문자 단위의 Neural Machine Translation

이창기○, 김준석, 이형규, 이재송
강원대학교○, 네이버 랩스
leeck@kangwon.ac.kr, {jun.seok, hg.lee, jaesong.lee}@navercorp.com

Character-Level Neural Machine Translation

Changki Lee○, Junseok Kim, Hyoung-Gyu Lee, Jaesong Lee
Kangwon National University○, NAVER LABS

요 약

Neural Machine Translation (NMT) 모델은 단일 신경망 구조만을 사용하는 End-to-end 방식의 기계번역 모델로, 기존의 Statistical Machine Translation (SMT) 모델에 비해서 높은 성능을 보이고, Feature Engineering이 필요 없으며, 번역 모델 및 언어 모델의 역할을 단일 신경망에서 수행하여 디코더의 구조가 간단하다는 장점이 있다. 그러나 NMT 모델은 출력 언어 사전(Target Vocabulary)의 크기에 비례해서 학습 및 디코딩의 속도가 느려지기 때문에 출력 언어 사전의 크기에 제한을 갖는다는 단점이 있다. 본 논문에서는 NMT 모델의 출력 언어 사전의 크기 제한 문제를 해결하기 위해서, 입력 언어는 단어 단위로 읽고(Encoding) 출력 언어를 문자(Character) 단위로 생성(Decoding)하는 방법을 제안한다. 출력 언어를 문자 단위로 생성하게 되면 NMT 모델의 출력 언어 사전에 모든 문자를 포함할 수 있게 되어 출력 언어의 Out-of-vocabulary(OOV) 문제가 사라지고 출력 언어의 사전 크기가 줄어들어 학습 및 디코딩 속도가 빨라지게 된다. 실험 결과, 본 논문에서 제안한 방법이 영어-일본어 및 한국어-일본어 기계번역에서 기존의 단어 단위의 NMT 모델보다 우수한 성능을 보였다.

주제어: Neural Machine Translation, 기계번역, Statistical Machine Translation, Deep Learning

1. 서론

기계번역에 신경망을 적용하는 방식은 주로 번역모델이나 언어모델 등의 일부분에 신경망을 적용하는 방식이 주로 연구되었고, 최근에 End-to-end 방식의 신경망 구조만을 사용하는 Neural Machine Translation (NMT) 모델이 개발되어 영어-프랑스와 같이 어순이 유사한 언어쌍에서 좋은 성능을 보였다[1][2]. NMT 모델은 입력 언어 문장을 단어 단위로 읽어(Encoding) 출력 언어 문장을 단어 단위로 생성(Decoding)하는 단일 신경망으로 구성되어 있으며, 병렬 코퍼스를 학습데이터로 사용하여 입력 언어 문장이 주어졌을 때 올바른 출력 언어 문장을 생성할 확률이 최대가 되도록 학습된다.

NMT 모델은 전통적인 방식의 Statistical Machine Translation (SMT) 모델에 비해서 다음과 같은 장점을 가진다. 첫 번째로, NMT 모델은 최소한의 전문 지식(Domain Knowledge)만이 필요하다. 전통적인 방식의 SMT는 많은 Feature Engineering이 필요한데, 이러한 Feature Engineering에는 전문적인 지식이 필요하고 시간도 많이 소요된다. 그러나 NMT는 이런 작업이 필요 없이 신경망의 구조만 결정해 주면 학습되는 파라미터들에 번역에 필요한 모든 정보들이 포함되게 된다. 두 번째로, NMT 모델은 입력 언어 문장이 주어졌을 때, 올바른 출력 언어 문장이 생성되도록 단일 신경망을 직접 학습한다. 전통적인 방식의 SMT는 단어 정렬(Word Alignment)을 최적화시키기 위한 기계학습, 언어 모델을 최적화시키기 위한 기계학습, 디코더에서 각 Feature들의 가중치(Weight)를 최적화시키기 위한 기계학습을 각각 수행하는 문제점이 있다. 세 번째로, NMT 모델의 디코더는 구조가 간단하다. 전통적인 방식의 SMT는 번역 모델, 언어 모델 등의 리소스가 필요하고 번역 방식에 따라서 다양한 형태의 디코더가 필요하고, 경우에 따라서는 구문분석기가 필요해지고 어순의 변경도 필요하다.

그러나 NMT 모델은 출력 언어의 사전(Target Vocabulary)의 크기가 커질수록 학습 및 디코딩의 속도가 느려지기 때문에 출력 언어 사전의 크기에 제한을 갖는다는 단점이 있다. 예를 들어, [1]에서는 입력 언어와 출력 언어의 사전을 빈도수가 높은 15,000단어로 구성하였고, [2]에서는 빈도수가 높은 30,000 단어를 사전으로 사용하였으며, 두 연구 모두 사전에 포함되지 않는 단어는 UNK 기호로 대체시켰다. NMT 모델의 사전 크기 제한 문제를 해결하기 위해서 최근에 많은 연구가 진행되었다. [3]에서는 병렬 코퍼스에 단어 정렬(Word Alignment)을 적용하여 입력 언어와 출력 언어의 Out-of-vocabulary(OOV) 단어들 간의 매핑 정보를 구축하여 OOV 사전을 구축하고, NMT의 출력 언어 문장의 UNK 기호에 대응되는 입력 언어의 단어를 (단어 정렬로부터 구함) OOV 사전에서 검색하여 UNK 기호를 출력 언어 단어로 바꾸어 주는 후처리(Post-processing) 기술을 제안하였다. [4]에서는 사전의 크기가 커지더라도 학습 속도가 떨어지지 않기 위해 Softmax 계산을 근사적으로 수행하는 Importance Sampling 기반의 방법을 제안하였으며, 영어-프랑스어 및 영어-독일어에 50만 단어의 사전을 이용하여 최고 수준의 성능을 보였다.

본 논문에서는 NMT 모델의 출력 언어 사전의 크기 제한 문제를 해결하기 위해서 입력 언어는 단어 단위로 읽

고 출력 언어를 문자(Character) 단위로 생성하는 방법을 제안한다. 출력 언어를 문자 단위로 생성하게 되면 NMT 모델의 출력 언어 사전에 모든 문자를 포함할 수 있게 되어 출력 언어의 OOV 문제가 사라지고 출력 언어의 사전 크기가 줄어들어 학습 및 디코딩 속도가 빨라지게 된다는 장점을 얻게 된다. 본 논문에서 제안한 방법은 기존의 NMT 모델을 변경할 필요 없이 그대로 사용할 수 있으며, 추가적인 학습 혹은 후처리(Post-processing) 등이 필요 없다.

본 논문의 구성은 다음과 같다. 2장에서는 NMT 모델에 대해서 설명하고, 3장에서는 본 논문에서 제안하는 문자 단위 NMT 모델에 대해서 설명하고, 4장에서는 한국어-일본어 기계번역과 어순이 상이한 영어-일본어 기계번역에 NMT 모델을 적용한 결과를 설명한다.

2. Neural Machine Translation

NMT는 Recurrent Neural Network(RNN)등의 신경망을 이용하여 P(**y**|**x**)를 직접 최적화하는 모델로(**x**는 입력 언어 문장, **y**는 출력 언어 문장), 그림1은 NMT 모델 중에 하나인 RNN Encoder-decoder 모델을 나타낸다[1]. 첫 번째 RNN(Encoder)은 입력 언어 문장을 다음과 같이 실수의 벡터 표현(Continuous-space Representation) c로 인코딩(encoding)한다:

$$c = f_{enc}(x)$$

두 번째 RNN(Decoder)은 이로부터 P(**y**|**x**)를 최대화하는 출력 언어 문장을 생성한다. RNN에서는 Long Term Dependency를 학습하기 위해서 Long Short-Term Memory (LSTM)나 Gated Recurrent Unit(GRU)를 사용하며, 전체 시스템은 한번에(End-to-end) 학습된다. 학습이 끝난 후, 실제 번역을 수행할 때는 주어진 입력 언어 문장으로부터 Beam Search 등을 이용하여 P(**y**|**x**)이 가장 높은 출력 언어 문장을 찾는다.

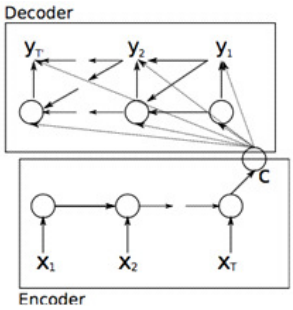

그림1. RNN Encoder-decoder model [1]

RNN Encoder-decoder 모델은 입력 언어의 문장을 길이에 상관없이 항상 고정된 차원의 단일 벡터로 인코딩하는데, 이로 인해 입력 언어 문장이 길어질 경우 번역의 성능이 떨어진다는 문제가 있다. 또한 입력 언어 문장으로부터 고정된 길이의 벡터만을 생성하고, 이로부터 출력 언어 문장을 생성하기 때문에 번역이 잘못되었을 경우에 원인을 분석하기 어렵다는 문제가 있다.

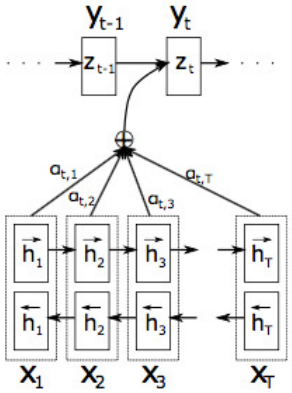

그림2. RNN Search model [2]

RNN Search 모델에서는 인코더와 디코더 사이에 Attention mechanism을 두어 이러한 문제들을 해결하였다[2]. 그림2는 RNN search 모델을 나타낸다. 인코더에서는 Bidirectional RNN을 사용하여 Forward Network에서는 Hidden State Vector Set $\{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_T\}$를 생성하고 Backward Network에서는 Hidden State Vector Set $\{\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_T\}$를 생성하여, 각각의 단어 별로 두 벡터들을 합하여 Context Vector Set $\{c_1, c_2, \ldots, c_T\}$ $(c_t = [\vec{h}_t; \overleftarrow{h}_t])$를 생성한다. Attention Mechanism은 인코더가 생성한 각각의 Context vector c_t와 디코더가 현재까지 생성한 출력 언어 문장($y_1, y_2, \ldots, y_{t-1}$)의 정보를 포함하고 있는 Hidden State Vector z_{t-1}을 입력으로 받아서 다음 출력 언어 단어 y_t를 예측하기 위해서 주의해서 봐야 할 Context Vector c_t의 Attention Weight를 결정한다. 이러한 Attention Weight를 결정하기 위해서 Feed-Forward Neural Network(FFNN)와 같은 신경망(f_{ATT})이 내부적으로 사용되고, Attention Weight를 이용하여 Context Vector Set의 가중치 합(Weighted Sum)을 구하여 새로운 Context Vector c^t를 아래와 같이 구한다:

$$e_i^t = f_{ATT}(z_{t-1}, c_i),$$

$$\alpha_i^t = \frac{\exp(e_i^t)}{\sum_{j=1}^{T} \exp(e_j^t)},$$

$$c^t = \sum_{i=1}^{T} \alpha_i^t c_i.$$

디코더는 새로 구한 Context Vector c^t와 디코더의 이전 Hidden State Vector z_{t-1}와 이전 출력 단어 y_{t-1}을 입력으로 받아서 Hidden State Vector z_t를 갱신하고 이를 이용하여 새로운 출력 단어 y_t를 Beam Search 등을 이용하여 결정한다. RNN Search 모델은 Attention Mechanism을 도입하여 출력 언어의 각 단어별로 Context Vector c^t를 새로 계산하기 때문에 RNN Encoder-decoder 모델에 비해서 긴 입력 언어 문장이 들어오더라도 성능 하락이 적으며, Attention Weight를 단어 정렬(Word alignment)로 사용할 수 있어 잘못된 번역의 원인 분석

이 쉽다.

3. 문자 단위의 Neural Machine Translation

본 논문에서는 NMT 모델의 출력 언어 사전의 크기 제한 문제를 해결하기 위해서 입력 언어는 단어 단위로 읽고(Encoding) 출력 언어를 문자(Character) 단위로 생성하는 문자 단위 NMT 모델을 제안한다. 문자 단위 NMT 모델은 출력 언어 사전의 크기가 줄어들어 모든 문자를 사전에 포함할 수 있게 되어 출력 언어의 OOV 문제가 사라지고 학습 및 디코딩 속도도 빨라지게 되며, 기존의 NMT 모델을 변경할 필요 없이 그대로 사용할 수 있으며, 추가적인 학습 혹은 후처리(Post-Processing) 등이 필요없이 학습데이터(병렬코퍼스)의 출력 언어 부분만을 문자 단위로 바꾸어 주는 전처리(Pre-Processing) 작업만이 필요하다. 출력 언어를 문자 단위로 변경할 때는 단순 문자 단위로 변경하는 것 보다 문자에 단어 분리(Word Segmentation) 정보를 추가한 '문자+Begin/Inside' 형태로 변경하는 것이 더 좋은 성능을 보여, 본 논문에서는 '문자+Begin/Inside' 형태를 사용하였다.

입력 언어의 경우는 사전의 크기가 커지더라도 학습 및 디코딩의 속도에는 큰 영향을 주지 않으며, 입력 언어를 문자 단위로 인코딩할 경우에는 실험 결과 큰 성능 하락을 보여서, 본 논문에서는 입력 언어는 단어 단위로 인코딩하며 충분히 큰 크기의 입력 언어 사전을 이용하였다(한국어 60,527 단어, 영어 245,111 단어).

다음은 영어-일본어 기계번역에 사용된 병렬코퍼스의 한 문장에 대해서 단어 단위 및 문자 단위로 인코딩한 문장의 예이다.

영어: The/DT details/NNS of/IN the/DT result/NN were/VBD described/VBN ./.
일본어: その/UN 結果/NCA を/PS 詳細/NCD に/VX 記し/VC た/VX 。/OP
일본어 '문자+B/I' 형태 변환: そ/B の/I 結/B 果/I を/B 詳/B 細/I に/B 記/B し/I た/B 。/B

4. 실험

본 논문에서는 기존의 SMT와 NMT 및 본 논문에서 제안한 문자 단위의 NMT의 성능을 비교 평가하기 위해서, ASPEC(Asian Scientific Paper Excerpt Corpus) 영어-일본어 병렬 코퍼스와 JPO(Japan Patent Office) Patent 한국어-일본어 병렬코퍼스를 이용하여 영-일 및 한-일 기계번역 시스템을 학습 및 평가하였다[5]. ASPEC 코퍼스는 과학 기술 분야의 논문에서 수집된 300만 문장으로 구성되어 있고, JPO Patent 코퍼스는 100만 문장으로 구성되어 있다.

본 논문에서는 SMT 시스템과 NMT 시스템을 각각 구현하였으며, 학습 데이터는 동일하게 번역 품질 상위 100만 문장만을 이용하였다. SMT 시스템은 오픈소스 엔진인 Moses[6]을 이용하여 구현되었으며, 구문 기반(Syntax-based) 모델 중 하나인 Tree-to-string 모델[7]을 학습하였고, MERT 알고리즘[8]을 이용하여 파라미터 튜닝을 수행하였으며, Chart 파싱 디코딩[9]을 통해 번역문을 생성하였다. Tree-to-string 모델에서는 소스 언어의 구문 분석 정보를 필요로 하기 때문에 영어 구문 분석을 위해서 Berkeley 파서[10]를 이용하였다.

NMT 시스템은 RNN search 모델[2]과 유사하게 Theano[11]를 이용하여 자체적으로 구현하였으며, 디코더 부분에서 학습 속도를 위해 Maxout network 대신 ReLU를 사용하였다. 학습은 Stochastic Gradient Decent(SGD)를 사용하였으며, 입력/출력 언어 모두 200차원의 Word Embedding을 Projection Layer에 사용했고, Hidden Layer Unit수는 1000을 사용했으며, Dropout은 사용하지 않았다.

번역 결과의 성능 평가를 위해 일본어 형태소 분석기로 JUMAN을 사용하여 테스트 데이터에서의 BLEU[12]와 RIBES[13]를 측정하였다. BLEU는 번역 평가에서 가장 널리 사용되고 있는 척도이며, RIBES는 BLEU에 비해 영어-일본어와 같이 어순 차이가 큰 언어 쌍에서 더욱 정확한 평가가 가능하다고 알려진 척도이다.

표 1. 영어-일본어 기계번역 성능

모델	BLEU	RIBES
PB SMT [5]	27.48	0.6837
HPB SMT [5]	30.19	0.7347
Tree-to-string SMT	32.63	0.7833
Word-level NMT	29.78	0.7877
Character-level NMT	**33.14**	**0.8073**
Tree-to-string SMT + NMT Re-ranking	**34.60**	0.8000

표1은 영-일 기계번역에서의 구 기반 SMT(PB SMT)[14], 계층적 구 기반 SMT (HPB SMT)[9], 구문 기반 SMT(Tree-to-string SMT)[7], NMT, 문자 단위 NMT의 비교 평가 결과를 보여준다. SMT의 기본 모델인 구 기반 모델과 계층적 구 기반 모델의 결과는 동일한 코퍼스로 학습하고 평가되어 WAT 2014[5]에서 보고된 결과를 참조하였다. 구문 기반 모델은 구 기반 모델이나 계층적 구 기반 모델에 비해 확연히 좋은 성능을 보여주었다. 실험 언어 쌍이 어순 차이가 큰 영어-일본어이기 때문에 입력 언어 문장의 구문 분석 정보가 활용되는 구문 기반 모델이 더 좋은 번역문을 만들어 내었다고 분석된다. NMT는 명시적인 구문 분석을 수행하지 않음에도 불구하고 SMT의 구 기반 모델과 계층적 구 기반 모델을 능가하였다. 특히 문자 단위 NMT는 기존의 단어 단위 NMT에 비해서 BLEU 3.36점이 높았으며 구문 기반 모델보다 0.51점이 높아 최고의 성능을 보였으며, RIBES에서도 최고의 성능을 보였다. 또한 구문 기반 모델의 결과를 문자 단위 NMT로 Re-ranking 한 경우 BLEU 1.46점이 추가로 상승하였다. 다만 RIBES 척도에서는 여전히 문자 단위 NMT가 가장 높은 점수를 보였다. 이러한 결과를 통해 NMT의 RNN과 Attention Mechanism이 구문 분석 없이도 문장 내의 원거리 의존성을 잘 학습하고, 문자 단위의 NMT의 경

우 OOV 문제를 해결하여 단어 단위 NMT 보다 좋은 성능을 보임을 알 수 있다.

표 2. 한국어-일본어 기계번역 성능

모델	BLEU	RIBES
PB SMT [15]	69.22	0.9413
HPB SMT [15]	67.41	0.9372
Word-level NMT	61.52	-
Character-level NMT	65.72	0.9346
PB SMT + NMT re-ranking	**71.38**	**0.9438**

표2는 한-일 특허 기계번역에서의 구 기반 SMT(PB SMT), 계층적 구 기반 SMT (HPB SMT), 문자 단위 NMT의 비교 평가 결과를 보여준다. 한-일 특허 번역의 경우 도메인이 제한적이고 한국어와 일본어의 어순이 유사하여 SMT의 기본 모델인 구 기반 모델이 계층적 구 기반 모델이나 NMT 보다 우수한 성능을 보였으나, 여전히 문자 단위 NMT가 단어 단위 NMT 보다 높은 성능을 보였고, 구 기반 모델의 결과에 문자 단위 NMT로 Re-ranking 한 경우 BLEU 2.16점이 상승하였다.

5. 결론

본 논문에서는 NMT 모델의 출력 언어 사전의 크기 제한 문제를 해결하기 위해서, 입력 언어는 단어 단위로 읽고(Encoding) 출력 언어를 문자(Character) 단위로 생성(Decoding)하는 문자 단위 NMT 모델을 제안하였다. 실험 결과, 문자 단위 NMT 모델이 영어-일본어 및 한국어-일본어 기계번역에서 기존의 단어 단위의 NMT 모델보다 우수한 성능을 보였다.

향후 연구로는 NMT 모델의 성능을 개선하고, 한국어나 일본어, 중국어와 같은 언어에 알맞은 NMT 모델을 개발할 계획이다.

참고문헌

[1] Cho, K. et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” Proceedings of EMNLP ’14, 2014.

[2] Bahdanau, D. et al., “Neural machine translation by jointly learning to align and translate,” Proceedings of ICLR’15, arXiv:1409.0473, 2015.

[3] Luong, M. et al., “Addressing the Rare Word Problem in Neural Machine Translation,” Proceedings of ACL’15, 2015.

[4] Jean, S. et al., “On Using Very Large Target Vocabulary for Neural Machine Translation,” Proceedings of ACL’15, 2015.

[5] Nakazawa, T. et al., “Overview of the 1st workshop on Asian translation,” Proceedings of WAT’14, 2014.

[6] Koehn, P., et al., “Moses: Open source toolkit for statistical machine translation,” Proceedings of ACL ’07, 2007.

[7] Liu, Y., et al., “Tree-to-string alignment template for statistical machine translation,” Proceedings of Coling-ACL ’06, 2006.

[8] Och, F. J., "Minimum error rate training in statistical machine translation." Proceedings of ACL ’03, 2003.

[9] Chiang, D., "A hierarchical phrase-based model for statistical machine translation," Proceedings of ACL ’05, 2005.

[10] Petrov, S. et al., "Learning Accurate, Compact, and Interpretable Tree Annotation," Proceedings of Coling-ACL ’06, 2006.

[11] Bastien, F. et al. “Theano: new features and speed improvements,” Deep Learning and Unsupervised Feature Learning NIPS 2012 Workshop. 2012.

[12] Papineni, K, et al., "BLEU: a method for automatic evaluation of machine translation," Proceedings of ACL ’02, 2002.

[13] Isozaki, H. et al., “Automatic Evaluation of Translation Quality for Distant Language Pairs,” Proceedings of EMNLP ’10, 2010.

[14] Koehn, P. et al., "Statistical phrase-based translation," Proceedings of NAACL-HLT ’03, 2003.

[15] Nakazawa, T. et al., “Overview of the 2nd workshop on Asian translation,” Proceedings of WAT’15, 2015.