부트스트래핑 기반의 단어-임베딩 투영 학습에 의한 대역어 사전 구축

이종서[O†], 왕지현[††], 이승진[††]

KAIST 전산학부[†]
NLP Center Language AI Lab, ㈜ 엔씨소프트[††]
leejseo@kaist.ac.kr, {korjhwang, sjlee927}@ncsoft.com

Bootstrapping-based Bilingual Lexicon Induction by Learning Projection of Word Embedding

Jongseo Lee[O†], JiHyun Wang[††], Seung Jin Lee[††]
School of Computing, KAIST[†]
NLP Center Language AI Lab, NCSOFT Corp[††]

요 약

대역사전의 구축은 저자원 언어쌍 간의 기계번역의 품질을 높이는데 있어 중요하다. 대역사전 구축을 위해 기존에 제시된 방법론 중 단어 임베딩을 기반으로 하는 방법론 대부분이 영어-프랑스어와 같이 형태적 및 구문적으로 유사한 언어쌍 사이에서는 높은 성능을 보이지만, 영어-중국어와 같이 유사하지 않은 언어쌍에 대해서는 그렇지 못하다는 사실이 널리 알려져 있다. 본 논문에서는 단어 임베딩을 기반으로 부트스트래핑을 통해 대역사전을 구축하는 방법론을 제안한다. 제안하는 방법론은 소량의 seed 사전으로부터 시작해 반복적인 과정을 통해 대역사전을 자동으로 구축하게 된다. 이후, 본 논문의 방법론을 이용해 한국어-영어 언어쌍에 대한 실험을 진행하고, 기존에 대역사전 구축 용도로 많이 활용되고 있는 도구인 Moses에 사용된 방법론과 F1-Score 성능을 비교한다. 실험 결과, F1-Score가 약 42%p 증가함을 확인할 수 있었으며, 초기에 입력해준 seed 사전 대비 7배 크기의 대역사전을 구축하였다.

주제어: 부트스트래핑, 대역사전, 이중 언어 사전, 사전 구축, 단어 임베딩, 기계번역

1. 서론

고품질의 기계번역 성능을 위해서는 대량의 병렬 코퍼스 확보가 매우 중요하다. 서유럽이나 영어권의 병렬 코퍼스 획득은 수월한 편이지만 그 외의 대다수의 언어는 병렬 코퍼스 획득이 쉽지 않은 것이 현실이다. 이와 같은 저자원 언어(low resource language)의 번역 문제를 보완하기 위해 전이 학습 기반의 번역 방법[1,2,3], 소량의 병렬 코퍼스에 대역사전을 이용한 번역 방법[4,5,6] 등이 있으며, 단일언어 코퍼스로부터 기계번역 학습을 위한 병렬 코퍼스를 자동 구축하는 시도도 있다[7,8].

병렬코퍼스를 자동 구축하는 연구는 번역 대상인 두 개 언어의 유사한 문서 집합 내에서 문장 간의 유사성을 측정하여 문장쌍을 추출하는 방법이 있다[8]. 이 방법은 유사성을 측정하기 위해서 두 문장을 구성하는 단어들의 정렬을 시도하기도 하며, 두 문장의 길이 차 비교, 단어의 어원 유사성 측정, 통계적 단어 확률분포 측정, 별도의 대역 사전의 대역어 포함 여부 검사 등을 수행한다. 또 다른 방법으로 소스 언어와 타겟 언어의 단어 또는 문장을 각 언어의 벡터 공간에 각각 임베딩한 후, 타겟 언어의 벡터 공간 또는 동일 벡터 공간으로 투영(projection)하여 두 언어의 문장 간 유사성 계산 문제를 벡터 간 유사성 문제로 변환하는 방법이 있다.

형태적, 구문적으로 유사한 언어쌍(similar language pairs) 간에 유사성을 비교할 때에는 비록 소스 또는 타겟 언어 중 어느 한쪽의 언어가 상대적으로 코퍼스 양이 부족하더라도 학습의 단위가 되는 단어 또는 서브워드(sub-word)의 형태가 유사하여 어휘사전과 문맥을 공유하는 정도가 높기 때문에 성능 향상을 얻을 수 있다. 그러나 형태적, 구문적으로 서로 다른 언어쌍(distant language pairs)간에 유사성을 비교할 때에는 서로의 단일언어 코퍼스만으로는 유사성을 측정하기 어렵기 때문에 대역사전 등의 외부 지식을 이용하는 것이 필요하다[9]. 대역사전은 유사성이 부족한 두 언어 간의 대역 표현을 동일 벡터 공간 위에 투영되도록 학습하는 과정에서 anchor 역할을 하게 됨으로써 두 언어 사이의 부족한 유사성을 크게 보완하는 역할을 하기 때문에 저자원 문제를 보완하는 중요한 역할을 한다.

대역사전을 구축하는 방법은 통계적 확률분포로부터 대역 단어 쌍을 추출하는 비지도 학습 기반의 방법들[10,11]이 우세하였으나, 최근에는 신경망 연구가 활발해짐에 따라 적대적 신경망 학습을 기반으로 한 연구도 보고되고 있다[12,13,14]. 또한, 기 보유한 소량의 대역사전을 활용하여 추가적인 대역 쌍을 추출하는 지도 학습 기반의 연구도 있다[12,15,16]. 지도 학습 기반의 대역사전 구축 연구는 Ground-Truth seed 사전을 통해 동일한 의미를 갖는 대역 단어 쌍들을 입력하게 되며, 이

들 단어 쌍이 최대한 같은 벡터 공간 상에 위치하게 하기 위하여 두 단어 벡터 간의 거리가 최소화되도록 학습한다[12,15].

본 논문은 형태적, 구문적으로 서로 다른 언어쌍의 대역사전 구축에 효과적인 지도 학습 방법을 제안한다. 특히, 어휘가 서로 다른 언어쌍의 언어들이 동일한 벡터 공간의 동일한 위치에 최대한 많이 투영되도록 하기 위해서 소량의 seed 대역사전으로부터 점진적으로 사전을 확장하는 부트스트래핑(bootstrapping) 방법을 도입하였다.

유사하지 않은 언어쌍인 한국어-영어 언어쌍에 대한 실험을 통해 성능을 정량적으로 평가하고, [11]과 같이 널리 활용되고 있는 기존의 대역사전 구축 방법론에 비해 의미있는 성능 개선이 있는지를 정량적으로 평가한다.

2. 관련 연구

2.1 MUSE

MUSE(Multilingual Unsupervised and Supervised Embeddings)는 [12]에서 제시된 방법론의 공식 구현체[17]로 소스 언어와 타겟 언어의 단어 임베딩으로부터 소스 언어의 임베딩 공간에서 타겟 언어의 임베딩 공간으로의 projection을 구한다. MUSE는 projection을 구하는 방법에 따라 지도 학습에 기반한 MUSE-supervised와 비지도 학습에 기반한 MUSE-unsupervised로 나누어진다. MUSE-supervised는 Procrustes Problem에, MUSE-unsupervised는 적대적 신경망에 기반을 두고 있다. 본 연구에서는 형태가 서로 다른 언어쌍의 대역사전 추출에 유리한 MUSE-supervised를 사용하였다.

MUSE-supervised는 소스 언어의 단어 임베딩 X와 타겟 언어의 단어 임베딩 Y, 그리고 대역 단어 쌍을 담은 seed 사전 $D = \{(src_1, tgt_1), \dots, (src_n, tgt_n)\}$을 입력으로 받으며, 다음의 과정을 수행한다.

1. $X_D = [x_1, x_2, \dots, x_n]$과 $Y_D = [y_1, y_2, \dots, y_n]$을 D와 X, Y로부터 구성한다. 이 때, x_i와 y_i는 각각 대역 단어 쌍 (src_i, tgt_i)에 대응되는 소스 및 타겟 언어의 단어 임베딩이다.
2. $W^* := \arg\min_{w \in O_d(\mathbb{R})} ||WX_D - Y_D||$를 구한다. (단, d는 임베딩의 차원을 의미한다.)
   - 이 때, W^*를 계산하는 문제는 Orthogonal Procrustes Problem으로 잘 알려져 있으며, $U\Sigma V^T = SVD(YX^T)$에 대해 $W^* = UV^T$와 같이 SVD를 이용해 닫힌 형태의 해를 계산할 수 있다[18].
3. 이후, Cross-domain similarity local scaling 작업을 통해 2에서 계산한 projection W^*를 refine 한다.

[12]에 제시된 각 언어쌍 별 MUSE-supervised의 성능은 그림 1과 같다. 형태적 및 구문적으로 비교적 유사한 언어쌍인 영어-스페인어 및 영어-프랑스어 에서의 성능과 비교적 덜 유사한 언어쌍인 영어-러시아어 및 영어-중국어 에서의 성능의 차이가 나타남을 확인할 수 있다.

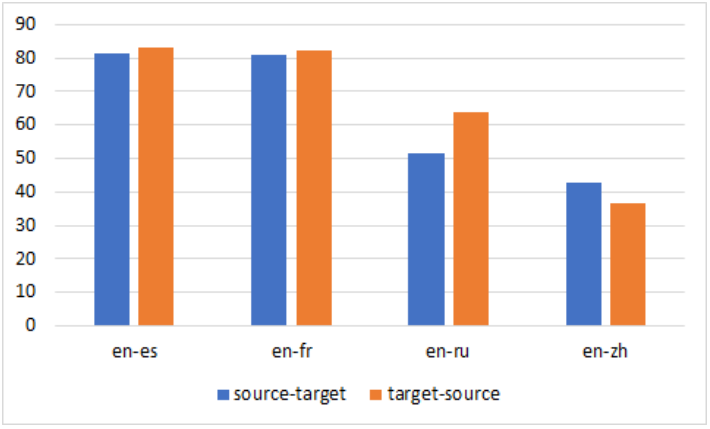

그림 1. MUSE-supervised의 언어쌍별 precision

2.2 부트스트래핑

[19]에서는 레이블이 없는 데이터로 모델을 반복적으로 개선해 나가는 부트스트래핑 방법론을 제안하였다. 부트스트래핑이 이루어지는 과정은 initialization 단계와 iteration 단계로 나누어 볼 수 있다.

Initialization 단계에서는 적은 수의 데이터에 직접 레이블을 붙여 seed 데이터를 구성하게 된다. Iteration 단계는 여러 방식으로 구성할 수 있는데, 예를 들어, (1) 레이블을 통한 모델의 개선과 (2) 개선된 모델을 통한 재 레이블링이 반복적으로 이루어지게 구성할 수 있다. 부트스트래핑 기법을 적용한 사례로는 부트스트래핑 기반의 병렬 문장 추출 시스템[20] 등이 있다.

3. 제안 방법론

본 연구는 소스 언어 임베딩 공간과 타겟 언어 임베딩 공간 각각의 서로 대응되는 점들(anchor points)을 최대한 많이 mapping 하고자 하였다. 이를 위해 주된 접근방법은 부트스트래핑 방법을 적용하여 feed 되는 외부 대역사전의 대역 쌍의 수를 점진적으로 증가시키는 것이다. 우리는 그림 2와 같은 부트스트래핑 기반의 반복적인 대역사전 구축 방법론을 제안한다.

우리는 이로부터 iteration이 반복되며 anchor point의 수가 증가함에 따라 projection matrix W^*가 보다 정교화되어 projection 된 소스 언어의 임베딩 공간 WX와 타겟 언어의 임베딩 공간 Y가 더 유사해지기를 기대하였다. 또한, 부트스트래핑 과정에서의 결과물로 초기 seed 사전보다 유의미하게 많은 대역 단어 쌍을 담고 있는 사전을 얻어내는 것도 기대하였다.

덧붙여, 한국어-영어 언어 쌍에 대해 MUSE-supervised

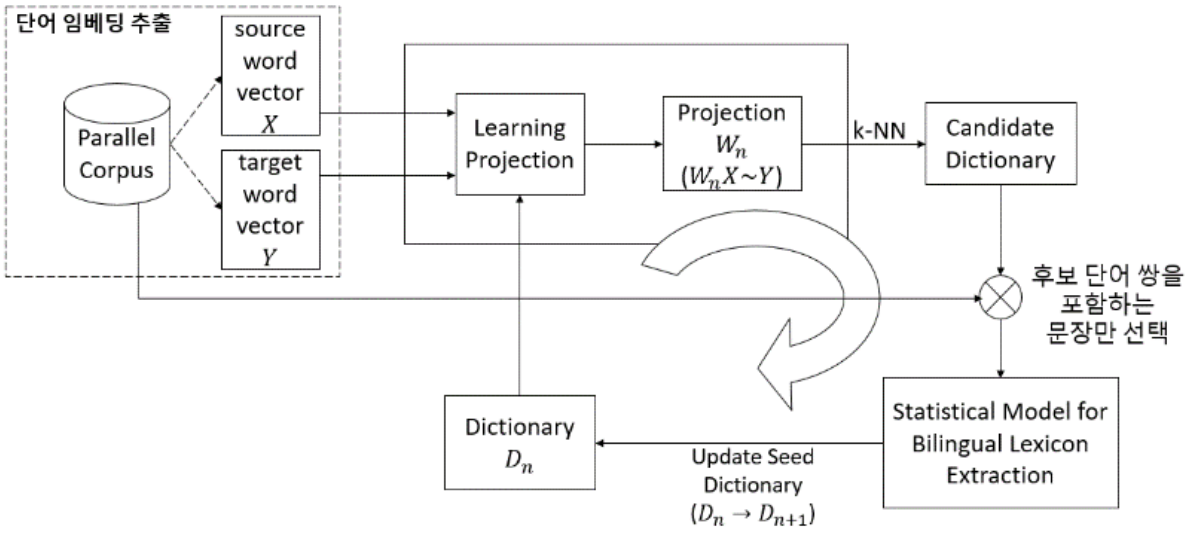

그림 2. 부트스트래핑 기반 대역사전 자동 구축 프로세스

대비 정밀도 등의 성능 수치 향상을 기대하였고, [11]과 같이 널리 사용되는 word-alignment 기반 대역사전 구축 도구 대비 유의미하게 향상된 성능도 기대하였다.

우리의 방법론이 작동하는 구체적인 과정은 다음과 같다.

- Initialization 단계
  1. 코퍼스로부터 소스 언어의 단어 임베딩 X와 타겟 언어의 단어 임베딩 Y를 구한다.
  2. 초기 seed 사전 D_0를 입력한다.

- Iteration 단계: n번째 iteration
  1. MUSE-supervised로 Procrustes Problem을 해결하여 X, Y, D_n으로부터 소스 언어 임베딩 공간에서 타겟 언어 임베딩 공간으로의 projection W_n을 학습한다.
  2. 소스 언어 단어 벡터의 projection W_nX와 타겟 언어 단어 벡터 Y에 대해 mutual k-NN 기법을 적용하여 대역 단어 쌍들을 얻어낸다.
  3. 대역 단어 쌍들을 필터링하는 과정을 거쳐 사전을 업데이트($D_n \rightarrow D_{n+1}$) 한다.

여기에서, mutual k-NN은 다음과 같이 정의된다:

- 벡터 공간 상의 두 벡터들의 집합 $V = \{v_1, v_2, \dots, v_n\}$와 $W = \{w_1, w_2, \dots, w_m\}$이 주어질 때, $\{(v_i, w_j) | w_j \in kNN(v_i) \ and \ v_i \in kNN(w_j)\}$.

필터링 단계에서는 단어 임베딩만을 사용했을 때 발생할 수 있는 노이즈를 제거해줌을 목표로 한다. 필터링 단계는 소량의 병렬 코퍼스에 적절한 척도를 적용하여 대역 단어 쌍의 점수를 계산하고, 일정 수치 이상의 대역 단어 쌍만을 선택하는 방법으로 구현할 수 있다.

본 연구에서는 필터링 단계에서 대역 단어 쌍에 대한 점수를 계산함에 있어 [21]에서 제안된 CPE(Controlled Predictive Effect) 척도를 활용하였다. CPE 척도는 기존에 많이 활용되던 PMI 척도를 개선한 척도이다. 소스 언어의 단어들의 집합 X와 소스 언어 단어 x 및 타겟 언어 단어 y에 대해 CPE 척도는 다음과 같이 정의된다.

$$CPE(y \mid x) = p(y \mid x) - \sum_{x' \in X} p(y \mid x')p(x' \mid x)$$

이 때, 조건부 확률 $p(y \mid x)$의 계산에는 상호 출현 횟수 및 출현 횟수를 다음과 같이 활용하게 된다. 단, $\#(x)$는 x가 출현하는 횟수이고, $\#(x,y)$는 소스 언어 문장에는 x가 출현하고, 이에 대응되는 타겟 언어 문장에는 y가 출현하는 횟수이다.

$$p(y \mid x) = \frac{p(x,y)}{p(x)} \approx \frac{\#(x,\ y)}{\#(x)}$$

4. 실험

본 장에서는 성능을 평가하기 위한 실험 방법 및 실험 결과를 다룬다.

4.1 실험 데이터

실험에는 자사 내부적으로 보유하고 있는 크기 20만의 한-영 병렬 코퍼스와 크기 7,798의 한-영 Ground-Truth 사전을 이용하였다. Ground-Truth 사전은 80:20 비율로 분할하여 각각 seed 사전 및 평가 데이터 셋으로 활용하였다. 병렬 코퍼스 및 사전 모두 형태소 분해하여 사용하였으며, 형태소 분해 시 한국어는 kmat[22], 영어는 Porter Stemmer[23]를 이용하였다.

실험에 사용한 병렬 코퍼스의 경우 약 6만 2천개의 한국어 단어와 약 3만 9천개의 영어 단어가 포함되어 있다.

4.2 실험 구성

제안한 방법론과의 비교 대상이 되는 baseline으로는 다음의 두 가지 방법론을 선택하였다.

- MUSE-supervised + mutual k-NN
  - k: 5, 10
- Moses: *lex(target word | source word)* 값이 cut-off 이상인 (소스 언어 단어, 타겟 언어 단어) 쌍 만을 선택
  - cut-off: 0.6, 0.5, 0.4, 0.3

MUSE-supervised + mutual k-NN 방법론은 그림 3의 과정을 따라 실험하였으며, MUSE-supervised + mutual k-NN을 이용한 실험과 본 연구의 방법론을 이용한 실험 모두 같은 단어 임베딩과 같은 seed 사전 데이터를 활용하였다.

단어 임베딩의 경우 형태소 분해된 코퍼스의 소스 및 타겟 문장들로부터 [24]의 공식 구현체인 fastText[25]의 skip-gram 모델을 사용해 생성하였다. 또한, k-NN을 효율적으로 계산하기 위해 두 실험 모두에서 GPU 기반의 nearest neighbor 탐색 알고리즘[26]의 구현체인 faiss[27]를 사용하였다.

MUSE-supervised + mutual k-NN을 이용한 실험의 구체적인 과정은 다음과 같다.

1. MUSE-supervised를 통해 소스 언어의 단어 임베딩 X, 타겟 언어의 단어 임베딩 Y, 그리고 seed 사전 D로부터 projection W를 학습한다.
2. 소스 언어 단어 벡터의 projection WX와 타겟 언어 단어 벡터 Y에 대해 mutual k-NN 기법을 적용하여 결과물 사전을 얻는다.

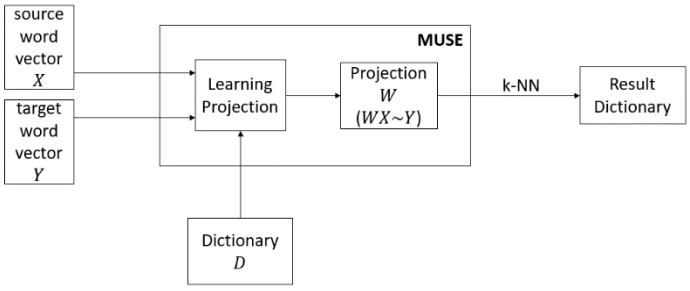

그림 3. MUSE-supervised + k-NN(baseline) 방법론을 이용한 실험의 과정

성능 평가의 대상이 되는 또 다른 방법론인 Moses의 경우에는 형태소 분해된 병렬 코퍼스를 입력했으며, cut-off 값을 0.1 단위로 변화시켜 가며, *lex(target word | source word)* 값이 cut-off 이상인 단어들만 택해 얻은 결과물 대역사전을 이용하여 성능을 측정하였다.

4.3 성능 평가 척도

모든 실험에서 성능 평가를 위해 Ground-Truth 사전을 통해 만든 공통적인 평가 데이터 셋을 활용하였다. 정량적인 성능 평가를 위한 척도로는 각 실험에서의 결과물 사전의 평가 데이터 셋에 대한 정밀도(Precision)와 재현율(Recall), 그리고 F1-Score를 활용하였다.

4.4 실험 결과

먼저, 제안한 방법론인 (1) Bootstrapping + MUSE + 10-nn과 baseline 방법론인 (2) MUSE + 5-nn, (3) MUSE + 10-nn, 총 3가지 방법 각각의 정밀도, 재현율 및 F1-Score는 그림 4와 같다.

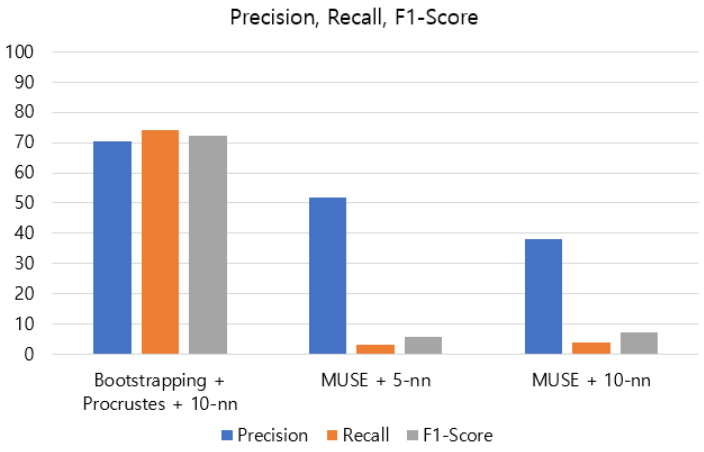

그림 4. MUSE-supervised와의 성능 비교: 정밀도, 재현율 및 F1-Score

우리는 부트스트래핑을 통해 보다 정교한 projection을 구할 수 있었다. 정밀도의 경우 70.5%로 MUSE-supervised + mutual 5-nn의 51.7%에 비해 약 19%p가 증가하였고, 재현율의 경우에도 74.0%로 유의미하게 개선되었다. 그리고 그로 인해 F1-Score 척도가 증가함을 확인할 수 있었다.

성능 비교를 위해 사용한 방법론인 Moses의 접근방법의 성능 비교 실험의 결과는 그림 5와 같다. Moses의 경우, *lex(target word | source word)*의 cut-off를 0.6부터 시작해 0.3까지 0.1단위로 낮추어 가며 cut-off 이상의 대역 단어쌍들만을 이용해 대역사전을 구축한 후 성능을 평가하였다.

Moses로 구축한 대역사전 중 가장 F1-Score가 높은 Moses-0.3과 비교했을 때, F1-Score가 약 42%p 증가함을 확인할 수 있었다. 그리고 Moses로 구축한 대역사전 중 가장 정밀도가 높은 Moses-0.6에 비해서도 높은 정밀도를 달성함을 확인할 수 있었다.

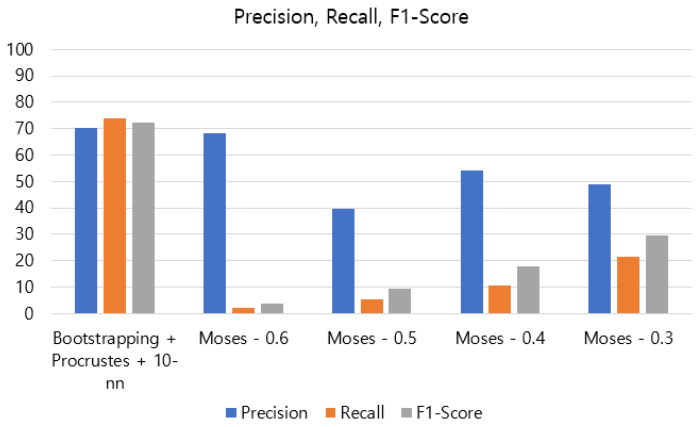

그림 5. Moses와의 성능 비교: 정밀도, 재현율 및 F1-Score

실험 과정에서 얻은 실제 결과물 사전에서 무작위로 추출한 샘플 대역 단어쌍의 예는 표 1과 같다. 결과물 사전으로부터 총 16개의 샘플 대역 단어 쌍을 추출하였다.

표 1. 결과물 사전의 대역 단어쌍 예시

한국어	영어	한국어	영어
케어	care	독서	read
캠프	camp	상실	loss
분기	quarter	조립	assembl
관행	trial	환자	patient
결혼	marri	빗썸	kdb
유니세프	unicef	성탄절	christma
탑승객	passeng	삭감	cut
책무	duti	히데키	tg

Iteration 횟수에 따른 결과물 사전의 크기 (대역 단어 쌍 개수)는 그림 6과 같이 나타났다. 약 6천개의 대역 단어 쌍을 담은 초기 seed 사전을 입력하고 13번의 iteration 후 약 4만 4천개의 대역 단어 쌍을 담은 사전을 얻을 수 있었다.

Iteration 횟수 별 대역 단어 쌍의 증가 폭은 그림 7과 같다. 초기에 큰 폭으로 증가하고, 이후 증가 폭이 감소하며 대역 단어 쌍의 개수가 수렴함을 확인할 수 있었다.

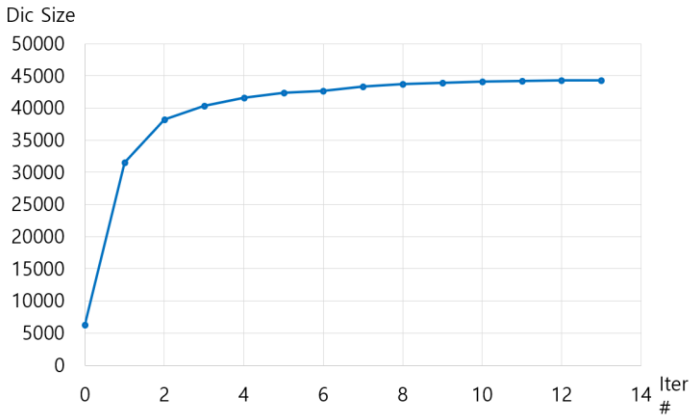

그림 6. Iteration 횟수에 따른 결과물 사전의 대역 단어 쌍 개수

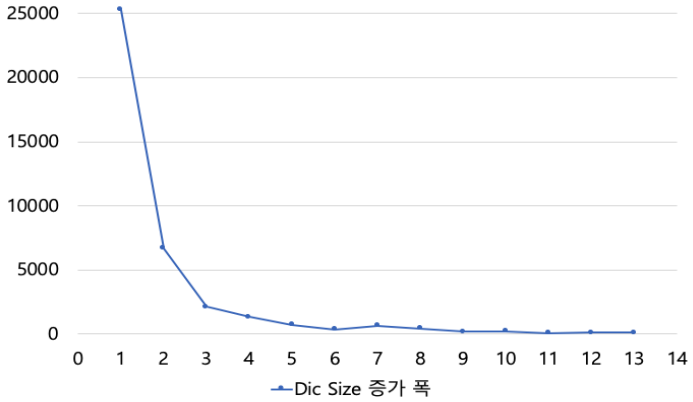

그림 7. Iteration 횟수에 따른 대역 단어 쌍 개수의 증가 폭

5. 결론

본 연구에서는 distant한 언어쌍의 번역 모델에 활용할 수 있는 대역사전을 자동으로 구축하는 방법론을 제안하였다. Distant한 언어쌍은 형태적, 구문적 유사도가 낮아 비지도 학습으로 좋은 성능을 내기 힘들기 때문에 Procrustes Problem 기반의 지도 학습 기법을 활용하였다. 그리고 소스 언어와 타겟 언어의 공통된 임베딩 공간에서 최대한 많은 점들의 쌍이 서로 mapping되게 하고자 하였다. 이를 위해 부트스트래핑 기법을 이용해 feed되는 대역사전의 크기를 반복적으로 증가시켰다.

본 연구의 방법론이 MUSE만 활용하는 것에 비해 실제 성능이 개선되는지를 확인하였다. 또한, 대역사전 구축 용도로 많이 사용되는 기존의 방법론에 비해 유의미한 성능 개선을 보이는지도 실험을 통해 평가하였다.

실험 결과, 성능 평가의 baseline 방법론인 Moses의 방법론 대비 F1-Score가 약 42%p 증가함을 확인할 수 있었고, 차용한 방법론인 MUSE-supervised + mutual k-NN 대비 정밀도가 약 19%p 증가함도 확인할 수 있었다. 덧붙여, 초기 seed 사전 대비 약 7배의 대역 단어 쌍을 추출하였다.

참고문헌

[1] T. Kocmi and O. Bojar, “Trivial Transfer Learning for Low-Resource Neural Machine Translation”, Proceedings of the Third Conference on Machine Tranlation: Research Papers, pp. 244-252, 2018.

[2] T. Q. Nguyen and D. Chiang, “Transfer Learning across Low-Resource, Related Languages for Neural Machine Translation”, Proceedings of the 8th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 296-301, 2017.

[3] B. Zoph et al, “Transfer Learning for Low-Resource Neural Machine Translation”, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1568-1575, 2016.

[4] S. H. Ramesh and K. P. Sankaranarayanan, “Neural Machine Translation for Low Resource Languages using Bilingual Lexicon Induced from Comparable Corpora”, arXiv preprint arXiv:1806.09652, 2018.

[5] A. Irvine and C. C. Burch, “Combining Bilingual and Comparable Corpora for Low Resource Machine Translation”, Proceedings of the Eighth Workshop on Statistical Machine Translation, pp. 262-270, 2013.

[6] A. Imankulova et al, “Improving Low-Resource Neural Machine Translation with Filtered Pseudo-Parallel Corpus”, Proceedings of the 4th Workshop on Asian Translation, pp. 70-78, 2017.

[7] Munteanu D.S., Marcu D. "Improving machine translation performance by exploiting non-parallel corpora", Computational Linguistics. 2005.

[8] C Hoang et al, “Exploiting Non-Parallel Corpora for Statistical Machine Translation”, IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future, 2012.

[9] S. Ruder et al, “Unsupervised Cross-Lingual Representation Learning”, Proceedings of the 57th annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, 2019.

[10] E. Morin and E. Prochasson, “Bilingual Lexicon Extraction from Comparable Corpora Enhanced with Parallel Corpora”, Proceedings of the 4th Workshop on Building and Using Comparable Corpora, 49th Annual Meeting of the Association for Computational Linguistics, pp. 27-34, 2011.

[11] P. Koehn et al, “Moses: Open Source Toolkit for Statistical Machine Translation”, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 177-180, 2007.

[12] A. Conneau et al, “Word Translation without Parallel Data”, 6th International Conference on Learning Representations, 2018.

[13] M. Zhang et al, “Adversarial Training for Unsupervised Bilingual Lexicon Induction”, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 1959-1970, 2017.

[14] R. Xu et al, “Unsupervised Cross-lingual Transfer of Word Embedding Spaces”, Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 2465-2474, 2018.

[15] T. Mikolov, Q. V. Le, and I. Sutskever, “Exploiting Similarities among Languages for Machine Translation”, arXiv preprint arXiv:1309:4168, 2013.

[16] X. Duan et al, “Bilingual Dictionary Based Neural Machine Translation without Using Parallel Sentences”, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 1570-1579, 2020.

[17] https://github.com/facebookresearch/MUSE

[18] J. Gower and G. Dijksterhuis, Procrustes Problems. Oxford University Press, 2004.

[19] E. Riloff, “Bootstrapping for Text Learning Tasks”, IJCAI-99 Workshop on Text Mining: Foundations, Techniques, and Applications, pp. 52-63, 1999.

[20] P. Fung, P. Cheung, “Mining Very-non-parallel Corpora: Parallel Sentence and Lexicon Extraction via Bootstrapping and EM”, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 57-63, Jul. 2004.

[21] Y. J. Choe, K. Park, and D. Kim, “word2word: A Collection of Bilingual Lexicons for 3,564 Language Pairs”, Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020), 2020.

[22] 이도길, “한국어 형태소 분석과 품사 부착을 위한 확률 모형”, 박사 학위 논문, 고려대학교, 2005.

[23] M. F. Porter, “An Algorithm for Suffix Stripping”, Program, Vol. 14, No. 3, pp. 130-137, 1980.

[24] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, “Enriching Word Vectors with Subword Information”, Transactions of the Association for Computational Linuistics, Vol. 5, pp. 135-146, 2017.

[25] https://github.com/facebookresearch/fastText

[26] J. Johnson, M. Douze, and H. Jegou, “Billion-scale Similarity Search with GPUs”, arXiv preprint arXiv:1702.08734, 2017.

[27] https://github.com/facebookresearch/faiss