신경망 기계번역에서 최적화된 데이터 증강기법 고찰

박찬준○, 김규경, 임희석

bcj1210@naver.com, overmind22@korea.ac.kr, limhseok@korea.ac.kr

고려대학교 컴퓨터학과

Optimization of Data Augmentation Techniques in Neural Machine Translation

Chanjun Park○ ,Kim Kuekyeng, Heuiseok Lim

Korea University Dept.Computer Science

요 약

딥러닝을 이용한 Sequence to Sequence 모델의 등장과 Multi head Attention을 이용한 Transformer의 등장으로 기계번역에 많은 발전이 있었다. Transformer와 같은 성능이 좋은 모델들은 대량의 병렬 코퍼스를 가지고 학습을 진행하였는데 대량의 병렬 코퍼스를 구축하는 것은 시간과 비용이 많이 드는 작업이다. 이러한 단점을 극복하기 위하여 합성 코퍼스를 만드는 기법들이 연구되고 있으며 대표적으로 Back Translation 기법이 존재한다. Back Translation을 이용할 시 단일 언어 데이터를 가상 병렬 데이터로 변환하여 학습데이터의 양을 증가 시킨다. 즉 말뭉치 확장기법의 일종이다. 본 논문은 Back Translation 뿐만 아니라 Copied Translation 방식을 통한 다양한 실험을 통하여 데이터 증강기법이 기계번역 성능에 미치는 영향에 대해서 살펴본다. 실험결과 Back Translation과 Copied Translation과 같은 데이터 증강기법이 기계번역 성능향상에 도움을 줌을 확인 할 수 있었으며 Batch를 구성할 때 상대적 가중치를 두는 것이 성능향상에 도움이 됨을 알 수 있었다.

1. 서 론

딥러닝을 이용한 Sequence to Sequence 모델의 등장[1,2]과 Multi head Attention을 이용한 Transformer[3]의 등장으로 기계번역에 많은 발전이 있었다. Transformer와 같은 성능이 좋은 모델들은 대량의 병렬 코퍼스를 가지고 학습을 진행하였는데 대량의 병렬 코퍼스를 구축하는 것은 시간과 비용이 많이 드는 작업이다. 이러한 단점을 극복하기 위하여 합성 코퍼스를 만드는 기법들이 연구되고 있으며 대표적으로 Back Translation 기법이 존재한다.

Back Translation이란 기존의 훈련된 반대 방향 번역기를 사용해 단일 언어 코퍼스에 대한 번역을 진행하며 합성 synthetic 병렬 코퍼스를 만든 후, 이것을 기존 양방향 병렬 코퍼스에 추가하여 훈련에 사용하는 방식이다.[4,5] 즉 Back Translation을 이용할 시 단일 언어 데이터를 가상 병렬 데이터로 변환하여 학습데이터의 양을 증가 시킨다. Back Translation 같은 경우 NMT의 장점인 번역기를 만들 때 하나의 병렬 코퍼스로 두 개의 번역 모델을 만들 수 있다는 특성에 기반한다. Copied Translation이란 반대 방향 번역기의 활용 없이 단일 언어 코퍼스 활용하는 방법론이다. 즉 소스 쪽과 타깃 쪽에 똑같은 데이터를 넣어 훈련시키는 방법이다.[6] 하지만 소스 언어의 어휘 vocabulary에 타깃 언어의 어휘가 포함되는 불필요함을 감수해야 한다는 단점이 존재한다. 본 논문은 Back Translation 뿐만 아니라 Copied Translation 방식을 통한 다양한 실험을 통하여 데이터 증강기법이 기계번역 성능에 미치는 영향에 대해서 살펴본다.

2. Transformer 기반 기계번역

Transformer[3]란 Convolution과 Recurrence 없이 오직 Attention만을 이용한 기계번역 모델로 구글에서 2017년 제안하였다. Query, Key, Value를 기반으로 하는 Multi Head Attention을 기반으로 입력과 출력에 대해 각각

Self Attention을 학습하고 이후 입력과 출력 사이의 Attention을 학습하는 구조이다.

$$MultiHead(Q, K, V) = Concat(head_1, .., head_n) W^O$$
$$head_i = Attn(QW_i^Q, KW_i^K, VW_i^V)$$
$$Attn(Q, K, V) = softmax(QK^T/\sqrt{d_k})V$$

연산의 병렬화가 가능하여 다른 모델보다 학습시간이 빠르다는 장점이 존재하며 현재 기계번역 분야에서 좋은 성능을 보이고 있는 모델이다.

3. 신경망 기계번역에서 데이터 증강기법

기계번역에서 데이터 증강기법으로 크게 2가지가 존재하며 Back Translation[4]과 Copied Translation[5]이 존재한다. Back Translation이란 기존의 훈련된 반대 방향 번역기를 사용해 단일 언어 코퍼스에 대한 번역을 진행하며 합성 synthetic 병렬 코퍼스를 만든 후, 이것을 기존 양방향 병렬 코퍼스에 추가하여 훈련에 사용하는 방식이며 NMT의 장점인 번역기를 만들면 하나의 병렬 코퍼스로 두 개의 번역 모델을 만들 수 있다는 특성에 기반한다. Copied Translation이란 반대 방향 번역기의 활용 없이 단일 언어 코퍼스 활용하는 방법론이다. 즉 소스 쪽과 타깃 쪽에 똑같은 데이터를 넣어 훈련시키는 방법이다. 위와 같은 데이터 증강기법 같은 경우 NMT의 장점인 번역기를 만들면 하나의 병렬 코퍼스로 두 개의 번역 모델을 만들 수 있다는 특성에 기반하여 최대 3배까지 데이터를 증강시킬 수 있다. 즉 100만 병렬 말뭉치가 있을 시 Backtranslation을 위한 역방향 기계번역 모델을 제작 후 데이터를 증강시키고 Copied Translation까지 적용할 경우 학습데이터가 300만 문장까지 늘어나게 된다.

4. 실험 및 실험 결과

기계번역에서 데이터 증강기법인 Back Translation과 Copied Translation 방식을 조합하여 가장 좋은 성능을 보이는 방법론을 실험을 통해 분석한다.

4.1 데이터

데이터 같은 경우 한-영 기계번역을 대상으로 실험을 진행할 것이기에 오픈 된 한-영 병렬 코퍼스를 사용한다. 영어 자막 한-영 병렬 코퍼스인 OpenSubtitles2018[1]을 사용한다. 전체 코퍼스에서 3 어절 이하인 한국어 문장은 제외시켰고 Uniform 확률로 각각 5,000 문장을 선택하여 개발(Dev) 셋과 평가(Test)셋을 구성하였다. 이 구성방법은 [7]의 논문과 동일한 구성 방법이다.

Back translation을 진행했을 시 1839242개의 학습데이터 Copied translation을 진행했을 시 2758863개의 학습데이터가 구축되었다. Back Translation에 사용한 영-한 기계번역의 BLEU1 ~ BLEU4 까지의 성능은 각각 35.5, 17.1, 9.6, 5.6이다.

<표1> 데이터 구성

	문장 개수
학습데이터	939140
+Back Translation	1839242
+Copied Translation	2758863
Validation	5000
Test	5000

4.2 모델

모델은 모든 실험에 동일하게 Transformer 모델[3]을 사용하며 Hyperparameters는 아래와 같다. [3]의 모델에서 제안하는 모델의 Hyper-Parameter를 그대로 사용하였다. Copied Translation을 이용할 경우 Vocab은 Share 한다. Tokenization의 경우 한국어, 영어 모두 BPE[7]를 적용한다.

<표2> 모델 Hyper-Parameter

Hyper-parameter	Setting
Source Vocabulary	32,004
Target Vocabulary	32,002
Batch Size	4,096

[1] http://opus.nlpl.eu/OpenSubtitles-v2018.php

Word Vector Size	512
Attention Head	8
Transformer FF	2,048
Dropout	0.1
Optimizer	Adam
Decay Method	Noam

4.3 실험 결과

실험결과 같은 경우 BLEU 점수를 성능 평가 지표로 사용하며 어절 단위로 점수를 계산하였다. 본 논문에서 사용한 동일한 데이터셋을 이용한 논문들과도 상호비교를 진행해보았다. 실험결과는 아래와 같다.

<표3> 실험 결과

	BLEU	BLEU1	BLEU2	BLEU3	BLEU4
안휘진(서강대) [9]	X	31.32	20.89	14.75	10.64
허광호(서강대) [8]	X	35.95	24.64	18.46	14.39
허광호(서강대) [8]	X	35.70	24.48	18.48	14.49
BASE (Transformer)	12.22	36.3	19.2	12.0	8.1
+Back Translation	**12.77**	**36.4**	19.4	12.1	8.0
+Copied Translation	9.69	32.8	16.5	10.2	6.8
+Back + Copied	12,48	35.2	18.5	11.5	7.7

*어절단위 BLEU 계산한 값 / [7] 형태소 단위

BASE 같은 경우 데이터증강기법을 사용하지 않고 Transformer로 학습을 진행한 모델이며 Back Translation과 Copied Translation을 각각 적용했을 때와 함께 적용했을 때의 성능비교를 진행하였다. 실험결과 Back Translation을 이용한 모델과 Copied Translation을 함께 사용한 모델의 경우 기존 BASE 보다 높은 BLEU점수를 보였으나 Copied Translation만을 이용했을 경우 오히려 BASE보다 성능이 많이 떨어지는 모습을 보였다. 이는 Copied Translation은 Back Translation과 함께 사용을 해야 효과를 볼 수 있음을 시사하며 원인으로는 한-영 기계번역에서 언어 쌍 끼리의 Character set이 다르기에 애초에 Share되는 Vocab이 적기 때문으로 판단된다. 또한 기존 동일한 데이터셋으로 연구된 논문보다 BLEU1에서 높은 성능을 보였는데 기존 연구 같은 경우 형태소 단위로 BLEU 점수를 계산 하였으며 BLEU1 ~BLEU4 까지만 공개되어 있다. Length Penalty가 반영된 전체 BLEU 점수는 공개되어 있지 않다. 또한 기존 연구 같은 경우 학습 데이터 외 추가 리소스를 이용하였으나[8] 본 논문은 추가 리소스를 사용하지 않았다.

<표4> 원본코퍼스와 합성코퍼스의 상대적 비율을 적용한 실험

	BLEU	BLEU1	BLEU2	BLEU3	BLEU4
Back Translation 2:1 비율	13.05	36.6	19.8	12.4	8.2
Back Translation 3:1 비율	12.83	36.2	19.5	12.1	7.8
Back Translation 4:1 비율	12.79	36.0	19.3	12.1	7.8
Back Translation 3:2 비율	12.60	36.5	19.8	12.4	8.2
Back Translation 4:3 비율	12.81	36.8	19.9	12.3	7.9
(Back+Copied) 2:1 비율	12.86	**36.8**	**20.3**	**12.9**	**8.8**
(Back+Copied) 3:1 비율	12.95	36.6	19.8	12.4	8.2
(Back+Copied) 4:1 비율	12.67	35.9	19.2	12.0	7.9
(Back+Copied) 3:2 비율	12.46	36.0	19.0	11.9	7.8
(Back+Copied) 4:3 비율	**13.09**	36.4	19.8	12.4	8.1

*어절단위 BLEU 계산한 값

원본 코퍼스와 합성 코퍼스의 비율을 상대적으로 두고 학습을 진행해보았다. Batch를 구성 할 때 2개의 코퍼스가 있을 시 상대적 비율을 두어 구성하는 방법이다.

2대1, 3대1, 4대1, 3대2, 4대3의 비율로 구성하였을 때의 모든 실험을 진행했으며 표3의 실험결과에서 Back Translation만을 사용했을 때 가장 좋은 성능이 나왔음으로 Back Translation만을 적용했을 때와 Back Translation과 Copied Translation을 함께 적용했을 때의 실험을 나누어서 진행하였다.

실험결과 Back Translation과 Copied Translation을 함께 적용하여, 4대3의 상대적 비율을 적용하여 학습을 진행했을 때 가장 높은 BLEU 점수를 보였다. 이는 합성 코퍼스의 양만 많다고 높은 성능의 모델이 만들어짐이 아님을 시사하며 원본 코퍼스와 합성 코퍼스의 적당한 비율을 가지고 훈련을 진행하는 것이 좋은 성능의 모델을 만들 수 있음을 시사한다.

<표3>에서 Back Translation과 Copied Translation을 함께 적용했을 때는 Back Translation만을 이용했을 때 보다 오히려 BLEU 점수가 떨어졌는데 <표4>와 같이 상대적 비율을 적용하여 학습을 진행했을 시 Copied Translation을 함께 사용하는 것이 효과가 있음을 확인하였다. 결론적으로 Back Translation과 Copied Translation과 같은 데이터 증강기법이 기계번역 성능향상에 도움을 준다.

5. 결론

본 논문은 기계번역의 데이터 증강기법들에 대해 다양한 실험을 진행하였다. Back Translation과 Copied Translation과 같은 데이터 증강기법이 기계번역 성능향상에 도움을 줌을 볼 수 있었으며 Batch를 구성할 때 상대적 가중치를 두는 것이 성능향상에 도움이 됨을 알 수 있었다. 추후 합성 코퍼스에 병렬 코퍼스 필터링 기법을 적용하여 기계번역 성능 향상 여부에 대해 연구해볼 예정이다.

6. 감사의 글

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터지원사업 (IITP-2018-0-01405), 2017년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.NRF-2017M3C4A7068189).

7. 참고문헌

[1]Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation By Jointly Learning To Align and Translate. In ICLR, pages 1–15

[2]Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proc of EMNLP.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008, 2017.

[4] Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Improving neural machine translation models with monolingual data." arXiv preprint arXiv:1511.06709 (2015).

[5Edunov, Sergey, et al. "Understanding back-translation at scale." arXiv preprint arXiv:1808.09381 (2018).

[6] Currey, Anna, Antonio Valerio Miceli Barone, and Kenneth Heafield. "Copied monolingual data improves low-resource neural machine translation." Proceedings of the Second Conference on Machine Translation. 2017.

[7] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proc. of ACL

[8] 허광호, (2018), 신경망 기반 기계 번역을 위한 역-번역을 이용한 한영 병렬 코퍼스 확장, HCLT2018

[9] 이재환, 김보성, 허광호, 고영중, 서정연. (2018). Subword 유닛을 이용한 영어-한국어 신경망 기계번역. 한국정보과학회 학술발표논문집, (), 586-588.