통계적 기계번역을 위한 변환 기반 문장 분할 방법*

이종훈 이동현 이근배
포항공과대학교 컴퓨터공학과
{jh21983, semko, gblee}@postech.ac.kr

A Transformation based Sentence Splitting method for Statistical Machine Translation

Jongoon Lee, Donghyeon Lee and Gary Geunbae Lee
Department of Computer Science and Engineering
Pohang University of Science & Technology

요 약

최근 활발하게 연구 되고 있는 통계 기반의 기계 번역 시스템에서는 입력 문장이 길어지면 번역 성능이 떨어지는 현상이 나타난다. 이를 완화하기 위해 긴 문장을 같은 의미의 짧은 문장들로 분할하여 각각 번역하면 기계 번역 성능을 향상 시킬 수 있다. 본 논문에서는 통계적 기계 번역을 위한 변환 기반의 문장 분할 방법을 제안한다. 변환 기반의 문장 분할 방법은 사람이 직접 분할한 예문으로부터 변환 규칙을 학습하여 기계 번역의 입력 문장에 적용함으로써 구절 기반의 통계적 기계 번역 성능을 최대화 한다.

1. 서론

통계적 방식의 기계 번역은 Brown[1]에 의해 정립 된 이후 최근 몇 년간 활발하게 연구 되고 있다. 통계적 기계 번역 방식은 그 동안 많은 연구들이 수행되면서 성능 향상을 거듭해 왔다. 그러나 여전히 많은 문제점이 산재해 있으며, 특히 긴 문장을 번역할 때 그 번역 성능이 떨어지는 문제가 있다.

이러한 현상의 원인은 다양한 관점에서 분석 할 수 있지만 주된 원인은 단어 재배열의 경우의 수가 많아지기 때문이라고 할 수 있다. 한 문장을 번역함에 있어서 가능한 단어 배열의 경우의 수는 문장에 포함된 단어의 수에 기하급수적으로 비례하여 증가한다. 그러나 전통적인 방식의 구절 기반 기계 번역(Phrase-based Machine Translation) 시스템[2]의 단어 재배열 모델은 복잡한 재배열을 제대로 처리하기에 충분하지 않다.

이러한 약점을 보완하기 위한 시도로 구문 기반 기계 번역 방식(Syntax-based Machine Translation)[3] 이나 Inversion Transduction Grammar[4]의 연구를 적용 하여 번역 성능을 향상시킨 바가 있었다.

그러나 이러한 방식들도 궁극적으로는 문장이 너무 길어지면 계산량의 증가를 감당 할 수 없으므로 필연적으로 가지치기 등의 근사 기법을 사용해야 하며, 이는 곧 오류를 포함할 수 있는 가능성으로 이어진다.

따라서, 근본적으로 문제를 해소 또는 완화하기 위해서는 복잡도 자체를 줄일 수 있는 방법을 강구해야 하며, 이것은 입력 문장의 길이를 줄임으로서 달성할 수 있다. 기계 번역의 본래 목적을 해치지 않기 위해 입력 문장의 의미를 보존 하면서 길이를 짧게 하기 위해서는 하나의 긴 문장을 같은 의미의 짧은 문장 여러 개로 다시 써야한다.

한국어의 문장은 그 구조에 따라 단문, 중문, 복문의 3가지 분류로 나눌 수 있다. 이 중에서 단문은 여러 개의 완전한 문장으로 분할하는 것이 불가능하다. 복문의 경우에는 그 구조가 복잡하여 문장 분할 작업에 문장의 전체 구조 분석 작업이 필연적으로 요구된다. 이러한 분석 작업은 통계적 기계 번역 방식과 비슷하거나, 그보다 많은 계산량이 필요할 수 도 있고, 그만큼 오류가 발생할 가능성도 높다. 따라서 복문을 분할하는 것은 통계적 기계 번역의 관점에서 현재로서는 큰 의미가 없다고 할 수 있다. 남은 것은 중문으로서, 중문의 경우는 원래 독립된 문장들이 접속하여 이루어지는 형태이므로 비교적 분할

* 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT 연구센터 지원사업의 연구결과로 수행되었음(IITA - 2006 - C1090 - 0603 - 0045)

하기 쉽다고 할 수 있다. 본 논문에서는 이러한 중문의 분할에 관한 내용을 다루고자 한다.

문장 분할을 통해 성능을 향상시키기 위한 방법들은 여러 가지 형식의 기계 번역 방식에서 연구 되어 왔다. Furuse 등[5]은 음성 통역 연구에서 문장 분할을 적용했고, Doi 등[6]은 예제 기반 기계 번역에 문장 분할 기법을 사용했다. 또한 Jin 등[7]은 SVM을 이용해 중국어 문장 분할을 연구한 바 있다. 그러나 Furuse 등의 방법은 구문 분석을 바탕으로 한 방법론 이므로 앞에서 언급한 이유에 따라 전처리로서의 문장 분할에는 적합하지 않다. 그리고 Doi 등의 방법은 영문에 적용되는 방식으로 단순히 분할 위치만을 잡는 방식이다. 이는 영문에 적용할 때는 유용하나 문장 분할에 따라 연결 어미를 적절한 형태의 종결 어미로 바꿔 써야하는 한국어 문장 분할에는 적용하기 힘들다. 이는 Jin 등의 방법에서도 마찬 가지이다.

한국어 문장 분할을 위한 방법론은 문장 분할 위치를 정확히 추정하는 능력과 함께 연결 어미를 종결 어미로 바꿔 쓰기위한 장치를 제공해야만 한다. 이러한 조건을 만족시킬 수 있는 방법론으로서 변환 기반 학습(Transformation-based Learning)이 있다. 본 논문에서는 한국어를 원시 언어(Source Language)로 하는 언어 쌍에 대한 번역 성능을 향상시키기 위해 변환 기반 학습을 이용한 문장 분할 방식을 제안한다.

2. 문장 분할 방법

본 연구의 목표는 긴 문장들 중에서 복문을 분할하여 짧은 문장들로 바꿔 씀으로서 구절 기반 통계적 기계 번역 시스템의 성능을 향상시키는 것으로, 이를 달성하기 위해 변환 기반의 접근 방법을 사용한다.

2.1. 변환(Transformation)의 소개

변환 기반 학습은 일종의 규칙 학습 방법으로 Brill[8]에 의해 정립된 방법론 이다. 변환 기반 학습은 형태소 분석이나 구문 분석[9] 등의 자연어 처리의 여러 가지 문제를 푸는데 사용된 바 있다.

변환 기반 학습에서 정의하는 변환은 유인 환경(triggering environment)과 다시 쓰기 규칙(rewriting rule)으로 구성되며, 다시 쓰기 규칙은 원본 패턴과 목적 패턴으로 구성 된다. 여기서는 이러한 변환 규칙을 얻는 방법과 그것을 적용하는 방법을 설명한다.

변환은 다음과 같은 형식으로 동작한다. 만약 어떤 입력 패턴이 유인 환경에 기술된 조건과 부합 한다면 다시 쓰기 규칙을 입력 패턴에 적용한다. 그 결과로 입력 패턴은 다시 쓰기 규칙의 원본 패턴과 일치하는 부분이 목적 패턴으로 치환된 형태로 변형된다. 예를 들어 유인 환경이 조건 A이고, 다시 쓰기 규칙이 B를 C로 변경하는 것인 변환이 있다면 이는 다음과 같은 문장으로 표현할 수 있다. “만약 조건 A가 충족 된다면 입력에서 패턴 B를 찾아 C로 치환한다.”

2.2. 변환 기반의 문장 분할 방법

보통, 변환을 적용할 때는 두 가지 선택지가 주어진다. 하나는 해당 되는 모든 변환을 동시에 적용하는 것이고, 또 하나는 미리 마련된 기준에 따라 하나씩 순서대로 적용하는 것이다. 이러한 선택은 풀고자 하는 문제의 성격에 달린 것으로, 매우 중요하다 할 수 있다. 그러나 문장 분할 문제에 있어서는 어느 쪽을 택해도 관계가 없다. 그것은 문장 분할에 있어서 두 개 이상의 변환이 한 문장에 동시에 적용되면서 그들이 서로 영향을 주고받는 경우가 거의 발생하지 않기 때문이다.

본 논문에서는 앞에서 언급한 두 가지 선택지 중에서 변환을 하나씩 적용하는 방법을 택했다. 이것은 알고리즘을 재귀적으로 설계하는 데 있어서 더 유리하기 때문이다. 재귀적으로 설계된 문장 분할 과정은 다음과 같다. 먼저, 주어진 입력에 대해 변환들을 적용하여 분할을 시도한다. 하나의 변환은 한 번에 한 문장을 최대 2개로 분할하고, 한 번에 하나의 변환만 적용된다. 따라서 입력 문장은 첫 단계에 최대 두 개의 문장으로 분할 될 수 있다. 만약 첫 번째 시도에서 분할되지 않았다면 과정은 종료되고, 분할되었다면 결과로 나온 각각의 문장들에 대해 다시 분할을 시도한다. 이러한 과정은 더 이상 문장 분할이 일어나지 않을 때 까지 계속 된다.

앞의 2.1. 절에서 일반적인 변환의 적용에 대해서 설명하였으나, 문장 분할을 위한 변환은 조금 다르다. 문장 분할을 위해서는 다시 쓰기 규칙뿐만 아니라 문장을 나누어야 할 위치 및 나누어진 문장들의 접속 패턴에 관한 정보도 필요하다. 따라서 문장 분할을 위한 변환은 추가적으로 2개의 구성 요소를 더 포함하게 되어 유인 환경, 다시 쓰기 규칙, 접속 패턴, 분할 위치로 구성 된다.

이러한 변환의 각 구성 요소들은 다음과 같은 내용으로 구성된다. (1) 유인 환경은 품사 태그가 달린 형태소 순열이다. (2) 다시 쓰기 규칙은 연결 어미를 종결 어미로 변환 하는 규칙이다. (3) 접속 방식은 ‘그리고’, ‘또는’,‘그러나’, ‘NULL' 중의 하나의 값을 가진다. (4) 분할 위치는 음이 아닌 정수로서 분할 후 두 번째 문장의 첫 단어가 될 단어의 위치를 뜻한다.

위에 기술된 변환이 문장 분할에 적용되는 과정은 다음과 같다. 어떤 입력이 주어지면 유인 환경에 부합 되는지 검사를 한 후, 다시 쓰기 규칙을 적용한다. 이렇게 변환 된 문장을 미리 지정된 위치에서 분할하여 두 문장으로 만들고 그 사이에 접속 패턴에 따른 접속사를 삽입하여 적용을 완료한다.

2.3. 문장 분할을 위한 변환의 학습

본래의 변환 기반 학습에서는 훈련 데이터에 대한 오류를 최소화 하는 변환을 먼저 찾고, 이를 적용하여 훈련 데이터를 변경하고, 변경된 데이터에 다시 오류를 최

```
BaseBLEU :=  BLEU score of the baseline system
S := Split example sentence
T := Extracted initial transformation
for each t∈ T
    for each s∈S
        while true
            try to split s with t
            if mis-splitting is occurred
                Expand environment
            else exit while loop
            if environment cannot be expanded
                exit while loop
S' := apply t to S
Decode S'
BLEU := measure BLEU
Discard t if BLEU < BaseBLEU
sort  T w.r.t. BLEU
```

그림 1. 문장 분할을 위한 변환 기반 학습 알고리즘

소화 하는 변환을 찾아내어 적용하는 과정을 반복함으로써 변환들의 우선순위를 정한다. 그러나 통계적 기계 번역을 위한 문장 분할에서는 직접적으로 문장 분할 자체의 성능을 높이기보다 분할 후 번역 성능을 최대화 하는 방법을 택하는 것이 타당하다. 번역 성능을 측정하는 기준으로는 BLEU* 점수[10]를 사용한다..

변환 기반 문장 분할 방식의 학습 단계에서 가장 첫 번째로 수행해야 하는 것은 우선순위를 정할 대상인 변환을 구축 하는 작업이다. 변환을 구축 하는 과정은 다시 쓰기 규칙을 추출하는 것으로 시작 된다. 다시 쓰기 규칙은 미리 분할 된 예제 문장에서 최소 편집 거리(Edit Distance)를 구하기 위한 행렬을 계산하여 추출 할 수 있다. 즉, 원문과 분할 된 문장을 최소 편집 거리로 정렬했을 때, 서로 다른 부분이 있다면 그것이 바로 다시 쓰기 규칙이 된다. 분할 위치와 접속 패턴은 미리 분할 된 예제로부터 직관적으로 얻을 수 있다. 특별한 과정이 필요한 부분은 유인 환경인데, 초기에는 이것을 다시 쓰기 규칙의 원본 패턴과 똑같이 설정한다. 처음 이 상태의 변환을 앞으로 초기 변환이라 하기로 한다.

유인 환경은 초기 변환에서 간단하게 정의 된 후 주어진 분할 예제에 대해서는 문장 분할 오류를 일으키지 않도록 확장된다. 문장 분할에 있어서 유인 환경은 문장 분할로 변경되는 부분의 형태소 패턴이며, 입력 문장에서 그와 같은 패턴이 나타날 때 조건이 충족된 것으로 간주한다. 따라서 유인 환경의 확장은 앞뒤의 형태소를 더 포함 시키는 것을 의미한다. 확장 과정은 각 변환을 분할 예제에 적용 시킨 뒤, 잘못 분할하는 경우가 있는

* 정답 문장은 각 1개를 사용하였다.

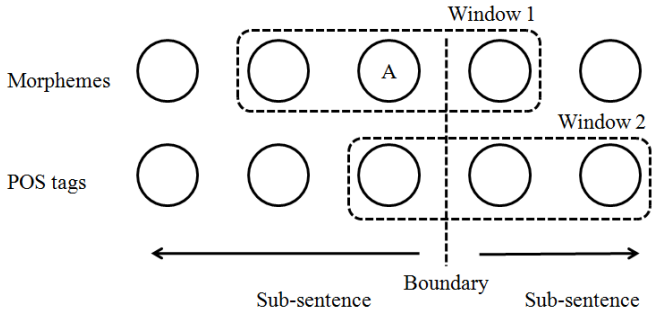

그림 2. 2개의 window를 이용한 유인 환경 확장 방식

지를 확인 하여, 잘못 분할하는 경우가 일어날 때 마다 유인 환경을 확장하는 것의 반복으로 이루어진다. 이러한 과정을 통해 확장된 변환들은 적어도 주어진 분할 예제에 대해서는 오류를 발생 시키지 않게 된다.

위의 확장 과정을 통해서 얻은 변환들에 대해 우선순위를 정하는 과정은 다음과 같다. 각 변환을 모든 예제에 적용 시킨 뒤, 그 결과물을 번역하여 BLEU 점수를 측정한다. 이 때 문장 분할을 적용하지 않은 경우보다 오히려 BLEU 점수를 감소시키는 변환들은 제외하고, 나머지 변환을 BLEU 점수가 큰 것부터 순서대로 정렬하여 BLEU 점수가 높은 것이 높은 우선순위를 가지도록 한다. 위 과정은 원래의 변환 기반 학습보다 간략화 된 것이다. 전체적인 과정은 그림 1에 의사 코드 형태로 기술 하였다.

2.4. 유인 환경의 확장

유인 환경이 너무 과하게 확장되면 그 변환은 거의 적용될 기회가 없게 되고 반대로 너무 적게 확장되면 그 변환은 오류를 포함할 가능성이 크게 된다. 따라서 유인 환경을 확장하는 과정은 매우 주의 깊게 다루어야 한다.

이러한 특성을 감안하여 유인 환경의 확장을 정밀 하게 다루기 위해서 본 연구에서는 유인 환경의 확장을 2개의 window를 통해 좀 더 정밀하게 조절 하는 방법을 사용했다. 하나의 window는 형태소 패턴을 조정 하고, 또 하나의 window는 품사 태그 패턴을 조절 한다. 이러한 개념은 그림 2에 나타나 있다. 그림 2에서 윗줄의 원들은 형태소 패턴을 나타내고 아랫줄의 원들은 품사 태그 패턴을 나타낸다. 점선으로 표시된 사각형들은 각 window의 현재 상태를 나타낸다. 다시 쓰기 규칙이 패턴 A를 바꿔 쓰는 것이므로 초기 변환에서 window는 둘 다 A 가 위치한 부분만을 덮고 있었다는 것을 알 수 있다. 그림 2에 나타난 현재 상태는 window1 이 앞뒤로 한번씩, window2가 뒤로 두 번 확장된 것이다. 이처럼 각각의 window는 훈련 과정에서 변환이 오류를 발생 시킬 때 마다 독립적으로 확장된다. 이때 각 window마다 전혀 확장하지 않거나, 앞으로만 확장하거나, 자유롭게 확장하

		기계 번역 모델		문장 분할	
		한국어	영어	분할 전	분할 후
훈련	문장 수	123,425		1,577	1,906
	단어 수	1,083,912	916,950	19,918	20,243
	어휘 수	15,002	14,242	1,956	1,952
테스트	문장 수	1,577		-	-

표 1. 데이터 통계

실험번호	형태소 window 확장 제약	품사 window 확장 제약
실험1	확장 안함	확장 안함
실험2		앞쪽 확장
실험3		모든 방향
실험4	앞쪽 확장	확장 안함
실험5		앞쪽 확장
실험6		모든 방향
실험7	모든 방향	확장 안함
실험8		앞쪽 확장
실험9		모든 방향

표 2. 실험 설정

실험번호	분할된 문장 수	BLEU	
		분할전	분할후
실험1	209	0.1778	0.1838
실험2	142	0.1564	0.1846
실험3	110	0.1634	0.1863
실험4	9	0.1871	0.2150
실험5	96	0.1398	0.1682
실험6	100	0.1452	0.1699
실험7	8	0.2122	0.2433
실험8	157	0.1515	0.1727
실험9	98	0.1409	0.1664

표 3. 문장 분할로 변화된 문장들에 대한 BLEU 점수 변화

는 3단계의 제약을 걸 수 있다. 3개의 제약을 2개의 window에 대해 조합하여 사용할 수 있으므로 이를 통해 도합 9가지 확장 방식을 정의 할 수 있다.

2개의 window를 정의함으로서 한 번의 확장이 일어날 때 마다 형태소 window를 앞이나 뒤로 확장하거나, 품사 태그 window를 앞이나 뒤로 확장하는 총 4가지의 선택이 가능하다. 확장 과정에서는 이 선택지 중의 하나를 어떤 고정된 순서에 따라서 돌아가며 하나씩 선택하는데 그 순서는 다음과 같다. 품사 태그 window를 앞으로 확장, 형태소 window를 앞으로 확장, 품사 태그 window를 뒤로 확장, 형태소 window를 뒤로 확장. 마지막 것을 선택한 다음에는 다시 처음으로 돌아가서 선택하여 위의 순서를 반복한다. 그런데 위의 9가지 중 하나가 먼저 결정 되면 그 결과로 이 4가지의 선택은 제약 받을 수 있다. 예를 들어 확장 방식이 형태소 window를 앞으로 확장하는 것만 허용하고 품사 태그 window를 확장하지 않는 것이라면 가능한 선택지는 형태소 window를 앞으로 확장하는 것 밖에 없다.

3. 실험

2. 절 에서 기술된 문장 분할 방식이 실제 번역 결과에 미치는 영향을 알아보기 위해서 일련의 실험을 진행했다. baseline 시스템은 가장 잘 알려진 구절 기반 통계적 번역기인 Pharaoh[11]를 사용하여 구축했고 언어모델은KN-discounting[12]을 이용한 trigram 모델을 SRILM [13]를 이용해서 생성하여 사용했다.

실험에 사용된 데이터 통계는 표 1에 기술 했다. 번역기 훈련용 말뭉치는 한글 원문을 수집하여 수동으로 번역하여 얻은 것으로 약 12만 문장쌍이다. 또한 문장 분할 위한 변환을 학습하기 위해 비교적 긴 문장 위주로 1,577 문장을 택하여 수동으로 분할했다. 또한 테스트를 위해서, 이와 다른 1,577 문장을 역시 긴 문장 위주로 뽑아 번역문과 함께 준비했다.

앞에서 언급한 9가지의 확장 방식을 비교하기 위한 실험 설정은 표 2에 나열 하였으며 각 실험 번호마다 한 가지 확장 방식이 대응 된다. 실험은 각각의 방식을 이용해서 변환을 학습한 후 테스트 데이터에 적용 한 뒤 번역 및 평가 하는 것으로 진행 되었다.

문장 분할이 번역에 미친 영향을 확인하기 위해서 변환에 의해 영향을 받은 문장들만을 따로 모아서 BLEU 점수의 변화를 측정한 결과를 표 3에 기술 하였다. 표 3의 결과들은 모든 경우에 대해서 일관적인 BLEU 점수의 증가를 보여 주고 있다.

사람에 의한 평가결과는 표 4에 정리했다. 사람에 의한 평가 에서는 BLEU 점수에 비해 좀 더 직관적으로 결과를 확인 할 수 있다. 표 4에서 개선된 변화는 문장 분할 이후에 더 나아진 문장의 수이며, 악화된 변화는 번역이 더 나빠진 경우, 비슷한 변화는 번역 결과에 큰 차이가 없는 경우를 의미한다. 사람에 의한 평가에서도 역시 일관되게 개선된 경우가 더 많은 것으로 나타나

실험 번호	변환의 수	분할 된 문장 수	개선된 변화	악화된 변화	비슷한 변화	개선/악화 비율	변환/변화 비율
1	34	209	60	30	119	2.00	6.15
2	177	142	43	9	90	4.78	0.802
3	213	110	29	9	72	3.22	0.516
4	287	9	4	1	4	4.00	0.031
5	206	96	25	4	67	6.25	0.466
6	209	100	23	8	69	2.88	0.478
7	256	8	3	1	4	3.00	0.031
8	177	157	42	10	102	4.20	0.887
9	210	98	21	4	73	5.25	0.467

표 4. 사람에 의한 평가 결과

개선된 변화	원문	유리창에 일부품목할인이라고 적혔는데 어떤 품목이 할인되는 거죠?
	정답번역	I saw that some items are on sale on window . what are they ?
	분할 전	What kind of items do you have this item in OOV some discount, I get a discount ?
	분할 후	You have this item in OOV some discount . what kind of items do I get a discount ?
비슷한 변화	원문	신용카드를 만들려고 하는데 무엇이 필요합니까?
	정답번역	What is necessary to be issued a new credit card ?
	분할 전	I ‘d like to make a credit card . What do I need ?
	분할 후	I ‘d like to make a credit card . What is necessary ?
악화된 변화	원문	전화 예매하려고 하는데 번호 좀 알려주세요.
	정답번역	I ‘d like to make a reservation by phone and tell me the phone number please .
	분할 전	I ‘d like to make a reservation but can you tell me the phone number , please .
	분할 후	i ‘d like to make a reservation . can you tell me the , please .

표 5. 번역례

BLEU 점수와 같은 결과를 보였다. 개선된 경우와 악화된 경우의 비율로서 따질 경우 실험 1과 실험 6을 제외하고는 모두 3이 넘는 것으로 나타났다. 즉, 개선된 경우가 악화된 경우가 3배 이상 많았다. 이 비율에 있어서는 실험 5가 가장 높은 결과를 보였다. 변환과 영향 받은 문장 수의 비율은 한 개의 변환이 평균적으로 영향을 준 문장의 수를 의미하며 실험 1을 제외 하고는 1에 못 미치는 것으로 나타나 적용성에 있어서는 다소 떨어지는 결과를 보였다.

문장이 개선된 경우는 여러 가지 경우가 있었지만 한 가지 흥미로운 결과가 미등록어와 관련하여 나타났다. 보통 미등록어가 입력 문장에 섞여 있으면 번역 결과는 어순이 뒤섞여 전체적인 의미를 파악하기 힘들게 되는 경향이 있다. 그런데 문장 분할로 인해 미등록어가 한쪽 문장에 집중됨으로서 다른 한쪽 문장의 의미는 살아나는 경우가 관찰되었다. 특히, 미등록어가 하나뿐일 때는 반드시 분할 후에 한쪽 문장에만 미등록어가 존재 하므로 나머지 문장은 더 좋은 결과를 얻는 경향을 보였다. 이러한 예가 표 5에 제시되어 있다.

더 악화된 경우는 주로 문장 분할이 잘못 된 경우에 발생 하였는데, 분할 지점이 잘못 된 경우는 양쪽 문장 모두 잘못된 의미를 나타내거나 비문이 되므로 악영향을 끼쳤다. 분할이 잘 된 경우에도 간혹 번역 결과에 악영향을 준 경우가 있었는데 원래 좋은 번역예가 모델에 포함 되어 있었으나 문장 분할로 인해 그것이 번역에 사용되지 못하여 발생한 것으로 파악 되었다.

4. 결론

본 논문에서는 한국어 문장 분할을 통해 한-영 통계적 기계 번역 성능을 향상시키기 위한 방법으로 변환 기반의 문장 분할 방식을 제안하였다. 변환은 유인 환경과 다시 쓰기 규칙으로 구성 되며, 변환의 학습은 미리 분할된 훈련 데이터에서부터 초기 변환을 학습 하여 이를 주어진 테스트 데이터에 대해서 오류가 없도록 확장 한 뒤 이들을 BLEU 점수를 최대화 하도록 골라내고 순서 매김 하는 과정을 통해 이루어진다.

일련의 실험을 통해 여러 가지 유인 환경 확장 방식

중에서 연결 어미 앞쪽의 형태소 및 품사 태그 정보가 가장 큰 도움을 주며, 변환 기반의 문장 분할 방식이 구절 기반 통계적 기계 번역의 성능을 향상시키기 위한 일종의 전처리로서 유용하게 적용될 수 있다는 결론을 얻을 수 있었다.

참고 문헌

[1] Peter F. Brown, Stephen A. Della Pietra, Vincent J.Della Pietra and Robert L. Mercer. 1993. The Mathematics of Statistical Machine Translation: Parameter estimation. *Computational Linguistics, 19(2):263-312.*

[2] Philipp Koehn, Franz Josef Och and Kevin Knight. 2003. Statistical Phrase-Based Translation. *In Proc of the of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology.*

[3] Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation Model. *In Proc. of the conference of the Association for Computational Linguistics (ACL).*

[4] Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics 23(3):377-404.*

[5] Osamu Furuse, Setsuo Yamada and Kazuhide Yamamoto. 1998. Splitting Long or Ill-formed Input for Robust Spoken-language Translation. *In Proc of the 36th annual meeting on Association for Computational Linguistics.*

[6] Takao Doi and Eiichiro Sumita. 2004. Splitting input sentence for machine translation using language model with sentence similarity. *In Proc. of the 20th international conference on Computational Linguistics.*

[7] Mexixun Jin, Mi-Young Kim, Dongil Kim, and Jong-Hyeok Lee. 2003, Segmentation of Chinese Long Sentence Using Support Vector Machine. *제 15회 한글 및 한국어 정보처리 학술대회 발표 논문집.*

[8] Eric Brill. 1995. Transformation-based error-driven learning and natural language processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics 21(4):543-565.*

[9] Eric Brill. 1993. Transformation-based error-driven parsing. *In Proc. of third International Workshop on Parsing.*

[10] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2001. BLEU: A method for automatic evaluation of Machine Translation. *Technical Report RC22176, IBM.*

[11] Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *In Proc. of the 6th Conference of the Association for Machine translation in the Americas.*

[12] Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram lnguage modeling. *In Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

[13] Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. *In Proc. of the 7th International Conference on Spoken Language Processing (ICSLP).*