한영 자동 번역을 위한 보조 용언 생성

신종훈○, 양성일, 서영애, 김창현, 김영길
한국전자통신연구원, 언어처리연구팀
{jhshin82,siyang,yaseo,chkim,kimyk}@etri.re.kr

English Auxiliary Verb Generation for Korean-to-English Machine Translation

Jong-Hun Shin○, Seong-Il Yang, Young-Ae Seo, Chang-Hyun Kim, Young-Kil Kim
Natural Language Processing Research Team, ETRI

요 약

본 논문에서는 한국어로 입력된 문장을 분석한 결과로부터 그에 해당하는 영어 대역문을 생성하는 과정에서, 어떻게 한국어의 보조용언을 영어 대역문에 반영 할 것인가를 다룬다. 특히 대화체 분야를 다루는 한영 자동번역 시스템에서는 한국어의 보조용언 생성이 대역문의 품질을 향상시키는데 중요한 위치를 차지하기 때문에, 한영 자동 번역에서의 자연스러운 영어 보조용언 생성을 위한 방법론을 제안한다. 첫째, 기존 패턴 기반 한영 자동 번역 엔진과 한국어 말뭉치를 형태소 분석한 결과를 살펴보고, 자연스러운 보조용언 대역어 생성의 어려움을 살펴본다. 둘째, 자연스러운 보조용언 생성에 필요한 양상을 규칙화 한 지식을 기반으로 자연스러운 단일 보조용언 생성을 위한 방법을 제시한다. 셋째, 두 개 이상의 보조용언이 연속해서 나타나는 다중 보조용언의 생성 방법을 제시한다. 마지막으로, 실험과 결론을 통하여 본 논문이 제안하는 방법론을 사용했을 때, 자동 번역 엔진의 성능 평가 지표 중 하나인 BLEU와 NIST점수의 변화를 나타내봄으로 그 성능을 보인다.

주제어: 자동 번역, 기계 번역, 영어 형태소 생성, 보조용언

1. 서론

보조용언은 본용언과 연결되어 그 뜻을 더해주는 역할을 해 주는 보조 동사와 형용사를 일컫는다. 한국어는 어휘적 요소에 문법적 요소를 결합하여 단어나 어절을 만드는 첨가어이기 때문에, 조사와 어미, 보조용언이 매우 발달해있다. 특히 보조용언은 굴절 접사(inflectional suffix)로 많은 문법 기능을 담당, 화자의 양상을 나타내어 전달하고자 하는 의미의 축을 가지는 등 이들 보조용언의 정확한 분석과 번역은 자동 번역 품질을 향상시키는데 큰 도움이 된다[1]. 한국어의 보조용언은 영문 대역문에서 주로 시제, 양태, 부정의 형태로 나타나며, 대역문 내에 존재하는 본동사의 활용형 생성에 영향을 주거나 조동사의 형태로 나타나 본동사의 인접한 곳에 자리를 잡게 된다.

한편, 보조용언을 잘 표현하는 것이 자동 번역 품질을 향상시키는데 도움이 되지만 때로는 번역 대상 도메인에 따라서 보조용언을 표현하지 않는 경우가 도움이 될 때가 있다. '-게하'와 같은 사동적 의미를 가지는 보조용언을 기업에서 사용하는 설명서를 번역하거나, 논문을 번역할 때 대표 대역어인 make를 표현하지 않음을 예로 들 수 있다.

지금까지의 규칙기반 및 패턴기반 자동 번역 연구[2]에서는 대역어 형태소 생성과 관련해 대표에 해당하는 고빈도 출현 대역어를 제시하는 방법론을 바탕으로, 어의 중의성 해소(WSD; word sense disambiguation)을 통해 다른 의미를 지니는 어휘를 올바르게 연결하는데 중점을 두었다. 보조용언의 대역어 생성에 대해서도 해당 방법론은 일관적으로 적용되는데, 한국어에서 연속적으로 복수의 보조용언이 나타날 때 앞에서 언급한 방법론으로 인해 대역문 생성 과정에서 대표 보조용언만 채택함으로 원 문장의 의미가 누락되는 문제가 있었다. 본 논문에서는 규칙을 포함하는 보조용언 대역 지식을 활용하고, 복수의 보조용언이 나타났을 때 대역문에 표현할 보조용언을 선택하거나 혼합하여 원문의 의미를 대역문 생성에 반영하는 방법을 제시하고자 한다.

2. 관련연구

2.1 패턴기반 자동 번역 엔진

본 연구의 기반이 되는 패턴 기반 자동 번역 엔진[3]은, 원문을 대역문으로 번역하기 위해 한국어 입력문에 대해 형태소 분석과 구문 구조 분석을 수행하고, 부분 대역문 연결, 부분 대역문 생성, 대역문 선택 및 정련, 그리고 마지막 단계로 대역어 형태소 생성을 통해 최종 대역문을 생성한다.

패턴 기반 자동 번역 엔진의 특징은 단문 제약 조건을 갖는 용언구 번역 패턴[4]을 구축하여, 중심어인 동사를 기준으로 하는 대역 문장을 생성하는 것이다. 보조용언 생성을 위해 이들 용언구 패턴을 확장하여 적용하는 것

으로 본 논문이 해결하고자 하는 문제를 해결할 수 있도록 접근할 수 있다. 그러나 이런 경우, 본용언에 결합 가능한 보조용언의 범위가 매우 넓어 용언구 패턴의 적용 범위(coverage)가 매우 낮으며, 또한 용언구 패턴을 구축한다고 하더라도 데이터 중복(redundancy) 문제가 발생한다. 이 때문에, 보조용언을 생성하기 위해서는 용언구 패턴을 기준으로 문장의 형태를 잡은 뒤 영어 형태소 생성부에서 본용언에 붙은 보조용언을 생성하는 방식을 사용하는 것이 더 효율적이며, 본 논문에서는 이 처리 순서를 변경하지 않는 대신 기존 대역어 형태소 생성부의 처리 방식을 개선하여 더 나은 번역 품질을 가지는 것을 목표로 한다.

2.2 말뭉치 분석

영어 대역문을 생성하기 전에, 한국어 원문에 대한 형태소 / 품사 분석을 통해 원문이 가지고 있는 기초 정보를 추출한다. 본 논문에서는 수집된 IRC 채팅(chatting) 말뭉치를 분석하여, 자주 출현하는 단일 보조용언 및 다중 보조용언을 살펴봄으로 올바른 보조용언 생성을 위한 요소를 추출하는 기초자료로 사용하였다. 사용된 말뭉치의 총 문장 수는 약 242만 8천 문장으로, 단일 보조용언 출현 빈도의 합은 378,608번이며, 두 개 이상의 다중 보조용언 출현 빈도의 합은 37,496번으로 나타났다. 나타난 단일 / 다중 보조용언을 고빈도 순서대로 상위 5개의 목록을 각각 [표 1], [표 2]에 나타내었다.

표 1 단일 보조용언 출현 빈도 상위 5개

순위	빈도	어휘	대역어
1	18307	지않	not
2	17575	고있	<ger>
3	16028	어야하	have_to<inf>
4	11578	어주	
5	10195	ㄹ수있	can<inf>

표 2 다중 보조용언 출현 빈도 상위 5개

순위	빈도	어휘	대역어
1	370	게하+어주	make<inf>
2	292	면되+지않	can<inf>+not
3	240	어보+아야겠	have_to<inf>
4	224	어야하+지않	have_to<inf>+not
5	198	어주+어야하	have_to<inf>

[표 1], [표 2]에 순위와 빈도, 한국어 어휘와 그에 상응하는 기존 대역 지식을 나열한 것이다. 대역어에 <>기호 내 표기는 본동사의 활용 양상을 나타내는 것이다. <inf>는 원형을, <ger>은 현재진행형으로 생성하라는 명령으로 작용한다. 이들 표에서 보는 바와 같이 대역어가 없는 어휘가 존재하며, 구축이 되지 않거나 표현하지 않는 것이 존재한다.

[표 1]에 있는 단일 보조용언 중 청자가 가져야 하는 의무를 표현하는데 사용되는 '-어야하'의 경우, 현재 대역어가 have_to<inf> 만 존재하나 동일한 의미를 가지는 영어 대역어인 must, ought to, should 중 하나가 들어가는 것이 자연스러운 문장이 되는 경우도 있다. 지금까지의 연구에서는 위 보조용언의 대역어 선택 문제 보다는, 보조용언의 생성에 한정하지 않고 넓은 범위에서 문장의 유창성(fluency)을 향상시키기 위한 방안으로 단일 언어 말뭉치나 이중 언어 말뭉치 기반의 후처리(post-processing)를 적용한다는 관점에서 이를 해소하기 위한 노력이 있어왔다[5]. 본 논문에서는 이 문제를 해결하기 위해 규칙이 내재된 대역 지식을 기반으로 하는 방법론을 제시한다.

또한, 다중 보조용언 생성 문제를 다루기 위해, [표 2]에서 나타나지 않은 세 개 이상의 보조용언이 나타난 경우 일부를 [표 3]에 추가로 나타내었다.

표 3 세 개 이상의 다중 보조용언

빈도	어휘	대역어
14	ㄹ수있+게하+어주	can<inf>+make<inf>
12	지않+을까싶+기도하	not+may<inf>
9	어주+어야하+지않	have_to<inf>+not
9	면되+지않+을까싶	can<inf>+ not+may<inf>
8	어야하+지않+을까하	have_to<inf>+ not+thinking_about<inf>

[표 3]에 있는 다중 보조용언 중'면되+지않+을까싶'의 경우, 보조용언 결합과 선택 문제를 보여주는 예이다. 원 문장에서 "(생성)만 하면 되지 않을까 싶습니다."라는 표현이 들어왔을 때, 위의 보조용언을 단순히 결합하게 될 경우 can<inf>와 may<inf>가 조동사로 문법상 동일한 위치에 사용됨으로 충돌이 일어나게 될 뿐만 아니라, 보조용언'지않'으로 인해 부정 표현이 포함되어, "It may not-"이나 "It can not-"과 같이 틀린 의미를 전달하는 대역문이 생성된다. 따라서 이러한 다중 보조용언이 나타났을 때 이를 선별하거나 표현하지 않아야 할 정보를 제거하는 단계를 추가하여야 한다.

3. 영어 보조 용언 생성

앞서 살펴본 내용을 토대로, 본 논문이 해결하고자 하는 문제의 접근 방법은 다음과 같다. 첫째, 규칙이 내재된 대역 지식을 기반으로, 단문 내에 존재하는 정보를

규칙과 비교하여 최선의 대역어를 선택하게 한다. 둘째, 다중 보조용언이 원문에 나타난 경우, 이를 생성하기 전에 억제시키거나 특정 보조용언을 선택하는 처리구조를 사용한다. 그리고 다중 보조용언으로 결합되거나 선택 생성된 정보 역시 앞서 구축한 대역 지식을 통해 올바르게 생성되도록 한다.

본 논문에서는 단문 내 정보 중 보조용언의 대역어를 결정하기 위해 다음과 같은 요소를 참조하였다. 첫째, 해당 원문의 도메인으로, 기업문서와 대화체와 같이 각기 다른 도메인에서의 대역어를 다르게 한다. 둘째, 단문 내 본용언의 어휘(surface word)를 참조하여 결정하게 한다. 셋째, 서법과 시제 및 부정, 그리고 주어의 인칭에 따라 달라질 수 있게 한다. 넷째, 원문이 단문으로 구성되어있는지, 혹은 복문으로 구성되어 있는지에 따라서도 다른 대역어가 나타나도록 한다. 마지막으로, 문미의 종속 연결 어미로 인한 접속사 여부를 검사할 수 있도록 하였다. 이는 대화체의 처리를 위한 방편으로, 대화체에서는 문미의 종속 연결 어미로 인해 보조용언의 생략된 의미를 복원하는 요소로 활용되기 때문이다.

이들 요소를 참조하는 규칙을 대역어 DB에 포함시키기 위해, 보조용언 대역 지식 구축에 사용된 표기법을 [표 4]에 BNF로 기술하였다.

표 4 규칙 내재 대역 지식 표기법의 BNF

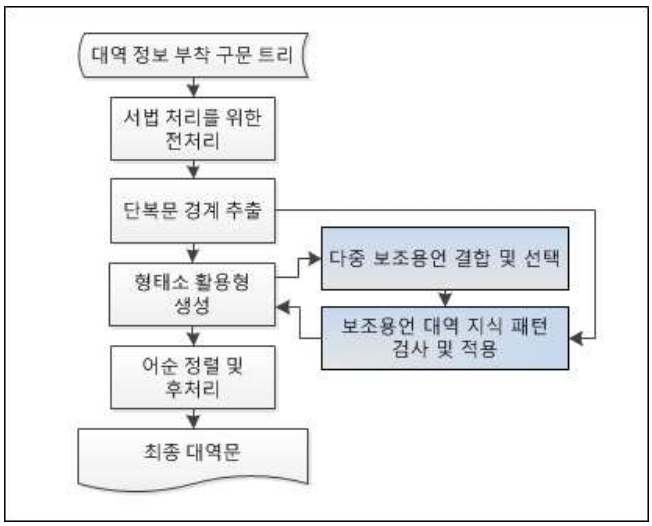
```
<targetwords> :== <expression>
      | <expression> <targetwords>
<expression> :== <rules> <eroot> ';'
<rules> :== '(FEAT' <match-statement>
      '(' <conditions> ')' ')'
<eroot> :== '(EROOT ' <literals> ')'
<match-statement> :== 'DEFAULT' |
      'MATCH' ( 'ALL' | 'PARTIAL' | 'ONCE' )
<conditions> :== <feature> '=' <value>
      | <feature> '=' <value> <contitions>
<literals> :== <literal> ',' <literal>
<feature> :== 'DOM' | 'TENSE' | 'MOOD'
      | 'NEG' | 'EPOS' | 'KPOS' | 'EROOT'
      | 'KROOT' | 'TARGET_RANGE'
<value> := <literal>
```

보조용언 '고싶'의 대표 대역어가 'want_to<inf>'이고, 도메인이 대화체 문장이고 평서문일 경우 'would_like_to<inf>'를 사용해야 한다면, [표 4]에서 기술한 표기법을 통해'(FEAT DEFAULT) (EROOT want_to<inf>) ; (FEAT MATCH ALL (TENSE=PRES DOM=DIALOGUE)) (EROOT would_like_to<inf>) ;'와 같이 지식을 구축 가능하다. 현재 이러한 규칙 기반의 대역 지식은, 표현하고자하는 대상이 되는 보조용언을 포함하는 예문 말뭉치를 사용하여 사람이 구축하고 있으나 더 많은 대역 지식을 구축하기 위해 이중 언어 말뭉치를 사용하여 대역 지식을 추출해 내는 방법을 구현할 필요가 있으며, 이에 대한 연구가 이루어져야 한다. 참고로, 본 논문에서는 <value>에 별도의 규칙을 넣지 않았으나 어순 정렬이 필요한 경우 관련된 추가 정보를 삽입하여 사용해야한다.

다음으로, 다중 보조용언을 올바르게 생성하기위해, 불규칙 형태소 활용을 처리하는 방식과 같이 키-값(Key-Value) DB를 구축하여, 특정 보조용언 패턴이 나타나는 경우 올바른 생성이 되는 방향으로 보조용언의 대역어 정보를 교정하도록 한다. 또한, '-할 수 있을 텐데'와 같은 예문으로 인해 출현하는 'ㄹ수있(can<inf>)'+'을테(will<inf>)'이 같이 나타나는 경우 'ㄹ수있'의 대역어를 'be_able_to<inf>'로 변형하여 'will be able to~'와 같은 표현이 생성될 수 있어야 한다. 이처럼 다중 보조용언의 결합으로 인해 전혀 다른 대역 지식이 필요한 경우 이를 생성하기 위해, [표 4]에 기술한 표기법을 기반으로 대역 지식을 표현해야 한다. 단순히 대역어를 선택하는데 사용되는 문자 상수열 표현과, 규칙 기반의 대역 지식이 같은 DB 내 혼재되어 있기 때문에, 규칙이 포함된 대역 지식을 파싱하는데 실패할 경우 문자열 상수처럼 취급하거나, 아니면 다른 구분자를 삽입하여 이를 구분하도록 한다.

지금까지 제시한 방법론을 바탕으로 재구성한 영어 형태소 생성 루틴의 구조를 [그림 1]로 도시화하였다.

그림 1 영어 형태소 생성 루틴 처리 구조도



기존의 형태소 활용형 생성 처리부를 강화하여, 본용언의 생성 시점에 보조용언을 선택하고, 생성하게 된다. 이때 대부분의 특질은 기존의 보조용언 생성과 동일하게 대역 정보 부착 구문 트리를 통해 얻어오나, 추가적으로 보조용언 선택에 단문 / 복문 여부를 따지고 그 경계를 전달받아 생성에 필요한 단문 구조만을 살피도록 한 뒤 보조용언 대역어 선택을 하도록 하였다.

4. 실험 및 결과

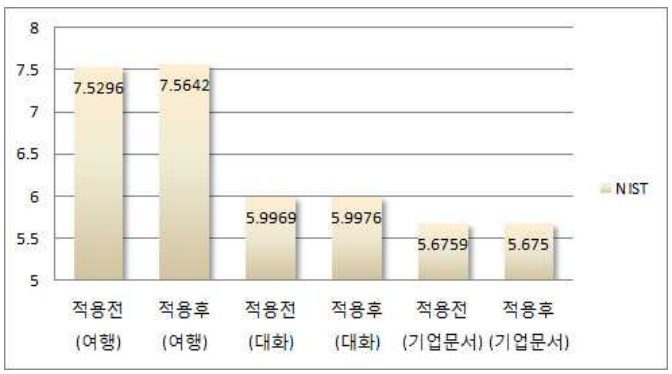
실험에는 각 도메인별로 해당 분야 전공자들에 의해 구축된 평가문장을 사용하여, BLEU[6]와 NIST[7] 점수를 통해 번역률을 계산, 객관적인 평가를 보도록 하였다. 이들 도메인은 각각 여행자 대화체 2000 문장, 메신저 대화체 말뭉치 3998문장, 기업문서 말뭉치 로 구성되어 있으며, 말뭉치 내 보조용언 출현 비중은 여행자 도메인 대화체 평가 집합이 39.4%, 메신저 대화체 말뭉치가 36.5%, 기업문서 말뭉치에서는 72.4%로 나타났다.

성능 비교를 위해 기존의 보조용언 생성방식을 적용한 한영 자동 번역 엔진을 기준치(baseline)로 설정하고, 새로운 방법론을 적용한 지식과 엔진을 통해 점수를 측정하였다.

또한, 본 실험을 위해 구축 및 정제된 지식의 수는 2장의 말뭉치 분석에서 사용된 IRC 채팅 말뭉치에서 나타난 9개의 고빈도 단일 보조용언과 20개의 다중 보조용언 처리 패턴만을 수정한 후 측정하였으며, 그 결과를 [그림 2]의 BLEU 점수와, [그림 3]의 NIST 점수로 나타내었다.

그림 2 보조용언 생성 적용 전 후 BLEU score

그림 3 보조용언 생성 적용 전 후 NIST score



본 실험에서는 여행자 도메인의 대화체 말뭉치에서 변화가 크게 나타났고, 대화체 도메인에서는 작은 변화만 관측되었다. 보조용언의 표현이 제한적인 기업문서 도메인의 경우, 점수의 변화가 거의 없으며, NIST는 오히려 떨어지는 경우가 관측되었다. 이러한 결과는 평가 말뭉치 내 활용된 보조용언에 따라 변화 폭이 크고, 지식 구축이 완료되지 않음으로 인해 생성된 보조용언의 수가 많지 않기 때문이다. 지식 구축에서는 IRC 채팅 대화체 말뭉치가 사용되었기 때문에, 기업문서 도메인의 평가 말뭉치에서 주로 사용된 보조용언과 상대적으로 다른 양상을 가지고 있음을 알 수 있다.

5. 결론

본 논문에서는 한국어 원문을 분석한 결과를 통해 적합한 한국어 보조용언을 영어 대역문으로 생성하는 방법론을 기술하였다. 실험을 통해 대화체에서의 보조용언 생성은 다른 기업 문서 번역이나 논문 번역에 비해 중요한 부분을 차지하고 있음을 보였으며, 또한 다중 보조용언 처리와 규칙이 내재된 지식을 통해 번역 성능이 개선됨을 보였다. 현재 보조용언 생성을 위한 지식이 추가 구축되고 있으며, 이러한 구축 결과에 따라 그 성능이 개선될 수 있는 여지가 존재한다. 사람이 예문을 보고 직접 구축하고 있으나, 추후 규칙이 내재된 지식을 이중언어 말뭉치와 사전에 매칭된 보조용언 대역 어휘 후보를 추출하여 반자동으로 구축하는 방법에 대한 추가적인 연구가 필요하다.

참고문헌

[1] 안동언, “기계번역 시스템 한국어 보조용언 생성”, 정보과학회논문지(B), 제24권, 제 1호, pp. 90-103, 1997.

[2] 최승권, 홍문표,“언어학도를 위한 기계 번역 입문”, 제6회 한국어 정보화 아카데미 강의 자료집, pp. 215-245, 2005.

[3] Changhyun Kim, Munpyo Hong, et al., “Korean-Chinese machine translation based on verb patterns.”, Machine translation: from research to real users: 5th conference of the Association for Machine Translation in the Americas, AMTA 2002, pp. 94-103, October 2002.

[4] 양성일, 김영길 외 4명, “한영 자동 번역을 위한 동사구 번역패턴의 활용”, 2001년도 한국정보과학회 가을 학술발표논문집 Vol.28, No.2, pp. 178-180, 2001.

[5] Behrang Mohit, 외 2명, “Language Model Adaptation for Difficult to Translate Phrases”, In The Proceedings of the 13th Annual Conference of the European Association for Machine Translation (EAMT-09), Barcelona, Spain, pp.160-167, 2009.

[6] Papieni, K. A, 외 3명, “Bleu: a Method for Automatic Evaluation of Machine Translation”, Proceedings of the ACL-02, pp.311-318, 2002.

[7] George Doddington, “Automatic Evaluation of machine translation quality using n-gram

co-occurrence statistics", Proceedings of HLT 2002, pp.138-145, 2002.