Unsupervised Machine Translation using Monolingual Corpora Only

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, Marc'Aurelio Ranzato

ICLR 2018

오지은

Introduction

WHY monolingual?

- 이제까지의 Neural Machine Translation: **supervised** learning:
  - 아주 많은 **parallel sentence**가 필요함
  - 그러나 parallel sentence는 기본적으로 만들기 어렵고(전문 번역가 필요), low-resource 언어의 경우 특히 찾기 힘들다
- 한편 **monolingual** corpora: 많고, 찾기 쉬움
  - low-resource 언어라도 monolingual data는 어느 정도 보유하고 있음

→ **monolingual corpora를 NMT에 이용**할 수 있게 해보자!

Back-translation

- **Back-translation**: monolingual corpora를 NMT에 이용하려고 한 여러 시도 중, 가장 주목할 만한 것
  - Auxiliary translation system을 따로 학습해서, target 언어로부터 source 언어로 번역된 문장을 생성 (synthetic data) - 즉 **현재의 모델을 사용해 번역 데이터를 생성**
  - 이 인위적으로 만들어진 데이터를 원래의 parallel data에 섞어서 학습에 사용
  - Back-translated 문장의 품질은 최종 학습 결과에 어느 정도 영향을 미치지만, 대단히 큰 영향을 주지는 않음

- BLEU score가 6점 차이나는 synthetic 데이터들로 학습했을 때, 최종 결과의 BLEU는 0.6~0.7점 차이

Unsupervised

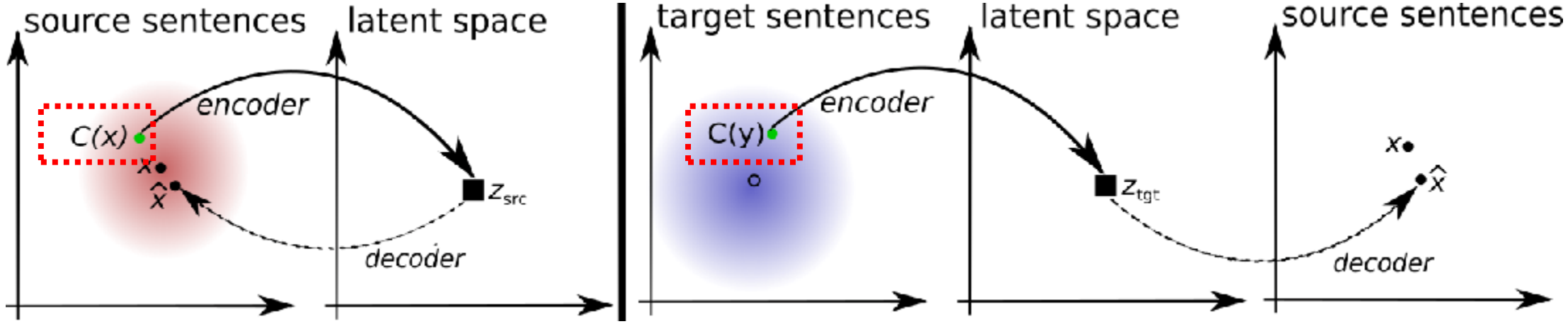
- Back-translation은 monolingual corpora를 활용하긴 했지만, 어쨌거나 **supervised learning**임
- 그렇다면, **아무런 종류의 supervision 없이**도 NMT를 수행하도록 학습할 수 있을까?
- 그렇게 할 수 있다면:

  Annotation 없는 새로운 language pair를 만날 때마다 이 방법을 사용하여 NMT를 수행할 수 있음

  Semi-supervised approach들에 대해 좋은 lower bound를 제공함

KEY idea

- 핵심: 두 언어 사이에 **common latent space**를 구축하고, 그 공간으로부터 각 도메인(언어)에 대해 **reconstruct**함으로써 번역을 학습한다
- 이때 따르는 원칙 2가지:
  1. 모델은 어떤 문장을 그 문장의 noisy 버전으로부터 원문을 재구성한다 (**denoising**)
  2. 모델은 어떤 문장을 그 문장의 noisy translation으로부터 원문을 재구성한다



Method

method

- Architecture: **sequence to sequence with attention** (Bahdanau et al. 2015)
  - 인코더: source/target sentence를 latent space에 인코딩
  - 디코더: latent space로부터 source/target sentence를 디코딩

- Overview:
  1. **Denoising Auto-Encoding**: noisy input으로부터 원래의 문장 복원
  2. **Cross-domain training**: noisy translation으로부터 원래의 문장 복원
  3. **Adversarial training**: 두 언어가 같은 공간에 매핑되도록 **discriminator** 사용

1. Denoising Auto-Encoding (**monolingual**)
   - 아무 제약 없이 x로부터 x를 예측하도록 학습하면, 모델은 단순히 입력을 출력으로 복사한다
   - 이것을 막기 위해 x를 noise로 왜곡시켜 **C(x)로부터 원문 x를** 만들어내도록 학습한다 (=x^)
   - Noising의 방법:
     ① Drop (일부 단어 삭제)
     ② Shuffle (문장 내 일부 단어 순서 변경)
   - 이 항목을 위한 objective function:

$$\mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, \ell) = \mathbb{E}_{x \sim \mathcal{D}_\ell, \hat{x} \sim d(e(C(x), \ell), \ell)} [\Delta(\hat{x}, x)]$$

x^은 noised x의 reconstruction이라는 뜻

x^과 x 사이의 token-level cross-entropy loss

x는 언어 ℓ 에 속한다는 뜻
ℓ = 소스/타겟

X^: reconstructed x
C(x): 노이즈가 있는 x
e(x, ℓ): 언어 ℓ 에 대한 x의 인코딩
d(x, ℓ) 언어 ℓ 에 대한 x의 디코딩

2. Cross Domain Training (**cross-lingual**)

- 언어 1에 속한 문장 x에 대해 언어 2에 속한 translation y를 만들어낸다
- 그 translation에 noise를 적용한 **C(y)로부터 원문 x를 재구성**한다 (=x^)
- Translation y는 **back-translation**으로 얻어낸다

→ 즉, 현재의 모델 M을 이용해 번역을 생성한다. M(x) = y

$$\mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, \ell_1, \ell_2) = \mathbb{E}_{x \sim \mathcal{D}_{\ell_1}, \hat{x} \sim d(e(C(M(x)), \ell_2), \ell_1)} [\Delta(\hat{x}, x)]$$

X는 언어 1에 속함

언어2에 대한 noisy translation을 인코딩 후 디코딩(재구성)

x^과 x 사이의 token-level cross-entropy loss

M(x) : 모델의 이전 아웃풋=번역=y
C(y) : noisy translation (언어 2에 속함)
e(x, ℓ) : 언어 ℓ에 대한 인코딩
d(x, ℓ) : 언어 ℓ 에 대한 디코딩

3. Adversarial Training

- 두 언어가 어지간히 비슷하지 않으면, 디코더가 다른 언어의 latent vector에서 제대로 된 번역을 생성하기 힘들다
- **다른 언어더라도 비슷한 문장이면 가까운 latent space 안에 들어가야 함**
- 인코딩된 latent space가 **원래 어떤 언어인지를 구별**하는 discriminator 사용
- 인코더는 이 discriminator를 속이도록 학습
- 인코더는 어떤 언어든지 관계없이 같은 공간 안에 feature를 학습하도록 하고, 디코더는 그 feature부터 어떤 언어든지 관계없이 재구성하도록 하는 것이 목표

Cross-entropy loss

인코더 아웃풋에 대해 적용됨

$$\mathcal{L}_{adv}(\theta_{\text{enc}}, \mathcal{Z}|\theta_D) = -\mathbb{E}_{(x_i,\ell_i)}[\log p_D(\ell_j|e(x_i,\ell_i))]$$

세타 enc: 인코더의 파라미터
Z: 인코더 아웃풋(즉 latent vector)
세타 D: discriminator의 파라미터

PD: binary prediction. 0 (소스)또는 1(타겟)

이 항은 discriminator loss이기도 함

- Final objective function

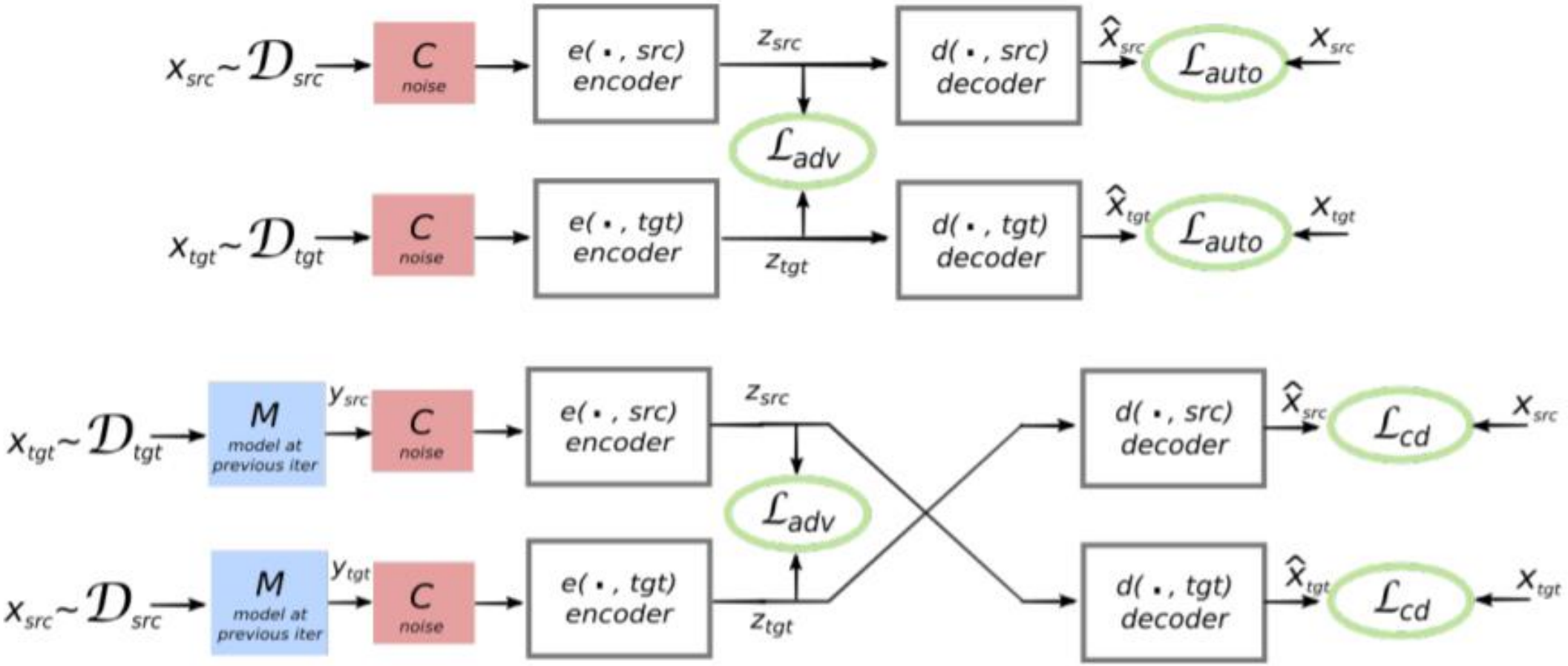
$$\mathcal{L}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}) = \lambda_{auto}[\mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src) + \mathcal{L}_{auto}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt)] +$$
$$\lambda_{cd}[\mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, src, tgt) + \mathcal{L}_{cd}(\theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z}, tgt, src)] +$$
$$\lambda_{adv}\mathcal{L}_{adv}(\theta_{\text{enc}}, \mathcal{Z}|\theta_D)$$

hyperparameter (가중치)
실험은 모두 1로 진행되었음

+ 여기에 discriminator loss가 병렬로 업데이트된다

Training

training


$x_{src} \sim \mathcal{D}_{src}$
C
noise
e(• , src)
encoder
z_{src}
d(• , src)
decoder
$\hat{x}_{src}$
$\mathcal{L}_{auto}$
x_{src}
$\mathcal{L}_{adv}$
$x_{tgt} \sim \mathcal{D}_{tgt}$
C
noise
e(• , tgt)
encoder
z_{tgt}
d(• , tgt)
decoder
$\hat{x}_{tgt}$
$\mathcal{L}_{auto}$
x_{tgt}
$x_{tgt} \sim \mathcal{D}_{tgt}$
M
model at
previous iter
y_{src}
C
noise
e(• , src)
encoder
z_{src}
d(• , src)
decoder
$\hat{x}_{src}$
$\mathcal{L}_{cd}$
x_{src}
$\mathcal{L}_{adv}$
$x_{src} \sim \mathcal{D}_{src}$
M
model at
previous iter
y_{tgt}
C
noise
e(• , tgt)
encoder
z_{tgt}
d(• , tgt)
decoder
$\hat{x}_{tgt}$
$\mathcal{L}_{cd}$
x_{tgt}

- Iterative training
- 모델의 이전 아웃풋을 사용해 현재 모델을 업데이트하면서 **반복적으로** 모델을 개선

① 초기화: word-by-word translation 시스템으로 모델 M 초기화

② 모델 M으로 데이터셋 번역

③ Discriminator, encoder, decoder 학습하여 업데이트 (=minimize objective)

④ 학습된 인코더와 디코더로 업데이트된 모델 M(t+1) 생성

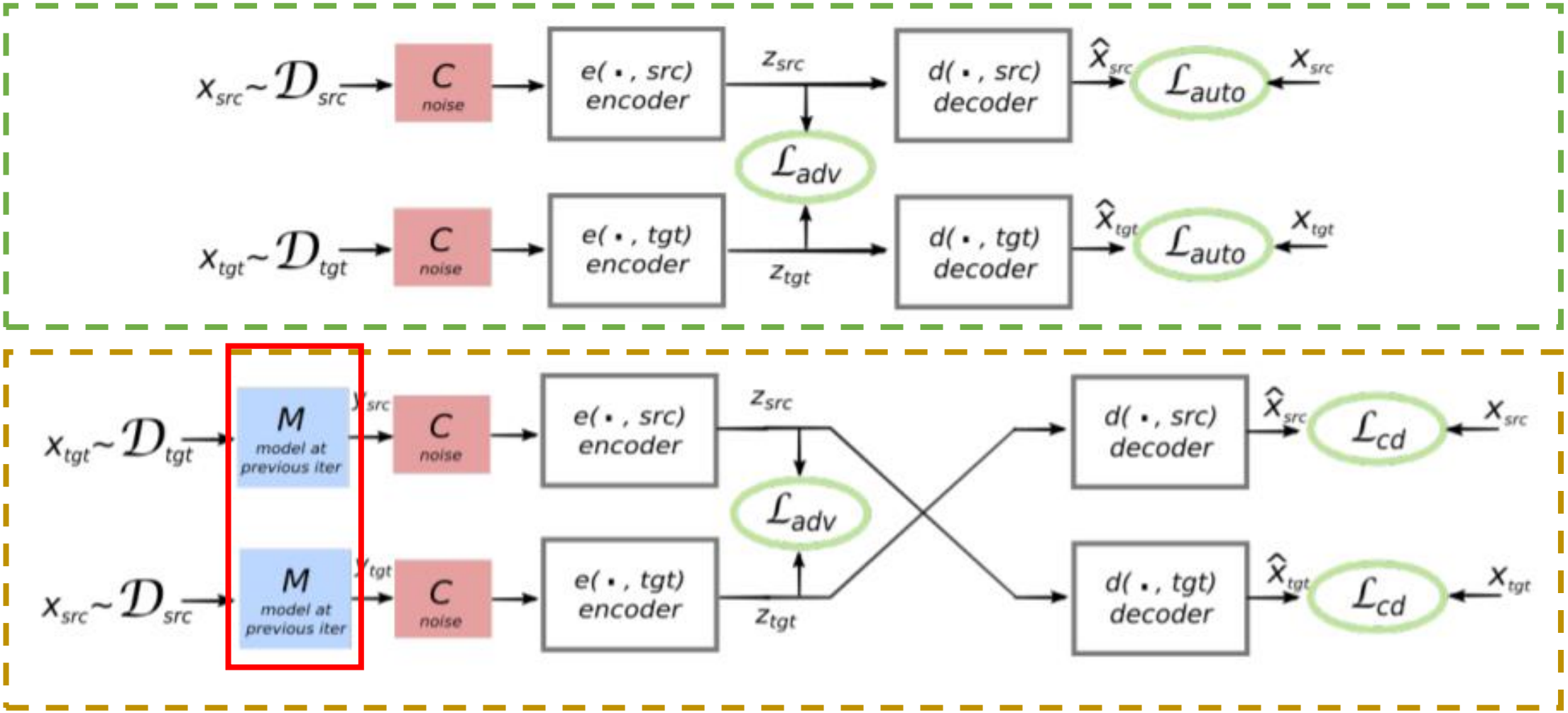
⑤ 반복

Algorithm 1 Unsupervised Training for Machine Translation

procedure TRAINING($\mathcal{D}_{src}, \mathcal{D}_{tgt}, T$)
    Infer bilingual dictionary using monolingual data (Conneau et al., 2017)
    $M^{(1)} \leftarrow$ unsupervised word-by-word translation model using the inferred dictionary
    for $t = 1, T$ **do**
        using $M^{(t)}$, translate each monolingual dataset
        // discriminator training & model training as in eq. 4
        $\theta_{\text{discr}} \leftarrow \arg\min \mathcal{L}_D, \quad \theta_{\text{enc}}, \theta_{\text{dec}}, \mathcal{Z} \leftarrow \arg\min \mathcal{L}$
        $M^{(t+1)} \leftarrow e^{(t)} \circ d^{(t)}$ // update MT model
    end for
    return $M^{(T+1)}$
end procedure

training

Auto-encoding (noisy x로부터 깨끗한 x 생성)



translation (noisy y로부터 깨끗한 x 생성)
이때 y는 모델의 **이전 iteration에서 나온 번역**

training

- 모델은 (최초의 모델 M_0이라도) 입력 문장에 대해 최소한의 어떤 정보를 얻는다
- 인코더는 denoise하도록 학습했으므로, 좀더 깨끗한 버전의 representation을 feature space에 만들어낼 것이다
- 디코더는 noisy feature를 가지고 noiseless output을 생성하도록 학습했으므로, 이 인코더와 디코더를 합하면 더 나은 translation이 만들어질 것이다

→ 다음 iteration에는 더 나은 back-translation을 할 수 있을 것이다

→ 반복!

Validation

validation

❖ Hyperparameter를 고를 때 혹은 stop criterion을 정할 때 어떤 기준으로 할 것인가?

1. 언어 1의 입력 x에서 언어 2로 번역 후, 그 결과를 다시 언어 1로 번역(x^)
2. x와 x^으로 BLEU 스코어 측정
3. 위의 과정을 언어1 → 언어2와 언어2 →언어1로 각각 진행 후 평균을 취함

$$MS(e, d, \mathcal{D}_{src}, \mathcal{D}_{tgt}) = \frac{1}{2}\mathbb{E}_{x\sim\mathcal{D}_{src}}\left[\mathrm{BLEU}(x, M_{src\to tgt} \circ M_{tgt\to src}(x))\right] + \frac{1}{2}\mathbb{E}_{x\sim\mathcal{D}_{tgt}}\left[\mathrm{BLEU}(x, M_{tgt\to src} \circ M_{src\to tgt}(x))\right]$$

MS: model selection criterion

Experiments

experiment

- Datasets:
  - WMT'14 English-French
  - WMT'16 English-German
  - Multi30k-Task1

- Baselines:
  - Word-by-word translation: 저자들의 이전 연구. 비슷한 언어에 대해 잘 작동함
  - Word reordering: 위의 WBW로 번역한 결과를 permutation해서 학습한 LM (WMT만 학습)
  - Oracle Word Reordering: WBW에서 나온 단어로 만든 best possible generation

    (모델이 단어 교체 없이 만들 수 있는 upper-bound 성능)

experiment

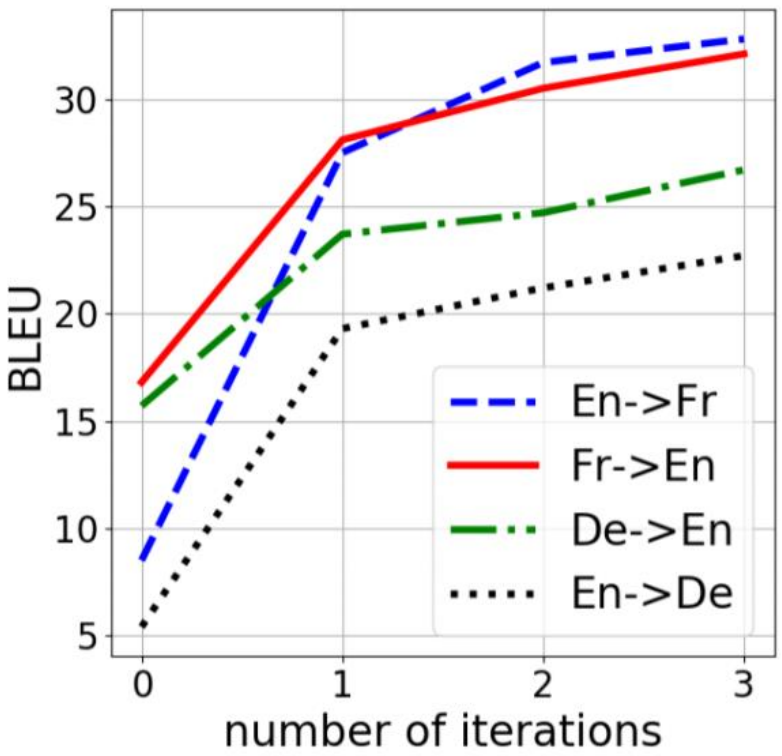
	Multi30k-Task1				WMT			
	en-fr	fr-en	de-en	en-de	en-fr	fr-en	de-en	en-de
Supervised	56.83	50.77	38.38	35.16	27.97	26.13	25.61	21.33
word-by-word	8.54	16.77	15.72	5.39	6.28	10.09	10.77	7.06
word reordering	-	-	-	-	6.68	11.69	10.84	6.70
oracle word reordering	11.62	24.88	18.27	6.79	10.12	20.64	19.42	11.57
Our model: 1st iteration	27.48	28.07	23.69	19.32	12.10	11.79	11.10	8.86
Our model: 2nd iteration	31.72	30.49	24.73	21.16	14.42	13.49	13.25	9.75
Our model: 3rd iteration	32.76	32.07	26.26	22.74	15.05	14.31	13.33	9.64

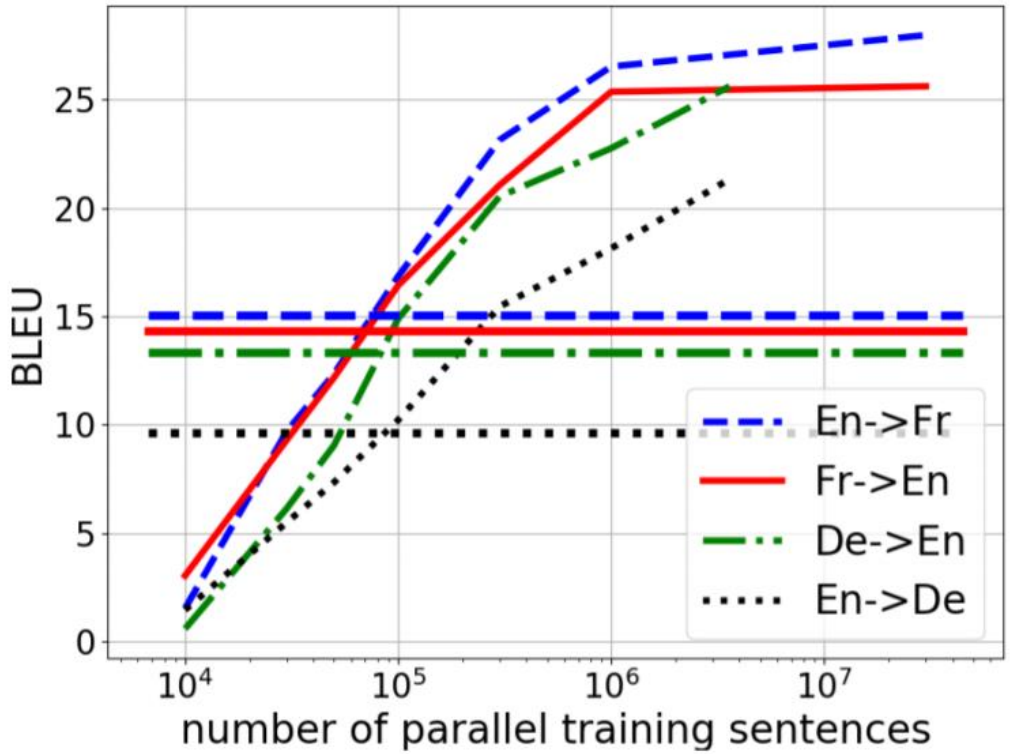
① Word-by-word는 타겟 언어가 영어일 때 성능이 좋다

② Word reordering은 WBW의 성능을 약간만 개선한다

③ Baseline들보다 unsupervised training 모델이 더 좋다 (→ 이 모델이 어순을 좀더 잘 맞출 뿐 아니라(reorder) 알맞은 substitution도 수행한다)

experiment





- 첫 번째 iteration부터 성능이 높다 (빨리 수렴한다)
- 10만 parallel 데이터로 학습한(supervised) 수준의 성능을 1.5천만 monolingual 데이터로 학습했을 때 달성한다

experiment

Source	un homme est debout près d' une série de jeux vidéo dans un bar .
Iteration 0	a man is seated near a series of games video in a bar .
Iteration 1	a man is standing near a closeup of other games in a bar .
Iteration 2	a man is standing near a bunch of video video game in a bar .
Iteration 3	a man is standing near a bunch of video games in a bar .
Reference	**a man is standing by a group of video games in a bar .**
Source	une femme aux cheveux roses habillée en noir parle à un homme .
Iteration 0	a woman at hair roses dressed in black speaks to a man .
Iteration 1	a woman at glasses dressed in black talking to a man .
Iteration 2	a woman at pink hair dressed in black speaks to a man .
Iteration 3	a woman with pink hair dressed in black is talking to a man .
Reference	**a woman with pink hair dressed in black talks to a man .**
Source	une photo d' une rue bondée en ville .
Iteration 0	a photo a street crowded in city .
Iteration 1	a picture of a street crowded in a city .
Iteration 2	a picture of a crowded city street .
Iteration 3	a picture of a crowded street in a city .
Reference	**a view of a crowded city street .**

Iter0: WBW

(어순, 즉 word reordering에 문제가 있음을 확인할 수 있음

Iter3에서 상당히 좋은 수준의 번역을 내는 것을 확인할 수 있음

ablation

	en-fr	fr-en	de-en	en-de
$\lambda_{cd} = 0$	25.44	27.14	20.56	14.42
Without pretraining	25.29	26.10	21.44	17.23
Without pretraining, $\lambda_{cd} = 0$	8.78	9.15	7.52	6.24
Without noise, $C(x) = x$	16.76	16.85	16.85	14.61
$\lambda_{auto} = 0$	24.32	20.02	19.10	14.74
$\lambda_{adv} = 0$	24.12	22.74	19.87	15.13
Full	**27.48**	**28.07**	**23.69**	**19.32**

- most critical component: unsupervised word alignment technique (back-translation 또는 워드임베딩)

  → pretrained embeddings과 back-translation 둘 중 하나만 사용하면 성능이 별로 떨어지지 않지만, 둘 다 사용하지 않으면 성능이 비약적으로 떨어짐

- Adversarial training과 auto-encoding 둘 다 성능에 영향을 미치는 요소임
- Noise 역시 크게 중요한 요소임

Conclusion

conclusion

- 결론: **아무런 supervision 없이도** 기계번역을 학습할 수 있다!
- Contribution:
  1. Low-resource 언어에 대해 유용함 (parallel data가 없어도 학습 가능)
  2. Semi-supervised machine translation의 길이 열림

- Comment: 제목의 monolingual corpora라는 말이 misleading한 것 같다

  non-parallel이라고 하는 편이 더 맞지 않을까?

End of Document