pISSN: 2287-5905



http://dx.doi.org/10.3745/KTSDE.2013.2.6.433

# An Automatic Extraction of English-Korean Bilingual Terms by Using Word-level Presumptive Alignment

Kong Joo Lee†

## ABSTRACT

A set of bilingual terms is one of the most important factors in building language-related applications such as a machine translation system and a cross-lingual information system. In this paper, we introduce a new approach that automatically extracts candidates of English-Korean bilingual terms by using a bilingual parallel corpus and a basic English-Korean lexicon. This approach can be useful even though the size of the parallel corpus is small. A sentence alignment is achieved first for the document-level parallel corpus. We can align words between a pair of aligned sentences by referencing a basic bilingual lexicon. For unaligned words between a pair of aligned sentences, several assumptions are applied in order to align bilingual term candidates of two languages. A location of a sentence, a relation between words, and linguistic information between two languages are examples of the assumptions. An experimental result shows approximately 71.7% accuracy for the English-Korean bilingual term candidates which are automatically extracted from 1,000 bilingual parallel corpus.

**Keywords : English-Korean Bilingual Terminologies, Sentence Alignment, Word Alignment, Presumptive Alignment**

# 단어 단위의 추정 정렬을 통한 영-한 대역어의 자동 추출

이 공 주†

## 요 약

기계번역 시스템 구축에 가장 필수적인 요소는 번역하고자 하는 언어간의 단어쌍을 담고 있는 대역어 사전이다. 대역어 사전은 기계번역뿐만 아니라 서로 다른 언어간의 정보를 교환하는 모든 응용프로그램의 필수적인 지식원(knowledge source)이다. 본 연구에서는 문서 단위로 정렬된 병렬 코퍼스와 기본적인 대역어 사전을 이용하여 영-한 대역어를 자동으로 추출하는 방법에 대해 소개한다. 이 방법은 수집된 병렬 코퍼스의 크기에 영향을 받지 않는 방법이다. 문서 단위로 정렬된 병렬 코퍼스로부터 문장 단위의 정렬을 수행하고 다시 단어 단위의 정렬을 수행한 후, 정렬이 채 되지 않은 부분에 대해 추정 정렬을 수행한다. 추정 정렬에는 문장에서의 위치, 다른 단어와의 관계, 두 언어간의 언어적 정보 등 다양한 정보가 사용된다. 이렇게 추정 정렬된 단어쌍으로부터 영-한 대역어를 추출할 수 있다. 약 1,000개로 구성된 병렬 코퍼스로부터 추출한 영-한 대역어는 71.7%의 정확도를 얻을 수 있었다.

**키워드 : 영-한 대역어, 문장 정렬, 단어 정렬, 추정 정렬**

## 1. 서 론

기계번역 시스템 구축에 가장 필수적인 요소는 번역하고자 하는 언어간의 단어쌍을 담고 있는 대역어 사전이다. 대역어 사전은 기계번역뿐만 아니라 서로 다른 언어간의 정보를 교환하는 모든 응용프로그램의 필수적인 지식원(knowledge source)이다. 기계번역에 대한 연구가 오랫동안 진행되어 왔고 온라인 형태의 사전들을 쉽게 접할 수 있어서 기본적인 대역어 사전을 구축하는 것이 예전에 비해 쉬워졌다. 그러나 이와 같이 쉽게 구축할 수 있는 대역어 사전은 가장 기본적인 대역어만을 담고 있어서 언어가 갖고 있는 다양한 표현성을 모두 담기 어렵다. 또한 분야마다 조금씩 달라지는 대역어라든지, 요즘과 같이 새로운 기술에 따른 신조어가 끊임없이 만들어지고 있는 상황에서 이에 대해 적절히 대처할 수 있는 대역어 사전을 만드는 일은 쉽지 않다.

최근에는 대량의 이중 언어 병렬 코퍼스를 이용하여 대역어 사전을 구축하는 연구들이 활발하게 진행되어 오고 있다[1, 2, 3]. 대표적인 방법으로 대량의 문장 정렬 코퍼스에 SMT (Statistical Machine Translation)[4] 구축 기법을 적용하여 기계번역 시스템 구축과 함께 동시에 대역어 사전을 구

† 종신회원 : 충남대학교 정보통신공학과 부교수
논문접수 : 2012년 12월 5일
수 정 일 : 1차 2013년 1월 30일, 2차 2013년 3월 5일
심사완료 : 2013년 3월 5일
* Corresponding Author : Kong Joo Lee(kjoolee@cnu.ac.kr)

축할 수 있다. 이와 같은 방법은 매우 효과적으로 대역어 사전을 구축할 수 있다. 그러나 코퍼스에 자주 발생하지 않는 단어들에 대해서는 정확한 대역어 추출이 어렵다. 특히 신조어나 자주 발생하지 않는 외래어 표기 등에 대한 대역어 구축은 이와 같은 방법을 통해서는 구축되기 매우 어렵다.

교류가 많지 않은 언어 간에는 대량의 이중 언어 병렬 코퍼스를 수집하는 것 자체가 어렵다. 그렇기 때문에 대안으로 병렬 코퍼스는 아니더라도 공통되는 부분이 존재하는 비교 코퍼스(comparable)를 이용하여 두 언어 간의 대역어 사전을 구축하고자 하는 시도들이 주목을 받고 있다[1, 5, 6].

본 연구에서는 문서 단위로 정렬된 코퍼스로부터 문장 정렬, 단어 정렬 및 추정 정렬을 통해 효과적으로 대역어를 추출할 수 있는 방법을 제안한다. 문서 단위 정렬 코퍼스와 기본 대역어 사전으로부터 문장 단위의 정렬과 단어 단위의 정렬을 수행한다. 정렬이 되지 않은 부분들을 대상으로 가능한 추정 정렬을 수행한다. 추정된 정렬로부터 얻어진 두 언어간의 단어쌍은 대역어 사전의 엔트리로 사용될 수 있다. Ex. 1의 정렬된 두 문장의 예를 살펴보자.

[Ex. 1]
영문: SK Hynix will transform its M8 factory to a 100% system semiconductor fabrication facility.
국문: SK 하이닉스 M8 공장이 시스템 반도체 팹으로 100% 전환한다.

밑줄 쳐 있는 부분이 기본 사전과 스트링 비교를 통해서 수행된 단어 단위의 정렬이다. 'transform'의 경우에는 기본 사전에 [transform: 바꾸다, 변형시키다, 변화시키다, 변환하다]의 대역어만이 존재하기 때문에 예제 문장에서 쓰인 '전환하다'와의 정렬이 수행되지 못했다. 정렬이 되지 않은 부분들에 대해 다양한 방법을 통해 정렬을 추정해 본다. 추정 정렬을 통해, [Hynix:하이닉스], [semiconductor:반도체], [fabrication facility: 팹] 등과 같은 대역어 사전에 추가할 수 있는 단어쌍을 추출하는 것이 본 연구의 목적이다.

본 연구에서는 정확한 추정 정렬을 위해 사용할 수 있는 다양한 방법에 대하여 제안한다. 또한 본 연구에서 제안하는 방법은 병렬 코퍼스의 크기에 상관없이 사용할 수 있는 방법이다. 즉, 수집된 병렬 코퍼스의 크기가 작더라도 정확한 대역어를 추정할 수 있다는 장점을 갖고 있다. 영어-한국어로 구성된 약 1,000개의 병렬 문서에 대해 실험한 결과를 제시하고 본 연구의 제안 방법이 유용함을 보인다.

본 논문은 다음과 같이 구성된다. 2장에서 기존의 유사한 연구들에 대해 소개하고 3장에서는 문장/단어 정렬을 통한 대역어 사전 구축에 대해 제안하고, 4장에서 실험 결과를 보여준다. 마지막으로 5장에서 결론을 맺는다.

## 2. 관련 연구

이중 언어 병렬 코퍼스(bilingual parallel corpus)로부터 정렬 과정을 통해 다양한 언어 지식을 추출하고자 하는 시도는 오래 전부터 계속되어 왔다. 두 코퍼스의 내용 및 구성의 유사도가 높을수록 양질의 언어 정보를 추출할 수 있지만, 이와 같이 유사도가 높은 양질의 병렬 코퍼스를 구하는 것이 어렵다. 그렇기 때문에 최근의 연구에는 병렬 코퍼스는 아니지만 비교 코퍼스(comparable corpus)를 통해 두 언어간의 대역어 사전과 같은 언어 지식을 추출하고자 하는 많은 시도들이 진행되었다.

비교 코퍼스(comparable corpus)[7]는 동일한 내용을 다른 언어로 담고 있는 병렬 코퍼스(parallel corpus)와는 달리 서로 동일한 내용은 아니지만, 동일한 수집 과정을 거쳐 모아진 데이터의 대표성이 유사한 코퍼스이다. 예를 들어 동일한 기간 동안 동일한 도메인의 동일한 장르의 데이터를 영어와 한국어로 각각 모았을 때, 이를 비교 코퍼스라고 할 수 있다. 그렇기 때문에 이렇게 수집된 비교 코퍼스의 내용은 서로 동일하다고 보장할 수 없다.

[2]의 연구에서는 이중 언어 문장 정렬 코퍼스(bilingual sentence-aligned corpus)로부터 번역 시스템을 위한 변환 규칙(transfer rules)과 어휘 매핑(lexical mapping) 정보를 추출하는 방법을 제안하고 있다. 정렬된 두 언어에 대해 각각 구문 분석을 수행하여 두 언어의 Logical Form(LF)을 생성한다. LF은 각 언어가 갖고 있는 언어적 특성을 최대한 배제하고 모든 언어에서 공통적으로 사용하기 위한 문장의 의미 표현 방식이다. 그렇기 때문에 정렬된 두 언어의 문장은 다른 형태로 표출되지만 그 LF는 유사하거나 동일할 수 있다. LF로 표현한 두 문장에 유사한 LF가 존재하면 (가령 동일한 relation type과 유사한 단어) 이를 정렬시킨다. 제안한 방법론에 대한 평가는 추출된 정렬결과를 기계번역 시스템에 직접 적용하여 기계번역 시스템의 성능 향상을 통해 수행하였다. 변환 규칙과 어휘 매핑을 추출하기 위한 학습 데이터는 수동으로 작성된 161,606 개의 Spanish-English 문장 단위 정렬 코퍼스이다. 전체적인 평가는 기계번역 시스템의 번역 결과를 보고 수행하였다. 상용 시스템인 Babelfish와 200개 문장에 대해 비교했을 때, 약 10% 정도 성능 향상을 보였다.

[1]의 연구에서는 비교 코퍼스로부터 단어대 단어(word-to-word) 대역어 사전을 추출하는 방법을 제안하였다. 비교 코퍼스를 사용하기 때문에 다음과 같은 가정을 도입하여 대역어 사전을 구축하였다. "source1이 target1의 대역어라면 source1과 자주 함께 등장하는 단어들의 대역어는 대상 코퍼스에서 target1과 자주 등장할 것이다". 이와 같은 가정을 갖고 원시언어 코퍼스에서 대역어가 아직 알려지지 않은 unknown 단어에 대해 대역어가 이미 알려져 있는 known 단어들과의 co-occurrence matrix을 구축한다. 대상 언어인 목적언어 코퍼스에서도 동일한 작업을 수행한다. 이때, 기능어(functional word)들은 고려하지 않는다. 한 단어가 갖고 있는 다의성(polysemy)를 고려하기 위해 co-occurrence matrix에서 [$W_1$, $W_2$]의 값을 코퍼스에서 발생한 절대 빈도가 아닌 단어쌍 ($W_1$, $W_2$)사이의 log-likelihood 값을 이용하여 계산하였다. 기본 대역어 사전(seed lexicon)을 이용하여 원시(source) 언어의 단어 $W_s$의 co-occurrence

vector와 가장 거리가 가까운 대상(target) 언어의 co-occurrence vector $W_t$을 찾는다. 이렇게 찾아진 $W_t$가 $W_s$의 대역어이다. [1] 연구에서 사용한 기본 대역어 사전은 GAZA++을 이용해서 자동으로 추출한 사전이다. 1,500개의 정답 대역어 사전(gold standard lexicon)을 구축한 후, 다양한 크기와 종류의 코퍼스를 대상으로 실험을 수행한 결과, 각 품사별로 추출한 대역어 사전의 정확도(precision)은 약 62%에서 82%의 결과를 얻을 수 있었다.

[3] 연구는 주로 neoclassical 어원(root)을 갖는(Greek이나 Latin 어원을 갖고 있는) neoclassical term의 대역어를 추출하는 방법에 대한 연구이다. 비교 코퍼스로부터 대역어(single term과 multiword term 모두 추출)의 좋은 결과를 얻기 위해서는 term의 다양성을 고려해야 한다. 변형 단어(variant term)라는 것은 본래 단어(original term)와 의미적 또는 개념적으로 연관이 있는 단어를 의미한다. 예를 들어 “generation of power”은 “power generation”의 변형단어라고 할 수 있다. 단어변형패턴(term variation pattern)을 이용해서 변형 단어를 인식해 낸다. 각 언어별로 term 후보군을 얻어 내면, 이를 입력으로 하여 term alignment을 수행할 수 있다. 기본적인 term alignment는 기본이 되는 대역어 사전을 이용하여 수행하다. term의 alignment를 수행하기 위해 (1) 여러 언어간에 걸쳐 나타나는 스트링의 유사도와 (2) Greek이나 Latin 어원을 갖는 단어 형성 규칙을 이용하여 수행한다. 또한 각 언어쌍에 대해 단어 형성 규칙에 사용되는 root형태에 대한 매핑 정보를 유지하고 있어야 한다. Source 언어의 neoclassical term을 Greek이나 Latin 어원으로 분리하고 각 언어의 어원 매핑 테이블을 이용하여 대상언어로 간단한 매핑을 수행한다. 입력으로 주어진 후보군들 중, 매핑이 완료된 스트링이 있으면, 그것이 대역어라고 간주할 수 있다. 이와 같은 방법으로 추출한 매핑은 거의 항상 올바른 결과를 유도했다.

## 3. 문장/단어 정렬을 통한 대역어 사전 확장

대역어 사전 구축을 위해서는 Fig. 1과 같은 처리 과정을 거친다. 첫번째로 L1, L2 언어로 정렬된 문서쌍으로부터 각각 문장을 분리하고 각 문장에 대해 형태소 분석과 품사 태깅을 수행한다. 본 연구에서 문장 단위 정렬은 Champollion [8]을 그대로 사용하여 수행하였다. 문장 단위의 정렬 결과는 N:M의 정렬이다. 즉, 반드시 1:1의 문장 정렬이 아니고 정렬된 문서의 특성에 맞게 N:M의 문장 정렬 결과를 얻게 된다. 본 연구에서 N과 M은 0에서 10의 범위를 갖도록 설정하였다. 각 문장 단위의 정렬 결과로부터 단어 단위의 정렬을 수행한다. 단어 단위의 정렬이 수행되지 않은 부분에 대해서 추정 정렬을 통해 대역어 사전에 새로 추가할 수 있는 정보를 추출할 수 있게 된다. 문장 분리와 형태소 분석/품사 태깅 단계, 그리고 문장 단위의 정렬은 본 논문의 핵심 부분이 아니기 때문에 자세한 설명을 제외하고, 단어 단위의 정렬 과정부터 다음의 절에서 자세히 설명한다.

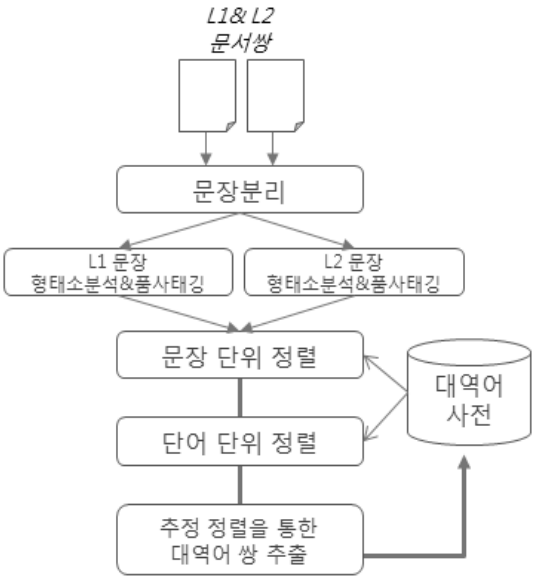


Fig. 1. The overall processing of the system

### 3.1 단어 단위 정렬

단어 단위의 정렬을 수행하는 알고리즘을 Table 2에 제시하였다. 알고리즘의 입력은 문장 단위로 정렬된 문장 쌍이다. 정렬 결과는 N:M이기 때문에 01줄의 S와 T에는 각각 1에서 최대 10개까지의 문장이 들어갈 수 있다. N:M의 정렬 결과 중 N이나 M의 값이 0인 경우는 단어 단위 정렬을 수행할 필요가 없기 때문에 정렬 알고리즘의 입력에서 제외한다. 출력으로는 최종 단어 단위의 정렬 결과를 담고 있는 word_alignment_list이다. 04부터 08줄까지는 정렬된 문장쌍에서 나올 수 있는 모든 가능한 단어쌍에 대해 기본 대역어 사전을 이용하여 정렬 가능한지 (즉, 대역어 사전에 존재하는지) 검사한다. 정렬이 가능할 경우 각 정렬에 대하여 점수를 매기고 그 결과를 word_alignment_list에 담는다.

본 연구에서는 단어 단위 정렬의 정확도를 높이기 위해 부가표현과 같은 예외 사항을 제외하고는 중복 정렬을 허용하지 않는다. 즉, 입력 문장쌍 [S, T]에 대해 단어 단위의 정렬을 수행할 때, S의 한 정렬 단위는 T의 한 정렬 단위와만 정렬을 이룰 수 있다. 또한 S에서 각각의 정렬 단위들은 서로 겹쳐져서는 안되며, T에서 각각의 정렬 단위 또한 서로 겹쳐져서는 안 된다.

Table 2 알고리즘의 입력으로 들어오는 정렬된 문장쌍은 Table 1과 같이 품사[1] 태깅이 된 형태이며, 대역어 사전 참조를 위해 영어 활용 형태에 대해서는 그 원형 정보도 함께 입력으로 들어온다. 또한, 영어 입력의 경우에는 품사 태깅 단계 이후에, 영어 분석 사전에 존재하는 복합명사, 복합어 등도 인식하여 알고리즘의 입력으로 함께 사용한다.(Table 1의 ‘MW’는 Multi Word를 의미한다.)

1) 본 연구에서 영어 품사 집합은 Penn Treebank의 품사 집합을 따르며 한국어는 세종품사태그 집합[9]을 사용하였다.

Table 1. The example of the input for the algorithm

| S | | T |
|---|---|---|
| Individual/JJ | | 개별/NNG |
| component/NN | | 성분/NNG |
| silicon/NN | | 무/NNG+-/SW+실리 |
| -/SYM | | 콘/NNG |
| free/JJ | | 유기/NNG |
| organic/JJ | organic compound/NN | 화합물/NNG+은/JX |
| compound/NN | | 각각/NNG+의/JKG |
| has/VBZ:have | | 분자/NNG+에/JKB |
| at/IN | at least two/MW | 적어도/MAG |
| least/JJS | | 두/MM |
| two/CD | | 개/NNG+의/JKG |
| silanes/NNPS:silane | | 사일레인/UNK |
| unsaturated/JJ | | 불/XP+포화/NNG |
| bonds/NNS:bond | | 결합/NNG+을/JKO |
| in/IN | | 갖/VV+는다/EF+./SF |
| eact/DT | | |
| molecule/NN | | |

Table 2. Word alignment algorithm

```
00  // Word Alignment Algorithm
01  // INPUT: a pair of aligned sentences [S, T]
02  // OUTPUT: word-level alignment list: word_alignment_list
03
04  for Ws in S:
05      for index t in T:
06          if found(LexTrans(Ws), t) or found(Ws, t) then
07              score = get_word_alignment_score(Ws, Wt)
08              save triple [Ws, Wt, score] into word_alignment_list
09
10  sort(word_alignment_list, score);
11  adjust_alignment(word_alignment_list);
12  decide_final_alignment(word_alignment_list);
```

04번째 줄의 $W_s$는 Table 1의 입력 문장 S의 분석 결과이며(복합어까지 포함), 06번째 줄의 found(X, t) 함수는 문장 T의 인덱스 t번째 단어부터 검색하여 스트링 'X'을 찾으면 '참'(True)을 리턴하며, 그때 찾아진 문장 T의 인덱스 t부터의 단어를 $W_t$로 표시한다. 06번째 줄의 LexTrans()는 L1-대-L2의 대역어 사전 정보를 의미하며 그 일부를 아래에 나열하였다.

LexTrans (silane/NNP) = {시레인, 실란}
LexTrans(individual/JJ) = {개별, 각각}
LexTrans(individual/NN) = {개인, 개체, 사람}
LexTrans(each/DT) = {각각}
LexTrans(organic compound/NN) = {유기 화합물}
LexTrans(free/JJ) = {자유롭, 무료, 개방되}
LexTrans(free/VB) = {석방하, 면제하}
LexTrans(bond/NN) = {결속, 속박, 계약, 결합}
LexTrans(at least two/MW) = {적어도 두 개}

07번째의 get_word_align_score($W_s$, $W_t$) 함수는 단어 $W_s$와 $W_t$가 정렬될 때, 정렬의 적합도를 수치화하여 점수로 돌려주는 함수이다. 이 함수는 다음과 같은 휴리스틱을 이용하여 점수를 산출한다.

(1) 정렬에 참여하는 $W_s$와 $W_t$의 단어 길이에 비례 (즉, 여러 단어로 구성되는 정렬에 더 높은 점수 부여)
(2) 기본 사전의 출처가 다르다면, 신뢰성 높은 사전에서 추출한 정보에 의해 정렬되었는지의 여부
(3) 정렬된 단어들이 정확히 매칭되었는지 아닌지의 여부

08번째까지 수행한 결과로 얻어지는 word_alignment_list을 아래에 나열하였다. 여기에 나열되어 있는 정렬들은 아직까지 후보 정렬들이다. word_alignment_list는 점수값에 의해 내림차순으로 정렬된 상태를 유지한다 (10번째 줄의 정렬함수 이용). 정렬에 참여하지 못한 부분은 밑줄로 표시하였다. 단어 'silane' 은 대역어 사전에 포함되어 있으나, 그 대역어가 문장 T에 똑같은 형태로 출현하지 않았기 때문에 word_alignment_list에 포함되지 못했다.

S: Individual component silicon-free organic compound has at least two silanes unsaturated bonds in each molecule .
T: 개별 성분 무-실리콘 유기 화합물 은 각각 의 분자 에 적어도 두 개 의 사일레인 불포화 결합 을 갖 는다 .

```
word_alignment_list = {
   [ [at least two], [적어도 두 개], 128.2 ]
   [ [organic compound], [유기 화합물], 100.2 ]
   [ [silicon], [실리콘], 22.0 ]
   [ [component], [성분], 20.0 ]
   [ [molecule], [분자], 20.0 ]
   [ [organic], [유기], 20.0 ]
   [ [compound], [화합물], 20.0 ]
   [ [bond], [결합], 18.0 ]
   [ [have], [갖], 15.0 ]
   [ [individual], [개별], 10.0 ]  [ [individual], [각각], 10.0 ]
   [ [two], [두 개], 5.0 ]
   [ [each], [각각], 3.0 ]
}
```

word_alignment_list의 후보 정렬 결과를 살펴보면, 'individual'은 '개별'과 '각각' 2개의 한국어와 정렬이 된 반면, 'each'는 '각각'하고만 정렬이 되어 있다. 본 연구에서는 중복 정렬을 허용하지 않기 때문에 'individual'의 최종 정렬을 선택해야 한다. 이 경우, [[individual], [각각]]을 최종 정렬로 결정하게 되면, 단어 'each'와 '개별'은 정렬에 실패하게 된다. 주어진 문장쌍에서 최대한 많은 정렬을 수행하기 위해서는 [[each], [각각]]을 정렬하고 [[individual], [개별]]로 정렬해야 한다. 이렇게 하기 위해서는 문장 S의 단어 중, 문

장 T와의 정렬이 단 하나로만 이루어져 있는 정렬 후보의 점수를 높여준다. Table 2 알고리즘의 11번째 adjust_alignment()함수가 이 역할을 수행한다. 즉, 최대한의 정렬을 수행할 수 있도록 반드시 정렬이 이루어져야 하는 후보 정렬에 가중치를 부여한다. 이렇게 한 후, 높은 점수를 가진 후보 정렬부터 최종 정렬로 결정한다. 이때, 최종 정렬에는 중복되는 정렬이 없어야 하며, 겹쳐지는 정렬이 없어야 한다. word_alignment_list 중 중복/겹쳐지는 정렬 결과를 제거한 것이 Table 3의 최종 출력 결과이다.

다음 단계를 위하여, 문장쌍 [S, T]에서 정렬되지 못한 단어 리스트를 s_unaligned_list와 t_unaligned_list로 구성한다.

Table 3. The output of the word alignment algorithm

| | |
|---|---|
| at least two | 적어도 두 개 |
| organic compound | 유기 화합물 |
| silicon | 실리콘 |
| component | 성분 |
| molecule | 분자 |
| bond | 결합 |
| have | 갖 |
| individual | 개별 |
| each | 각각 |

### 3.2 추정 정렬을 통한 대역어 사전 추출

다음은 추정 정렬을 통한 대역어 추출 과정이다. 크게 4가지의 휴리스틱을 사용하여 추정 정렬을 수행하였다.

1) 부가설명을 이용한 추정 정렬

두 정렬 문장 S와 T 사이에 가장 쉽게 추정 정렬을 수행할 수 있는 경우가 문장 T에서 문장 S의 스트링을 그대로 사용하고 그에 해당하는 대역어를 함께 작성하는 경우이다. 다음과 같은 예제를 살펴보자. 문장 S에서 “flexible smartphone”이라는 표현이 사용되었고 정렬 문장 T에 “플렉시블(flexible)”이라는 표현이 사용되었을 경우, [flexible, flexible]의 정렬은 완료가 된 반면, 문장 T의 ‘플렉시블’은 정렬되지 못한 채 남아있게 된다. S와 T에 동일한 스트링이 사용되었으며 부가표현 표시 (괄호표시)를 사용하고 그 앞쪽에 나타나는 스트링이 미정렬 상태로 남아 있게 되면 [flexible, ‘플렉시블’]의 정렬을 추정해 볼 수 있다.

| | |
|---|---|
| **COND** | S:$W_s$ *AND* T: $W_t$ ( $W_{t+1}$ ) *AND* $W_s$ == $W_{t+1}$ |
| **ACTION** | add [$W_s$, $W_t$] into word_guess_alignment_list |

[Ex. 2] S: flexible smartphone
T: 플렉시블(flexible) 스마트폰
word_alignment_list = { [flexible, flexible], [smartphone, 스마트폰] }
➔ add [flexible, 플렉시블] inot word_guess_alignment_list

2) 최소편집거리를 이용한 추정 정렬

최신의 기술 문서나 신조어의 경우 외래어의 쓰임이 많으며 이러한 외래어는 발음 그대로 표기하는 경우가 빈번하다. 이와 같은 경우에도 추정 정렬을 통해서 외래어 표기의 다양한 형태에 대한 대역어 사전을 구축할 수 있다. 기본사전에 LexTrans(flexible/NNP) = {플렉시블}만이 존재하게 되면 다음의 예제 문장 S와 T에서 적절한 정렬이 이루어지지 못한다. 이때, 기본사전의 ‘flexible’ 대역어인 ‘플렉시블’과 문장 T의 ‘플렉서블’의 최소편집거리(minimum edit distance)를 구해보고 이 값이 threshold 이하일 때, [flexible, ‘플렉서블’] 정렬을 추정해 볼 수 있다.

| | |
|---|---|
| **COND** | S:$W_s$ *AND* T:$W_t$ *AND* EditDistance($W_s$, $W_t$) < threshold |
| **ACTION** | add [$W_s$, $W_t$] into word_guess_alignment_list |

[Ex. 3] S: flexible
T: 플렉서블
➔ add [flexible, 플렉서블] inot word_guess_alignment_list

3) 위치 및 관계 정보를 이용한 추정 정렬

다음에서는 정렬된 단어의 위치와 관계 정보를 이용하여 추정 정렬을 수행한다. 이에 대한 알고리즘을 Table 4에 제시하였다. 이 알고리즘에서 사용되는 함수 Forward(F), Backward(B)는 다음과 같이 정의하였다.

$F_{Align}(W)$ : Sentence S에서 W가 [i, j]를 차지할 때, 기정렬된 j+1번째 단어
$F_{Unalign}(W)$ : Sentence S에서 W가 [i, j]를 차지할 때, 미정렬된 j+1번째 단어
$B_{Align}(W)$ : Sentence S에서 W가 [i, j]를 차지할 때, 기정렬된 i-1번째 단어
$B_{Unalign}(W)$ : Sentence S에서 W가 [i, j]를 차지할 때, 미정렬된 i-1번째 단어

즉, F(W)는 문장에서 W 다음에 나타나는 단어를 추출하는 함수이고, B(W)는 문장에서 W 이전에 나타나는 단어를 추출해 주는 함수이다. 아래 첨자로 ‘Align’이 주어질 때는 추출하는 단어가 이미 정렬이 되어 있어야 하는 단어임을 의미하며, ‘Unalign’일 경우에는 정렬되지 않은 단어를 추출하도록 한다.

AlignMap(W) 함수는 문장 S의 단어 W에 대해 word_alignment_list의 [W, Y]가 존재할 경우 Y를 돌려준다. 즉, 문장 S의 단어 W와 정렬된 문장 T의 단어를 찾아준다.

Table 4 알고리즘의 08번째부터 16번째 줄에서 얻어지는 각각의 정렬 후보를 도식화하면 Fig. 2와 같다. 두 문장 사이에 정렬된 단어들은 서로 실선으로 연결되어 있다. 그림에서 보는 바와 같이 문장 S의 미정렬 단어 $W_s$에 대해 문장 T의 가능한 정렬 후보는 $W_{bb}$, $W_{bf}$, $W_{fb}$, $W_{ff}$의 4 단어가 가능하다.

Table 4. Presumptive alignment algorithm

```
01  // Word Presumptive Alignment Algorithm
02  // INPUT: s_unaligned_list & t_unaligned_list
03  // OUTPUT: word_guess_alignment_list
04
05  for each Ws in s_unaligned_list:
06
07     // Using position and structure
08     Wfb = BUnalign(AlignMap(FAlign(Ws)))
09     scorefb = get_forward_backward_guessing_score(Ws, Wfb)
10     Wff = FUnalign(AlignMap(FAlign(Ws)))
11     scoreff = get_forward_backward_guessing_score(Ws, Wff)
12
13     Wbb = BUnalign(AlignMap(BAlign(Ws)))
14     scorebb = get_forward_backward_guessing_score(Ws, Wbb)
15     Wbf = FUnalign(AlignMap(BAlign(Ws)))
16     scorebf = get_forward_backward_guessing_score(Ws, Wbf)
17
18     // node with the highest score
19     Wt=max[(Wfb,scorefb), (Wff,scoreff), (Wbb,scorebb), (Wbf scorebf) ]
20     add [Ws, Wt] into word_guess_alignment_list
```

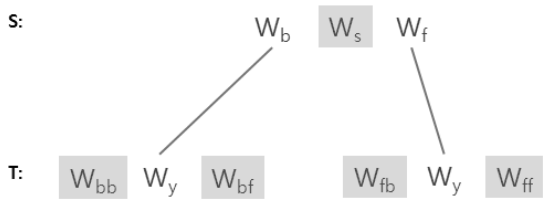


Fig. 2. The possible alignment nodes

다음 그림은 실제 예를 통해 추정 정렬을 설명하고자 한다. Fig. 3은 문장 S와 T에 대해 기본 정렬 이후의 결과 모습이다. 회색으로 칠해져 있는 부분이 정렬이 되지 않고 남은 부분이며, S와 T의 정렬된 단어들은 선으로 서로 연결되어 있다.

Fig. 4는 Fig. 3의 결과에서 문장 S의 미정렬 단어 “phablets”에 대한 추정 정렬을 수행하는 과정이다. 이 경우에는 “phablets”로부터 추출할 수 있는 $F_{Align}$ 정보가 없다. $B_{Align}$(“phablets”)을 수행하면 정렬된 단어 “Note2”을 찾을 수 있고 “Note2”의 문장 T의 정렬 단어가 역시 “Note2”이다. 여기서 $F_{Unalign}$(“Note2”)와 $B_{Unalign}$(“Note2”)의 연산을 통해서 “phablets”에 대한 2개의 정렬 후보 단어 ‘파블렛’과 ‘겔럭시’를 구할 수 있다. 이 각각의 추정 정렬에 대한 적합도 점수는 함수 get_forward_backward_guessing_score()를 통해서 구할 수 있다. 이 함수는 영어와 한국어 사이의 변환 구조 정보와 [“phablets”, “파블렛”]과 [“phablets”, “겔럭시”] 쌍 중에서 어떤 것이 대역어로서 더 적합한지를 측정하여 점수로 환산하여 알려준다. 이 때 사용되는 변환 관련 구조 정보의 일부를 Table 5에 제시하였다. 영어에서 [A/명사류 B/명사류]의 한국어로의 변환은 어순이 바뀌는 [LexTrans(B) LexTrans(A)]보다는 어순이 그대로 유지되는 [LexTrans(A) LexTrans(B)]의 구조가 더욱 선호되며, 이 구조를 얻기 위한 연산자는 $F_{Unalign}$(AlignMap($B_{Align}$()))와 $B_{Unalign}$(AlignMap($F_{Align}$()))이다. 이와 같은 변환 구조 정보를 활용하면 [“phablets”, “파블렛”]이 [“phablets”, “겔럭시”]보다 선호된다.


Fig. 3. The output of the word alignment algorithm


Fig. 4. Presumptive aligning based on Backward and Forward

Table 5. Forward/Backward functions according to English/Korean transfer patterns

| English/Korean transfer patterns | Applicable Forward/Backward function |
|---|---|
| A/NOUN B/NOUN<br>LexTrans(A)LexTrans(B) | $F_{Unalign}(AlignMap(B_{Align}()))$<br>$B_{Unalign}(AlignMap(F_{Align}()))$ |
| A/NOUN **of** B/NOUN<br>LexTrans(B) 의 LexTrans(A) | $F_{Unalign}(AlignMap(F_{Align}()))$<br>$B_{Unalign}(AlignMap(B_{Align}()))$ |
| A/VERB **by** B/NOUN<br>LexTrans(B) 에_의해 LexTrans(A) | $F_{Unalign}(AlignMap(F_{Align}()))$<br>$B_{Unalign}(AlignMap(B_{Align}()))$ |
| A/VERB **as** B/NOUN<br>LexTrans(B) 로써/으로써 LexTrans(A) | $F_{Unalign}(AlignMap(F_{Align}()))$<br>$B_{Unalign}(AlignMap(B_{Align}()))$ |
| A/NOUN **from** B/NOUN<br>LexTrans(A) 에서부터 LexTrans(B) | $F_{Unalign}(AlignMap(B_{Align}()))$<br>$B_{Unalign}(AlignMap(F_{Align}()))$ |
| **by** A/V-ING<br>LexTrans(A) ㅁ/음으로써 | $B_{Unalign}(AlignMap(B_{Align}()))$ |

Fig. 5은 Fig. 3에서 정렬되지 않은 단어 “listed”의 추정 정렬을 수행하는 과정이다. 이 경우에는 $B_{Align}$(“listed”)는 없다. $F_{Align}$(“listed”)의 정보로 단어 “as”를 추출하고, 이에 대한 한국어 정렬 정보 “으로써”를 찾을 수 있다. Table 5를 보면 원문인 영어가 [A/동사류 as]의 구조를 가질 경우 이에 대한 한국어 구조는 [로서/으로서 LexTrans(A)]의 구조를 갖는 경우가 많다. 이 예에서와 같이 “listed as”에서 정렬되지 않은 단어가 ‘listed’이며, “as”와 “으로써”가 정렬된 경우는 $F_{Unalign}(AlignMap(F_{Align}()))$ 연산에 의해 나온 단어가 가장 높은 추정 정렬 후보가 된다.

전체 코퍼스에 대해 1)부터 3)에서 얻어진 word_guess_alignment_list() 중, 일정 빈도 이상 발생한 추정 정렬들을 LexTrans 사전에 추가한다. 본 연구에서는 2회 이상 발생한 추정 정렬을 사전에 추가하였다.

4) 품사 및 기타 정보를 이용한 추정 정렬

1)에서 3)까지의 작업을 수행하고도 정렬되지 않은 부분에 대해서는 다음의 휴리스틱을 이용하여 마지막 추정 정렬을 수행하였다. 본 연구에서는 영어와 한국어 간의 대역어 지식을 추정 정렬에 사용하여 최대한 정확도를 얻고자 하였다. 아래의 세 가지 휴리스틱은 (a), (b), (c)의 순서대로 적용 우선 순위를 갖는다. 즉, 모든 남은 후보 정렬 [$W_s$, $W_t$]에 대해 (a)의 규칙을 우선 적용하여 적용되는 경우 추정 정렬 수행한 후, 나머지에 대해 (b)를 모두 적용해 보고, 마지막으로 (c) 규칙으로 추정 정렬을 수행한다.

| | |
|---|---|
| **COND** | S: $W_s$ AND T: $W_t$ AND Info_Condition ($W_s$, $W_t$) is satisfied |
| **ACTION** | add [$W_s$, $W_t$] into word_guess_alignment_list |

a) prefix, suffix에 따른 발음 정보 및 이에 따른 한국어 대역어 패턴

예) $W_s$가 “~cal/~tic/~tion/~ly/{poly~ multi~}”와 같은 접두/접미사를 포함하고 있으면서,
$W_t$가 “~적/~의/~션/{~하게 ~되게 ~히}/다~”와 같은 접두/접미사를 포함하고 있으면
➔ add [$W_s$, $W_t$] into word_guess_alignment_list

b) 영어와 한국어 대역어의 발음 정보: 영어의 알파벳에 대해 가능한 모든 bigram을 구하고, 이에 대한 한국어 발음을 발음기호 정보를 이용하여 구축하였다. 다음은 그 중 하나의 예제이다.

예) $W_s$가 ‘ca~’로 시작하고
$W_t$의 첫글자의 초성: {‘ㅋ’} 이고 중성: {‘ㅏ’, ‘ㅋ’, ‘ㅗ’, ‘ㅐ’, ‘ㅖ’ } 이면
➔ add [$W_s$, $W_t$] into word_guess_alignment_list

c) 동일한 품사 정보: $W_s$와 $W_t$가 동일한 품사인 경우; 다음은 영어와 한국어 사이에 동일한 품사로 간주한 모든 품사 쌍이다.


Fig. 5. Presumptive aligning based on Forward and Forward

| 영어 | 한국어 | 영어 | 한국어 |
|---|---|---|---|
| 명사류(N*) | 명사(NNG, NNP) | 형용사류(J*) | 형용사(VA)+관형형어미 |
| 동사류(V*) | 동사(VV) | 부사류(R*) | 일반부사(MAG) |

## 4. 실험 및 평가

### 4.1 실험 환경

본 논문에서 제안하는 접근 방법은 정렬된 문장쌍이 매우 유사할 때 효과적이다. 즉 번역문이 지나치게 의역을 한 경우에는 좋은 결과를 얻을 수 없다. 그렇기 때문에 정렬된 문장 쌍의 유사도 - 즉, 정렬의 결과 - 에 따라 얻을 수 있는 결과가 확연히 다르다. 본 연구에서는 양질의 결과를 얻기 위해 정렬된 문장 쌍 중, 문장 정렬 유사도가 높은 정렬쌍만을 사용하고자 한다. 본 논문에서 문장 정렬 유사도는 다음과 같이 정의하였고, 이 문장 정렬 유사도 값이 높을수록 단어 단위의 정렬 결과가 좋다고 간주할 수 있다.

$$\frac{num\ \_aligned\_words}{(num\ \_unaligned\_L1\_words + num\ \_unaligned\_L2\_words)} \quad (1)$$

본 논문에서는 Equation (1)의 값이 1.0 이상인 값을 갖는 정렬쌍에 대해서만 3장에서 설명한 추정 정렬을 통한 대역어 추출을 수행하도록 하였다. num_unaligned_L1_words와 num_unaligned_L2_word를 구할 때 한국어의 조사, 어미와 같은 기능어는 고려하지 않았다.

실험 대상으로 사용한 문서 단위 정렬 코퍼스는 전자/기술 분야의 신문 기사와 영-한으로 작성된 각종 매뉴얼로 구성되어 있다. 전체 약 1,000개의 문서 단위 정렬 코퍼스로부터 추출한 정렬쌍은 다음과 같다.

### 4.2 추정 정렬을 통해 추출한 대역어의 정확도

다음 Table 7은 본 연구에서 추정 정렬을 통해 추출한 영-한 대역어의 일부이다. 잘못된 영-한 대역어의 대부분은 영어와 한국어 대역어 사이의 단어 개수가 차이가 나는 경우이다. “unallocated”는 한국어로 “할당되지 않다/못하다”로 번역이 되어야 하는데 추출된 결과는 “할당되다”로만 나왔다. 이는 본 연구의 방법이 부정어 ‘않다/못하다’를 하나의 대역어로 포함하여 추출하지 못했기 때문이다.

성능은 정확률(precision)과 재현율(recall)로 평가하였다. 정확률은 Table 6의 “중복 제거한 추정 정렬”로 나온 결과 14,004개 모두에 대해서 평가하였다. 대역어가 추출된 단계별로 나온 정확률과 전체 정확률은 다음 Table 8과 같다.

Table 6. The result of alignment

| | |
|---|---|
| Total number of sentences | English: 52,320<br>Korean: 49,176 |
| Total number of aligned sentences | 52,120 |
| Total number of aligned sentences which meet Equation (1)>1.0 | 35,845 |
| Total number of bilingual terms extracted through presumptive alignment | 53,601 |
| Total number of unique bilingual terms extracted through presumptive alignment | 14,004 |

Table 7. English-Korean bilingual terms extracted through the presumptive alignment algorithm

| Correct Korean-English bilingual terms | | Incorrect Korean-English bilingual terms | |
|---|---|---|---|
| implementable | 구현가능하다 | LTE | 에벌루션 |
| scalability | 확장성 | unallocated | 할당되다 |
| ferroelectrics | 강유전체 | oxygen | 하나 |
| nonvolatile | 비휘발성 | inaccessible | 없다 |
| Hectopascal | 헥토파스칼 | want | 설명하다 |
| certified e-mail | 공인전자주소 | Microsoft | 클라우드 |
| boosting | 활성화하다 | ranked | 높다 |
| digital signature | 전자서명 | provide | 파트너 |
| eject | 분사하다 | datacenter | 동시 |
| visualize | 시각화하다 | version | 후속작 |
| store | 스토어 | time | 런타임 |
| cloud | 클라우드 | amount | 적정량 |

Table 8. The precision of bilingual terms extracted in presumptive alignment

| presumptive aligning steps | precision |
|---|---|
| (가) Using parentheses | 98.5% |
| (나) Using minimum edit distance | 100.0% |
| (다) Using Forward and Backward function | 72.8% |
| (라) Using POS and etc info. | 60.4% |
| **TOTAL** | 71.7% |

"최소편집거리를 이용한 추정 정렬"의 경우에는 100%의 정확도를 보였다. "부가설명을 이용한 추정 정렬"의 경우에는 추정 정렬의 경계를 정확히 인식하지 못하여 약간의 오류가 있었다. (가)와 (나)의 경우는 정확도는 높은 반면 추정할 수 있는 경우는 전체 중 약 15% 정도만을 차지하였다. 전체적으로 약 71.7%의 정확도를 얻을 수 있었다.

본 논문의 접근 방법이 얼마나 많은 사전 정보를 추정해 낼 수 있는지 판단하기 위해 재현율(recall)을 평가해 보았다. Equation (1)이 1.0 이상을 만족하는 정렬 문장 중, 임의로 1,000개의 문장을 뽑아서 수동 단어 정렬을 수행하였다. 이 중, 사전 정보로 이미 저장되어 있는 정렬쌍을 제외하고 난 후, 중복된 것을 제외하면 모두 453개의 정렬쌍을 얻을 수 있었다. 이 중 시스템이 추정 정렬로 대역어를 맞게 생성해 낸 것은 모두 320개로 약 70.6%의 recall의 결과를 얻었다.

## 5. 결 론

본 논문에서는 문서 단위의 정렬 코퍼스로부터 영-한 대역어 사전에 추가할 수 있는 대역어를 자동으로 추출할 수 있는 방법을 소개하였다. 문서 단위로 정렬된 코퍼스로부터 문장 단위의 정렬을 수행한다. 정렬된 문장쌍으로부터 기본 대역어 사전을 이용하여 단어 단위의 정렬을 수행한다. 이렇게 한 후, 양쪽 문장에서 정렬되지 못한 단어들을 대상으로 추정 정렬을 수행하여 대역어 사전에 추가할 수 있는 대역어 쌍을 추출할 수 있다.

본 논문에서 제안한 방법은 적은 규모의 병렬 코퍼스에서도 양질의 대역어 후보를 추출할 수 있다는 장점이 있다. 그렇기 때문에 아직 병렬 코퍼스가 충분히 확보되지 못한 새로이 등장하는 분야에서 대역어를 추출할 때 매우 유용하게 사용될 수 있다. 또는 전문 분야의 문서 병렬 코퍼스로부터 해당 전문 분야의 대역어를 추출하는 데 매우 유용하다. 다만, 수집된 병렬 코퍼스의 정렬된 문장들이 지나치게 의역을 하여 문장의 구성 성분이 많이 차이가 날 경우에는 좋은 성능을 발휘하지 못한다는 단점이 있다.

본 연구에서는 여러 단어로 구성되는 대역어에 대해서는 많은 고려를 하지 못했다. 그러다 보니 추출된 대역어의 오류 대부분이 여러 단어로 구성되어야 하는 대역어인 경우가 많았다. 특히 영어와 한국어 사이의 대역어가 서로 다른 품사 구조를 가질 경우 대역어 추출이 좋지 않은 결과를 보였다.

향후 연구에서는 부분 구문 분석 등을 통해 여러 단어로 구성되어야 하는 대역어 추출에 대해 연구를 수행할 것이다. 또한, 병렬 코퍼스가 아닌 비교 코퍼스로부터도 정확한 대역어를 추출할 수 있도록 추정 방법을 확장하도록 할 것이다.

## 참 고 문 헌

[1] Elena Irimia "Experimenting with Extracting Lexical Dictionaries from Comparable Corpora for English-Romanian language pair", in LREC2012 Workshop, 2012.

[2] Arul Menezes and Stephen D. Richardson, "A best-first alignment algorithm for automatic extraction of transfer mappings from bilingual corpora," in DDMR Workshop, ACL, 2003.

[3] Marion Weller, Anita Gojun, Ulrich Heid, Beatrice Daille, Rima Harastani, "Simple methods for dealing with term variation and term alignment," in Proceedings of the 9th International Conference on Terminology and Artificial Intelligence, 2011.

[4] P. Koehn, "Statistical Machine Translation," Cambridge University Press, 2010.

[5] B´eatrice Daille "Building bilingual terminologies from comparable corpora: The TTC TermSuite." in LREC 2012 Workshop, 29-32, 2012.

[6] Pascale Fung and Percy Cheung, Mining "Very-Non-Parallel Corpora: Parallel Sentence and Lexicon Extraction via Bootstrapping and EM," in Proceedings of EMNLP, 2004.

[7] Paola Carrion Gonzalez and E. Cartier, "Technological tools for dictionary and corpora building for minority languages: example of the French-based Creoles," in Proceedings of Workshop on Language Technology for Normalisation of Less-Resourced Languages, 2012.

[8] X. Ma, "Champollion: A Robust Parallel Text Sentence Aligner," in Proceedings of LREC, 2006.

[9] Sejoong Project 21, http://www.sejong.or.kr/


## 이 공 주

e-mail : kjoolee@cnu.ac.kr

1992년 서강대학교 전자계산학과(학사)

1994년 한국과학기술원 전산학과(공학석사)

1998년 한국과학기술원 전산학과(공학박사)

1998년~2003년 한국마이크로소프트(유) 연구원

2003년 이화여자대학교 컴퓨터학과 대우전임강사

2004년 경인여자대학 전산정보과 전임강사

2005년~현 재 충남대학교 정보통신공학과 부교수

관심분야 : 자연언어처리, 기계번역, 정보검색, 정보추출