Journal of the Research Institute of Industrial Technology, Vol. 24, 2007

대역사전이 한영 문장정렬에 미치는 영향 분석

서형원* · 조희영** · 김재훈***

*/**한국해양대학교 컴퓨터공학과 대학원, ***한국해양대학교 컴퓨터공학과 부교수

Empirical Impact Analysis of Bilingual Dictionary on Korean-English Sentence Alignment

*Hyung-Won Seo** · *Hee-Young Cho*** · *Jae-Hoon Kim****

*/***Department of Computer Engineering, Korea Maritime University, Busan 606-791, Korea*

****Department of Computer Engineering, Korea Maritime University, Busan 606-791, Korea*

요 약 : 최근 웹에서는 대역문서들을 흔히 찾을 수 있으며 이런 대역문서는 중요한 언어자원이 될 수 있다. 예를 들면 정렬기법을 이용해서 이와 같은 대역문서로부터 병렬말뭉치를 구축할 수 있다. 병렬말뭉치란 한 언어로 쓰인 문장을 그 문장의 번역과 함께 나란히 정렬하여 모아둔 말뭉치를 말한다. 이 논문은 병렬말뭉치를 구축하는데 있어서 대역사전이 문장정렬에 미치는 영향을 분석한다. 문장정렬 방법은 크게 대역사전을 이용하는 어휘기반 문장정렬과 문장의 단순한 길이 정보만을 이용하는 길이기반 문장정렬이 있다. 실험에서 전자와 후자를 위해서 각각 Champollion과 Align_regions이라는 문장정렬 도구를 사용한다. Champollion에서 필요한 대역사전은 한영사전으로부터 대응되는 단어쌍을 추출하여 구축되며, Align_regions의 경우는 특별히 준비해야 할 것은 없다. 정확률과 재현율을 이용한 평가에서는 Champollion이 Align_regions에 비해 좀더 좋은 성능을 보였으며, 대역사전이 문장정렬에 커다란 영향을 주는 것으로 관찰되었다. 앞으로 좀 더 질 좋은 대역사전을 사용한다면 상당히 좋은 결과를 보일 수 있을 것이다.

핵심용어 : 문장정렬, 병렬말뭉치, champollion, Align_regions

KEY WORDS : sentence alignment, parallel corpus, Champollion, Align_regions

ABSTRACT : *On the Web, there are a lot of translated documents in several languages, which can be very useful language resources. For example, a parallel corpus can be constructed from the translated corpus using sentence alignment algorithms. The parallel corpus is a collection of texts in one language with its translation in another language. In this paper, we empirically analyze an impact of bilingual dictionary on Korean-English sentence alignment. In general there are two major methods for sentence*

* psyence24@nate.com 051)410-4896

** serensis@hhu.ac.kr 051)410-4896

*** jhoon@hhu.ac.kr 051)410-4514

alignment: lexicon-based and length-based methods. In the experiments, we use Champollion and Align_regions as sentence alignment tools for the former and the latter, respectively. Champollion needs a Korean-English bilingual dictionary, which is constructed to extract pairs of words corresponding to each other in the two languages, but Align_regions does not do anything. We use precision and recall rate as measures for evaluation. Champollion outperforms Align_regions a little in both precision and recall. We have observed that the bilingual dictionary has a big impacts on Korean-English sentence alignment. In the future, we should refine the Korean-English bilingual dictionary to get better performance.[1)]

1. 서 론

최근 인터넷의 급속한 성장으로 다양한 언어로 의사를 전달할 필요성이 점점 더 늘어나고 있으며, 인터넷에는 같은 내용을 다양한 언어로 표현한 문서들을 자주 발견할 수 있다. 예를 들면 제품을 소개하는 매뉴얼, 뉴스 기사[1)] 등이 있으며 이를 병렬문서(parallel documents)라고 한다. 이와 같은 병렬문서들은 매우 중요한 언어자원이 될 수 있다. 예를 들면 이와 같은 병렬문서로부터 병렬말뭉치(parallel corpus)를 구축할 수 있는데, 이를 위해서는 문장정렬(sentence alignment) 기법들[1]이 이용된다. 병렬말뭉치란 한 언어로 쓰인 문장을 그 문장의 번역과 함께 나란히 정렬하여 모아둔 말뭉치를 말한다. 이 논문은 병렬말뭉치를 구축하는데 있어서 대역사전(bilingual dictionary)이 문장정렬에 미치는 영향을 분석한다. 문장정렬 방법은 크게 대역사전을 이용하는 어휘기반 문장정렬[2]과 문장의 단순한 길이 정보만을 이용하는 길이기반 문장정렬[3]이 있으며 이들 두 방법을 이용해서 병렬말뭉치를 구축하고 구축된 말뭉치의 성능을 비교해봄으로써 대역사전이 한영 문장정렬에 미치는 영향을 분석하고자 한다. 병렬말뭉치는 웹에 공개된 대역문서를 이용해서 구축되며 다음과 같은 단계를 거친다. 1)웹 문서수집기(document collector)를 이용해서 웹으로부터 한영 웹문서(html 문서)를 각각 수집한다. 2) 수집된 각 언어의 웹문서에서 불필요한 내용(태그와 광고 문구 등)을 제거하여 문장을 추출하고, 추출된 문장을 단락단위로 정렬한다. 3) 단락단위로 정렬된 문서를 문장정렬 방법을 이용해서 문장을 정렬한다. 4) 정렬된 병렬문장을 단어 단위로 분리하여 병렬말뭉치를 구축한다. 이런 방법으로 구축된 말뭉치는 통계기반 기계번역[1][4]에 그대로 이용할 수 있으며, 대역사전 구축[5][6]이나 다국어 정보검색 시스템의 색인어 번역 등에 이용될 수 있을 것이다. 여기서 말하는 문장정렬이란 웹이나 일반문서로부터 수집된 원시문서와 대역문서를 문장단위로 정렬하는 것이며, 이를 위해서는 많은 정렬 도구들[2)]이 있다. 이 논문에서는 문장정렬을 위한 도구로서 Champollion[2]과 Align_regions[3]을 이용한다. Champollion은 Align_regions와는 다르게 대역사전을 이용한다. 이 논문에서는 한영 대역사전을 웹에 공개된 각종 한영사전으로부터 자동으로 구축되었으며 다소 오류가 포함되어 있으나 실험에는 큰 영향이 없을 것으로 판단되어 그대로 사용하였다. 정확률과 재현율을 이용한 성능평가에서는 Champollion이 Align_regions에 비해 좀

1) english.donga.co.kr
english.etnews.co.kr
joins.com/cnn

2) http://www.cs.unt.edu/~rada/wa

더 좋은 성능을 보였으며, 대역사전이 문장정렬에 커다란 영향을 주는 것으로 관찰되었다. 앞으로 좀 더 질 좋은 대역사전을 사용한다면 상당히 좋은 결과를 보일 수 있을 것이다. 2장에서는 Align_regions과 Champollion에 관련된 연구에 대하여 소개하고, 3장에서는 두 도구를 이용한 병렬말뭉치 구축 시스템에 대하여 간략히 소개할 것이다. 그리고 4장에서는 대역사전이 문장정렬에 어떠한 영향을 주는지를 살펴보고 5장에서 결론을 맺고, 향후 연구 계획에 대하여 기술한다.

2. 관련 연구

2.1 병렬 말뭉치

병렬말뭉치는 같은 내용이 두 개 이상의 언어로 표현된 말뭉치를 말한다. 최초의 병렬말뭉치는 똑같은 내용을 이집트어와 그리스어로 표기한 Rogetta Stone[3]이라고 말할 수 있다. 오늘날 병렬말뭉치의 대표적인 예는 성경이다[4]. 성경은 같은 내용이 수십 가지의 언어로 표현되어 많은 사람들에게 읽혀지고 있다. 병렬말뭉치는 자연언어처리와 기계번역에서 언어정보 구축 분야에서 널리 사용되고 있으며 그 밖에도 언어 교육, 사전 편찬, 대조언어학 연구 등에서 널리 활용되고 있다. 국내에서 개발된 대표적인 병렬 말뭉치는 세종 병렬 말뭉치[5], KAIST 병렬 말뭉치[6] 등이 있다. 외국의 경우에는 유엔 병렬 말뭉치[7], Europarl[8], Hansards[9] 등이 있다.

3) http://en.wikipedia.org/wiki/Rosetta_Stone
4) http://www.kidok.info./BIBLE
5) http://www.sejong.or.kr
6) http://bola.or.kr
7) http://www.ldc.upenn.edu./Catalog/reame_files/un.readme.html
8) http://people.csail.mit.edu/koehn/publications/europarl
9) http://www.isi.edu/natural-language/download/hansard

2.2 병렬 말뭉치 구축

병렬 말뭉치를 구축하는 방법은 말뭉치 구축 도구를 이용해서 수동으로 구축하는 방법과 자동으로 구축하는 방법이 있다[7][8][9]. 수동으로 구축하는 방법은 정확하지만 많은 인력과 시간 그리고 경비가 지출되기 때문에 자주 사용하지 않는다. 병렬말뭉치 구축은 원시문서의 종류에 따라서 조금씩 다를 수 있다. 원시문서가 책이나 기타 문서일 경우는 대개 원시문서와 원시문서의 번역본으로 병렬문서를 이룬다. 또 다른 경우는 원시문서로 웹 문서를 사용하는 경우이다. 웹 문서로부터 양국어 문서를 수집하여 병렬문서를 구축할 수 있다[7]. 그러나 웹으로부터 구축된 병렬문서는 대량의 문서를 쉽게 구할 수는 있으나 두 문서의 내용이 다를 경우가 종종 발생한다. 따라서 웹을 통해 자동으로 구축하는 방법은 짧은 시간 동안에 많은 양의 말뭉치를 구축할 수 있으나, 구축된 말뭉치에는 항상 어느 정도의 오류가 포함되어 있을 수 있다.

2.3 문장정렬

문장정렬은 웹이나 일반문서로부터 수집된 원시문서와 대역문서로부터 문장단위로 정렬하는 것이다. 대부분 1-1 대응하지만, 1-n, n-1, 1-0, 0-1 와 같은 다양한 대응이 존재한다. 이전의 문장정렬 알고리즘 중에 가장 널리 알려진 Gale와 Church의 알고리즘[3]은 기본 문장이 길면 이에 대응하는 문장도 길고, 기본 문장이 짧으면 역시 짧은 문장으로 번역된다는 가정을 기본으로 하고 있다. 이런 알고리즘의 장점은 수행 속도가 빠르고 문장 구조와 길이가 비슷할 경우 정렬이 잘된다는 점이 있다. 실제로 이렇게 문장의 길이에 기반 하여 확률을 계산하는 알고리즘은 기본적으로 96% 정도의 정확성을 보인다. 하지만 영어와 불어처럼 언어적인 특성이 같은 언어에서는 두 언

어 사이에는 정렬이 잘되는 편이지만 문장 구조가 다른 영어와 한국어 사이에는 비교적 정확성이 떨어진다. Kay와 Rocheisen의 알고리즘 [10]에서는 단어의 대역쌍이 포함되어 있는 정도를 이용하여 문장을 정렬하였다. 이는 전자보다 좋은 결과를 얻은 반면 속도가 느린 단점을 보였다. Simard 등의 알고리즘[11]에서는 문장부호나 한국어 내에 사용된 영어 혹은 한자 정보 같은 유사한 단어나 철자를 이용하였다. 그러나 이 알고리즘 역시 효율이 좋지 않아서 Gale과 Church의 알고리즘 [3]에 동종의 개념을 결합하여 성능을 향상시켰다. Wu의 알고리즘 [12]은 이런 문장을 기반으로 하여 정렬하는 알고리즘의 약점을 보완하기 위해 어휘 번역을 통한 정보(사전)를 활용하여 접근하였다. 이렇게 길이와 어휘를 같이 이용하여 접근한 사례를 소개해 보자면, 주어진 확률을 이용하는 문장 기반으로 하는 번역 모델 [13]이 있고, 패턴 인지 방법을 이용하여 원래의 문장과 그에 따른 번역본 사이에서 일치하는 토큰을 찾는 모델[14]이 있다. 여기서 말한 일치하는 토큰은 모델은 문장이 일치하는 것을 찾기 위해 음의 경계에 대한 정보와의 결합에 이용되고 있다. 이런 문장정렬의 결과는 병렬 말뭉치이다. 병렬 말뭉치는 수작업으로 구성되는 경우가 정확하지만 많은 인력과 시간 그리고 경비가 지출되기 때문에 대체적으로 자동으로 구축하여 좋은 결과를 얻었다.

2.4 문장정렬 도구

문장정렬을 위해서 사용되는 도구는 여러 종류가 있으나[10] 이 논문에서는 Align_regions[3]과 Champollion[2]을 이용하므로 이들에 대한 간단한 특징들을 살펴보고자 한다. Align_regions[3]은 단어 길이의 매우 간단한 통계 모델을 기반으로 하고 있다. 이 프로그램은 한 언어에서 문장이 길면 다른 언어에서도 그에 대응하는 문장 역시 길게 번역된다는 점을 이용하였다. 문장의 대응은 선형적(Linear) 이라는 성질을 이용하고 있다. 하지만 문장의 길이 외에는 특별한 정보를 이용하지 않기 때문에 모든 언어에 쉽게 적용할 수 있는 장점이 있으나 웹 문서와 같이 오류를 많이 포함하고 있는 문서에는 적합하지 않다는 단점이 있다. Align_regions는 두 가지의 단계로 프로그램을 실행한다. 먼저 단락을 정렬한 후, 그 다음 단락 안에 있는 문장을 정렬한다. 그러나 짧은 표제나 서명들은 단락인지 구별하기 쉽지만은 않다. 게다가 이런 짧은 표제나 서명 같은 것들은 항상 모든 언어로 잘 정렬되는 것은 아니다. 그렇지만 다행히도 이런 것들은 간단한 길이의 값에 의해 진짜로부터 매우 쉽게 구별될 수 있다. 이런 단락을 정렬하는 알고리즘은 매우 간단하다. 먼저 한 쌍의 문서에서 두 단락을 고려하는 것으로 시작한다. 두 단락을 고려할 때, 만약 100개의 문자 이상의 길이를 포함하거나 50개의 문자 이하의 길이라면 정렬하고 다음으로 넘어간다. 그러나 만약 단락 중의 하나가 긴 반면에 다른 나머지가 짧다면 짧은 단락은 null로 정렬되고 넘어갈 것이다. 실제로 이 간단한 알고리즘은 영어와 독일어를 시험으로 했을 때, 아무런 오류도 없었다. Champollion[2]은 어휘를 기반으로 문장을 정렬하며 앞에서 소개한 다른 문장정렬 도구와 크게 두 가지가 다르다. 첫째로, noisy input이 있다고 가정을 한다. 실제로 정렬된 문장의 상당수(대부분)가 1-1 매칭이 일어나지 않을 수 있다는 것을 의미한다. 따라서 정렬하는 도중에 삽입과 삭제가 자주 일어날 수 있는데 이것이 중요한 역할을 담당하는 경우가 있다. 이 가정은 어휘의 의미를 따지지 않고 정렬을 하는 것에 대항하는 것이다. 어휘의 정보를 고려하지 않고 단순히 문장의 정보를 가지고 정렬하는 방법은 가끔 noisy data를 다룰 때, 신뢰할 수 없는 결과를 보여줌에도 불구하고 여전히 사용된다.

10) http://www.cs.unt.edu/~rada/wa

그러나 이런 정렬 방법은 현재 Champollion처럼 어휘 정보를 사용하는 도구를 도와주는 역할 밖에 못하고 있는 실정이다. 둘째는, Champollion은 번역된 단어에 가중치를 주어 접근한다는 것이 다른 어휘기반 접근 방법과 다르다. 번역된 어휘는 보통 다음과 같은 특징을 가지고 있다. 첫째, 번역된 단어는 어휘를 번역하는 과정에 어귀를 사용함으로써 확인되어 진다. 둘째, 번역된 단어의 통계들은 문장이 일치하는지 확인하는데 사용된다. 대부분 알려진 문장 정렬 알고리즘들은 번역된 단어를 동등하게 취급한다. 이 말은 문장이 일치하는 지를 계산할 때, 같은 가중치를 번경된 단어 쌍에 적용한다는 것을 나타낸다. 하지만 Champollion 의 경우에는 빈번히 일어나는 번역 쌍을 줄이기 위해 가중치를 주는 것이 일반적이다. 하나, 혹은 더 많은 문장으로 구성되어 있는 어떤 두 조각이 유사한지를 계산하기 위해 이런 방법을 사용한다. 그리고 정렬의 결과 중에는 1-1 번역 외의 번역이나 문장의 길이가 다른 문장을 정렬할 때 어려움(alignment_penalty)이 있는데, 이런 것들은 경험적으로 결정하게 된다. Champollion은 이런 것을 해결하기 위해 동적 프로그래밍(dynamic programming)을 사용한다. 이것은 원시문장과 번역된 문장 사이에서 최대한 가장 유사한 것을 찾아 최선의 정렬을 하기 위함이다. 동적 알고리즘 프로그래밍은 Gale과 Church의 알고리즘[3]과 매우 흡사하다. 그러나 가장 가까운 거리를 갖는 경로를 찾는 방법 대신에 가장 유사한 경로를 찾는다. Champollion은 1-0, 0-1, 1-1, 2-1, 1-2, 1-3, 3-1, 1-4, 4-1까지의 정렬을 지원한다. Champollion은 최상의 정렬을 계산하기에 앞서 두 언어의 복수 문서 모두 단어를 분리한다. 예를 들면, 영어와 아라비아어와 같은 형태학적으로 복잡한 언어들을 가지고 먼저 문장을 단어 단위로 나눈 뒤, 약간의 stemmer를 적용한다. 여기서 말한 stemmer는 각 단어를 그 언어의 사전 형태에 맞게 표준화하는데 사용되는 것이다. 그래서 그렇게 한 후, 사전적인 대응수를 최대화한다. 이는 영어와 아랍어 그리고 중국어에 대해 적용하여 좋은 결과를 보였다.

3. 병렬말뭉치 구축 시스템

이 절에서는 추출된 한영 병렬문서와 문장정렬 도구들을 이용하여 한영 병렬말뭉치 시스템을 설계하고 구현한다. 이 시스템의 구성은 크게 세 단계로 나타낼 수 있다(Fig. 1). 첫 번째, 두 가지의 언어로 되어 있는 병렬문서의 입력을 수정하는 정제단계, 두 번째로 문장정렬을 수행하는 병렬말뭉치 구축단계, 그리고 마지막으로 두 정렬도구의 결과를 비교하여 수정하는 수정단계로 구성된다. Align_regions의 상세한 시스템 구조는 비교적 간단하기 때문에 이 논문에서는 자세히 설명하지 않지만 나중에 설명될 Champollion의 상세 구조를 설명함으로써 쉽게 추론할 수 있을 것이다. Fig. 2는 Champollion의 상세한 시스템 구조이다. 먼저 정제단계에서 단어분리(word segmentation)가 수행된다. 이 때 영어의 경우는 etoken.pl, lowercase.pl, english_stemmer.pl이라는 프로그램이 이용되고, 한국어의 경우는 TokenizerK.pl과 Final_tok.pl이 이용된다. etoken.pl은 영어 문장을 공백과 특수기호 단위로 분리하고, lowercase.pl을 대문자을 소문자로 변경하며, english_stemmer.pl을 영어 단어의 원형을 찾아준다(예: learning → learn). TokenizerK.pl을 한국어 문장을 음절을 바탕으로 단어를 분리하며[15], Final_Tok.pl은 TokenizeK.pl에서 분리된 결과에서 품사정보와 사전 검색에 적합하도록 변경한다. dic.giza.kmu는 한영 대역사전이며, merge_alignment.pl은 Align_regions과 Champollion의 결과가 서로 일치하지 않으므로 이들을 일치시킨다. 이하의 절에서 이들 각 단계를 자세히 설명할 것이다.

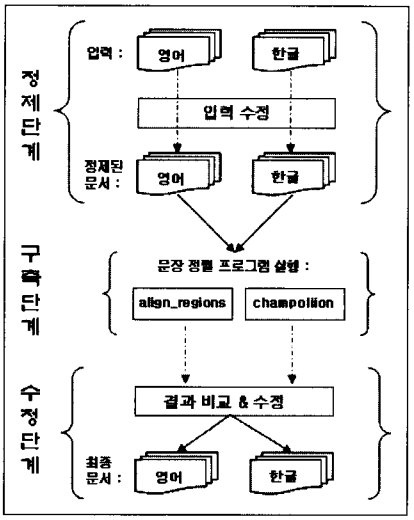

Fig.1 The flow diagram of the proposed system

3.1 병렬문서의 정제 단계

Align_regions과 Champollion을 이용하여 병렬문서를 문장정렬 하기 위해서는 가장 먼저 각각의 입력형태에 맞게 수정해야 하는 작업이 필요하다. Align_regions의 경우를 보면, 한 라인 당 한 단어를 표시해야하며, 문장과 문단이 종료되었음을 표시하는 별도의 분리자(delimiter)가 필요하다. 일반적으로 문단 분리자(hard region, </p>)는 양 언어의 문서에 같은 수가 포함되어야 하며, 문장 분리자(soft region, </s>)은 반드시 같을 필요가 없다. Champollion은 입력형태가 한 라인에 한 문장이 들어가야 한다. 그리고 Align_regions과는 다르게 대역사전 검색을 위해서 단어가 분리되어야 한다. 단어분리는 영어와 한글 부분이 서로 다르다. 앞에서 간단히 언급했듯이 영어 단어분리는 공백과 특수 기호를 기준으로 분리되므로 비교적 간단하다. 즉, 불필요한 데이터를 정제한 후 단어와 단어 사이를 띄우고, 모든 대문자를 소문자로 바꾸고, 단어와 기호(쉼표, 마침표, 세미콜론 등)들을 분리하여 영어사전을 검색한다. 한글 단어분리는 형태소 분석기[15]을 이용하여 형태소 단위로 어절을 분리한다. 각각의 한영 문장 사이에 유사한 단어를 비교하기 위하여 대역사전이 필요하며, 일반 한영사전에서 원형을 모두 뽑아내고, 정확한 사전 비교를 위하여 조사를 모두 없애고, 동사에 있어서는 '다'를 없앴으며 불규칙 동사와 같은 변이형을 모두 사전에 추가한다.

3.2 병렬말뭉치 구축 단계

Align_regions은 정제된 쌍의 병렬 말뭉치를 입력으로 받아 별다른 정보 없이 단지 문장의 길이의 유사함을 계산하여 문장정렬을 한다. 이때에는 특별히 어절은 특별히 분리할 필요가 없고 단지 불필요한 태그나 완전히 번역이 잘못된 문장들만 제거하고 Align_regions를 실행한다.

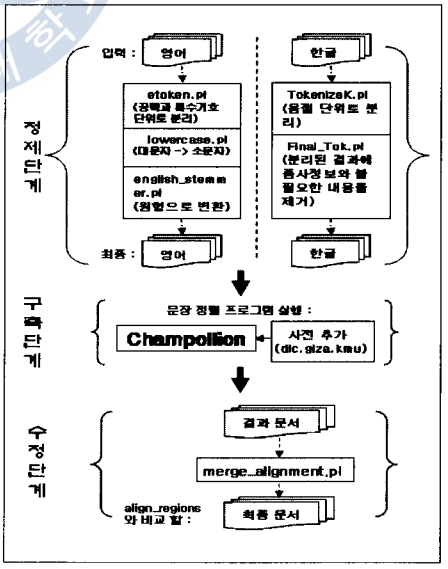

Fig. 2 The system configuration using Champollion as a sentence alignment tool

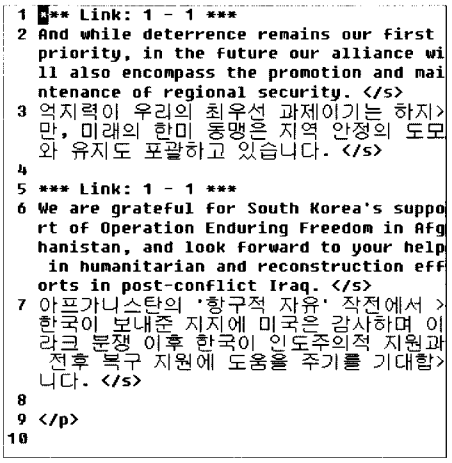

Fig. 3 The output of Align_regions

결과로는 Fig. 3과 같이 문장정렬의 대응관계가 출력되고(예: *** Link: 1 - 1***) 각 언어의 정렬된 문장이 출력된다. 이에 반해, Champollion은 앞서 설명했듯이 한 개의 혹은 그 이상으로 구성된 각각 2개의 문장 사이에 유사함을 계산한다. 이런 계산을 위해 '동적 프로그래밍 기법'(dynamic programming method)을 이용하였는데, Champollion 에서는 경로를 검색하기 위하여 Gale과 Church[3]가 고안한 가장 짧은 거리를 계산하는 알고리즘과는 다르게 가장 유사함(maximum similarity)을 이용하여 계산하였다. Champollion은 1-0, 0-1, 1-1, 2-1, 1-2, 1-3, 3-1, 1-4, 4-1까지의 정렬을 지원하며, Fig. 4와 같이 단순히 두 언어 간의 정렬된 문장 번호만 출력한다. 그런 다음 출력된 문장 번호를 기반으로 하여 최종 영한 병렬 말뭉치를 얻을 수 있다.

Fig. 4 The output of Champollion

3.3 각 정렬도구의 수정단계

이 단계에서는 Align_regions과 Champollion은 출력 형식이 서로 다르게 때문에 서로 비교하기 매우 어렵다. 따라서 이 절에서는 두 정렬도구 사이의 결과 파일을 비교하고 분석하기 위해 출력 형식을 수정하여 맞추는 과정이다. Align_regions 같은 경우에는 확장자가 al 인 파일로 파일 안에 정렬이 된 문장이 나란히 출력이 되며 그 위에 각각 문장들이 몇 대 몇으로 정렬이 되었는지 숫자로 표기가 되어있다. 이와는 좀 다르게 Champollion은 일단 출력 파일이 각각 대응된 문장들은 생략되어 있고 정렬된 문장의 번호만이 출력된다. 이런 결과를 토대로 Merge_alignment.pl이라는 프로그램을 수행할 시 Align_regions과 같은 형식으로 문장과 숫자가 동시에 출력이 된다.

4. 대역사전의 영향 평가

이 논문에서 대역사전이 문장정렬에 미치는 영향을 분석하기 위해서 간단히 문장정렬의 성능을 평가한다. 평가를 위해서 세종 병렬말뭉치[11]를 이용했다. 전체 세종 병렬말뭉치 중에서 241개의 파일만 이용하였고, 단락 수는 약 5만 개이다. Table 1은 세종 병렬말뭉치에 포함된 한국어와 영어 문장 수를 보여주며, Table 2는 각 문장정렬 도구의 정확률과 재현율이다.

Table 1 Statistics of the evaluation corpus

언어	단락 수	문장 수
한국어	51,498	149,930
영어	51,498	150,387

11) http://www.sejong.or.kr/

Table 2 The precision and recall of each sentence alignment tool

문장 정렬도구	정확률 (precision)	재현율 (recall)
Align_regions	95.07%	94.95%
Champollion	95.43%	95.00%

정확률 P와 재현율 R은 Table 3과 같은 분할표(contingency table)를 이용해서 식 (1)과 (2)와 같이 계산된다. Table 3에서 X는 임의의 정렬을 의미하고, ¬X는 X와 다름을 의미한다. A, B, C, D는 정답 정렬과 정렬도구의 결과가 일치하거나 불일치한 수를 의미한다. 예를 들어 A는 정렬도구의 결과와 정답 정답이 정확히 일치한 정렬의 개수이고 B는 정렬도구에서 찾아준 정렬 중에서 정답 정렬과 일치하지 않는 정렬의 개수이다. C와 D도 A와 B와 같이 해석된다.

Table 3 A 2 by 2 contingency table

		정렬 정답	
		X	¬X
정렬 도구 결과	X	A	B
	¬X	C	D

$$P = \frac{A}{A+B} \quad (1)$$

$$R = \frac{A}{A+C} \quad (2)$$

Align_regions과 Champollion사이에는 근소한 차이로 Champollion의 성능이 더 좋게 나왔다. 하지만 여기서 빠진 부분이 있는데, 예를 들면, 왼쪽이 정답 정렬이고 오른쪽이 Champollion의 정렬 결과로 본다면 다음과 같은 경우가 있다.

1,2 <=> 1	1 <=> 1
	2 <=> 0

이 경우에는 영어의 두 문장이 한글의 한 문장과 대응되는 경우이다. 이처럼 2-1이나 1-2 정렬이 일어났을 때, 1-1 정렬이 일어난 것처럼 나오고 나머지 한 문장은 빠져버리는 경우가 발생하였다. 그리고 정답과 비교하여 사람이 실제적인 의미를 봤을 때 틀린 부분들은 오히려 정답이 틀리고 Champollion이 잘 대응되는 경우가 많았다. 이처럼 Champollion은 대역사전을 통해서 전혀 대응이 일어나지 않을 경우에는 '0'을 처리하여 대응문장이 없음을 표시할 수 있었다. 길이 기반의 정렬은 대략적인 정렬을 어느 정도 가능하지만 정확한 정렬에는 다소 오류가 포함될 수 있음을 알 수 있었다. 특히 웹 문서에서 추출된 병렬문서는 정확하게 번역되지 않는 경우가 종종 발생하는데 이 경우는 가능하면 어휘기반 문장정렬 방법을 이용하지 않으며 구축된 병렬말뭉치에 오류가 포함될 수밖에 없다.

5. 결론 및 향후 연구

이 논문은 병렬말뭉치를 구축하는데 있어서 대역사전이 문장정렬에 미치는 영향을 분석한다. Align_regions은 길이 기반 문장정렬 방법으로 문장정렬을 위한 특별히 필요한 정보가 없다. 그러나 Champollion의 경우에는 어휘기반 문장정렬 방법으로 먼저 대역사전이 준비되어야 하며 대역사전을 검색하기 위해서는 병렬문서들에 대해서 단어가 분리되어야 한다. 여기서 반드시 형태소 분석을 필요하지 않는다. 왜냐하면 병렬문서에 포함된 모든 단어가 대역사전에 모두 포함되지 않아도 Champollion

이 처리할 수 있다. 그러나 성능 면에서는 어느 정도 영향을 미칠 것으로 본다. Champollion의 결과는 단순히 정렬된 문장 번호만 나타내기 때문에 결과를 영어와 한글로 나누는 작업을 통해 원하는 문장 정렬 결과를 얻을 수 있었다. 실험에서 정확률과 재현율을 이용한 평가에서는 Champollion이 Align_regions에 비해 좀 더 좋은 성능을 보였으며, 대역사전이 문장정렬에 커다란 영향을 주는 것으로 관찰되었다. 앞으로 좀 더 질 좋은 대역사전을 사용한다면 상당히 좋은 결과를 보일 수 있을 것이다.

참 고 문 헌

[1] Manning, C. D. & Schütze, H. *Foundations of Statistical Natural Language Processing*, Cambridge, MA: MIT Press, 1999.

[2] Ma, X. (2006) "Champollion: A Robust Parallel Text Sentence Aligner," *Proceeding of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*.

[3] Gale, W. A. & Church, K. W. "A program for aligning sentences in bilingual corpora," Computational Linguistics, vol. 19, no. 1, pp. 75-102, 1991

[4] Brown, P., Della Pietra, V., Della Pietra, S., & Mercer, R. "The mathematics of statistical Machine Translation: Parameter estimation," *Computational Linguistics*, vol. 19, no. 2, pp. 263-311, 1993

[5] Nagata, M, Saito, T. & Suzuki, K. "Using the Web as a Bilingual Dictionary," *Proceedings of the ACL 2001 Workshop on Data-Driven Methods in Machine Translation*, pp. 95-102, 2001.

[6] Shin, J. H., Han, Y. S., & Choi, K.-S. "Bilingual Knowledge Acquisition from Korean-English Parallel Corpus Using Alignment Method (Korean-English Alignment at Word and Phrase Level)," *Proceedings of the The 16th International Conference on Computational Linguistics*, pp. 230-235, 1996.

[7] Resnik, P. "Parallel strands: A preliminary investigation into mining the web for bilingual text," *Proceedings of the Third Conference of the Association for Machine Translation in the Americas (AMTA-98)*, 1998.

[8] Yang, C. C. & Li, K. W. "Automatic construction of English/Chinese parallel corpora," *Journal of the American Society for Information Science and Technology*, vol. 54, no. 8, pp. 730 - 742, 2003.

[9] Yang, C. C. & Li, K. W. "Conceptual analysis of parallel corpus collected from the Web," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 5, pp. 632 - 644, 2006.

[10] Kay, M. & M. Roscheisen. "Text-translation alignment," *Computational Linguistics*, vol. 19, no. 1, pp. 121-142, 1993

[11] Simard, M., Foster, G. & Isabelle, P. "Using Cognates to Align Sentences in Bilingual Corpora," *Proceedings of the Fourth International Congress on Theoretical and Methodological Issues in Machine Translation (TMI 92)*, pp. 67-81, 1992.

[12] Wu, D. "Aligning a parallel English-Chinese corpus statistically with lexical criteria," *Proceedings of the 32nd Annual Meeting for Association for Computational Linguistics*, pp. 80-87, 1994.

[13] Chen, S. Building Probabilistic Models for Natural Language. Ph.D. dissertation, Harvard University, Cambridge, MA, 1996.

[14] Melamed, I. D. "Bitext maps and alignment via pattern recognition," *Computational Linguistics*, vol. 25, no 1, pp 107-130, 1999.
[15] 김재훈 & 이공주 "사례기반 학습을 이용한 음절기반 한국어 단어 분리 및 범주 결정", 정보처리학회논문지 B, 제10-B권, 제1호, pp. 47-56, 2003

원고접수일 : 2007년 1월 8일
원고채택일 : 2007년 1월 23일