



DSBA CS224n 2021 Study

[Special Lecture 1]

Low Resource Machine Translation

(CS224n 2020 Winter)



고려대학교 산업경영공학과
Data Science & Business Analytics Lab

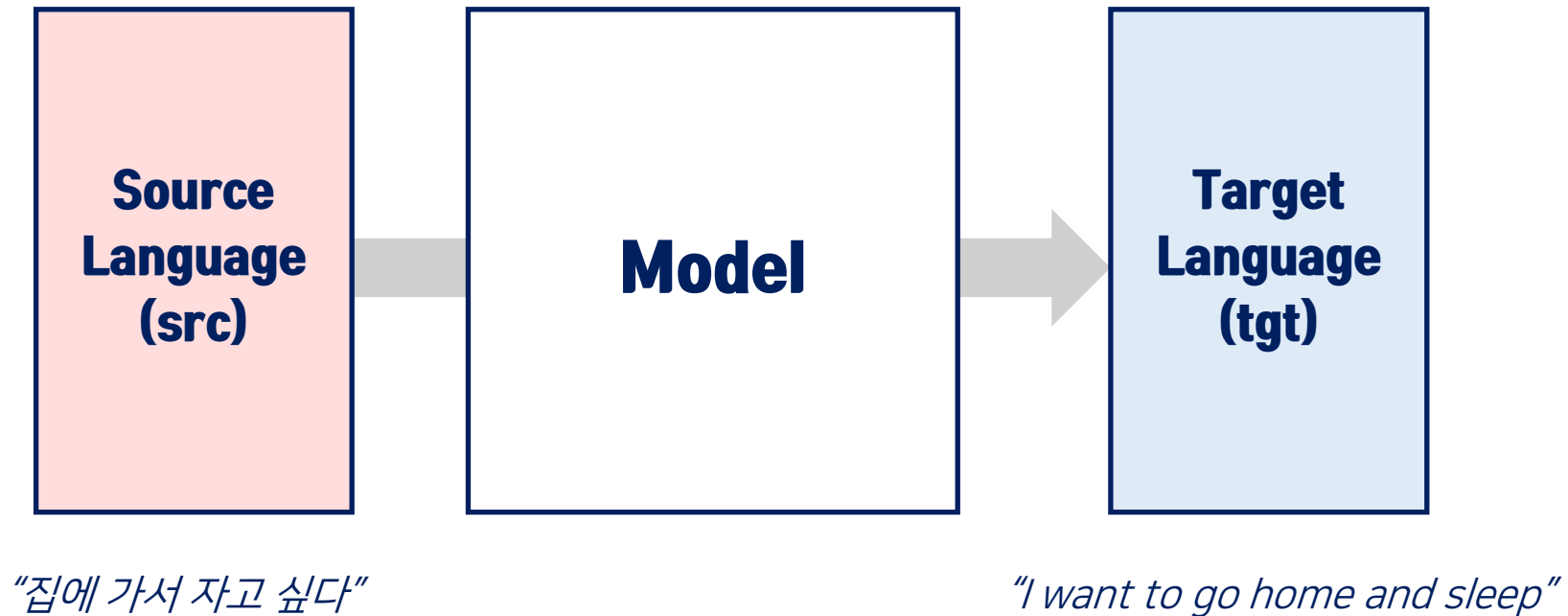
발표자 : 고유경

- 1 Low Resource Machine Translation
- 2 Learning Methods for LRMT
- 3 Paper 1
- 4 Paper 2

1

Low Resource Machine Translation

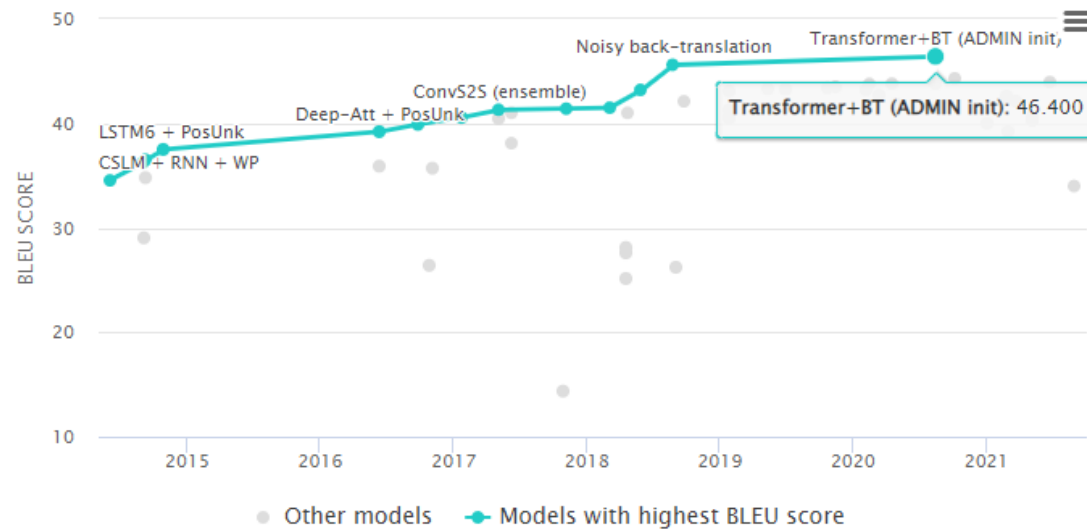
- **NMT(Neural Machine Translation):** source 언어를 target 언어로 번역하는 모델 학습



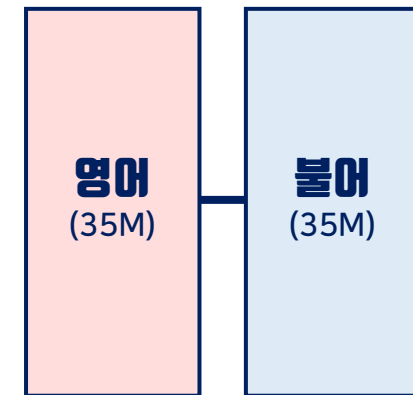
High performance of modern MT systems

- 비약적 발전을 이룬 기계 번역(NMT) 연구: 높은 성능 달성
- 현재 번역 시스템은 **대량의 parallel corpus(High Resource Language pair)**로 학습된 언어에 대해서 잘 작동함 (power of supervision)

WMT 2014 English-French Leaderboard



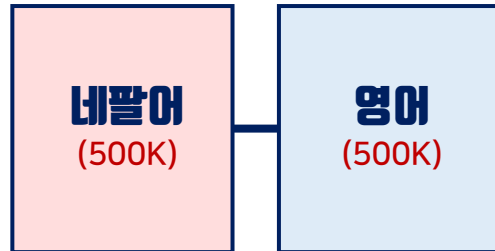
parallel(병렬) corpus



High Resource Languages

What if Low Resource Languages?

① 소량의 parallel corpora



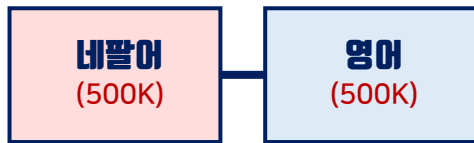
② monolingual(단일) corpora



Low Resource Languages: parallel corpora를 기준으로 10K개 이하인 언어 쌍

① 소량의 parallel corpora

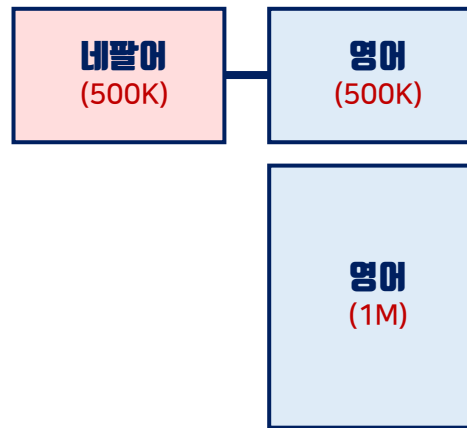
Supervised MT



En-Ne	Ne-En	En-Si	Si-En
4.3	7.6	1.0	6.7

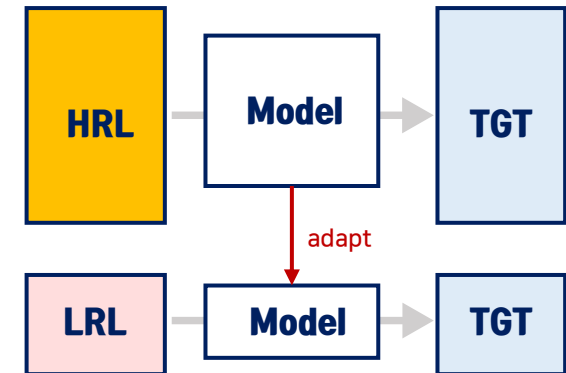
- 낮은 성능

Semi-supervised MT



- Feat. Data Augmentation
- 대부분의 경우 추가적인 monolingual 데이터 활용

Transfer Learning

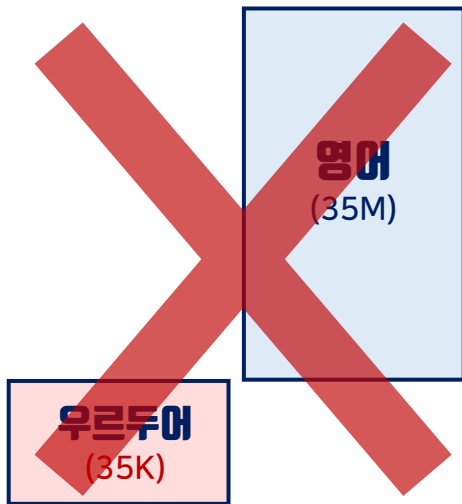


- High resource language pair로 학습한 번역 모델을 fine-tuning,
- HRL과 LRL이 비슷한 언어일 때만 가능 (ex. HRL: 터키어, LRL: 아제르바이잔어)

② monolingual(단일) corpora

Supervised MT

Semi-supervised MT



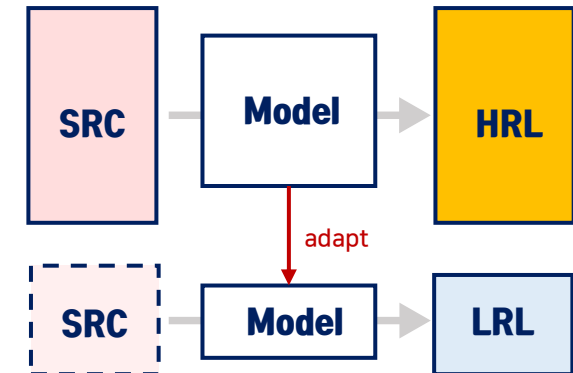
- parallel 데이터의 부재로 불가능

Unsupervised MT



- HRL을 중심으로 연구
- Monolingual 데이터 양에 따라 성능 차이 크기 때문에 LRL의 경우 성능 매우 낮음(< 1)

Transfer Learning



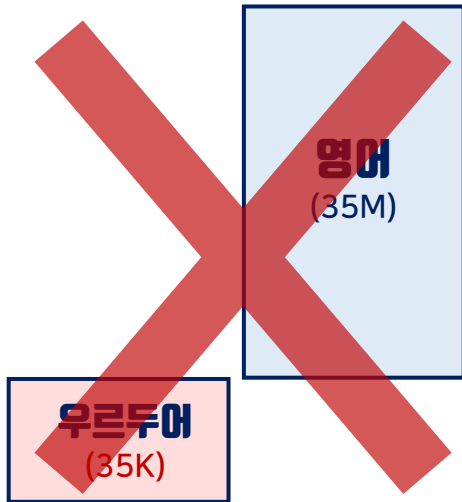
- HRL로부터 embedding 및 model을 transfer 받아 pseudo src 생성 후 번역 모델 fine-tuning
- HRL과 LRL의 유사성 매우 중요

② monolingual(단일) corpora

평가용 parallel 데이터셋이 없으면 test 불가능!

Supervised MT

Semi-supervised MT



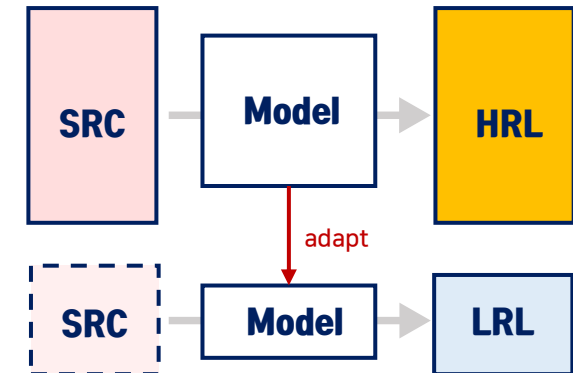
- parallel 데이터의 부재로 불가능

Unsupervised MT



- HRL을 중심으로 연구
- Monolingual 데이터 양에 따라 성능 차이 크기 때문에 LRL의 경우 성능 매우 낮음(< 1)

Transfer Learning



- HRL로부터 embedding 및 model을 transfer 받아 pseudo src 생성 후 번역 모델 fine-tuning
- HRL과 LRL의 유사성 매우 중요

② monolingual(단일) corpora

- ❖ Real world에서는 parallel data에 포함되지 않는 언어들이 무수히 많이 존재함 (심지어 같은 언어 내에도 방언과 같은 variety 존재)



parallel



monolingual



- ❖ 좋은 퀄리티의 학습, 평가 데이터 구축 필요 (Facebook AI Research가 선도)



- FLORES (2019): 영어-네팔어, 영어-싱할라어
- FLORES-101 (2021): 101개 언어로 번역(multilingual), 도메인 다양화

About FLORES Dataset

- Evaluation이 중요한 Low Resource Machine Translation, 평가할 때는 parallel 데이터셋 필수
 - 학습용 parallel data는 많지만, domain이 다양한 high-quality Evaluation 데이터셋 부재
- ✓ **기존 Evaluation 데이터셋 한계**
- Evaluation 데이터셋 양 자체가 적고, 도메인과 언어 종류가 다양하지 않음
 - 영어로만 parallel 데이터셋 구성 (to and from only English)
 - 대부분 자동화된 과정으로 데이터셋 구축 -> 낮은 퀄리티의 번역문
 - 2019년 Facebook이 제작한 LRMT 평가 데이터셋 FLORES ver1: 영어-싱할라어 / 영어-네팔어
- ✓ **2021년 Facebook에서 Multilingual 버전의 도메인 다양화한 FLORES-101 공개**
- 영어 위키피디아에서 3001개의 문장 추출하여 101개의 언어로 번역
- ✓ **Contribution**
1. 다양한 도메인 및 주제를 담고 low resource 언어도 포함한 high-quality 데이터셋
 2. 10,100개의 many-to-many translation 조합 구축 가능(multilingual)
 3. annotation 과정을 문서화하여 향후 새로운 데이터셋 구축에 도움이 되도록 함
 4. meta data도 함께 제공하여 번역 뿐만 아니라 다른 task에도 적용 가능
 5. Sentence piece tokenization을 기반으로 한 BLEU metric 제안



Test on Baseline Model

M2M-124 model with 615M parameters

101

101

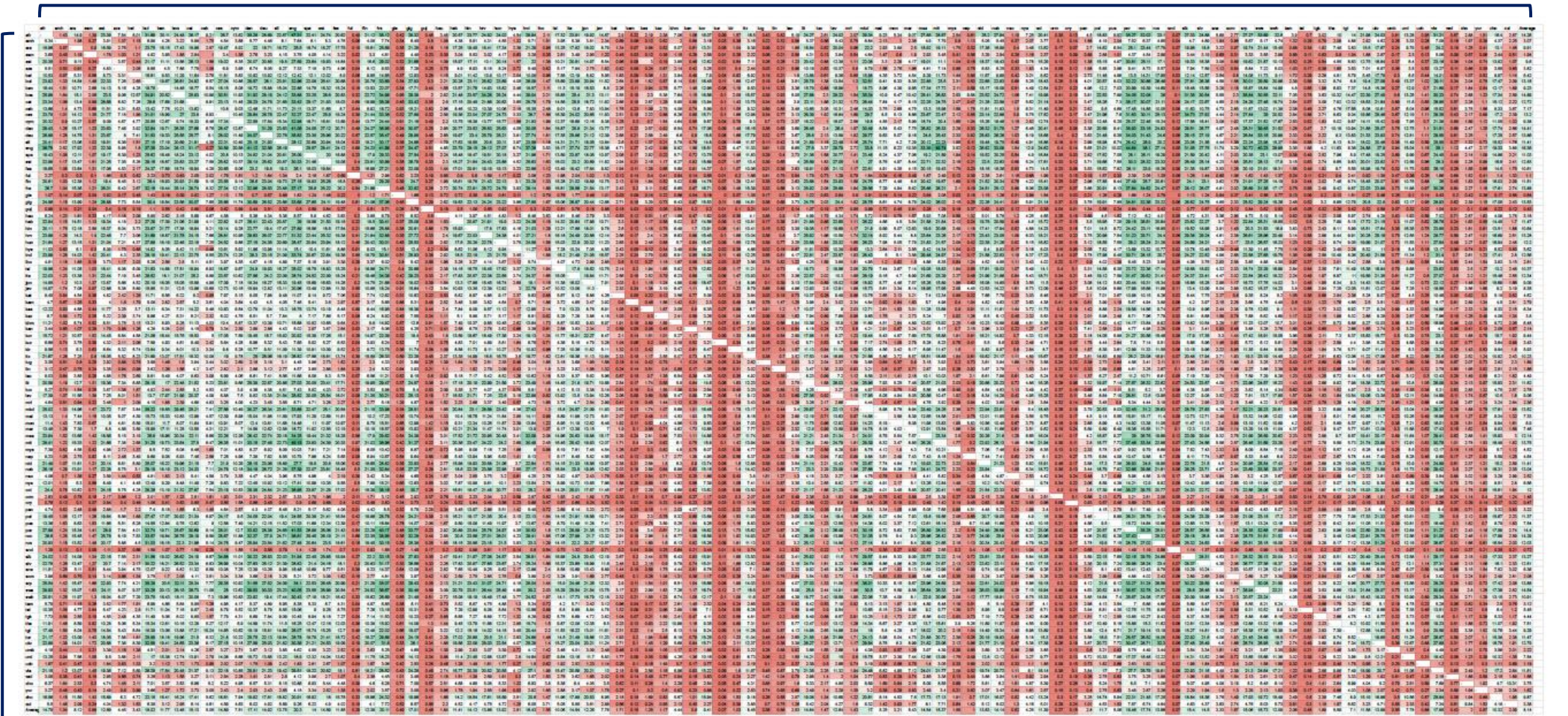




Figure 8: spBLEU on the M2M MMT model on all the language pairs of FLORES-101 dev-test set. Cell (i,j) reports spBLEU for translating from language i to language j. Therefore, each column shows spBLEU for translating in the same target language but using various source languages. Vice versa, each row shows spBLEU for translating into various target languages when starting from the same source language.


Low Resource Machine Translation

WMT 2021

- EMNLP 2021 SIXTH CONFERENCE ON MACHINE TRANSLATION (WMT21): Nov 2021 [[link](#)]
- Shared Task: Large-Scale Multilingual Machine Translation
 - **Small Track #1** : 5 Central/East European languages, 30 directions: Croatian, Hungarian, Estonian, Serbian, Macedonian, English
 - **Small Track #2**: 5 South East Asian languages, 30 directions: ~~Sundanese~~, Javanese, Indonesian, Malay, Tagalog, Tamil, English (n
 - **Large Track**: All Languages, to and from English. Full list at the bottom of this page.
- **Training data**: Parallel data from **Opus** Monolingual data from Wikipedia
- **Evaluation Data**: **Flores 101**, dev and devtest
- 리더보드: Microsoft의 DeltaLM (a generic encoder-decoder pretrained language model, using high-quality parallel data and back-translation data)

MODEL LEADERBOARD - FLORES MT EVALUATION (FULL) 	
Model	▼ Average BLEU
DeltaLM+Zcode (Microsoft)	16.63
615m (Baohao Liao)	7.55
m2m-124-175m (Guillaume Wenzek)	6.05

MODEL LEADERBOARD - FLORES MT EVALUATION (SMALL TASK 1) 	
Model	▼ Average BLEU
DeltaLM+Zcode (Microsoft-Small)	37.59
615m (Baohao Liao)	34.96
distill (Wen Lai)	31.86
m2m-615m (Guillaume Wenzek)	28.23
m2m-124-175m (Guillaume Wenzek)	21.30

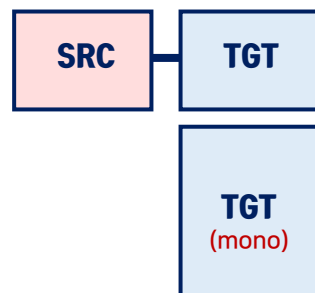
MODEL LEADERBOARD - FLORES MT EVALUATION (SMALL TASK 2) 	
Model	▼ Average BLEU
DeltaLM+Zcode (Microsoft-Small)	33.89
615m (Baohao Liao)	33.34
TenTrans (Wanying Xie)	28.89
adaavg (Danni Liu)	28.64
huawei-tsc1 (huaweitsc)	28.40
srph-large (jcbaiseacruz02)	22.97

2

Learning Methods for LRMT

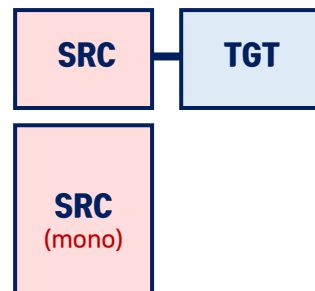
Semi-supervised Learning: parallel (+ monolingual)

Mono(TGT)



Back-Translation

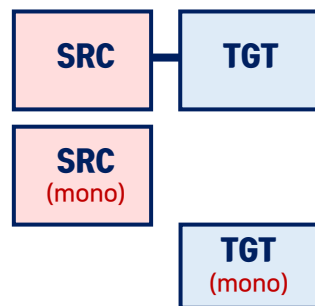
Mono(SRC)



DAE
(Denoising Auto Encoding)

Self-training

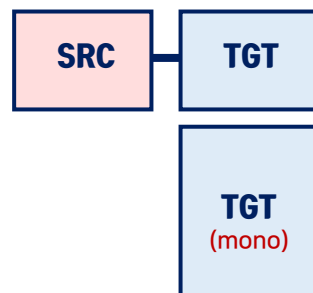
Mono(TGT, SRC)



Iterative
Back-Translation

Semi-supervised Learning: parallel (+ monolingual)

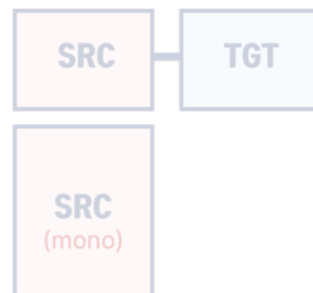
Mono(TGT)



Back-Translation

- parallel 데이터로 역번역(tgt->src) 모델 학습하여 tgt(mono)로 pseudo src 생성
- 증강된 데이터로 번역(src->tgt) 모델 학습

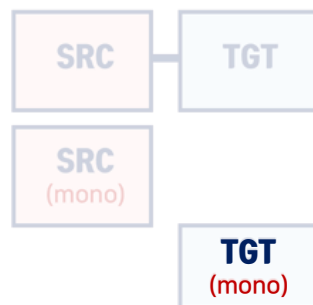
Mono(SRC)



DAE
(Denoising Auto Encoding)

Self-training

Mono(TGT, SRC)



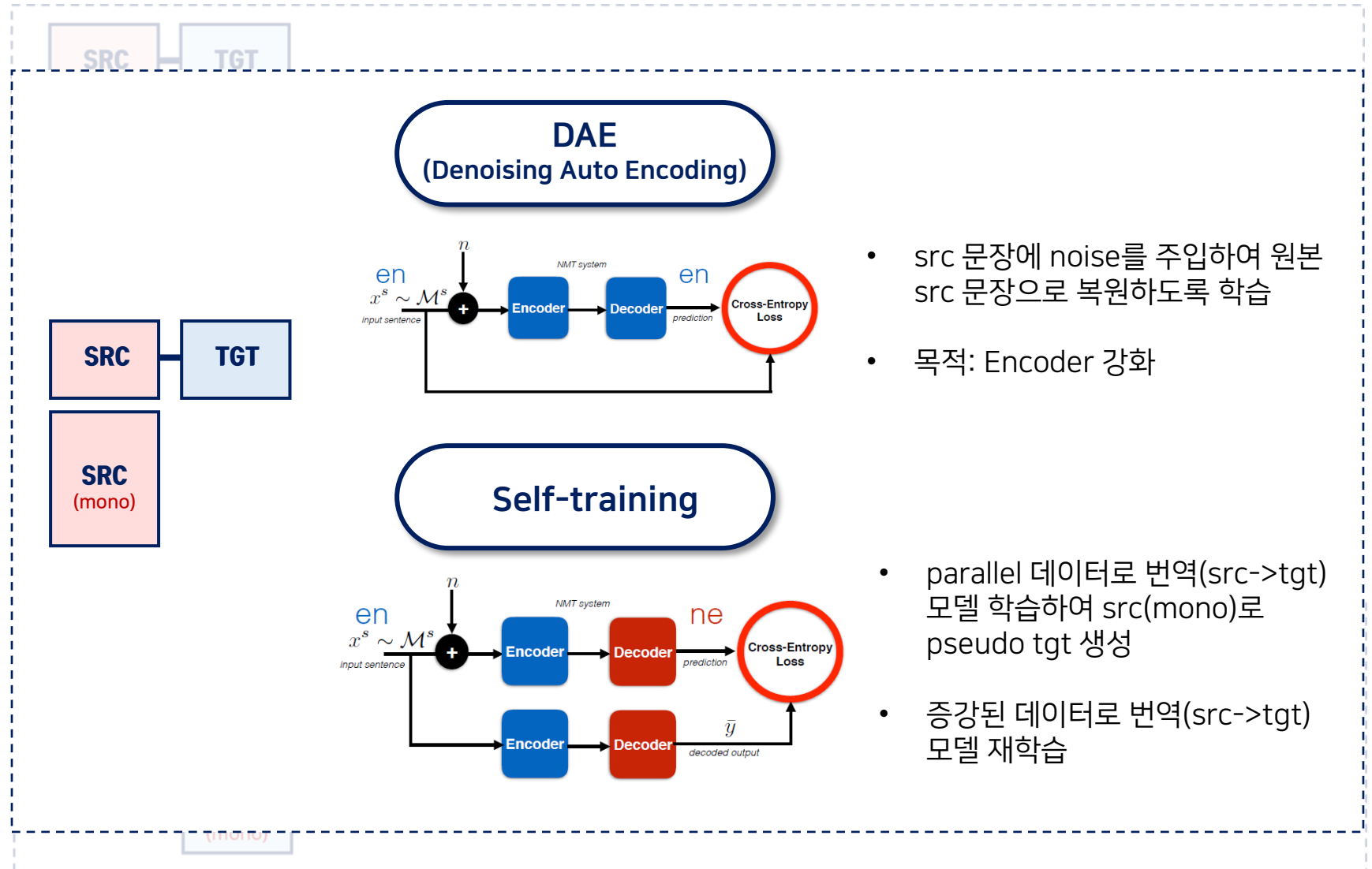
Iterative
Back-Translation

Semi-supervised Learning: parallel (+ monolingual)

Mono(TGT)

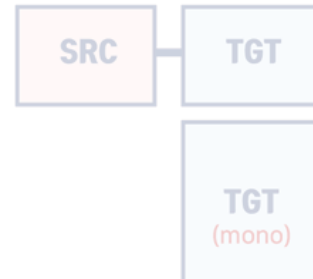
Mono(SRC)

Mono(TGT, SRC)



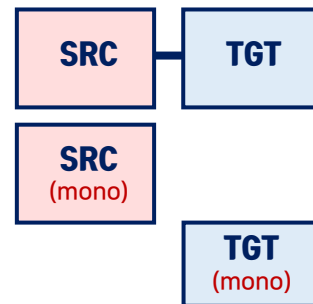
Semi-supervised Learning: parallel (+ monolingual)

Mono(TGT)

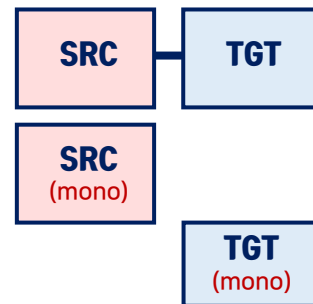


Back-Translation

Mono(SRC)

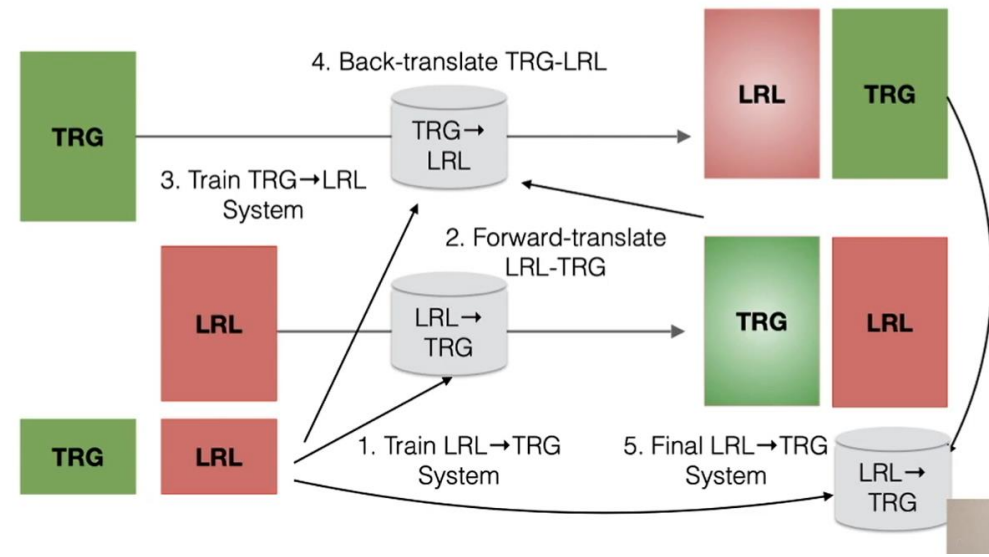


Mono(TGT, SRC)



Iterative Back-Translation

- self training과 back translation의 결합



Unsupervised Learning: only monolingual

Mono(TGT, SRC)

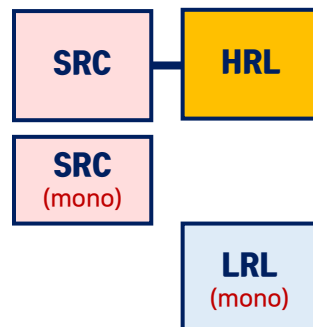
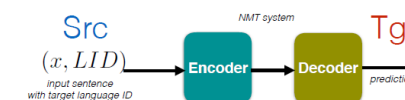
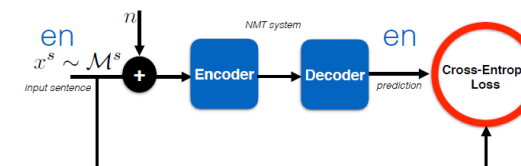
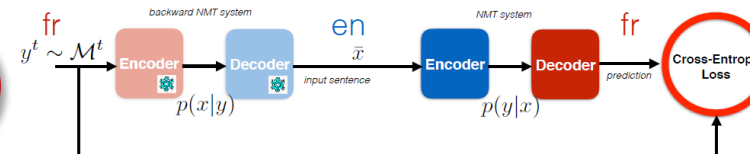
SRC
(mono)

TGT
(mono)

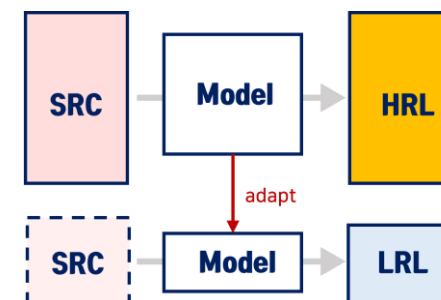
Iterative
Back-Translation

DAE
(Denoising Auto Encoding)

Shared enc/dec
(multilingual)



Transfer learning



Papers

■ **Unsupervised Machine Translation using Monolingual Corpora only**

(ICLR 2018, 817회 인용, Facebook AI Research)

DAE
(Denoising Auto Encoding)

**Iterative
Back-Translation**

Shared enc/dec
(multilingual)

■ **Machine Translation into Low-resource Language Varieties**

(ACL 2021)

Transfer learning

Published as a conference paper at ICLR 2018

UNSUPERVISED MACHINE TRANSLATION USING MONOLINGUAL CORPORA ONLY

Guillaume Lample ^{† ‡}, **Alexis Conneau** [†], **Ludovic Denoyer** [‡], **Marc'Aurelio Ranzato** [†]

[†] Facebook AI Research,

[‡] Sorbonne Universités, UPMC Univ Paris 06, LIP6 UMR 7606, CNRS

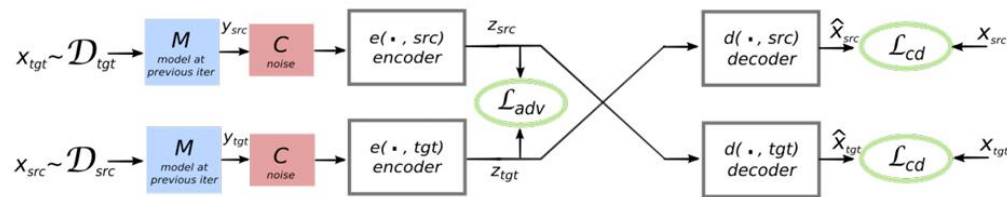
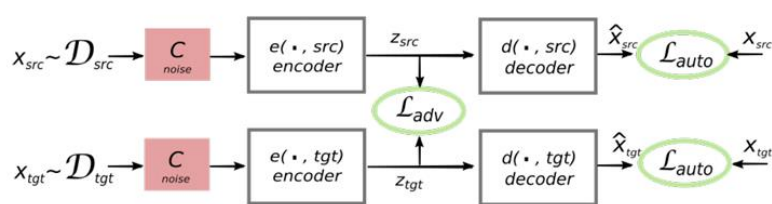
{gl, aconneau, ranzato}@fb.com, ludovic.denoyer@lip6.fr

Unsupervised Machine Translation using Monolingual Corpora only

(ICLR 2018, 817회 인용, Facebook AI Research)

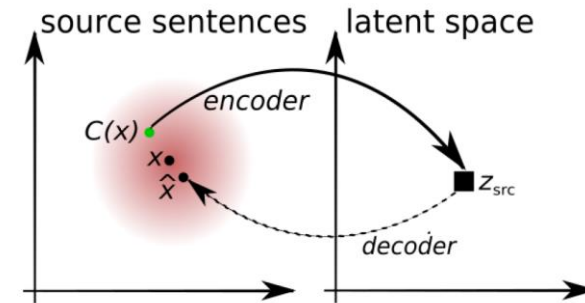
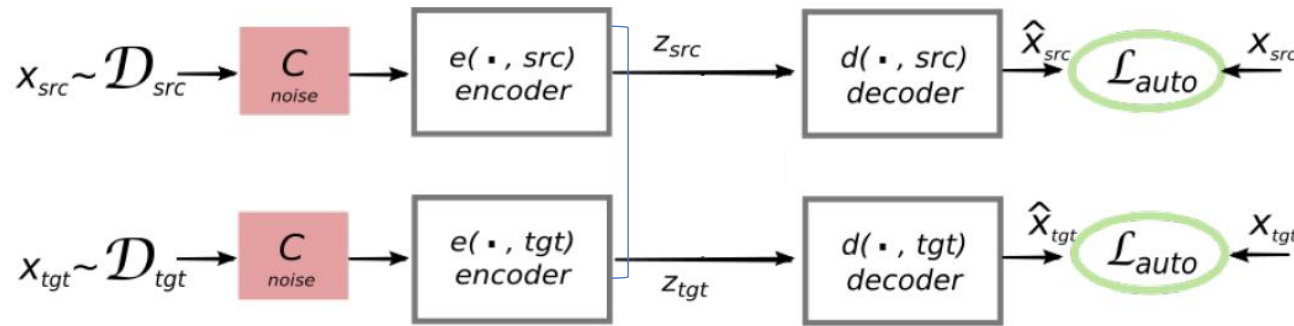
Overview

- ✓ 본 논문에서 다루는 task: monolingual data만 존재하는 unsupervised Machine Translation(En-De, De-En, En-Fr, Fr-En)
- ✓ 본 논문이 제안하는 방법
 - 완전한 unsupervised learning 세팅에서 오직 mono(each src, tgt) data만 활용하여 보조 task를 수행하며 번역 모델의 encoder/decoder 학습
 - shared encoder/decoder : src \rightarrow tgt, tgt \rightarrow src 모델의 encoder/decoder가 동일한 파라미터 공유 (임베딩 제외)
 - 3가지 보조 task: reconstruction 작업을 반복하여 언어(도메인) 간 공통의 latent space를 구축하는 것을 목표로 함
 1. **Denoising Auto-Encoding(DAE)**: input 문장에 noise 추가하여 input과 유사하게 복원하도록 학습
 2. **Cross Domain Training(CD)**: input 문장을 input domain에서 output domain으로 매핑
 3. **Adversarial Training(Adv)**: input 문장을 language와 관계 없이 같은 latent feature space로 매핑
 - Contribution: 3단계에서 산출된 loss로 학습을 반복적으로 진행하여 supervision 없이도 우수한 번역 성능 달성



Task 1. Denoising Auto-Encoding (DAE)

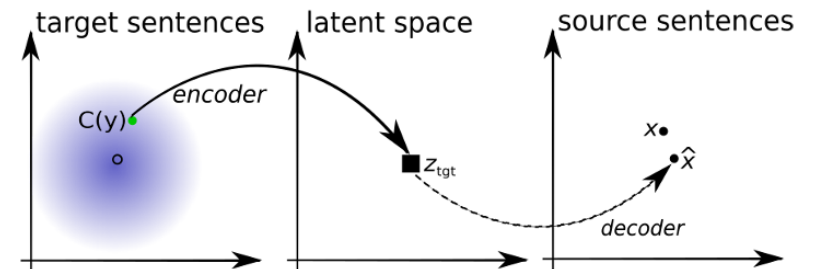
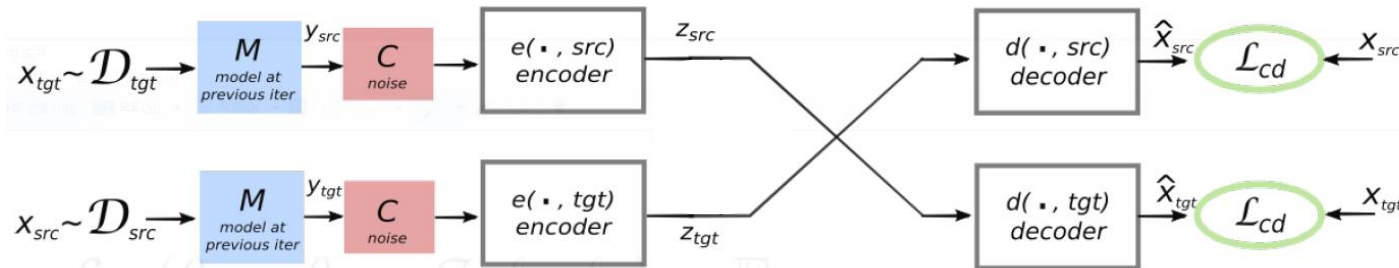
encoder/decoder의 파라미터는 공유하지만, 언어에 따라 임베딩이 다르기 때문에 구분하여 표기



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

- **학습 대상:** encoder, decoder
- **Idea:** input 문장에 noise 추가하여 encoder/decode를 거쳐 input과 유사하게 복원하도록 학습
 - noise: drop(일정 확률로 단어 삭제), swap(단어 순서 변경)
 - src와 tgt 두 언어 DAE를 거쳐 2개의 loss 값 산출
- **Loss function:** sum of token-level cross entropy(원문, 복원문)
 - 두 시퀀스 간 discrepancy(불일치) 정도를 측정하여 유사해지도록 학습

Task 2. Cross Domain Training (Back Translation)



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

- **학습 대상:** encoder, decoder

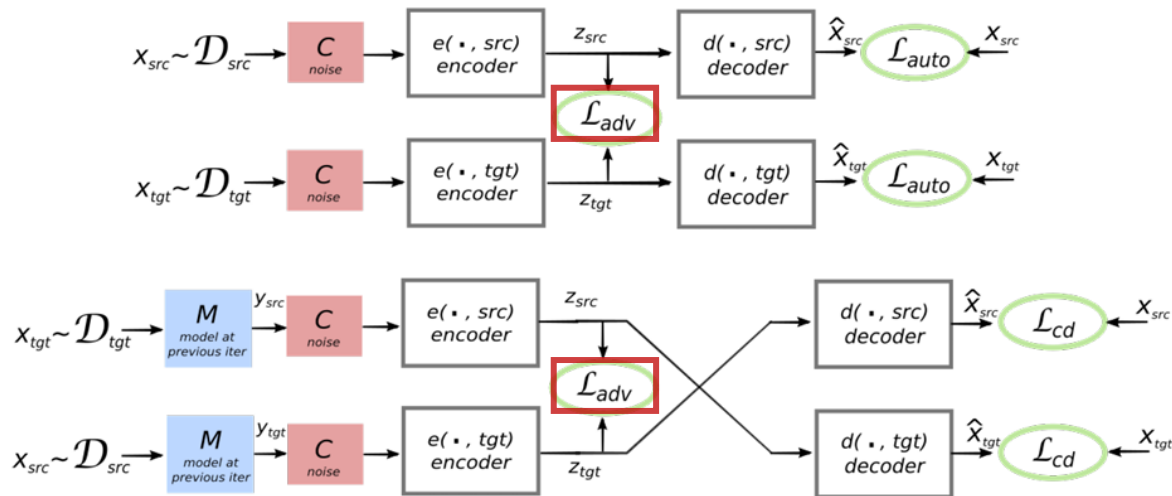
- **Idea:** input 문장(src/tgt)을 input domain(src/tgt)에서 output domain(tgt/src)으로 매핑

- src 언어 예시
- ① input 문장(src)을 현재까지 업데이트된 번역 모델 $M(\text{src} \rightarrow \text{tgt})$ 에 넣어 번역문(new tgt) 생성
 - ② 해당 번역문(new tgt)에 noise(C)를 입혀 encoder/decoder를 거쳐 input 문장(src)을 복원 (back translation)

- **Loss function:** sum of token-level cross entropy(원문, 복원문)

- 두 시퀀스 간 discrepancy(불일치) 정도를 측정하여 유사해지도록 학습, src와 tgt 2개의 loss 산출

Task 3. Adversarial Training



- Adversarial Loss (discriminator)

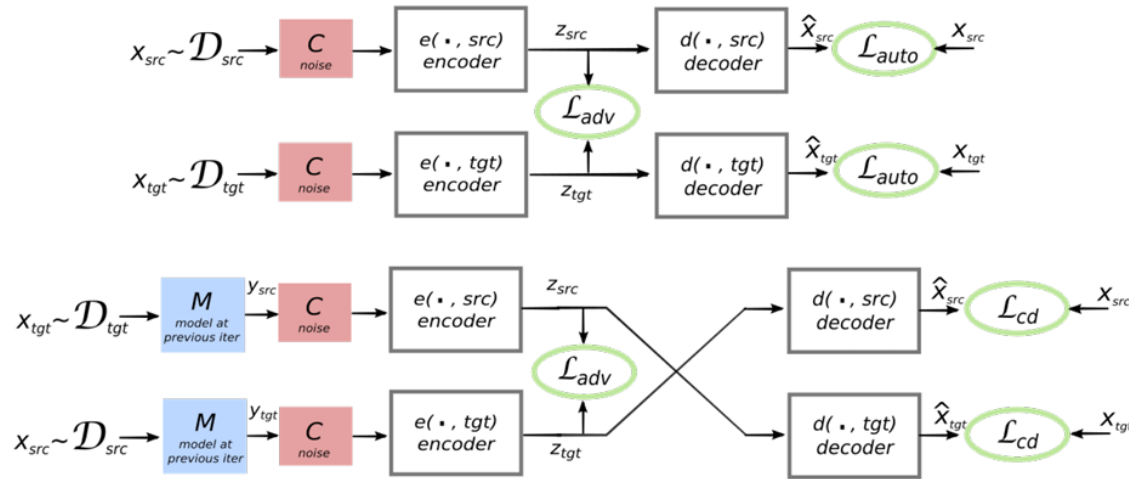
$$\mathcal{L}_D(\theta_D | \theta, \mathcal{Z}) = -\mathbb{E}_{(x_i, \ell_i)} [\log p_D(\ell_i | e(x_i, \ell_i))]$$

- Adversarial Loss (encoder)

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

- 학습 대상:** encoder, discriminator
- Idea:** input 문장의 언어(src, tgt)와 관계 없이 동일한 특징 공간(same feature space)로 매핑
 - decoder는 유사한 분포를 가지는 encoder로부터 input이 연산되어 전달될 때, 잘 작동함
 - 따라서, encoder의 결과(latent representation)를 동일한 공간으로 매핑하여 latent distribution이 같아지도록 조정
 - discriminator: 특정 latent representation의 언어가 src인지, tgt인지 판별
 - 활성함수와 dropout을 포함한 3개의 MLP 은닉층으로 구성 → 1개의 출력(sigmoid)
 - encoder: discriminator를 더 잘 속이도록 학습을 진행
- Loss function:** 이진 분류 task - BCE Loss (encoder, discriminator 2개)

Final Objective Function



- Embedding: pre-trained Fasttext
- Encoder: 3 layers Bi-LSTM
- Decoder: 3 layers LSTM
- Translation: seq2seq + attention

- Encoder/Decoder 파라미터 업데이트

$$\mathcal{L}(\theta_{enc}, \theta_{dec}, \mathcal{Z}) = \underbrace{\lambda_{auto}}_{\text{하이퍼파라미터}(=1)} [\mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, src) + \mathcal{L}_{auto}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, tgt)] + \underbrace{\lambda_{cd}}_{\text{하이퍼파라미터}(=1)} [\mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, src, tgt) + \mathcal{L}_{cd}(\theta_{enc}, \theta_{dec}, \mathcal{Z}, tgt, src)] + \underbrace{\lambda_{adv}}_{\text{하이퍼파라미터}(=1)} \mathcal{L}_{adv}(\theta_{enc}, \mathcal{Z} | \theta_D)$$

- ... 1. DAE Loss
- ... 2. Cross Domain Loss
- ... 3. Adversarial Loss (Encoder)

- Discriminator 파라미터 업데이트

$$\mathcal{L}_D(\theta_D | \theta, \mathcal{Z}) = -\mathbb{E}_{(x_i, \ell_i)} [\log p_D(\ell_i | e(x_i, \ell_i))]$$

- ... Adversarial Loss (Discriminator)

Training Algorithm

Algorithm 1 Unsupervised Training for Machine Translation

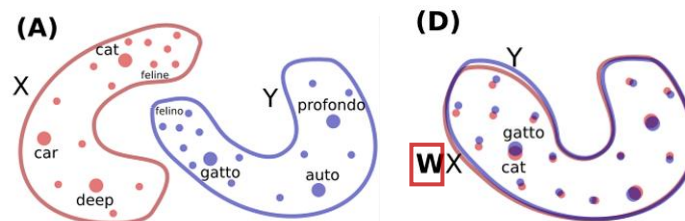
```

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

```

학습 초기 단계 - **Unsupervised Word translation**: 두 언어의 단어들 간 bilingual mapping 학습

- 가정: 두 언어의 word 임베딩 공간은 isomorphic함
 - isomorphic: 두 벡터 공간에 대해 bijective(one-to-one) linear transformation이 존재
- 따라서, X 언어의 임베딩 공간은 Y 언어의 임베딩 공간으로 선형 변환 가능 $\rightarrow WX=Y$
- Adversarial Training: 특정 임베딩 벡터의 언어가 Y인지, WX인지 예측하는 discriminator와 그를 속이는 W 학습
 - 잘 align되어 있다면 discriminator는 어느 언어인지 맞힐 수 없을 것
- 두 연구 모두 실험 언어가 알파벳 기반이기 때문에(동일한 어원) 어느 정도 유사도가 반영되어 mapping이 가능했을 것이라 추측



Experiment&Evaluation

	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

Table 2: **BLEU score on the Multi30k-Task1 and WMT datasets** using greedy decoding.

데이터셋: 영어-불어 / 영어-독일어

- *WMT'14 English-French, WMT'16 English-German, Multi30k*
- 가장 높은 성능을 기록한 supervised 모델(Full dataset)
- 반복 횟수 늘어날 수록 점차 BLEU 점수 상승
- 1회의 iteration 만으로도 word-by-word (WBW)나 word reordering(WBW 이후 LSTM 기반 LM으로 단어 순서 조정)보다 높은 성능 기록

Machine Translation into Low-resource Language Varieties

Sachin Kumar♣ Antonios Anastasopoulos◇ Shuly Wintner♡ Yulia Tsvetkov♠

♣Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA

◇Department of Computer Science, George Mason University, Fairfax, VA, USA

♡Department of Computer Science, University of Haifa, Haifa, Israel

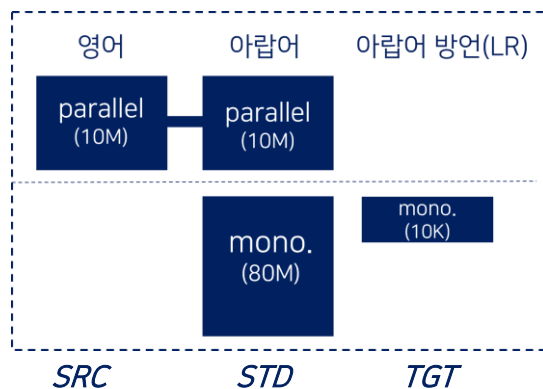
♠Paul G. Allen School of Computer Science & Engineering, University of Washington

sachink@cs.cmu.edu, antonis@gmu.edu, shuly@cs.haifa.ac.il, yuliats@cs.washington.edu

 **Machine Translation into Low-resource Language Varieties** (ACL 2021, 3회 인용)

Overview

- ✓ 본 논문이 다루는 **task**: Unsupervised Machine Translation (영어->방언)
- ✓ 본 논문이 제안하는 방법: Transfer learning 프레임워크 (high resource 정보 -> low resource 정보)
 1. high resource(STD, 표준어) **embedding** -> low resource(TGT, 방언) **embedding**
 - high resource 언어의 knowledge를 transfer
 - word embedding adaptation: softmax 기반의 discrete output이 아닌 continuous word vector의 유사도를 기반으로 단어 예측
 2. high resource SRC(영어)->STD(표준어) 번역 모델 -> **low resource SRC(영어)-> TGT(방언) 번역 모델**
 - SRC(영어)-TGT(아랍어 방언) parallel data 없이 mono(TGT, 방언) data만 활용하여 빠르고 효과적인 adaptation 도모

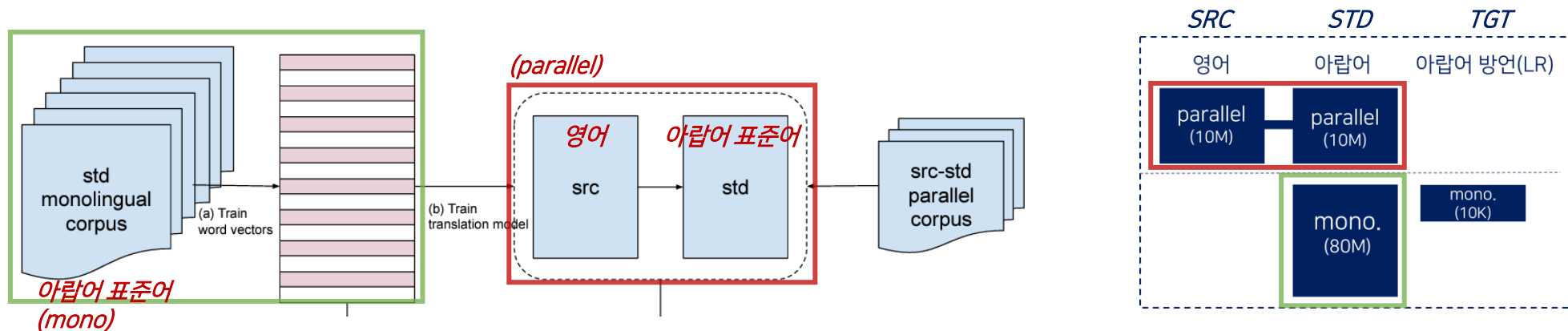


Dataset

- parallel: (src, std) pair -> *high-resource model 학습에 활용*
 - ① (영어, 아랍어) 10M / ② (영어, 러시아어) 630K / ③ (영어, 노르웨이어) 10M
- monolingual: (std), (tgt) -> *fine-tuning에 활용*
 - ① (아랍어) 80M, (아랍어 방언) 10K
 - ② (러시아어) 80M, (우크라이나어, 벨라루스어) 10K, 100K, 1M
 - ③ (노르웨이어) 26M, (노르웨이어 방언) 310K

Training Process

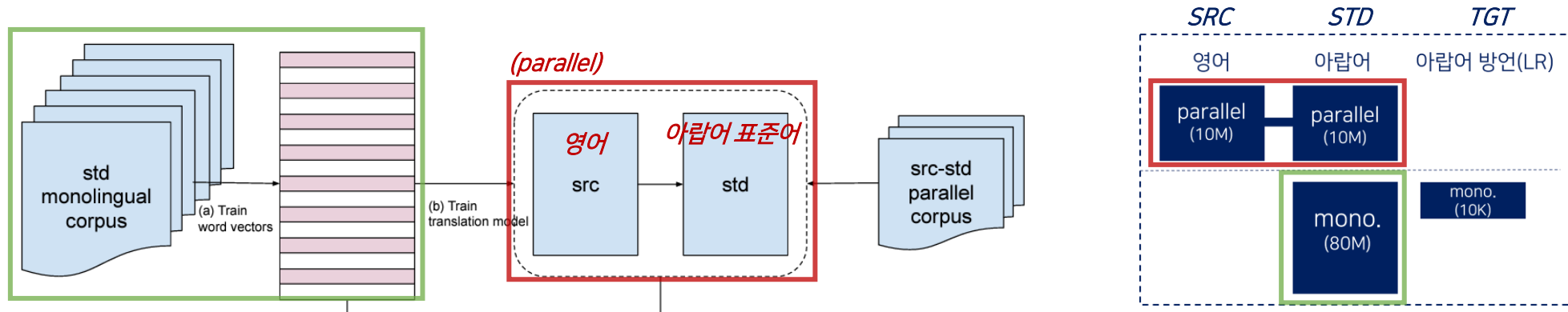
1. SRC(영어)->STD(아랍어 표준어) 모델 학습: Supervised learning w/ parallel data



- ✓ 주어진 영어(SRC)-아랍어(STD) parallel data로 번역 모델(SRC->STD) 학습
 - Transformer based encoder/decoder
 - 디코더 입력(STD): d차원의 pre-trained word vector (unit length)
 - ✓ parallel이 아닌 monolingual 표준어 data로 Fasttext 임베딩 학습
 - 디코더 출력(STD): d차원의 continuous word vector
 - softmax로 prob. 단어 예측하지 않는 이유: transfer learning 시에 std(표준어)와 tgt(방언)간 vocab mismatch 발생
 - 예측 벡터와 가장 가까운(코사인 유사도 기준) 벡터를 pre-trained embedding table에서 추출
 - Loss function: vMF(von Mises-Fisher) Loss

Training Process

1. SRC(영어)->STD(아랍어 표준어) 모델 학습: Supervised learning w/ parallel data



- **vMF(von Mises-Fisher) Loss** (ICLR 2019, [link](#)): 예측 벡터와 pre-trained 벡터 간의 probabilistic variant of cosine distance

At each generation step, the decoder of our model produces a continuous vector $\hat{\mathbf{e}} \in \mathbb{R}^m$. The output word is then predicted by searching for the nearest neighbor of $\hat{\mathbf{e}}$ in the embedding space:

$$w_{\text{predicted}} = \underset{w}{\operatorname{argmin}} \{d(\hat{\mathbf{e}}, \mathbf{e}(w)) | w \in \mathcal{V}\}$$

Normalizing Constant

$$C_m(\kappa) = \frac{\kappa^{m/2-1}}{(2\pi)^{m/2} I_{m/2-1}(\kappa)},$$

$$\text{vMF}(\mathbf{e}(w); \hat{\mathbf{e}}) = C_m(\|\hat{\mathbf{e}}\|) e^{\hat{\mathbf{e}}^T \mathbf{e}(w)}$$

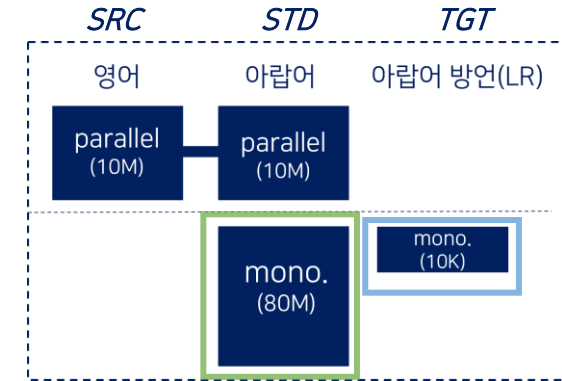
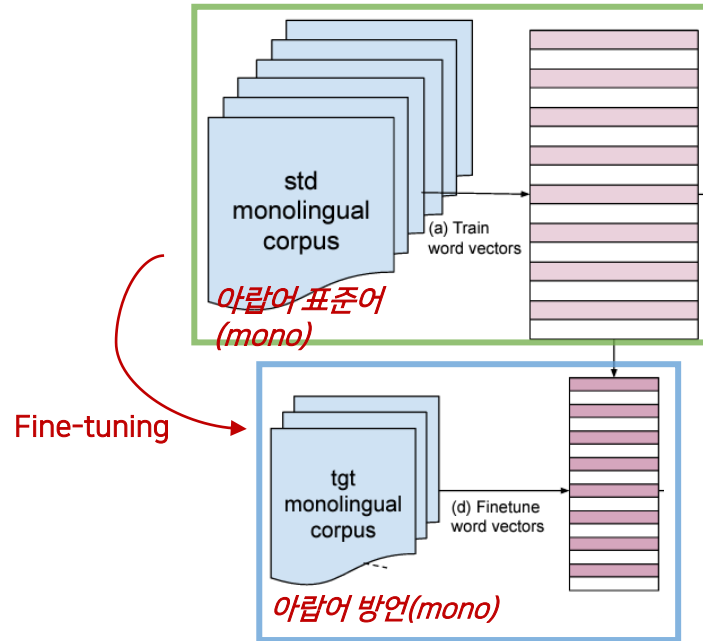
von Mises Fisher 분포의 확률 밀도 함수

$$\text{NLLvMF}(\hat{\mathbf{e}}; \mathbf{e}(w)) = -\log(C_m(\|\hat{\mathbf{e}}\|)) - \hat{\mathbf{e}}^T \mathbf{e}(w)$$

Neg. Log Likelihood Loss

Training Process

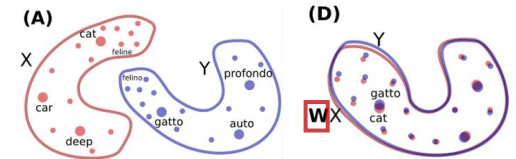
2. STD(표준어) 임베딩을 Fine-tuning하여 TGT(방언) embedding vectors 학습



- SRC(영어)->STD(아랍어) 모델을 SRC->TGT(방언)으로 fine-tuning하기 위해서는 TGT(방언) monolingual corpus의 임베딩 필요
- TGT(아랍어 방언)과 유사한 STD(아랍어 표준어)의 임베딩 활용
- STD의 정보를 포함하는 tgt의 임베딩 벡터 학습

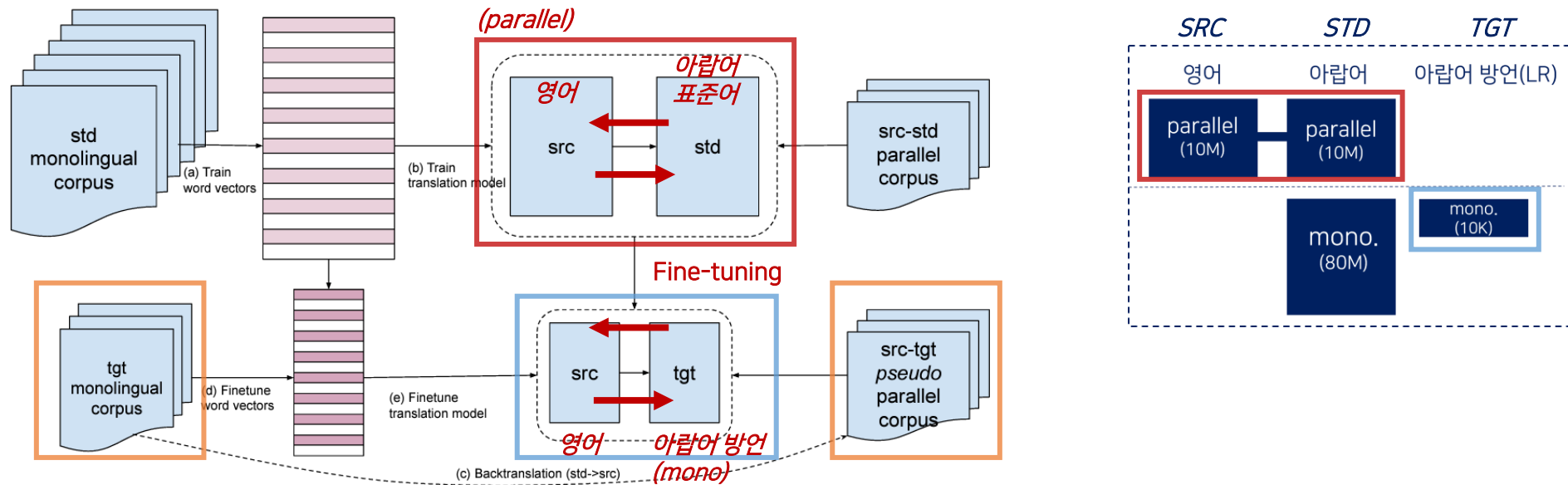
과정

- ① 전처리: 표준어 STD(BPE), 방언 TGT(joint BPE 모델 학습 with STD+TGT)
- ② STD의 Fasttext 임베딩으로 초기화 및 fine-tuning
- ③ TGT 임베딩을 STD 임베딩 공간으로 projection (supervised embedding alignment-위 그림)



Training Process

3. SRC(영어)->STD(아랍어) 모델을 Fine-tuning하여 SRC(영어)->TGT(아랍어 방언) 번역 모델 학습



- ① STD(아랍어 표준어)->SRC(영어) back translation 모델 학습 (large parallel data 활용) ✓
- ② TGT(아랍어 방언)을 SRC(영어)로 역번역하여 pseudo parallel 데이터 생성 ✓
 - 표준어와 방언의 유사도에도 불구하고 noisy하나, 성능 향상
- ③ 생성한 pseudo parallel set으로 SRC(영어) -> STD(아랍어 표준어) 모델을 fine-tuning하여 SRC(영어) -> TGT(아랍어 방언) 번역 모델 학습

Experiment&Evaluation

러시아어-> 우크라이나어, 벨라루스어

노르웨이어 -> 노르웨이어 방언

아랍어 -> 아랍어 방언

Size of TGT corpus	UK			BE			NN 300K	Arabic Varieties (10K)			
	10K	100K	1M	10K	100K	1M		Doha	Beirut	Rabat	Tunis
SUP(SRC→STD)	1.7	1.7	1.7	1.5	1.5	1.5	11.3	3.7	1.8	2.0	1.3
UNSUP(SRC→TGT)	0.3	0.6	0.9	0.4	0.6	1.4	2.7	0.2	0.1	0.1	0.1
PIVOT	1.5	8.6	14.9	1.15	3.9	8.0	11.9	1.8	2.1	1.7	1.1
SOFTMAX	1.9	12.7	15.4	1.5	4.5	7.9	14.4	14.5	7.4	4.9	3.9
LANGVARMT	6.1	13.5	15.3	2.3	8.8	9.8	16.6	20.1	8.1	7.4	4.6

sup(src->std) + unsup(std->tgt)

- ✓ 언어 간 유사도 높을 수록 (NO-NN) easy adaptation of model and embedding -> BLEU 높음
- ✓ TGT가 Low resource일 수록 overfitting되는 경향이 있음
- ✓ 유사하지 않은 언어(SRC, TGT)를 Unsupervised(paper1, MLM으로 초기화)로 학습하면 성능 낮음
- ✓ softmax를 기반으로 한 discrete 예측 모델보다 성능 높음

감사합니다