論文2000-37CI-5-1

효율적인 예제 기반 기계번역을 위한 패턴의 사용

(An Use of the Patterns for an Efficient Example-Based Machine Translation)

李 起 榮 *, 金 漢 宇 *

(Ki-Young Lee and Han-Woo Kim)

요 약

예제 기반 기계번역 기법은 기존의 규칙 기반 기계번역에서 발생되는 다양한 문제점들을 해결하기 위해 제안된 새로운 기계번역 패러다임이다. 하지만 기존의 순수 예제 기반 기계번역의 경우 적당한 크기의 병렬 코퍼스를 사용하여 입력문과 거의 유사한 예문을 발견하는데는 한계가 있으며, 이러한 점이 번역문 생성 단계에서 부담으로 작용하게 된다. 본 논문에서는 예제 기반 기계번역 기법의 문제점을 보완하기 위한 새로운 대안으로서 패턴과 예문을 함께 사용하여 영한 변환을 수행하는 새로운 영한 변환 기법을 제안한다. 패턴은 크게 문장 패턴과 구 패턴으로 구분되며, 패턴의 메타 부분은 유사 예문 발견 확률을 높여서 예제 기반 기계번역 기법을 보다 실용적으로 만들어준다. 실험 결과 기존의 표층 어휘 비교에 의한 순수 예제 기반 기계번역에 비해 비교적 적은 양의 예문을 가지고도 유사 예문 발견 확률이 높다는 것을 알 수 있었다.

Abstract

An example-based machine translation approach is a new paradigm for resolving various problems caused by the rules of conventional rule-based machine translation. But, in pure example-based machine translation, it is very hard to find similar examples matched with input sentences by using reasonable parallel corpus. This problem causes large overheads in the process of sentence generation. This paper proposes new method of English-Korean transfer using both patterns and examples. The patterns are composed of sentence patterns and phrase patterns. Meta parts of the patterns make the example-based machine translation more practical by raising the probability to find similar examples. The use of patterns and examples can reduce the ambiguities in source language analysis and give us a high quality of MT. And experimental results with a test corpus are discussed.

I. 서 론

전통적인 규칙 기반 기계번역 방법은 형태소 해석, 구문 해석, 의미 해석의 3단계 과정을 통해 번역을 수행한다. 이러한 규칙 기반 기계번역 방법은 각 단계에서 발생하는 모호성이 서로 결합하여, 전체적으로 보다 많은 모호성을 생성할 뿐만 아니라, 원시 언어 해석에 사용되는 문법 규칙의 수가 너무 많아서 이를 유지 및 관리하는데 많은 문제점을 드러내고 있다. 또한 규칙 기반의 기계번역 방법은 시스템의 도메인(domain)을 다른 도메인으로 변경 또는 확장할 경우, 새로 도입되

* 正會員, 漢陽大學校 電子計算學科

(Dept. of Computer Science & Engineering, Hanyang Univ.)

接受日字:1999年8月23日, 수정완료일:2000年5月3日

는 규칙과 기존 규칙간의 충돌을 세심하게 신경 써야 할뿐만 아니라, 전체 시스템을 개발하기까지 걸리는 시간과 노력이 막대하다는 단점을 지니고 있다. 이러한 이유로 인해 전통적인 규칙 기반 기계번역 방법과는 다른 다양한 기계번역 방법론들이 제시되었으며, 대표적인 것으로는 예제 기반 기계번역(example-based MT), 통계 기반 기계번역(statistics-based MT) 및 코퍼스 기반 기계번역(corpus-based MT) 등이 있다.

예제 기반 기계번역 방법은 Nagao[1]에서 제안된 새로운 기계번역 방법이다. 예제 기반 기계번역에서는 원시 언어 문장과 그에 대한 대역문의 쌍(pair)으로 구성된 대규모 번역 데이터베이스를 번역의 지식원(knowledge base)으로 사용한다. 즉, 예제 기반 기계번역 방법은 데이터베이스에 존재하는 가장 유사한 예문의 대역문을 번역의 모범(模範)으로 사용하여 번역을 수행한다. 그러나 실생활에서 일반적으로 사용하는 문장이든 제한된 도메인에서 사용되는 문장이든 그 문장의 길이가 예제 기반 기계번역 방법을 적용할 경우 효과적일 만큼 짧지는 않다. 유사 예문을 발견할 확률은 문장의 길이가 길어질수록 낮아진다고 할 때, 이러한 문제는 결국 입력문에서 예문과 일치하지 않는 부분에 대한 처리를 수행하기 위한 오버헤드(overhead)를 초래하며, 번역문 생성 과정을 매우 어렵게 만드는 요인이 된다.

본 논문에서는 실험을 통하여 순수한 예제 기반 기계번역 방법이 갖는 문제점들을 언급하고, 이를 해결하기 위한 새로운 기계번역 방법으로서 패턴과 예문을 함께 이용하는 번역 방법을 제안한다. 본 논문에서 제안하는 접근 방법은 모든 문장은 문법적 또는 의미적 중심 부분을 뼈대로 하여 기타 부가적인 어휘들로 구성되어 있다는 것이다. 본 논문에서는 기본적으로 이러한 문장의 문법적 또는 의미적 중심 부분을 패턴화하여 번역에 사용한다. 더 나아가서 특정 도메인에서 자주 사용되는 빈출 표현도 역시 패턴으로 간주한다. 즉, 기존의 규칙 기반 접근 방법에서 취하는 원시 언어 위주의 해석적 관점을 취하는 것이 아니라 번역 관점에서 원시 언어 표현과 이에 대응하는 목표 언어 표현을 번역 패턴으로 정의한다. 이러한 점에서, 제안되는 패턴은 기존의 규칙에 크게 의존하지 않게 되며, 이는 기존의 규칙 기반 기계번역 시스템에서 규칙으로 인해 발생되는 다양한 문제들, 예를 들어, 규칙 유지 및 관리의 어려움, 번역 시스템 확장의 어려움 등과 같은 문제들을 자연스럽게 해결해 줄 수 있으며, 문어체에 비해 비문법적인 표현을 많이 포함하고 있는 대화체 등에 적용할 경우, 많은 효과가 있을 것이다.

본 논문의 구성은 다음과 같다. 2장에서는 예제 기반 기계번역과 관련된 다양한 연구들을 소개하며, 3장에서는 문장 패턴 및 구 패턴에 대한 정의를 한다. 그리고 4장에서는 패턴을 사용하여 번역을 수행하는 전반적인 과정을 예문과 함께 보이며, 5장에서는 본 논문에서 제안하는 패턴과 예문을 사용하는 기계번역 방법에 대한 실험 결과에 대한 토의를 한다. 마지막으로 6장에서는 본 연구와 관련해서 향후 연구 과제를 제시하고 결론을 맺는다.

Ⅱ. 관련 연구

기계번역을 위해 예문을 사용하려는 접근 방법은 Nagao. M.[1]에서 처음으로 제안되었다. Nagao. M.[1]에서 제안된 번역 방법은 원문과 번역문의 쌍으로 구성되어 있는 대규모 병렬 코퍼스를 지식 베이스로 사용하여 입력문과 가장 유사한 예문을 찾아서 조정 과정을 거친 후, 번역문을 생성한다. 그러나 입력문의 길이가 길어질수록 문장 단위 비교에 의해서는 입력문과 한두 단어 차이가 나는 유사한 예문을 발견할 확률은 매우 떨어지게 된다. 결국, 입력문과 차이가 많이 나는 예문을 사용하여 번역문을 생성할 경우, 이는 번역문 생성 부분에 있어서 상당한 부담으로 작용하게 된다.

이를 해결하기 위해서 데이터베이스의 예문과 입력문과의 비교를 문장 단위가 아닌 문장의 일부, 즉, 청크(chunk)로 나누어서 비교하자는 방법이 제안되었다[2]. 이 방법은 입력문을 청크로 나누어서 각각의 청크를 독립적으로 번역하고, 이후에 각각의 번역된 청크들을 결합해서 최종 번역문을 생성하게 된다. 이 방법은 문법은 별로 고려하지 않고 주로 표층 단어의 일치 여부에만 의존하는 경향이 있기 때문에, 청크를 잘못 나눌 경우에는 번역의 품질에 매우 좋지 않은 영향을 끼치게 되는 단점이 있다.

佐藤 理史[3]에서는 예문 데이터베이스의 원문과 번역문을 의존 문법을 사용하여 구문 분석한 결과인 의존 트리로 구축하며, 원시 언어에 대한 구문 트리 노드와 목표 언어에 대한 구문 트리 노드간의 대응관계를 따로 나타내었다. 佐藤 理史[3]에서 제안하는 아이디어는 하나의 입력문을 번역하는데 있어서 하나 이상의 예문을 사용하자는 것이며, 이러한 번역 단위(translation

unit)의 분할은 기존의 문법에 의해서 구현된다. 즉, 이 번역 방법은 기존의 문법을 사용하여 문장을 몇몇 구 단위로 분할한 뒤, 각각의 구를 서로 다른 예문을 사용하여 번역하고, 최종적으로 하나의 문장으로 통합해서 번역문을 생성하게 된다. 이렇게 할 경우, S. Nirenburg, C. Domashnev, D. J. Grannes[2]에서 발생하는 문제점을 기존의 문법을 사용하여 해결할 수 있는 장점이 있다. 그러나 이 방법의 문제점은 첫째로 데이터베이스를 구축하는데, 너무 많은 비용이 든다는 것이고, 둘째로 기존의 규칙에 기반하여 원시 언어 및 목적 언어를 해석하기 때문에 그 모호성도 계속 존재하게 된다는 것이다.

또 다른 기계번역 방법으로 패턴을 번역에 사용하려는 시도가 古瀬 藏, 隅田 英一郎, 飯田 仁[4], 古瀬 藏, 飯田 仁[5], Osamu Furuse and Hitoshi IIDA[6], Koich Takeda[7] 등에서 제안되었다. 古瀬 藏, 隅田 英一郎, 飯田 仁[4], 古瀬 藏, 飯田 仁[5], Osamu Furuse and Hitoshi IIDA[6] 등에서는 특정 도메인 상에서 자주 사용되는 빈출 표현들을 패턴으로 정의하였으며, 메타 부분에 올 수 있는 연어의 리스트를 따로 만들어 두고 적당한 번역을 시도하였다. Koich Takeda[7]에서는 패턴 기반의 문맥자유문법을 설정하여 이를 기계번역에 사용하였다. Hideo Watanabe[8]에서는 번역 패턴을 추출하는 방법을 제안하였다. Hideo Watanabe[8]에서는 원문과 번역문을 의존문법을 사용하여 파스 트리 형태로 나타낸 뒤에 각 노드간의 대응 관계를 패턴화하였다. 하지만 Hideo Watanabe[8]도 역시 기존의 규칙 기반의 해석을 통해 패턴을 추출하므로 해석에서 발생하는 모호성으로 인한 문제는 피하기 어렵다고 할 수 있다.

우리 나라의 경우, 패턴을 사용하여 문장을 해석하고자 하는 연구로 이호석, 김영택[10], 서병락, 김영택[11], 김나리, 김영택[12], 송영빈, 최기선[13] 등이 있다.

위에서 언급한 대부분의 제안들이 공통으로 다루고 있는 것은 기존의 규칙 기반 기계번역 시스템들이 많은 양의 규칙을 사용하고 있으므로 그 유지 및 관리가 힘들며, 이후의 시스템 확장 또한 매우 어렵다는 것이다. 또한 기존의 규칙 기반의 기계번역 시스템들은 시스템 개발에 걸리는 시간이 기타 소프트웨어에 비해 특히 길어서, 상용 제품으로의 개발에 많은 문제점을 지니고 있다. 이러한 점은 佐藤 理史[9]에서도 규칙 기반 기계번역 시스템의 문제점으로 제기되었다.

Ⅲ. 패 턴

우리가 사용하는 문장들은 일반적으로 문법적·의미적 핵심 부분을 중심으로 하여 기타 수식어구들로 구성되어 있는 것으로 볼 수 있다. 본 논문에서는 이러한 문장의 중심 부분을 패턴화하여 문장의 구조 변환에 사용한다. 이때, 패턴을 구성하는 문법적·의미적 중심 부분을 고정 부분이라고 하고 나머지 부분을 메타 부분이라고 부른다. 하지만 기존의 문법 규칙을 사용하여 패턴을 정의하고 이를 그대로 기계번역에 적용하는 데에는 약간의 무리가 따른다. 왜냐하면 기존의 문법에 의존해서 패턴을 정의할 경우에는 이미 서론 부분에서 언급한 기존의 규칙 기반 시스템에서 발생하는 다양한 문제점들을 피할 수 없기 때문이다.

따라서 본 논문에서는 패턴을 통상적인 의미보다 확장된 개념을 갖는 것으로 정의하였다. 본 논문에서는 패턴을 두 가지 의미를 갖는 것으로 정의하였다. 첫 번째 의미로서의 패턴은 특정 제한된 도메인 내에서 빈번하게 사용되는 표현이 있을 때 그것을 패턴으로 정의하였다. 두 번째 의미로서의 패턴은 번역 관점에서 하나의 번역 단위로 취급되어야 하는 표현을 패턴으로 정의하였다. 본 논문에서 제안하는 패턴은 기존의 문법에 충실히 따르는 경우도 있지만, 그렇지 않은 경우도 충분히 존재한다. 이렇게 정의된 문장의 패턴은 제한된 도메인에서는 더욱 뚜렷하게 드러난다. 결국, 이러한 패턴을 사용하여 기존의 예제 기반 기계번역을 수행할 경우, 메타 부분의 사용으로 인해, 적당한 크기의 병렬 코퍼스를 가지고도 높은 유사 예문 발견 확률을 유지시킬 수 있으며, 번역문 생성 단계에서의 부담을 줄일 수 있다.

본 논문에서는 문장을 형성하는 패턴을 크게 문장 패턴과 구 패턴으로 분류하였다. 문장 패턴은 한 문장 전체의 구조적 중심 패턴을 나타내며, 입력문의 구조 변환을 위해 사용된다. 구 패턴은 문장 패턴이 적용되었을 때, 메타 부분만을 다시 분석하여 얻어진다. 구 패턴은 문장 패턴에 비해 상대적으로 높은 우선 순위를 가지므로, 입력문에 대한 패턴 적용 과정에서 패턴 적용의 모호성을 해결하는데, 도움을 줄 수 있다.

표 1과 표 2는 'ITU-T X계열 권고'의 문장들로 구성된 학습 코퍼스에서 나타나는 문장 패턴 및 구 패턴의

일부를 나타낸다. 표에서 보이는 SX1, SX2, SX3, PX1 등은 패턴의 메타 부분이며, 입력 문장에 대한 패턴 적용시, 이 메타 부분에는 다양한 단어열들이 적용될 수 있다.

다음절에서는 문장 패턴 및 구 패턴에 관한 보다 자세히 설명한다.

표 1. 문장 패턴 예
Table 1. Examples of sentence patterns.

영어 문장 패턴	한국어 대역 패턴
SX1 be defined using SX2	SX1'은 SX2'을 사용하여 정의된다
SX1 be provided through SX2	SX1'은 SX2'을 통해 제공된다
SX1 apply to SX2	SX1'은 SX2'에 적용된다
SX1 be referenced by SX2	SX1'은 SX2'에 의해 참조된다
SX1 be derived from SX2	SX1'은 SX2'로부터 유도된다

표 2. 구 패턴 예
Table 2. Examples of phrase patterns.

영어 구 패턴	한국어 대역 패턴
a number of PX1	많은 PX1'
acronym for PX1	PX1'의 약어
information on PX1	PX1'에 대한 정보
reference to PX1	PX1'에 대한 참조
relationship between PX1	PX1' 간의 관계

1. 문장 패턴

본 논문에서 제안하는 문장 패턴은 문장 전체를 커버하는 것으로, 해당 문장의 문법적·의미적 핵심 부분을 패턴화함으로써 얻어진다. 여기서 문장 패턴은 문장의 구조 변환을 위해 사용한다. 이미 앞에서 언급했듯이 문장 패턴은 원시 언어의 해석적 관점이 아니라 번역적 관점을 취한다. 즉, 기존의 해석 문법을 벗어나서 해당 도메인의 코퍼스 상에서 빈번하게 발생하거나, 번역의 관점에서 항상 일관되게 번역되어야 하는 단어열들을 패턴으로 고정시킨다.

(1) *The inner type is identified by means of its identifier.*

(1)' *내부 유형은 식별자에 의해 명시된다.*

문장 (1)의 경우, 적용되는 영어 문장 패턴은 "*SX1 be identified by means of SX2*"이고, 이에 대응되는 한국어 대역 패턴은 "SX1'은 SX2'에 의해 명시된다"이다. 이때 SX1 및 SX2는 메타 부분에 해당하며, SX1'과 SX2'은 각각 SX1과 SX2에 대한 대역 표현에 해당한다. 여기서, 동사 'identify'는 실제 영어 문법에서 "*by means of* ~"를 하위범주화하지 않는다. 하지만 본 논문에서 도메인으로 정한 학습 코퍼스에서는 매우 빈번하게 발생되는 표현이므로 이를 패턴으로 정의하였다.

(2) *This Recommendation defines a number of simple types.*

(2)' *본 권고는 다수의 단순 유형을 정의한다.*

(3) *Encoding rules may define a different type.*

(3)' *부호화 규칙은 다른 유형을 정의할 수 있다.*

문장 (2), (3)을 살펴보면, 두 문장 모두 "*SX1 define SX2*"라는 영어 문장 패턴을 적용할 수 있으며, 대응되는 한국어 번역 패턴은 "*SX1'*은 *SX2'*을 정의한다"이다. 하지만, 문장 (2)와 (3)은 모달리티(modality)가 다르다. 이 경우, 두 문장을 서로 다른 문장 패턴이 적용되는 것으로 취급하면, 가능한 모든 모달리티를 갖는 예문이 있어야만 임의의 모달리티를 갖는 입력문을 올바르게 처리할 수 있다는 문제점이 있다. 본 논문에서는 이러한 단점을 해결하기 위해서 문장 (2)와 (3)을 동일한 문장 패턴이 적용될 수 있는 것으로 하였고, 모달리티에 대한 처리는 번역문 생성 단계에서 처리하도록 하였다.

(4) *The ASN.1 notation is referenced by other standards.*

(4)' *ASN.1 표기법은 다른 표준에 의해 참조된다.*

(5) *Any part of any ASN.1 type definition can be referenced by use of the "AbsoluteReference" syntactic construct.*

(5)' *ASN.1 유형 정의의 임의의 일부가 "Absolute-Reference" 구문 구성의 사용에 의해 참조될 수 있다.*

문장 (4)와 (5)를 서로 비교해 볼 때, 문장 (5)는 문장 (4)에 비해 다소 복잡한 구조를 가지고 있는 것처럼 보인다. 하지만 두 문장 모두에 동일한 문장 패턴이 적용된다. 즉, 문장 (4)와 (5)에 동일하게 적용되는 영어 문장 패턴은 "*SX1 be referenced by SX2*"이며 대응되는 한국어 대역 패턴은 "SX1'은 SX2'에 의해 참조된

다"이다. 문장 (5)에 대해 해당 패턴을 적용하면, 메타 부분인 SX1과 SX2에 대응하는 것은 각각 "*Any part of any ASN.1 type definition*" 및 "*use of the "AbsoluteReference" syntactic construct*"이며, SX1′과 SX2′에 대응하는 것은 각각 "*ASN.1 유형 정의의 임의의 일부*" 및 ""*Absolute- Reference*" 구문 구성의 사용"이다. 이 문장에서 알 수 있는 것은 패턴에 의해 원시 언어 문장을 해석할 경우, 기존의 문법에 비해 원시 언어 해석 단계에서 발생할 수 있는 모호성을 상당히 줄일 수 있다는 것과 패턴 적용 이후, 변환 과정에서 메타 부분을 어느 정도 독립적으로 데이터베이스의 예문을 사용하여 정확하게 어휘 변환할 수 있다는 것이다.

2. 구 패턴

본 논문에서는 문장 패턴을 적용한 이후, 메타 부분에 해당하는 단어열들을 구 패턴으로 다시 분류하였다. 구 패턴은 학습 코퍼스의 각 문장들에 대해서 문장 패턴을 적용하고, 메타 부분에 해당하는 단어열들의 사용 유형을 조사하여 얻어진다.

(6) *A number of constraints can be provided.*
(6)′ *많은 제한이 제공될 수 있다.*
(7) *This Recommendation defines a number of useful types.*
(7)′ *본 권고는 많은 유용한 유형을 정의한다.*

문장 (6)의 경우 적용될 수 있는 문장 패턴은 "*SX1 be provided*"이며, 이에 대한 번역 패턴은 "SX1′이 제공된다"이다. 이 경우, 메타 부분인 SX1에는 "*a number of constraints*"가 할당된다. 문장 (7)에 대해서는 적용될 수 있는 문장 패턴이 "SX1 define SX2"이며, 이에 대한 번역 패턴은 "SX1′은 SX2′을 정의한다"이다. 이 경우에도 앞에서와 마찬가지로 SX1에는 "*this Recommendation*"이, SX2에는 "*a number of useful types*"가 해당된다. 이때, "a number of ~"가 두 문장 모두에서 "많은 ~"으로 번역되는 것을 알 수 있으며, 이를 통해 "*a number of PX1*"이라는 구 패턴의 추출이 가능하며 이에 대한 대역 패턴은 "많은 PX1′"으로 결정된다.

본 논문에서는 패턴 적용에 있어서 구 패턴에 문장 패턴보다 높은 우선 순위를 부여하였기 때문에, 구 패턴은 패턴 적용 과정에서 문장 패턴보다 먼저 적용된다.

(8) *CCIR defines the acronym for Coordinated Universal Time as UTC.*

문장 (8)의 경우, "*acronym for PX1*"이라는 구 패턴을 우선 적용함으로써, "*the acronym for*"를 하나의 단어열로 취급할 수 있다. 즉, "*the acronym for*"를 하나의 단어열로 취급할 경우, 문장 패턴 적용시 'for'에 의한 모호성이 발생되는 것을 막을 수 있다.

이와 같이, 문장 패턴과 구 패턴을 함께 사용하여 입력문을 분석할 경우, 패턴 적용의 우선 순위에 의해 구 패턴을 문장 패턴보다 먼저 적용함으로써 패턴 적용 과정에서 발생할 수도 있는 모호성을 상당량 줄일 수 있다.

Ⅳ. 패턴과 예문을 사용한 영한 변환

그림 1은 본 논문에서 제안하는 변환 기법의 전체적인 흐름을 나타낸다.

그림 1과 같이 본 논문에서는 영한 변환을 위한 변환 지식으로서 패턴으로 표현된 영어 문장과 그에 대응하는 한국어 문장의 쌍으로 구성되어 있는 병렬 코퍼스를 사용한다.

그림 2는 입력문이 번역되는 과정을 구체적으로 나타낸다. 그림 2에서 보이는 패턴과 예문을 사용하여 입력문을 변환하는 과정을 간략히 설명하면 다음과 같다.

입력문이 들어오면 첫 번째 단계에서, 입력문에 대해 패턴을 적용한다. 물론 이 과정은 앞에서 설명했듯이 구 패턴을 먼저 적용하고, 다음에 문장 패턴을 적용한다. 입력문을 커버하는 문장 패턴이 발견되면 일차적으로 문장 전체의 구조적인 변환 과정은 적용된 문장 패턴의 대역 패턴을 사용하여 이루어진다.

두 번째 단계에서는 패턴의 메타 부분에 대한 어휘 변환을 위해 동일한 표층 표현을 사용하고 있는 예문을 검색해서 그 대응 대역 표현을 어휘 변환에 사용한다. 메타 부분의 번역을 위해서 예문을 사용하는 이유는 해당 도메인에서의 가장 적절하면서도 자연스러운 어휘 변환을 수행하기 위해서이다.

다음 절에서는 그림 2의 입력 문장을 가지고 본 논

문에서 제안하는 변환 과정에 대해 자세히 설명한다.

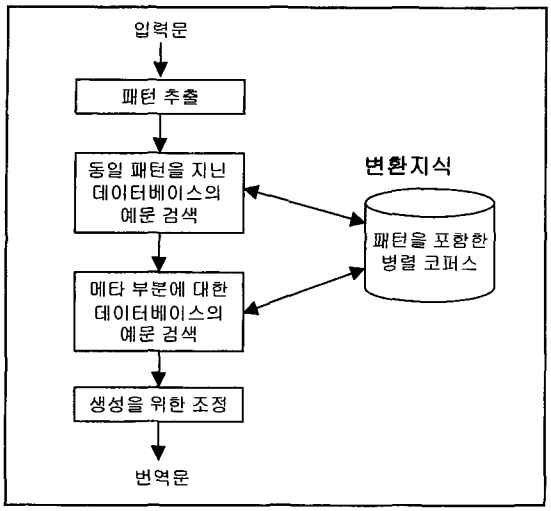

그림 1. 변환 과정

Fig. 1. Transfer procedure.

<입력문장>
This Recommendation defines a number of useful types.

<Step 1: 구 패턴 적용>
This Recommendation defines *a number of* useful types.
PX1

<Step 2: 문장 패턴 적용>
This Recommendation *defines* *a number of* useful types.
SX1 SX2

<구 패턴 및 문장 패턴 적용 결과>
PX1
This Recommendation *defines* *a number of* useful types.
SX1 SX2

<Step 3: 예문을 사용한 메타부분의 변환>
This Recommendation : 본 권고
useful types : 유용한 유형

<각각의 대역 패턴>
SX1 define SX2 : SX1′은 SX2′을 정의한다
a number of PX1 : 다수의 PX1′

<조정 단계를 거친 최종 번역문>
본 권고는 다수의 유용한 유형을 정의한다.

그림 2. 변환 과정 적용 예

Fig. 2. An example of applying transfer procedure.

1. 구 패턴 및 문장 패턴의 적용

번역 대상인 영어 문장이 입력문으로 들어오면 시스템은 우선 입력문을 커버하는 패턴을 추출한다. 이 과정에서 시스템은 문장 패턴보다 우선 순위가 높은 구 패턴을 우선적으로 적용하기 위해 입력 문장의 각 단어를 조사한다. 그 결과 시스템은 그림 2의 입력문에 대해 *"a number of PX1"* 이라는 구 패턴을 적용할 수 있음을 발견한다. 이렇게 구 패턴이 우선적으로 적용될 경우, *"a number of"*는 하나의 단어열로 취급되기 때문에, 문장 패턴 적용시 발생할 수 있는 모호성을 상당량 줄일 수 있다.

구 패턴이 적용되고 나면 입력문 중에서 구 패턴의 고정 단어열을 하나의 단어열로 취급해서 문장 패턴에 대한 적용을 시도한다. 그림 2의 경우는 입력 문장에 대해서 적용 가능한 문장 패턴으로 *"SX1 define SX2"* 가 선택된다. 그리고 선택된 문장 패턴에 대한 한국어 대역 패턴은 "SX1′은 SX2′을 정의한다"로 결정되며, SX1, SX2 등의 메타 부분에는 각각 *"This Recommendation"*, *"a number of useful types"*가 할당된다.

1) 패턴 적용에서 발생하는 모호성의 해소

참고로, 그림 2의 입력문에서는 구 패턴이나 문장 패턴 적용시에 모호성이 나타나지 않지만, 패턴을 사용하여 원시 언어 문장을 해석하는 경우에도 규칙 기반 접근 방법에서와 마찬가지로 동일 문장에 대해서 하나 이상의 패턴을 적용할 수 있는 경우가 발생한다. 본 논문에서는 패턴 적용 단계에서 발생하는 모호성을 해결하기 위해 적용 가능한 패턴 중 표층 어휘(lexical term)를 보다 많이 포함하고 있는 패턴을 선호한다. 그 이유는 표층 어휘를 많이 포함하고 있는 패턴일수록 해당 표현을 커버할 가능성이 높기 때문이다. 문장 (9)는 표층 어휘를 많이 포함하고 있는 패턴을 사용함으로써 모호성을 해결하는 예를 보인다.

(9) Summarization applies algorithms to observed attribute values.

문장 (9)의 경우, 문장 패턴을 적용하면 적용 가능한 문장 패턴 후보로는 *"SX1 apply SX2"* 및 *"SX1 apply SX2 to SX3'*가 가능하다. 이 경우 패턴 적용의 모호성이 발생하게 되며, 이를 해결하기 위해 표층 어휘를 보다 많이 포함한 *"SX1 apply SX2 to SX3'*를 문장 전체를 커버하는 문장 패턴으로 선택한다.

2. 예문을 사용한 메타 부분의 변환

문장 패턴 및 구 패턴이 결정되었으면, 패턴의 메타 부분에 대한 어휘 변환이 수행된다. 이 과정에서는 병렬 코퍼스의 예문을 사용하여 직접적으로 어휘 변환을 수행한다. 예문을 사용하여 어휘 변환을 수행하는 알고리즘을 그림 3에 보였다.

그림 3의 알고리즘을 살펴보면 다음과 같다. 우선, 특정 표현이 병렬 코퍼스에 존재하는지 여부를 리턴하는 SearchDB() 함수를 사용하여 예문을 사용한 메타 부분의 직접 변환이 가능한지를 조사한다. 만약 동일 표현이 병렬 코퍼스 상에서 발견되는 경우에는 직접 변환이 가능하며, *GetTargetExpression()* 함수에 의해 직접 변환이 수행된다. 만약 메타 부분과 동일한 표현이 병렬 코퍼스에서 발견되지 않는 경우에는 번역 단위를 보다 작게 나누어서 각각의 서브 메타 부분을 독립적으로 번역하고 결과를 조합하게 된다. 이때, 메타 부분이 고빈도 기능어를 포함하고 있을 경우에는, *GetSubTargetExpression()* 함수에 의해 번역 표현을 얻는다. 이 과정에서, 번역 단위의 축소에 의해서도 동일 표현이 병렬 코퍼스에서 발견되지 않을 때에는 사전 참조에 의한 디폴트 번역을 수행한다. 그러나 고빈도 기능어의 부재로 번역 단위의 축소가 불가능한 경우에는 사전 참조에 의한 번역을 수행하게 되며, 이 과정은 *GetDefaultTargetExpression()* 함수에 의해 수행된다.

```
Function LexicalTransform(Meta: String): String
var
    Matched, HasFunctionalWord: Boolean;
    TargetExpression, SubMeta1, SubMeta2, FunctionalWord: String;
begin
    Matched := SearchDB(Meta);
    if(Matched == TRUE) then begin
        TargetExpression := GetTargetExpression(Meta);
        return (TargetExpression);
    end
    else begin
        HasFunctionalWord := CheckFunctionalWord(Meta);
        if(HasFunctionalWord == TRUE) then begin
            DivideMeta(Meta, SubMeta1, SubMeta2, FunctionalWord);
            TargetExpression := GetSubTargetExpression(SubMeta1, SubMeta2,
                                FunctionalWord);
            return (TargetExpression);
        end
        else begin
            TargetExpression := GetDefaultTargetExpression(SubMeta1, SubMeta2,
                                FunctionalWord);
            return (TargetExpression);
        end {if}
    end {if}
end {LexicalTransform}
```

그림 3. 어휘 변환 알고리즘
Fig. 3. An algorithm for lexical transformation.

1) 동일 표층 표현이 예문에서 발견되는 경우

그림 2의 입력 예문에 대해서 설명하면 다음과 같다. SX1, SX2에 할당된 메타 부분인 *"This Recommendation", "a number of useful types"*를 어휘 변환하기 위해서 병렬 코퍼스를 검색하여 일치하는 표현을 찾는다. 이 중, SX2에 해당하는 *"a number of useful types"*의 경우는 이미 구 패턴 *"a number of PX1"*이 적용되었으므로 나머지 메타 부분인 *"useful types"*에 대한 대역 표현을 얻기 위한 검색이 이루어지며, 입력문의 각각의 메타 부분과 동일한 표현이 발견될 경우에는 예문의 대역 표현을 사용하여 직접적인 어휘 변환이 가능하다. 즉, 동일한 표층 표현이 병렬 코퍼스 상에서 발견되는 경우에는 그 대역 표현을 사용하여 최종적으로 문장 (10)과 같은 번역문을 생성하게 된다.

(10) 본 권고는 다수의 유용한 유형을 정의한다.

2) 동일 표층 표현이 예문에서 발견되지 않는 경우

입력 문장의 메타 부분이 데이터베이스의 예문에 존재하지 않는 경우에는 해당 메타 부분을 보다 작은 번역 단위로 나누는 과정이 필요하다. 번역 단위를 보다 작게 나누는 이유는 데이터베이스 예문에서의 동일 표현 발견 확률을 높이기 위해서이다. 예를 들어, *"the canonical order for tags"*라는 메타 부분이 데이터베이스의 예문에 존재하지 않는다고 가정하자. 이 경우, 고빈도 기능어 'for'를 중심으로 메타 부분의 표층 표현을 분할하게 된다. "the canonical order for tags"의 경우, 전치사 'for'를 사용하여 분할하면 해당 표현은 "the canonical order"와 "tags"로 나뉘어진다. 시스템은 보다 작은 단위로 분할된 번역 단위인 "the canonical order"와 "tags"에 대한 적당한 대역 표현을 찾기 위해 예문을 검색하여 어휘 변환을 수행한다. 그리고 전치사 'for'에 대한 올바른 해석은 'for'를 포함하고 있는 예문을 검색하여 'for'의 앞, 뒤에 오는 단어간의 의미 관계를 참조하여 최종적인 어휘 변환을 수행한다. 만약 분할 과정 이후에도 변환을 위해 예문을 이용할 수 없으면, 사전을 참조하여 어휘 변환을 수행한다.

이와 같이, 예문을 사용하여 메타 부분에 대한 변환을 수행할 경우, 해당 도메인의 예문을 사용하여 어휘 변환을 수행하므로, 자연스러운 번역이 가능하다는 장점을 지닌다.

3. 생성을 위한 조정

본 논문에서 제안하는 방법의 최종 단계인 조정 단계에서는 하나 이상의 번역 단위로 나뉘어서 번역된 대역 부분들을 적당하게 결합하는 처리를 수행한다. 또한 조정 단계에서는 문장 패턴에 명시되어 있지 않은 모달리티나 부정어구 또는 시제 등의 처리를 수행한다. 이미 앞에서도 언급했듯이, 동일한 패턴이지만 모달리티가 다른 문장들을 서로 다른 패턴을 가지는 것으로 간주한다면 만족스러운 결과를 얻기 위해서는 번역 데이터베이스의 크기가 매우 커야한다는 단점이 있다. 이러한 문제는 순수한 예제 기반 기계번역 방법이 가지는 문제점과 동일한 것이라고 할 수 있다. 따라서, 본 논문에서는 이와 같은 문제점을 해결하기 위해, 시제나 모달리티 등의 차이가 패턴 적용 과정에서는 영향을 미치지 않도록 하고, 번역에 있어서의 미세한 차이를 조정 단계에서 조정하는 것으로 하였다.

V. 실 험

본 장에서는 본 논문에서 제안한 방법의 검증을 위해 수행한 실험에 관해 설명한다. 실험을 위한 제한된 도메인으로서 'ITU-T X계열 권고'의 문장 및 그에 대한 대역 문장을 병렬 코퍼스로서 사용하였다. 표 3은 패턴 추출을 위해 사용한 병렬 코퍼스의 특성을 나타낸다.

표 3. 병렬 코퍼스 구성
Table 3. Parallel corpus figures.

	문장 수	단어 수
병렬 코퍼스	7,587 문장	242,459 단어

구 패턴 및 문장 패턴의 자동 추출은 현재 연구 단계이므로 사람이 반자동으로 추출하였다. 패턴 추출은 비교적 출현 빈도가 높은 17개의 동사를 본동사로 사용하고 있는 문장들 중 복합 관계사절을 포함하고 있지 않은 단문을 그 대상으로 하였다. 이를 위해 선택된 19개의 동사와 해당 동사가 본동사로 사용된 문장의 문장 패턴에 대한 내용이 표 4에 요약되어 있다.

실험에서 사용한 테스트 코퍼스는 표 4의 17개의 동사 중 9개의 동사를 선택하여 해당 동사가 본동사로 사용된 문장들로 구성되어 있으며, 마찬가지로 복합 관계사절을 포함하지 않은 단문으로 구성되어 있다. 표 5는 테스트 코퍼스의 특성을 나타낸다. 표 5에서 동사 뒤의 괄호 안에 있는 숫자는 해당 동사가 본동사로 사용된 문장의 수를 나타낸다.

표 4. 추출된 문장 패턴
Table 4. Extracted sentence patterns.

동사	문장 수	추출된 패턴 수
appear	115	12
apply	182	14
assign	104	13
carry	64	10
consist	93	2
contain	240	2
define	632	23
derive	90	4
identify	180	10
include	221	8
occur	174	12
provide	327	24
reference	147	16
represent	99	6
require	154	8
specify	705	19
use	582	18
전체	4109	201

표 5. 테스트 코퍼스 특성
Table 5. Test corpus figures.

단어 수	5 이하	6~10	11~15	16~20	21 이상
문장 수	6	134	155	18	8
전체	321				
본동사 종류	apply(40), consist(3), contain(34), define(76), derive(13), include(40), provide(29), reference(10), specify(76)				

표 6은 테스트 코퍼스의 각 문장에 대해서 패턴을 적용할 경우, 패턴 적용의 성공 여부 및 발생되는 모호성에 관해 실험한 결과를 나타낸다.

표 6. 문장 패턴 적용 결과

Table 6. Results for applying patterns.

단어 수	5이하	6~10	11~15	16~20	21이상	전체
문장 수	6	134	155	18	8	321
성공	5	114	132	12	4	267
실패	1	1	8	2	2	14
모호성	0	19	15	4	2	40

표 6에서, 문장 패턴의 적용이 실패한 경우를 해당 원인에 따라 분류해 보면 다음과 같다. 첫째, 입력 문장의 문장 패턴이 이미 추출된 문장 패턴의 범위를 벗어나는 경우이다. 둘째, 구 패턴이 적용되어 하나의 고정 단어열로 취급되어야할 단어들에 대해서 구 패턴의 미추출로 인해 문장 패턴이 적용된 경우이다. 셋째, 입력 문장이 매우 복잡한 명사구를 포함하고 있어서 문장 패턴이 잘못 적용된 경우이다.

모호성이 발생한 경우에 대해서도 몇 가지 원인으로 분류할 수 있다. 모호성을 일으킨 가장 큰 원인은 명사구가 다양한 전치사로 결합되어 있는 복잡한 경우이다. 그리고, 빈도는 낮지만, 다른 문장 패턴의 구성 단어들로부터 영향을 받는 경우이다. 이러한 경우는, 현재 고려중인 형태소 태깅 결과를 패턴 적용시 함께 사용하면 충분히 해결될 수 있으리라 생각된다.

표 7은 패턴 적용시 모호성이 발생되는 경우, 모호성 해결에 관한 실험 결과이다. 본 논문에서는 구 패턴을 우선적으로 적용하고, 표층 어휘를 보다 많이 포함하고 있는 패턴을 선호함으로써, 패턴 적용 과정에서 발생할 수도 있는 모호성을 해결하고자 하였다. 표 7이 나타내는 바와 같이 본 논문에서 제안한 방식을 사용하여 패턴 적용시 발생하는 모호성을 어느 정도 해결할 수는 있다. 하지만, 보다 정확한 패턴의 적용을 위해서는 표층 어휘 정보뿐만 아니라, 형태소 정보도 함께 이용하면 보다 효율적인 모호성 해결이 가능하리라 생각된다.

표 7. 패턴 적용 모호성 해결 실험

Table 7. Results for ambiguity resolution.

단어 수	5이하	6~10	11~15	16~20	21이상	전체
모호한 문장 수	0	19	15	4	2	40
모호성 해결	-	17	12	3	0	32
실패	-	2	3	1	2	8

표 8은 병렬 코퍼스의 크기 변화에 따른 메타 부분의 직접 어휘 변환 가능성 여부를 나타낸다. 즉, 구 패턴과 문장 패턴이 적용된 이후, 패턴에 의해 적당한 비교 단위로 나누어진 각 메타 부분들이 병렬 코퍼스의 예문들에 의해서 얼마만큼 커버되는지를 보여주고 있다. 표 8을 보면 병렬 코퍼스의 크기와 병렬 코퍼스에서 메타 부분의 발견 여부는 병렬 코퍼스의 크기가 상당히 커야만 만족스러운 결과를 얻을 수 있음을 나타낸다. 동시에 표 8의 결과는 순수 예제 기반 기계번역 방법에서의 단점을 나타내는 것이기도 하다. 또한 표 8을 통해, 패턴을 사용하여 비교 단위를 줄인다고 하더라도, 동일한 표현이 예문에서 발견되기는 상당히 어렵다는 것을 알 수 있다. 물론 병렬 코퍼스의 크기가 충분히 크다면 만족할만한 결과를 얻을 수 있지만, 그러기 위해서는 병렬 코퍼스 구축 비용 및 검색 비용이 너무 크다.

표 8. 병렬 코퍼스 크기와 메타 부분 발견 가능성의 관계

Table 8. The relation between the parallel corpus size and the possibility to find example matched with meta parts of input sentence.

병렬 코퍼스 크기	병렬 코퍼스에서 발견되는 메타 부분의 수	테스트 코퍼스의 메타 부분의 총수
1000 문장	84 개 (13.4%)	628 개
2000 문장	87 개 (13.9%)	
3000 문장	112 개 (17.8%)	
4000 문장	182 개 (29.0%)	
5000 문장	185 개 (29.5%)	
6000 문장	186 개 (29.6%)	
7587 문장	189 개 (30.1%)	

표 8에서, 예문에서 발견되는 메타 부분에 대해서는 해당 대역 표현을 그대로 사용하여 어휘 변환을 수행할 수 있다. 하지만 그렇지 않은 경우에 대해서는 앞에서 이미 언급했듯이 고빈도 기능어를 사용하여 번역 단위를 나눈 뒤에 유사 표현을 찾아서 변환을 수행하거나, 기존의 사전을 이용해야 한다. 이 과정에서 예제 기반 기계번역 방법은 기존의 규칙 기반 기계번역 방법과 결합될 수 있을 것이다.

Ⅵ. 결 론

본 논문은 영한 기계번역을 위해 패턴과 예문을 함께 사용하여 변환을 수행하는 새로운 방법을 제안하였다. 제안된 방법은 순수 예제 기반의 시스템이 가지고 있던 문제점들을 패턴을 사용하여 해결하고자 하였다. 패턴과 예문을 함께 사용할 경우 얻을 수 있는 장점으로는 다음과 같은 것들이 있다.

첫째, 패턴은 표층 어휘를 포함하고 있기 때문에 패턴을 사용하여 원시 언어를 해석할 경우, 기존의 규칙 기반 기계번역에 비해 발생되는 모호성을 상당량 줄일 수 있었다. 둘째, 패턴은 메타 부분을 포함하고 있기 때문에, 적당한 크기의 번역 데이터베이스를 가지고 유사 예문 발견 확률을 높일 수 있었다. 셋째, 예문을 사용하여 메타 부분의 어휘 변환을 수행함으로써, 해당 도메인에서의 해당 표현에 대한 가장 자연스러운 대역 표현을 얻을 수 있다. 넷째, 기존의 문법 규칙으로 처리하기에는 예외적인 경우가 많았던 대화체(spoken language) 등에 제안된 방식을 적용할 경우, 기존의 대화 시스템에 비해 보다 견고하고(robust), 실용적인 시스템의 개발이 가능하다.

본 논문에서 제안된 방법은 병렬 코퍼스의 각 문장으로부터 문장 패턴과 구 패턴을 자동으로 추출하고 원시 언어 문장과 목표 언어 문장간의 대응 관계를 자동으로 추출하는 기술과 함께 결합될 경우, 번역 시스템의 구축 및 확장을 매우 용이하게 할 수 있을 것으로 생각된다. 따라서 향후 연구 과제로는 주어진 병렬 코퍼스로부터 패턴 및 양언어간 대응 관계를 자동으로 추출하는 기술에 대한 연구가 함께 진행되어야 할 것이다. 또한 본 논문에서 대상으로 한 제한된 도메인에서는 별로 나타나지 않지만, 일반 도메인에서는 비교적 자주 나타나는 번역 모호성도 해결해야 할 문제이다. 또한 보다 실용적인 기능을 갖추기 위해서는 본 논문의 주된 실험 대상이었던 영어의 단문 이외에 하나 이상의 동사 패턴을 지니고 있는 복합문 등의 처리를 위해 기존의 연구를 확장할 필요가 있다.

참 고 문 헌

[1] Nagao. M., "A Framework of a mechanical translation between Japanese and English by Analogy principle," A. Elithorn and R. Barnerji, eds., Artificial and Human Intelligence, pp.173-180, 1984.

[2] S. Nirenburg, C. Domashnev, D. J. Grannes, "Two Approach to Matching in Example-Based Machine Translation," Proceeding of TMI'93, pp.47-57, 1993.

[3] 佐藤 理史, "實例に基づく飜譯,"情報處理, Vol.33, No.6, pp.673-681, 1992.

[4] 古瀨 藏, 隅田 英一郎, 飯田 仁, "變換主導型機械飜譯の實現手法," 自然言語處理80-8, pp.1-8, 1990.

[5] 古瀨 藏, 飯田 仁, "變換と解析の協調的處理による飜譯手法," 自然言語處理87-4, pp.27-34, 1992.

[6] Osamu Furuse and Hitoshi IIDA, "Constituent Boundary Parsing for Example-Based Machine Translation," Proceeding of COLING94, pp.105-111, 1994.

[7] Koich Takeda, "Pattern-Based Context-Free Grammars for Machine Translation," http://xxx.lanl.gov/cmp-lg/.

[8] Hideo Watanabe, "A Method for Extracting Translation Patterns from Translation Examples", Proceeding of TMI'93, pp.292-301, 1993.

[9] 佐藤 理史, アナロジーによる機械飜譯, 共立出版株式會社, 1997.

[10] 이호석, 김영택, "영한 변환사전 생성을 위한 말뭉치에 기반한 연어와 관용어의 자동 추출," 한국정보과학회논문지, 제21권, 제11호, pp.2110-2117, 1994

[11] 서병락, 김영택, "한영 기계 번역을 위한 번역 패턴에 기반한 영어 문장 생성기," 한국정보과학회 논문지, 제23권, 제5호, pp.520-529, 1996

[12] 김나리, 김영택, "한국어 동사 패턴에 기반한 한국어 문장 분석과 한영 변환의 모호성 해결," 한국정보과학회논문지, 제23권, 제7호, pp.766-775, 1996

[13] 송영빈, 최기선, "일한 문형사전을 위한 구문연구," 제10회 한글 및 한국어 정보처리 학술대회, pp.295-303, 1998

저 자 소 개

李 起 榮(正會員)

1994년 한양대학교 전자계산학과 학사. 1997년 한양대학교 전자계산학과 석사. 1997년~현재 한양대학교 전자계산학과 박사과정. 관심분야는 기계번역, 자연어 인터페이스, 정보검색 등

金 漢 宇(正會員) 第25卷 第 10號 參照