구간 단위 문장대응을 통한 위키피디아 문서 내 한국어-영어 병렬문장의 효과적인 추출

Effective Korean-English Parallel Sentence Extraction from Wikipedia by Consecutive Sentence Sequence Matching

요 약

본 논문은 한국어-영어 위키피디아(Wikipedia) 문서로부터 병렬문장(parallel sentence)를 효과적으로 추출하는 새로운 방법으로, 대응 관계에 있는 한국어-영어 문장의 연속 수열 탐지를 제안한다. 제안하는 기법은 한국어 위키피디아 본문에는 구간 단위로 영어 본문과 대응 관계가 높은 경우가 많다는 관찰에 기반 하여, 높은 정확도의 병렬문장 추출을 목표로 한다. 실제 한국어-영어 위키피디아 문서 50쌍을 이용한 실험에서, 본 논문이 제안한 구간 단위 문장대응 방식은 기존의 문장 단위 대응 기법이나 문서 정렬 기법보다 8.4~56.1% 높은 정확도로 병렬문장을 추출하여, 다종언어 위키피디아 자원으로부터 한국어 병렬문장 말뭉치 생성에 효과적임을 확인할 수 있었다.

1. 서론

기계학습을 통한 자동번역 기술과 이를 이용한 지능형 서비스가 급격히 발전함에 따라, 자동번역을 위한 기계학습의 근간이 되는 이종언어 대조 코퍼스(bilingual parallel corpus) 자원의 중요성이 부각되고 있다. 신경망 기반 기계번역은 번역 관계에 있는 대량의 병렬문장(parallel sentence)을 학습 데이터로부터 원시어 문장에서 특정 단어 수열이 출현할 때 도착어 문장에서 특정 단어가 수열이 출현할 확률이 최대가 되도록 신경망을 학습시킴으로써 이루어진다. 효과적인 번역을 목표로 신경망을 안정적으로 학습시키기 위해서는, 다양하고 방대한 이종언어 대조 코퍼스의 입력이 필수적으로 요구된다. 심층신경망을 통해 높은 성능을 보이는 것으로 알려진 영어-프랑스어 기계번역 연구의 경우, 신경망 학습에 13억 단어 규모의 영어-프랑스어 병렬 코퍼스가 사용되었다[1].

현재 공개된 한국어 대조 코퍼스는, 이와 같은 기계학습 기반의 자동번역 기술의 연구와 구현에 요구되는 규모와 다양성에서 한계가 명확한 상황으로, 지속적인 한국어 병렬문장 코퍼스 확충이 요청되는 실정이다. 한국어-영어 병렬문장을 제공하는 몇 안 되는 공개 코퍼스 중 하나인 세종말뭉치[2]의 경우, 한국어-영어 병렬문장을 한국어 기준 692,852어절을 제공하고 있는데 그치고 있다. 현재 한국어 대조 코퍼스의 또 다른 문제점으로 현시적 언어의 반영이 제한적이라는 점을 들 수 있다. 세종말뭉치의 경우, 2007년 이후 개발이 중단된 상황이다. 이에 더하여, 영어 외의 다른 언어와 한국어 간의 대조 코퍼스가 매우 부족한 면도 현재 한국어 이종언어 코퍼스가 당면한 문제 중 하나다. 이종언어 말뭉치의 부족은 한국어 기계번역 연구에 제약점으로 대두되고 있는 실정이다.

이러한 한국어 이종언어 말뭉치 구축을 위해 위키피디아를 활용하는 방안이 제시되고 있다. 위키피디아(Wikipedia)는 다양한 주제에 대한 다종언어 문서 자원으로, 현시적 주제에 대한 양질의 텍스트가 지속적으로 생산되고 배포되는 공간이다. 하지만, 명시적으로 번역된 글과 달리, 같은 주제어에 대한 위키피디아 내 다종언어 문서의 경우, 일부 공통적인 내용을 포함하는 반면, 상당 부분에 있어서는 각 국가의 특정 상황에 기인하여 개별적인 내용으로 작성되어 있으므로, 문서 간에 명확한 번역 관계를 찾기 어려운 경우가 대부분이다. 이러한 문제를 해결하기 위해, 다종언어 위키피디아 문서 쌍에서 문장 쌍의 번역 확률[3]을 활용하거나, 문장 내 단어의 대응도[4]를 활용하여 의미적 대응 관계가 높을 것으로 추정되는 문장 쌍을 탐색하는 방법이 제안되어 왔다. 하지만 최근에 개발된 병행문장 탐지기법에서도 탐지된 문장 쌍이 실제 대응 관계에 있는 비율(정확도)가 50% 미만으로, 보다 정확한 추출 기법의 개발이 요청되는 상황이다[3][4].

본 논문은 한국어 위키피디아 문서가 영어 문서의 일부 구간을 선별적으로 번역하여 경우가 많다는 관찰에 착한하여, 한국어-영어 본문에서 연속적인 문장수열로서 대응관계가 높은 구간을 탐지하고, 이 구간에 속한 문장 쌍을 병렬문장으로 추출하는 **구간 단위 문장대응 기법**을 제안한다. 임의로 선택한 50개 실제 한영 위키피디아 문서 쌍에 대하여, 제안하는 기법은 기존에 제시된 두 가지 방법(4장)의 병행문장 추출 성능을 실험한 결과(5장), 본 연구가 제안하는 방식이 문장 단위 대응 기법보다 1.53~3.49배, 문서 전체 문장정렬 방식보다 1.20~2.11배 높은 정확도로 병렬문장 추출을 달성함을 확인할 수 있었다.

2. 위키피디아 문서에서 구간 단위 문장대응 관찰

동일한 주제어에 대한 한국어-영어 위키피디아의 문서에서, 대응관계에 있는(같은 의미를 갖는) 문장이 구간 단위(연속된 수열)로 나타나는 경우가 빈번함을 정량적으로 보이기 위해, 50개의 위키피디아 주제어를 임의로 선택하여, 해당 한국어-영어 문서에 존재하는 문장 간 대응 관계의 양과 형태를 조사하였다. 구체적인 조사방법은 다음과 같다.

- 조사대상 수집: 한국어 위키피디아에서 '임의문서로 이동' 기능을 이용해, 50개의 한국어-영어 문서 쌍을 수집했다. 이 때 임의문서로 제시된 한국어 문서 혹은 영어 문서가 5개 미만의 문장으로 구성된 경우, 수집에서 배제하였다.
- 대응 문장 파악: 수집된 50개의 한국어-영어 문서 쌍에 대하여 1명의 대학생이 병렬문장 관계를 표기한 정답(ground truth)을 만들었다. 정답은 한국어 문장과 영어 문장 쌍의 집합으로, 각 한국어 문장에 대해 영어 문서 내에 가장 의미적 대응도가 높은 문장을 택했으며, 한국어(영어) 문장의 내용이 50% 이상이 영어(한국어)에 반영된 경우만 정답에 포함시켰다. 한 개의 한국어(영어) 문장이 최대 2개의 연속된 영어(한국어) 문장으로 나뉘어 대응되는 경우, 복수 개의 대응관계를 정답에 포함하였다. 그 외 복잡한 대응 관계가 있는 문장은 병렬문장 코퍼스에 부적합한 것으로 판별하여, 정답에 포함시키지 않도록 하였다.
- 구간 단위 대응관계에 있는 문장 비율 측정: 50쌍의 문서에 대해 사람이 작성한 정답에서 연속된 한국어 문장수열이 연속된 영어 문장수열에 대응되는 비율을 측정했다. 한국어 문서의 k번째 문장과 k+1번째 문장이 영어 문서의 e번째 문장과 e+1번째 문장에 각각 대응되는 경우, k번째 한국어 문장과 k+1번째 한국어 문장을 구간 단위 대응관계가 있는 문장으로 판단했다.

조사대상으로 수집된 50개 문서 쌍은 평균 15.64개의 한국어 문장(최소 5, 최대 50)과 평균 54.6개의 영어 문장(최소 5, 최대 376)로 이루어졌다. 50개 한국어 문서에는 총 782개의 한국어 문장이 있으며, 대응 문장을 포악한 결과, 총 390개 한국어 문장(전체의 49.9%)이 특정 영어 문장과 대응 관계에 있음을 파악할 수 있었다.

표1은 영어 문장과 대응되는 한국어 문장 중 구간 단위 대응에 속한 경우(대응되는 한국어-영어 문장이 2개 이상, 연속적으로 등장하는 경우)가 어느 정도인 지 조사한 결과다. 표1의 2행에서 조사대상의 58%의 문서(=(10+19)/50)에서 60% 이상의 한국어-영어 문장 대응관계가 구간대응 형태로 나타남을 알 수 있다. 또한, 구간 대응에 속한 한국어 문장 비율이 80% 이상인 문서(전체의 38%)에서는 평균 2.26개 구간이 등장하며, 이 때 구간의 평균 길이는 4.91임을 알 수 있다. 결론적으로, 영어 문서와 대응관계에 있는 한국어 위키피디아 문서의 경우, 대체로 복수 개의 대응되는 구간(연속된 문장수열 쌍)이 발견되며, 따라서, 본 논문의 가정이 실제적임을 확인할 수 있다.

구간 대응에 속한 한국어 문장비율	20% 이하	20%초과, 40%이하	40%초과, 60%이하	60% 초과, 80% 이하	80% 초과
문서 수	12개	3개	6개	10개	19개
평균 구간 개수	0개	1개	1.2개	1.7개	2.26개
평균 구간 길이	0문장	2문장	2.17문장	2.83문장	4.91문장

표1. 한국어 문장 대비 구간 대응 관계에 있는 한국어 문장에 따른 조사대상 분포

3. 구간 단위 문장대응 탐색 기법

본 논문은 주어진 한영 문서 쌍에서 대응도가 높은 연속된 한국어 문장수열(구간)과 연속된 영어 문장수열의 쌍을 병렬문장의 가능성이 높은 조건으로 인식하고, 이를 효과적으로 탐지하여 병렬문장을 추출하는 구간 단위 문장 대응 방법을 제안한다. 제안하는 기법은 다음 3가지 과정을 거쳐 병렬문장을 추출한다.

- **과정1. 문장 정의**: 한국어-영어 문서 쌍의 텍스트를 문장 단위로 분절하여 한국어 문장수열 $(K_1, K_2, \ldots, K_n)$과 영어 문장수열 $(E_1, E_2, \ldots, E_m)$을 정의한다.
- **과정2. 한국어-영어 문장 간 대응도 계산**: 분절된 한국어-영어 문장의 모든 쌍에 대하여, 두 문장이 같은 의미를 내포하고 있는 정도를 나타내는 대응도 $Corr(i,j)$를 실수 값으로 구한다.
- **과정3. 한국어-영어 문장 간 대응도 합이 최대인 문장수열 쌍에서 연속 문장수열 추출**: 한국어 문장수열과 영어 문장수열의 쌍으로, 같은 위치(index)의 한국어-영어 문장 대응도의 총합이 최대가 되는 경우를 탐색한다. 그리고 이 한국어-영어 수열 쌍에서 한국어 문장과 영어 문장이 연속적으로 나타나는 최대 구간을 탐색하여, 그 탐색 결과를 병렬문장으로 추출한다.

  먼저, 전체 한국어 문장수열의 부분수열로서 대응도 총합이 최대가 되는 문장수열 쌍은 다음 조건을 만족하는 부분수열 쌍 $\langle(K_{s_1}, K_{s_2}, \ldots, K_{s_k}), (E_{t_1}, E_{t_2}, \ldots, E_{t_k})\rangle$으로 정의할 수 있다:

  - $s_i < s_{i+1}$ and $t_i < t_{i+1}$ for $1 \leqq i < k$
  - $k \leqq m$ and $k \leqq n$
  - maximizes $\sum_{i=1}^{k} Corr(s_i, t_i)$

  위의 식을 만족하는 $K_{s_1}, K_{s_2}, \ldots, K_{s_k}$과 $E_{t_1}, E_{t_2}, \ldots, E_{t_k}$의 부분수열로서, 연속적이며 대응도 합이 최대가 되는 문장구간(consecutive subsequence) 쌍은 다음조건을 만족하는 부분순열 $\langle(K_{s_b}, K_{s_{b+1}}, \ldots, K_{s_e}), (E_{t_{b'}}, E_{t_{b'+1}}, \ldots, E_{t_{e'}})\rangle$로 정의할 수 있다:

  - $1 \leqq b < e \leqq k,\ 1 \leqq b' < e' \leqq k,\ e-b = e'-b'$
  - $s_{i+1} = s_i + 1$ for $b \leqq i < e$
  - $t_{j+1} = t_j + 1$ for $b' \leqq j < e'$
  - maximizes $\sum_{i=0}^{k-1} Corr(s_b+i, t_{b'}+i)$

  위 식으로 정의된 한국어-영어 문장구간으로부터 같은 순서에 있는 한영 문장 쌍 $(K_{s_b}, E_{t_{b'}}), (K_{s_b+1}, E_{t_{b'}+1}), \ldots, (K_{s_e}, E_{t_{e'}})$를 병렬문장으로 추출한다. 그리고 주어진 횟수만큼, 앞서 추출된 문장을 배제하고, 과정3을 반복하여 병렬문장을 추출한다.

과정1과 과정2는 기존의 문장정렬 기법이나 문장 단위 대응 기법과 동일한 방식으로 볼 수 있다. 한국어-영어 문장 간 대응도는 대역어 사전을 이용한 방식, 문장의 특징집합을 이용한 방식[4], 혹은 번역확률[3]을 이용하는 방식 등이 사용될 수 있다.

알고리즘1은 과정3에서 한국어-영어 문장 쌍의 대응도 값을 입력으로 받은 후 문장 간 대응도 합이 최대가 되는 한국어-영어 문장수열 쌍을 탐지하는 알고리즘으로, 문장정렬에 널리 사용되는 방식대로, 최장 부분 수열 탐색과 유사한 형태의 동적 프로그래밍이다. 본 논문에서는 공간의 제약으로 문장수열 쌍의 대응도의 최

입력: $Corr(1..n, 1..m)$: 한영 문장 쌍의 대응도
τ : 대응 문장수열 탐색에서 문장 단위 대응도 하한 값

```
1  w[0..n, 0..m] = 0
2  for i = 1 to n do
3    for j = 1 to m do
4      if Corr(i, j) ≧ τ then
5        w[i, j] ← max(w[i−1, j], w[i, j−1]) + Corr(i,j)
6      else
7        w[i, j] ← max(w[i−1, j], w[i, j−1])
8      end if
9    end for
10 end for
11 return w[n,m]
```

알고리즘1: 대응도 합이 최대가 되는 문장수열 쌍 탐색

입력: $Corr(1..n, 1..m)$: 한영 문장 쌍의 대응도
τ : 대응 문장수열 탐색에서 문장 단위 대응도 하한 값
$(K_{s_1}, K_{s_2}, \ldots, K_{s_k})$, $(E_{t_1}, E_{t_2}, \ldots, E_{t_k})$: 한국어-영어 문장수열 쌍

```
1  w[0..n, 0..m] = 0
2  for i = 2 to k do
3    if s_i = s_{i−1} + 1 then
4      for 2 = 1 to k do
5        if t_j = t_{j−1} + 1 then
6          if Corr(s_i, t_j) ≧ τ and  then
7            w[s_i, s_i] ← max(w[s_{i−1}, t_j], w[s_i, t_{j−1}]) + Corr(s_i, t_j)
8          else
9            w[s_j, t_j] ← 0
10         end if
11       end if
12     end for
13   end if
14 end for
15 return a maximum value of w[1..n, 1..m]
```

알고리즘2: 대응도 합이 최대가 되는 연속문장순열(구간) 쌍 탐색

대 값을 구하는 부분만 간단히 기술하였다. 알고리즘1에서 $w[i,j]$는 $(K_1, K_2, \ldots, K_i)$의 부분수열과 $(E_1, E_2, \ldots, E_j)$의 부분수열 중 대응도 총합이 최대가 되는 값을 나타내며, 1행에서는 $w[i,j]$의 점화식 전개를 위한 초기 값을 설정하고 있다. 2행에서 10행에 걸쳐있는 분기문에서는 $w[i,j]$값을 $w[1,1], w[1,2], \ldots, w[n,m]$ 순서로 계산한다. 4행에서 K_i와 E_j의 대응도가 입력으로 주어진 문장 대응 하한 값 τ보다 큰 지를 검사한다. $Corr(i,j)$가 τ 이상인 경우에는, $(K_1, K_2, \ldots, K_i)$와 $(E_1, E_2, \ldots, E_j)$ 간의 최대 부분순열에 K_i와 E_j가 각각 포함되는 경우를 생각하여, $w[i,j]$을 $w[i-1,j] + Corr(i,j)$와 $w[i,j-1] + Corr(i,j)$ 중 큰 값으로 결정된다. 반면, $Corr(i,j)$가 τ보다 작을 경우, K_i와 E_j를 최대 부분순열에서 배제하여 $w[i,j]$는 $w[i-1,j]$와 $w[i,j-1]$ 중 큰 값을 구한다.

알고리즘2은 알고리즘1의 결과로 정의된 최대 문장수열 쌍 $(K_{s_1}, K_{s_2}, \ldots, K_{s_k})$와 $(E_{t_1}, E_{t_2}, \ldots, E_{t_k})$에 포함된 연속된 문장수열(구간) 쌍으로 대응도의 합이 최대가 되는 한국어-영어 문장 구간 쌍을 찾는 알고리즘의 개요다(간단한 표현을 위해, 한국어-영어 대응 구간 쌍의 최대 대응도 합만을 구함). 알고리즘2는 알고리즘1과 유사한 방식으로 $(K_{s_1}, K_{s_2}, \ldots, K_{s_k})$과 $(E_{t_1}, E_{t_2}, \ldots, E_{t_k})$의 부분수열 쌍을 탐색하나, 3행과 5행의 조건식을 통해서 연속적인 부분수열만을 탐색하여, 결과적으로 연속순열 쌍 중 대응도 합이 최대인 경우를 구하게 된다.

4. 실험 설계

제안한 구간 단위 문장대응 기법이 위키피디아 문서에서 얼마만큼 효과적으로 실제 병렬문장을 탐지하는 지, 또 기존 기법과 비교하여 어떠한 성능을 보이는지 확인하기 위하여 다음과 같은 실험을 설계하였다.

- **실험 대상**: 본 실험에서는 실험 대상으로, 앞선 2장에서 조사 연구의 재료로 사용한 50개의 한국어-영어 위키피디아 문서쌍과 정답 데이터를 실험대상으로 사용하였다.
- **비교 대상**: 본 논문에서 제안한 기법과 함께 한영 위키피디아 문서로부터 병렬문장을 추출할 수 있는 다음의 2가지 기존 기법을 함께 사용하여, 결과를 비교하였다:

기법		정확도	재현도
문장 단위 대응	Jac, ε =0.01	22.5%	36.9%
	Jac, ε =0.02	26.6%	28.5%
	Jac, ε =0.03	32.9%	17.4%
	Jac, ε =0.04	33.0%	9.2%
	Jac, ε =0.05	25.5%	3.6%
	Op2	27.3%	51.8%
문장 정렬	Jac, ε =0.01	37.8%	52.3%
	Jac, ε =0.02	37.2%	50.3%
	Jac, ε =0.03	38.4%	50.8%
	Jac, ε =0.04	41.2%	48.5%
	Jac, ε =0.05	42.1%	42.8%
	Op2	41.6%	**55.6%**

기법		정확도	재현도
연속 문장 수열 대응	n=1, ε =0.01	66.7%	23.6%
	n=1, ε =0.02	66.2%	23.6%
	n=1, ε =0.03	68.0%	22.3%
	n=1, ε =0.04	**78.6%**	23.6%
	n=1, ε =0.05	72.5%	20.3%
	n=2, ε =0.01	58.4%	35.6%
	n=2, ε =0.02	56.6%	35.4%
	n=2, ε =0.03	60.8%	34.6%
	n=2, ε =0.04	62.6%	33.1%
	n=2, ε =0.05	65.4%	32.1%
	n=3, ε =0.01	51.8%	40.5%
	n=3, ε =0.02	50.5%	40.3%
	n=3, ε =0.03	53.6%	40.0%
	n=3, ε =0.04	54.3%	38.5%
	n=3, ε =0.05	59.3%	38.5%

표2. 3가지 병렬문장 추출 기법의 실험결과 결과

- 문장 단위 대응 기법: 모든 한국어-영어 문장 쌍의 대응도를 측정한 후, 각 한국어 문장에 가장 높은 대응도를 보이는 영어 문장을 대응하는 기법 [4]
- 문장 정렬 기법: 모든 한국어-영어 문장 쌍의 대응도를 측정한 값을 바탕으로, 한국어 문서 전체와 영어 문서 전체로부터 문장별 대응도 합이 최대가 되는 문장순열 쌍을 구하는 기법 [5]

- **기법의 구현**: 제안한 기법과 기존의 2가지 기법의 성능을 상호 비교하기 위해, 본 실험에서는 실험대상 위키피디아 문서에서 한영 문장 쌍의 대응도를 동일한 방식으로 계산한 데이터를 생성한 후, 각 기법은 이를 바탕으로 병렬문장 쌍을 출력하는 형태로 구현하도록 하였다. 한국어-영어 문장 쌍의 대응도는 각 문장에서 고유명사와 숫자를 특징점(feature)로 추출하고, 대역사전(dictionary)를 통해 공통된 특징점의 개수와 공통되지 않은 특징점의 개수에 따른 유사도를 Jaccard값[4]과 Op2[6]값의 두 가지 계측방법으로 측정했다.

  문장 단위 대응 기법과 문장 정렬 기법의 경우, 다음의 두 가지 방식으로 구현하여 실험에 사용하였다:
  - 한영 문장 간 대응도를 Jaccard 값으로 측정한 후, ε 이상의 대응도를 갖는 한영 문장 쌍으로부터 병행문장을 추출. 이 때 ε 값으로 0.01, 0.02, 0.03, 0.04, 0.05를 사용함(사전조사 실험(pilot study) 결과를 바탕으로 파라미터 선정)
  - 한영 문장 간 대응도를 Op2 값으로 측정하여 사용함

  본 논문에서 제안한 구간 단위 대응 기법의 경우, 다음과 같이 파라미터를 다양하게 하여 실험에 사용함
  - 한영 문장 간 대응도를 Jaccard 값으로 측정한 후, ε 이상의 대응도를 갖는 한영 문장 쌍으로부터 병행문장을 추출. 이 때 ε 값으로 0.01, 0.02, 0.03, 0.04, 0.05를 사용함.
  - 구간 추출 횟수(3장의 과정3) n로 1, 2, 3을 사용함 (ε 와 독립적으로 조정함).
- **측정**: 구현된 각 기법을 50쌍의 실험대상 한영 문서 쌍에 적용하여 병렬문장 쌍의 집합을 각각 추출하였다. 추출된 각 병렬문장 쌍 집합을 정답 문장 쌍 집합과 유사도를 비교하였다. 유사도는 구체적으로 다음 두 가지 값을 측정한다:
  - 정확도(precision): 각 기법이 추출한 병렬문장 쌍 중 정답에 포함되는 문장 쌍의 비율
  - 재현도(recall): 정답의 문장 쌍 중 각 기법이 추출한 문장 쌍의 비율

5. 실험 결과

표2는 50개의 실험대상 한영 위키피디아 문서 쌍에 대하여 총 27개의 병렬문장 추출 기법을 적용한 결과다. 표2의 좌측/우측의 두 번째 열은 정확도(precision), 세 번째 열은 재현도(recall)를 각각 나타낸다. 표2의 좌측 2-7행은 문장 단위 대응 기법에 해당하며, 2-6행은 Jaccard값을 사용한 경우, 7행은 Op2값을 사용한 경우의 결과다. 문장정렬로 표시된 8행과 13행은 각각 Jaccard값(8-12행)과 Op2값(13행)을 바탕으로 병렬문장을 추출한 결과다. 표2의 우측 2~16행은 본 논문이 제안한 구간 단위 대응 기법의 결과다.

표2의 실험결과로부터, 실험에 사용된 총 27개의 병렬문장 추출

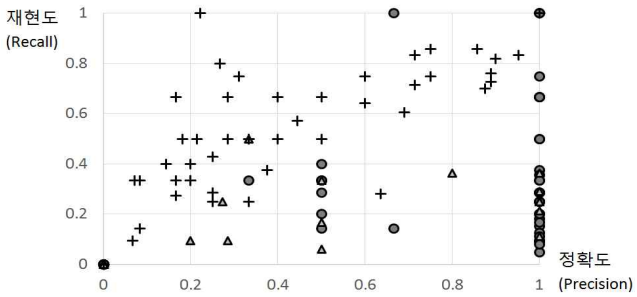

그림1: 3가지 기법의 문서별 정확도, 재현도의 분산

기법 중, 본 논문에서 제안한 연속 문장수열 대응 방식으로 n=1, ε =0.04의 파라미터 값으로 설정된 기법이 가장 높은 정확도(78.6%)를 보였음을 알 수 있다. 전체적으로 볼 때, 본 논문에서 제안한 연속 문장수열 대응 기법의 정확도는 파라미터에 따라서 최소 50.5%, 최대 78.6%으로, 문장 단위 대응 기법(최소 22.5%, 최대 33.0%)과 문장정렬 기법(최소 37.2%, 최대 42.1%)을 상회하는 결과를 보였다. 구문 대응으로 파라미터를 다르게 한 15개의 기법에서는, 전반적으로 추출 횟수가 작을수록 높은 정확도를 보였으며, Jaccard값이 0.04 이상인 한영 문서 쌍을 활용한 경우가 그 외의 경우보다 항상 높은 정확도로 병렬문장 추출을 달성하였다.

재현도 결과의 경우, Op2값을 이용한 문장정렬이 가장 높은 재현도(55.6%)를 달성하였음을 알 수 있다. 본 논문에서 제안한 연속 문장수열 대응 기법의 경우, 재현도가 최소 20.3%(n=1, ε =0.05의 경우), 최대 40.5%(n=3, ε =0.01의 경우)로, 문장 정렬 기법보다는 다소 낮은 결과 재현도를 보였다. 연속 문장수열 대응에 해당하는 15개 기법에서는, 추출 횟수가 증가하고 최소 Jaccard값 기준(threshold)이 낮아질수록, 재현도가 증가함을 볼 수 있다. 이와 유사하게, 문장 단위 대응 기법에서도 최소 Jacard값 기준이 낮아짐에 따라, 재현도가 최대 36.9%까지 증가함을 볼 수 있다.

그림1은 50개 실험대상 한영 문서 쌍 별로 3가지 종류의 병렬문장 추출기법이 달성한 정확도와 재현도의 분포를 표현하고 있다. 그림1의 횡축은 정확도를 나타내며, 종축은 재현도를 나타내며, 각 점은 한 기법의 특정 한영 문서 쌍에 대한 결과다. 그림1에서 십자 표식의 점은 최소 Jaccard값이 0.04인 한영 문장쌍을 바탕으로 한 문장 단위 대응 기법에 해당하는 결과를 뜻하며, 세모 표식의 점은 Op2를 사용한 문장 정렬 기법, 그리고 동그라미 표식은 구문 단위 대응 기법 중 Jaccard값 0.04 이상의 한영 문장 쌍을 이용한 기법의 결과다. 그림1에서 볼 수 있는 듯이, 본 논문에서 제안한 구문 단위 기법(동그라미 표식)이 50개 실험대상 전반에 대하여, 다른 두 개의 기법보다 높은 정확도를 달성하는 방향으로 병행문장을 추출하였음을 확인할 수 있다.

6. 결론

본 논문에서는 한국어 위키피디아 본문이 구간 단위로 영어 본문과 대응 관계가 높은 경우가 많다는 관찰에 기반 하여, 구문 단위 대응을 통한 보다 정확한 한국어-영어 병렬문장 추출 기법을 제안하였다. 실제 한국어-영어 위키피디아 문서 50쌍을 이용한 실험에서, 본 논문이 제안한 구간 단위 문장대응 방식은 기존의 문장 단위 대응 기법이나 문서 정렬 기법보다 8.4~56.1% 높은 정확도로 병렬문장을 추출하여, 다종언어 위키피디아 자원으로부터 한국어 병렬문장 말뭉치 생성에 효과적임을 확인할 수 있었다.

참조문헌

[1] K. Cho et al, Learning Phrase Representation using RNN encoder-decoder for Statistical Machine Translation, EMNLP, 2014

[2] 세종 코퍼스 https://ithub.korean.go.kr/user/guide/corpus/guide1.do

[3] 배경만, 김성호, 천주룡, 고영중, 위키피디아 기반의 의미분석을 위한 언어 자원 구축, 정보과학회지, 제 34권, 제 8호, 2016

[4] 김성현, 양선, 고영중, 위키피디아로부터 한국어-영어 병렬 문장 추출, 정보과학회논문지: 소프트웨어 및 응용, 제 41권, 제 8호, 2014

[5] 홍진표, 차정원, 길이 및 어휘 정보와 번역 모델을 이용한 한영 문장 정렬, 정보과학회지, 소프트웨어 및 응용, 제 40권, 제 8호, 2013

[6] L. Naish et al., A model for spectra-based software diagnosis, TOSEM, 20(3):11, 2011