영한기계번역에서의 전처리에 관한 연구

김 성 동

한성대학교 컴퓨터공학과

sdkim@hansung.ac.kr

A Study on Preprocessing in English-Korean Machine Translation

Sung-Dong Kim

Dept. of Computer Engineering, Hansung University

요 약

영한기계번역은 영어와 한국어 사이에 많은 언어적인 차이가 존재하며 이를 효과적으로 해결해야 한다. 규칙기반의 영한기계번역에서는 언어간의 차이를 어휘, 구문, 변환 등의 규칙을 이용하고 숙어 등의 사전정보를 활용하는 방법이 적용되고 있으나 한계가 있다. 본 논문에서는 두 언어간의 차이를 해소하는 방안으로 전처리를 적용하였으며 규칙기반의 영한기계번역에서 요구되는 전처리작업에 대해서 연구하였다. 전처리작업은 전처리문제와 해결방안으로 구성되는데, 언어간의 차이해소에 필요한 전처리문제를 조사하여 전처리문제가 영한기계번역의 어떤 단계에서 다루어져야 할지에 의해 문제들을 구분하였으며 이를 해결하기 위한 방안을 고안하여 본 논문에서 제시하였다.

1. 서 론

최근의 영한기계번역시스템은 짧은 문장에 대해서는 비교적 좋은 번역을 생성하지만 긴 문장이나 특수한 패턴을 포함한 문장들은 정확한 번역을 하지 못한다. 규칙기반의 영한기계번역 시스템은 특수한 패턴들의 올바른 분석 및 번역을 위해 어휘(lexical), 구문(syntactic), 변환(transfer) 등의 기계번역 과정에서 활용되는 규칙과 숙어 등의 사전정보를 이용하였으나 규칙의 수가 증가하고 숙어인식의 어려움과 숙어인식 과정에서 수반되는 부작용으로 인해 특수한 패턴에 대한 효과적인 해결방법이 되지 못한다. 또한 영한번역 대상이 되는 실제 문장들은 괄호, 인용부호, 리스트 표식 등 단어가 아닌 요소를 포함하기도 하는데 이들 요소들은 구문분석을 어렵게 하기 때문에 번역과정 이전에 적절하게 처리되어야 한다.

본 논문에서는 위에서 언급한 영어 문장이 가지는 특수한 패턴과 단어가 아닌 요소 등 영어와 한국어 사이의 차이들에 의한 번역의 어려움을 해결하기 위한 방안으로 전처리(preprocessing)에 대해 연구하였다. 본 논문에서의 전처리 대상은 HTML 같은 형식화된 문서가 아닌 일반 문서에 있는 영어 문장이다. 그리고 목적하는 영한기계번역시스템은 어휘, 구문, 변환 규칙을 가지는 규칙기반의 시스템으로서 언어간의 차이해소를 위해 숙어번역(idiom translation) 방식[1]을 적용하며 긴 문장의 효율적인 번역을 위해 문장분할(sentence segmentation)[2, 3] 방법을 적용하고 있는 시스템이다.

본 연구에서는 영한기계번역의 모든 과정, 즉 어휘분석(lexical analysis), 문장분할, 구문분석(parsing), 변환(transfer) 등의 과정에서 전처리를 필요로 하는 문제들을 조사하고 각각의 전처리 문제에 대한 해결방안을 고안하였다. 이를 통해 전처리 문제와 그에 대한 해결방안으로 구성되는 전처리작업(preprocessing task)을 정의하였다. 그리고 전처리작업을 수행하는 전처리 모듈과 기존의 영한기계번역시스템의 통합을 용이하게 하기 위해 전처리작업을 수행되어야할 시기에 따라 분류하였다. 또한 전처리에 의해 수반되는 후처리작업(post-processing)을 연구하여 제시함으로써 전처리에 의한 효과적인 언어차이 해소를 가능하도록 하였다.

2장에서 본 논문에서 조사한 전처리작업을 제시하고 분류하였다. 3장에서는 전처리에 수반되는 후처리작업을 제시하였으며 4장에서 앞으로의 할일 등을 제시하며 논문을 결론짓는다.

2. 전처리작업 (Preprocessing Tasks)

본 논문에서 전처리를 적용하려고 하는 대상이 되는 영한기계번역시스템은 그림 1의 구조를 가진다. 어휘분석(lexical analysis)을 통해 입력 문장의 각 단어에 대한 품사 및 기타 정보를 추출하고 문장분할(sentence segmentation) 단계에서 긴 문장을

분할하고 각 분할은 부분파싱(partial parsing) 단계에서 분석되어 전체구조 합성(global structure construction) 단계에서 하나의 파싱트리를 생성한다. 이후 변환(transfer) 단계에서는 변환규칙에 의해 한국어 생성(generation)에 적합한 트리로 변환이 이루어진다.

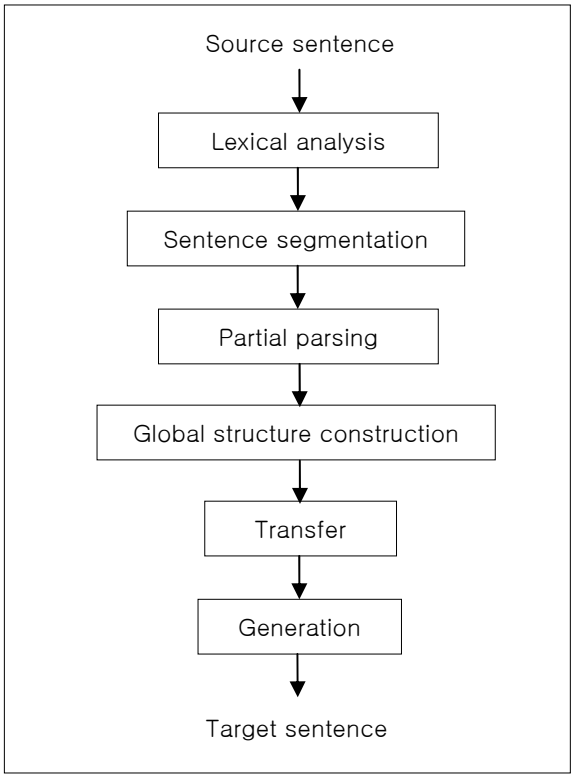

그림 1. 영한기계번역시스템의 논리적구조.

본 절에서는 영어와 한국어의 차이를 효과적으로 다루기 위해 필요한 전처리문제와 해결방안을 단계별로 분류하여 제시한다.

3.1 어휘분석 이전에 필요한 전처리작업

괄호, 인용부호, 하이픈(hyphen), 리스트 표시, 세미콜론, 콜론 등 단어가 아닌 요소를 포함한 문장이 있으며 단어가 아닌 요소는 문장을 여러 개의 번역단위(translation unit: TU)로 분할하는 역할을 한다. 따라서 번역단위 별로 나누어 번역을 수행하지 않으면 정확한 번역을 얻을 수 없다. 그리고 단위와 숫자가 함께 사용되거나 합성어, 번역을 할 때 번역하지 않아도 되는 의미없는 단어를 문장이 포함할 수도 있다. 또한 어휘분석 이전에 발견할 수 있는 영어에 고유한 특별한 패턴을 포함한 문장도 가능하다. 아래에 어휘분석 이전에 전처리를 필요로 하는 7가지 문제를 제시하였다.

첫째, 세미콜론, 콜론, 바(bar) 등이 하나의 문장의 복수개의 번역단위로 분할하는 경우이다. 그림 2의 예는 콜론에 의해 하나의 문장이 2개의 번역단위로 나뉘는 모습을 볼 수 있다. 하나의 복수의 번역단위를 포함하는 경우에 대처하기 위해 그림 1의 영한번역 과정을 둘러싸서 제어하는 상위 모듈이 필요하다.

The issue is in two parts : 200 million Swiss francs of privately placed notes, 100 million Swiss francs of publicly listed bonds.

TU1: The issue is in two parts
TU2: 200 million Swiss francs of privately placed notes, 100 million Swiss francs of publicly listed bonds.

그림 2. 복수의 번역단위를 가지는 문장.

둘째, 문장에서 인용부호로 둘러싸인 부분은 독립적인 번역단위가 될 수 있다. 그러나 첫번째 경우와는 다르게 인용부호로 둘러싸인 번역단위는 개별적으로 번역되지만 원래의 문장은 인용부호로 둘러싼 부분을 포함한 상태로 번역되어야 한다. 그림 3에 예를 제시하였는데, TU2는 개별적으로 번역될 수 있지만, TU1은 TU2에 해당하는 부분을 Q1이라는 것으로 표시하여 포함한 상태로 번역되어야한다. 따라서 TU2의 번역된 결과가 후처리(post-processing)를 통해 TU1의 번역된 결과의 Q1에 해당하는 부분에 대치되어야 한다.

"Next year, I may evaluate it a little closer," said Stan Guest, an uninsured farmer in East Nantmeal Township, Pa.

TU1: Q1, said Stan Guest, an uninsured farmer in East Nantmeal Township, Pa.
TU2 = Q1: Next year, I may evaluate it a little closer

그림 3. 인용부호의 의한 번역단위를 포함하는 문장.

셋째, 리스트의 나열을 나타내기 위한 표식(mark)을 포함하는 문장이 존재한다. 표식이 기호라면 쉽게 제거될 수 있으나 숫자 또는 알파벳 숫자(예를 들어, i, ii, iii, I, II, 등)이라면 인식과 제거과정이 필요하다.

넷째, 합성어(composition words)를 어휘분석 이전에 인식하여 하나의 단어로 처리할 수 있도록 해야 한다. 두 단어 이상이 모여 명사, 동사, 부사, 전치사 또는 접속사의 역할을 하는 경우들이 있다. 합성어를 어휘분석 이전에 인식하기 위해 합성어와 그 의미 및 품사를 포함하는 합성어 사전이 필요하다.

다섯째, 영어에는 한국어로 번역할 때 의미전달에 영향을 미치지 않거나 의미번역이 매우 모호한 단어들이 있다. 예를 들어, “ever”은 문장의 내용을 강조하기 위해 사용되는 단어인데, 이에 대한 적절한

한국어 대역어가 없으며 이를 특별하게 번역하지 않아도 문장 의미전달에 문제가 없으므로 번역 전에 제거되어도 된다. 이러한 종류의 단어들을 모아서 어휘분석 이전에 제거할 필요가 있다.

여섯째, 영어 문장은 문법으로 분석하기 어려운 특수한 문장 패턴이 존재한다. 이들 중 많은 것들은 숙어처리 방식을 적용하여 번역하지만 어떤 것들은 어휘분석 이전에 인식하여 처리하는 것이 보다 효과적일 수 있다. 예를 들어, [~ so that ~] 패턴을 포함하는 문장은 [~, so, ~]와 같은 의미를 가지지만 후자의 경우는 전자에 비해 문법으로 처리하는 것이 용이하다. 이러한 패턴들을 수집하고 적절한 패턴으로 문장 다시쓰기(sentence rewriting)를 수행하는 전처리가 어휘분석 이전에 필요하다.

일곱째, 날짜나 지명 등을 나타내는 영어의 구(phrases)들이 존재하며 일반적으로 쉼표를 포함하고 있다. 이러한 구들을 파싱 과정에서 구조를 파악하고 변환사전에서 한국어 대역어를 얻어서 번역하는 것은 파싱의 어려움 뿐만 아니라 적절하게 번역하는 것도 매우 어렵다. 따라서 이러한 구들을 어휘분석 이전에 인식하여 대응하는 한국어 번역문으로 번역하고 하나의 단어로 간주하여 이후의 번역과정에서 다루어야 파싱이 간단해지고 정확한 번역을 얻을 수 있다.

3.2 어휘분석 이후에 필요한 전처리작업

단어의 품사, 품사 확률 등의 어휘분석 결과를 필요로 하는 전처리작업이 있으며 따라서 이들은 어휘분석 이후에 수행되어야 한다. 여기서는 4가지의 전처리작업을 제시하였다.

첫째, 사람 이름과 나이를 표현하는 구를 포함하는 문장이 존재한다. 예를 들어, “Vincent, 32, was very smart”라는 문장에서 “Vincent, 32”는 “32살인 Vincent”로 번역되어야 하는데 구문 규칙이나 숙어로 처리하기가 어렵다. 이를 패턴으로 정의하고 대응하는 번역 패턴에 의해 대응 번역을 생성해야 한다. 전처리과정에서는 [사람+나이] 패턴을 찾아 하나의 단위로 만들고 대응 패턴에 의한 번역을 위한 후처리과정이 필요하다.

둘째, 영어에서 지명을 나타내는 특정한 패턴이 존재한다. 예를 들어, “I lived in Brynmawr, PA.”라는 문장에서 “Brynmawr, PA.”은 하나의 지명을 나타내는데, 이를 인식하기 위해서는 쉼표 앞, 뒤 단어가 고유명사여야 하며 따라서 품사정보를 필요로 한다. 지명 패턴도 위의 경우와 마찬가지로 하나의 단위로 간주되어야 하지만 특별한 후처리는 필요없다.

셋째, 3.1절에서 언급하였듯이 문법 및 숙어번역 방식으로 처리하기 어려운 패턴들이 존재한다. 그 중 일부는 어휘분석 이후에 처리될 수 있다. 예를 들어, [not only ~ but (also) ~], [no sooner had ~ than ~] 등의 패턴에서 ‘~’ 부분에 대한 일치 검사를 위해 어휘분석 정보가 필요하다. 패턴이 인식되는 경우 적절한 대응 패턴으로 다시쓰기가 수행되어야 하는데, [no sooner had ~ than ~]은 [as soon as ~, ~]로 다시 쓰여지면 분석이 보다 용이하게 수행될 수 있다.

넷째, 쉼표 다시쓰기(comma rewriting)이 필요한 경우가 있다. 예를 들어, “I need small, fast computer”라는 문장은 “I need small and fast computer”와 같은 의미이다. 그러나 쉼표를 포함하여 번역과정이 수행되면 문장분할에서 문장 구성요소(constituents)가 아닌 분할이 생성될 수 있고 이로 인해 잘못된 분석결과가 생성될 수 있다. 쉼표 다시쓰기를 위해 쉼표의 용도에 대한 정보가 필요하다[4].

3.3 문장분할 이후에 필요한 전처리작업

문장분할은 쉼표를 가지는 문장들을 쉼표에 의해 분할한다. 여기서는 문장분할 이후에 수행될 수 있는 전처리작업으로 아래 3가지를 제시하였다.

첫째, 영어에는 문장의 의미를 강조하기 위해 부가의문문이라는 것이 존재하는데 이는 도치 형식(inverted form)으로 표현되기에 문법으로 기술하기 어렵다. 따라서 구문분석 이전에 인식되어 대응하는 번역으로 변환되어야 한다. 부가 의문문은 [, VERB (not) Subject-word ?] 패턴을 가지므로 문장분할 이후에 마지막 분할에 대해 패턴 매칭(pattern matching)을 시도하여 인식할 수 있다. 부가 의문문으로 인식된 경우, 분할 리스트에서 분리되어 대응 번역으로 변환되고, 문장 번역이 끝난 후 번역 결과에 추가된다.

둘째, [so ~ that ~], [it BE-verb ~ ADJ that ~], [it BE-verb ~ that~] 등의 패턴들은 문법으로 기술하기도 어렵고 숙어로 처리하는 것도 인식의 부작용(side-effect)로 인해 적절하지 않다. 또한 이들 패턴을 포함한 문장은 그림 1의 “Partial parsing” 단계에서 하나의 파싱 단위가 되기 보다는 문장분할에 의해 분할될[1] 경우가 더 많은데 분할되어 따로 분석된다면 패턴에 의한 의미를 전달하기가 어려워진다. 따라서 쉼표에 의한 분할 이후에 생성된 분할에 대해 패턴 인식이 되어야 한다. 이들 패턴의 경우 분할 위치와 파싱 결과를 결합하는 방식에 대한 정보를 포함하도록 하여 패턴 인식 후 분할하여 각 분할을 독립적으로 파싱한 후 결과를 합성하도록 한다.

셋째, “say”류의 동사들은 몇몇 패턴을 형성한다. 이들 동사가 인용부호로 둘러싸인 목적어를 가지면, 목적어는 쉼표로 분리되고 주어-동사 어순이 도치될 수 있어 문법규칙으로 다루기 어렵다. 이 경우 문장분할

[1] 쉼표에 의한 문장분할 이후 일정 길이 이상의 분할에 대해서는 추가적인 분할이 수행되기 때문이다.

위치조정(segments repositioning)이 필요한데 그림 4에 예를 제시하였다. 우선 번역단위가 인식된 후 구문 규칙으로 기술될 수 있는 순서로 문장분할들의 위치를 조정한다. 위치조정 과정에서 각 문장분할의 역할에 대한 정보를 얻게 되며 이는 “Partial parsing” 후 전체구조 합성단계에서 이용된다.

"Next year, I may evaluate it a little closer," said Stan Guest, an uninsured farmer in East Nantmeal Township, Pa.

⇨ Q1, said Stan Guest, an uninsured farmer in East Nantmeal Township, Pa.

⇨ Stan Guest, an uninsured farmer in East Nantmeal Township, Pa., said, Q1.

그림 4. 문장분할 위치조정의 예.

3. 후처리작업 (Post-processing Tasks)

2장에서 제시한 전처리작업 중 일부는 대응하는 후처리작업을 필요로 하며 여기서는 5가지로 분류하였다.

첫째, 단어가 아닌 요소에 의해 분리된 번역단위의 번역결과는 번역단위의 순서에 맞게 결합되어야 한다. 이러한 후처리작업은 그림 1의 “Generation” 단계 후에 수행된다.

둘째, 인용부호나 괄호로 둘러싸인 부분이 번역된 후 번역결과는 원래 문장의 번역결과에 포함되어야 한다. 전처리작업에서 확인된 인용부호나 괄호와 연관된 단어의 번역결과와 결합되어야 하므로 목적 단어의 위치를 확인할 수 있는 파싱 트리(parsing tree) 상에서 결합되어야 한다. 따라서 이러한 후처리작업은 그림 1의 “Transfer” 단계 후에 수행되어야 한다.

셋째, 이름-나이 패턴과 날짜 패턴으로 구성되는 구는 함께 번역패턴(translation patterns)을 이용하여 번역한다. 그림 5에 번역패턴에 의한 번역방법을 제시하였다. 이러한 후처리작업은 그림 1의 “Transfer” 단계에서 수행한다.

[Month NUM1, NUM2] → [NUM2년 Month월 Num1일]: January 1, 1998
[NUM1 Month, NUM2] → [NUM2년 Month월 Num1일]: 15 January, 1998

[PRONOUN, NUM] → [NUM살인 PRONOUN]
***Antonio L. Savoca , 66** , was named president*

그림 5. 날짜와 이름-나이 패턴에 대한 번역패턴의 예.

넷째, 3.3절에서 소개한 특수한 패턴을 가지는 문장분할은 패턴이 제공하는 분할정보에 의해 추가 분할되어 분선된다. 그리고 패턴에 기술된 결합 규칙(combination rules)에 의해 분할의 파싱 결과들을 결합하여 하나의 구조를 생성한다. 따라서 이러한 후처리작업은 그림 1의 “Partial parsing” 단계 후에 수행된다. 그림 6은 세 부분으로 구성되는 특수 패턴의 예를 보여준다. 첫번째 부분은 문장 패턴이고 두번째 부분은 분할 방식을 알려주며, 마지막 부분은 파싱결과를 결합하는 방식을 의미한다. 예에서 ‘+’ 기호는 두 트리를 결합하라는 의미이고 ‘1_SUBJ_2’는 첫번째 트리의 주어(subject)가 두번째 트리라는 의미이다. 패턴에서 처음 두 부분은 전처리작업을 위한 것이며 세번째 부분은 후처리작업에서 이용된다.

[~ so A that B], ([~ so A] , [and B]) , +;

[it BE-verb A ADJ that B], ([it BE-verb A ADJ], [B]), 1_SUBJ_2;

[it BE-verb A that B], ([it BE-verb A], [B]), 1_SUBJ_2;

그림 6. 문장분할 내의 특수한 패턴의 예.

다섯째, 전처리작업에서 문장분할 위치조정을 수행할 때, 각 분할의 역할을 확인한다. 후처리작업에서는 파악된 역할에 따라 각 분할의 파싱 결과를 결합한다. 그림 7은 그림 4에 대한 예가 나타나는데, 여기서는 쉼표에 의해 분리된 문장분할들의 역할에 대한 정보를 보여준다.

"Next year, I may evaluate it a little closer," said Stan Guest, an uninsured farmer in East Nantmeal Township, Pa.

⇨ Stan Guest, an uninsured farmer in East Nantmeal Township, Pa., said, Q1.

⇨ SEG1: SUBJECT, SEG2: APPOSITIVE, SEG3: VERB, SEG4: OBJECT

그림 7. 위치조정된 문장분할의 역할 정보의 예.

그림 8은 2, 3장에서 제시한 전처리 및 후처리작업이 결합된 영한기계번역 시스템의 논리적 구조를 보여준다. 입력 문장은 복수개의 번역단위를 가질 수 있으며 각 번역단위에 대해 “lexical analysis”부터 “generation”까지의 번역사이클(translation cycle)이 수행된다. 따라서 “Pre-lexical preprocessing”과 마지막 “post-processing”은 입력 문장에 대해 단 한번만 수행된다.

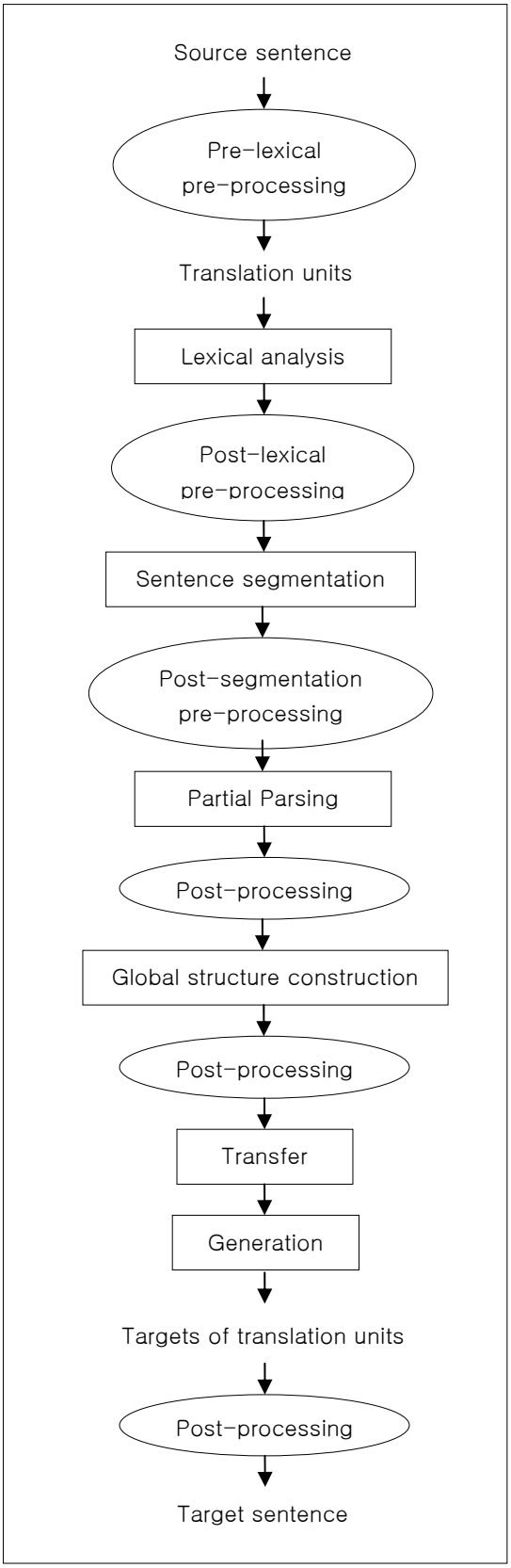

그림 8. 전처리, 후처리작업이 통합된 영한기계번역 시스템의 논리적 구조.

4. 결론

본 논문에서는 영어와 한국어간의 차이를 해소하기 위한 방안으로 전처리를 적용하였으며 필요한 전처리작업과 이에 수반되는 후처리작업을 조사하여 분류하였다. 영한기계번역의 여러 단계에서 수행되는 전처리와 후처리작업에 의해 문법이나 사전정보에 의해 적절하게 해결하지 못하는 어려운 문제를 효과적으로 다룰 수 있다. 다양한 전처리문제에 대한 해결방안으로 문장분리(sentence split), 기호 및 단어제거, 다시쓰기, 문장분할 리스트로부터 특정분할의 분리, 문장분할 위치조정 등이 제시되었다. 그리고 복합어 사전, 문장패턴, 번역패턴 등의 정보 구축이 필요하다고 제안하였다.

일부 전처리작업은 이미 개발되었으며 일부는 연구중이다. 효과적인 정보 구축과 효과적인 활용을 위해 패턴과 패턴에 부속되는 정보에 대한 간단하고 일관성 있는 표현 방법을 고안할 필요가 있다. 어떤 방법은 부작용을 유발할 수도 있으므로 이를 피할 수 있는 방법도 연구해야 할 것이다. 추가적으로 제안된 전처리, 후처리작업이 통합된 영한기계번역 시스템으로 번역 테스트를 수행하여 번역품질의 개선을 측정하여 제안된 방법의 유용성을 검증해야 한다.

참고문헌

[1] 윤성희. 영한기계번역에서 숙어기반의 효율적인 파싱. 서울대학교 컴퓨터공학과 박사학위논문. 1993.

[2] S.-D. Kim, B.-T. Zhang, Y. T. Kim, Reducing Parsing Complexity by Intra-Sentence Segmentation based on Maximum Entropy Model, Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora, pp. 164-171, 2000.

[3] S.-D. Kim, B.-T. Zhang, Y. T. Kim, Learning-based Intrasentence Segmentation for Efficient Translation of Long Sentences. *Machine Translation*, Vol. 16, No. 2, pp. 151-174, 2001.

[4] 김성동, 박성훈, 영한 기계번역에서 긴 문장의 구문분석 정확성 향상을 위한 쉼표의 용도 분류, 한국정보과학회 추계학술대회, 2006.