



## Exploit Duality



# Exploit Duality

## What is Duality?

Duality란 무엇일까요? 우리가 보통 기계학습을 통해 학습하는 것은 어떤 도메인의 데이터  $X$ 를 받아서, 다른 도메인의 데이터  $Y$ 로 맵핑(mapping)해주는 함수를 근사(approximation)하는 것이라 할 수 있습니다. 따라서 대부분의 기계학습에 사용되는 데이터셋은 두 도메인 사이의 데이터로 구성되어있기 마련입니다.

Task( $D_1 \rightarrow D_2$ )	Domain 1	Domain 2	Task( $D_1 \leftarrow D_2$ )
기계번역	source 언어 문장	target 언어 문장	기계번역
음성인식	음성 신호	텍스트(transcript)	음성합성
이미지 분류	이미지	class	이미지 합성
요약	본문(content) 텍스트	제목(title) 텍스트	본문 생성

위와 같이 두 도메인 사이의 데이터의 관계를 배우는 방향에 따라서 음성인식이 되기도 하고, 음성합성이 되기도 합니다. 이러한 두 도메인 사이의 관계를 duality라고 우리는 정의 합니다. 대부분의 기계학습 문제들은 이와 같이 duality를 가지고 있는데, 특히 기계번역은 각 도메인의 데이터 간에 정보량의 차이가 거의 없는 것이 가장 큰 특징이자 장점 입니다. 따라서 duality를 가장 적극적으로 활용할 수 있습니다.

## CycleGAN

먼저 좀 더 이해하기 쉬운 duality의 활용 예로, 컴퓨터 비전(Computer Vision)쪽 논문[Zhu et al.2017]을 설명 해 볼까 합니다. Cycle GAN은 아래와 같이 unpaired image set이 여러개 있을 때,  $Set X$ 의 이미지를  $Set Y$ 의 이미지로 합성/변환 시켜주는 방법 입니다. 사진을 전체 구조는 유지하되 모네의 그림풍으로 바꾸어 주기도 하고, 말과 얼룩말을 서로 바꾸어 주기도 합니다. 겨울 풍경을 여름 풍경으로 바꾸어주기도 합니다.

아래에 이 방법을 도식화 하여 나타냈습니다.  $Set X$ 와  $Set Y$  모두 각각 Generator( $G, F$ )와 Discriminator( $D_X, D_Y$ )를 가지고 있어서, min/max 게임을 수행합니다.

$G$ 는  $x$ 를 입력으로 받아  $\hat{y}$ 으로 변환 해 냅니다. 그리고  $D_Y$ 는  $\hat{y}$  또는  $y$ 를 입력으로 받아 합성 유무(Real/Fake)를 판단 합니다. 마찬가지로  $F$ 는  $y$ 를 입력으로 받아  $\hat{x}$ 으로 변환 합니다. 이후에  $D_X$ 는  $\hat{x}$  또는  $x$ 를 입력으로 받아 합성 유부를 판단 합니다.

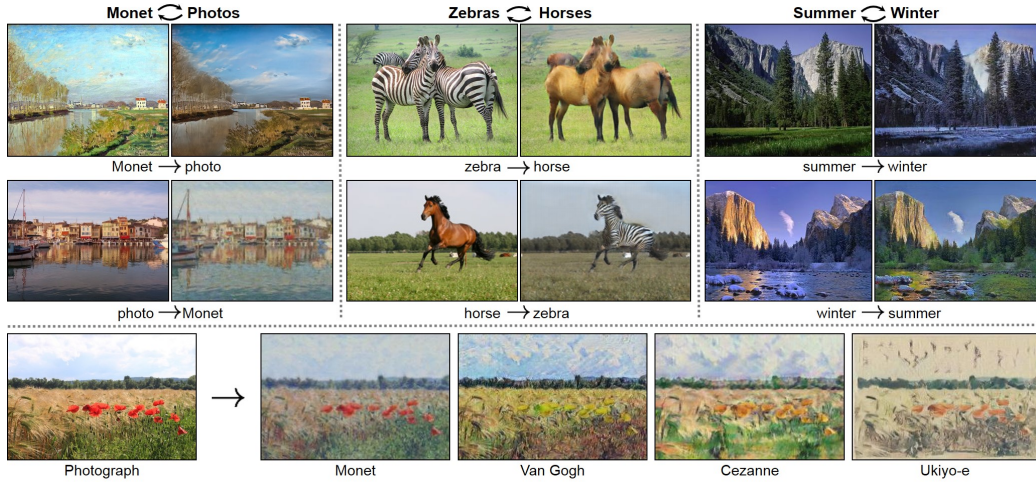


Figure 1: Cycle GAN – image from web

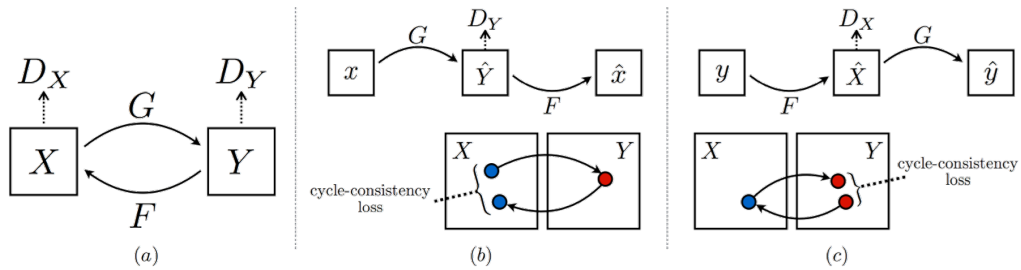


Figure 3: (a) Our model contains two mapping functions  $G : X \rightarrow Y$  and  $F : Y \rightarrow X$ , and associated adversarial discriminators  $D_Y$  and  $D_X$ .  $D_Y$  encourages  $G$  to translate  $X$  into outputs indistinguishable from domain  $Y$ , and vice versa for  $D_X$  and  $F$ . To further regularize the mappings, we introduce two *cycle consistency losses* that capture the intuition that if we translate from one domain to the other and back again we should arrive at where we started: (b) forward cycle-consistency loss:  $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ , and (c) backward cycle-consistency loss:  $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$

Figure 2:

이 방식의 핵심은  $\hat{x}$ 나  $\hat{y}$ 를 합성 할 때에 기존의 Set  $X, Y$ 에 속하는 것 처럼 만들어내야 한다는 것 입니다. 이것을 기계번역에 적용 시켜 보면 어떻게 될까요?

## Dual Supervised Learning

이번에 소개할 방법은 Dual Supervised Learning (DSL) [Xia et al.2017] 입니다. 이 방법은 기존의 Teacher Forcing의 문제로 생기는 어려움을 강화학습을 사용하지 않고, Daulity로부터 regularization term을 이끌어내어 해결하였습니다.

베이즈 정리(Bayes Theorem)에 따라서 우리는 아래의 수식이 언제나 성립함을 알고 있습니다.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

$$P(Y|X)P(X) = P(X|Y)P(Y)$$

따라서 위의 수식을 따라서, 우리의 데이터셋을 통해 훈련한 모델들은 아래와 같은 수식을 만족해야 합니다.

$$P(x)P(y|x; \theta_{x \rightarrow y}) = P(y)P(x|y; \theta_{y \rightarrow x})$$

이 전제를 우리의 번역 훈련을 위한 목표에 적용하면 다음과 같습니다.

$$\begin{aligned} objective1 : \min_{\theta_{x \rightarrow y}} \frac{1}{n} \sum_{i=1}^n \ell_1(f(x_i; \theta_{x \rightarrow y}), y_i), \\ objective2 : \min_{\theta_{y \rightarrow x}} \frac{1}{n} \sum_{i=1}^n \ell_1(g(y_i; \theta_{y \rightarrow x}), x_i), \\ s.t. P(x)P(y|x; \theta_{x \rightarrow y}) = P(y)P(x|y; \theta_{y \rightarrow x}), \forall x, y. \end{aligned}$$

위의 수식을 해석하면, 목표(objective1)은 베이즈 정리에 따른 제약조건을 만족함과 동시에,  $\ell_1$ 을 최소화(minimize) 하도록 해야 합니다.  $\ell_1$ 은 번역함수  $f$ 에 입력  $x_i$ 를 넣어 나온 반환값과  $y_i$  사이의 손실(loss)를 의미 합니다. 마찬가지로,  $\ell_2$ 도 번역함수  $g$ 에 대해 같은 작업을 수행하고 최소화하여 목표(objective2)를 만족해야 합니다.

$$\mathcal{L}_{duality} = ((\log \hat{P}(x) + \log P(y|x; \theta_{x \rightarrow y})) - (\log \hat{P}(y) + \log P(x|y; \theta_{y \rightarrow x})))^2$$

그러므로 우리는  $\mathcal{L}_{duality}$ 와 같이 베이지 정리에 따른 제약조건의 양 변의 값의 차이를 최소화(minimize)하도록 하는 MSE 손실함수(loss function)을 만들 수 있습니다. 위의 수식에서 우리가 동시에 훈련시키는 신경망 네트워크 파라미터를 통해  $\log P(y|x; \theta_{x \rightarrow y})$ 와  $\log P(x|y; \theta_{y \rightarrow x})$ 를 구하고, 단방향(monolingual) corpus를 통해 별도로 이미 훈련시켜 놓은 언어모델을 통해  $\log \hat{P}(x)$ 와  $\log \hat{P}(y)$ 를 근사(approximation)할 수 있습니다.

이 추가적인 제약조건의 손실함수를 기존의 목적함수(objective function)에 추가하여 동시에 minimize 하도록 하면, 아래와 같이 표현 할 수 있습니다.

$$\begin{aligned} \theta_{x \rightarrow y} &\leftarrow \theta_{x \rightarrow y} - \gamma \nabla_{\theta_{x \rightarrow y}} \frac{1}{n} \sum_{j=1}^m [\ell_1(f(x_i; \theta_{x \rightarrow y}), y_i) + \lambda_{x \rightarrow y} \mathcal{L}_{duality}] \\ \theta_{y \rightarrow x} &\leftarrow \theta_{y \rightarrow x} - \gamma \nabla_{\theta_{y \rightarrow x}} \frac{1}{n} \sum_{j=1}^m [\ell_2(g(y_i; \theta_{y \rightarrow x}), x_i) + \lambda_{y \rightarrow x} \mathcal{L}_{duality}] \end{aligned}$$

여기서  $\lambda$ 는 Lagrange multipliers로써, 고정된 값의 hyper-parameter 입니다. 실험 결과  $\lambda = 0.01$  일 때, 가장 좋은 성능을 나타낼 수 있었습니다.

**Table 2.** Summary of some existing En→Fr translations

Model	Brief description	BLEU
NMT[1]	<i>standard NMT</i>	33.08
MRT[2]	<i>Direct optimizing BLEU</i>	34.23
DSL	<i>Refer to Algorithm 1</i>	<b>34.84</b>
[1] (Jean et al., 2015); [2] (Shen et al., 2016)		

Figure 3:

위의 테이블과 같이 기존의 Teacher Forcing 아래의 cross entropy 방식([1]번)과 Minimum Risk Training(MRT) 방식([2]번) 보다 더 높은 성능을 보입니다.

이 방법은 강화학습과 같이 비효율적이고 훈련이 까다로운 방식을 벗어나서 regularization term을 추가하여 강화학습을 상회하는 성능을 얻어낸 것이 주목할 점이라고 할 수 있습니다.

## Dual Unsupervised Learning

### Dual Learning for Machine Translation

공교롭게도 CycleGAN과 비슷한 시기에 나온 논문[Xia et al. 2016]이 있습니다. NLP의 특성상 CycleGAN처럼 직접적으로 gradient를 전달해 줄 수는 없었지만 기본적으로는 아주 비슷한 개념입니다. 짝이 없는 단방향(monolingual) corpus를 이용하여 성능을 극대화 하고자 하였습니다.

즉, monolingual sentence( $s$ )에 대해서 번역을 하고 그 문장( $s_{mid}$ )을 사용하여 복원을 하였을 때( $\hat{s}$ ) 원래의 처음 문장으로 돌아올 수 있도록(처음 문장과 차이 최소화 하도록) 훈련하는 것입니다. 이때, 번역된 문장  $s_{mid}$ 는 자연스러운 해당 언어의 문장이 되었는지도 중요한 지표가 됩니다.

위에서 설명한 알고리즘을 따라가 보겠습니다. 이 방법에서는 *Set X*, *Set Y* 대신에 *Language A*, *Language B*로 표기하고 있습니다.  $G_{A \rightarrow B}$ 의 파라미터  $\theta_{AB}$ 와  $F_{B \rightarrow A}$ 의 파라미터  $\theta_{BA}$ 가 등장합니다. 이  $G_{A \rightarrow B}$ ,  $F_{B \rightarrow A}$ 는 모두 parallel corpus에 의해서 pre-training이 되어 있는 상태 입니다. 즉, 기본적인 저성능의 번역기 수준이라고 가정합니다.

우리는 기존의 policy gradient와 마찬가지로 아래와 같은 파라미터 업데이트를 수행해야 합니다.

$$\begin{aligned}\theta_{AB} &\leftarrow \theta_{AB} + \gamma \nabla_{\theta_{AB}} \hat{\mathbb{E}}[r] \\ \theta_{BA} &\leftarrow \theta_{BA} + \gamma \nabla_{\theta_{BA}} \hat{\mathbb{E}}[r]\end{aligned}$$

$\hat{\mathbb{E}}[r]$ 을 각각의 파라미터에 대해서 미분 해 준 값을 더해 주는 것을 볼 수 있습니다. 이 reward의 기대값은 아래와 같이 구할 수 있습니다.

$$\begin{aligned}r &= \alpha r_{AB} + (1 - \alpha) r_{BA} \\ r_{AB} &= LM_B(s_{mid}) \\ r_{BA} &= \log P(s | s_{mid}; \theta_{BA})\end{aligned}$$

---

**Algorithm 1** The dual-learning algorithm

---

- 1: **Input:** Monolingual corpora  $D_A$  and  $D_B$ , initial translation models  $\Theta_{AB}$  and  $\Theta_{BA}$ , language models  $LM_A$  and  $LM_B$ , hyper-parameter  $\alpha$ , beam search size  $K$ , learning rates  $\gamma_{1,t}, \gamma_{2,t}$ .
  - 2: **repeat**
  - 3:    $t = t + 1$ .
  - 4:   Sample sentence  $s_A$  and  $s_B$  from  $D_A$  and  $D_B$  respectively.
  - 5:   Set  $s = s_A$ .  $\triangleright$  *Model update for the game beginning from A.*
  - 6:   Generate  $K$  sentences  $s_{mid,1}, \dots, s_{mid,K}$  using beam search according to translation model  $P(\cdot|s; \Theta_{AB})$ .
  - 7:   **for**  $k = 1, \dots, K$  **do**
  - 8:     Set the language-model reward for the  $k$ th sampled sentence as  $r_{1,k} = LM_B(s_{mid,k})$ .
  - 9:     Set the communication reward for the  $k$ th sampled sentence as  $r_{2,k} = \log P(s|s_{mid,k}; \Theta_{BA})$ .
  - 10:    Set the total reward of the  $k$ th sample as  $r_k = \alpha r_{1,k} + (1 - \alpha) r_{2,k}$ .
  - 11:   **end for**
  - 12:   Compute the stochastic gradient of  $\Theta_{AB}$ :
 
$$\nabla_{\Theta_{AB}} \hat{E}[r] = \frac{1}{K} \sum_{k=1}^K [r_k \nabla_{\Theta_{AB}} \log P(s_{mid,k}|s; \Theta_{AB})].$$
  - 13:   Compute the stochastic gradient of  $\Theta_{BA}$ :
 
$$\nabla_{\Theta_{BA}} \hat{E}[r] = \frac{1}{K} \sum_{k=1}^K [(1 - \alpha) \nabla_{\Theta_{BA}} \log P(s|s_{mid,k}; \Theta_{BA})].$$
  - 14:   Model updates:
 
$$\Theta_{AB} \leftarrow \Theta_{AB} + \gamma_{1,t} \nabla_{\Theta_{AB}} \hat{E}[r], \Theta_{BA} \leftarrow \Theta_{BA} + \gamma_{2,t} \nabla_{\Theta_{BA}} \hat{E}[r].$$
  - 15:   Set  $s = s_B$ .  $\triangleright$  *Model update for the game beginning from B.*
  - 16:   Go through line 6 to line 14 symmetrically.
  - 17: **until** convergence
- 

Figure 4:



위와 같이  $k$ 개의 sampling한 문장에 대해서 각기 방향에 대한 reward를 각각 구한 후, 이를 선형 결합(linear combination)을 취해줍니다. 이때,  $s_{mid}$ 는 sampling한 문장을 의미하고,  $LM_B$ 를 사용하여 해당 문장이 *language B*의 집합에 잘 어울리는지를 따져 reward로 리턴합니다. 여기서 기존의 cross entropy를 사용할 수 없는 이유는 monolingual sentence이기 때문에 번역을 하더라도 정답을 알 수 없기 때문입니다. 또한 우리는 다수의 단방향(monolingual) corpus를 갖고 있기 때문에,  $LM$ 은 쉽게 만들어낼 수 있습니다.

$$\begin{aligned}\nabla_{\theta_{AB}} \hat{\mathbb{E}}[r] &= \frac{1}{K} \sum_{k=1}^K [r_k \nabla_{\theta_{AB}} \log P(s_{mid,k}|s; \theta_{AB})] \\ \nabla_{\theta_{BA}} \hat{\mathbb{E}}[r] &= \frac{1}{K} \sum_{k=1}^K [(1 - \alpha) \nabla_{\theta_{BA}} \log P(s|s_{mid,k}; \theta_{BA})]\end{aligned}$$

이렇게 얻어진  $\mathbb{E}[r]$ 를 각 파라미터에 대해서 미분하게 되면 위와 같은 수식을 얻을 수 있고, 상기 서술한 파라미터 업데이트 수식에 대입하면 됩니다. 비슷한 방식으로  $B \rightarrow A$ 를 구할 수 있습니다.

Table 1: Translation results of En $\leftrightarrow$ Fr task. The results of the experiments using all the parallel data for training are provided in the first two columns (marked by “Large”), and the results using 10% parallel data for training are in the last two columns (marked by “Small”).

	En $\rightarrow$ Fr (Large)	Fr $\rightarrow$ En (Large)	En $\rightarrow$ Fr (Small)	Fr $\rightarrow$ En (Small)
NMT	29.92	27.49	25.32	22.27
pseudo-NMT	30.40	27.66	25.63	23.24
dual-NMT	<b>32.06</b>	<b>29.78</b>	<b>28.73</b>	<b>27.50</b>

Figure 5:

위의 테이블은 이 방법의 성능을 비교한 결과 입니다. Pseudo-NMT는 이전 챕터에서 설명하였던 back-translation을 의미합니다. 그리고 그 방식보다 더 좋은 성능을 기록한 것을 볼 수 있습니다.

또한, 위 그래프에서 문장의 길이와 상관 없이 모든 구간에서 baseline NMT를 성능으로 압도하고 있는 것을 알 수 있습니다. 다만, 병렬(parallel) corpus의 양이 커질수록 단방향(monolingual) corpus에 의한 성능 향상의 폭이 줄어드는 것을 확인할 수 있습니다.

이 방법은 강화학습과 Duality를 접목하여 적은 양의 병렬(parallel) corpus와 다수의 단방향(monolingual) corpus를 활용하여 번역기의 성능을 효과적으로 끌어올리는

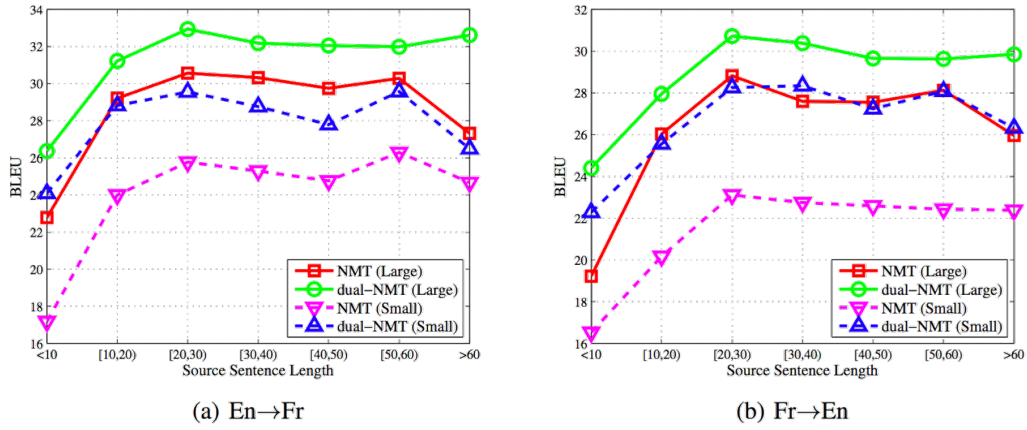


Figure 1: BLEU scores w.r.t lengths of source sentences

Figure 6:

방법을 제시하였다는 점에서 주목할 만 합니다.

## Dual Transfer Learning for NMT with Marginal Distribution Regularization

Dual Supervised Learning (DSL)은 베이지 정리에 따른 수식을 제약조건으로 사용하였다면, 이 방법[Wang et al.2017]은 Marginal 분포(distribution)의 성질을 이용하여 제약조건을 만듭니다.

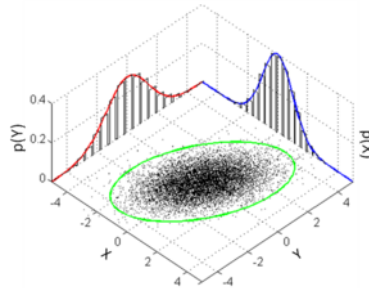


Figure 7: Marginal Distribution from Wikipedia

$$P(y) = \sum_{x \in \mathcal{X}} P(x, y) = \sum_{x \in \mathcal{X}} P(y|x)P(x)$$

Marginal 분포는 결합확률분포(joint distribution)를 어떤 한 variable에 대해서 합 또는 적분 한 것을 이릅니다. 이것을 조건부확률로 나타낼 수 있고, 여기서 한발 더 나아가 기대값 표현으로 바꿀 수 있습니다. 그리고 이를  $K$ 번 샘플링 하도록 하여 Monte Carlo로 근사 표현 할 수 있습니다.

$$\begin{aligned} P(y) &= \sum_{x \in \mathcal{X}} P(y|x; \theta)P(x) = \mathbb{E}_{x \sim P(x)} P(y|x; \theta) \\ &\approx \frac{1}{K} \sum_{i=1}^K P(y|x^i; \theta), \quad x^i \sim P(x) \end{aligned}$$

이제 위의 수식을 기계번역에 적용해 보도록 하겠습니다. 우리에게 아래와 같이  $N$ 개의 source 문장  $x$ , target 문장  $y$ 으로 이루어진 양방향 corpus  $\mathcal{B}$ ,  $S$ 개의 target 문장  $y$ 로만 이루어진 단방향 corpus  $\mathcal{M}$ 이 있다고 가정 해 보겠습니다.

$$\begin{aligned} \mathcal{B} &= \{(x^n, y^n)\}_{n=1}^N \\ \mathcal{M} &= \{y^s\}_{s=1}^S \end{aligned}$$

그럼 우리는 아래의 목적함수(objective function)을 최대화(maximize)하는 동시에 marginal 분포에 따른 제약조건 또한 만족시켜야 합니다.

$$\begin{aligned} \text{Objective} &: \sum_{n=1}^N \log P(y^n|x^n; \theta), \\ \text{s.t. } P(y) &= \mathbb{E}_{x \sim P(x)} P(y|x; \theta), \forall y \in \mathcal{M}. \end{aligned}$$

위의 수식을 DSL과 마찬가지로 Lagrange multipliers와 함께 기존의 손실함수(loss function)에 추가하여 주기 위하여  $S(\theta)$ 와 같이 표현합니다.

$$\mathcal{S}(\theta) = [\log \hat{P}(y) - \log \mathbb{E}_{x \sim \hat{P}(x)} P(y|x; \theta)]^2$$

$$\mathcal{L}(\theta) = - \sum_{n=1}^N \log P(y^n|x^n; \theta) + \lambda \sum_{s=1}^S [\log \hat{P}(y) - \log \mathbb{E}_{x \sim \hat{P}(x)} P(y|x; \theta)]^2$$

이때, DSL과 유사하게  $\hat{P}(x)$ 와  $\hat{P}(y)$ 가 등장합니다.  $\hat{P}(y)$ 는 단방향(monolingual) corpus로 만든 언어모델(language model)을 통해 확률값을 구합니다. 위의 수식에 따르면  $\hat{P}(x)$ 를 통해 source 문장  $x$ 를 샘플링(sampling)하여 네트워크  $\theta$ 를 통과시켜  $P(y|x; \theta)$ 를 구해야겠지만, 아래와 같이 좀 더 다른 방법으로 접근합니다.

$$\begin{aligned}
P(y) &= \mathbb{E}_{x \sim \hat{P}(x)} P(y|x; \theta) = \sum_{x \in \mathcal{X}} P(y|x; \theta) \hat{P}(x) \\
&= \sum_{x \in \mathcal{X}} \frac{P(y|x; \theta) \hat{P}(x)}{P(x|y)} P(x|y) \\
&= \mathbb{E}_{x \sim P(x|y)} \frac{P(y|x; \theta) \hat{P}(x)}{P(x|y)} \\
&= \frac{1}{K} \sum_{i=1}^K \frac{P(y|x_i; \theta) \hat{P}(x_i)}{P(x_i|y)}, x_i \sim P(x|y)
\end{aligned}$$

위와 같이 target 문장  $y$ 를 반대 방향 번역기( $y \rightarrow x$ )에 넣어  $K$ 개의 source 문장  $x$ 를 샘플링(sampling)하여  $P(y)$ 를 구합니다. 이 과정을 다시 하나의 손실함수(loss function)으로 표현하면 아래와 같습니다.

$$\mathcal{L}(\theta) \approx - \sum_{n=1}^N \log P(y^n|x^n; \theta) + \lambda \sum_{s=1}^S [\log \hat{P}(y^s) - \log \frac{1}{K} \sum_{i=1}^K \frac{\hat{P}(x_i^s) P(y^s|x_i^s; \theta)}{P(x_i^s|y^s)}]^2$$

Table 1: BLEU scores on En→Fr and De→En translation tasks.  $\Delta$  means the improvement over the basic NMT model, which only used bilingual data for training. The basic model for En→Fr is the RNNSearch model (Bahdanau, Cho, and Bengio 2015), and for De→En is a two-layer LSTM model. Note that all the methods for the same task share the same model structure.

System	En→Fr	$\Delta$	De→En	$\Delta$
Basic model	29.92		30.99	
<i>Representative semi-supervised NMT systems</i>				
Shallow fusion-NMT (Gulcehre et al. 2015)	30.03	+0.11	31.08	+0.09
Pseudo-NMT (Sennrich, Haddow, and Birch 2016)	30.40	+0.48	31.76	+0.77
Dual-NMT (He et al. 2016a)	32.06	+2.14	32.05	+1.06
<i>Our dual transfer learning system</i>				
This work	<b>32.85</b>	<b>+2.93</b>	<b>32.35</b>	<b>+1.36</b>

Figure 8:

위의 테이블과 같이, 이 방법은 앞 챕터에서 소개한 기존의 단방향 corpus[Gulcehre et al.2015][Sennrich et al.2016]를 활용한 방식들과 비교하여 훨씬 더 나은 성능의 개선을 보여주었으며, 바로 앞서 소개한 Dual Learning[He et al.2016a]보다도 더

나은 성능을 보여줍니다. 마찬가지로, 불안정하고 비효율적인 강화학습을 사용하지 않고도 더 나은 성능을 보여준 것은 주목할 만한 성과라고 할 수 있습니다.