실용적인 영한 기계번역을 위한 전처리기의 설계 및 구현

A Preprocessor for Practical English-to-Korean Machine Translation

여상화, 정한민, 채영숙, 김태완, 박동인

Sanghwa Yuh, Hanmin Jung, Youngsoog, ChaeTaewan Kim, Dong-In Park

기계번역연구실/자연어정보처리연구부/시스템공학연구소

Machine Translation Lab., Dept. of Natural Language Information Processing, SERI

요약

본 논문에서는 실용적인 기계번역 시스템을 위하여 다양한 입력 형태에서 나타나는 여러 현상을 전처리하는 기법을 설명한다. 전처리기는 문장 분리, Title 및 나열문 인식, HTML Tag 의 처리, 하이픈처리, 숫자 표현 처리, 대소문자의 정규화, 고유명사 인식, 복합 단위 인식 등을 수행하여 형태소 분석기의 처리 부담을 줄인다.

1. 서론

실용적인 기계 번역 시스템을 위해서는 번역하고자 하는 입력 문서의 자동화가 이루어져야 한다. 최근 WWW(World-Wide Web)의 비약적인 확산과 OCR(Optical Character Recognition) 기술의 발달 및 스캐너의 보급으로 실용적인 기계번역 시스템의 걸림돌이 되어 온 입력 문서의 자동화는 어느 정도 만족할 만한 수준에 이르렀다. 그러나, 현재까지 출시된 상용 영한 기계 번역 시스템들은 번역 문서의 입력 형태에 따른 고려가 미비하여 사용자가 번역하고자 하는 문서를 재편집해야 하는 불편이 따랐다. 웹을 통한 HTML 문서의 경우에는, HTML Tag 의 처리가 안되어 Tag 를 사전에 제거하거나 Drag&Drop 기능을 이용하여 화면에 나타난 Text 만을 번역시키고 있다[IBM96][정소프트 96]. OCR 로 인식한 문서인 경우에는, 라인의 끝에서 단어를 잇기 위해 사용된 하이픈(-)을 처리 하지 못하여 사용자가 일일이 입력 문서를 수정해 줘야하는 불편이 있었다.

본 논문에서는 입력 문서에서 나타나는 다양한 현상을 형태소 분석기가 용이하게 처리할 수 있도록 전처리하는 기법에 대해 설명한다. 2 장에서는 입력 문서에서의 고려 사항과 기존 시스템에서의 처리 능력을 보이고 3 장에서는 전처리기의 기능을 제시하고 각 기능별로 문장 분리, 단어의 분리, 대소문자의 정규화 방법 그리고 복합 단위 인식의 방법을 설명

한다. 4 장에서는 실험 결과를 제시한다.

2. 상용 영한 기계번역시스템의 문제점

표 1 에서는 실용적인 기계번역을 위해 입력 문서에 대한 몇가지 고려 사항과 현재 상용화된 영한 기계번역 시스템에서 처리 능력을 보여 주고 있다. mates/ek 는 당 부서와 KAIST 전산학과가 공동개발한 영한 번역시스템으로 상용화되지는 않았다[시스템 92][과기원 92].

표 1. 입력 문서에 대한 고려 사항 및 상용 영한 번역 시스템의 성능

입력 문장에서의 고려 사항	mates/EK	EnKor V1.0	번역마당 V1.5	WordChange 2.5
한 문장이 여러 줄인 경우의 처리	X	○	○	○
TITLE/나열문 처리	X	X	△	X
HTML 문서의 처리	X	X	X	X
대문자로 작성된 문장 처리	X	○	X	△
단어 연결을 위한 하이픈 처리	X	X	○	X
사전에 수록되지 않은 고유명사 처리	○	○	○	○

X: 처리 못함, △: 일부 처리함, ○: 처리함

대부분의 시스템이 미등록어에 대한 처리는 잘 수행하지만 입력 문서에 대한 고려가 부족함을 알 수 있다. 이는 사용자의 개입을 요구함을 의미하며 기계번역 시스템의 이용에 큰 장애가 된다.

3. 전처리기의 기능

전처리기는 입력 문서에서 나타나는 다양한 현상을 형태소 분석 전단계에서 미리 처리하여 형태소 분석기를 보조한다. 전처리기가 수행해야 하는 기능은 크게 문장의 분리와 단어의 인식 그리고 문자의 정규화 그리고 복합 단위의 인식으로 나눌 수 있다.

기존의 번역 시스템에서는 위의 기능 중 일부를 형태소 해석 단계에서 수행하고 있다[김용변 94][과기원 92]. 그러나, 이들을 형태소 해석 단계에서 처리하기에는 다루어야 할 예외 현상이 많고 Domain 에 따라, Text 에 따라 특이한 경우가 많다. 또한, 새로운 예외 현상들이 계속 발견될 수 있으므로 형태소 해석 프로그램과 분리하여 전처리 단계에서 수행하는 것이 바람직하다. 또한, 이를 위해서는 전처리기 자체가 새로운 언어 현상에 쉽게 대처할 수 있는 형태로 작성되어야 한다.

3.1 문장 분리

영어의 문장은 주로 종결기호('.', '!', '?' 등)로 구분되지만 Title 이나 나열문인 경우에는 종결기호가 사용되지 않는다. 이경우, 이들을 별도의 문장으로 분리하지 않고 처리하면 파싱에 실패하게 된다. 문장 분리 단계에서는 입력 문서를 문장 단위로 분리하며 복합 단위 인식 단계의 편이성을 위해 한 문장이 한 라인이 되도록 한다.

Title 이나 나열문인 경우에는 다음과 같은 경험적인 정보(Heuristics)를 이용하여 문장을 분리한다.

(1) title 문장은 모두 대문자로 작성되거나, 기능어를 제외한 모든 단어의 첫글자를 대문자로 나타낸다.
(2) title 문장의 선두에는 장, 절 등을 표시하는 번호가 나타난다.
(3) HTML 문서인 경우, <TITLE> 과 같은 Tag 를 이용한다.
(4) 나열문인 경우 문두에 글머리표가 나타난다.

문장 분리 단계에서는 종결기호가 단어의 일부로 사용되는 경우(예: I.B.M., O.K.)에 대한 예외 처리도 수행해야 한다. 특히 이들 단어가 문장의 마지막에 사용된 경우에는 생략된 종결 기호를 복원한다.

3.2 단어의 인식

3.2.1 하이픈 처리

하이픈은 복합어에 사용되기도 하고, 라인의 끝에서 단어가 미처 끝나지 않은 경우, 그 단어가 다음 라인에 이어짐을 표시하기도 한다. 라인의 마지막 단어가 하이픈으로 끝나는 경우, 하이픈을 단어의 일부로 보아야 할 지, 아니면 무시해야 할 지 판단해야 한다.

라인의 끝에서 하이픈으로 끝나는 경우, 이 하이픈이 단어의 일부인지 아니지를 판단하기 위해 하이픈 단어 사전을 이용한다. "re-creation(재창조; 개조)" 과 "recreation(휴양;오락)"과 같이 하이픈의 유무에 따라 별개의 단어로 인식되는 경우에는 두가지 모두를 출력한다.

3.3.2 고유 명사의 인식

기존의 기계번역 시스템에서는 고유 명사 인식을 별도로 수행하지 않았다. 단순히 사전 탐색에 실패한 단어에 대해 첫 글자가 대문자이면 고유명사로 추정하였다. 본 시스템에서는 사전과 규칙에 의해 고유 명사를 인식하며 인식된 고유 명사에 대하여 FEMALE, MALE, HUMAN, PET, ORG(조직) 등의 정보를 부여한다.

고유 명사 사전에는 흔히 사용되는 남자와 여자의 First name 과 애완 동물 이름이 수록되어 있다. 표 2 와 같은 형태의 고유명사는 정규 표현(Regular Expression)을 통해 인식되며

"John F. Kennedy" 또는 " Stephen Akerfeldt"와 같이 First name 이 사용된 경우에는 이를(예: John, Stephen)을 분리하여 고유 명사 사전을 탐색하여 FEMALE, MALE, HUMAN 의 정보를 부가한다. 또한, Mr., Ms., Mrs., Dr. 등의 tiltle 과 함께 last name 이 사용된 경우에는 이들 title 을 이용하여 FEMALE, MALE 정보를 부가 한다.

[보기] ("John F. Kennedy" :PROP FEMALE), ("Mr. Kim" :PROP MALE)

표 2. 고유 명사의 표현(일부)

John F. Kennedy	Aho, A. V.	Bach, E.	J.D. Ullman
K. Jones	Stephen Akerfeldt	Mr. Kim, Mr. O'Kicki	T. Marshall Hahn Jr.

사람 이름 뒤에 "& Co." 등이 나오는 경우에는 조직명으로 인식하며 ':PROP ORG' 정보를 부여한다. 두개의 First name 이 '&'로 묶이면 사람이름이거나 회사명이므로 단순히 ':PROP' 정보를 부여한다. (예: '**Johnson & Johnson**', 'Aho, **Sethi & Ullman**')

3.2.3 축약형 처리

Be 동사, 조동사에 사용되는 축약형은 결합되어 있던 주어를 분리하고 원래의 형태로 복원한다. 이 경우 "I'd"의 경우에서와 같이 조동사의 축약형, "'d"는 would, should, could, had 의 4 가지가 가능하다. 원형이 여러 가지가 가능한 경우 동사의 축약형을 분리하여 출력한다. 그렇지만 "I'd like to"와 같이 다음에 오는 단어를 보아 하나로 결정할 수 있는 경우에는 하나의 형태로 복원한다.

[보기] "He'd" ==> ("He") ("'d"), "I'm" ==> ("I) ("am"), "I'd like to" ==> ("I") ("would") ("like") ("to")

3.2.4 HTML Tag 의 처리

Start Tag, End Tag , Attribute 를 가지는 Tag 그리고 HTML 의 Comment 문을 정규 표현을 사용하여 HTML 의 Tag 로 인식한다. 인식된 Tag 들에는 ':TAG HTML'의 정보를 부여하여 형태소 분석기가 이들을 불필요하게 분석하지 않도록 한다.

3.2.5 특수 기호의 처리

특수문자는 고유 명사에서처럼, 특수 문자가 단어의 일부로 사용되지 않은 경우에는 이를 단어로 인식한다. 복수형에 사용되는 따옴표는 제거하고 단어에 복수형이라는 정보를 부가한다. HTML 문서에서 사용하는 Character Entity(예: ø = ø)는 단어의 일부로 처리한다.

[보기] "AT&T" ==> ("AT&T"), "Strøe" ==> " Strøe "

3.3 대소문자의 정규화

영어 문서에는 알파벳의 대문자와 소문자가 사용되며 대문자인 경우, 문장의 처음 단

어와 고유명사에서 사용된다. 그러나 실세계 문서에서는 고유명사가 아니라 하더라도 Title 이나 자신이 강조하고 싶은 단어에는 흔히 대문자가 사용된다.

[보기]

CHAPTER 7 AMBIGUITY RESOLUTION :STATISTICAL METHODS

7.1 Basic Probability Theory
Intuitively, the probility of some event is the likelihood that it will occur.

따라서 이들 문서는 대소 문자가 정규화된 형태로 변형하여 처리해야만 동일한 단어가 사전에 여러 번 등록되는 것을 방지할 수 있고 대문자로만 작성된 문서도 일반 문서와 동일하게 처리할 수 있다.

정규화는 단순히 모든 문자를 소문자로 바꾸어 처리할 수도 있다. 그러나, 대문자(Capitalization)는 문장 내에서 단어를 규명하는데 매우 중요한 역할을 한다[GORREL65]. 예를 들어, 'White House'는 백악관이라는 의미이지만 'white house'는 단순히 하얀 집을 의미한다. 이와 같이 Capitalization 은 고유 명사를 인식하는 중요한 단서로서 사용된다. 따라서, 입력 문서에서 나타난 대문자를 모두 무시하는 것은 바람직하지 않다. 그러나, 문장이 대문자로만 작성된 경우에는 Capitalization 을 이용한 고유명사 인식은 불가능하고 문맥을 통하여 판단할 수 밖에 없다.

본 시스템에서는 고유명사를 제외한 모든 영어 단어(문장의 첫 단어를 포함)는 소문자로 정규화 한다. 이를 위해 고유명사 사전을 이용한다. 고유명사 사전에는 한 단어 짜리 고유 명사와 두 단어 이상의 고유 명사들이 수록되어 있으며, 두 단어 이상의 고유 명사들은 검색의 편의를 위해 Under line 으로 묶여 있다. (예: "United_States_of_America")

대소문자 정규화 방법은 첫글자가 대문자인 여러 단어를 처리하는 경우와 한 단어가 모두 대문자(또는 첫글자만 대문자)인 경우의 두가지로 분리하여 처리한다.

```
Algorithm  Normalization_WORDS(WORDS);  /* 두 단어 이상 */
N_WORDS = Blank 를 Underline 으로 변형하여 한 단어로 만듦;
N_WORDS 를 고유명사 사전에서 탐색;
if (탐색 성공시) N_WORDS 와 사전정보를 출력;
else { Head = WORDS 의 첫 단어; Tail = WORDS 의 나머지 단어;
     Normalization_A_WORD(Head) ;
     Normalization_WORDS(Tail);
}

Algorithm  Normalization_A_WORD(A_WORD); /* 한 단어 */
switch(A_WORD)
case 모두 대문자   :          A_WORD 의 첫 글자만 대문자로 변형;
case 첫글자만 대문자 :        A_WORD 를 고유명사 사전에서 탐색;
                              if (탐색성공) {Segment(A_WORD); break;}
                              else { A_WORD 를 모두 소문자로 변형;
default:                      Segment(A_WORD)
}
```

위의 알고리즘에서 정규화된 단어를 바로 출력하지 않고 별도의 함수 Segment를 통해 출력하는 것은 대소문자의 정규화 문제는 다른 문제와 복합된 형태로 나타날 수 있기 때문이다.

..... OF COUPS AGAINST ANTI-AMERICAN LEADERS.	(..... ("of") ("coups") ("against") ("anti-American")("leaders"))

a. 입력 문서 b. 전처리 결과

그림 1. 하이픈과 대문자 문제가 복합된 형태의 처리 예

위의 예에서 "AMERICAN"은 "American"으로 정규화 되고 Segment()함수에 의해 출력되는 과정에서 하이픈 단어의 일부라는 것이 판명되어 앞 라인에서 정규화된 "anti"와 결합하여 "anti-American"으로 출력된다.

3.4 복합 단위의 인식

여러 단어로 이루어진 수 표현(Numeric Expression), 단위 명사를 포함하는 수 표현, 날짜 표현, 인명, 기관명 그리고 고정 형태의 표현(Frozen Expression) 등은 각각의 단어를 개별적으로 해석하는 것은 의미가 없다. 따라서, 이들은 하나의 단어로 취급하는 것이 이후의 해석 모듈의 부담을 줄인다.

[보기] " Aug. 31, 1987 " ==> ("Aug. 31, 1987" :DATE :K_LEX "1987 년 8 월 31 일")
"$133.8 million" ==> ("$133.8 million" :NUM :K_LEX "1 억 3 천 3 백만 8 천 달러")
"one hundread and seventy-five" ==> ("175")

복합 단위의 인식은 지역적으로 인접한 단어들에 대해서만 제한적으로 수행하며 문장분리, 대소문자 정규화가 수행된 결과에 대해 인식을 수행한다. 따라서, 복합 단위 사전의 표제어도 대소문자가 정규화된 형태로 수록된다.

4. 구현 및 실험

본 논문에서 제안한 전처리기는 Unix의 flex와 bison 툴을 이용하여 c언어로 구현되었다[JOHN93]. flex와 bison을 이용한 전처리기는 프로그램의 유지 보수가 매우 용이하여 새로운 현상에 대해 쉽게 대처할 수 있고 수많은 예외 현상들을 매우 쉽고 빠르게 처리할 수 있다.

4.1 시스템의 구성

그림 2는 전처리기의 구성도이다. 전처리 과정은 PASS1과 PASS2의 2단계로 수행된다. PASS1에서는 하이픈 단어 사전과 고유명사 사전을 이용하여 문장의 문리, 단어의 분리, 대소문자 정규화를 수행한다. PASS 2에서는 복합 단위 인식을 수행한다.

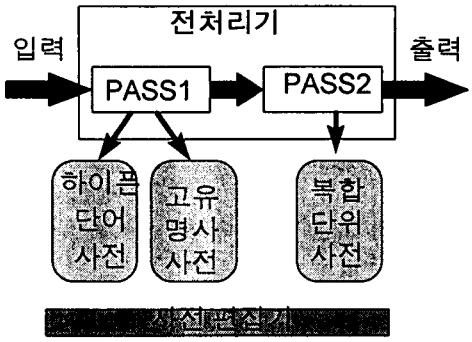

그림 2. 전처리기의 구성도

고유 명사 사전은 Mark Kantrowitz 가 수집한 영어의 이름 리스트(V1.3)와 mates/ek 및 PennTree Bank 에서 수집한 두 단어 이상의 고유 명사들로 구축되었다. Mark Kantrowitz 리스트에는 흔히 사용되는 남자 이름 2,940, 여자 이름 4,987 그리고 애완 동물 이름 18 개가 수록되어 있다.

하이픈 단어 사전에는 PennTree Bank 와 mates/ek 에서 수집한 단어 7,254 개가 수록되어 있다. 하이픈이 사용된 단어 중에서 분수와 숫자 21~99 를 나타내는 경우와 '1%-a-year', '1,400-lawyer', '1.5-mile', '1/2-inch', '1:30-6', '3rd-Period' 와 같이 숫자와 결합된 형태는 정규 표현에 의해 처리하므로 사전에는 수록하지 않는다. 하이픈 단어가 문장의 중간이나 문장의 선두에 나타날 수 있으므로(예: BUY-OUT, Buy-out, buy-out) 대소문자의 정규화와 병행하여 처리되므로 정규화된 형태로 수록된다. 고유 명사 사전과 하이픈단어 사전은 unix 의 dbm 을 개선한 GNU 의 gdbm library 를 이용하여 hash 사전으로 구축되었다.

복합 단위 사전은 flex 의 specification rule 로 기술되어 있으며 PennTree Bank 와 mates/ek 에서 수집한 복합 단위어 8,000 여개로 구성하였다.

4.2 실험

전처리기의 성능을 처리 속도와 정확률의 관점에서 실험하였다. PennTree Bank 의 Raw Corpus 중 Wall Street Journal 전체 문장(6,895,590 Byte)을 Sun Ultra 1(167Mhz)에서 처리하는데는 23.8 초(Real Time:23.8, User Time:21.8, System Time:1.0)가 소요되었다. 정확률을 Test 하기 위해 3 종류의 문서를 선정하였다. 문서 1 은 PennTree Bank 에서 추출한 WSJ 중 200 문장이고, 문서 2 는 코펜하겐 관광 가이드에 대한 HTML 문서이고, 문서 3 은 Unix 의 wc(word count program)의 매뉴얼 전문이다.

표 3. 전처리기의 실험 결과

구분	문서 1(WSJ)					문서 2(HTML)					문서 3(WC)				
	R	S	C	I	M	R	S	C	I	M	R	S	C	I	M
문장 및 title 분리	200	200	200	0	0	48	28	28	0	20	47	22	22	0	25
라인끝의 하이픈	0	0	0	0	0	0	0	0	0	0	4	4	4	0	0
고유명사 인식	100	51	51	0	49	42	22	22	0	20	0	0	0	0	0
축약형 처리	51	51	51	0	0	1	1	1	0	0	0	0	0	0	0
HTML Tag 처리	0	0	0	0	0	141	141	141	0	0	0	1	0	1	0
특수기호 처리	432	432	432	0	0	15	15	15	0	0	102	102	100	0	2
대소문자 정규화	491	491	486	5	0	145	132	132	0	13	45	45	45	0	0
복합단위 인식		72	72				10	10				8	8		
크기(Byte)	17,388					4,966					2,770				
수행 시간	Real :0.3 User:0.2 Sys:0.0					Real :0.2 User:0.1 Sys:0.0					Real :0.1 User:0.0 Sys:0.0				

주) R :실제수, S:시스템에서 인식한 수, C:옳게 인식한 수, I:잘못인식한 수, M:인식하지 못한 수

실험 결과 라인끝에서의 하이픈 처리, 축약형 처리, HTML Tag 처리, 특수기호 처리, 대소문자의 정규화 등이 매우 성공적임을 알 수 있다. 반면에 문장 분리와 고유 명사 인식 기능이 상대적으로 미약하였다. 특히, 문서 3 의 경우 문장이 분리되어야 하나 이를 인식할 만한 근거가 전혀 없는 경우가 많았다.

[보기]

```
EXAMPLE
        example%
        wc /usr/share/man/man1/{csh.1,sh.1,telnet.1}
             1876   11223   65895  /usr/share/man/man1/csh.1
              674    3310   20338  /usr/share/man/man1/sh.1
              260    1110    6834  /usr/share/man/man1/telnet.1
             2810   15643   93067  total
        example%
```

위의 보기와 같은 문장들은 번역 대상도 아니므로 이들을 분리하기 위한 별도의 처리가 필요하다.

5. 결론

본 논문에서는 실용적인 기계번역 시스템을 위하여 문장 및 Tiltle 인식, 고유명사 인식, HTML Tag 의 분리, 하이픈 처리, 대소문자의 정규화, 복합 단위 인식 등을 수행하는 전처리기를 구현하였다. 전처리기는 HTML 문서, OCR 문서 등 다양한 종류의 번역 대상 문서를 형태소 분석기가 쉽게 처리할 수 있는 형태로 바꾸어 형태소 분석기의 처리 부담을 크게 줄인다. 또한, 다양한 정규 표현과 문법 규칙을 제공하는 flex 와 bison 을 이용하여 전처리기를 구현함으로써 새로운 현상을 쉽게 반영할 수 있고 프로그램의 유지 보수가 용이하며 처리속도가 매우 빠르다.

실험 결과 고유명사 인식, HTML Tag 의 분리, 하이픈 처리, 대소문자의 정규화, 복합 단위 인식 기능이 우수하였으나 문장 및 Title 분리 기능이 미약하였다. 앞으로의 연구 과제

는 문장 및 Title 분리 기능을 강화하는 것과 번역 대상이 아닌 부분을 사용자가 미리 Mark-up 하여 불필요한 처리를 수행하지 않도록 하는 사용자 인터페이스를 개발하는 것이다.

참고 문헌

[COLLINS90] Collins Cobuild English Grammar, Collins, 1990

[GORREL65] Gorrel and Laird, Modern English Handbook, Prentice Hall, 1965

[IBM96] IBM, IBM 영한기계번역시스템 앙꼬르 버전 1.0 사용자 안내서, IBM, 1996

[JOHN93] John R. Levine, Tony Mason & Doug Brown, *lex & Yacc*, O'Reilly & Associates, Inc, 1993

[과기원 92] 한국과학기술원, 영한기계번역시스템(III):문법개발지원환경 및 해석문법개발, 과학기술처, 1992

[시스템 92] 시스템공학연구소, 기계번역을 위한 언어모델링 및 골격시스템 개발(3), 과학기술처, 1992

[김용변 94] 김용변, 연어정보와 숙어에 기반한 영어 어휘 분석기의 구현, 서울대 컴퓨터공학과 석사학위 논문, 1994

[번역마당 96] HY 정보기술, 번역마당 V1.5 사용자 설명서, HY 정보기술, 1996

[정소프트 96] 정소프트, 워드체인지 2.5 사용설명서, 정소프트, 1996