JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 19, No. 6, pp. 1161-1167, Jun. 2018

숫자 기호화를 통한 신경기계번역 성능 향상

강청웅 · 노영헌 · 김지수 · 최희열*
한동대학교 전산전자공학부

Symbolizing Numbers to Improve Neural Machine Translation

Cheongwoong Kang · Youngheon Ro · Jisu Kim · Heeyoul Choi*
School of Computer Science and Electrical Engineering, Handong Global University, Pohang 37554, Korea

[요 약]

기계 학습의 발전은 인간만이 할 수 있었던 섬세한 작업들을 기계가 할 수 있도록 이끌었고, 이에 따라 많은 기업체들은 기계 학습 기반의 번역기를 출시하였다. 현재 상용화된 번역기들은 우수한 성능을 보이지만 숫자 번역에서 문제가 발생하는 것을 발견했다. 번역기들은 번역할 문장에 큰 숫자가 있을 경우 종종 숫자를 잘못 번역하며, 같은 문장에서 숫자만 바꿔 번역할 때 문장의 구조를 완전히 바꾸어 번역하기도 한다. 이러한 문제점은 오번역의 가능성을 높이기 때문에 해결해야 될 사안으로 여겨진다. 본 논문에서는 Bidirectional RNN (Recurrent Neural Network), LSTM (Long Short Term Memory networks), Attention mechanism을 적용한 Neural Machine Translation 모델을 사용하여 데이터 클렌징, 사전 크기 변경을 통한 모델 최적화를 진행하였고, 최적화된 모델에 숫자 기호화 알고리즘을 적용하여 상기 문제점을 해결하는 번역 시스템을 구현하였다. 본 논문은 데이터 클렌징 방법과 사전 크기 변경, 그리고 숫자 기호화 알고리즘에 대해 서술하였으며, BLEU score (Bilingual Evaluation Understudy score) 를 이용하여 각 모델의 성능을 비교하였다.

[Abstract]

The development of machine learning has enabled machines to perform delicate tasks that only humans could do, and thus many companies have introduced machine learning based translators. Existing translators have good performances but they have problems in number translation. The translators often mistranslate numbers when the input sentence includes a large number. Furthermore, the output sentence structure completely changes even if only one number in the input sentence changes. In this paper, first, we optimized a neural machine translation model architecture that uses bidirectional RNN, LSTM, and the attention mechanism through data cleansing and changing the dictionary size. Then, we implemented a number-processing algorithm specialized in number translation and applied it to the neural machine translation model to solve the problems above. The paper includes the data cleansing method, an optimal dictionary size and the number-processing algorithm, as well as experiment results for translation performance based on the BLEU score.

색인어 : 신경 기계 번역, 숫자 번역, 오번역, 기호화, 모델 최적화
Key word : Neural Machine Translation, Number Translation, Mistranslation, Symbolization, Model Optimization

http://dx.doi.org/10.9728/dcs.2018.19.6.1161

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 20 May 2018; **Revised** 24 June 2018
Accepted 25 June 2018

***Corresponding Author; Heeyoul Choi**

Tel: +82-54-260-1303
E-mail: hchoi@handong.edu

Ⅰ. 서 론

기계 학습의 발전을 통해 추천 등의 기계학습 알고리즘은 기업에서 매출 향상에 활용한지 오래 되었고, 현재도 계속해서 발전하고 있다 [1], [2]. 최근 기계학습 분야의 일종인 딥러닝 기술의 고도화를 통해 이전엔 인간만이 할 수 있었던 음성인식이나 이미지 인식등 섬세한 작업들을 기계가 할 수 있게 되었고, 특정 분야에서는 인간의 수준을 뛰어 넘는 성능을 보이기도 한다 [3], [4], [5], [6]. 기계 번역은 이러한 분야 중 하나로써 인간 수준에 근접한 성능을 보이고 있다 [7], [8].

이에 따라 많은 기업체들은 기계 학습 기반의 번역기를 개발하고 있고, 시중에는 네이퍼 파파고 (papago.naver.com) 나 구글 번역기 (translate.google.com) 와 같은 번역서비스들이 많이 출시되었다. 이 번역기들은 전반적인 성능의 우수함을 자랑하나, 두 가지의 문제점이 있다. 첫 번째로 문장 내에 큰 숫자가 있을 경우, 그 숫자를 종종 다른 숫자로 오번역한다는 것이고, 두 번째로 같은 구조의 문장에서 숫자만 바꿔 번역할 때, 문장의 구조가 완전히 바뀌어 번역된다는 것이다.

첫 번째 문제점의 예시는 아래와 같다. 참고로 구글 및 파파고 번역은 2018년 6월1일 기준 결과이다.

입력 문장: "이 제품에는 약 3억 2백만 개의 다양한 미생물들이 있습니다."

구글 번역: "There are about 32 million different microorganisms in this product."

위 번역 결과는 구글 번역기를 사용한 것으로 '3억 2백만'을 '32 million' (3천 2백만) 으로 잘못 번역하는 문제를 보인다.

입력 문장: "나는 그에게 3,091만원의 빚을 지게 되었다."

파파고 번역: "I owe him 30.1 million won."

위 파파고 번역은 '3,091만원'을 '30.1 million' (3,010만원) 으로 오역하는 문제를 보인다.

두 번째 문제점의 예시는 아래와 같다.

입력 문장: "이 그룹은 모두 합쳐 30명이었다."

파파고 번역: "There were 30 people in total."

입력 문장: "이 그룹은 모두 합쳐 32명이었다."

파파고 번역: "In total, the group had 32 members."

위 번역 결과는 네이버 파파고를 사용한 것으로 입력 문장의 구조는 같지만 숫자에 따라 번역 결과의 문장 구조 자체가 달라지는 문제를 보인다.

본 논문의 목표는 상기 두 가지 문제인 숫자 오번역과 번역 후 문장 구조가 바뀌는 것들을 방지하여 숫자 번역을 보다 정확하게 구현하는 것이다.

본 논문에서 구현한 번역 모델은 bidirectional recurrent neural networks (RNN), long short-term memory (LSTM), 그리고 attention mechanism으로 구성된 neural machine translation (NMT) 모델이고, 사용한 데이터는 웹에서 크롤링한 데이터이다 (hancorpus.github.io).

데이터의 유효성을 검증하기 위해서 bi-gram과 uni-gram 기반의 BLEU score를 기준으로 데이터 클렌징을 수행한다. 또한 번역에 사용되는 사전의 크기를 조정함으로써 BLEU score가 가장 높은 모델을 선택해 모델을 최적화한다.

본 논문은 서론부터 시작하여 배경 지식, 제안 모델과 결과를 거쳐 결론에 이르며 연구에 대한 구체적인 방법들을 서술한다. 배경 지식부에서는 번역 모델에 사용된 다양한 방법들에 대해 서술하고, 제안 모델부에서는 데이터 클렌징과 사전 크기 조정을 통한 데이터 설정과 숫자 기호화 알고리즘에 대해 서술한다. 결과부에서는 제안된 모델에 대한 성능 검증을 위해 다양한 실험 결과들을 제시하고, 결론부에서는 요약이 제시된다.

Ⅱ. 배 경 지 식

인공신경망 기계 번역 시스템에 주로 사용되는 신경망은 RNN 과 convolutional neural network (CNN)이다. 본 논문에서는 번역 데이터를 다루기 때문에 순차적인 데이터의 학습에 적합한 구조인 RNN을 사용한다 [7].

본 논문에서 사용하는 시스템은 RNN을 양방향으로 적용하는 encoder 와 단방향으로 적용하는 decoder를 포함하고, 그 사이를 attention mechanism 이 연결한다. 이때, encoder 와 decoder에 사용하는 RNN 으로는 LSTM을 사용한다.

2-1 Bidirectional RNN

RNN은 데이터의 현재 값과 이전 값에 의해 결과 값이 결정되므로 번역 문장과 같은 순차적인 데이터에 효과적이다. 하지만 데이터의 이후 값을 알지 못한 채 이전 값과 현재 값만 이용되므로 번역의 정확성이 떨어진다는 문제점이 있다.

이러한 문제를 해결하기 위해 bidirectional RNN을 사용한다. Bidirectional RNN은 기존 RNN을 역방향으로 한 backward RNN을 추가해 이전 값과 이후 값을 모두 사용하여 결과 값을 결정하게 하므로 한 방향을 사용하는 경우보다 더 정확한 번역이 가능하다 [8].

2-2 LSTM

RNN은 어떤 정보와, 해당 정보가 필요한 곳의 거리가 짧을 때는 과거 정보를 이용하여 잘 학습할 수 있지만, 거리가 멀 때는 학습이 어려워지는 장기 의존성 문제가 발생한다. 즉, 번역 모델을 학습할 때 입력 문장의 길이가 길수록 학습이 어려워질 가능성이 높아진다.

이를 해결하기 위해 본 논문에서는 LSTM 방식의 RNN을 사용한다. LSTM은 cell, input gate, output gate, forget gate가 서로 상호작용하여 오랫동안 정보를 기억하게 하여 장기 의존성 문제를 해결하므로 번역 모델에 자주 사용된다 [7].

2-3 Attention Mechanism

Attention mechanism은 모델이 원문의 모든 단어에 집중하면 정확도가 떨어질 수 있다는 문제점을 해결하기 위해, 모든 단어가 아닌 중요 단어에만 집중하여 다음 단어를 예측하는 방법이다. 예를 들어, 한국어 "나는 행복하다"를 영어 "I am happy"로 번역하는 모델을 만들 때, "happy"라는 단어를 예측할 때 "행복"에 주목하게 만드는 방법이다.

2-4 Word Embedding

Word embedding은 하나의 단어를 n차원 벡터로 변환하여 나타내는 방법이다. 변환된 각 단어들은 각각의 벡터 값을 가진다. 아래 Figure 1은 기존의 neural machine translation 기반의 번역모델을 학습하여 단어를 500차원의 벡터로 변환한 뒤 principle component analysis (PCA)를 이용하여 2차원 벡터로 축소하여 나타낸 이미지이다. 각 단어들은 각각의 벡터 값을 지니는데, 각 숫자도 숫자 값에 따라 서로 다른 단어로 인식되어 각각의 벡터 값을 지닌다. 즉, 다른 숫자는 다른 의미를 가짐으로써 번역에서 숫자만 달라지더라도 문장 전체 구조가 다른 번역 결과를 얻게 되는 이유가 된다. 또한 이들 숫자들의 의미는 그 숫자를 포함하는 많지 않은 문장들에 기반하여 만들어지므로, 숫자의 변경은 문장 구조의 변경을 넘어서 오역으로 이어질 수 있다.

또한 자주 나타나지 않는 숫자들은 (예를 들면 3억2천만) 사전에 포함되지 않고 word embedding 과정을 거칠 수가 없게 된다. 이러한 경우를 위해 subword 모델 [9]을 활용하여 긴 숫자를 작은 숫자들의 연결로 표현하는데, 이때 숫자를 잘못 번역할 가능성이 높아진다.

Ⅲ. 제 안 모 델

3-1 데이터 설정

1) 데이터 클렌징

학습에 사용한 데이터가 웹에서 크롤링한 데이터이기 때문에 유효하지 않은 데이터를 삭제함으로써 데이터의 유효성을 확보할 필요가 있다. 우선 주어진 데이터에서 번역 모델을 학습 한 뒤, 학습에 사용된 데이터들을 다시 번역하여, 전혀 정답을 맞추지 못할 경우 삭제하는 방식으로 이루어진다. 즉, 주어진 원문과 번역문이 유효하다면, 번역문과 모델이 번역한 문장에서 최소 한 단어 혹은 두 단어가 연속으로 일치해야 유

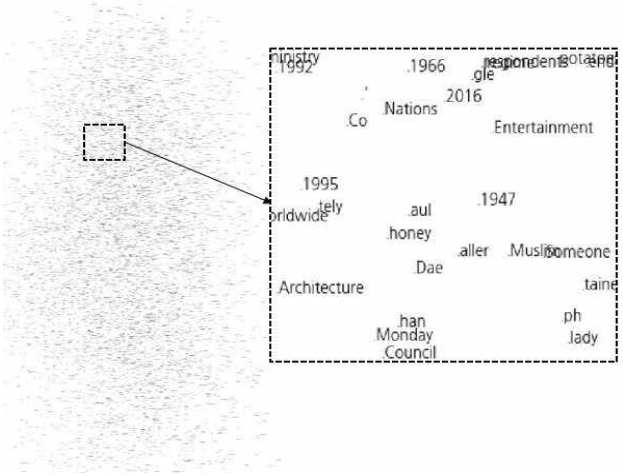

그림 1. 주요성분분석을 사용하여 2 차원에 표현된 word embedding. '1995' 나 '2016' 등의 숫자들도 'honey' 나 'lady' 등의 다른 단어들과 같은 공간에 표현된다. 즉, 숫자들도 같은 공간에서 semantic 의미를 갖게 된다.

Fig. 1. Word embedding representation in 2 dimensional space by principal component analysis. Numbers like '1995', '1947', and '2016' are presented in the same space as the other words like 'honey' and 'lady'. That is, the numbers have semantic meaning in the same space.

효한 문장이라는 생각으로 데이터 클렌징 방법을 제안했다. 해당 방법은 데이터 속에 유효하지 않은 문장이 어느 정도 존재하더라도 유효한 문장이 더 많을 것이고, 학습된 모델이 어느 정도 신뢰성이 있을 것이라는 전제에 기반한다.

또한, 데이터 클렌징에 사용하는 모델은 한영 번역 모델인데, 그 이유는 영한 번역은 한글의 언어적 특성상 점수가 높게 나오지 않기 때문에 차이를 알기 힘들기 때문이다.

데이터 클렌징 방법은 다음과 같다. 먼저 웹에서 크롤링한 데이터로 번역모델을 학습하고 그 모델로 데이터를 번역한다. 이때 번역 모델의 사전크기는 1만개로 고정한다. 그 후 번역문과 실제 번역문에서 uni-gram과 bi-gram을 이용하여 점수를 측정한다. 측정된 점수를 본 논문에서는 '클렌징 기준 점수'라고 칭한다. 클렌징 기준 점수를 바탕으로 특정 점수에 미치지 못하는 문장들을 삭제한다.

기준 점수 0, 5, 10, 20을 기준으로 데이터를 클렌징하고 그 데이터로 각 번역 모델을 학습하였을 때, 기준 점수 5로 클렌징한 데이터로 학습한 모델의 BLEU score가 가장 높았기에, 이 데이터를 사용하여 모델 최적화를 계속 진행한다.

2) 사전 크기 조정

번역 모델에 사용되는 사전은 일반적으로 사용하는 사전이 아닌 학습 데이터의 모든 단어를 최대한 포괄하는 임의의 문자열로 이루어진 사전이다.

추가적인 모델 최적화를 위하여 사전 크기를 다양하게 변

경하여 모델을 학습한다. 사전 크기를 각각 1만 개, 2만 개, 3만 개로 설정하여 모델을 학습한다. 여기서 학습하는 모델은 앞선 데이터 클렌징에서 기준 점수 5로 클렌징한 데이터를 사용하며, 한영 번역, 영한 번역 모델 둘 다 학습한다. 한영 번역 모델에서는 사전 크기가 1만 개일 때 BLEU score가 가장 높았고, 영한 번역 모델에서는 사전 크기가 3만 개일 때 BLEU score가 가장 높았다. 참고로, 모든 경우에 subword 모델인 BPE를 사용하여 사전을 만들었다 [9].

3-2 숫자 기호화

데이터 클렌징과 사전 크기를 다양하게 설정하여 최적화한 번역 모델을 찾아 학습할 수 있다. 이후 숫자 번역의 문제를 해결하기 위해 숫자 기호화 방식을 제안한다 [10].

기본적인 과정은 다음과 같다. 번역 모델이 번역할 때 입력으로 하나의 문장이 들어오면, 문장에 있는 숫자를 기호화하고, 기호화된 문장을 번역 모델을 이용하여 번역한다. 번역 후에 번역 문에 남아있는 기호를 입력문장에 있는 숫자를 이용해 다시 원래 숫자로 바꾼다.

숫자를 위와 같이 기호화한 다음 모델을 이용하여 번역하면 숫자 값에 따라 입력 벡터가 달라지지 않기 때문에 입력 문장의 숫자 값이 변해도 출력 문장이 달라지지 않는다. 이렇게 숫자 기호화를 적용시킴으로써 상기 서론부에서 제시했던 두 가지 문제를 해결할 수 있다.

숫자 기호화의 과정은 모델을 학습할 때와 새로운 문장을 번역할 때 두 가지 경우로 나누어진다. 첫번째로, 숫자 기호화가 적용된 모델을 학습할 때는 기호화가 되어있는 데이터가 필요하기 때문에 입력 문장과 번역 문장에서 숫자들을 기호화하는 전처리과정을 거친 후 번역모델을 학습한다. 두 번째로, 새로운 문장을 번역할 때는 입력 문장에 있는 모든 숫자를 기호화하여 사용한다. 두 가지 경우에 대한 자세한 기호화 과정은 아래와 같다.

1) 모델을 학습할 때

본 논문에서는 학습 데이터의 숫자를 최대한 기호화하기 위하여 여러 방법을 사용한다. 데이터에 있는 숫자를 기호화할 때 사용하는 기능은 다음과 같다.

(1) 숫자 표현 영단어를 숫자로 치환

영어 데이터에서 숫자를 나타내는데 자주 쓰이는 영어 단어들은 숫자로 치환한다. 예를 들어 'hundred', 'thousand', 'million' 과 같은 단어들은 숫자를 나타내는 경우가 대부분이기 때문에 이와 관련된 영어 단어들은 숫자로 치환한다.

(2) 숫자와 단위의 조합을 숫자로 치환

숫자를 표현할 때 숫자만을 사용하여 숫자를 표현하지 않고, 숫자와 단위의 조합으로 숫자를 표현하는 경우가 있다. 이러한 경우를 정규표현식을 이용하여 숫자로 치환한다. 예를 들어 '3천'과 '3 thousands'는 3000'으로 치환하고, '5만 7천'과 '57 thousands'는 '57000'으로 치환한다.

(3) 양쪽 문장에 같은 숫자가 있으면 해당 숫자를 기호화

이 기능은 숫자 기호화에서 가장 핵심적인 부분이다. 앞의 두 단계는 이 기능이 더 잘 수행되도록 돕는 부분이다. 원문과 번역문에 같은 숫자가 있으면 이 숫자를 기호화한다.

예를 들어 "나는 21살입니다."와 "I am 21 years old"라는 문장이 쌍을 이루면 그 안의 같은 숫자 '21'을 기호화하여 "나는 __N1살입니다."와 "I am __N1 years old."로 문장을 바꾼다.

(4) 1~20의 숫자 추가 기호화

1 ~ 20의 숫자는 해석이 다양하게 될 수 있기 때문에 따로 추가적인 기호화가 필요하다. 앞의 세 단계를 거친 후 한글 문장의 남은 숫자 중 1에서 20 사이의 숫자가 있고 영어 문장에 그에 해당하는 숫자가 문자로 표현되어 있으면 기호화한다.

예를 들어 "그녀는 17살에 죽었다."와 "She died at the age of seventeen."라는 문장이 쌍을 이루면 'seventeen'과 '17'이 같은 숫자이기 때문에 기호화하여 "그녀는 __N1살에 죽었다."와 "She died at the age of __N1."로 바꾼다.

2) 번역할 때

학습이 끝난 후 새로운 문장을 번역 할 때는 학습 데이터를 다룰 때와는 다른 알고리즘을 적용한다. 학습할 때와 달리 입력 문장만 주어지기 때문에 입력 문장의 숫자를 기호화하고, 기호화한 문장을 모델을 이용하여 번역하여 결과로 나온 번역문의 기호를 다시 원래 숫자로 치환한다.

예를 들어 "There are 3 chairs."라는 문장이 입력으로 들어오면 숫자를 기호화하여 입력을 "There are __N1 chairs."로 바꾸고, 그 후 번역 모델로 번역하여 "__N1개의 의자가 있다." 라는 번역 결과를 얻고, 기호를 다시 원래 숫자로 바꾸어 "3개의 의자가 있다."라는 최종 번역문을 얻는다.

Ⅳ. 실 험 결 과

4-1 데이터

본 논문에서 사용한 데이터의 양은 가장 초기 모델을 학습할 때 학습 데이터 3,218,337 문장, valid 데이터 5,000 문장, test 데이터 3,409 문장이다. 학습된 모델로 학습 데이터를 번역한 후 BLEU score 5를 기준으로 데이터 클렌징을 거치고 난 후에는 학습 데이터 2,201,423 문장, valid 데이터 2,590 문장, test 데이터 1,857 문장이 남아, 이들 문장으로 모든 모델들에 대한 평가를 진행하였다.

다음의 문장들은 데이터 클렌징에서 삭제된 문장들 중 일부이다.

영어 문장 1 : "And I want to talk to my readers about them."

한글 문장 1 : "당신은 심지어 뉴스 앵커들로부터도 이 표현을 들을 수 있다."

영어 문장 2 : "They liked the same books music films."

한글 문장 2 : "내가 너를 정말로 좋아했다는 걸 너가 알기를 바란다."

영어 문장 3 : "The fairy tale written."

한글 문장 3 : "생쥐가 말로, 호박이 마차로 바뀌는 샤를 페로의 동화는 너무나 유명하다."

데이터 클렌징을 통하여 위 문장 1, 문장 2와 같이 의미가 서로 다른 문장이나 위 문장 3과 같이 중간에 끊긴 문장들이 제거되었다.

4-2 번역 결과

본 논문에서 번역 모델의 성능 향상을 위해 적용한 방법은 데이터 클렌징, 사전 크기 설정, 숫자 기호화이다. Table 1은 데이터 클렌징과 사전 크기 설정에 따른 valid 데이터의 BLEU score 결과이다. 한영 번역 모델을 사용하여 얻은 클렌징 기준 점수 5점을 이용하였을 때 한영 번역 모델의 valid 데이터의 BLEU score는 0.35점 상승하였다. 영한 번역 모델도 한영 번역 모델을 사용하여 클렌징한 데이터로 학습하였을 때 valid 데이터의 BLEU score 7.94점이 나왔다.

표 1. 데이터 클렌징과 사전크기 조정을 통한 한영 및 영한 번역의 성능 향상. 모델들은 BLEU 점수로 평가됨.

Table 1. Translation performance for Kr2En and En2Kr improved by data cleansing and a proper vocabulary size. The models are evaluated in BLEU scores.

	Kr2En	En2Kr
Original	21.27	-
+Data Cleansing	**21.62**	7.94
+Dictionary Size Setting	21.62	**9.05**

사전 크기를 다양하게 실험하였을 때 한영 번역 모델의 최적 사전 크기는 동일하게 1 만이었고, 영한 번역 모델의 최적 사전 크기는 3 만이었다. 한영 번역 모델은 valid 데이터의 BLEU score는 동일했고, 영한 번역 모델은 valid 데이터의 BLEU score는 1.11점 상승하였다.

Table 2는 데이터 클렌징과 사전 크기 설정으로 최적화된 모델과 그에 추가적으로 숫자 기호화를 적용한 모델의 valid 데이터와 test 데이터의 BLEU score 결과이다. valid 데이터의 BLEU score는 한영 번역에서는 숫자 기호화를 적용하지 않은 모델이 0.34점 더 높고 영한 번역에서는 숫자 기호화를 적용한 모델이 1.10점 더 높다. test 데이터는 숫자 기호화의 효과를 알기 위하여 숫자가 포함된 문장 위주로 구성하였다. test 데이터의 BLEU score는 숫자 기호화를 적용한 모델이 한영 번역에서 1.86점, 영한 번역에서 4.51점 더 높아 숫자 기호화의 성능을 확인할 수 있다.

표 2. 한영과 영한 번역에서 숫자 기호화의 효과. Table 1 에서와 같은 사전크기와 데이터 클렌징을 사용함.

Table 2. Effects of number symbolization in Kr2En and En2kr translation. The models are the same as in Table 1. with the same vocabulary size after data cleansing.

	Kr2En		En2Kr	
	Valid	Test	Valid	Test
Optimized Model	21.62	22.21	9.05	12.39
+Number Symbolization	21.28	**24.07**	10.15	**16.90**

숫자 기호화를 적용할 경우에 BLEU score 가 향상되는 것을 통해 성능 개선을 확인할 수 있다. 추가적으로, BLEU score에는 보이지 않는 개선을 확인하기 위해 아래의 실제 번역 예시들을 통해 질적인 차이를 확인할 수 있다.

시중의 번역기에서 문제를 보이는 번역 예시들에 대해 본 논문에서 제안한 숫자 기호화를 적용하면 다음과 같이 정확히 번역하는 것을 볼 수 있다.

서론부에서 구글 번역기는 '3억 2백만'을 '32 million'(3200만)으로 오번역하나, 본 논문의 번역기는 '3020000000'라고 올바르게 번역한다.

입력 문장 : "이 제품에는 약 3억 2백만 개의 다양한 미생물들이 있습니다."

번역 결과 : "There are about 3020000000 different microbes in the product."

서론부에서 네이버 파파고는 '3,091만'을 '30.1 million'(3010만)으로 오번역하나, 본 논문의 번역기는 '30910000'으로 올바르게 번역한다.

입력 문장 : "나는 그에게 3,091만원의 빚을 지게 되었다."

번역 결과 : "I owe him 30910000 won."

서론부에서 지적한 두 번째 문제 예시에 대해서 우리가 제안한 번역모델의 결과는 아래와 같다.

입력 문장 : "이 그룹은 모두 합쳐 30명이었다."

번역 결과 : "The group consisted of 30 people."

입력 문장 : "이 그룹은 모두 합쳐 32명이었다."

번역 결과 : "The group consisted of 32 people."

서론부에서 네이버 파파고는 입력문장에서 '30'을 '32'로만 바꾸어도 문장 구조를 완전히 바꾸어서 번역하지만 본 논문의 번역기는 문장 구조를 그대로 유지하면서 숫자만 바뀐 번역결과를 생성한다.

Ⅴ. 결 론

본 논문에서는 기존 번역기들의 숫자 오번역 문제를 지적하고 이를 해결하는 숫자 기호화 알고리즘을 제안하였다. 그리고 번역 모델에 숫자 기호화 알고리즘을 적용하기에 앞서, 데이터 클렌징과 사전 크기 조정을 통해 정제된 데이터와 최적화된 모델을 생성 한다.

데이터를 클렌징할 때 기준 점수 5점을 사용하였을 때, BLEU score가 21.62점으로 가장 높았고, 사전 크기를 한영은 1만 개, 영한은 3만 개로 조정하였을 때 BLEU score가 각각 21.62점, 9.05점으로 가장 높았다.

데이터 클렌징과 사전 크기 설정으로 최적화한 모델과 해당 모델에 숫자 기호화를 적용한 모델을 비교하였을 때 숫자가 포함되지 않은 문장의 BLEU score는 비슷하였지만, 숫자가 들어간 문장의 BLEU score의 경우는 숫자 기호화를 적용한 모델의 BLEU score가 더 높았다. 또한 숫자 기호화를 통해 숫자의 오번역하는 문제와 숫자만 바뀌는 경우에도 번역 문장 구조가 바뀌는 문제를 해결함으로써 사람이 기대하는 번역에 좀 더 유사한 번역 결과를 확보할 수 있었다.

앞으로 다른 번역 모델에도 이 논문의 제안된 방법을 적용할 수 있고, 언어모델 등 다른 자연어 처리에도 확장 가능하다.

감사의 글

이 논문은 2018년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2017R1D1A1B0303331).

참고문헌

[1] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer Society*, Vol. 42, No. 8, pp.30-37, August 2009.

[2] H. Choi, Y. Kang, and M. Kang, "Pet shop recommendation system based on implicit feedback," *Journal of Digital Contents Society*, Vol. 18, No. 1, pp. 1-4, 2017.

[3] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, Vol. 521, No. 7553, pp. 436–444, 2015.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Neural Information Processing Systems, pp. 1097–1105, 2012.

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbur, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, pp.82-97, 2012.

[6] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural Image Caption Generation with Visual Attention," in *Proceeding of the International Conference on Machine Learning*, Lille: France, pp. 2048-2057, July 2015.

[7] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks", NIPS 2014.

[8] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015.

[9] R. Sennrich, B. Haddow, and A. Birch. " Neural machine translation ofrare words with subword units," arXivpreprint arXiv:1508.07909, 2015.

[10] H. Choi, K. Cho, and Y. Bengio, "Context-dependent word representation for neural machine translation", *Computer Speech & Language*, Vol. 45, p. 149-160, 2017.

강청웅(Cheongwoong Kang)

2015년～현 재: 한동대학교 전산전자공학부 재학

※ 관심분야 : 머신러닝, 딥러닝, 인공지능

노영헌(Youngheon Ro)

2012년～현 재: 한동대학교 전산전자공학부 재학

※ 관심분야 : 머신러닝, 딥러닝, 인공지능

김지수(Jisu Kim)

2013년～현 재: 한동대학교 전산전자공학부 재학

※ 관심분야 : 딥러닝, 신경 기계 번역

최희열 (Heeyoul Choi)

2005년: 포항공과대학교, 컴퓨터공학과 (이학석사)
2010년: Dept. of Computer Science and Engineering, Texas A&M University (Ph.D)
2010년 ~ 2011년: Indiana University (PostDoc)
2015년 ~ 2016년: University of Montreal (Visiting Researcher)

1998년 ~ 2001년: OromInfo (Programmer)
2011년～2016년: 삼성전자 종합기술원 (Research Staff Member)
2016년～현 재 : 한동대학교 전산전자공학부 조교수
※ 관심분야 : 머신러닝, 딥러닝, 인공지능