한영 병렬 코퍼스 구축을 위한 하이브리드 기반 문장 자동 정렬 방법

박정열(시스트란), 차정원(창원대)

<차 례>

<Abstract>

A Hybrid Sentence Alignment Method for Building a Korean-English Parallel Corpus

Jungyeul Park, Jeong-Won Cha

The recent growing popularity of statistical methods in machine translation requires much more large parallel corpora. A Korean-English parallel corpus, however, is not yet enoughly available, little research on this subject is being conducted. In this paper we present a hybrid method of aligning sentences for Korean-English parallel corpora. We use bilingual news wire web pages, reading comprehension materials for English learners, computer-related technical documents and help files of localized software for building a Korean-English parallel corpus. Our hybrid method combines sentence-length based and word-correspondence based methods. We show the results of experimentation and evaluate them. Alignment results from using a full translation model are very encouraging, especially when we apply alignment results to an SMT system: 0.66% for BLEU score and 9.94% for NIST score improvement compared to the previous method.

* Keywords: Sentence alignment, Hybrid alignment method, Parallel corpus.

1. 서 론

최근에 들어 통계적 기계 번역에 대한 연구가 활발해짐에 따라 자동으로 병렬 코퍼스를 구축하고 문장을 정렬하는 방법에 대한 관심이 다시 높아지고 있다. 영어와 불어로 제공되는 캐나다와 의회 기록물(Canadian Hansards)[1]이나 영어, 불어, 독일어 등 12개 국어 이상의 유럽어를 제공하는 유럽 의회 기록물(European Parliament Proceedings)[2]과 같이 특정 언어들 간의 병렬 코퍼스를 구축한 사례는 쉽게 찾을 수 있다. 또한 중국어와 영어의 개별 코퍼스에서 기사 간의 유사도와 문장들의 유사도를 계산하여 병렬 코퍼스로 구축한 사례도 있다[3]. 하지만 한국어의 경우, 21세기 세종계획[4]에서 일부 구축한 한국어와 영어, 한국어와 일본어 병렬 코퍼스와, 전문용어언어공학연구센터(Korterm)에서 구축한 한국어와 영어, 한국어와 중국어 6만 문장 정도의 병렬 코퍼스[5]만 존재하여 통계적 기계 번역 학습에 사용하기에는 턱없이 부족한 형편이다.

병렬 코퍼스 구축에는 대량의 리소스를 구할 수 있어야 하며 또한 문장 단위로 정렬이 되어야 한다. 병렬 코퍼스를 구축하는데 있어서 가장 쉽게 생각할 수 있는 리소스로 여러 언어로 번역되어 제공되는 신문이나 잡지 기사와 독해 공부를 위한 대역 문서, 한국어로 번역된 기술 문서나 도움말 파일 등이 있다. 병렬 코퍼스 구축에서는 문학 작품이나 종교 관련 서적들의 사용은 배제되는데, 이는 대체로 직역인 의회 기록물이나 기술 문서와 달리 문학 작품이나 종교 관련 서적들은 주로 의역을 사용하기 때문에 기계 번역과 같은 자동 처리에서는 기대하는 만큼 적절한 결과를 얻을 수 없기 때문이다.

문장 정렬은 병렬 코퍼스를 만들고 사용하는데 있어서 가장 중요한 작업이다. 정렬된 문장으로 구축된 병렬 코퍼스는 기계 번역뿐만 아니라 단어 모호성 해결 문제[6], 다국어 정보 검색[7]과 같은 다른 분야에서도 사용될 수 있는 유용한 자원이 될 수 있다. 문장 정렬은 수동으로 하는 것이 가장 좋으나 대용량의 코퍼스를 작성해야 하기 때문에 자동 정렬에 관심을 가지게 되었다. 그러나 자동 정렬에는 해결해야 할 문제는 다음과 같다. 보통 사람이 문장을 번역할 때 원시 언어에서 하나의 문장을 대상 언어에서도 하나의 문장으로 항상 번역을 하지 않는다는 점이다. 따라서 문장 정렬에서의 문제점은 원시 언어의 문장 묶음이 대상 언어의 문장 묶음으로 사상(mapping)되며 이들 문장 묶음은 0개 이상의 문장을 허용하기 때문에 그 외에는 번역자의 의도에 따라 번역된 텍스트에 새로운 문장의 삽입이나 삭제가 가능하다. 일반적으로 하나의 문장이 하나의 문장으로 번역되는 경우를 1:1 대응이라 하며 이전 연구에서 병렬 코퍼스에서 90%정도가 이에 해당한다고 한다[8]. 하나의 문장이 두 개로 나뉘거나 두 개의 문장이 하나로 합쳐질 수도 있으며(1:2 또는 2:1), 심지어는 1:3이나 3:1 정렬도 가능하다. 이렇게 문장들이 쌍으로 묶이는 단위를 비드(bead)라고 한다.

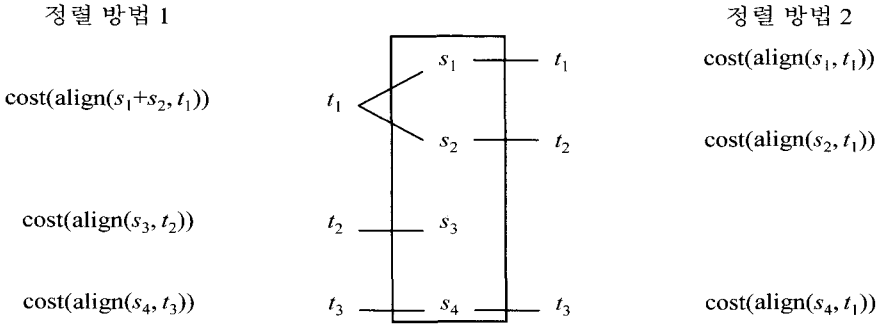
본 연구에서는 기존의 방법들의 이러한 단점을 보완하기 위해, 단순한 단어 수준의 번역을 사용하는 것이 아니라 영한 번역기를 사용하여 원시 코퍼스의 언어(본 논문에서는 한국어)를 대상 코퍼스의 언어(영어)로 완전히 번역한 후, 길이 기반 및 단어 기반의 유사도를 계산하여 문장을 정렬한다.

이후 본 논문의 구성은 다음과 같다. 2장에서는 이전 연구에서 사용한 문장 정렬 방법을 소개하고, 3장에서는 본 논문에서 사용하는 새로운 방법을 제안한다. 그리고, 4장에서는 실험 및 평가를 보이고 5장에서는 본 논문의 결론을 맺는다.

2. 관련 연구

본 장에서는 이전 연구에서 제안되었던 문장 정렬 방법들을 간단하게 정리하여 소개한다. 문장을 자동으로 정렬하는 방법으로는 크게 길이에 기반으로 하거나 어휘 정보를 사용하거나 이들 두 방법을 결합한 하이브리드 정렬 방법이 있다.[1)]

문장을 정렬하는 방법은 문장 정렬에 필요한 비용의 모든 경우를 계산하여 최소한의 비용을 가지는 문장 정렬 경우를 선택한다. <그림 1>에서는 문장 s_1, s_2, s_3, s_4를 가지는 언어 L1과 문장 t_1, t_2, t_3, t_4를 가지는 언어 L2에 대해 두 가지 가능한 문장 정렬의 예를 보인다. 정렬 방법 1에서는 s_1, s_2와 t_1이 정렬(2:1), s_3과 t_2가 정렬, s_4와 t_3이 정렬(1:1)되고, 정렬 방법 2에서는 s_1과 t_1이 정렬, s_2와 t_2가 정렬, s_4와 t_3이 정렬되며 s_3은 L2의 어떤 문장과도 정렬(1:0)되지 않는다.



<그림 1> 문장 정렬의 예[15]

1) 그밖에도 신호 처리에서 사용하는 오프셋을 통한 정렬 방법도 있다[17][18]. 이 기술은 문장의 비드를 정렬하기 보다는 두 개의 병렬 텍스트에서 오프셋의 위치를 정렬하여 원시 언어의 오프셋이 대상 언어의 오프셋과 정렬되는 것을 보인다. 이 논문에서는 이 방법에 대해서는 기술하지 않는다.

2.1. 길이기반

길이를 기반으로 문장을 정렬하는 방법에서는 위의 <그림 1>과 같이 문장 정렬에 필요한 비용을 "길이"에 기반하여 계산한다. Gale and Church 알고리즘[10]으로 대표되는 길이 기반의 정렬 방법은 한 언어에서 긴 문장은 다른 언어에서 긴 문장과 대응한다는 가설을 가지고 비드의 확률을 원문과 번역문의 문장의 길이에 기반하여 문장 간의 유사도를 계산한다.[2] 이 알고리즘은 적어도 비슷한 언어이고 직역인 경우에는 문장을 제대로 정렬할 수 있다. Gale and Church 알고리즘에서 제안하는 문장 정렬의 가능성은 {0:1, 1:0, 1:1, 1:2, 2:1, 2:2}로 한정하며 거리 측정치 δ는 이들 비드에 대해 insert, delete, substitute, expand, contract, merge 연산을 하는데 필요한 비용으로 계산한다. 이는 주어진 비드 l1과 l2에서 각 언어의 문장들의 길이에 기반한 계산으로, 먼저 원시 언어의 각 문자는 대상 언어에서 랜덤 숫자로 나타난다고 가정하고, 이 랜덤 숫자가 독립적이고 동일하게 분포된다고 할 때 정규 분포로 모델링하여 평균 c와 편차 s_2를 구한다. 평균 c에 대해 [10]에서는 독일어/영어 및 불어/영어의 텍스트 길이를 비교하여 각각 1.1 및 1.06을 얻었지만 계산을 단순화하기위해 c=1로 가정하여 사용하며, 편차 s2는 독일어/영어 및 불어/영어에 대해 7.3 및 5.6을 얻었지만 역시 단순화하기위해 언어에 관계없이 s_2=6.8로 사용한다. 따라서, 거리 측정치 δ를 $(l_2 - l_1 c)\sqrt{l_1 s_2}$로 정의할 때 두 비드 간의 비용은 식 (1)과 같이 정의할 수 있다.

$$\mathrm{cost}(l_1, l_2) = -\log Prob(match|\delta(l_1, l_2, c, s_2)) \quad (1)$$

비용 계산에 log 값을 사용하여 비용 측정을 거리 측정으로 간주할 수 있게 된다. 다시 말해, 가장 높은 비율을 가지는 정렬이 가장 짧은 거리에 대응하기 때문에 위의 식에 따르면 정렬 비율을 단순히 비드 거리의 합으로 계산할 수 있다. $Prob(match|\delta)$는 베이즈 이론에 따라 $Prob(match)Prob(\delta|match)$로 계산된다. $Prob(match)$는 [10]에서 정의한 {0:1, 1:0, 1:1, 1:2, 2:1, 2:2}와 같은 비드 구성 방법에 따라 스위스 유니온 은행(USB) 코퍼스에서 얻었다.[3] 조건부 확률 $Prob(\delta|match)$는 $Prob(|\delta|)$이 표준화된(평균 0, 표준편차 1) 정규 분포를 가지는 랜덤 변수 z가 최소한 $|\delta|$와 같은 크기를 가진다고 할 때 $Prob(\delta|match) =$

2) Gale and Church 알고리즘에서의 문장의 길이는 단어수가 아닌 문자열의 길이이다. Gale and Church 알고리즘에서 단어수를 사용하지 않은 이유는 동일 어족의 비슷한 언어라도 단어 구성의 변이가 다르기 때문에(예를 들어, 영어-독일어) 단어수를 사용할 경우 오류 비율을 높을 수 있다. [9]는 단어(토큰)의 수로 길이를 계산한다.

3) $Prob(match)$는 {1:1, 1:0(또는 0:1), 2:1(또는 1:2), 2:2}에 대해 각각 0.89, 0.0099, 0.089, 0.0011이다.

$2\times(1-Prob(|\delta|))$으로 계산될 수 있다.

결론적으로 [10]의 방법은 최소한 동족 언어 간(예를 들어, 자신들의 논문에서 사용한 영어, 불어, 독일어)에서는 어느 정도 높은 성능을 보였다.[4] 이 알고리즘은 1:1 정렬에서 가장 높은 정확성을 보였지만 1:0이나 0:1 정렬은 정확한 정렬을 얻을 수 없었다.

그밖에도 길이 기반을 사용하는 다른 연구로는 문장의 길이가 아닌 단어의 수로 유사도를 계산하는 [9]가 있고, Gale and Church 알고리즘을 사용하여 영어와 광동어를 정렬한 [20]이 있다. [20]은 길이 기반의 단점을 보완하기위해 다음 절에서 설명할 어휘 정보를 사용하여 정확도를 높였다.

2.2. 어휘정보 사용

길이 기반 알고리즘은 특히 짧은 문장의 연속일 때 단어의 속성을 고려하지 않고 길이만 고려하기 때문에 문장 정렬에 있어서 정렬 오류가 발생할 확률이 높아진다. 또한 한쪽 코퍼스에서 문장의 삭제가 있다면 정렬 오류가 발생한 확률이 더욱 높아지며 길이 기반에서는 이를 복구할 방법이 전혀 없다. [9]에서는 이런 문제를 해결하기 위해 앵커(anchor)를 사용하여 코퍼스를 작은 단위로 나누지만 이러한 앵커의 선택 역시 정확한 어휘 정보에 기반하지 않는다.[5]

Chen 알고리즘[12]에서는 문장 정렬을 단어 대 단어 번역 모델을 구축한 다음 주어진 번역 모델이 생성한 코퍼스로 최대 유사도를 가지는 정렬을 선택한다. Chen의 모델은 기본적으로 [14]와 유사하나 비용을 계산하는데 있어서 길이를 사용하는 대신 "단순한" 번역 모델을 사용한다.[6] Chen은 단순함과 효율성을 위해 어순을 고려하지 않고 한단어가 둘 이상의 단어와 대응하는 가능성을 모두 무시한다. 따라서 가능한 단어 비드는 1:0, 0:1, 1:1로 한정된다. 이 번역 모델의 핵심은 어떤 단어가 특정 단어로 번역될 때 이들 단어의 1:1 단어 비드는 확률이 높고, 동일한 단어의 1:0이나 0:1 단어 비드의 확률 곱보다는 매우 높다는데 있다. Chen

---

4) 기본적으로 4%의 오류를 보였고, 최상의 결과를 얻을 수 있는 80%의 코퍼스에서는 오직 0.7%의 오류만을 보였다.

5) [9]에서는 코퍼스의 *Author = Mr. Speaker* 및 *Author = M. le Président*와 같은 *Author* 주석 부분을 앵커로 정의했지만 번역 문장들의 특성상 이들 앵커가 정확하게 동일한 숫자로 출현하지 않는 경우가 빈번하다. 예를 들어, 유럽 의회 코퍼스[21]에서는 *Resumption of the session* 및 *Reprise de la session*과 같이 회의 시작을 알리는 문장을 앵커로 삼을 수 있지만 정확하게 동일한 숫자로 코퍼스에 출현하지 않는다(영어 122번, 불어 126번 출현). 따라서 이런 유의 앵커는 어휘 앵커가 되기 힘들다.

6) [8]에서는 [12]와 같이 현재 통계적 기반의 기계 번역에서 사용하고 있는 IBM 번역 모델 [14]를 사용하지 않고 보다 단순한 번역 모델을 사용한다. 번역 모델에 대한 논의는 이 논문의 범위를 벗어나기 때문에 여기에서는 자세히 논하지 않는다.

은 문장 정렬을 위해서 모든 가능한 단어 비드의 합을 사용하지 않고 최상의 값만을 선택해서 사용한다. 사실 최상의 단어 비드를 찾기 위해 Chen의 알고리즘은 추정할 수 있는 정렬의 1:0과 0:1 비드로 시작하여 greedy하게 1:0과 0:1 비드를 1:1 비드와 교체하여 더 이상 개선되지 않을 때까지 정렬 확률을 계산한다. Chen의 번역 모델에서 파라미터는 비터비(Viterbi) 버전의 EM 알고리즘을 사용하여 계산하고 수동으로 정렬한 초기 100 문장을 사용하여 번역 모델을 부트스트래핑한다. 그런 다음, 점진적 버전의 EM 알고리즘을 통해 20,000 문장에 대해 다시 파라미터를 계산한다. 번역 모델은 최종적으로 한 번의 패스를 통해 코퍼스를 정렬하게 된다.

[8]에서는 캐나다 의회 기록 코퍼스, 유럽 의회 기록 코퍼스의 수백만 문장에 대해 정렬을 시도하여 오류 비율이 전체 코퍼스에서 0.4%만 얻었다고 보고했다. 대부분의 정렬 오류는 잘못된 문장 경계 인식에서 왔기 때문에 자신이 사용한 알고리즘은 견고하다고 주장했고 IBM 번역 모델과 같은 복잡한 번역 모델을 사용하는 경우에는 문장 정렬 과정이 상당히 느려짐에도 불구하고 정렬 결과의 개선은 없을 것으로 평가했다. 또한, 문장 정렬 비드를 1:0, 0:1, 1:1, 2:1, 1:2로 한정하여 다른 정렬 비드의 경우를 고려하지 않았다. 하지만 Chen 방법을 다른 정렬 비드로 확장하는 것은 어렵지 않겠지만 [10]의 예에서와 같이 복잡한 문장 정렬 비드는 오류를 발생할 가능성이 많으리라 예상된다.

그밖에 어휘 정보를 사용하는 정렬 방법으로는 동계어(同系語) 정보를 사용하는 [7]이 있고, 어휘 대응을 통해 영어와 일본어를 정렬한 [22] 등이 있다.

2.3. 하이브리드 방법

최근에 발표된 문장 정렬 방법들은 이전의 정렬 방법들을 결합한 하이브리드 방법을 사용한다. 대표적인 예가 [12]이나 [13]으로서 문장 정렬을 위해 문장의 길이와 어휘 정보를 모두 사용해 문장의 유사도를 계산한다. [12]에서는 문장의 길이에 기반하여 문장의 유사도를 계산한 초기 정렬 결과를 바탕으로 IBM 번역 모델 1[14]로 학습하고 이를 사용하여 원시 문장을 번역하여 대상 문장과의 유사도를 계산해 병렬 코퍼스를 구축한다. [13]에서는 기존의 대역 사전을 사용하여 단어 간의 부분 번역을 한 다음, 문장의 길이와 번역된 단어(토큰)간의 유사도를 계산하여 문장을 정렬한다.

[13]과 같이 기존에 존재하는 대역 사전을 사용하여 단어 간의 번역 후 문장 간의 유사도를 계산하는 경우에는 단어 모호성이 있는 대상 문장에서 빈도수가 가장 높은 단어를 선택하기 때문에 정확한 단어 번역에 기반하여 문장의 유사도를 계산한다고 볼 수가 없다. 반면, [12]와 같이 IBM 번역 모델 1을 사용하는 경우에는 기존의 다른 정보나 자료가 없이도 이 방법을 사용할 수 있는 장점이 있

으나 [13]에서도 비판된 것처럼 병렬 코퍼스에서 길이에 기반하여 일차적으로 문장을 유사도를 계산한 결과를 번역 모델 학습에 사용하기 때문에 문장이 정확하게 정렬되었다는 보장이 없고 따라서 자동으로 구축된 번역 모델의 정확성까지 의심받게 된다. 또한 [12]는 자신의 논문에서는 빠르고 정확한 문장 정렬(fast and accurate sentence alignment)이라고 주장하지만 IBM 번역 모델로 학습한 다음 원시 코퍼스를 번역을 하여 문장 정렬을 하고 이를 계속해서 반복하기 때문에 결코 빠르다고 할 수 없다.

본 연구에서는 본 장에서 설명한 기존의 방법들의 단점을 보완하기 위해 기존의 영한 번역기를 사용하여 단순한 단어 수준의 번역이나 번역 모델을 새로이 구축하는 어휘 정보 수준의 내용을 사용하는 것이 아니라 원시 코퍼스의 언어(본 논문에서는 한국어)를 대상 코퍼스의 언어(영어)로 완전히 번역한 다음 길이 기반 및 단어 기반의 유사도를 계산하여 문장을 정렬한다. 다음 장에서는 본 논문에서 제안하는 방법을 구체적으로 설명하기로 한다.

3. 제안 방법

이 장에서는 원시 언어인 한국어를 번역기를 사용하여 대상 언어인 영어로 번역한 다음에 길이 기반을 통해 문장의 유사도를 계산하고 번역된 단어(토큰)를 비교하여 최종적인 정렬 문장을 생성하는 알고리즘에 대해 설명한다.

Gale and Church 알고리즘[10]과 같이 문장의 길이를 비교하거나 Kay and Roscheisen 알고리즘[11]과 같이 동계어를 사용하여 문장을 정렬하는 경우에는 같은 어족의 언어끼리는 어느 정도의 만족할 만한 결과를 얻을 수 있지만 한국어, 영어와 같이 단어나 문장의 구성 및 성립 과정이 전혀 다른 경우에는 이들 방법을 직접 적용하는 것은 문제가 된다.

본 연구에서는 한국어를 문장 정렬에 직접 사용하지 않고 한국어의 영어 번역문을 영어와 비교해 정렬하는 방법을 사용한다. [13]에서 제안하는 시스템을 기본 시스템으로 차용하지만, 이전의 연구에서처럼 단순한 단어 수준의 번역이나 일부 트레이닝 코퍼스에서 학습한 번역 모델을 통한 신뢰할 수 없는 방법의 번역 모델을 사용하는 것이 아니라 기존의 규칙 기반 영한 번역기를 사용하여 한국어의 영어 번역문을 얻은 다음 이를 대상 언어인 영어 원문과 비교하여 정렬한다. 문장에서의 몇몇 단어 수준의 번역으로 정렬을 시도하는 [13]과는 달리, 본 논문에서 제안하는 방법은 영어 번역문-영어 원문의 정렬이 되기 때문에 [13]에서 사용하는 길이에 기반한 알고리즘이나 동계어를 사용하는 알고리즘을 사용해도 한국어-영어 또는 단어 수준의 번역-영어 원문[7]의 정렬보다는 좋은 결과를 얻을 수 있다. 또한

7) 단어 수준의 번역은 한국어 문장의 원래 틀을 그래도 유지한다. 예를 들어, "우리 나라

수차례의 반복을 통해 정렬된 결과를 다시 정렬하여 향상된 정렬 결과를 얻고자 한다. [13]의 시스템도 반복 정렬을 하지만, 본 논문에서는 반복 정렬한 다음, 두 개 이상의 문장이 하나의 문장과 정렬되는 경우에는 가중치를 두어 반복 정렬을 한다. 반복적 정렬 실험에 대한 결과는 본 장의 뒷부분에서 제시하고 논의하도록 한다.

3.1 한국어-영어 병렬 코퍼스

서론에서도 기술한 바와 같이 통계적 방법을 사용하는 기계 번역 연구에 사용할 만큼의 충분히 한국어가 포함된 병렬 코퍼스는 현재까지 거의 없다고 해도 과언이 아니다. 따라서 본 연구에서는 한국어와 영어를 동시에 제공하는 신문이나 잡지의 기사, 독해 공부를 위한 대역 문서, 번역된 기술 문서 등을 웹에서 찾아 문장을 자동으로 정렬하여 한영 병렬 코퍼스를 구축한다. 웹에서 대역 문서를 자동으로 찾아 병렬 코퍼스를 구축하는 것은 새로운 것이 아니며 [15]에서 시작하여 [7] 및 [16]에서 충분한 양의 병렬 코퍼스를 웹에서 찾아 자동으로 구축할 수 있음을 보였다.

웹에서 찾을 수 있는 한국어와 영어 대역 문서로는 신문 기사의 경우에는 미국 통신사의 기사의 한글 번역문을 제공하거나 한글 기사를 영문으로 번역 제공하는 것이 있다. 영어 독해 공부를 위한 영어 본문과 이에 대한 해석을 같이 기재하는 사이트들을 찾을 수 있다. 최근에는 오픈 문서 프로젝트를 통해 오픈 소스 프로그램의 문서나 기술 백서들을 한글로 번역한 사이트도 많이 찾아 볼 수 있고 컴퓨터 프로그램의 도움말이나 기술 문서 등을 번역한 사이트들도 많다. 또한 웹뿐만 아니라 지역화된 소프트웨어에서 제공하는 번역된 도움말들도 어렵지 않게 찾을 수 있다. 이들 기술 문서나 소프트웨어의 도움말은 대체적으로 직역되어 있어 이 논문의 목적에 부합하며 무엇보다도 상당히 많은 양의 문서를 쉽게 찾을 수 있어 해당 분야의 문서를 통계적 방법에 기반한 기계 번역에 적용할 때 좋은 결과를 얻을 수 있을 것으로 예상된다.

그밖에 웹에서 찾을 수 있는 대역 문서로는 종교 관련 서적도 쉽게 찾을 수 있지만 앞에서 언급한 것처럼 주로 의역을 사용하기 때문에 기계 번역과 같은 자동 처리에서 유용한 결과를 얻을 수 없다는 결론 하에 본 논문에서는 사용하지 않기로 한다. 또한, 영화의 영어/한글 통합 자막 같은 자료도 쉽게 웹에서 찾을 수 있지만 한글로 번역된 영화 자막은 구어나 통신어 사용이 많고 오타나 의역도 적지 않아 자동 처리에 사용할 만한 품질을 찾기 힘들다.

만세를 외쳤다"의 경우 "we nation 만세를 외쳤다"로 번역된다. 이는 [13]의 연구에서 형태소 분석을 고려하지 않기 때문에 단순히 2개 국어 사전만으로 대응 단어를 치환하기 때문이다.

위와 같은 방법으로 수집된 한국어 영어 병렬 코퍼스의 크기는 <표 1>에 정리한 바와 같다. <표 1>에서는 본 논문에서 찾은 코퍼스의 페이지(문서) 수, 한국어 및 영어의 어절/단어 수, 전체 파일 크기 등을 제공한다. <표 1>에서 코퍼스의 페이지수는 한국어/영어 문서 쌍을 하나의 페이지로 계산하며, 파일 크기는 한국어/영어 페이지 크기 모두를 합산한 결과이다. 기술 백서나 소프트웨어 도움말의 경우 프로그래밍 코드와 같은 원시 언어와 대상 언어에 동일한 내용이 있을 때는 문장 정렬 후에 이들을 삭제하며, 1:0이나 0:1로 정렬된 문장 비드의 경우에도 병렬 코퍼스로서의 효용을 찾을 수 없기 때문에 삭제한다. 따라서 최종적으로 얻어지는 병렬 코퍼스의 크기는 달라질 수 있다.

<표 1> 수집된 한국어 영어 병렬 코퍼스의 크기

코퍼스	페이지수	한글 어절수	영어 단어수	파일 크기 (KB)
미국 통신사 기사 번역	4,922	812,621	1,289,566	39,972
한글 기사 영문 번역	2,504	657,098	1,053,064	20,644
영어 독해 공부를 위한 영한 대역	2,342	198,042	292,260	18,840
기술 백서 및 기술 문서	8,386	4,525,442	5,501,780	122,288
컴퓨터 소프트웨어 도움말	16,182	2,790,609	3,856,848	147,244
총 계	34,336	8,983,812	11,993,518	348,988

3.2. 코퍼스 분할

문장 정렬 알고리즘은 동적 프로그래밍 기법을 사용하기 때문에 두 문장 길이의 곱에 해당하는 시간이 필요하다. 따라서 큰 코퍼스를 한 번에 정렬하는 것보다 가능한 작은 부분으로 나누어서 정렬하는 것이 훨씬 시간을 절약할 수 있다. 본 논문에서는 다른 연구들과 달리 수집된 코퍼스를 하나의 거대한 코퍼스로 합치지 않고 수집된 페이지 단위로 정렬을 시도한다. 정렬 시간만 따진다면 수 천 또는 수 만개의 비교적 작은 코퍼스를 정렬하기 때문에 하나의 거대한 코퍼스를 정렬하는 것보다 빠른 정렬을 할 수 있고 큰 코퍼스를 한 번에 정렬할 때 발생할 수 있는 메모리 문제도 해결할 수 있다.

3.3. 전처리기로서의 번역기 사용

앞에서 설명한 문장 정렬 알고리즘들은 이미 문장 경계가 구분되었다고 가정하고 문장 간의 유사도를 계산하기 때문에 이들 알고리즘을 적용하기 전에 문장의 경계를 구분해야 한다.[8] 웹에서 수집한 문서들의 특성상 문장이나 단락의 경계

가 표시되어 있지 않기 때문에 본 연구에서는 따로 문장 경계를 위한 전처리 과정을 거치지 않고는 문장 정렬을 수행할 수가 없다. 기존의 규칙 기반 한영 번역기를 사용하여 한국어의 영어 번역문을 얻을 뿐만 아니라 원시 언어 및 대상 언어의 문장 경계까지 구분한다. 원시 언어인 한국어는 문장 경계 구분 및 대상 언어인 영어로 번역하는 과정을 거치고 대상 언어인 영어는 번역 과정을 거치지 않고 번역기의 문장 경계 기능만 사용한다.

3.4. 문장 수준의 정렬

한국어를 번역한 영어와 영어 원문을 문장 수준에서 정렬을 시도한다. 본 논문에서 사용하는 문장 정렬 방법은 [13]의 방법을 차용한다. [13]과 다른 점이 있다면 대역 사전을 사용한 단어 수준의 번역을 원시 언어로 대상 언어와 정렬하는 것이 아니라 전체 번역을 사용하여 번역된 원시 언어를 대상 언어와 정렬한다.

먼저, 문장 간의 유사도 측정은 단어(토큰) 단위와 길이 단위로 나눌 수 있는데 단어 단위의 경우에는 원시 및 대상 문장에서의 동일 단어의 상대적 개수를 사용하여 유사도를 측정하고, 길이 단위의 경우에는 가장 긴 문장에서 가장 짧은 문장 사이의 비율로 유사도를 계산한다. 문장 사이의 유사도는 정렬 매트릭스에서 모든 문장 쌍에 대해 대각선 모양을 형성하도록 하여 1 대 다수의 문장 정렬이 가능케 하고 특정 문장이 너무 멀리 떨어진 문장과의 정렬을 시도하지 않도록 한다.

적절한 문장 쌍으로 유사도 매트릭스가 얻어지면 동적 프로그래밍을 통해 정렬 최적화를 수행한다. 정렬 최적화는 매트릭스에 특정 문장을 정렬에서 배제할 것인지 여러 문장으로 합칠 것인지를 결정하는 페널티를 할당한다. 문장 배제에 대한 스코어는 고정된 파라미터로 트레이닝 코퍼스에서 학습되며 문장 연결은 식 (2)와 같이 연결할 두 문장의 토큰 기반 스코어(identityScore)와 길이 기반 스코어(galeScore) 합을 사용한다.

$$\mathrm{cost}(l_1,l_2) = identityScore(l_1,l_2) + galeScore(l_1,l_2) \quad (2)$$

또한, [13]에서와 같이 성능 문제로 한 번에 두 문장 이상이 한 문장과 결합하는 것을 고려치 않는다. 이렇게 정렬 최적화가 끝나면 다시 이 과정을 반복적으로 수행해 새로운 결과(두 문장이 합쳐진 결과)가 길이 기반 비율에서 보다 높은 스코어를 가지는 경우에는 이웃하는 문장쌍들과 결합을 반복한다. 이런 방법을 통해 1 대 다수의 문장 정렬을 찾아낼 수 있다.

---

8) 앞에서 설명한 이전 연구들에서는 문장 정렬에 필요한 문장 경계의 구분에 대해 전혀 언급이 없었다.

3.5. 반복 정렬

본 연구에서 제안된 문장 수준의 정렬에서는 문장 쌍들을 반복적으로 결합하여 유사도를 계산하기 때문에 이미 이전 연구에서 한정하던 {0:1, 1:0, 1:1, 1:2, 2:1, 2:2} 비드를 넘어 다수 대 다수까지 정렬이 가능하다. 이제 두 개 이상의 문장이 합쳐졌을 때 일정한 가중치를 부여하여 다시 반복적으로 정렬을 시도한다. 이렇게 반복적으로 정렬을 시도하는 이유는 앞에서 수행했던 문장 수준의 정렬에서 1:0이나 0:1로 정렬된 결과가 오류였을 경우([10]의 결과에서 1:0이나 0:1 정렬은 모두 오류를 보였다.) 이를 해결해보기 위함이다. 두 문장 이상이 하나로 합쳐져 정렬된 경우에는 반복 정렬 동안은 이전 결과를 그대로 사용하여 정렬을 하기 때문에 합쳐진 문장이 다시 분리되지는 않는다. <그림 2>는 반복 정렬 과정의 예를 보여 준다.

<그림 2>의 주기 1에서는 원시 및 대상 언어의 텍스트가 {1:2, 1:3, 0:1, 1:2, 1:2}로 문장이 정렬되었다.[9] 이제 이렇게 정렬된 결과에 두 문장 이상 합쳐진 경우에는 가중치(본 연구에서는 가중치를 5×(결합된 문장의 수)를 해당 문장 길이에 더하여 계산했다.)를 부여하여 다시 정렬한다. 이렇게 가중치를 사용하여 문장 수준의 정렬을 다시 실행할 경우, 주기 2에서는 {1:5, 1:1, 1:2, 1:2}로 문장이 정렬되었다. 주기 1의 처음 두 정렬 비드가 합쳐지면서 두번째 원시 텍스트의 문장은 0:1 비드에 포함되어 주기 1의 {1:2, 1:3, 0:1} 비드가 주기 2에서는 {1:5, 1:1}로 변경되었다. 이론상으로 프로그램은 문장 비드의 변경이 없을 때까지 반복해서 정렬을 수행한다. 본 연구에서는 실험적으로 주기 5까지 수행하여 결과를 분석한 결과(주기 5 이후로는 결과의 변화가 없었다), 주기 3 이상에서는 정렬 결과에는 아무런 변화가 없었고 문장 정렬 유사도만 일부 변경되었다. 따라서 본 연구는 실제 반복 정렬의 주기를 2로 제한하여 수행한다.

이제 두 번의 주기에 걸친 반복 정렬 결과에서 최상의 정렬 결과를 선택해야 한다. 주기 2의 결과에 대해 confirmed alignment (CA)와 doubtable alignment (DA)로 나눌 수 있다. CA는 주기 2의 정렬 3번과 4번처럼 주기 1과 동일한 정렬 결과를 가지지만 문장 유사도가 주기 1보다 높다. 따라서 CA는 최종 정렬 결과로 확정한다. DA는 주기 2의 정렬 1번과 2번처럼 주기 1과 결과가 다른 경우를 지칭한다. 주기 2의 정렬 2번은 DA이지만 이전 주기보다 문장 유사도가 높기 때문에 최종 정렬 결과로 확정한다. 주기 2의 정렬 1번의 경우에는 DA이면서 이전 주기보다 문장 유사도가 낮기 때문에 최종 정렬 결과로 확정할 수 없다. 이런 경우에는 이전 주기(주기 1)의 결과로 돌아간다. 위의 설명은 아래와 같이 정리할 수 있다. 따라서 <그림 2>에 대한 최종 정렬 결과는 {1:2, 1:1, 1:2, 1:2}로 <그림 3>에서 제시한다.[10] 이것을 요약하면 다음과 같다.

9) 문장의 끝에 있는 숫자는 문장 정렬 유사도이다.

주기 1:

1. 전설적인 자동차 제작자 보이드 카딩턴은 "차를 보면 상태가 엉망진창이라는 걸 손수 말해주고 있다"며 원래 계획대로 시동을 걸지 못해 안타까워했다. "I'll tell you what, she's a mess. &&& Look at her," said legendary car builder Boyd Coddington, who was unable to start the car as planned. 0.44452

2. 차의 내부 수납공간과 트렁크에서는 여자 핸드백, 머리핀, 립스틱, 20L의 석유, 슬리츠 맥주 등 오클라호마주 50주년을 기리기 위해 담아뒀던 물품들이 있었다. -- the trunk, organizers pulled out some of the objects buried to celebrate Oklahoma's 50 years of statehood &&& -- &&& a 5-gallon can of leaded gasoline, which went for 24 cents a gallon in those days, and rusted cans of Schlitz beer. 0.58288

3. The contents of a "typical" woman's handbag, including 14 bobby pins, lipstick and a bottle of tranquilizers, were supposed to be in the glove box, but all that was found looked like a lump of rotted leather. -0.3

4. 차외에 다른 타임캡슐에 묻혔던 보관함에는 전혀 색이 바래지 않은 성조기와 각종 역사 문서 및 지도가 발견됐다. A separate time capsule buried with the car was opened, and organizers removed and unfolded an unfaded American flag. &&& Other historical documents, aerial maps of the city and postcards were in good condition. 0.391071

5. 이번 행사에 참석한 밥 페트리는 "자동차가 녹이 슬던 안 슬던 상관없다"며 "1957년에 타임캡슐을 만든 생각 자체가 훌륭한 것이고 그들이 생각했던 삶을 우리가 지금 살고있는 것이 중요한 것"이라고 말했다. "We don't care what condition it's in," said Bob Petri. &&& "It's just the whole idea somebody thought of it in 1957 and here we are living it." 1.11449

주기 2:

1. 전설적인 자동차 제작자 보이드 카딩턴은 "차를 보면 상태가 엉망진창이라는 걸 손수 말해주고 있다"며 원래 계획대로 시동을 걸지 못해 안타까워했다. "I'll tell you what, she's a mess. &&& Look at her," said legendary car builder Boyd Coddington,

10) 어떤 의미에서 <그림 2>는 반복 정렬의 효용을 설명하는데 좋은 예는 아니다. 실제 반복 정렬 후의 프로그램 결과에서는 대상 코퍼스의 "- *the trunk, organizers pulled out some of the objects buried to celebrate Oklahoma's 50 years of statehood &&& -- a 5-gallon can of leaded gasoline, which went for 24 cents a gallon in those days, and rusted cans of Schlitz beer.*"이 주기 2에서 주기 1의 결과를 복원하는 과정에서 삭제되었다. 정확하게 정렬되려면 주기 1에서 정렬 2번과 3번이 합쳐져야 한다(1:4). 하지만 문장 수준의 정렬 결과보다 반복 정렬을 수행한 이후에 보다 좋은 결과를 얻었다. (자세한 내용은 4장 참조)

who was unable to start the car as planned. &&& -- the trunk, organizers pulled out some of the objects buried to celebrate Oklahoma's 50 years of statehood &&& -- &&& a 5-gallon can of leaded gasoline, which went for 24 cents a gallon in those days, and rusted cans of Schlitz beer. 0.403017

2. 차의 내부 수납공간과 트렁크에서는 여자 핸드백, 머리핀, 립스틱, 20L의 석유, 슐리츠 맥주 등 오클라호마주 50주년을 기리기 위해 담아뒀던 물품들이 있었다. The contents of a "typical" woman's handbag, including 14 bobby pins, lipstick and a bottle of tranquilizers, were supposed to be in the glove box, but all that was found looked like a lump of rotted leather. 0.905567

3. 차외에 다른 타임캡슐에 묻혔던 보관함에는 전혀 색이 바래지 않은 성조기와 각종 역사 문서 및 지도가 발견됐다. A separate time capsule buried with the car was opened, and organizers removed and unfolded an unfaded American flag. &&& Other historical documents, aerial maps of the city and postcards were in good condition. 0.41875

4. 이번 행사에 참석한 밥 페트리는 "자동차가 녹이 슬던 안 슬던 상관없다"며 "1957년에 타임캡슐을 만든 생각자체가 훌륭한 것이고 그들이 생각했던 삶을 우리가 지금 살고있는 것이 중요한 것"이라고 말했다. "We don't care what condition it's in," said Bob Petri. &&& "It's just the whole idea somebody thought of it in 1957 and here we are living it." 1.93326

<그림 2> 반복 정렬의 예

1) 이전 주기와 문장 정렬 결과가 같으면 CA로 확정한다.
2) 문장 정렬 결과가 다르고, 문장 유사도가 이전 주기보다 높으면 DA이지만 CA로 확정한다.
3) 문장 정렬 결과가 다르고, 문장 유사도가 이전 주기보다 낮으면 DA이며, 이전 주기 결과로 돌아가 CA로 확정한다.[11]

1. 전설적인 자동차 제작자 보이드 카딩턴은 "차를 보면 상태가 엉망진창이라는 걸 손수 말해주고 있다"며 원래 계획대로 시동을 걸지 못해 안타까워했다. "I'll tell you what, she's a mess. &&& Look at her," said legendary car builder Boyd Coddington, who was unable to start the car as planned. 0.44452

11) 문장 유사도는 원시 언어를 기준으로 비교한다.

2. 차의 내부 수납공간과 트렁크에서는 여자 핸드백, 머리핀, 립스틱, 20L의 석유, 슬리츠 맥주 등 오클라호마주 50주년을 기리기 위해 담아뒀던 물품들이 있었다. The contents of a "typical" woman's handbag, including 14 bobby pins, lipstick and a bottle of tranquilizers, were supposed to be in the glove box, but all that was found looked like a lump of rotted leather. 0.905567

3. 차외에 다른 타임캡슐에 묻혔던 보관함에는 전혀 색이 바래지 않은 성조기와 각종 역사 문서 및 지도가 발견됐다. A separate time capsule buried with the car was opened, and organizers removed and unfolded an unfaded American flag. &&& Other historical documents, aerial maps of the city and postcards were in good condition. 0.41875

4. 이번 행사에 참석한 밥 페트리는 "자동차가 녹이 슬던 안 슬던 상관없다"며 "1957년에 타임캡슐을 만든 생각자체가 훌륭한 것이고 그들이 생각했던 삶을 우리가 지금 살고 있는 것이 중요한 것"이라고 말했다. "We don't care what condition it's in," said Bob Petri. &&& "It's just the whole idea somebody thought of it in 1957 and here we are living it." 1.93326

<그림 3> <그림 2>의 최종 정렬

4. 실험 및 평가

본 장에서는 정렬 결과를 보이고 이를 바탕으로 평가를 한다. 일반적으로 정렬 결과를 평가하려면 비교할 수 있는 정렬된 코퍼스(일반적으로 사람이 수동으로 직접 정렬한 코퍼스)가 있어야 한다. 하지만 본 논문에서는 비교할 수 있는 정렬된 코퍼스가 없기 때문에 다른 연구에서 사용한 정렬 방법의 결과를 statistical machine translation (SMT)에 적용하여 결과를 비교하는 방법으로 평가한다.

4.1. 정렬 결과 및 비교

<그림 4>는 [13]에서 사용한 정렬 결과(A) 및 본 연구에서 제안하는 정렬 결과(B)를 비교한다. [13]의 정렬 결과에서 0:1로 정렬된 두 개의 문장(A2 및 A3)은 모두 잘못 정렬된 예이다. 이에 반해 이 논문에서 제안하는 방법은 이를 모두 정확하게 정렬하였다(B2 및 B3). 또한 A6은 "병력 수를 15,000 명으로 재조정 했다"라는 부분에 대한 정렬에 실패했지만 B4의 경우에는 1:2로 정확하게 정렬했다. 또한 B3와 같이 정확하게 정렬했지만 필요없는 부분까지 정렬에 포함하는 오류를 보였다. 반대로 동일한 문장에 대해 A5에서는 원시 언어의 내용을 전혀 포함하지 못했

A
1. 정부 통계에 따르면 2004년 18,528명이 살해됐다. According to government statistics, there were 18,528 murders in 2004. 0.203759 2. Ø more than 50 per day. -0.3 3. Ø Top government and police officials repeatedly insist that the crime rate is coming down. -0.3 4. 이는 하루에 50명 이상이 사망하는 수치다. But this is met with widespread skepticism. 0.189474 5. 최고 정부 경찰 관계자들은 범죄율이 떨어지고 있다고 주장하지만 많은 사람들은 아직 이에 대해 회의적인 입장을 보이고 있다. Copyright 2007 The Associated Press. &&& All rights reserved &&& This material may not be published, broadcast, rewritten, or redistributed. -0.180233 6. 보고서는 28,000명의 병력이 20,000명의 전투 병력에 추가로 포함될 것이라고 밝혔었지만 병력 수를 15,000명으로 재조정했다. It estimates that under past proportions, 28,000 support troops would be added to the 20,000 combat troops. 0.447464
B
1. 정부 통계에 따르면 2004년 18,528명이 살해됐다. According to government statistics, there were 18,528 murders in 2004. 4.49016 2. 이는 하루에 50명 이상이 사망하는 수치다. more than 50 per day. 2.98333 3. 최고 정부 경찰 관계자들은 범죄율이 떨어지고 있다고 주장하지만 많은 사람들은 아직 이에 대해 회의적인 입장을 보이고 있다. Top government and police officials repeatedly insist that the crime rate is coming down. &&& But this is met with widespread skepticism. &&& Copyright 2007 The Associated Press. &&& All rights reserved. 0.0944805 4. 보고서는 28,000명의 병력이 20,000명의 전투 병력에 추가로 포함될 것이라고 밝혔었지만 병력 수를 15,000명으로 재조정했다. It estimates that under past proportions, 28,000 support troops would be added to the 20,000 combat troops. &&& But it revises that figure to 15,000 support troops for a new deployment. 0.24692

<그림 4> 정렬 결과 비교

다. <표 2>는 본 논문에서 수집한 병렬 코퍼스에 대한 정렬 결과를 정리한 것이다. <표 2>에서는 정렬 결과가 0:1 또는 1:0으로 정렬되는 경우, 1:1로 정렬되는 경우 및 그 외의 경우인 n:m(n, m은 2 이상)으로 정렬되는 경우로 나누어 각각의 코퍼스에 대한 수치를 표시한다.[12]

<표 2> 정렬 결과

코퍼스	1:0 or 0:1	1:1	그 외	총합
미국 통신사 기사 번역	6,034	40,837 (71,56%)	10,195	57,066
한글 기사 영문 번역	8,982	50,201 (78,97%)	4,388	63,571
영어 독해 공부를 위한 영한 대역	137	10,569 (88,90%)	1,182	11,888
기술 백서 및 기술 문서	48,054	357,217 (82.54%)	27,525	432,796
컴퓨터 소프트웨어 도움말	23,430	215,922 (76.02%)	44,688	284,040

4.2. SMT 시스템에의 적용

일반적으로 정렬 결과를 평가할 때에는 수동으로 정렬한 결과와 비교하여 정렬 결과의 오류 비율을 측정하는 방법을 사용한다. 아직까지 한국어-영어의 정렬 결과를 비교할 수 있는 성능 비교 집단이 없기 때문에 이 절에서는 정렬 결과를 통계적 방법의 기계 번역(SMT) 시스템에 직접 적용한다.[13]

SMT에 적용하기 위해 미국 통신사 영문 기사 코퍼스를 사용하여 [13]에서 사용한 정렬 결과(A) 및 본 논문에서 제안하는 정렬 결과(B)로 SMT를 위한 기본 시스템을 사용하여 한국어를 영어로 번역한 결과를 비교하는 방법을 사용한다. SMT를 통해 번역 결과가 향상된다면 정렬 결과가 낫다고 가정하고 정렬 결과를 평가한다. MT 기본 시스템은 오픈 소스는 모시스(Moses) 시스템으로 병렬 코퍼스를 사용하여 SMT 시스템을 구축할 수 있게 한다.[14]

12) <표 2>에서는 문장 간의 유사도에 대해 임계값(threshold)을 설정하지 않고 정렬되는 모든 경우에 대한 수치를 표시한다. 보다 높은 신뢰도를 가지는 병렬 코퍼스를 구축하기위해서는 역치를 설정하여 기대에 못 미치는 유사도를 가지는 문장 정렬 쌍은 삭제할 수 있다.

13) 정렬 결과를 비교할 수 있는 비교 집단의 구축은 이후 과제로 남겨 두기로 한다.

14) SMT 기본 시스템과 함께 유로 의회 코퍼스를 훈련 집단, 검증 집단, 성능 평가 집단 및 성능 비교 집단으로 분리하여 배포한다[25].

정렬 결과 A 및 B에서 중복되는 병렬 문장 38,523개는 올바르게 정렬되었다고 가정하여 이들 중에서 500 문장을 검증 집단, 다른 500 문장을 성능 평가 집단으로 추출하고 남은 병렬 문장인 37,523개를 훈련 집단으로 사용하여 SMT 시스템에 적용한다. <표 3>은 이들 정렬 결과 A 및 B를 기본 SMT 시스템에 적용하여 한국어를 영어로 번역한 결과에 대한 수치이다. 번역 결과에 대한 평가는 NIST 및 BLEU 기계 번역 자동 평가 도구를 사용했다[26].[15]

<표 3> A 및 B를 사용한 SMT 실험 결과

정렬 방법	NIST	BLEU
A	1.0227	0.0152
B	1.1192	0.0153

4.3. 평가

4.1절에서 보인 정렬 결과에서는 이전 연구보다 본 연구에서 제안하는 정렬 방법을 사용하는 경우에 나은 결과를 보인다는 것을 알 수 있었다. 단순히 정렬 결과를 비교하는 것보다 최근에 병렬 코퍼스가 가장 많이 사용되고 요구되는 SMT를 통해 이 논문에서 제안하는 방법과 이전 연구의 정렬 방법을 번역 결과로 비교 했다. 이 경우에도 본 연구에서 제안하는 방법으로 정렬한 코퍼스가 NIST 및 BLEU 스코어에서 모두 근소한 차이로 향상된 결과를 얻었다.

정렬 결과 A 및 B를 SMT 시스템에 적용할 결과는 MT 시스템의 성능만으로 판단한다면 결코 만족할 만한 결과는 아니다. 이는 훈련 집단의 코퍼스 양에 기인하며 이 문제는 SMT의 특징상 코퍼스 양이 증가하면 어느 정도 해결되리라 생각한다.[16]

15) NIST[23]와 BLEU[24]는 기계 번역의 품질을 판단하기 위해 평가 비교 집단과 비교하여 수치적으로 계산하는 평가 메트릭이다.

16) 코퍼스 크기가 훨씬 큰 컴퓨터 관련 기술 문서를 SMT를 사용한 실험에 적용하지 않은 이유는 기술 문서의 특징 상 프로그래밍 코드를 포함하고 있는데 이들 부분을 자동으로 인식하여 대상 언어로 번역되지 않아야 하는 문제를 해결해야 한다. 또한 일반적으로 지역화된 소프트웨어에서 제공되는 번역된 도움말은 자동 번역된 경우가 있어 오히려 SMT의 학습 데이터로 사용하기에 성능을 떨어트리는 경우도 있을 수 있다. SMT의 번역 결과는 훈련 집단의 양 뿐만 아니라 같은 크기의 코퍼스라도 훈련 집단의 선택에서도 크게 좌우된다. 따라서 훈련 집단의 선택 및 기준 설정은 앞으로의 SMT에서 지속적으로 연구해야 할 부분이다.

5. 결 론

본 논문에서는 한국어와 영어를 동시에 제공하는 신문이나 잡지 기사, 독해 공부를 위한 대역 문서, 영문을 한글로 번역하여 제공하는 기술 문서 등을 웹에서 찾아 이들을 자동으로 정렬하여 한영 병렬 코퍼스를 구축했다. 자동으로 문장을 정렬하는 방법으로 길이 기반 방법과 문장 번역을 사용하는 방법을 결합하여 하이브리드 방법을 사용하며, 기존의 문장 정렬 방법에서 사용하는 부분적 단어 번역에 의존하지 않고 번역기를 사용한 전체 번역을 통해 정렬 알고리즘을 적용하여 보다 신뢰할 수 있는 결과를 얻었다. 본 논문에서는 단순히 문장 수준에서의 정렬에 머무르지 않고 이를 다시 반복적으로 정렬을 시도해 이전 연구에서 가장 큰 문제점이였던 1:0이나 0:1 비드에서 발생할 수 있는 오류를 해결하려 노력했다.

또한, 본 논문에서 제안하는 정렬 방법을 사용하여 보다 정확하게 정렬된 문장을 가지는 병렬 코퍼스로 SMT 시스템에 적용하여 향상된 결과를 얻을 수 있는 가능성을 본 연구에서는 제시했다. 본 논문에서 구축된 한국어 영어 병렬 코퍼스가 앞으로의 통계적 방법을 사용하는 한국어 영어 기계 번역 연구에 사용될 수 있기를 바란다.

참 고 문 헌

[1] http://www.parl.gc.ca.

[2] http://www.europarl.europa.eu.

[3] D. S. Munteanu, D. Marcu, "Improving machine translation performance by exploiting non-parallel corpora", *Computational Linguistics*, Vol. 31, No. 4, pp. 477-504, 2005.

[4] http://www.sejong.or.kr.

[5] http://catalog.elra.info.

[6] N. Ide, T. Erjavec, D. Tufis, "Sense discrimination with parallel corpora", *Proc. ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions*, Vol. 8, pp. 61-66, 2002.

[7] J. Chen, J.-Y. Nie, "Automatic construction of parallel English-Chinese corpus for cross-language information retrieval", *Proc. Sixth Conference on Applied Natural Language Processing*, pp. 21-28, 2000.

[8] S. F. Chen, "Aligning sentences in bilingual corpora using lexical information", *Proc. 31st Conference on Association for Computational Linguistics*, pp. 9-16, 1993.

[9] P. F. Brown, J. C. Lai, R. L. Mercer, "Aligning sentences in parallel corpora", *Proc. 29th Annual Meeting of the Association for Computational Linguistics*, pp. 169-176, 1991.

[10] W. A. Gale, K. W. Church, "A program for aligning sentences in bilingual corpora", *Computational Linguistics*, Vol. 19, No. 1, pp. 75-102, 1993.

[11] M. Kay, M. Roscheisen, "Text-translation alignment", *Computational Linguistics*, Vol. 19,

No. 1, pp. 121-142, 1993.

[12] R. C. Moore, "Fast and accurate sentence alignment of bilingual corpora", *Proc. Conference on Association for Machine Translation in the Americas (AMTA)*, pp. 135-244, 2002.

[13] D. Varga, L. Németh, P. Halácsy, A. Kornai, V. Trón, V. Nagy, "Parallel corpora for medium density languages", *Proc. Recent Advances in Natural Language Processing (RANLP) 2005 Conference*, pp. 590-596, 2005.

[14] P. E Brown, V. J. D. Pietra, S. A. D. Pietra, R. L. Mercer, "The mathematics of statistical machine translation: parameter estimation", *Computational Linguistics*, Vol. 19, No. 2, pp. 263-311, 1993.

[15] P. Resnik, "Parallel strands: a preliminary investigation into mining the web for bilingual text", *Machine Translation and the Information Soup*, D. Farwell, L. Gerber, E. Hovy (Eds.), Springer, pp. 72-82, 1998.

[16] P. Resnik, N. A. Smith, "The web as a parallel corpus", *Computational Linguistics*, Vol. 29, No. 3, pp. 349-380, 2003.

[17] K. W. Church, Char_align: "A program for aligning parallel texts at the character level", *Proc. 31st Conference on Association for Computational Linguistics*, pp. 1-8, 1993.

[18] P. Fung, K. NcKeown, "Aligning noisy parallel corpora across language groups: Word pair feature matching by dynamic time warping", *Proc. First Conference on Association for Machine Translation in the Americas (AMTA)*, pp. 81-88, 1994.

[19] C. D. Manning, H. Schütze, 1999. Foundation of Statistical Natural Language Processing.

[20] D. Wu, "Aligning a parallel English-Chinese corpus statistically with lexical criteria", *Proc. 32nd Conference on Association for Computational Linguistics,* pp. 80-87, 1994.

[21] P. Koehn, "Europarl: a parallel corpus for statistical machine translation", *Proc. 10th Machine Name Translation Summit (MT Summit X)*, pp. 79-86, 2005.

[22] M. Haruno, T. Yamazaki, "High-performance bilingual text alignment using statistical and dictionary information", *Proc. 34th Annual Meeting on Association for Computational Linguistics*, pp. 131-138. 1996.

[23] G. Doddington, "Automatic evaluation of machine translation quality using n-gram co-occurrence statistics", *Proc. Second Conference on Human Language Technology (HLT)*, pp. 128-132, 2002.

[24] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation", *Proc. 40th Annual Meeting on Association for Computational Linguistics*, pp. 311-318, 2002.

[25] http://www.statmt.org/wmt08/baseline.html.

[26] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b-2008-01-23.tar.gz.

접수일자: 2008년 11월 20일
게재결정: 2008년 12월 27일

▶ 박정열(Jungyeul Park)
주소: Paroi Nord - La Grande Arche. 1, Parvis de la Défense 92044 Paris La Défense Cedex
소속: 시스트란
전화: +33 (0)8 25 80 10 80
E-mail: park@systran.fr

▶ 차정원(Jeong-Won Cha) : 교신저자
주소: 641-773 경남 창원시 사림동 9번지 소나무5길 국립창원대학교
소속: 국립창원대학교 컴퓨터공학과
전화: 055) 213-3818
E-mail: jcha@changwon.ac.kr