"
NDd  ({NNN})+"여"?"일"

NTh  ({NNN}|{NNK})+"시"
NTm  ({NNN}{SP}?)+"여"?"분"
NTs  ({NNN}{SP}?)+"여"?"초"

NDymd       ({TENTWELVE}|{NDy}){SP}?{NDm}{SP}?{NDd}
NDym        {NDy}{SP}?{NDm}
NDmd        {NDm}{SP}?{NDd}

NThms       {NTh}{SP}?{NTm}{SP}?{NTs}
NThm        {NTh}{SP}?{NTm}
NTms        {NTm}{SP}?{NTs}
```

4.4 우회 기법

이전 층에서 인식한 확실한 내용에 대해서는 아무런 처리를 하지 않고 다음 단계로 넘기는 방법이 필요하다. 그렇지 않은 경우에는 이미 인식한 부분을 다시 인식하여 이중으로 인식하는 문제점을 가진다. 아래의 예에서 보는 것과 같이 한자어 수사에서의 '조'와 단위성 의존명사의 '조'는 그 형태가 동일하기 때문에 금전 표현 인식 층에서 인식한 내용을 수량 표현 인식 층에서 우회 방법을 사용하지 않으면 하나의 내용에 대해서 두 개의 인식 태그가 붙어서 나오는 결

과를 가져온다. 이런 문제를 해결하기 위하여 이전 단계에서 인식한 내용은 아무런 처리없이 다음 단계로 넘기는 우회 방법을 사용하였다.

입력 문장 : 7조60억원을 가지고
중간 문장 : <MNY>7조60억원</MNY>을 가지고
오류 출력 문장 : <MNY><MSR>7조</MSR>60억원</MNY>을 가지고
정상 출력 문장 : <MNY>7조60억원</MNY>을 가지고

4.5 어절 단위 인식

수사가 표현된 형식을 살펴보면, 어절의 시작에서 나타날 경우와, 어절의 중간, 그리고 어절의 끝에서 나타나는 경우가 있다. 그러나, 대부분의 경우 어절의 시작에서 시작하지 않는 경우는 인식하지 말아야 할 것을 인식하는 경우가 생기게 된다. 아래에 오류를 생성하는 예를 보였으며, 이런 잘못된 인식을 하지 않기 위해서, 인식의 시작을 어절의 시작에서부터 인식하도록 하였다.

(4-1) 진<MSR>열대</MSR>
(4-2) 추<MSR>세대</MSR>로라면
(4-3) 국<MSR>제사회</MSR>에서
(4-4) 연<MSR>세대</MSR>학교
(4-5) 초목이 무성<CRD>하나</CRD>
(4-6) 외<MSR>세배</MSR>격의
(4-7) 베<MSR>네통</MSR> 칼라
(4-8) 무역<MSR>제일주</MSR>의

그러나 다음과 같은 숫자의 경우는 어절의 가운데에 나타나더라도 명확하기 때문에 숫자로 시작하는 것에 대해서는 인식하도록 하였다.

(4-9) 주변<MSR>4강</MSR>의

4.6 최장 구절 인식

수사가 쓰인 표현에서 관련있는 표현의 조각 조각의 작은 정보를 표시하는 것보다는 의미있는 큰 덩어리를 인식해서 나타내는 것이 한번에 더 많은 정보를 정확하게 전달할 수 있다. 예를 들어, “2001년 12월 30일”과 같은 표현에 대해서 띄어쓰기 단위로 각각 날짜의 정보를 가지고 있지만 이것을 따로 따로 인식해서

표시하는 것보다는 전체를 하나의 날짜 단위로 인식해서 표시하는 것이 더 정확한 정보를 전달할 수 있다. 그리고 한글맞춤법표준안에서 명시된 것처럼 숫자를 우리글로 적을 때는 십진법에 따라 띄어쓰기 때문에 각각이 부분 부분의 덩어리로 나타날 수 있다. 이런 경우 각각에 대해서 서로 다른 정보를 표시하기보다는 가장 큰 단위에 대해서만 의미있는 정보를 표시하는 것이 더 정확할 때가 있다.

Lex에서 입력된 문장이 표현된 규칙에 대해서 매치가 이루어 질 경우, 가장 긴 규칙(longest match)과, 먼저 정의된 규칙(rule given first)을 사용하여 모호성을 해소하기 때문에 가장 긴 규칙을 먼저 선언하여, 최장 구절 인식을 하도록 하였다.

4.7 수사와 단위성 의존명사의 연접가능성

수사와 단위성 의존명사가 결합할 때는 모든 가능한 경우가 가능하기 보다는 약간의 제약이 존재한다. “10시 10분”으로 적혀 있는 것을 읽을 때에 “열시 십분”이라고 읽는 것이 보통이다. 똑같은 10을 읽는데도 시 앞에서는 열로 읽고, 분 앞에서는 십으로 읽는 것이다. 물론 말할 때에도 마찬가지이다. 이것이 정상적인 우리말이다. 한자어 수사와 고유어 수사를 사용하는 기준이 한결같지 않기 때문이다. 그러나 대체로 말한다면, 그 뒤에 오는 단위 명사와의 관계로 파악된다. 우리 겨레가 비교적 오랫동안 써 온 단위 명사 앞에서는 고유어 수사가 선택되고, 그 사용의 역사가 비교적 짧은 단위 명사 앞에서는 한자말 수사가 사용된다고 할 수 있다. 그 기준이 되는 시기가 갑오경장으로 그 당시 시간에 대한 개념은 있었지만, 분, 초에 대한 개념이 없었기 때문에, 시간 앞에는 고유어 수사가 사용되었으며, 분과 초에서는 한자어 수사가 차용되어 사용되었다[18, 19].

일반적으로 숫자는 고유어 단위성 의존명사와 한자어 단위성 의존명사와 비교적 자유롭게 결합하지만, 고유어 수사는 고유어 단위성 의존명사와만 결합하는 특징을 보인다. 또 수사가 결합하는 형태도 고유어 수사는 홀로 쓰이지만, 숫

자와 한자어 수사는 서로 혼용해서 사용되기도 한다.

작은 수를 셀 경우에는 고유어 수사가 자연스럽지만, 그 수가 20에서 30, 혹은 100을 넘는 수에 대해서는 고유어 수사와 한자어 단위성 의존명사가 결합해야 하는 경우가 있는데, 고유어를 세는 단위가 100이상의 수에 대해서는 잘 쓰이지 않기 때문이다. 이 같은 현상은 차용된 지 오래된 한자어 단위성 의존명사가 고유어 수사와 연결되는 편이 더 자연스럽기 때문이다.

그러나 최근 들어 한자어 수사 체계가 점차 우세해져 고유어 쪽을 대신해 가고 있다. 십 미만의 수에서는 고유어 수사 체계가 절대적으로 우세하지만, 이십 이상의 수에서는 한자어 수사가 고유어 쪽을 침투해 들어와 점차 우세해지고 있다. 고유어 수사와 결합하던 단위성 의존명사들도 이십 이상 되는 수와 결합할 때는 한자어 수사를 허용하는 일이 점점 더 빈번해지고 있다. 수사의 차용이 작은 수보다는 큰 수에서 더 쉽게 일어나고 있지만, 최근 서양에서 차용된 단위성 의존명사인 “미터, 그램, 킬로그램, 리터, 씨씨” 따위가 쉽 미만의 수에서마저 고유어 수사와 결합하지 않고 항상 한자어 수사와 결합하는 데에서도 잘 드러난다 [20, 21].

단위성 의존명사 중에는 고유어 수사와도 결합가능하고, 한자어 수사와도 결합 가능한 경우가 있는데, 각각에 대해서 서로 다른 의미를 가지게 된다. 예를 들어, 단위성 의존명사 “권”의 경우, “한 권”은 수량을 나타내며, “일 권”은 순서를 나타낸다.

본 논문에서는 한자어 수사만을 취하는 것과 고유어 수사만을 취하는 것으로 분류하여, 각각 한자어 단위성 의존명사와 고유어 단위성 의존명사로 구분하였다. 그리고 한자어 수사와 한자어 단위성 의존명사, 고유어 수사와 고유어 단위성 의존명사가 어울리는 규칙을 만들었고, 고유어 수사와 한자어 수사와 함께 어울리는 규칙도 만들었다.

- 도량형 단위성 의존명사 (기호) : UNIT

“㎕”|“㎖”|“㎗”|“ ℓ ”|“㎘”|“cc”|“㎣”|“㎤”|“㎥”|“㎦”
“㎙”|“㎚”|“㎛”|“㎜”|“㎝”|“㎞”|“㎟”|“㎠”|“㎡”|“㎢”|“㏊”
“㎍”|“㎎”|“㎏”|“㏜”|“㎈”|“㎉”|“㏈”|“㎧”|“㎯”|“㎐”|“㎱”

- 도량형 단위성 의존명사 (글자) : UNIT E
“다스”|“마이크로그램”|“메가바이트”|"럭스”|“루멘”|“미리”|“미크론”|“페이지”
“미터”|“밀리”|“밀리리터”|“박스”|“볼트”|“비트”|“세트”|“헥타르”|“마일”
“센치”|“센티”|“센치미터”|“스텝”|“암페어”|“야드”|“에르그”|“에르스텟”

- 고유어 단위성 의존명사 : UNIT K
“필”|“표”|“포기”|“편”|“판”|“톨”|“통”|“통화”|“토막”|“탕”|“클래스”|“개비”
“컵”|“켤레”|“자”|“자루”|“손”|“송이”|“발”|“발짝”|“방”|“방울”|“바가지”
“가구”|“가지”|“개”|“건”|“곳”|“공기”|“과목”|“구”|“군데”|“그루”|“그릇”

- 한자어 단위성 의존명사 : UNIT C
“도”|“호”|“반”|“호선”|“호실”|“회”|“푼”|“타”|“층”|“쪽”|“차”|“부작”|“호점”
“집”|“주”|“주일”|“조”|“종”|“과”|“절”|“점”|“정보”|“보”|“부”|“분기”|“중”|“범”
“승”|“열”|“선”|“문”|“몰”|“교시”|“국”|“급”|“동”|“위”|“위권”|“문항”|“차원”

- 정규표현식
({SP}|{SYMBOL})({NNN}|{NNC}{SP}?)+({UNIT }|{UNIT E}|{UNIT C})
({SP}|{SYMBOL})({NNN}|{NNK})+{SP}?“여”?{UNIT K}

제 5 장 실험 및 평가

5.1 실험 환경

제안한 방법의 성능을 평가하기 위한 평가 집합은 조선일보 94년 코퍼스 중 일부를 가지고 만들었으며, 73,547개의 어절로 이루어진 5,657개의 문장을 사람이 수동으로 태깅하여 만들었다. 평가 집합에서 모든 태그를 제거한 후, 이것을 시스템의 입력으로 해서 수행시킨 결과를 시스템 결과로 하여 평가를 하였다. 본 실험은 SUN Sparc 10 Workstation에서 이루어졌다.

5.2 평가 방법

본 논문에서는 제안한 시스템의 성능을 평가하기 위하여 재현율(*Recall*) R과 정확률(*Precision*) P, 그리고 재현율과 정확률 모두를 이용하여 시스템의 성능을 하나로 평가하는 $Fscore$를 사용하며, 각각 식 (1), (2) 그리고 (3)으로 정의한다[22].

$$R = \frac{N_R}{N_C} \tag{1}$$

$$P = \frac{N_R}{N_S} \tag{2}$$

$$Fscore = \frac{(\beta \times \beta + 1)PR}{\beta \times \beta (P + R)} \tag{3}$$

여기서 N_S 는 시스템이 인식한 문서의 전체 어절의 개수이고, N_R은 평가 집합과 시스템이 공통으로 인식한 어절의 개수이다. N_C는 평가 집합에 표시된 어절의

개수를 의미한다. 여기서 β의 의미는 재현율 R과 정확률 P의 비중을 선택할 수 있게 하는 변수로 $\beta > 1$이면 정확률의 비중을, $\beta < 1$이면 재현율의 비중을 높게 둔다는 의미이다. 일반적으로 기술적인 성능평가를 위한 β 값으로 1, 2, 5를 사용한다. 본 실험에서는 β값을 1로 두고 실험을 하였다.

5.3 성능 평가

5.3.1 전체 시스템의 성능

표 4는 시스템 전체에 대해서 평가한 재현율(*Recall*), 정확률(*Precision*), *Fscore* 이다. 재현율과 정확률이 비교적 낮은 것은 규칙을 작성할 때 신문에서 발생하는 수사에 대해서 정확히 기술하지 못했기 때문이며, 또한 장르가 비교적 제한된 점도 그 이유로 들 수 있다. 시스템 결과의 개수가 평가 집합의 개수보다 많은 이유는 시스템이 과잉생성 하였기 때문이며, 이 과잉생성은 또한 정확률을 떨어드리는 한 요인이 되었다.

표 4 전체 시스템 성능
Table 4 System performances

평가 집합 태그 수	3,862개
시스템 결과 태그 수	4,036개
일치된 태그 수	3,507개
Recall	90.8%
Precision	86.9%
Fscore	88.8%

5.3.2 각 태그별 성능

표 5에 각 태그에 따른 성능을 나타내었다. 성능이 다른 것에 비해 비교적 높게 나타난 것은 신문에서의 출현 패턴이 다양하지 않은 것이며, 상대적으로 낮게 나타난 것은 규칙을 정의할 때, 정의하지 않음으로 해서 생긴 오류들이다. 재현율의 경우 대부분은 90%이상의 성능을 나타내지만, 기간(DUR), 계수(CRD)에 대해서는 그 성능이 80% 이하로 나타난 것은 신문에 나타나는 다양한 패턴에 대해서 규칙을 첨가하지 않음으로 생긴 결과이다. 정확률의 경우 시간(TME), 계수(CRD) 항목이 현저하게 낮은 이유는 시스템이 불필요한 인식을 하였기 때문이다. 대표적인 원인으로는 잘못된 수관형사를 과잉 인식했기 때문이다.

표 5 각 태그 별 성능

Table 5 Performance for each tag

	평가 집합 태그 수	시스템 결과 태그 수	일치된 태그 수	*Recall*	*Precision*	*Fscore*
TME	73	96	72	98.6	75.0	85.2
DTE	1317	1258	1243	94.4	98.8	96.5
DUR	206	168	154	74.8	91.7	82.4
MNY	261	267	242	92.7	90.6	91.7
PCT	212	207	206	97.2	99.5	98.3
MSR	1372	1449	1291	94.1	89.1	91.5
CRD	421	591	311	73.9	52.6	61.5

5.4 오류 분석

오류 분석은 모델을 개선하거나 시스템의 성능을 개선하는 데에 많은 도움을 줄 수 있기 때문에 본 절에서는 구현된 시스템의 단계에서 발생되는 오류를 분석하고자 한다. 시스템이 과잉 생성한 태그의 수는 시스템이 생성한 전체 태그 수에서 일

치한 태그 수를 뺀 값으로 530개가 있으며, 시스템이 인식하지 못한 태그는 평가 집합의 총 태그 수에서 일치한 태그 수를 뺀 값으로 356개가 있다. 그러나 전체 분석된 오류의 수는 501개로 개수에서 차이를 보인다. 그 이유는 시스템이 생성한 태그와 평가 집합의 태그를 비교할 때, "56대44"의 경우, 평가 집합에는 <PCT>56대44</PCT>로 되어있고, 시스템의 결과는 <MSR>56대</MSR><CRD>44</CRD>로 되어 있으며, 또 "4분의 1"의 경우도 평가 집합은 <CRD>4분의 1</CRD>, 시스템 결과는 <TME>4분</TME>의 <CRD>1</CRD>의 결과를 출력하기 때문에 차이가 났다.

그림 4 오류 분석

Fig. 4 Error Analysis

5.4.1 의미적 모호성

수식 표현 중에서 형태적으로는 날짜나 수량을 나타내는 것이지만, 문장에서의 의미는 날짜나 수량을 표현하지 않는 경우이며, 다음과 같은 예가 있다.

(5-1) <MSR>한편</MSR>으로는
(5-2) 베이비 붐 <MSR>세대</MSR>는
(5-3) 명절날 <MSR>세배</MSR>하기 위하여 모였다.
(5-4) <CRD>하나</CRD>님

5.4.2 수관형사 오류

수관형사로 인한 오류는 39%를 차지하며, 성능향상을 위해서 반드시 해결해야 하며, 해결할 수 있는 문제이다. 이는 숫자에서 나타나기보다는 한자어 수사와 일반 글자가 형태적으로 동일하기 때문에 발생한 것이다. 오류로 인식된 예를 보이면 다음과 같다.

(5-5) <CRD>백</CRD>범 시해
(5-6) <CRD>백</CRD>제시대
(5-7) <CRD>둘</CRD>러싸고
(5-8) 다르기는 <CRD>하나</CRD>
(5-9) 어떻게 해야 <CRD>하나</CRD>
(5-10) <CRD>하나</CRD>회

5.4.3 범위 오류

범위 오류는 크게 두 가지로 나누어 관찰할 수 있으며, 잘못된 최장일치와, 필요한 최장일치가 있다. 필요한 최장일치는 규칙을 정의함으로써 쉽게 해결될 수 있지만 잘못된 최장일치를 방지하기 위해서는 다른 방법이 필요하다. 아래에는 시스템이 생성한 범위 오류의 예이다.

(5-11) <MSR>25개항</MSR>목에 대해서
(5-12) 정무<MS>1장</MSr>관은
(5-13) <MSR>제1차</MSR>관보
(5-14) <MSR>제2도</MSR>약
(5-15) <DUR>3일 간</DUR>염백신

5.4.4 규칙 없음 오류

규칙이 없음으로 해서 생긴 오류는 시스템이 생성한 오류를 분석하여 규칙을 첨가함으로서 쉽게 수정하여 성능향상을 기대할 수 있다. 처음에 규칙을 설계할 때 고려하지 않은 많은 새로운 표현이 신문과 같은 자연언어에서는 빈번히 발생하기 때문에 완전한 규칙을 작성하는 것은 불가능할지 모르나, 특정 장르에 발생하는 표현이 그렇게 다양하지 않은 점을 고려할 때 어느 정도의 성능 향상은 기대할 수 있

는 부분이다. 그리고, 기타 오류의 유형으로는 사람이름, 수사 + 단위성 의존명사"로 시작하는 글, 수사 + 조사 등이 있다. 아래에는 시스템에 규칙이 없음으로 인해 생긴 오류이며, 단순한 규칙의 첨가로 성능향상을 기대할 수 있는 부분이다.

(5-16) <DTE>4・19</DTE>
(5-17) <CRD>4</CRD>・<CRD>19</CRD>
(5-18) <PCT>56대44</PCT>
(5-19) <MSR>56대</MSR><CRD>44</CRD>
(5-20) B<MSR>777기</MSR>
(5-21) ISO<MSR>9001인</MSR>증
(5-22) <MSR>이세기</MSR> 정책위원장은
(5-23) <MSR>이세</MSR>중
(5-24) <MNY>이원</MNY>종 정무수석
(5-25) <MSR>한 병</MSR>원에서
(5-26) <MNY>사원</MNY>들은
(5-27) <CRD>수천만이</CRD> 넘는

5.4.5 시스템 개선 방안

본 절에서는 시스템이 생성한 오류를 분석을 통해서 개선할 방법에 대해서 살펴보고자 한다. 의미적으로 모호한 경우에 대해서는 주변 명사를 고려한다거나, 다음 글자에 대한 품사 정보를 이용한다면, 어느 정도 해결할 수 있을 것으로 생각된다. 수관형사 문제는 다음의 글자가 조사인지 아닌지를 사용한다면 시스템이 잘못 생성하는 과잉생성 문제를 쉽게 해결할 수 있을 것이다. 범위 오류에서도 잘못된 최장일치의 경우에도 조사를 사용하지 않아서 생긴 것이므로, 조사 정보의 사용은 필수적이라 하겠다. 규칙이 없음으로 해서 생긴 오류는 오류를 분석해서 이전 규칙과 충돌하지 않는 규칙을 첨가함으로써 해결될 것이라고 본다. 기타의 오류에 대해서도 조사 정보와 사람의 이름 전후에 나타나는 직함 정보나 접사 정보를 사용할 경우 어느 정도 문제가 해결될 것이라 생각된다.

시간 표현은 정보 추출과 같은 응용분야에서 필히 인식을 필요로 하는 중요한 요소가 되며, 구문 분석에 있어서는 구성 성분의 중의적 해석이 두드러지게 나타나는 부분이며, 이러한 결과로 구문 분석의 오류를 빈번히 야기하기도 한다.

(5-28) 10월 9일 저녁 7시 비행기표를 예약할 수 있다.
(5-29) 10월 9일 저녁 7시 김 대통령의 담화가 있다.

위 문장들을 시간명사와 관련하여 구문 분석해보면 크게 두 가지 경우로 나뉘는데, 첫 번째(5-28)는 시간 표현이 다음 명사를 수식하는 관형어 역할을 하여 하나의 명사구를 이루고, 두 번째(5-29)는 시간 표현이 용언을 수식하는 부사로 사용된 경우이다. 즉, 시간 표현은 다른 명사와 결합하여 복합어를 이루는 경우가 있고, 시간 부사의 역할을 하는 경우가 있다. 이는 시간 표현이 명사를 수식할지 아니면 동사를 수식할지를 결정하는 것이므로 실제 구문분석을 하기 전에 대단히 중요한 정보가 된다. 지금의 수식 표현 인식기에는 수식을 인식하는 기능만 있지만 이 인식기에 위와 같이 인식된 수식의 기능을 결정하는 기능이 첨가된다면 구문 분석의 정확률을 효과적으로 향상시킬 수 있을 것이다.

제 6 장 결 론

본 논문에서는 유한상태 오토마타를 사용하여 한국어 문서에서 시간 표현과 수량 표현에 관련된 부분을 찾아서 인식하는 방법에 대해서 살펴보았다. 대상 문장이 어느 정도 정형화된 형태를 갖는다면 문장에 대한 복잡한 파싱에 의하지 않아도 수사를 중심으로 주변의 품사를 고려해서 어절 간의 형태적인 패턴을 찾아서 인식할 수 있었다.

본 시스템은 구현이 간단하면서도 빠르며, 추출 신뢰도가 비교적 높다는 장점을 가진다. 문장 전체에 대한 파싱이나 의미 분석에 의존하지 않고, 수사를 중심으로 주변품사에 대해서 파악하기 때문에 처리 과정이 간단하며 정확한 정보를 인식할 수 있다. 또한 일반 파싱에서 문제가 되는 어절 수에 제한 받지 않으므로 긴 문장 내의 정보도 오류없이 추출할 수 있다.

반면 유한상태 오토마타의 제한점으로 인해 정형화 되어있지 않은 형태로 작성된 문서에서는 효과적으로 정보를 인식하는 것이 어렵다는 단점이 있다. 즉, 일관성이 없는 패턴의 문장에 대해서는 새로운 패턴을 다시 정의해서 사용해야 한다.

본 논문에서는 시스템의 성능을 평가하기 위하여 조선일보 94년 기사 중 일부를 수동으로 태깅하여 평가 집합으로 사용하였다. 객관적인 평가를 위해서 정보검색에서 사용하는 평가 기준인 정확률과 재현율을 사용하였으며, 평가 집합과 시스템 결과물을 비교하여 얻은 전체 성능은 정확률과 재현율이 각각 86.9%와 90.8%를 보였다. 오류 분석을 통해 단순한 규칙의 첨가나 조사와 같은 주변 품사 정보를 고려한다면 성능향상을 기대할 수 있을 것이다.

본 시스템에서는 수사와 단위성 의존명사같은 비교적 단순한 정보만을 사용하였지만, 조사와 같이 좀 더 확장된 정보를 사용한다면 인식 대상이 되는 문서

가 보다 복잡한 경우에도 성능향상을 기대할 수 있다. 그리고 전처리 과정으로 띄어쓰기에 대한 처리[23-25]를 하거나 셈의 대상이 되는 명사에 대한 정보[15]를 사용함으로써 정확한 수식 표현의 범위를 인식할 수 있을 것으로 기대할 수 있다.

참고 문헌

[1] Steven Abney, "Parsing by Chunks," Kluwer Academic Publisher, 1991.

[2] Salah Ait-Mokhtar, J. P Chanod, "Incremental finite state parsing," In ANLP'97, 1997.

[3] 김재훈, "부분 구문분석 방법론," 정보처리학회지, 제7권, 제6호, pp. 83-96, 2000.

[4] 최명석, "한국어 부분 구문분석," 한국과학기술원, 전산학과, 석사학위 논문, 1998.

[5] 오효정, 임정묵, 이만호, 맹성현, "유한 오토마타를 이용한 정보 추출 시스템의 구현 및 분석," 제10회 한글 및 한국어 정보 처리 학술대회 발표 논문 pp. 3-8, 1998.

[6] Douglas E. Appelt, Jerry R. Hobbs, John Bear, David Israel, and Mabry Tyson, "FASTUS : A finite-state processor for information extraction from real-word text," in Proceedings IJCAI '93, Chambery, France, August 1993.

[7] 강승식, 우종우, 윤보현, 박상규, "정보 추출," 정보과학회지, 제19권, 제10호, pp. 27-39, 2001.

[8] N. Chinchor, P. Robinson, and E. Brown, Hub4 "Named Entity Task Definition (version 4.8)," SAIC, August 1998.

[9] Lesk, M.E. & Schmidt, E. "Lex : A Lexical Analyzer Generator. Anonymous (Ed.), Unix Research System Papers, Tenth Edition. Murray Hill, NJ: AT&T Bell Laboratories," 1990.

[10] 김대수, 오토마타와 계산이론, 생능출판사, 1996.

[11] 오세만, 컴파일러 입문, 정익사, 1994.

[12] 김영길, 강석훈, 우요섭, 김한우, 최병욱, "자연언어에 의한 질의응답 시스템의 설계," 제4회 한글 및 한국어 정보 처리 학술대회 발표 논문 pp.469-477, 1992.

[13] Donald Hindle. "User manual for Fidditch. Technical Report 7590-142, Naval Research Laboratory," 1998.

[14] 김윤관, 윤준태, 송만석, "시간 표현에 대한 부분 문법 기술 및 FST를 이용한 시간 구문 분석," 제11회 한글 및 한국어 정보 처리 학술대회 발표 논문 pp.231-236, 1999.

[15] 최민우, 강범모, "분류사와 명사 의미 부류," 제12회 한글 및 한국어 정보 처리 학술대회 발표 논문 pp.395-401, 2000.

[16] 유동준, "국어의 분류사와 수량화," 국어국문학 89, 국어국문학회, 1983.

[17] 서정수, 국어문법, 서울, 1994.

[18] 류영남, 이론과 실제로 본 우리글 적기의 바른 길, 태화 출판사, 1986

[19] 리의도, 말을 잘하고 글을 잘 쓰려면 꼭 알아야 할 것들, 석필, 2000.

[20] 방성원, 호정은, 김종인, "세종 의존명사/대명사/수사 전자사전의 정보표상 구조," 제13회 한글 및 한국어 정보 처리 학술대회 발표 논문 pp.341-347, 2001.

[21] 유재원, "자연어 처리를 위한 의존 명사 하위범주 분류," 제9회 한글 및 한국어 정보 처리 학술대회 발표 논문 pp.136-142, 1997.

[22] Frakes, W. B. and Baega-Yates, R., Information Retrieval : Data Structures & Algorithms, Prentice Hall, 1992.

[23] 심광섭, "음절간 상호 정보를 이용한 한국어 자동 띄어쓰기," 정보과학회

논문지(B), 제23권, 제9호, pp.991-1000, 1996.

[24] 강승식, "한글 문장의 자동 띄어쓰기," 제10회 한글 및 한국어 정보처리 학술대회, pp. 137-142, 1998.

[25] 강승식, "음절 bigram 특성을 이용한 띄어쓰기 오류의 인식," 제12회 한글 및 한국어 정보처리 학술대회, pp. 85-88, 2000.

감사의 글

본 논문이 완성되기까지 먼 곳에서 ICQ로, 메일로 질책해 주시고 이끌어 주신 김재훈 교수님께 깊이 감사드립니다. 교수님의 무한한 인내심과 자상한 지도가 없었다면 지금의 이 논문은 완성되지 못하였을 것입니다. 그리고 바쁘신 중에도 심사를 맡아 지도와 조언을 해주신 박휴찬 교수님과 신옥근 교수님, 그리고 손주영 교수님께도 감사를 드립니다. 또한 학부와 대학원에서 제게 가르침을 주신 류길수 교수님께도 감사를 드립니다. 너무 부족한 제가 여러 훌륭한 교수님들로부터 가르침을 받을 수 있었던 것을 큰 행운으로 여깁니다.

저희 자연언어 처리 연구실에서 같이 지낸 후배 박호진에게는 지난 3년 동안 부족한 선배를 잘 따라 주어서 고맙다는 말을 전하고 싶습니다. 그리고 먼저 졸업한 김준홍 선배님께는 대학원 생활을 통해서 그 시간동안 관심과 인내로 지켜봐 주신 것에 대해서 감사함을 전하고 싶습니다. 학부와 대학원에서 좋은 모습을 보여주신 하동경 선배님, 안명석 선배님께도 감사한 마음을 전하고 싶습니다. 작은 일에도 신경 써 주신 강군호 조교님과 김경언 조교님께도 이 자리를 빌어서 고마운 마음을 전해야 할 것 같습니다. 힙합 음악을 전염시킨 원룡이형, 학부 동기이며 대학원 생활을 같이 했던 대근이와 재환이, 대학원 후배(?)님인 언규형, 형석이형, 실험실로 찾아와 커피를 함께 마셨던 동기 정태, 기영, 영조, 만식, 한우형, 멀리 군대에 있는 동기 승현이, 그리고 형의 실수를 이해해 준 순현이와 실험실의 동재에게도 고마움을 전하고 싶습니다.

철든 귀염둥이 규술이, 사실을 거짓말처럼 이야기하는 기훈이, 말만 안하면 킹카인 동식이, 너무나 착한 동기 은실이, 투덜이 스머프 상훈이, 롱다리 민구, 아직 여자 손 한번 못잡은(?) 현이, 지금은 연락이 뜸한 정혜와 기쁨을 함께 하고 싶습니다. 군에 입대할 때 먼길을 같이 가주었던 고마운 친구 해준이와 또 힘들

때 나를 찾아 준 광태, 좋은 말동무가 되어준 준호, 귀엽고 착하고 재미있는 준석이에게도 짧은 군생활이었지만 커다란 기쁨이 되어주어서 고맙다고 전하고 싶습니다. 흔들리지 않는 나무같은 기형이형, 나이보다 젊게 사는 은희누나, 너무 착하게 살고있는 윤희누나, 풍부한 경험과 확신을 가진 영석이형, 밥 잘 사는 친구 재만이에게도 감사드리고, 또 여기에 미처 이름을 적지 못한 사람들에게도 감사의 마음을 전하고 싶습니다.

고등학교 때 영어 공부에 흥미를 갖게 해주신 이정명 선생님과 논리적으로 생각을 할 수 있게 가르쳐 주신 서만률 선생님, 그리고 아버지처럼 자상하게 학생들을 가르쳐 주셨던 김학봉 선생님께 깊이 감사드립니다. 청년의 때에 미래를 보며 비전을 품을 수 있게 해주신 정익주 목사님께도 감사의 마음을 전하고 싶습니다. 대학원에서 만난 영어과 이건수 교수님께도 수업을 통해서 많은 것을 배울 수 있었기에 고마운 마음을 전합니다.

항상 저 때문에 걱정하시는 부모님과 가족에게 고마움을 전하고 싶습니다. 아들을 믿고 지켜봐 주신 어머니와 아들 걱정을 보이지 않게 하셨던 아버지, 동생을 위해서 아낌없는 지원을 해 준 누나, 지금은 힘든 환경에 있지만 금방 털고 일어설 형, 저를 친동생처럼 잘 배려해 준 하나 뿐인 우리 매형과 미래의 꿈나무들인 조카들에게도 고마움을 전하고 싶습니다. 저의 모든 어려움과 필요를 아시고 충분히 넘치도록 채워주셨던 하나님께 부끄럽지만 이 모든 기쁨과 영광을 돌리고 싶습니다.

2년의 시간이 지나고 나서야 대학원 생활이 짧은 줄 알았습니다. 존경하는 교수님께 관심있는 분야를 더 배울 수 있었고, 실력있는 선배님과 후배님을 통해서 서로의 생각을 이야기 할 수 있었던 때가 너무 좋았었습니다. 생각해 보니 저는 지금까지 너무 많은 것을 받고만 살아온 것 같습니다. 이제는 소중한 사람들에게 작은 것 하나라도 함께 나누면서 살고 싶습니다.