영한 기계번역 시스템의 개선을 지원하는 영어 구문 규칙 관리 도구

김성동[O], 김창희, 김태완
한성대학교, 컴퓨터공학과
sdkim@hansung.ac.kr, kimch3617@naver.com, madplay@naver.com

English Syntactic Rule Management Tool for Improving English-Korean Machine Translation System

Sung-Dong Kim[O], Chang-Hee Kim, Tae-Wan Kim
Hansung University, Dept. of Computer Engineering

요 약

규칙 기반의 영한 기계번역을 위해서는 많은 영어 구문 규칙을 구축하고 관리해야 하는데, 이는 매우 많은 노력과 시간을 요구한다. 이 문제에 대한 해결방안으로 본 논문에서는 영어 구문 규칙의 효율적인 관리를 도와주는 도구를 제안한다. 영한 기계번역 시스템의 성능 개선 과정에서 영어 구문 규칙의 검색과 수정이 빈번하게 이루어지는데, 이러한 작업을 쉽게 할 수 있도록 제안하는 도구는 다양한 키를 이용한 규칙 검색과 규칙 수정 기능을 제공한다. 제안하는 도구는 영어 구문 규칙을 관리하는데 필요한 사람의 노력을 줄여 지속적인 영한 기계번역 시스템의 성능 개선 과정을 보다 손쉽게 할 수 있게 할 것이다.

주제어: 영한 기계번역, 규칙기반 기계번역, 구문 분석, 구문 규칙

1. 서론

규칙 기반 기계번역(rule-based machine translation: RBMT) 방법은 기계번역 방법 중 가장 오래된 방법으로서 언어학 기반(linguistic-based) 방법이라고도 불린다. 규칙 기반 기계번역 시스템을 개발하기 위해서는 많은 규칙을 전문가가 직접 구축해야 하며 이는 많은 시간과 노력을 필요로 한다. 그리고 번역 품질 개선 과정이나 새로운 영역에 적응시키는 작업에는 기존에 구축한 규칙을 지속적으로 개선하는 작업이 요구되는 지식 획득의 병목현상(knowledge acquisition bottleneck) 문제를 가진다. 다른 방법으로 1990년대 이후로 대량의 말뭉치가 구축됨에 따라 이국어 병렬 말뭉치(bilingual parallel corpus)를 이용한 통계적 기계번역 시스템에 대한 연구가 활발하게 진행되었다. 이 방법은 기계번역 시스템을 개발하고 유지하는 비용이 규칙 기반 방법에 비해 적다는 장점을 가지는데 비해 언어 구조나 문장 스타일 등을 적절하게 반영하여 번역하지 못하며 학습 과정이나 번역 에러를 조정하는 것이 어렵고 매우 많은 말뭉치 데이터를 필요로 한다는 단점을 가진다. 이에 비해 규칙 기반 방법은 전문가의 지식을 활용하여 보다 나은 번역을 생성할 수 있으며 예상되는 번역을 생성하기 때문에 에러를 수정하는 것이 통계적 방법에 비해 용이하다. 영한 기계번역이 다루는 두 언어는 매우 상이한 언어이므로 규칙 기반 방법을 이용하여 번역 시스템을 구축하고 이를 보강하는 측면에서 통계적 방법을 적용하는 것이 바람직하다는 판단이다. 이러한 하이브리드 방식의 기계번역 시스템으로는 SYSTRAN의 기계번역 시스템[1], Safaba의 EMTGlobal[2], PROMT의 기계번역 솔루션[3] 등이 있으며 실용화되고 있는 상황이다.

본 논문에서는 차트(chart) 기반의 구문 분석 방법[4]이 적용된 영한 기계번역 시스템의 효율적인 성능 개선 작업을 지원하기 위한 영어 구문 규칙 관리 도구를 제안한다. 규칙 기반의 영한 기계번역 시스템은 많은 영어 구문 규칙을 가지며 각 규칙마다 적절성 점수를 가지고 있어 파싱 트리를 생성하면서 결과 트리에 대한 적절성 점수를 계산한다. 문장 번역 테스트를 통해 번역 품질을 향상시키는 과정에서 영어 구문 규칙을 검색하여 규칙을 보완하고 수정하여 개선하는 과정이 필수적으로 요구되는데, 제안하는 도구는 이러한 과정을 손쉽게 수행할 수 있도록 지원하여 영한 기계번역 시스템의 성능 개선 작업을 도와주는 것을 목적으로 한다.

본 논문은 다음과 같이 구성된다. 2장에서는 기계번역 시스템의 개선 과정에서 사용될 구문 규칙 관리 도구가 갖추어야 할 요구 사항을 분석하고 3장에서는 이러한 요구사항을 수용하는 구문 규칙 관리 도구의 기능을 설명한다. 그리고 4장에서는 논문에서 제안하는 구문 규칙 관리 도구의 효과에 대해 설명하고 앞으로의 과제를 제시하며 논문을 마무리한다.

2. 구문 규칙 관리 도구의 요구 사항

문장 번역 테스트를 통한 영한 기계번역 시스템의 번역 품질 개선을 위한 구문 규칙 개선 작업 과정 다음과 같다. 첫째, 테스트 문장들을 일괄 번역(batch

translation)한 후에 번역 결과를 분석하여 잘못된 번역을 생성한 문장을 수집한다. 둘째, 이들 각각의 문장에 대해 개별 번역(interactive translation)을 수행하며, 이 때 구문 분석 결과인 차트와 파싱 트리(parsing tree)를 출력하면서 적용된 구문 규칙을 함께 출력하도록 한다. 셋째, 이 결과물을 이용하여 번역 오류가 잘못된 파싱 트리의 선택에 의한 것이었는지를 확인한다. 만약 번역 오류의 원인이 이 경우에 해당한다면, 올바른 구문 분석 트리를 생성하기 위한 규칙을 확인하여 이들 규칙이 적용된 구문 분석 트리를 찾고, 결과 트리와 비교하여 잘못된 트리가 선택된 원인을 파악한다. 이 과정에서 구문 규칙을 검색하여 분석하는 작업이 필요한데, 제안하는 도구는 규칙을 검색하여 그 내용을 쉽게 확인할 수 있는 기능을 제공해야 한다. 넷째, 올바른 트리를 생성하기 위한 구문 규칙과 결과 트리 생성에 적용된 규칙을 비교하고, 올바른 트리가 생성될 수 있도록 이들 규칙을 수정한다. 따라서 제안하는 도구는 규칙의 수정 기능을 제공해야 한다. 그리고 규칙이 수정되면 구문 분석기의 규칙 적용 모듈이 수정된 규칙에 맞게 수정되어야 하는데, 제안하는 도구는 규칙 적용 모듈의 수정 기능 역시 제공해야 한다. 그림 1은 영한 기계번역 시스템의 구문 규칙의 예를 보여준다.

```
NP010: NP(%NOUN, cat='NP, OBJP=1, SUBJP=1,
   RefIndicToScore NOUN POSSESIVE 1 0.5,
   score=1.02)
   <- NOUN(COORDED=0)
```

그림 1. 구문 규칙의 예

구문 규칙은 [규칙 이름: 규칙의 좌측(left-hand side) <- 규칙의 우측(right-hand side)]의 형식을 가진다. 규칙의 좌측과 우측은 [문법 기호(행동/조건)]의 형식을 가지는데, "문법 기호"는 NP, VP 등의 구를 나타내는 기호와 NOUN, ADJ 등의 품사를 나타내는 기호를 포함한다. 영한 기계번역 시스템의 구문 분석 모듈은 구문 규칙의 규칙 이름과 우측 기호들과 관련된 C 언어 함수로 구성된다. 예를 들어, 그림 1의 구문 규칙과 관련된 C 언어 함수는 NP010()과 NP010noun() 함수가 있다.

결과적으로 제안하는 구문 규칙 관리 도구는 다음과 같은 기능을 필요로 한다. 첫째, 구문 규칙 검색 기능이다. 구문 규칙의 검색을 쉽게 하기 위해 규칙의 좌측 기호, 우측 기호들, 규칙 이름 등을 키(key)로 하는 검색이 필요하다. 둘째, 구문 규칙의 비교를 위해 규칙의 모든 내용을 보기 좋게 보여줄 뿐만 아니라 여러 개의 규칙을 동시에 보여줄 수 있어야 한다. 이와 함께 규칙에 해당하는 구문 분석 모듈을 보여줄 필요가 있다. 잘못된 구문 분석의 원인을 파악할 때, 규칙을 분석하기 보다는 구문 분석 모듈, 즉 C 언어 함수를 분석하는 것이 더 용이할 수 있기 때문이다. 셋째, 구문 규칙과 해당하는 C 언어 함수의 수정 기능이 필요하다.

3. 구문 규칙 관리 도구의 기능

2장에서 제시한 구문 규칙 관리 도구의 요구사항을 만족시키기 위해서 제안하는 도구는 다음과 같은 기능을 제공한다: 규칙 검색 기능, 규칙과 대응하는 구문 분석 모듈 보여주기 기능, 규칙 수정과 대응하는 구문 규칙 관리 모듈 수정 기능.

본 장에서는 구문 규칙 관리 도구의 기능을 설명함에 있어 사용자 인터페이스의 모습을 제시하고 도구를 사용하는 방법을 중심으로 설명한다.

3.1 구문 규칙 검색 기능

적절하지 않은 구문 분석 트리 선택에 의해 잘못된 번역이 생성되었을 경우, 결과로 선택된 구문 분석 트리를 생성하는데 적용된 구문 규칙을 확인해야 한다. 올바른 번역 생성을 위한 트리가 선택되지 않았던 이유는 경쟁하는 구문 규칙들 중에서 적절한 구문 규칙의 점수가 선택된 구문 규칙의 점수보다 낮거나 적절한 구문 규칙을 적용하기 위한 조건이 맞지 않았기 때문이다. 따라서 경쟁하는 구문 규칙들을 모두 확인하여 규칙의 조건부와 점수를 비교, 분석해야 한다. 그림 2는 구문 규칙 검색 화면의 모습이다.

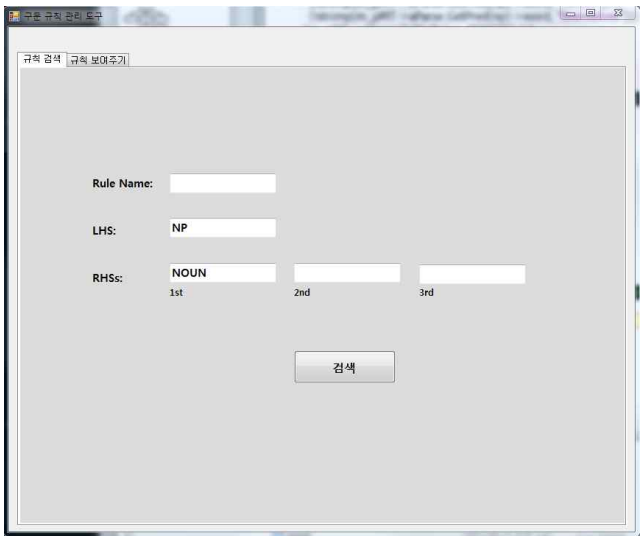

그림 2. 구문 규칙 검색 화면

경쟁하는 구문 규칙을 쉽게 확인하기 위해서 구문 규칙 검색 기능은 2장에서 제시한 것처럼 규칙 이름, 규칙의 좌측 기호(LHS), 우측 기호(RHS)들을 키로 하여 검색할 수 있다. 원하는 키를 입력하고 "검색" 버튼을 누르면 검색 키에 맞는 구문 규칙을 검색하고 결과는 "규칙 보여주기" 탭에서 보여준다. 우측 기호는 하나 이상을 키로 할 수 있으며, 좌측 기호와 우측 기호를 조합하여

키를 구성하여 검색할 수도 있다. 그림 3은 LHS가 NP이고 RHS의 첫 번째 기호가 NOUN인 구문 규칙을 검색하기 위해 키를 설정한 모습을 나타낸다.

3.2 구문 규칙과 구문 분석 모듈 보여주기 기능

검색한 구문 규칙을 보여줄 때, 대응하는 구문 분석 모듈의 C 언어 함수도 함께 보여준다. 그림 3은 규칙 검색 결과를 보여주는 화면의 모습이다.

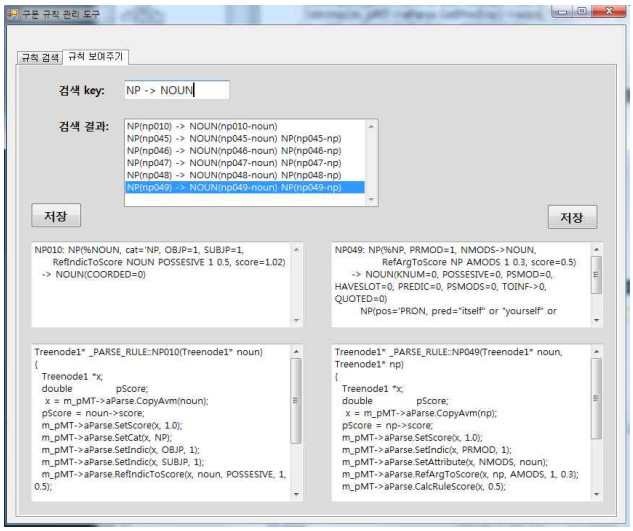

그림 3. 구문 규칙 보여주기 화면

그림에서 "검색 key" 부분은 검색에 사용된 key가 무엇인지를 확인할 수 있게 하며, "검색 결과" 부분은 검색 key와 일치한 구문 규칙들의 목록을 보여준다. 특정한 규칙을 선택하여 클릭하면 먼저 좌측에 구문 규칙과 해당하는 C 언어 함수를 보여주고 다른 규칙을 선택하여 클릭하면 오른편에 보여줌으로써 규칙의 비교를 용이하게 한다. 그림 4는 먼저 "np010" 규칙을 선택하여 좌측에 그 내용이 나타난 후, "np049" 규칙을 선택하여 우측에 내용이 나타난 모습을 보여준다.

3.3 구문 규칙과 구문 분석 모듈 수정 기능

구문 규칙들을 분석하여 잘못된 트리를 선택한 원인을 파악하였다면 구문 분석과 대응하는 C 언어 함수를 수정해야 한다. 원칙적으로는 규칙을 수정하면 대응하는 C 언어 함수를 직접 C 언어 함수를 수정한다. 그림 3에서 제시되는 C 언어 함수를 수정하면 영한 기계번역 시스템의 구문 분석 모듈이 수정되며, 수정된 번역 시스템으로 번역을 수행하여 수정에 의해 올바른 트리를 선택하여 올바른 번역이 생성되는지를 확인하는 과정으로 구문 규칙을 개선하는 작업을 수행한다.

그림 3에서 구문 규칙이나 C 언어 함수를 수정한 후 "저장" 버튼을 누르면 시스템의 규칙과 C 언어 함수가 수정 된다. 화면에 2개의 "저장" 버튼이 있는데 각각 해당하는 위치의 규칙과 함수를 저장한다. 그리고 수정에 대한 기록은 로그 파일에 저장되어 구문 규칙과 모듈에 대한 수정 내용을 나중에 확인할 수 있다.

4. 결론

본 논문에서는 규칙 기반의 영한 기계번역 시스템의 구문 규칙 개선을 통한 지속적인 성능 개선을 지원하기 위한 구문 규칙 관리 도구를 제안하였다. 제안하는 도구는 번역 테스트 과정에서 구문 규칙의 잘못된 적용으로 인해 잘못된 번역이 생성된 경우에 그 원인을 파악하여 올바른 번역을 생성할 수 있도록 기존의 구문 규칙을 수정하는 작업을 쉽게 할 수 있도록 지원한다. 즉 적용된 구문 규칙을 쉽게 검색할 수 있도록 다양한 키를 이용한 검색 기능을 제공하고 구문 규칙과 대응하는 C 언어 함수를 동시에 보여줌으로써 문제가 되는 구문 규칙과 모듈을 쉽게 확인하고 수정할 수 있도록 해준다. 이를 통해 구문 규칙의 개선에 의한 번역 시스템의 성능 개선 과정을 쉽게 한다.

현재는 기존 규칙을 확인하고 수정하기만 할 수 있으나 필요하면 새로운 구문 규칙을 추가하거나 필요 없는 규칙을 제거할 경우를 지원할 수 있도록 구문 규칙 관리 도구가 개선되어야 할 것이다. 구문 규칙이 삭제되면 이에 해당하는 구문 분석 모듈의 C 언어 함수가 삭제되어야 하며 추가되면 C 언어 함수가 새로 생성되어야 한다. 따라서 구문 규칙을 자동으로 C 언어 함수로 변환하는 도구에 대한 연구가 필요하다.

참고문헌

[1] SYSTRAN Translator, SYSTRAN, www.systransoft.com
[2] EMTGlobal, Safaba Translation Innovation, www.safaba.com.
[3] PROMT translator, PROMT, www.promt.com
[4] Terry Winograd, Language a s a Cognitive process: Vol. 1, Syntax, Addison Wesley, 1983.